

Navigation and geolocation within urban and semi-urban environments using low-rate wireless personal area networks

This dissertation is submitted to the University of Hertfordshire in partial fulfilment of the requirements of the degree of MSc by Research.

Thomas David Perrin

October 2016

Acknowledgements

The author would like to thank all of those that have supported him in the pursuit of this thesis. In particular, notable merit is due to the following people and organisations for their steer and direction throughout the project:

- The *Home Office Science: Centre for Applied Science and Technology* for initiating and sponsoring this stream of work and for providing the time and resources required to accomplish it.
- Mr Johann Siau (*University of Hertfordshire*), principle project supervisor.
- Dr Pandellis Kourtesis (*University of Hertfordshire*), Research Tutor and project supervisor.
- Dr Milos Milosavljevic (*University of Hertfordshire*), project supervisor.
- The *University of Hertfordshire* Researcher Development Programme and all of the staff involved for providing the structure and direction needed to complete this course of study.

Abstract

IEEE 802.15.4 defines networks and hardware capable of low power, low data rate transmissions. The use of these networks for the “Internet of Things”, machine to machine communications, energy metering, control and automation etc is increasing. In an urban environment, these networks may well soon become so popular and widespread in their usage that their discoverability and coverage density is sufficient for aiding geolocation – in the same way that IEEE 802.11 WiFi networks are used today. This research shows that although possible, there are some current inherent weaknesses in the use of IEEE 802.15.4 networks for location purposes particularly with respect to multilateration.

Related Publications

Regarding similar principles in WiFi ranging and received signal strengths, the author has jointly authored the following Home Office publications:

HM Government, Home Office Centre for Applied Science and Technology, "*FLAME User Manual*". Publication Number: 99/13, Home Office, 2013.

HM Government, Home Office Centre for Applied Science and Technology, "*FLAME Test Documentation*". Home Office, 2013.

Contents

Acknowledgements	2
Abstract	3
Related Publications.....	4
Contents.....	5
Figures	10
Tables	12
Chapter 1 – Thesis, Introduction and Context	13
1.1. Thesis and Hypothesis.....	14
1.1.1. Thesis statement	14
1.1.2. Hypothesis detail	14
1.2. Introduction	15
1.3. Contribution to Knowledge.....	16
1.4. Ethical Considerations	18
1.5. Why Location, and In What Context?.....	19
1.6. Why Consider IEEE 802.15.4 for Location?.....	22
1.7. Research Questions	24
Chapter 2 – Location Using Low-Rate Wireless Networks	25
2.1. Studying the Field of Location.....	26
2.2. Anticipated Performance and Limitations.....	28
2.3. Location Derivation Methodologies	33
2.3.1.1. Proximity	35
2.3.1.2. Trilateration	36
2.3.1.3. Triangulation	37
2.3.1.4. Scene Analysis	38
2.3.1.5. Other.....	39
2.3.1.5.1. Optical referencing.....	39
2.3.1.5.2. Image recognition	40
2.3.1.5.3. Predictive	40
2.3.1.5.4. Internet connection mapping.....	41
2.4. Radio Range Measurement Techniques.....	42
2.4.1.1. Time of Arrival / Time of Flight	42
2.4.1.2. Time Difference of Arrival.....	43
2.4.1.3. Received Signal Strength.....	43
2.5. Location Derivation Improvements.....	45

2.5.1.1.	Map Matching	45
2.5.1.2.	Particle Filtering	45
2.5.1.3.	Combining Sources.....	47
2.6.	Literature Review Findings.....	49
Chapter 3 – Investigating Data Collection Hardware and Energy Smart Meters		51
3.1.	Introduction to chapter	52
3.2.	Investigating hardware platforms	53
3.2.1.	Purpose	53
3.2.2.	Requirements	53
3.2.3.	Methodology	54
3.2.4.	Results.....	54
3.2.4.1.	IEEE 802.15.4 interfaces	54
3.2.4.2.	Mobile Computer Systems	57
3.2.4.3.	Embedded Operating Systems	58
3.2.4.4.	System on Chip (plus additional circuitry)	59
3.2.4.5.	Microcontroller Development Boards	60
3.2.4.6.	FPGA / CPLD Development Board	61
3.2.5.	Conclusions	62
3.3.	Discovering network transmissions.....	64
3.3.1.	Purpose	64
3.3.2.	Requirements	64
3.3.3.	Methodology	64
3.3.4.	Results.....	66
3.3.5.	Conclusions	68
3.4.	Extracting information for geolocation.....	69
3.4.1.	Purpose	69
3.4.2.	Requirements	69
3.4.3.	Methodology	70
3.4.4.	Results.....	71
3.4.5.	Conclusions	75
3.5.	Findings on the use of smart meters and the RZUSB Stick	77
3.5.1.	Achievements and impact.....	77
3.5.2.	Next steps.....	77
Chapter 4 – Correlation of Received Signal Strength and Range		79
4.1.	Chapter introduction.....	80
4.2.	Understanding RZUSB Stick received signal strengths	81
4.2.1.	Purpose	81
4.2.2.	Requirements	81

4.2.3. Methodology	82
4.2.4. Results.....	86
4.2.5. Conclusions	91
4.3. Measuring received signal strength with range	93
4.3.1. Purpose	93
4.3.2. Requirements	94
4.3.3. Methodology	95
4.3.3.1. Experimental parameters	95
4.3.3.1.1. Elevated height	95
4.3.3.1.2. Surroundings	95
4.3.3.1.3. Measurement samples per increment.....	96
4.3.3.1.4. Delta range step size	96
4.3.3.1.5. Radio module type.....	97
4.3.3.1.6. Antennae gain.....	97
4.3.3.1.7. Antennae orientation.....	98
4.3.3.2. Test setup	98
4.3.4. Results.....	100
4.3.4.1. Data presentation.....	100
4.3.4.2. Relationship between power and distance.....	104
4.3.4.3. Analysing sources of error	108
4.3.4.4. Interpreting the results against the thesis.....	110
4.3.5. Conclusions	112
4.4. Findings relating to IEEE 802.15.4 range determination	115
4.4.1. Achievements and impact.....	115
4.4.2. Next steps.....	116
Chapter 5 – Preparing for Scene Analysis Using Smart Meters	117
5.1. Chapter introduction.....	118
5.2. Database creation.....	119
5.2.1. Purpose	119
5.2.2. Requirements	119
5.2.3. Methodology	120
5.2.4. Results.....	123
5.2.5. Conclusions	125
5.3. Scene analysis findings	127
5.3.1. Achievements and impact.....	127
5.3.2. Next steps.....	127
Chapter 6 – Discussions and Conclusions	128

6.1.	Using IEEE 802.15.4 for Location	129
6.1.1.	Suitable methodologies for geolocation	129
6.1.1.	Practical implications	131
6.2.	Research Limitations	133
6.2.1.	Suitability of methodology	133
6.2.2.	Suitability of equipment.....	134
6.2.3.	Validity of data	134
6.2.4.	Value of research versus peers	135
6.3.	Further Avenues for Research	136
6.3.1.	Including link quality indicator	136
6.3.2.	Alternative logging equipment	136
6.3.3.	Live data capture	136
6.3.1.	Simulated locational testing	137
	References.....	138
	Appendix 1	149
7.1.	Location System Comparison	149
	Appendix 2	152
8.1.	Prototyping platform versus development from scratch	152
	Appendix 3	154
9.1.	Background scan of 2.4 GHz band	154
9.2.	Detection scan of 2.4 GHz band	159
	Appendix 4	164
10.1.	Installing the KillerBee Environment to a Raspberry Pi.....	164
10.1.1.	Console export	164
	Appendix 5	172
11.1.	Installing new firmware to the RZUSB	172
	Appendix 6	176
12.1.	MBed LCP1768 source listing.....	176
12.1.1.	Code listing	176
	Appendix 7	178
12.2.	Adapted KillerBee python scripts	178
12.2.1.	TPStumbler python script	178
12.2.1.1.	Template CSV file	178
12.2.1.2.	Source listing.....	178
12.2.2.	TPRange python script.....	183
12.2.2.1.	Template CSV file	184
12.2.2.2.	Source listing.....	184
12.3.	Multi-channel automation scripts.....	188

12.3.1.1. Source listing.....	188
-------------------------------	-----

Figures

Figure 1: Constandache et al. [7] present this battery life comparison between different location derivation technologies.	22
Figure 2: Example of location derivation by proximity	35
Figure 3: Example of location derivation by trilateration	37
Figure 4: Example of location derivation by triangulation	38
Figure 5: Example of location derivation by scene analysis	39
Figure 6: Example of forcing measured positions to permissible routes.....	45
Figure 7: Example of using particle filtering with map matching (A)	46
Figure 8: Example of using particle filtering with map matching (B)	46
Figure 9: Example of using particle filtering with map matching (C)	47
Figure 10: Example of using particle filtering with map matching (D)	47
Figure 11: WiSpy Channelizer with a 5 dBi 2.4 GHz antenna	66
Figure 12: Background scan of the 2.4 GHz band (aligned with IEEE 802.15.4 decimal channel numbering)	67
Figure 13: Scan of the 2.4 GHz band during network transmission (aligned with IEEE 802.15.4 decimal channel numbering)	67
Figure 14: Linux scripts operating an RZUSB Stick whilst two XBee modules communicate	70
Figure 15: Initial attempt at extracting PAN ID and Network ID; comparing captured data with Moltosenso.....	72
Figure 16: Partially refined data extraction; bytes still represented in little-endian ...	73
Figure 17: Raw data from the simulated wireless personal area network whilst demonstrating scanning channels	73
Figure 18: Sample data from a smart meter	74
Figure 19: Error handling classifies ACK (acknowledgement) message separately to data messages.....	75
Figure 20: RX and TX scripts script in operation during RSSI measurement	83
Figure 21: 1m directional RSSI testing setup	83
Figure 22: Orientation combinations for creating an omni-directional model.....	85
Figure 23: Histogram of the collated mean received power values	87
Figure 24: Histogram of the collated mean received power values split into the three receiver planes	88
Figure 25: Polar plot demonstrating the directionality of the RZUSB Stick.....	88
Figure 26: Received power vs. range relationship model using an omni-directional model of received power at 1 m	90
Figure 27: Test setup to determine the relationship between the received radio power of an IEEE 802.15.4 network and distance.....	98
Figure 28: RZUSB Sticks mounted upon canes to elevate to 1 m above ground.....	99
Figure 29: Orientations of the RZUSB Sticks during very close range measurement (RX in foreground, TX behind)	99
Figure 30: TX node during configuration	99
Figure 31: RX node being set up at 540 cm separation distance	99

Figure 32: Kotanen et al. presented received Bluetooth power vs. range data in a straight forward manner [86]	101
Figure 33: Benkic et al. present their wireless sensor network signal strength vs. range in an abstracted manner [2]	102
Figure 34: Ruiz et al. present their RFID signal strength vs range data in this format [45]	103
Figure 35: Plot showing the basic trend of received power vs. distance and also the spread of measurements at each range.....	104
Figure 36: Received power vs. linear range with error metrics.....	105
Figure 37: Received power vs. logarithmic range with error metrics	105
Figure 38: Comparing the measured results to predicted relationship models.....	107
Figure 39: Standard deviation of received power about the mean vs. distance	108
Figure 40: Assessing links between range and the delta from mean or standard deviation.....	109
Figure 41: Assessing links between standard deviation and delta from mean	110
Figure 42: Histograms of received power at different ranges	111
Figure 43: Histograms of range information separated by received power	112
Figure 44: A large batch of RZUSB Sticks were purchased and reprogrammed with customised firmware	120
Figure 45: RZUSB Stick before (top) and after (bottom) fitting a 50mm, 5-pin x 2-row JTAG header.....	172
Figure 46: Atmel JTAGICE Mk II programmer with an additional 100mm to 50mm header adapter.....	173
Figure 47: Reprogramming an Atmel RZUSB Stick with the KillerBee firmware	174
Figure 48: An off-the-shelf RZUSB Stick (left) and a modified one (right)	174
Figure 49: Checking the device recognition before and after reprogramming with the KillerBee firmware	175

Tables

Table 1: Comparing estimated transmission range of WiFi and smart meters	29
Table 2: Simplistic estimation of ability to attain location fix based upon speed	30
Table 3: Farid et al's comparison of localisation methodologies [29]	34
Table 4: Farid et al's comparison of positioning technologies [29]	49
Table 5: Comparing IEEE 802.15.4 radio modules for use in the investigation of this thesis.....	57
Table 6: Mean power received for each orientation combination	86
Table 7: Transmission characteristics of two RZUSB Sticks at 1 m separation	89
Table 8: Calculated Rx power at sample distances using the omni-directional model of received power at 1 m	91
Table 9: Measurement datasets with mean received power values potentially worthy of re-measurement.....	109
Table 10: Results of the surveying methodologies	123
Table 11: Channel dominance surveying results.....	125
Table 12: Summary channel dominance survey results	125
Table 13: Revised transmission range to a smart meter	130
Table 14: Revised estimation of ability to attain location fix based upon speed	131
Table 15: Comprehensive comparison of indoor location systems as contrasted in summary literature works	151

Chapter 1 – Thesis, Introduction and Context

Chapter Summary

Introducing the thesis behind this research, the potential impact of this research to the field, and the ethical considerations faced prior to carrying out the research.

This chapter also summarises the scope of this research and the reasoning behind the direction taken. Locational references and geographic environments are defined to provide greater clarity throughout the remainder of this dissertation.

1.1. Thesis and Hypothesis

1.1.1. Thesis statement

In future years, energy smart meters and other IEEE 802.15.4 domestic networks will provide a comparable framework for geolocation to that which WiFi does currently.

1.1.2. Hypothesis detail

In the context of an urban or semi-urban environment (defined in section 1.5) it is anticipated that there will be high density coverage of IEEE 802.15.4 networks. With a Government statement about energy smart meters that “most homes should have one by 2020” [1], network coverage should soon be comparable to domestic WiFi installations.

The author hypothesises that the same location derivation methodologies and techniques that are applied to IEEE 802.11 (WiFi) transmissions can equally be applied to IEEE 802.15.4 transmissions. Both systems predominantly operate upon 2.4 GHz frequency bands; they can both support 10-100m transmission ranges; and as argued above, they are both likely to command the same coverage density. Given these similarities, it seems likely that there should be a strong parallel between the capabilities of both systems in the application of location derivation.

1.2. Introduction

This research proposes an emergent mechanism to use transmissions from energy suppliers' smart metering systems to identify or geo-code a location.

As mentioned in the hypothesis, there is a future potential for every residence or commercial property to be supplied with one or several devices transmitting unique identifiers with every communication. This is a new technological opportunity; comparable to when the unique identification of WiFi access points were first catalogued into commercial geo-referenced databases.

The bulk of this dissertation is structured in six chapters as follows:

Chapter 1 – sets out the thesis and the considerations required prior to undertaking the research as well as identifying the field and describing the problem.

Chapter 2 – presents the prior art, literature of the field and the various approaches to the problem.

Chapter 3 – details the work required in preparation for testing and exploring the thesis such as developing the hardware and experimentally defining the available data.

Chapter 4 – covers the first hand investigations and analysis undertaken to explore the use of received signal strength with low-rate wireless personal area networks for deriving range information with which to further derive location.

Chapter 5 – extends upon Chapter 4 and examines the use of smart meters for scene analysis methodologies.

Chapter 6 – discusses the results of this research in context of the thesis and provides a critical review of the work, suggesting scope for future expansion.

Following this are the references cited in the dissertation and the full bibliography of papers and sources used in researching the field.

Finally, there are a number of appendixes which provide important information underpinning this work, but which do not directly add weight to the argument and discussion of the thesis.

1.3. Contribution to Knowledge

As best as can be identified, the approach of utilising energy suppliers' smart meter transmissions to derive the location of a third party device has not been considered in literature before.

With a few exceptions that are in identified Chapter 3, even the constituent aspects of using any IEEE 802.15.4 networks for trilateration, triangulation or scene analysis do not appear to have been investigated for geolocation in the perspective of uncontrolled networks outside of the users' ownership.

This thesis and dissertation uses existing literature from alternative technology fields to consider the entire principle of using IEEE 802.15.4 transmissions for geolocation via any methodology. Several papers have suggested that existing methodologies for radio frequency location are applicable to IEEE 802.15.4 transmissions; however most approach this from the perspective of locating nodes that are associated with the network - i.e. locating the positions of nodes within a single and geographically small network. Alternatively, where researchers have presented a means of locating devices based upon third party networks (i.e. where not connected or communicating directly with the network) then these studies have either not been in the context of IEEE 802.15.4 or have not involved any manner of practical investigation.

The primary research of this study focuses upon the practical use of received signal strength of IEEE 802.15.4 networks for trilateration and scene analysis methodologies. Although not with a thought to the roll out of smart meters, two notable papers [2] and [3] have undertaken this exercise in the past. Their findings were at odds with the theoretical propositions conveyed by other authors as they both concluded that received signal strength trilateration was not a viable option for wireless sensor networks. The author intends through investigation to clarify the position between the only other practical studies of this specific application, and the theoretical standpoints of the researchers of radio frequency location at large.

The author believes that some of the reason for a lack of prior study into this problem stems from the perceived disadvantages of IEEE 802.15.4 versus IEEE 802.11 networks from the perspective of geolocation. Wireless sensor networks are inherently by design lower power, lower data rate and lower frequency of transmission than WiFi networks. Combined, this means that on a mobile system, it is inherently less likely of a transmission occurring within detectable range during the window of opportunity that the target is passing a transmitter. WiFi on the other hand can be considered as constantly transmitting and at a higher power so the likelihood of capturing measurable data within that same window of opportunity is much greater. This makes IEEE 802.11 a more attractive target for investigation even though a focus on WiFi comes at the cost of lower spatial resolution as discussed in section 2.2.

Another difficulty faced by researchers, and perhaps prohibiting greater numbers of practical studies, is that of detecting and recording network traffic on IEEE 802.15.4 networks when not associated with the communicating network. To achieve this specialist IEEE 802.15.4 network analysis tools are required which are not commonly accessible. In the past researchers have had to create their own hardware to achieve this (as is the case of the TelosB Mote, a collaboration between Crossbow Technologies and University of California, Berkley [4]) or obtain one of the few commercial offerings; several of which are no longer available for sale.

Despite the potential failings mentioned above, most of the papers across the field of localisation considered in Chapter 3 acknowledged the benefits of conglomerating multiple measurements and technologies to attain a reliable and accurate location fix. Certainly none presented a counter argument to the effect that using additional data sources is not a worthwhile proposition where battery life, size and cost permit. In this merit the author believes that this study will be able to provide great value in identifying a new and emerging mechanism for geolocation within urban and semi-urban environments.

1.4. Ethical Considerations

There are limited ethical concerns that could relate to this work or the way in which it has been undertaken, and there has been no cause for application to the university board for ethical consideration.

No aspects of this research required third party paid or voluntary involvement. Neither did this research put at risk any intrusion upon the safety or privacy of any person or group.

The data collection undertaken in Chapter 4 was limited in part by the UK Wireless Telegraphy Act 2006 c.36 [5] and by Home Office policy governing the collection of personally identifying data.

Given the high profile sponsorship of this research, approaches such as sniffing third party network communications and wardriving¹ which could be construed by an outside observer as an invasion of privacy have been carefully avoided.

¹ Wardriving: identifying and, for the purposes of formulating a database, geographically tagging networks and telecommunications equipment that are broadcasting uniquely identifiable information over the air.

1.5. Why Location, and In What Context?

Accurate knowledge of the geographical location of an item, electronic device or person is frequently desirable and often critical in numerous applications.

The location of an object can be stated in multiple ways:

- In absolute terms with reference to an origin; for instance a position on a map.
- In relative terms in comparison to another object; such as when describing network architectures.
- Or in relative terms in comparison to a prior position; this may be of most interest when discussing events or changes.

It is possible to infer an absolute position based upon relative positioning if a location is known for the reference object or time.

This study is primarily concerned with position with respect to a map, however it is acknowledged that some crossover exists and that if a location can be determined for another object then location derivation via relative positioning becomes a possibility.

In the United Kingdom there are primarily three origin systems used in mapping on a geographical scale: the World Geodetic System 1984 (WGS84), the European Terrestrial Reference System 1989 (ETRS89) and the National Grid - based upon Ordnance Survey Great Britain 1936 (OSGB36). Whilst it is more than possible to convert between the three systems, it is notable that the Ordnance Survey demonstrate that significant differences of up to 200m in the coordinates of the same latitude and longitude can occur dependant on the reference system used [6]. This means that it is essential to know which system is in use if an accurate depiction of position is desired. It is also possible to define a bespoke origin for a particular application; however this is normally only resorted to within the bounds of a single room or machine. Throughout the papers read, and especially by methodologies used in parallel with global positioning satellites, the normal mode of referring to an absolute geolocation for academic purposes is by latitude and longitude using the WGS84 coordinates.

Throughout this study the terms '**geolocation**' and '**location**' have been used almost interchangeably. More precisely, the term geolocation has been taken at a macro (geographical) scale whereas the unqualified term location also encompasses the micro (local) scale of positional information and knowledge. The nano scale of location (such as the positioning and orientation of mechanical parts within a machine) is beyond the scope of this study.

When discussing location, it is helpful to define the environment within which positional knowledge is important to the intended application. It is also important to

understand to what extent or degree of precision and accuracy that knowledge of location exists and is required. These aspects will shape and define the technologies available for determining location and the efforts to which this needs to be resolved. The range of environments globally is infinite and non-discrete; many locations will share traits of multiple types of environment, however there are four major environments which can be loosely described as follows:

Remote, i.e. a location with very little human influence in the vicinity such as oceans, deserts, rainforests and mountain ranges. These locations are typically geographically large and sparse; end users of positional information may include military or search and rescue personnel, cartographers, scientific surveyors and animal tracking projects. It can readily be imagined that positional precision to within kilometres or hundreds of meters would provide useful and valuable information whereas centimetre or millimetre precision and accuracy is unlikely to be necessary.

Rural, i.e. moderately sparse areas but with regular man-made intrusions upon the surroundings such as agricultural land, road networks and external to single dwellings or out-buildings would fall into this category. It is conceivable that knowledge of the application could be used in such an environment to constrain the limits of possible location and improving the perception of accuracy – such as assuming the user must be upon a mapped road or within a field boundary. Some common and well-funded example usages of positional information in this context are crop spraying and monitoring systems, goods and freight tracking, and forestry commission sensor nodes.

Semi-urban, i.e. well inhabited locations but with regular breaks and only a limited height and density of buildings and structures; such as villages, industrial estates, recreational spaces and university campuses. It would also seem suitable to include indoor locations where the target location would likely be within relatively free-space compared to the infrastructure. Journey navigation systems, employee and lone worker tracking, asset tracking within a factory or warehouse and care for elderly monitoring systems are all applicable applications within this category. It would be common for positional information with precision to a single metre or perhaps even sub-metre precision to be of great value in this environment.

Urban, i.e. a densely populated and dominantly constructed surrounding; inner-cities, underground rail networks and multi-storey shopping centres all suit this environment. Some commonly cited examples for positional interest include virtual tourism guides, notification of local services and augmented reality, targeted and profiled advertising and social location sharing (such as photograph geo-tagging, searching for nearby friends or potential dates, location based games such as man-hunt and geocaching, etc.). With the density of existing infrastructure, greatest number of individual users and potentially highest availability of power this environment provides a rich focus for research, particularly on the micro scale of positional information.

There are many more end users and applications of positional information than have been mentioned above. Additionally most if not all of the examples cited above, and others such as sporting and racing metrics, fleet and insurance tracking, unmanned flight navigation systems or criminal offender tagging are by no means restricted to any one category.

This research shall focus primarily upon determining geolocational information in urban and semi-urban environments.

1.6. Why Consider IEEE 802.15.4 for Location?

An extract reproduced in Figure 1 below, shows the power consumption and battery life differences between assisted satellite positioning, WiFi positioning (labelled as Sky / Skyhook) and displacement positioning mechanisms. These are three possible mechanisms for location derivation commonly used in mobile electronic systems.

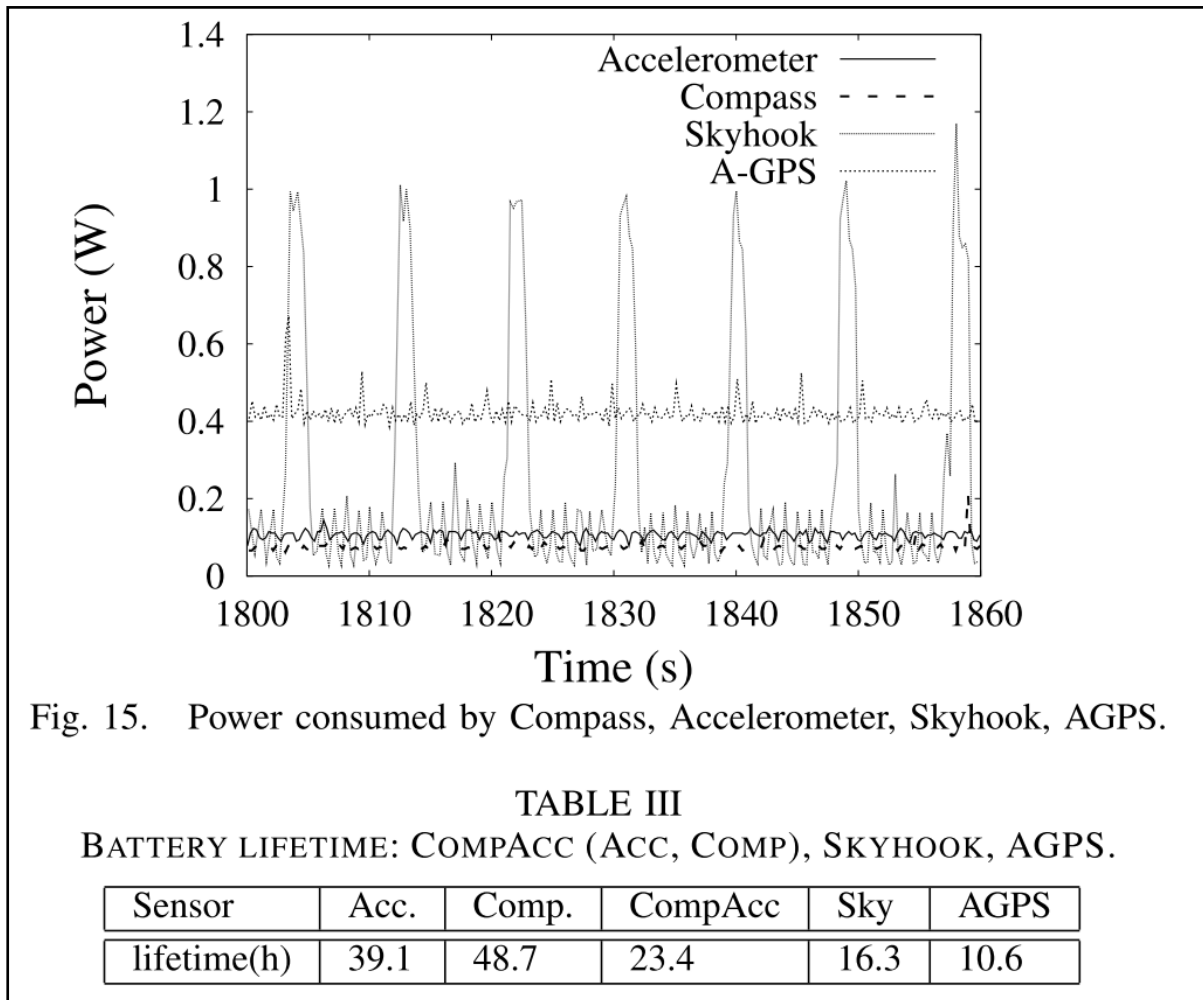


Fig. 15. Power consumed by Compass, Accelerometer, Skyhook, AGPS.

TABLE III
BATTERY LIFETIME: COMPACC (ACC, COMP), SKYHOOK, AGPS.

Sensor	Acc.	Comp.	CompAcc	Sky	AGPS
lifetime(h)	39.1	48.7	23.4	16.3	10.6

Figure 1: Constandache et al. [7] present this battery life comparison between different location derivation technologies.

The paper argues that significant battery life enhancements could be gained by using WiFi or displacement positioning in place of satellite positioning. Constandache et al. also argue that the speed of acquisition is also typically improved but at the cost of a trade off in spatial resolution accuracy.

In recent years there has been much discussion of the “Internet of Things”; a philosophy that there will be a complete and semi-autonomous network of appliances, sensors and machinery. This has been increasingly apparent in the technical news media, professional institutes’ addresses and the peaked interest of

researchers and their publication themes. Although not a new concept, there is a growing awareness for the need of standardisation and governance; generating increased hype as each manufacturer of “connected devices” releases new products utilising potentially incompatible systems.

Currently, IEEE 802.15.4 encompasses the majority of approaches to connected devices and the so called Internet of Things. With the updates and improvements that have evolved with the standards, the ZigBee protocols built upon IEEE 802.15.4 still hold the greatest dominance in the UK. Others such as 6LoWPAN, MiWi and Thread also exist and compete for dominance.

The IEEE 802.15.4 standard [8] is primarily intended for low data rate, mid to low range, low power devices. It is beneficial to embedded applications due to the significantly lower complexity of the networking stacks; the code (and hence memory size) required to operate a network is lowest of all the wireless networking standards. A complexity comparison is depicted most effectively by Gutierrez in his 2005 presentation series to the Computer Science department of the University of California, Berkeley [9].

IEEE 802.15.4 has been adopted by the Government [10], [11] and industry for residential and commercial smart meter use. As a consequence of this, it is anticipated that the battery life considerations of a smart meter location system could outperform those seen in the comparison of assisted satellite, WiFi and displacement positioning systems.

As eluded in the hypothesis statement for this research (see Chapter 1) by the year 2020 there is expected to be a high coverage of smart meters in the UK, approaching one per household and most commercial units also being fitted. This is similar to the way in which each of these properties is likely to possess a wireless router for internet traffic – better perhaps given that many businesses favour fixed line networking and so do not have routers installed at their premises.

As will be seen in the literature review (Chapter 3), IEEE 802.15.4 has not been widely considered before from the perspective of locating a user not connected to a network. Conversely, IEEE 802.11 (WiFi) has been studied in this context in detail by numerous researchers. Given their potential similarities (frequency, transmission range, coverage density, environments etc) it would appear worthwhile also considering the merits of IEEE802.15.4 for location.

1.7. Research Questions

The thesis and hypothesis statement in Chapter 1 shall be probed and explored through the undertaking of a literature review (Chapter 2) and first hand investigation (Chapters 3, 4 and 5).

The author has identified a number of complimentary research questions to direct these studies; the answers to which are intended to help prove or disprove the hypothesis:

- How is the proposed technology likely to compare on a location derivation perspective with other techniques such as global positioning satellites or accelerometers?
- How are IEEE 802.15.4 wireless sensor and smart meter networks likely to compare to other radio frequency alternatives such as IEEE 802.11 (WiFi)?
- Is there an identifiable means for determining location using IEEE 802.15.4 communications?
- Is there a suitable algorithm already available for processing measured data?

Chapter 2 – Location Using Low-Rate Wireless Networks

Chapter Summary

In this chapter the existing literature in the field is presented, including a summary of the major location systems that fit the identified research direction.

Also, different location derivation techniques from the literature are contrasted and some exemplar studies are identified which provide a basis for formulating the primary research methodologies.

To conclude, the research questions identified in Chapter 2 are revisited and a direction is set for the primary research.

2.1. Studying the Field of Location

With reference to the overall thesis, this study is concerned with a new and emerging mechanism for geolocation. In order to ascertain this, it was necessary to scope the breadth of the technical field and conduct a review into the techniques applied by others.

Location and geolocation techniques have been compared and contrasted heavily in literature, either specifically [12]–[20] or as an introductory text to most papers in this field. A considerable problem facing the authors of comparison papers spanning a broad cross-section of location and geolocation techniques is that of the rapidly evolving technology and high levels of interest in new methodologies driving numerous changes and improvements.

There are many well established methods of location determination. Wu et al [21] make a particularly good review of prior art in radio transmission analysis, light level measurement based upon Hill's initial work in 1994 [22] has evidenced popularity in remote areas and inertial dead reckoning systems as covered in depth by Harle [20] are a common option for pedestrians and mobile systems. However in recent years, satellite positioning (GPS / GNSS / Galileo etc) has become the default solution for most geolocation requirements and indeed it is the first method that springs to most people's minds.

In the past decade, and especially within the rapidly evolving realms of portable devices such as mobile phones, harvesting broadcast WiFi signals has also gained in popularity. Making use of existing communications hardware (such as Wifi and Bluetooth circuitry) makes a lot of commercial sense and makes use of existing infrastructure to provide additional services.

Additionally, triangulation of available communication structures (e.g. mobile phone cell towers) is commonly used to locate devices communicating within a network. According to Djuknic and Richton [23], "the driving force behind the development of this technology is a US Federal Communications Commission (FCC) mandate stating that by 1 October 2001 all wireless carriers must provide the geolocation of an emergency 911 caller". This presents an isolated perspective from the United States of America; the statement may be true in part, however there will be numerous other drivers including those touched upon earlier.

In terms of network based geolocation, most researchers have previously focused upon the goal of identifying the location of individual nodes within a network. This has several desirable outputs with many commercial advantages – particularly when dealing with mesh network architectures for sensor arrays or the "Internet of Things" philosophy. Other works have come at the problem from the perspective of locating mobile telephones – again because of the commercial backing and incentives.

Between all of the identified papers that compared and contrasted the different possible technological solutions a total of seventy four different prototype or commercial solutions were presented. Of these greater than fifty were aligned to the goal of locating a device within a geographical zone (some of the solutions were inappropriately listed as they were intended for the purposes of scene reconstruction or cinematography). The appropriate solutions have been agglomerated into a singular table in Appendix 1 (Table 15) with references for both the original source documentation and the paper(s) of collated solutions within which it is contained. Some ratification of cited metrics and additional comments have also been added, making this one of the most comprehensive comparison tables of indoor geolocation solutions to be found in literature.

Many of the more recent papers are now concerned with location in an indoor environment (urban and semi-urban). Of the papers considered from the last couple of years approximately 80% were concerned with indoor localisation versus approximately just 25% of selected relevant sources published prior to 2011. There were some heavy biases in the selection process that may distort these figures; broadly speaking however, these numbers parallel the author's perception of the available literature.

During the literature review undertaken, there appeared to be a distinct lack of research surrounding the ability of a device to locate itself using nearby third party low-rate wireless personal area networks. There were papers discussing how to identify spatial separation and arrangement of nodes within a network; however this is fundamentally quite a different question.

2.2. Anticipated Performance and Limitations

The author believes that the perceived problem of sporadic and infrequent transmissions from wireless sensor networks (discussed in section 1.3) will be alleviated to some degree by the Government's technical specifications for smart meters [10], [24].

The specifications demand a minimum communications interval of ten seconds for mains connected sensors (but targeting updates every five seconds) and every thirty minutes for battery powered gas sensors. Although not equivalent to WiFi which could be deemed in near constant communication, smart meters should present a much more frequent transmission rate than other wireless sensor networks.

One of the potential disadvantages of utilising WiFi detection for deriving location is that the transmission power of wireless routers has been designed to achieve greater than 100 m range of WiFi coverage. This means that upon detection of a service set identifier (SSID) the measurement equipment may be anywhere within a sphere of at least 200 m diameter. To obtain any greater resolution it is necessary to trilaterate between multiple detected routers or utilise some other methodology such as approximating a range from the transmitter based upon the received signal strength (see Chapter 4 for more discussion on these points). In comparison, smart meters transmit over a much smaller range and so with detection of just a single transmitter the location resolution will be less.

The range of a smart meter network is undefined in the IEEE specifications as it is too heavily dependent on the surrounding environment. Factors such as the density and amount of building materials blocking a free line of sight, the directionality of the antennae fitted or weather conditions will make a significant impact upon the detectable range of a smart meter. However a comparison with WiFi is possible by using the stated transmission power limits of both and assuming omni-directional antennae in similar environments (both being within domestic properties of the same manufacture for instance). This comparison has been shown in Table 1 where the following assumptions have been made:

- For 2.4 GHz WiFi the UK radio licensing limit of 20 dBm (100 mW) equivalent isotropically radiated power (EIRP) has been assumed as the transmitter (TX) power. It is assumed that wireless routers capitalise on the full permissible power in order to provide the greatest coverage (and so competitive advantage) possible.
- For smart meters, the transmitter power is much lower, defined in the IEEE standard as targeting -3 dBm (0.5 mW) or less to conserve power [8]. This is approximately two hundred and three times less powerful than WiFi routers.
- As both systems utilise 2.4 GHz hardware receivers it is assumed that the same standards of receiver technology are available to both. Therefore if a

more sensitive receiver than used in the calculation is available this would make an equal affect to both systems. A receiver sensitivity of -95 dBm has been used to represent a moderately priced technology.

- The same non-ideal environment has been assumed for both, with a fade margin of -30 dB chosen to represent signal losses through walls and multipath environments.
- The model for path loss is given in Equation 1 however cable losses and antenna gains have not been used in the calculations as these have been accounted for in the EIRP transmitter powers used.
- The model for range estimation used is displayed in Equation 2. It can be verified by checking that the output approximates that of Equation 4 which is a rearranged form of Equation 3 the Friis Transmission Equation (common place in telecommunication theory for calculating received power), plus the fade margin.

$$[Path Loss] = [Transmitter Power] - [Receiver Sensitivity] + [Transmitter Antenna Gain] + [Receiver Transmitter Gain] - [Cable Losses] + [Fade Margin]$$

Equation 1: Path loss estimation [25]

$$[Distance] = 10^{\left(\frac{[Path Loss] - 32.44 - 20\text{Log}([Frequency])}{20}\right)}$$

Equation 2: Simplistic estimation of transmission distance [25]

$$[Rx power] = [Tx Power] + [Rx Gain] + [Tx Gain] + 20\log\left(\frac{\lambda}{4\pi[Distance]}\right)$$

Equation 3: Friis Transmission Equation

$$[Distance] = \frac{\lambda}{4 \times \pi} \sqrt{\frac{([Rx power] - [Tx Power] - [Fade Margin] - [Rx Gain] - [Tx Gain])}{10}}$$

Equation 4: Transmission distance estimation obtained by rearranging Friis Transmission Equation and factoring in the fade margin

	IEEE 802.11 (WiFi)	IEEE 802.15.4 (Smart meter)
TX power - EIRP (dBm)	20	-3
RX sensitivity (dBm)	-95	-95
Fade margin (dB)	-30	-30
Frequency (MHz)	2.4	2.4
Wavelength (m)	0.121	0.121
Path loss (dB)	85	62
Range estimate (m)	176.9	12.5
Range using Friis Equation (m)	171.1	12.1

Table 1: Comparing estimated transmission range of WiFi and smart meters

From the calculations in Table 1 it is possible to see that based upon the published IEEE standards, smart meters will have a much smaller transmission range – an estimated diameter of 25 m as opposed to about 350 m for WiFi. For location purposes this is significantly better resolution from a single measurement and so methodologies such as proximity detection (discussed later in this chapter) are more applicable.

Assuming location could be derived in some manner based upon a single transmission then a measurement device could travel up to the full extent of a smart meter transmission sphere within the transmission period and still detect the communication.

Table 2 shows the speeds of travel that correlate to being within the transmission range of a smart meter for the entire duration between communication bursts.

With a time between transmissions of a five to ten seconds (or 1,800 seconds for a gas supply node) the cells highlighted in green indicate the distances travelled that are still within the transmission range calculated from Table 1.

The orange cell shows additional valid travel distances based upon a much more simplistic inverse square law model whereby the smart meter transmission range will be halved for every 6.02 dBm increase in transmission power that WiFi has above smart meters. With an equivalent isotropically radiated transmission power delta of 23 dBm this means that smart meters would have approximately an eighth of the range of WiFi routers so about a 44 m sphere.

Red cells in Table 2 indicate distances of travel between communication bursts which exceed the confines of a smart meter transmission range.

Transmission Frequency (S)	5	10	1800	Speed (MPH)	
Distance Travelled (m)	2.2	4.5	804.7	1	
	11.2	22.4	4023.4	5	
	22.4	44.7	8046.7	10	
	33.5	67.1	12070.1	15	
	44.7	89.4	16093.4	20	
	55.9	111.8	20116.8	25	
	67.1	134.1	24140.2	30	
	78.2	156.5	28163.5	35	
	89.4	178.8	32186.9	40	
	100.6	201.2	36210.2	45	
	111.8	223.5	40233.6	50	
	122.9	245.9	44257.0	55	
	134.1	268.2	48280.3	60	
	TX range < 12.5 m radius	145.3	290.6	52303.7	65
	TX range < 22 m radius	156.5	312.9	56327.0	70
	Distance travelled > than TX range	167.6	335.3	60350.4	75

Table 2: Simplistic estimation of ability to attain location fix based upon speed

Table 2 shows that IEEE 802.15.4 networks (and specifically smart meters) could be usefully used in a geolocation sense using a proximity type methodology as long as the measurement device is moving at less than ten to fifteen miles per hour. This means that static agricultural / scientific monitoring or pedestrian use may be possible but that vehicle navigation and autonomous flight type applications are likely to be unsuitable for smart meter geolocation systems.

It would appear that IEEE 802.15.4 networks have the potential, after the rollout of smart meters, to provide an additional means of location derivation. The author anticipated from these pre-investigatory calculations that a smart meter location system may be able to operate with lower power consumption than WiFi (which is already a technological solution with lower power consumption than global positioning satellites). This is as a result of requiring fewer measurements and less processing to achieve a finer resolution of location than WiFi would with the same methodology.

2.3. Location Derivation Methodologies²

Whilst focusing only on radio frequency techniques, Mao et al [26] provide a good depiction of some of the methodologies available. Their paper categorises methodologies into the measurement of distance, angle or time of flight of a radio transmission. This simplified classification does not allow scope for non-radio frequency techniques so this dissertation has adopted Al Nuaimi and Kamel's classifications of Proximity, Trilateration, Triangulation and Scene Analysis [12] in addition to some unique examples that do not readily fall into generalised categories.

Trilateration and triangulation are primarily concerned with calculating a position based upon measurable metrics of known reference locations. This is frequently computed in real-time and irrespective of whether distance, time or angle is measured the output location is predominantly calculated as a distance from other objects. A distinct advantage of these systems is that the methodology is equally effective in previously chartered and unchartered territories.

Some systems take advantage of abstracted measurements to known locations such as using inertial measurement to calculate the distances and directions travelled, otherwise termed as dead reckoning. Harle has contrasted many inertial system approaches and methodologies in his tables [20], however his paper is primarily focused upon body worn systems for pedestrian use and those based around Smartphone technology. Dead reckoning has been discussed in greater detail in the measurement techniques section as this is really a means of maintaining knowledge of a location as opposed to deriving a location from an unknown starting position.

Scene analysis, and to some extent proximity systems, use measurable sensor data to compare against existing tables and databases of readings to determine a location by "best fit". Farahani [27] provides much detail on the "best fit" and database search principles for scene analysis from a radio frequency perspective. A fundamental enabler to this approach is the process of pre-surveying the bounding area once or many times in able to build the reference databases. This takes time, access and resources up front and a live communication link to the databases or alternatively post processing is required to derive the location.

The field of robotics defined the term '**simultaneous location and mapping**' (SLAM) as a means of navigating and operating robotic platforms in new environments. This process removes the need for prior knowledge or survey of the area; in some cases this will save time and effort, but in other circumstances access may not be physically possible beforehand and maps of the area may not exist.

² Map data and imagery used throughout this section is copyright to Google TM 2015 and has been used for academic purposes within the bounds of the published terms and conditions

Whilst appealing, disadvantages to this process include an increased demand for processing power and time as the device is required to build a model of the environment on the fly. This process is not suitable for systems that do not monitor their location at a high sample rate - with intermittent location updates it would not be possible to continuously generate a model that can be cross-referenced with future positions. Sjö et al. present an optical SLAM approach in their paper which highlights the importance and difficulties of location and mapping to the field of robotics [28].

Farid et al [29] present this summary table contrasting different methodologies for deriving a location:

TABLE 1: Comparisons of indoor position methods.

Method	Measurement type	Indoor accuracy	Coverage	Line of sight (LOS)/nonline-of sight (NLOS)	Affected by multipath	Cost	Notes
Proximity	Signal type	Low to high	Good	Both	No	Low	(1) Accuracy can be improved by using additional antenna. However, it will increase the cost. (2) Accuracy is on the order of the size of the cells.
Direction (AoA)	Angle of arrival	Medium	Good (Multipath issues)	LOS	Yes	High	(1) Accuracy depends on the antenna's angular properties. (2) Location of antenna must be specified.
Time (ToA, TDoA)	Time difference of arrival	High	Good (Multipath issues)	LOS	Yes	High	(1) Time synchronization needs. (2) Location of antenna must be specified.
Fingerprinting	Received signal strength	High	Good	Both	No	Medium	(1) Need heavy calibration. (2) Location of antenna is not necessary.
Dead reckoning	Acceleration, velocity	Low to medium	Good	NLOS	Yes	Low	Inaccuracy of the process is cumulative, so the deviation in the position fix grows with time.
Map matching	An algorithm based on algorithms based on projection and pattern recognition	Medium	Medium (indoor) Good (outdoor)	NLOS	Yes	Medium	(1) Map matching purely focus on algorithms and not fully on position methods, coordinate transformation, and geocoding. (2) Using pattern recognition, high computing complex and poor real time issue occur.

Table 3: Farid et al's comparison of localisation methodologies [29]

Whilst a useful comparison, it is the author's view that Farid et al have blurred the distinctions between localisation methodologies, measurement techniques and means of improving results. As per the methodology classifications identified earlier, Proximity and Fingerprinting (aka Scene Analysis) are distinctive methodologies. Whereas Direction, Time and Dead reckoning from Table 3 are measurement techniques applied to the Trilateration and Triangulation methodologies. Map matching in Table 3 is a means of improving the accuracy and relevance of a location fix.

Each of the major methodology classifications identified (Proximity, Trilateration, Triangulation, Scene Analysis and Others) has been detailed by the aid of example in their respective subsections below. Measurement techniques and means of improving location fixes have been discussed in subsequent sections. All

technological solutions from GPS to WiFi to the measurement of times dawn and dusk can be categorised into the classifications identified (in these examples: trilateration via time difference of arrival for GPS, proximity for early implementations of WiFi location assistance, and scene analysis for Hill’s elephant seal tracking [22], [30]).

2.3.1.1. Proximity

Proximity detection is one of the least sophisticated forms of location derivation. By detecting that one is within the transmission or sensory ranges of a unique object it is possible to state a position with a degree of precision. This precision is at worst the maximum distance at which this object can be detected, however positional knowledge is improved if the range is non-uniform and the pattern of transmission is known or can be calculated. This can often be the case in radio frequency systems where directional antennae may be used or attenuators such as walls are present.

This methodology has been exemplified in Figure 2 below in terms of radio frequency transmission however it is also applicable to other technologies. Ultrasonic [31] and infrared beacons [32] have been used in this manner in place of radio transmission, however the latter (e.g. [33]–[35]) is more popular especially given infrastructure such as mobile cell towers and WiFi hotspots are already in place [19].

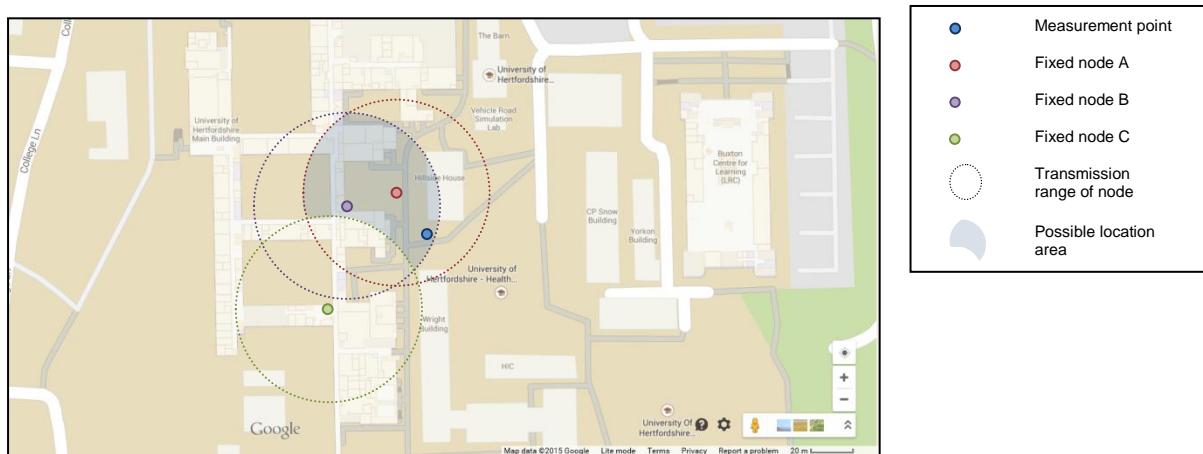


Figure 2: Example of location derivation by proximity

In Figure 2, nodes A, B and C are fixed radio transmitters. Each of the nodes have identical omni-directional radiation patterns as depicted by the dotted rings. At the measurement point denoted by the blue dot near the centre of the map, the only nodes within range are nodes A and B. Node C is outside the detectable range. With no further information, it is only possible to state that the measurement point is near to nodes A and B. In this example however, the map position of the nodes are also known so it is possible to ascertain that the measurement point must be within the blue shaded area shown. This is the only location where the transmission ranges of nodes A and B overlap but do not coincide with node C.

The author and others at the Home Office Centre for Applied Science and Technology have previously trialled this approach on a geographical scale using

different types of radio broadcast equipment [25], [36]. Numerous transmitters were attributed with unique identifiers and associated with a database of longitudes and latitudes. By cross-referencing detections with the database, it was evidenced that it is possible to report a useful location when in the proximity of a transmitter. Despite significant error handling, the Home Office detailed three cases of possible false-positive locations using this methodology [36]:

- The detected transmitter may have moved since the database was collated.
- Inadvertent duplicates may exist with regards to the unique identifications.
- Or the received identifier data may have been corrupted but coincidentally matched a true record in the database.

Using this methodology, and assuming continuous sampling, the only scope for precision enhancement is to increase the density of measurable nodes and / or decrease the detectable range of the objects. In a large area, or where finer detail is required, the infrastructure requirements of fine resolution location can be cost prohibitive and the resources required to maintain a database of locations may be excessive.

In contrast however, the computational load of the measurement system is can be very low which can be a significant advantage in terms of battery life and physical size requirements of a host device. The computational load in this methodology is proportional to the sample rate and duration of measurement, both variables which can improve performance at the cost of resources [36].

Operating in reverse, it is possible for the nodes (as in Figure 2) to detect and notify when the target equipment is within range of itself [19]. This is the mechanism for serving cell identification within mobile telecoms. It is possible by collating the details of the serving cell and neighbour cells to identify the shaded region shown in Figure 2 whilst performing all the computation on a remote system, maintaining “dumb” target equipment with superior battery life.

A proposal by Wu et al [21] uses the unique attenuation characteristics of 2.4GHz WiFi signals through walls to determine loss of proximity with a wireless access point. This novel approach is computationally inefficient however as continuous (or at least frequent relative to speed of motion) measurement is necessary in order to detect the threshold change in received signal strength.

2.3.1.2. Trilateration

Trilateration expands upon the proximity methodology by utilising range information. As with previously, this methodology can apply equally well to other technologies (e.g. the Dolphin [37], Cricket [38] and Active Bats [39] ultrasonic systems or the Active Badge infrared system [40]) however trilateration is frequently used in radio location and available metrics such as received signal strength make this especially convenient.

Trilateration and more generally multilateration are described by the radio frequency example depicted in Figure 3. As before nodes A, B and C are fixed radio transmitters. In this instance the distances from the measurement point are ascertained and indicated by their corresponding dotted rings.

Where two nodes are physically separated by less than the sum of their respective distances from the measurement point (as depicted for instance by the green and purple dotted rings) then there will be two possible points at which the measurement node must have been to obtain the measured values. To obtain a singular position three overlapping nodes must be measured as shown by the addition of the third node in Figure 3, hence the term trilateration. More than three points provides an advantage in terms of accuracy and precision.

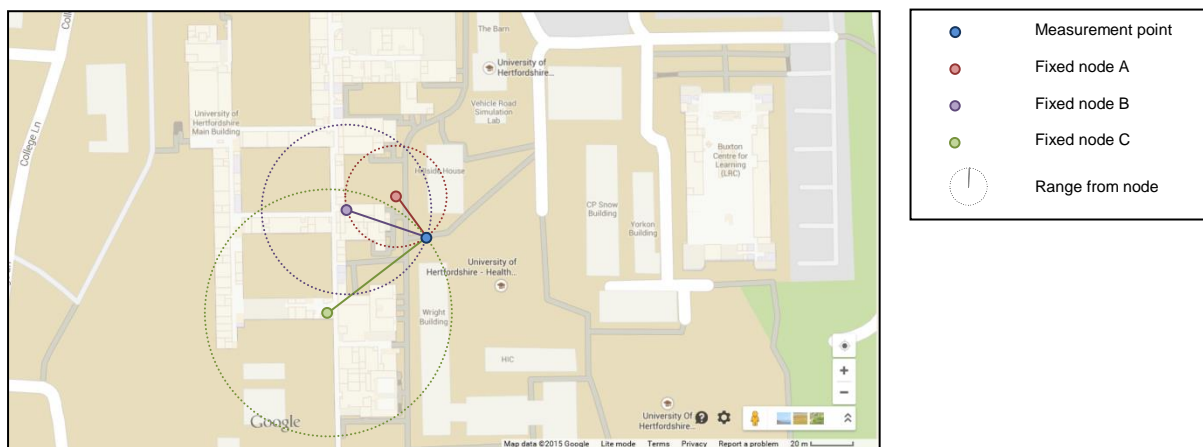


Figure 3: Example of location derivation by trilateration

Trilateration and multilateration are entirely dependent on the accuracy of ranging measurements obtained between the measurement point and the multiple references. The different techniques for measuring range may involve measuring the timing or strength of broadcast signals as are discussed in greater detail in section 2.4; or it may more literally be a measure of displacement through odometers or accelerometers. The latter is more popular with pedestrian systems as the regular pattern of footfall is a detectable metric for estimating distance with low cost inertial units [20], [41]–[49].

In radio frequency based location systems trilateration is a frequently used methodology due to the ease of obtaining a figure for received signal strength which is inversely proportional to range. RADAR [50] and COMPASS [51] are two of the older and more frequently cited / developed systems in this research space, however there are numerous others such as [52]–[60].

2.3.1.3. Triangulation

With triangulation, one measures the bearing to multiple non-parallel reference points and takes the point at which the bearings conform as the position from which the measurement was taken (see Figure 4). With radio frequency solutions this is termed as measuring the angle of arrival of target signals.

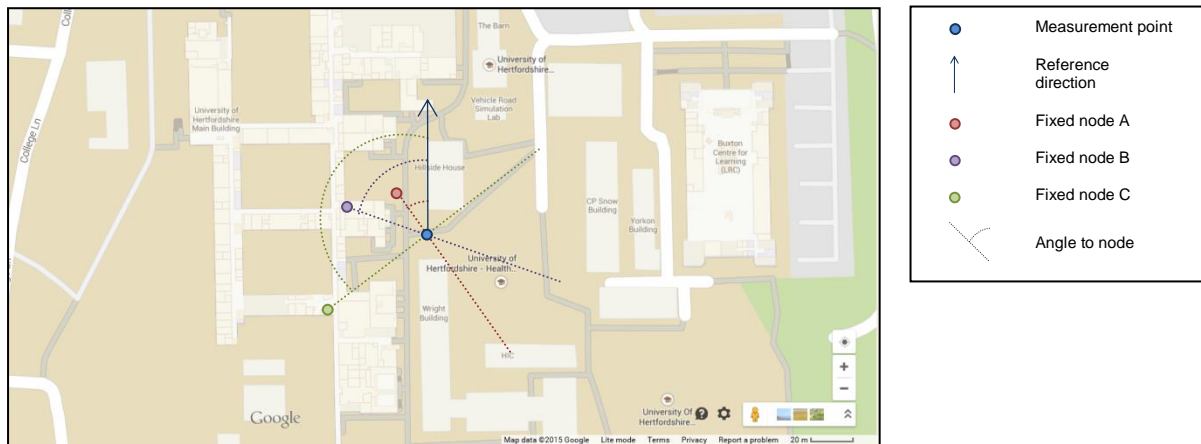


Figure 4: Example of location derivation by triangulation

This can be performed with just two linear bearings, however as directional measurement of any signal is prone to error it is commonplace to utilise three or more hence the term triangulation. With error margins accounted for, three bearings won't converge on the same exact point but will provide a small triangle within which the measured position lies.

In general terms measuring the angle of a signal (regardless of whether this were light, radio waves, sound etc) is more complex than measuring intensity as it requires a directional sensor or a sensor array and polar measurements.

Aitenbichler and Muhlhauser present an example of triangulation of infrared beacons detected by a stereo video system [61] whereas Yhang et al. have shown a radio frequency identity tag application of triangulation utilising directional antennae [62].

A naturally directional sensor would be the measurement of changes in magnetic flux via a Hall Effect sensor hence why this technique is popular with digital compasses. Given small physical size, the low cost and minimal complexity of accurate digital compasses they are widely added to portable electronic equipment such as smart phones and tablet computers. Due to this, triangulation is often used as an add-on methodology for improving the results of multilateration techniques such as in two radio frequency plus compass solutions Sapphire Dart and Ubisense [63], [64].

2.3.1.4. Scene Analysis

Scene analysis is performed by pre-sampling the geospatial zone of interested and creating a database of values with positions. Regardless of the entity being measured (light levels [30], electromagnetic flux density - covered in detail by Gu et al. [16], radio frequency samples [58], [65]–[67], or GSM signal measurements [68] etc) numerous premeasured positions are required. When attempting to locate the device of interest an instantaneous measurement sample is compared against the database of pre-recorded results to find either the closest fit or an interpolated position based upon similar data.

Figure 5 shows an analogy of using scene analysis with received signal strengths from multiple radio frequency beacons. Each grey dot on the map represents a

transmitting beacon; the rings represent the current transmission strengths / range limits of each beacon. Each yellow dot indicates a sample position that has previously been measured and recorded.

In the scenario depicted, the measurement point is able to detect weak signals from each of nodes A, B, and C and so can correlate these signal strengths with a lookup table of past measurement samples. The large shaded region indicates the geolocation area that would be returned based upon a binary proximity analysis. In contrast, the four yellow markers with red outlines identify the previous measurements with most similar likeness to the current data – the position is either calculated as the centroid of these points or as a weighted interpolation. As can be seen from Figure 5, the scene analysis approach can yield much more accurate positioning results but is affected by the separation and accuracy of previous sample data.

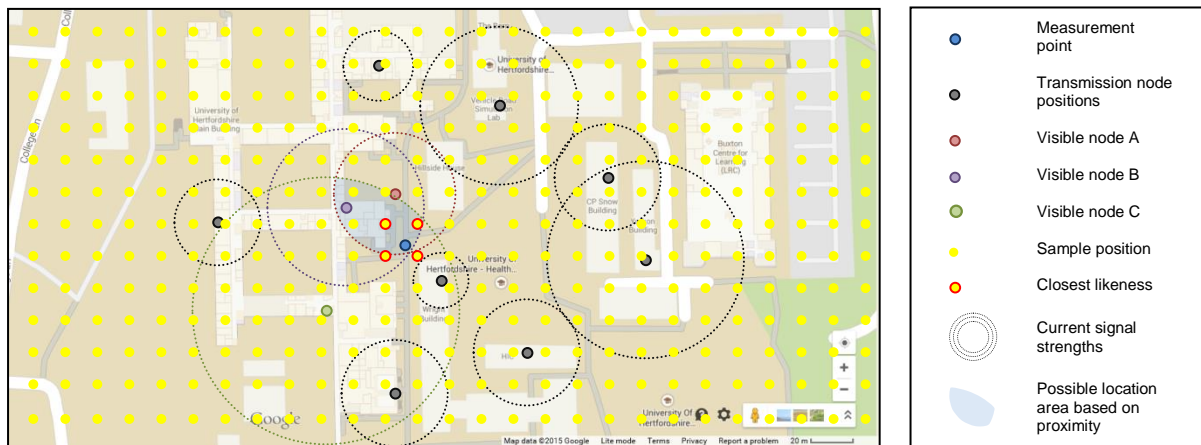


Figure 5: Example of location derivation by scene analysis

Due to the heavy reliance on historic data, this approach is not well suited to fluctuating measurement data or to continuously changing environments – for instance recording measurements of mobile transmitters into a database is of little value if their positions will have altered when the intended location exercise takes place.

2.3.1.5. Other

Besides the different combinations of those methodologies listed above, there are some additional alternative methods for deriving location in particular circumstances.

2.3.1.5.1. Optical referencing

Tilch and Mautz presented a “Camera and Laser based Indoor Positioning System” (CLIPS) [69] based upon their prior research and prototype [70]. A “reference map” was generated on the fly by projecting a calibrated laser pattern onto the environment, which allowed for optical comparison of the target’s view point within a room versus the projected pattern using stereo photogrammetric techniques to determine the relative location of the view point. A similar projected grid navigation

system is employed by the iRobot® house cleaning robot Braava and its accessory projector the NorthStar® Navigation Cube [71].

Tilch and Mautz's survey of optical systems [17] contrasts a large number of optical location systems and evidences the high levels of accuracies attainable in a comparative table. Many of the systems and proposals they included however were not location systems in the context of this study; they were instead developed for tasks such as optical inspection and measurement of physical objects or providing interactive information boards. As a result, some of the stated accuracies and the represented quantity of optical geolocation systems in their paper are misleading when compared to other technological approaches.

2.3.1.5.2. *Image recognition*

Urban street furniture and building designs could be captured by a camera and the images compared to a mapped image database such as Google StreetView™. Using image recognition tools and learning algorithms the strongest matches could be identified and subsequently filtered to provide a single match per predefined geospatial area. This could provide potential for geolocating where an image has been taken or for locating a device with a camera attached.

Clearly this concept would require access to a database of geocoded images such as the Google StreetView™ imagery. The physical possibility of this has been shown by Google's own research paper on automatic house number identification - where 600,000 images from StreetView™ have been fed through an image recognition system [72].

Other image recognition systems have been created for using in simultaneous location and mapping applications whereby particular features are recognised and used as a point of future reference [73]–[75]. Similarly, there are several systems using this technique by with easily recognised placed markers, particularly where these can be highlighted by the use of an infrared illuminator [76]–[78].

2.3.1.5.3. *Predictive*

Ashbrook and Starner [79], [80] and Scellato et al. [81] present the case for predictive location and tracking solutions. By monitoring lifecycle patterns of equipment, animals or people over large periods of time it is possible to accurately predict the next locations based upon the confirmative occurrence of other events.

For instance, if a person commutes to work by bus every morning it would be possible to assume their location based upon knowledge the bus timetable and confirmation that the person in question did get up and left the house for work at their normal time.

Vehicle navigation systems frequently assume that the driver will continue to follow the planned route at either the current or permissible speed. In the event of satellite coverage dips (such as passing beneath bridges) the system will continue to predict

the user's location despite a loss of current data. If the user deviates course during this period it can lead to significant error but otherwise to this the system can quite accurately position the user until sufficient satellite coverage is resumed.

2.3.1.5.4. *Internet connection mapping*

For internet connected devices, a possible alternative may involve the timing of messages and responses (also known as pings) to and from multiple static network addresses at known, fixed geographical locations. It is assumed that to observe any significant readings a large number (in the order of hundreds) of known target network addresses would be required and some potentially complex algorithmic techniques used to nullify the effect of multiple path communications. A conceivable commercial use for this process is in approximating the country of origin of online shoppers when faced with an increasing use of IP proxy servers. It is understood following a conference attended by the author that the approach is currently being trialled by the Dutch.

2.4. Radio Range Measurement Techniques

As will be seen in Chapter 3, the available hardware for testing did not lend itself to a triangulation approach. However, the three remaining major methodologies shall be investigated for use with IEEE 802.15.4 networks.

For proximity detection a binary existence or not of a unique network is required, scene analysis increases the measurement complexity by requiring a measurement of signal strength. However to measure distances for undertaking a multilateration approach there are a number of different methods presented in literature, the main of which are listed here:

2.4.1.1. Time of Arrival / Time of Flight

The time taken for a radio frequency transmission to pass from transmitter to receiver is directly proportional to the distance between them; measuring the time of flight thus enables the calculation of distance.

The most recognisable application of time of arrival is with GPS; very accurate atomic clocks continuously stream timing and self-position data from space to earth. Receivers on earth are sufficiently distant as to be able to measure the difference between the time the signal was received and the timestamp of the epoch sent - so measuring the time of flight. This system depends upon multiple clock synchronisation to perform multilateration. Generally, whilst the principle is simple (and used successfully with other radio frequency systems [53], [66], [82]) the implementation for trilateration is somewhat harder than other methods. This is due to the timing accuracies required to differentiate distances between two signals, both of which are travelling at speeds near the speed of light and especially where the distances are small.

An alternative two-way communications version of time of arrival exists whereby a signal is transmitted by the measuring device a nearby wireless node. As soon as the node receives the signal a return signal is sent back to the measurement device which measures the time period between sending the first message and receiving a response. This time period consists of twice the time of flight plus a constant amount taken by processing. This system does not require such accurate clocks in the wireless network and so would be closer aligned to smart meter measurements – unfortunately implementation with smart meters is still unlikely to be possible as the processing delay in response to a message will be uncontrolled and hence variable, accounting for large uncertainties in the distances calculated.

Time of arrival range measurement can apply to other mediums such as ultrasonic [39], audible tones [83], surface acoustic waves [82] and infrared transmissions [40]; however these are of lesser to this study which focuses upon the application of use with energy smart meter systems.

Güvenç and Chong present a concise table outlining the differences and accuracies of various algorithms for time of arrival systems [84].

2.4.1.2. Time Difference of Arrival

Time difference of arrival is a popular approach requiring less accurate clocks than time of arrival systems. At least two transmissions from the same source but via different wavelengths are simultaneously transmitted. The receiving node measures the time delay between receiving one frequency (e.g. radio waves) and the second (e.g. sound waves). Given both signals have travelled the same distance; the timing difference between them is proportional to the distance travelled and the velocities of the signals used.

Time difference of arrival can also be achieved by measuring the differences in arrival time of signals detected by multiple synchronised receivers.

Although used by multiple geolocational systems [37], [38], [63], [64], [85], time difference of arrival is not applicable to a smart meter based location system as there are multiple transmitters and a singular receiver, but only a single frequency is broadcast by each smart meter.

2.4.1.3. Received Signal Strength

The most applicable radio frequency range metric to smart meters (and arguably the most frequently studied) is that of correlating signal strength losses to range. Most of those systems cited in the triangulation subsection employ received signal strength to achieve the methodology [44], [50]–[60].

Assuming known (and fixed) transmitter / receiver gains, it is possible to measure received signal strengths with most wireless communication systems in order to calculate the drop in signal strength caused by the transition. In section 2.2 several equations were shown linking path losses to distance, several researchers (such as Bahl et al. [50]) have progressed these equations to account for different environments and systems.

Kotanen et al [86] present a Bluetooth Local Positioning Application (BLPA) based upon trilateration of Bluetooth communications. They use the received signal strength indication (RSSI) to determine the range from each fixed node. In their paper they present a mean error of position of 3.76m but continue to conclude that the system requires an alternative means to determine the range.

According to Kotanen et al, the Bluetooth RSSI figures were not directly correlated to received power and as such is “defined too loosely for positioning purposes” [86]. It is notable however that when measurements were taken to relate distance to a derived received power level they were undertaken in an undesirable environment - a moderately uncontrolled office without background references nor discussion of antenna directionality. This will have led to a poor distance model that may have adversely impacted their conclusions. Nonetheless, with simulated results, their

approach evidenced success and their experimental design is worthy of some note if similar were to be undertaken with wireless sensor networks.

Chintalapudi et al contrast WiFi received signal strengths obtained by seven different smartphone handsets [87]. They show a massive 20 dBm difference strongly arguing the need for accommodating receiver gain differences when performing ranging with received signal strength indications.

2.5. Location Derivation Improvements³

The following subsections briefly outline some techniques that can be applied to any of the methodologies outlined in section 2.3.

2.5.1.1. Map Matching

Perhaps the simplest improvement, utilised by multiple in-car navigation systems, is to take assumptions about the possible locations that a device might possibly be with relation to the known surroundings. In vehicle navigation it is frequently assumed that the vehicle will always be on a road and so systematic positional errors and low geospatial resolution can be combated by forcing the user's position to the nearest available road as depicted in Figure 6 below.

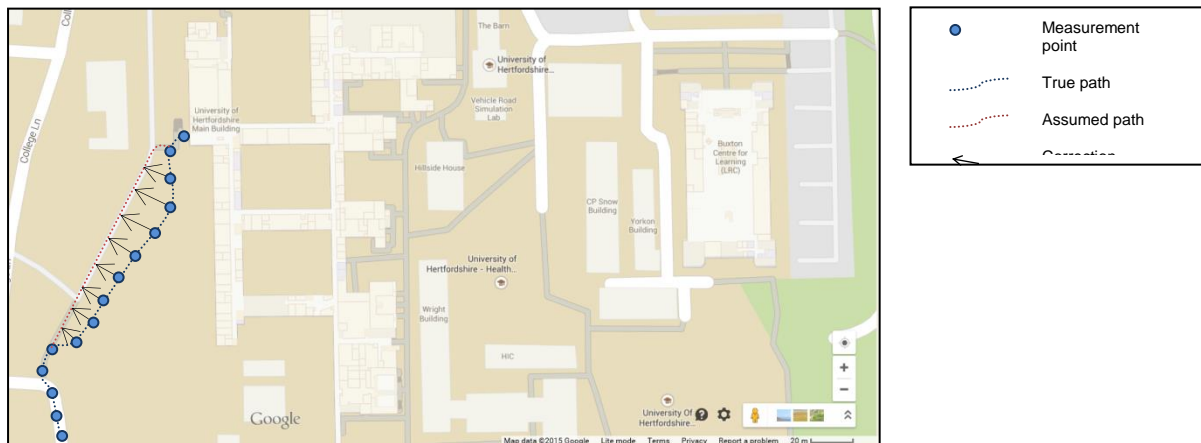


Figure 6: Example of forcing measured positions to permissible routes

The use of map matching techniques has been discussed in a few papers, most dominantly those relating to location and navigation within confined areas such as offices and shopping centres [20], [48], [88]–[90].

2.5.1.2. Particle Filtering

Using iterative techniques such as Recursive Bayesian Estimation and Kalman Filtering it is possible to refine a position over time. By using a very high number of particles to represent all the possible spatial positions within the target zone and then applying measured data (such as displacement values or received beacon signals) to every particle it is possible to rule out and destroy those particles which have resulted in impossible movement or positions. Iteratively performing this function

³ Map data and imagery used throughout this section is copyright to Google™ 2015 and has been used for academic purposes within the bounds of the published terms and conditions

with the application of probability statistics results in an ever narrowing particle field representing the possible locations of the target.

Figure 7 through to Figure 10 illustrates how with knowledge only of the vector of travel between snapshots it is possible to continually narrow the possible number of locations. Once a singular possible position is reached displacement tracking can continue and additionally the only possible path to the current position can be stated.

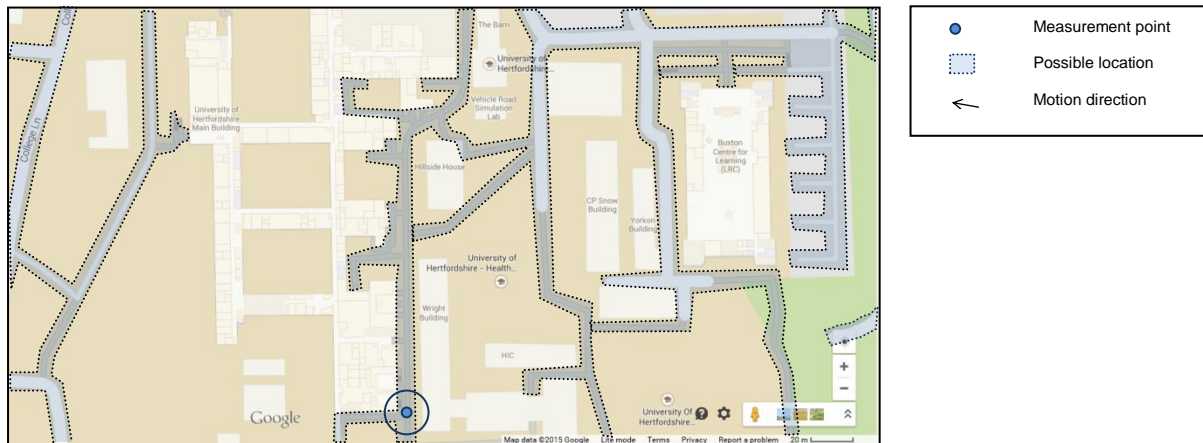


Figure 7: Example of using particle filtering with map matching (A)

(A) Initially the position cannot be known without an external input but it is assumed in this case for simplification purposes that the target device must be upon a road.

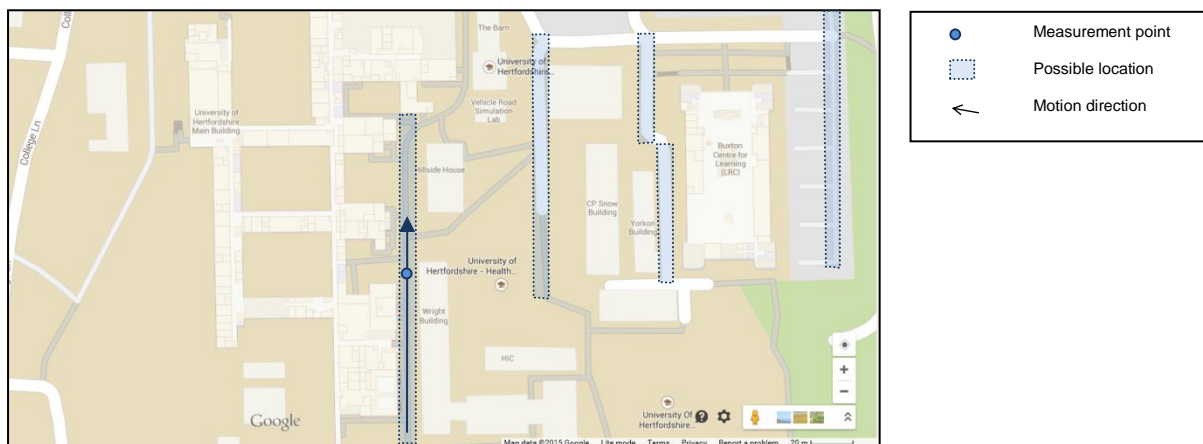


Figure 8: Example of using particle filtering with map matching (B)

(B) After movement or repeated measurement it is possible to narrow down the possible locations – in this case the only possible locations at which a displacement of a magnitude and direction indicated by the arrow in Figure 8 are those shaded. This represents a significant reduction already.

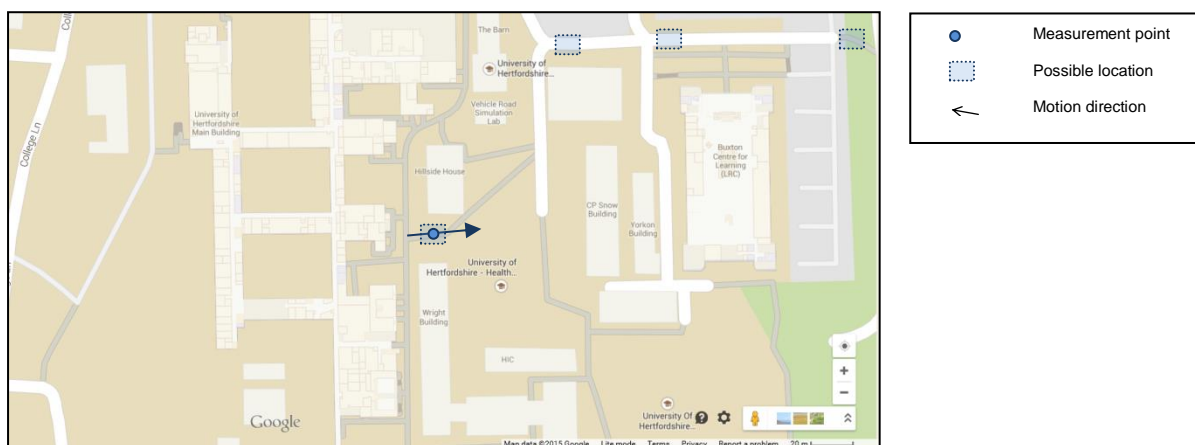


Figure 9: Example of using particle filtering with map matching (C)

(C) After iterative analysis the possible locations become fewer and more defined – there are only a handful of locations in this case at which a small displacement, following a right-hand turn, proceeded by a much larger northerly displacement are possible.

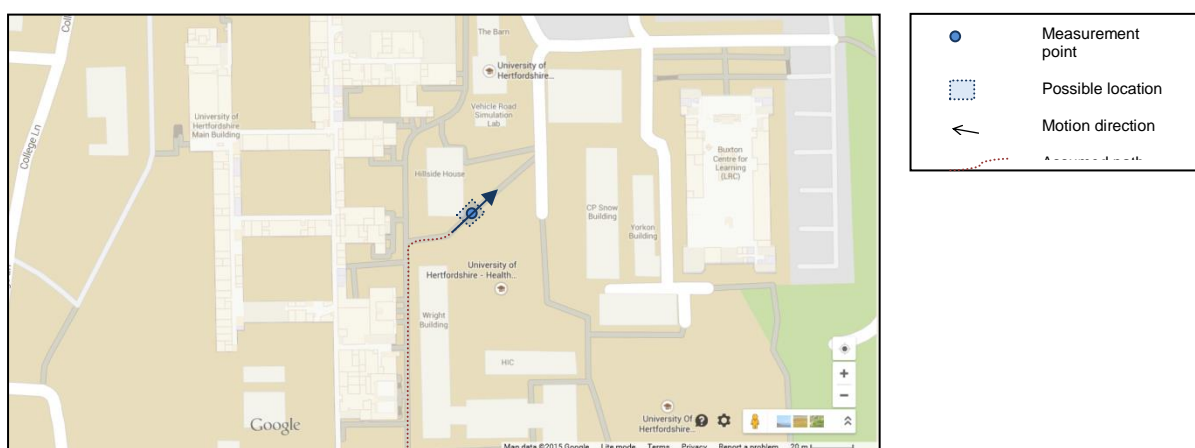


Figure 10: Example of using particle filtering with map matching (D)

(D) Finally, when the position is known and no other positions are possible based upon the sequence of events / measurements, historic location prediction of the path taken is possible.

The use of particle filtering is particularly powerful in improving the reliability of results at the cost of additional computing resources. Several researchers [7], [12]–[14], [20], [26], [41]–[47], [75], [91]–[95] have covered the use of such approaches in their papers; all with favourable comment.

2.5.1.3. Combining Sources

The agglomeration of multiple input sources to a geolocation service can provide distinct advantages in terms of accuracy, reliability and robustness. The use of GPS and WiFi data within smartphones has increased the speed and accuracy of geolocation and provides for situations where a view of the sky is not possible – such as indoors [96]. By cross-referencing multiple inputs, an in-vehicle data logger

designed for recording professional racing metrics can typically have a combination of GPS, CAN bus and accelerometer measurements to provide an incredibly accurate record of the vehicle's location, vector and acceleration [97]. This provides for a possible precision greater than achievable by GPS alone and also additional reliability when passing under objects such as gantries and bridges where the satellite signals can be wholly lost or reduced in number.

The COMPASS positioning system is an excellent example of corroborative combination of multiple sensors –a WiFi antenna and a digital compass. The team used a scene analysis methodology based upon measuring the received signal strengths of WiFi communications. They enhanced the base methodology by filtering the database of scene measurement recordings based upon the orientation of the measurement device. This meant that falsely low received signal strengths are negated where they were caused by the blocked line of sight path of where the operator is standing. Compared to the earlier radio frequency scene analysis system (RADAR [50]) King et al. show that their COMPASS system achieved significantly better results because of the combined sensors [51].

2.6. Literature Review Findings

In terms of electronic positioning and location technologies, Farid et al. presented the best overall summary of the differences. This has been reproduced below in Table 4 and shows that there is no overall clear leader in terms of which technology is “best”. Some systems are better for financial cost or portability, others better in terms of accuracy or coverage, and almost all bar GPS are suitable for indoor environments.

Although Faird et al. mention ZigBee in the table, it is worth noting that this is in the context of inter-network location and positioning – i.e. discovering the position and ranges of other associated nodes. Unfortunately this is not in the same context as smart meters, to which the target geolocation device will act as a third party to the smart meter network. Additionally, it is notable that the remarks against ZigBee are largely negative.

TABLE 3: Comparison of common position systems used for localization.

System	Accuracy	Principles used for localization	Coverage	Power consumption	Cost	Remarks
GPS	6 m–10 m	ToA	Good outdoor Poor indoor	Very high	High	(1) Satellite based Positioning. (2) Processing time and computation is slow.
Infrared	1 m–2 m	Proximity, ToA	Good Indoor	Low	Medium	(1) Short range detection. (2) No invasion of multipath.
WiFi	1 m–5 m	Proximity, ToA, TDoA, RSSI Fingerprinting, and RSSI theoretical propagation model	Building level (outdoor/indoor)	High	Low	(1) Infrastructure available everywhere. (2) Initial deployment is expensive. (3) Multipath susceptible slightly.
Ultrasound	3 cm–1 m	ToA, AoA	Indoor	Low	Medium	(1) Sensitive to environmental. (2) No invasion of multipath.
RFID	1–2 m	Proximity, TOA, RSSI theoretical propagation model	Indoor	Low	Low	(1) Real time location system. (2) Response time is high. (3) Manual programming.
Bluetooth	2 m–5 m	RSSI fingerprinting and RSSI theoretical propagation model	Indoor	Low	High	(1) Data transfer speed is high. (2) Limitation in mobility.
ZigBee	3 m–5 m	RSSI fingerprinting and RSSI theoretical propagation model	Indoor	Low	Low	(1) Low data transmission rate. (2) Nodes are mostly asleep.
FM	2 m–4 m	RSSI fingerprinting	Indoor	Low	Low	(1) Less susceptible to objects. (2) Signal is strong; due to this, it covers large areas.

cm: centimeters; m: meters.

Table 4: Farid et al’s comparison of positioning technologies [29]

Through a study of the literature the author has identified many potential means to derive a location from transmitted smart meter signals:

- Using a binary measurement of proximity to a smart meter.
- Using multilateration to multiple smart meter networks measuring the range via either time of arrival techniques or more likely receive signal strength correlation. This would appear without testing to present the most favourable approach, especially as it is the most popular technique for WiFi and Bluetooth systems which are closely related in terms of hardware.
- Using a scene analysis (fingerprinting) approach.

All three possibilities require pre-surveying of the geographical target zone (wardriving) however this is most resource demanding when applying a scene analysis approach. This study will attempt to further investigate the likelihoods and possibilities of undertaking all three of these approaches.

This literature review identified the work of Benkic et al. [2], Kotanen et al. [86] and Ruiz et al. [45] to be the most closely related or worthy of consideration when undertaking received signal strength versus range experiments for the purposes of multilateration. That said, Bahl et al. [50] present three models for receive signal strength positioning in order to attempt to account for rooms and multipath environments.

The next chapter shall be by undertaking an assessment of the possible hardware choices and the information available from smart meters. Some possible leads were identified with regards to suitable hardware:

- Benkic et al. [2] showed three IEEE 802.15.4 radio modules that were deemed inadequate for the task of received signal strength trilateration.
- Speers et al. [98] identified an Atmel radio module for IEEE 802.15.4 which they had provided some custom firmware and python scripts with which to undertake low level stack control and investigate ZigBee security.
- And several researchers [3], [43], [98]–[100] identified the Telos B-Mote (a research community development [4]) as applicable hardware for these type of investigations.

Chapter 3 – Investigating Data Collection Hardware and Energy Smart Meters

Chapter Summary

This chapter encompasses the tests and investigations undertaken to identify suitable means of capturing relevant data from smart meters.

Subsequently, a brief investigation into some of the real world practicalities associated with deriving location with IEEE 802.15.4 networks is presented.

3.1. Introduction to chapter

This chapter prepares for the primary research (testing based data collection) undertaken by the author in pursuance of this thesis.

The chapter is split into three parts: choosing a hardware platform, investigating the radio profile of IEEE 802.15.4 networks, and finally determining the useful data transmitted by a smart meter. This showcases a practical progression in identifying a means by which to carry out the research studies centred about the thesis in section 1.1.1.

Further detail has been provided the appendices surrounding the alteration of the firmware on board the chosen hardware platform, and the use and creation of python scripts for performing network monitoring tasks. These have not been included in the main text of this dissertation due to their deviation from the thesis questions.

3.2. Investigating hardware platforms

3.2.1. Purpose

In order to successfully and readily collect any meaningful primary data to support the proposed thesis it was necessary to possess some minimum equipment:

- an IEEE 802.15.4 compatible radio module (sensor)
- a computing platform for commanding the radio module
- a means of logging or displaying received data
- a means for transforming primary data into useful information
- a secondary means of location

A number of different hardware enablers were trialled to determine the most suitable setup for this research. By trying several systems it was possible to identify the most adaptable and easy to use (given the author's prior skill set) such that a greater proportion of time could be spent focusing on the research questions.

Given the rapidly expanding industry for the Internet of Things, IEEE 802.15.4 networks and computing platforms it would be quite possible to devote large amounts of time researching this area.

3.2.2. Requirements

To allow sufficient time to investigate the core thesis, a one month period of part time studies (approximately 40-50 hours) was allocated to choosing a platform and learning how to establish a network and communicate with other nodes.

To provide assurance that a workable solution could be found a diverse mix of hardware was needed. At least five radio modules and five significantly different computing platforms were trialled.

To preserve time and concentrate effort, minimal hardware design and manufacture was key and any software programming needed to be that utilising syntax familiar to the author. A fuller consideration of using commercial-off-the-shelf versus a bespoke testing platform is provided in Appendix 2.

This hardware platform investigation was required to identify a solution for a test system that should:

- be capable of two way communications via the IEEE 802.15.4 protocols
- provide low level stack control for packet sniffing
- be capable of some level of processing on the fly such that data packets are translated to second tier data of value to location derivation and this thesis

- be able to record the derived data to some form of transferable storage medium for further analysis
- similarly log additional sensor data suitable for referencing the location via an alternative means be capable of mobile field testing; although it is acknowledged that this does not include weather protection where the tests can be otherwise arranged to mitigate issues.

3.2.3. Methodology

The first step taken was to identify a number of available commercial-off-the-shelf radio modules and computing platforms. Systems were identified by a combination of internet searches, networking with other researchers and hardware design engineers, and observing the equipment and tools utilised in published research studies and papers.

Computing platforms can be categorised into many different types of system and intended purposes. A selection of systems has been considered for their suitability in this project and discussed in the results. The systems are split as best as possible into distinct areas; as technology evolves however cross-over and hybrid devices are becoming more prevalent and desirable. The categories are loosely arranged in order of increasing hardware design effort, meaning that the latter classes of system require intrinsically greater investment before return for the purposes of this project.

By performing a paper review of their capabilities and merits (using datasheets and published literature), the identified systems were narrowed to a much smaller selection. This was a subset deemed worthy of obtaining practical hands-on experience with such that a quantitative and subjective opinion could be formed. This process is shown pictorially in the results.

The final output of this investigation is a chosen pairing of computing platform and radio module that meet the requirements set out earlier.

3.2.4. Results

During the paper review of available systems a large number of contrasting possibilities were identified. This results section provides greater detail on the types of systems that were considered in order to arrive at the final decision which is later outlined in the conclusions.

3.2.4.1. IEEE 802.15.4 interfaces

There are four major approaches that could have been taken with this research for interfacing with IEEE 802.15.4 networks:

- using a computing platform (such as the TelosB mote described later) specifically designed for the task


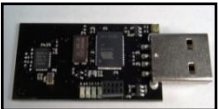
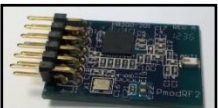
- plugging in a USB radio adapter (assuming the platform supports USB) to handle the physical layer and network stack and permit the computing platform to interact with the application layer at a much higher level
- using a prebuilt radio circuit module that is typically designed to output a serial interface to a controller which handles the application layer similarly to a USB adapter
- or designing a bespoke radio circuit from the ground up.

Of these, the USB adapter and prebuilt modules were the most accessible options.

The Atmel RZUSB Stick was an ideal choice out of the USB dongles; it had a strong level of support from prior researchers with accompanying open source software tools. Additionally, Atmel support the use of custom firmware allowing lower level access to the stack if required [101]. The alternative trialled was Microchip's Zena dongles; these had the advantage of spanning the full allocation of channels (868 MHz, 915 MHz and 2.4 GHz) however they were not as straightforward to command and interface with outside of the demonstration software.

There were many comparable radio modules for interfacing to development boards and embedded computing platforms. Most operated over a serial universal asynchronous receiver / transmitter port or via SPI / I2C buses. Of the available options, the Ciseco XRF, XBee Pro Series 1 and Digilent PModRF2 modules were trialled. The XBee modules transpired to be the simplest network to set up as a result of their comprehensive software package, however the Ciseco offerings had greater sensitivity and the Digilent module provided a the most versatile interface on a hardware level.

Table 5 contrasts the IEEE 802.15.4 interface options considered and their respective benefits. It also attempts to assign a subjective opinion score of how beneficial to the thesis investigation the option would be:

Option name	Pros and Cons	Subjective Score (1 low – 10 high)
<p data-bbox="204 286 392 349">Microchip Zena USB Dongle</p> 	<p data-bbox="454 286 1153 315">Between the three dongles the full channel set is covered:</p> <ul data-bbox="454 320 1153 472" style="list-style-type: none"> - 868 MHz hosts a single European channel - 915 MHz hosts thirty North American channels as of 2006 (<i>not licensed for use in Europe</i>) - 2.4 GHz hosts sixteen global channels that are of most interest in the UK <p data-bbox="454 501 1145 654">Zena network analysis application software allows some interpretation of communications (dominantly supporting Microchip's MiWi protocol) and of greatest interest, allows overlay of a pictorial network estimation onto a map graphic.</p> <p data-bbox="454 683 1082 745">Only able to analyse networks to which the dongle is associated.</p> <p data-bbox="454 775 1131 869">Some limited support available, receiver sensitivity data and the ability to interface with non-Microchip software is lacking.</p> <p data-bbox="454 898 1098 960">Possible to modify the hardware to accept an external antenna.</p> <p data-bbox="454 990 1099 1019">USB interface for computers but no low power modes.</p>	<p data-bbox="1273 320 1294 349">4</p>
<p data-bbox="204 1068 376 1131">Atmel RZUSB Stick</p> 	<p data-bbox="454 1068 975 1097">Only covers the global channels at 2.4 GHz</p> <p data-bbox="454 1126 1139 1189">Works well with multiple protocols, has built-in air capture mode so doesn't need to be associated with network.</p> <p data-bbox="454 1218 1062 1281">Atmel network analysis software is reasonably fully featured but not simple to use.</p> <p data-bbox="454 1310 1145 1435">Well documented firmware alteration and network penetration testing python scripts developed for academic researchers. Provides easy access to and logging of network data important to this thesis.</p> <p data-bbox="454 1464 1147 1527">Hardware choice of folded dipole antenna or chip antenna via the placement or removal of 0R resistors.</p> <p data-bbox="454 1556 1136 1619">USB interface and possibility of JTAG control. Does feature low power modes following firmware modification.</p>	<p data-bbox="1273 1102 1294 1131">9</p>
<p data-bbox="204 1664 427 1727">Digilent Peripheral Module PModRF2</p> 	<p data-bbox="454 1664 975 1693">Only covers the global channels at 2.4 GHz</p> <p data-bbox="454 1722 1098 1785">ZigBee and MiWi protocols supported but native IEEE 802.15.4 and 6LoWPAN not easily supported.</p> <p data-bbox="454 1814 1070 1877">Extremely high data rates possible compared to the standard.</p> <p data-bbox="454 1906 1155 2031">Well documented code libraries for interfacing available; however these do not enable low level stack control. Network data for joined networks is sufficient for this thesis investigation.</p> <p data-bbox="454 2060 1056 2089">SPI interface only, does include low power modes.</p>	<p data-bbox="1273 1697 1294 1727">6</p>

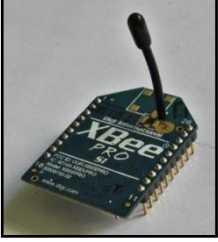

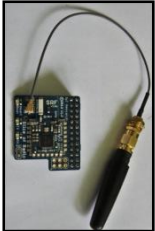
<p>XBee Pro Series 1</p> 	<p>Only covers the global channels at 2.4 GHz</p> <p>Well documented code libraries for interfacing available; however these do not enable low level stack control. Network data for joined networks is sufficient for this thesis investigation.</p> <p>Possible to modify the hardware to accept an external antenna.</p>	<p>7</p>
<p>Ciseco XRF / ARF / SRF</p> 	<p>Only covers the non-global channels (868 MHz and 915 MHz) so good coverage for North America, but limited to one less common channel in the UK and Europe.</p> <p>Works well with multiple protocols. Data for networks with no association is possible for this thesis investigation but not easy to acquire.</p> <p>Well documented code libraries for interfacing available. Well documented hardware specifications, good support availability and antenna patterns possible to obtain.</p> <p>Simple to modify the hardware to accept an external antenna.</p> <p>SPI or I2C communications and available in a range of footprints including surface mount or XBee form compatible. Very low power modes available.</p>	<p>5</p>
<p>Ciseco Slice of Radio</p> 	<p>Only covers the non-global channels (868 MHz and 915 MHz) so good coverage for North America, but limited to one less common channel in the UK and Europe.</p> <p>Works well with multiple protocols. Data for networks with no association is possible for this thesis investigation but not easy to acquire.</p> <p>Well documented code libraries for interfacing available. Well documented hardware specifications, good support availability and antenna patterns possible to obtain.</p> <p>Simple to modify the hardware to accept an external antenna.</p> <p>Specific interface for Raspberry Pi platform enabling easy use of penetration testing scripts for data collection and logging. No low power modes.</p>	<p>5</p>

Table 5: Comparing IEEE 802.15.4 radio modules for use in the investigation of this thesis

3.2.4.2. Mobile Computer Systems

At the highest level and most removed from any hardware development is the consumer computing category. This category broadly includes the following generic subcategories:

- Laptop and notebook computers

- Tablets, hybrid-tablets and phablets
- Smartphones and wearable devices

All of these subcategories have strong environments for developing custom software and applications.

Prior research into sensor network security provided a small number of suitable prebuilt tools and scripts for signal strength investigations and geolocation referencing. These were Linux based tools such as KillerBee, Scapy, OpenEar and zbWarDrive intended for laptop and desktop machines. These types of tools have begun to include the use of IEEE 802.15.4 protocols for detecting and connecting to ZigBee and XBee type networks. Whilst there may exist some scripts that would require very little alteration to perform some initial investigations, the support and help files for using or amending the tools is in the most part quite poor. A high degree of conversancy with Linux and the scripting languages used is expected and many areas are undocumented or awaiting further development.

Many laptops are available with built-in GPS, Bluetooth and WiFi; alternatively, these plus IEEE 802.15.4 interfaces are also available as USB devices. This makes system integration straightforward. Along with the use of the prebuilt tools, the available hardware offered a big advantage in terms of reduced development time before data collection and analysis could be carried out.

Conversely, a key advantage to smartphones or tablets in this project was the fact that University of Hertfordshire already had an ongoing development project using an Android application to provide a “Big Data” tool [102]. At the time of investigation this was in prototyping stages but did have working phone-sensor logging features and the output data could be manually passed to a computer for analysis.

Whilst still feasible, the use of a smartphone or tablet would present some difficulties when trying to interface with remote hardware providing the 802.15.4 network interface. It would have required custom hardware designs and additional application development to achieve.

Software development processes differ depending on the host operating system (Android, Windows or iOS) but typically require a verification or approval process before it can be used on more than one unit; this would have delayed and complicated development a little but was not a barrier in itself.

Additionally, battery life on most smartphones / tablets whilst continuously polling sensors may well have hindered prolonged tests unless the unit were continuously charging.

For these reasons, laptops and commercially available USB radio modules were a more obvious choice when compared to smartphones and tablet forms.

3.2.4.3. Embedded Operating Systems

There are several small, single board platforms designed to run an embedded operating system (by default, variants of Linux). The subset of examples considered for applicability to this thesis were:

- Raspberry Pi (chosen to represent the low cost option)
- Odroid U3 (chosen to represent the high computing power option)
- BeagleBone Black (chosen to represent the options with strong interfacing capabilities).
- Arduino Yun (chosen to represent the internet / cloud based operating system options)

Of the above, the BeagleBone Black represented the strongest choice due to the extensive examples database available and the ability to interface to both USB and non USB radio modules. For the Raspberry Pi there was a specific “Slice of Radio” IEEE 802.15.4 radio module by Ciseco which is supported by their software tools and represents a reasonably simple route to investigation. All of the options had broadly similar strengths and weaknesses to one another.

As with laptop computers, an embedded Linux platform such as this would allow the easy use of scripts and existing libraries / programs such as developed for networking and network infrastructure detection or penetration testing. The widely publicised concept of the “internet of things” has also really pushed development in this area and many tools are being developed with this in mind.

The main deterrent against using an embedded Linux style operating system for this project is the lack of integrated battery. When combined with their high power demand this made the option of fitting a bespoke battery difficult to achieve whilst maintaining a small mobile form factor. For the purposes of the thesis investigations this has inherent problems when trying to perform any sort of field testing – it would be necessary at all times to be near a source of permanent or replacement power.

Most of the embedded platforms to date have yet to fulfil reliance assurance and do not operate in real time (although there are some that do). The result of this is that it is not possible to be sure that running code will function asynchronously as it expected to and stable operation over prolonged periods of time is unproven. Neither of these issues were of direct consequence to this thesis, and indeed a real time operating system was not used, however would be worth bearing in mind for any expanded investigations a platform with ready access to such systems could have significant benefits. It was decided to plan for the possibility of further development from the outset as opposed to later needing to redesign much of the test bed created.

3.2.4.4. System on Chip (plus additional circuitry)

This category encapsulates modules that provide networking, storage, custom firmware and limited header ports on a small solderable daughterboard or IC thus creating an entire “system on a chip”. There were only a handful of possibilities identified as directly suitable for investigating this thesis without additional

processors. These still required additional circuitry to make them function and would require low level access to an IEEE 802.15.4 module's communications port. The major options identified were:

- Aria board (a breakout board for an Electric Imp module)
- Microchip RN131G (plus others in the RN and MRF ranges)
- Intel Edison Board

Out of these options, the Microchip modules (previously developed by Roving Networks) represented the best choice purely because of the author's prior experience. Some previous research, testing and development into IEEE 802.11 network investigations by the author were performed using these modules.

The Electric Imp module had a development model that was not compatible with the sponsorship of this research; it was the business model of this unit that the development environment, and all of the firmware created, resides upon the Electric Imp servers. In other respects, the usability and specifications of this module were very appealing.

As a rule, these devices typically consume little power and include power saving modes or functions. Combined with their small size, manufacturer provided development support and low cost (they are intended to be bought as components within mass produced products) some of these systems present some enticing possibilities. The lower level development entry point does mean that there is a step up in firmware development effort required when compared to those discussed earlier.

The major reason for deciding against the use of one of these systems is that none of the systems are capable of providing more than one means of location sensing without some bespoke circuitry and design to incorporate additional sensors at which point nearly all of the benefits of these products are negated. However a device such as this could be useful for other applications as a small and simple system with only a single sensor type – particularly if this were WiFi.

3.2.4.5. Microcontroller Development Boards

The options in this category are quite similar to the System on Chip options; typically they are larger, have more interfacing ports and represent a more beginner friendly development environment.

This was the largest category considered due to the extensive array of boards available and the low level networking control possible at entry level development skills. Some of the development boards particularly lent themselves towards specific sensor and radio modules, but most if not all were compatible with the majority of IEEE 802.15.4 development modules and USB modules. The list of development boards considered for this thesis investigation were:

- ARM
 - NXP LCPxpresso Board

- MBed boards e.g. LCP1768 (*the MBed environment is particularly supportive of the XBee IEEE 802.15.4 modules*)
- Atmel SAM3S-EK evaluation kit
- Atmel AVR
 - Arduino boards e.g. Mega / Duo
 - ATtiny boards e.g. Adafruit Trinket Pro
 - AVR32 boards e.g. Adafruit Bluefruit LE Micro
- Microchip PIC
 - chipKIT boards e.g. uC32 / Max32 (*the Digilent who make these boards also provide a specific IEEE 802.15.4 "PMod" accessory*)
 - Microchip Explorer 16 32-bit evaluation board
- Other / Network security specific
 - Netduino GO board
 - BusPirate
 - GoodFET
 - TelosB mote (*this has been made specifically as an integrated IEEE 802.15.4 development board for academic researchers*)

Of the selection, there were three strongest contenders, two of which were taken forward for trialling. On paper, the most logical choice appeared to be the TelosB mote; this had a strong link with other researchers' work and had been designed specifically for academic purposes. Unfortunately physically obtaining one to trial, let alone to use longer term, proved prohibitive. Alternatively, the author had greatest prior experience using MBed platforms. Combined with an XBee radio module and perhaps additionally a WiFi / GPS module, this would have represented a fast development route. In contrast, the chipKIT Max32 offered the lowest level stack control of IEEE 802.15.4 communications and supported the most powerful processor. This would have allowed for much more in depth control and investigation of the protocol than the others were offering.

Despite their individual advantages, all of the microcontroller development boards required a significant investment of time when compared to using a Linux based system and a USB radio dongle.

3.2.4.6. FPGA / CPLD Development Board

Only three choices were considered from this category as it was known at the start that this category represented the technology with which the author had least experience. This meant that any system chosen would have required a significant investment in time and learning. The options considered included an offering from each of the major FPGA vendors (Xilinx versus Altera) and additionally a representative of CPLD technology:

- Digilent Nexys 3 Spartan 6 FPGA development board
- Digilent CoolRunner 2 CPLD development board
- Terasic Altera DE2-115 FPGA development board

As the closest route to application specific integrated circuits considered, these represented the lowest level development choice combining an increased development effort with greatest flexibility.

As well as offering the use of third party radio modules and USB radio dongles, with an FPGA / CPLD it would have been possible to implement a software defined radio. This would have provided redefinable and fully configurable IEEE 802.15.4 connectivity with full control over all aspects of the radio stack. For the purposes of this thesis this could have meant the ability to run multiple simultaneous radio channels, quickly change protocols and retrieve low level information such as received signal strengths and packet loss statistics with relative ease. The multichannel and redefinable characteristics are not matched in versatility by any of the other technology options considered.

3.2.5. Conclusions

As a result of this investigation, it was decided to use a laptop computer with inbuilt WiFi, a USB GPS receiver and a USB IEEE 802.15.4 transceiver to undertake the bulk of the testing. The IEEE 802.15.4 transceiver chosen was the RZUSB Stick as it had provision for firmware alteration, but predominantly, it also had existing academic usage with open source code available for investigating ZigBee networks.

The chosen Atmel RZUSB Stick and accompanying python scripts successfully provided a versatile and almost-off-the-shelf solution for the purposes of investigating this thesis. The use of a laptop computer was by a long way the quickest and most adaptable from of computing platform available. However it is acknowledged that for field testing, with potentially multiple units, it could easily be possible for a small, low cost and low power system to have many practical advantages.

In addition, the testing undertaken at this stage provided a very simplistic manner for simulating ZigBee and ZigBee Enterprise networks. By using the default topologies provided with the XBee Pro radio modules, it was only necessary to connect power to the modules and a functioning network would be created. In combination with a demo application (of just a few lines) developed for an MBed board this then provided the data stream and frequency of transmission. Thus a fully synthetic network could be created for simulated testing with minimal effort or set up.

Although discounted at the time of investigation, once the University of Hertfordshire Big Data Android application has progressed and can take input from external sensors for IEEE 802.15.4 this may be an interesting option to pursue.

As already discussed, the hardware platform investigations were necessarily brief and could not attempt to cover all possible equipment. Subsequent to the majority of script writing and data collection, an alternative USB module for IEEE 802.15.4 packet sniffing was discovered by the author. Californian Eastern Laboratories' EM357 USB Stick [103] has not been tried in the scope of this project, but promises to be a well characterised and sensitive receiver with a host of useful software tools

for network sensing and investigation. The availability of published antennae patterns would have reduced the required effort of some of the subsequent testing undertaken and added greater confidence to the results.

3.3. Discovering network transmissions

3.3.1. Purpose

In order to use IEEE 802.15.4 network transmissions as a means of location, it must be possible for the sensor equipment (in this case an Atmel RZUSB Stick as identified in section 3.2) to detect and process messages sent by other networks.

Although the networks used during later tests were under the ownership of the author and this project, it was necessary to assume no prior knowledge of the systems. In this way it would be possible to simulate capturing real world transmissions from networks beyond the control of the end user such as the intended smart meter networks.

Before attempting to collate signal strengths, unique identifiers, locations, or any other data from a simulated third-party network, it is necessary to know which channel to listen for data on. A method that listens for transmissions on a channel for a period and then hops to the next channel may miss some or all networks. This is due to the sporadic nature of the IEEE 802.15.4 network messages and potentially not listening to the right channel at the right time.

3.3.2. Requirements

This test needed to show that it is possible to reliably identify a channel that has live IEEE 802.15.4 network traffic in order that the sensor equipment can be set to the settings for collecting further data.

Given that the global frequency allocation for IEEE 802.15.4 networks is within the 2.4 GHz band which is also shared with many other users, it was necessary to be able to differentiate desired transmissions from other radio emissions.

It was necessary to simulate the detection of a third-party network and so no prior knowledge of the network was assumed.

3.3.3. Methodology

The approach taken was to obtain a radio spectrum profile of the 2.4 GHz band and confirm whether a distinctive pattern or feature could be observed when a simulated smart meter network was activated. The radio spectrum profile would consist of frequency density plots, waterfall graphs and utilisation plots; combining to show information on frequency, data flow / rates and transmission power across the 2.4 GHz band.

In order to observe differences, it was necessary to obtain a background scan with no IEEE 802.15.4 network transmissions, followed by a detection scan with a live network. Both scans were made over a duration of fifteen minutes in order to fairly compare the reports, accounting for transient emissions and background variation.

A ZigBee Enterprise network was built using two XBee Pro Series 1 modules; one of which was mounted upon an MBed LCP1768 development board. This hardware was identified as suitable for the task during the earlier testing (see section 3.2.5). The network was programmed with ten second message transmissions and acknowledgement bursts on ZigBee channel twelve to replicate the smart meter operation as set out in the UK Government technical specifications [24]. A message transmit count was displayed upon the LCD of the MBed platform for information purposes. See Appendix 6 for the MBed code routines used. The messages delivered to the receiving were also displayed on the laptop screen using XCTU the XBee application software; this provided confirmation of a functioning network.

This test was performed using a USB 2.4 GHz WiSpy Channelizer (Figure 11) and its associated software. This equipment was used as opposed to laboratory test and measurement equipment due to its size and portability. The project test system defined in section 3.2.5 comprised of a laptop computer, USB radio module and USB GPS module. The choice of this USB WiSpy Channelizer also meant that the entire investigation process could operate from the same laptop without additional test platforms. In the event that this spectrum analyser had not proved successful, a standalone spectrum analyser would have been used to compare and contrast the results.



Figure 11: WiSpy Channelizer with a 5 dBi 2.4 GHz antenna

3.3.4. Results

The semi-automated reports produced by the WiSpy Channelizer software have been included in full in Appendix 3 however the key plots have been reproduced and annotated below.

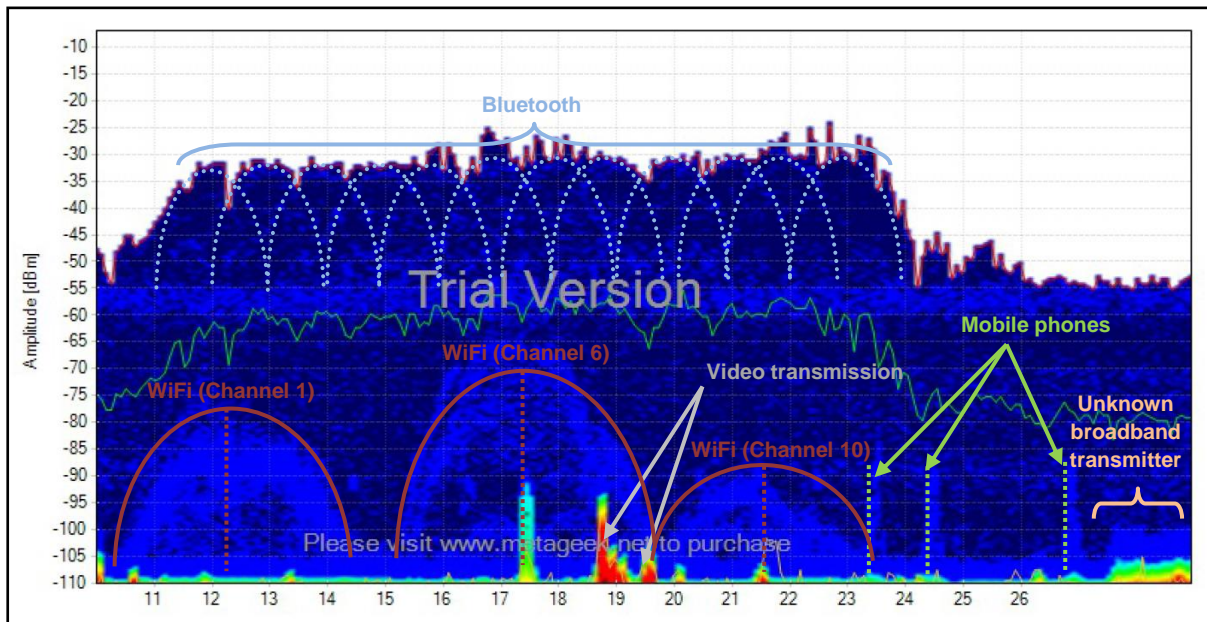


Figure 12: Background scan of the 2.4 GHz band (aligned with IEEE 802.15.4 decimal channel numbering)

Figure 12 shows a background scan of a computing laboratory environment containing WiFi routers, mobile phones and 2.4GHz digital video senders. The labelled profiles could be matched with equipment in the building; however the high frequency broadband transmitter labelled could not be identified.

In Figure 13, all the same noise sources are present; some even to a greater extent. Additionally however, there is a characteristic peak at 2.405 GHz corresponding to the Channel 12 setting used in the XBee radio module.

By observing the position of this peak it is possible to tell that a ZigBee network is operating over channel twelve as expected.

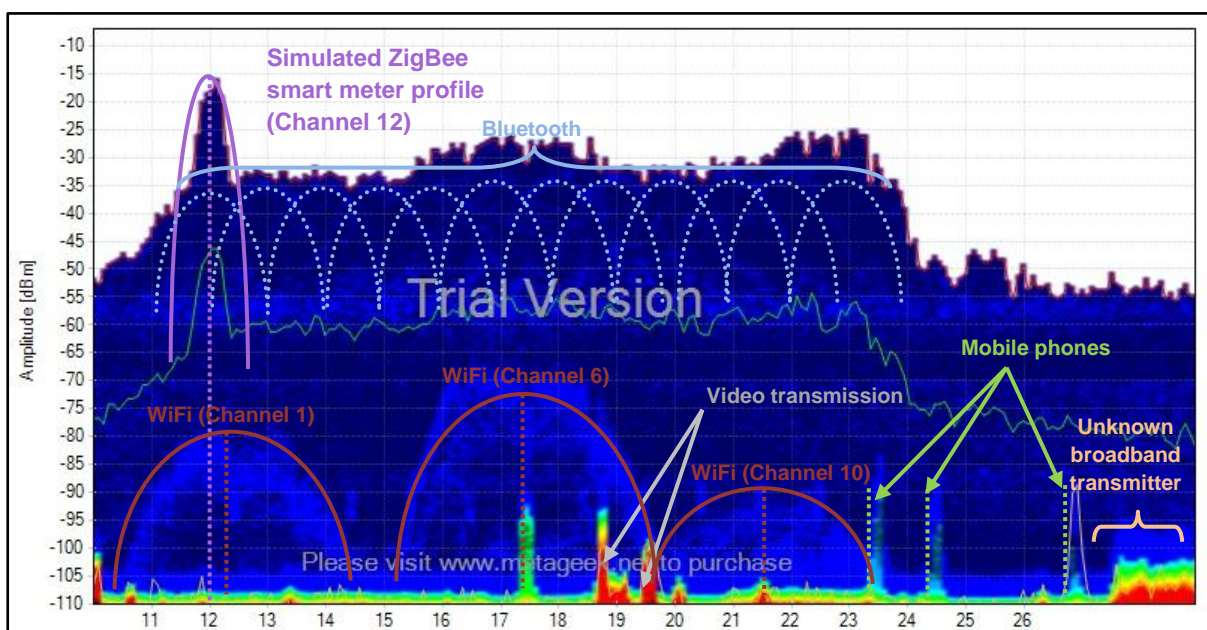


Figure 13: Scan of the 2.4 GHz band during network transmission (aligned with IEEE 802.15.4 decimal channel numbering)

When further from the ZigBee antenna, and no longer line of sight, the received amplitude of transmissions were much smaller. In this scenario the characteristic peak was not easily discernible from the background – particularly given the correlation with a Bluetooth hopping frequency. Neither the waterfall or utilisation graphs were of any aid to identify this as the transmitted data was negligible compared to the other noise sources.

3.3.5. Conclusions

This testing showed that with moderate proximity it is possible to differentiate target networks from the radio background using a spectrogram alone. The waterfall and utilization plots produced gave useful information regarding other emission sources, but showed a distinctive lack of data for the simulated ZigBee network. This was because of the low data rate, small packet size and infrequency transmissions. This matches with expectations based upon the IEEE 802.15.4 standards and the associated spectrum profile.

Using the described technique is an effective way to ascertain the channel in use by an unknown IEEE 802.15.4 network if one can gain a margin of clearance in terms of received signal strength. With the same caveat this technique works equally well where multiple channels are transmitting in the local vicinity.

In a realistic scenario (non line of sight and at unknown distance from the transmitter) it would be necessary to resort to a channel scanning approach with an IEEE 802.15.4 receiver.

Having performed this step it was possible to assume knowledge of the transmitter channel and progress to the tests in section 3.4 - identifying some of the transmitted information that can be used for geolocation.

3.4. Extracting information for geolocation

3.4.1. Purpose

Having identified the possible methodologies for deriving location in the literature review (Chapter 2) and chosen the hardware for use in the research investigations, it is necessary to confirm the data relevant to location that it is possible to extract.

From the IEEE 802.15.4 standards [104], [105] it is understood that a unique identifier for each transmitting node and network should be available and transmitted in the clear. This would make proximity type location analysis possible.

Some indication of signal strengths should also be measurable, although whether this is accessible at an application layer will be hardware dependant. As discussed in Chapter 2, signal strengths would allow for multilateration and advanced fingerprinting techniques.

If the hardware possesses a directional antenna (unpublished for the RZ USB but will be explored more in section 4.2) then this may make triangulation type techniques possible to investigate.

Finally, if there is any transmitted information about the time of transmission and an accurate hardware clock is accessible then it ought to be possible to investigate time of flight techniques for deriving location.

By understanding what data is accessible at the application layer using the Atmel RZUSB Stick it will be possible to confirm both the suitability of the hardware for this research project, and the potential applicability of different methodologies to the research goals.

3.4.2. Requirements

For this testing to prove successful, it should evidence first hand some form of information capture that could be used for location derivation; as a minimum, a unique network reference allowing proximity analysis possible.

Given at this stage all possible data will be captured, to comply with the need to mitigate intrusion of third party networks this testing needs to take place using a combination of simulated networks and an IEEE 802.15.4 sterile environment.

This testing should make use of the hardware and scripts intended for use in the remainder of the research in order to prove their suitability to the task.

3.4.3. Methodology

Figure 14 shows the equipment used for this testing (note that the distance between each of the three nodes was set to approximately 1m in a triangular formation as opposed to the immediate proximity shown for photographic purposes). The testing was performed in a radio frequency shielded laboratory in order to ensure a sterile environment whereby only the author's own networks would be investigated. No other radio equipment within the 2.4 GHz band was operational within the laboratory at the time of testing.



Figure 14: Linux scripts operating an RZUSB Stick whilst two XBee modules communicate

The RZUSB Stick hardware was first reprogrammed with the customised KillerBee firmware (See Appendix 5). This was required for gathering information regarding transmissions related to third party networks (in this case third part meaning simulated networks and those under the ownership of the author, but not in any way associated with the RZUSB Stick). In WiFi terms this would be to operate in a promiscuous mode and it is a methodology often employed for packet sniffing and penetration testing.

Some simple MBed code developed for the testing in section 3.3 was reused to observe the transmissions when sending known data (see Appendix 6). This code transmitted “Hello World” at regular intervals to the second XBee transceiver which would automatically send an ACK (acknowledgement message) in reply. By

comparing the packets sent between the XBee modules and those intercepted by the RZUSB Stick it was possible to identify the sting location of the bytes containing desired information.

The receiving XBee node was connected to an IEEE 802.15.4 network management tool called Moltosenso IRON. This performs a very similar function to XBee's own XCTU software, however it was possible to operate on a Linux operating system which was required for the KillerBee scripts controlling the RZUSB Stick. By using the network management software it was possible to both confirm proper operation of the simulated network, and to cross correlate the KillerBee data strings with those transmitted.

Once the data format was understood the KillerBee python scripts were adapted to begin classifying and recording relevant information. A source listing of the adapted script has been included in Appendix 7 section 12.2.1; given the proper parameters when running from a Linux console, the new script scans the named channel (if none is set as a parameter then the script will cycle through all channels) at a sample rate defined by the user. Any relevant profile network traffic is displayed to the console and optionally logged to a comma separated values (CSV) file – a template of which is also provided in Appendix 7 section 12.2.1.

To prove the aptitude and relevance of the adapted scripts, some sample transmissions of an actual smart meter⁴ were obtained over a short time period. The aforementioned simulated network was also transmitting in the vicinity at the same time to see if the two networks could be readily distinguished.

3.4.4. Results

Initial attempts at using the KillerBee scripts to extract personal area network identifiers (PAN ID), source identifiers and destination identifiers were not successful. Figure 15 shows a comparison of the identified information with that confirmed via the network manager software connected to the receiving XBee Pro module.

⁴ Although using a real smart meter, this was a unit under the ownership of the author, only the networking information was analyzed and no attempt at data decryption was undertaken. The data was not recorded other than by screen capture, and no metadata such as date, time or location has been attributed to the information captured. To further mitigate any risks of violating employer policies, full permission was additionally granted via the author's management chain for this specific test.

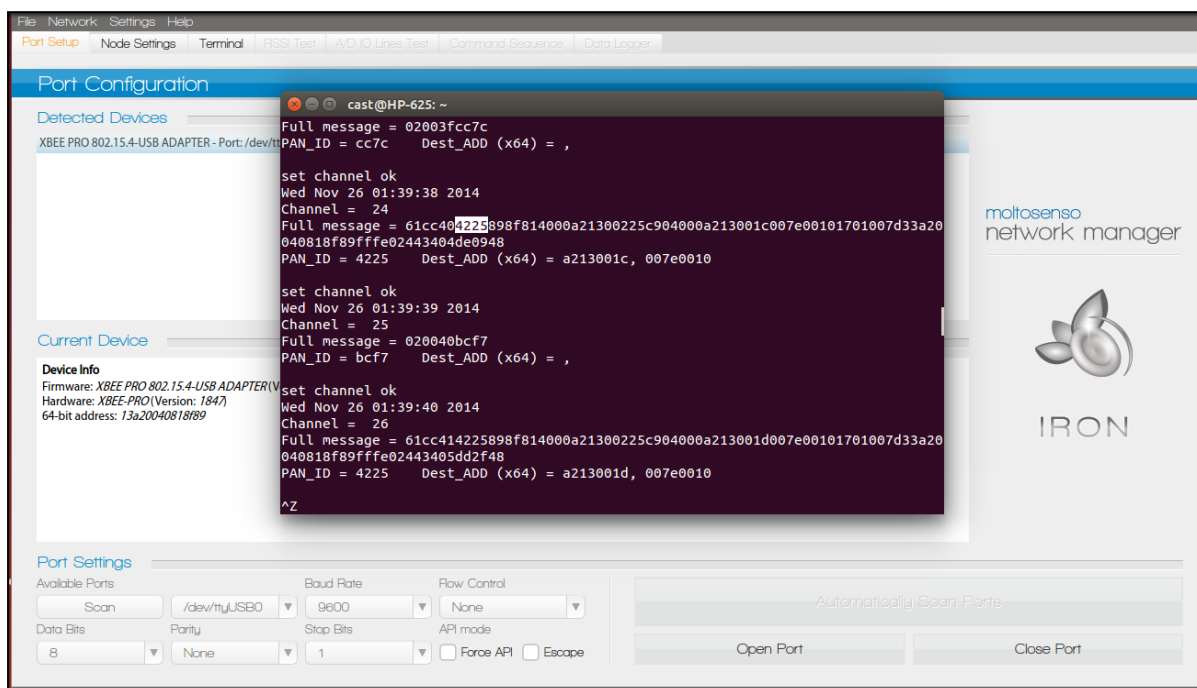


Figure 15: Initial attempt at extracting PAN ID and Network ID; comparing captured data with Moltosenso

Several issues were identified with this first attempt:

1. The script was trying to extract PAN IDs and Destination IDs from acknowledgement (ACK) messages (the second message received in Figure 15). The IEEE standard makes it clear that these message types do not contain this information.
2. The PAN ID and Destination ID were being reported in a slightly unusual format – the containing string is comprised of several concatenated two-bit hexadecimal numbers (each two-bit number representing one byte). The string was concatenated in little-endian format however the individual bytes were formatted in big-endian.

This means that whereas the PAN ID was supposed to be expressed as h:2542, it was actually represented as h:4225. Likewise, the sixty four bit destination address should have been h:0013A200 40818F89 but was actually represented as h:898F8140 00A21300 (note that unhelpfully Moltosenso does not display leading zeros).

3. The extracted string representing the Destination ID was mistakenly offset by several characters.

Figure 16 shows the output of a partially refined script whereby issue three from the list above has been resolved and the sixty four bit destination address is properly captured (albeit in the unusual format).

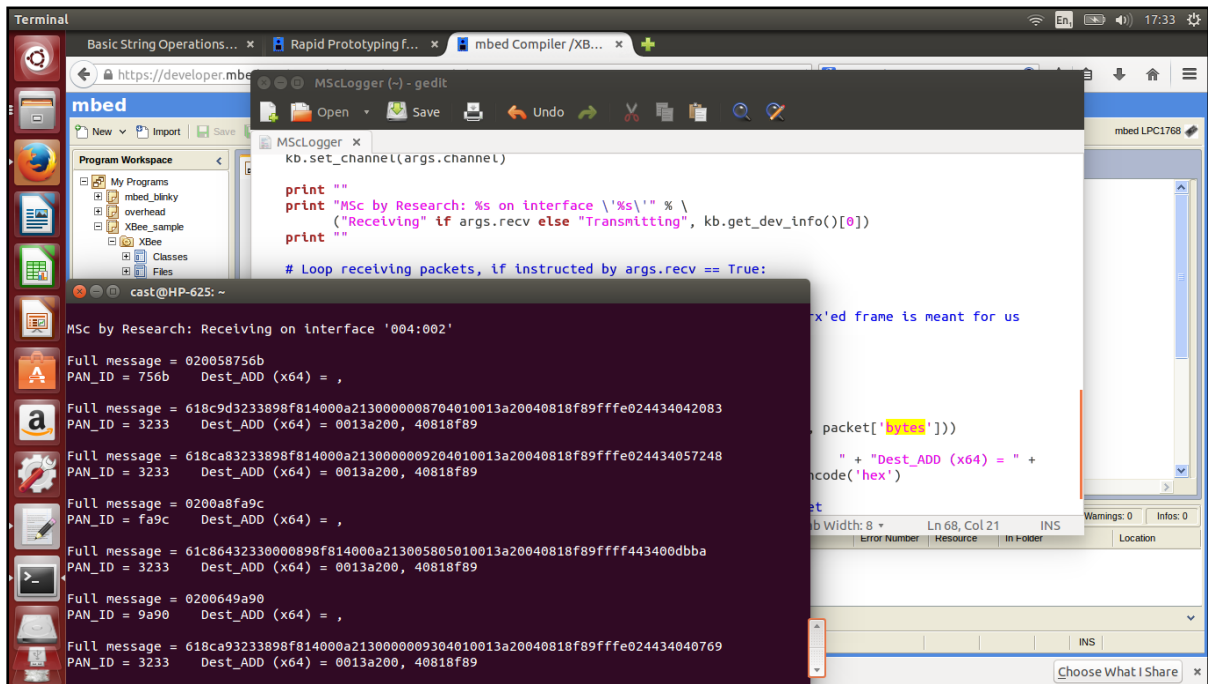


Figure 16: Partially refined data extraction; bytes still represented in little-endian

With the proper location relevant information being extracted ready for displaying or logging attention was turned to testing and refining the ability of the scripts to scan through multiple channels (channels 11 to 26 are within the 2.4 GHz band). This is shown in Figure 17.

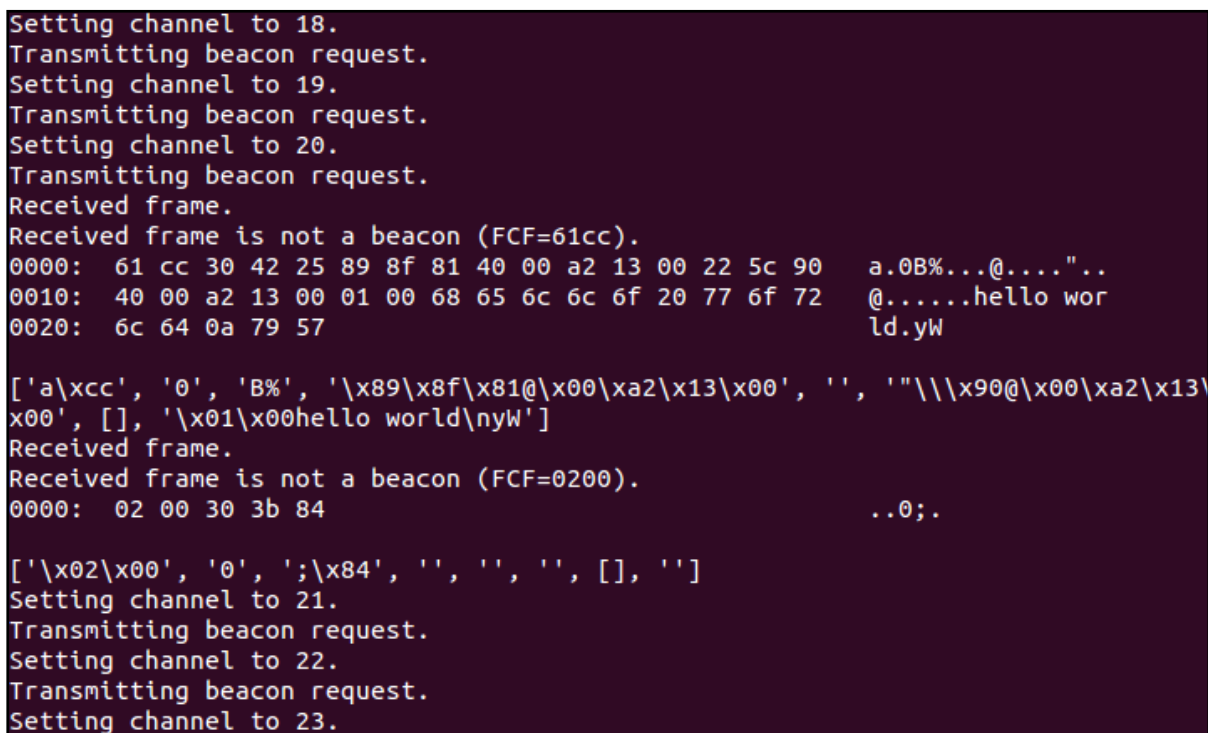


Figure 17: Raw data from the simulated wireless personal area network whilst demonstrating scanning channels

Following the success of this, the script was further refined so as to resolve issue two; with this achieved some effort was placed into understanding message types in

order to begin categorising messages. Initially all message types that did not correspond to a Beacon Frame were suppressed. This served two goals – it helped ensure that only responses to user initiated presence requests were displayed or logged, but it also temporarily resolved the issue of the scripts attempting to extract network information from message types that do not contain the required fields.

With the scripts sufficiently functional to perform a constrained data capture (without logging) of a real system, the screen capture in Figure 18 was obtained. This successfully demonstrated the following achievements:

- the system scanning through the channels, transmitting beacon requests and listening for responses
- the system identifying and discarding messages from the simulated network (still operating on Channel 20)
- the system identifying and discarding messages from the smart meter that were only intended for reception by the smart meter (i.e. not a response to the RZUSB Stick's beacon request)
- the system identifying and displaying responses to the RZUSB Stick's beacon request and successfully identifying the PAN ID, Source ID, Extended PAN ID and Stack Profile from each responsive node

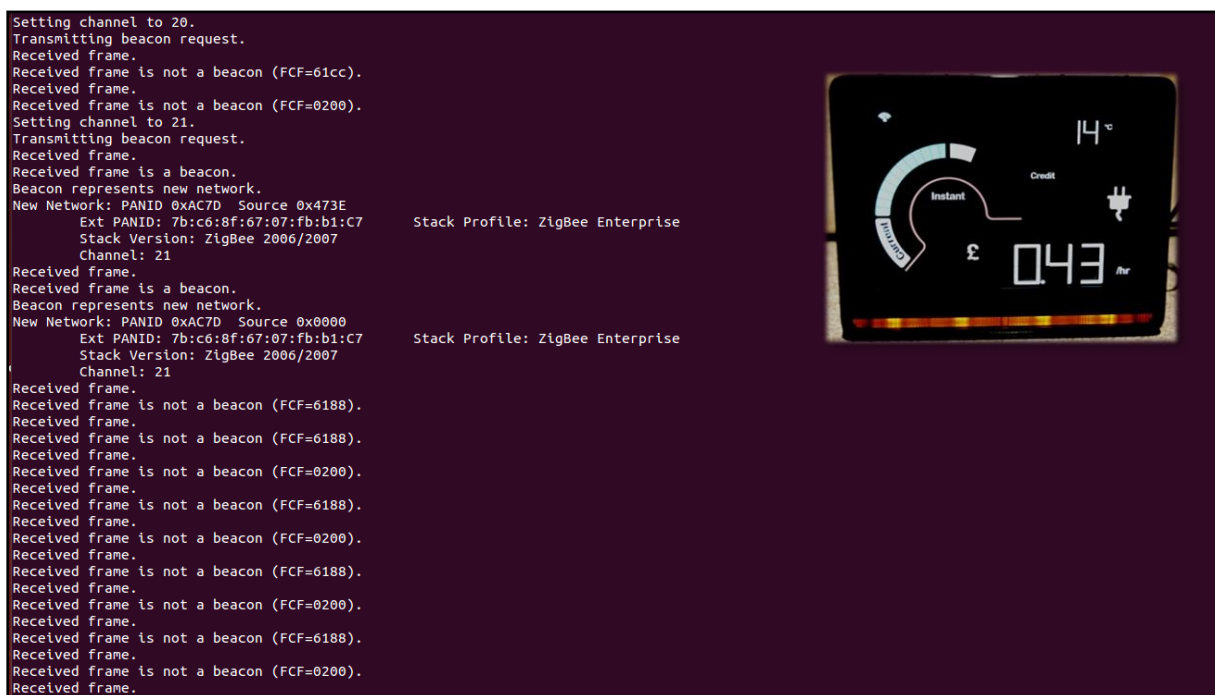


Figure 18: Sample data from a smart meter

As well as the above achievements, the results shown in Figure 18 also proved and informed the following points fundamentally important to the thesis:

- It is possible to read uniquely identifiable information from a smart meter without decrypting any of the data transmitted

- The smart meter network under test consisted of two nodes and both of these nodes transmitted relevant unique identifiers (PAN ID, Source ID, Extended PAN ID, Channel number, Stack type)
- The smart meter network communicated far more frequently than the requirements of the standard – as opposed to one message every ten seconds (and striving for one per every five seconds), the network communicated data frames (with extractable identifiers) approximately every second
- It was shown to be possible to solicit a beacon response with relevant unique identifiers upon demand – this means that information is available at any point in time regardless of the data frame frequency of the network

Finally, in preparation for the range testing in Chapter 4, other message types such as acknowledgement frames were separately classified such that their occurrence could be noted but no other information relating to these messages obtained. The adapted script is listed in Appendix 7 section 12.2.1; Figure 19 shows the resultant console window following the classification of acknowledgement messages.

```
zbstumbler: Transmitting and receiving on interface '005:003'
Transmitting beacon request.

Received frame.
Unrecognised message
0000: 61 cc 3c 42 25 89 8f 81 40 00 a2 13 00 22 5c 90   a.<B%...@...."..
0010: 40 00 a2 13 00 19 00 68 65 6c 6c 6f 20 77 6f 72   @.....hello wor
0020: 6c 64 0a 6a 74                                     ld.jt

Packet length, 37 bytes.

Received frame.
***ACK message***

Transmitting beacon request.
^C
2 packets transmitted, 2 packets received.
```

Figure 19: Error handling classifies ACK (acknowledgement) message separately to data messages

3.4.5. Conclusions

The results of this testing conclusively demonstrated that uniquely identifiable data is obtainable almost instantaneously from a smart meter network that is within range. The data is transmitted in the clear and is accessible from data or beacon frames.

The unique information such as PAN ID and Source IDs could be collated into a database along with geographical coordinates to use in a proximity methodology for deriving location. Using active scanning techniques (i.e. requesting beacon frames) it should be possible to obtain geocoded information at any speed so long as the device is within range of a network.

It is desirable to be able to identify details about the network detected such as what IEEE 802.15.4 profile the network belongs to. For instance if the difference between smart meters and medical or automation equipment could be distinguished then filters could be applied at the MAC layer to prevent the unintentional collection of untargeted data and focus purely upon smart meters. This would have an additional advantage in terms of location derivation benefits as it would ensure reliable information sources that will not be mobile or transient.

3.5. Findings on the use of smart meters and the RZUSB Stick

3.5.1. Achievements and impact

In section 3.2 a number of hardware choices were presented and compared; this was an enabling task to ascertain appropriate tools for performing the research. The chosen platform was the Atmel RZUSB Stick, which can be reprogrammed with customised firmware for third party network analysis as is shown in Appendix 5.

Section 3.3 began the first look at how IEEE 802.15.4 networks might be detected and possible methods for focusing the channel selection. Practically, this testing showed that a spectrum analysis can only determine transmitter channel when within an extremely close range. This means that for the purpose for navigation or geolocation a 2.4GHz spectrum analysis is not likely to be of much tangible use.

As a result of section 3.3, the methodology chosen (and successfully trialled in section 3.4) to detect the transmission channel is to cycle through the channels with a small dwell period whilst transmitting a beacon request. By sending a beacon request message any functional coordinator nodes within range are seen to respond with network details as is to be expected according to the IEEE standards [104].

Section 3.4 further went on to show that the difficulties of being in the right place at the right time to detect transmissions as anticipated in section 2.2 are unfounded. It is quite possible to receive an almost instantaneous response by actively listening for networks (using the beacon requests).

During the course of section 3.4, significant ground was also made in terms of filtering transmissions for only those relevant to the research questions. This had numerous advantages in terms of limiting necessary storage memory, reducing data processing requirements, ensuring reliable geocoding references and preventing unintentional intrusion of privacy.

In all, this chapter has shown that there exists consumer accessible hardware (albeit with firmware modification) that is capable of performing proximity type location methodologies.

3.5.2. Next steps

This chapter has shown the ability of an RZUSB Stick to use proximity detection of smart meters as a methodology to derive location (in the manner of current smartphone technology using WiFi). Ideally, the next step would undertake a practical analysis of the proximity method and compare the accuracy against a well documented alternative such as WiFi.

Section 2.2 showed that smart meters are expected to possess a much smaller transmission range than WiFi routers; this means proximity detection would have a much smaller scope in terms of possible locations of the detector equipment. The possible positions could be decreased further when mixed with a performance enhancing technique such as map matching (for instance to force the detector position to a path when a signatory gait pattern is detected by the accelerometer or forced to a road when travelling above walking speeds).

On the flip side, due to their decreased transmission range (and so geospatial coverage) there is likely to be less overlap of smart meter transmissions than there is with WiFi routers. In sparsely spaced areas there may be zero coverage whereas WiFi may have provided limited coverage. This means that position measurements may not be as continuously obtainable as WiFi.

By taking an approach such as wardriving with both WiFi detectors (a promiscuous mode WiFi card), a modified RZUSB Stick and a reference GPS device, it would be possible to undertake a comparative analysis. At present however there are two main prohibiting aspects:

1. The smart meter roll out is still in the initial phases and 90% United Kingdom coverage is not anticipated until post 2020. At this point in time it would be an unfair comparison between the two systems as most domestic properties will possess a WiFi router but not a smart meter.
2. To prevent collateral privacy intrusion and operate within the sponsor's policies, this testing would be significantly constrained to a point where alternative means of inferring a comparison would need to be considered. It would not be desirable for instance to create a geographically linked database identifying which households operate on which channels; what their smart meter networks are uniquely identified by; and potentially by analysis of this information, which supplier each household has chosen for their energy provision.

Mitigations could be taken to alleviate both issues such as to perform the measurement and analysis upon a new and unpopulated housing development, or to artificially replicate the same scenario in a controlled area. To attempt this would have taken an unpalatable amount of time, negotiation, and financial resourcing for the purpose of investigating this thesis.

Instead, the merits of alternate location methodologies using IEEE 802.15.4 networks have been investigated to a similar level as proximity has been here. In Chapter 4 the applicability of multilateration using received signal strength is explored, and in Chapter 5 an attempt is made to consider a scene analysis approach (aka fingerprinting).

Chapter 4 – Correlation of Received Signal Strength and Range

Chapter Summary

The range testing undertaken and the derived correlations between distance and the received signal power of an IEEE 802.15.4 receiver are outlined and compared to researchers in similar fields.

Some of the difficulties faced and suspected sources of error are investigated and the suitability of received signal strength geolocation methodologies for use with low-rate wireless personal area networks is determined.

4.1. Chapter introduction

This chapter attempts to address questions surrounding the expected performance of an IEEE 802.15.4 location system using multilateration or a similar range based location derivation methodology. This testing is performed with reference to the existing literature and contrasts against similar work in radio frequency identification (RFID) tag, Bluetooth and other radio frequency studies.

Received signal strength has previously been investigated for range determination in wireless sensor networks in two papers; one by Benkic et al. [2] and the other by Heurtefeux and Valois [3]. It has been used widely by researchers of alternate radio frequency technology, most notably WiFi and Bluetooth as was seen in Chapter 2.

The two papers named above were at odds with the general view of the literature which holds that received signal strengths of all radio frequency transmissions should have a strong and calculable correlation to range. The only researchers to have investigated using third party wireless sensor networks for geolocation derivation have stated the task as not possible.

Through these investigations the author shall attempt to utilise the best approaches identified during the literature review. The author intends by experimentation to disprove the assertions of Benkic et al. and Heurtefeux and Valois, and thus uphold the assumptions of the wider research community.

The chapter concludes with recommendations upon the use of the IEEE 802.15.4 standards for the purposes of location in urban and semi-urban environments based upon the data collected.

4.2. Understanding RZUSB Stick received signal strengths

4.2.1. Purpose

To be able to calculate any multilateration or radio fingerprint geolocation algorithms it is necessary to obtain a meaningful value for received signal strength.

The RZUSB Stick firmware provides a radio frequency power indication, for the channel in use, in the range 0 to 28. This could be used directly to create a model of range based upon the figure provided; however it would be much more beneficial to provide the radio frequency power in terms of decibel meters (dBm). This means that measurements could be compared with those of other researchers working with WiFi and RFID / beacon systems. For instance it should be possible to utilise the inverse square law of radio propagation to predict the distance between transmitter and receiver based upon the loss in radio frequency power measured at the receiver in comparison to the known (or rather assumed) power at the transmitter.

As well as converting to a standard measurement unit, it is useful to obtain knowledge regarding the polarity and directionality of the antenna on the RZUSB Stick. If the antenna were directional this would open up the possibility of triangulation techniques; whereas an omni-directional antenna is advantageous for proximity and multilateration techniques in order to mitigate differences in user orientation. Atmel do not provide any details regarding the directionality of the antenna on the RZUSB Stick.

As will be seen in section 4.3, directionality knowledge was also a necessity for more accurately and effectively generating a model of received signal strengths in relation to distance from the transmitter.

4.2.2. Requirements

The three fundamental goals of this testing were:

1. To be able obtain a measurement in decibel meters for the received signal power suitable for use in a simplified inverse square law model for RF power versus range.
2. To obtain a generalised view of the directionality of the receiver.
3. To identify the best combination of orientations for transmitter and receiver nodes to maximise high signal strength communications.

To achieve these goals accurately and reliably it was essential that there was minimal other radio traffic on the same frequency band which may have distorted the results. It was also important that measurements were taken in an accurate,

repeatable manner so that they could be validated at a later date if subsequent testing did not perform as expected.

4.2.3. Methodology

The RZUSB Stick circuit is based upon an AT86RF230 integrated circuit. The datasheet for this states the following about received radio power and received signal strength indication (RSSI) [106]:

The read value is a number between 0 and 28 indicating the received signal strength as a linear curve on a logarithmic input power scale (dBm) with a resolution of 3 dB. An RSSI value of 0 indicates an RF input power of < -91 dBm, [...] a value of 28 a power of ≥ -10 dBm.

[...]

For an RSSI value in the range of 1 to 28, the RF input power can be calculated as follows:

$$P_{RF} = \text{RSSI_BASE_VAL} + 3 \cdot (\text{RSSI} - 1)$$

The KillerBee framework created by Riverloop Security [98] was used and adapted to create a python script which would use two RZUSB Sticks and the formula for received power calculation (presented above) to investigate received signal strength indication with range. The source listing is presented in Appendix 7 section 12.2.2.

The script created was designed to operate in two modes dependant on the parameters passed by the operator:

1. As a transmitting node; the script will in this case transmit a defined number of beacon requests on a defined channel with a defined transmission interval. In the following testing the channel used was number 26 as this was the least congested. There were a thousand transmissions, one every half a second, to give a statistically significant measurement within a manageable timeframe.
2. As a receiving node; in this mode the script would identify beacon request frames upon a specified channel and log their occurrences with a representation of the received signal strength reported by the RZUSB Stick and the calculated radio power. Upon closure of the script, the total number of received beacon requests are presented in order to compare against the number transmitted.

The data filtering and classification processes developed in the previous chapter were essential for these scripts to be created.

Figure 20 shows both modes of this script being used to transmit and capture a thousand beacon requests from which to derive a distribution of received radio power at the range and orientation measured.

```

cast@HP-625: ~/Documents/Testruns
RSSI: 13, Power: -54 dbm (x981)
***Beacon Request***
RSSI: 13, Power: -54 dbm (x982)
***Beacon Request***
RSSI: 13, Power: -54 dbm (x983)
***Beacon Request***
RSSI: 13, Power: -54 dbm (x984)
***Beacon Request***
RSSI: 13, Power: -54 dbm (x985)
***Beacon Request***
RSSI: 13, Power: -54 dbm (x986)
***Beacon Request***
RSSI: 13, Power: -54 dbm (x987)
***Beacon Request***
RSSI: 13, Power: -54 dbm (x988)
***Beacon Request***
RSSI: 13, Power: -54 dbm (x989)
***Beacon Request***
RSSI: 13, Power: -54 dbm (x990)
***Beacon Request***
RSSI: 13, Power: -54 dbm (x991)
***Beacon Request***
RSSI: 13, Power: -54 dbm (x992)
^C
***END***
Total received: 992

cast@HP-625: ~/Documents/Testruns$

cast@HP-625: ~/Documents/Testruns$
Transmitting beacon request (x979)
Transmitting beacon request (x980)
Transmitting beacon request (x981)
Transmitting beacon request (x982)
Transmitting beacon request (x983)
Transmitting beacon request (x984)
Transmitting beacon request (x985)
Transmitting beacon request (x986)
Transmitting beacon request (x987)
Transmitting beacon request (x988)
Transmitting beacon request (x989)
Transmitting beacon request (x990)
Transmitting beacon request (x991)
Transmitting beacon request (x992)
Transmitting beacon request (x993)
Transmitting beacon request (x994)
Transmitting beacon request (x995)
Transmitting beacon request (x996)
Transmitting beacon request (x997)
Transmitting beacon request (x998)
Transmitting beacon request (x999)
Transmitting beacon request (x1000)
Transmitted 1000 times
cast@HP-625:~/Documents/Testruns$

```

Figure 20: RX and TX scripts script in operation during RSSI measurement

For the test, two RZUSB Sticks were placed 1m apart and 1 m above floor height in a radio frequency shielded lab whilst connected to the laptop operating the python scripts in Ubuntu Linux. The lab setup was as per Figure 21 below using a 1m³ non-metallic table designed for electromagnetic compatibility testing. During the testing the author additionally removed himself from the lab as the 2.4 GHz band is sensitive to water and the close presence of a human body. The script was written with a countdown timer and high volume audible completion alert to facilitate this manner of remotely taking measurements.

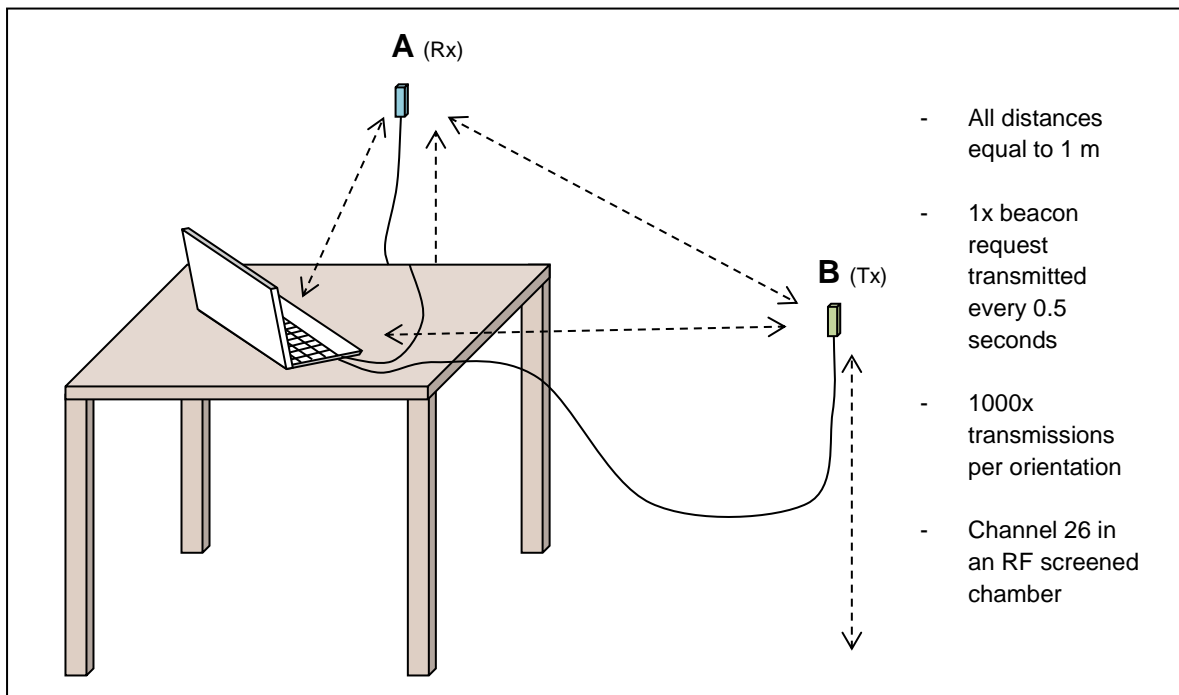


Figure 21: 1m directional RSSI testing setup

All sources of transmission in the lab which operated within the 2.4 GHz band were removed or turned off and a background scan was performed using the WiSpy Channelizer as described in section 3.3; the quietest channel was chosen for this test (channel 26). This was to prevent radio interference from distorting the received signal strength measurements as the Atmel datasheet for the radio chipset clearly states that the RSSI figure provided by the RZUSB is the sum of all radio frequency power at this frequency not just the power transmitted [106].

In order to obtain a representation of an omni-directional antenna, one thousand received power measurements were taken for each of a set of orientations. In each set the physical orientation of either the transmitter node or receiver node had been altered with respect to the other. The conceivable combinations measured are depicted below in Figure 22 comprising of 18,000 total measurements. The resulting receiver orientations XY, ZY and YX have thus been measured for each possible transmitter orientation and with the combined set of measurements represent an average model of the plane as depicted. A similar approach was taken with the Tmote-Sky hardware modules by An et al. [100] however the author does not believe they took sufficient measurements to obtain statistically reliable results. They also measured the differing received signal strengths with orientation at a distance of 4 m as opposed to the industry standard 1 m; this means that they did not measure the maximum practical change in signal.

Once histograms for three model receivers were obtained (one each for XY, ZY and YX) these were further combined to provide a completely omni-directional model of an RZUSB Stick. The resultant single histogram at a 1 m separation is representative of the received radio frequency power independent of either the transmitter or receiver orientation. This is important given that in the event of deriving location from smart meters, the orientation of a smart meter is deemed unknown and it is convenient not to require specification of the orientation of the sensor.

The histograms and mean power values used to create the final omni-directional representation were all treated in percentage occurrences of particular received power values. This is because although one thousand measurements were taken in each of the eighteen orientation combinations, not every measurement set returned one thousand received beacon requests. The number of missed transmissions for each orientation combination was recorded, but paled in significance compared to the valid data so retesting any measurement set was not considered necessary.














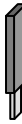

























A (Rx)	B (Tx)	A (Rx)	B (Tx)	A (Rx)	B (Tx)
 XY	 XY	 ZY	 XY	 YX	 XY
 XY	 XZ	 ZY	 XZ	 YX	 XZ
 XY	 ZY	 ZY	 ZY	 YX	 ZY
 XY	 YX	 ZY	 YX	 YX	 YX
 XY	 YZ	 ZY	 YZ	 YX	 YZ
 XY	 ZX	 ZY	 ZX	 YX	 ZX
 XY	Average of plane	 ZY	Average of plane	 YX	Average of plane

Figure 22: Orientation combinations for creating an omni-directional model

4.2.4. Results

Table 6 below summarises the measurements taken; for each orientation combination the thousand results occupied a spread of radio power values. The collated mean powers were then taken for further analysis.

Orientation Combination	Mean Power (dBm)	Lost Messages (%)
XY:XY	-48.0	-
XY:XZ	-47.9	-
XY:ZY	-40.8	-
XY:YX	-50.6	-
XY:YZ	-48.0	-
XY:ZX	-47.2	-
ZY:ZY	-45.0	-
ZY:ZX	-50.4	0.1
ZY:XY	-48.1	-
ZY:XZ	-51.0	-
ZY:YX	-48.0	-
ZY:YZ	-43.3	-
YX:YZ	-53.4	-
YX:YX	-46.6	-
YX:XY	-57.6	-
YX:XZ	-56.5	-
YX:ZY	-48.0	-
YX:ZX	-62.5	-
Omni	-49.6	0.006

Table 6: Mean power received for each orientation combination

The overall standard deviation of the mean powers in different combinations of transmitter and receiver orientation was 5.24 with a mean omni directional power of -49.6 dBm. This signifies a fairly distributed spread of possible received signal strengths at 1 m as can be seen visually in Figure 23.

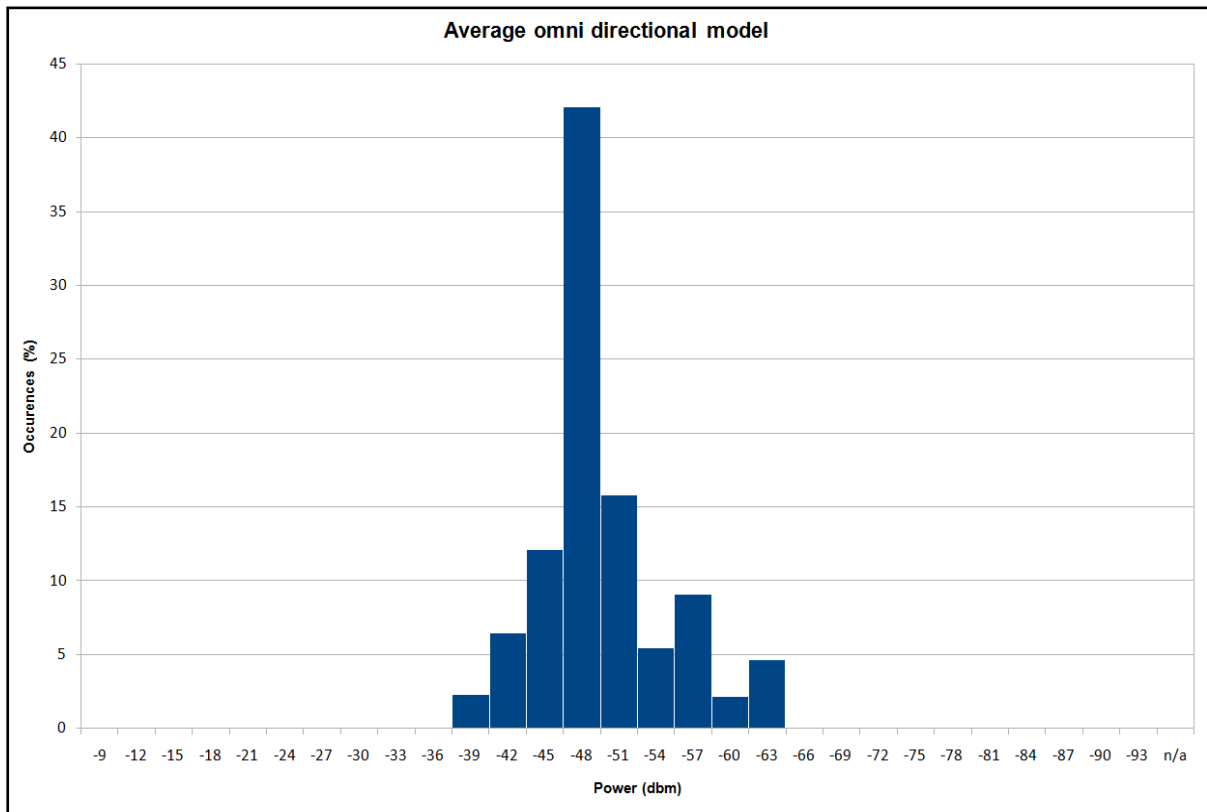


Figure 23: Histogram of the collated mean received power values

The mean power measured at 1 m ranged from -39 dBm to -63 dBm. Attempting to use this data for range derivation in a location derivation sense would return very little confidence in the devices' positioning. To understand this data a little more comprehensively and identify causes for the wide spread of mean values the data was analysed per plane (i.e. per each of the three orientations XY, ZY and YX for the receiving node – see the average planes depicted in Figure 22).

Figure 24 overlays the three resultant histograms of mean received radio power values per plane. It would appear that orientating the receiving node in the YX plane results in a significantly spread range of returned power values. The ZY plane has the most normal Gaussian response; however the XY plane of orientation has the strongest correlation to a singular mean despite having a skewed distribution.

Figure 25 shows an analysis of the effect of the orientation on the radio power measured by the receiving node. This highlights that the Z plane (aka a vertical RZUSB Stick) was the most omni-directional and strongest link for two nodes set at the same height. The Y plane (aka where the integrated circuits are facing outwards) provided the weakest returned power.

For the reasons displayed by these two analyses, for the remainder of testing throughout these studies the RZUSB Sticks were orientated vertically (in the ZY plane depicted in Figure 22) to obtain the most consistent results.

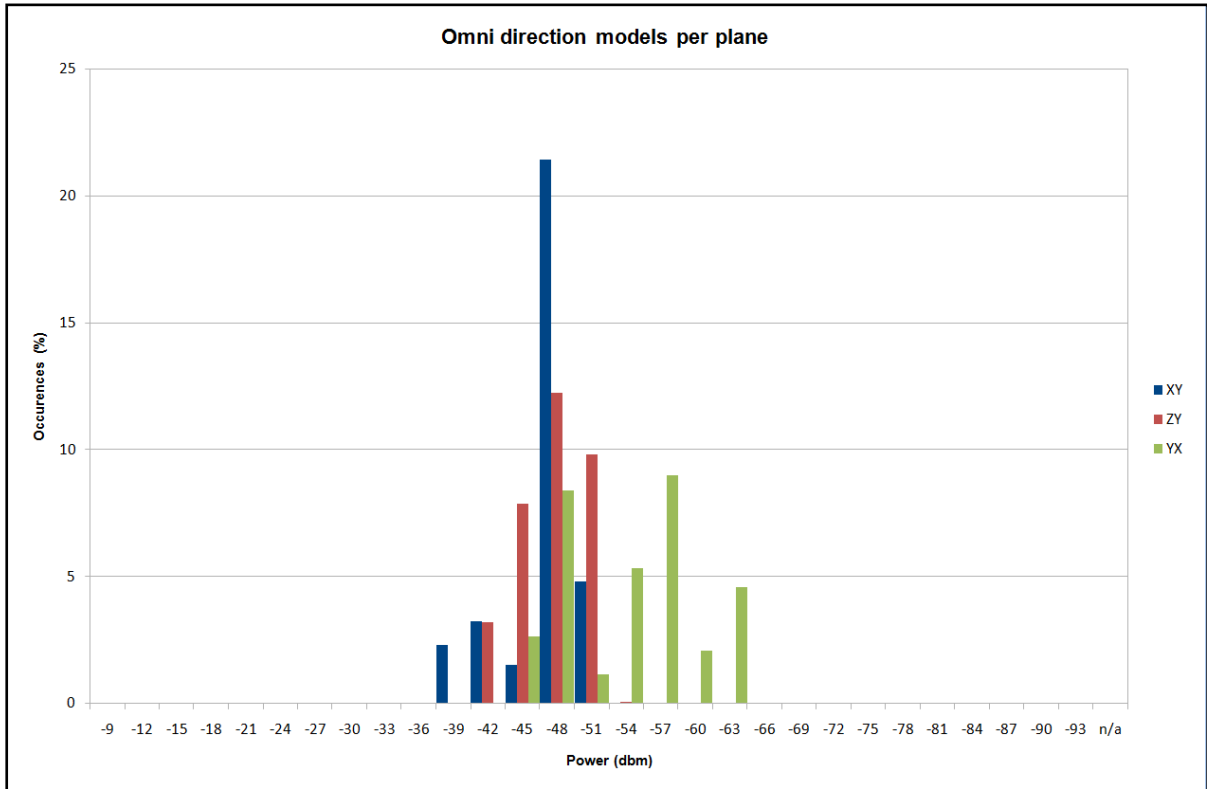


Figure 24: Histogram of the collated mean received power values split into the three receiver planes

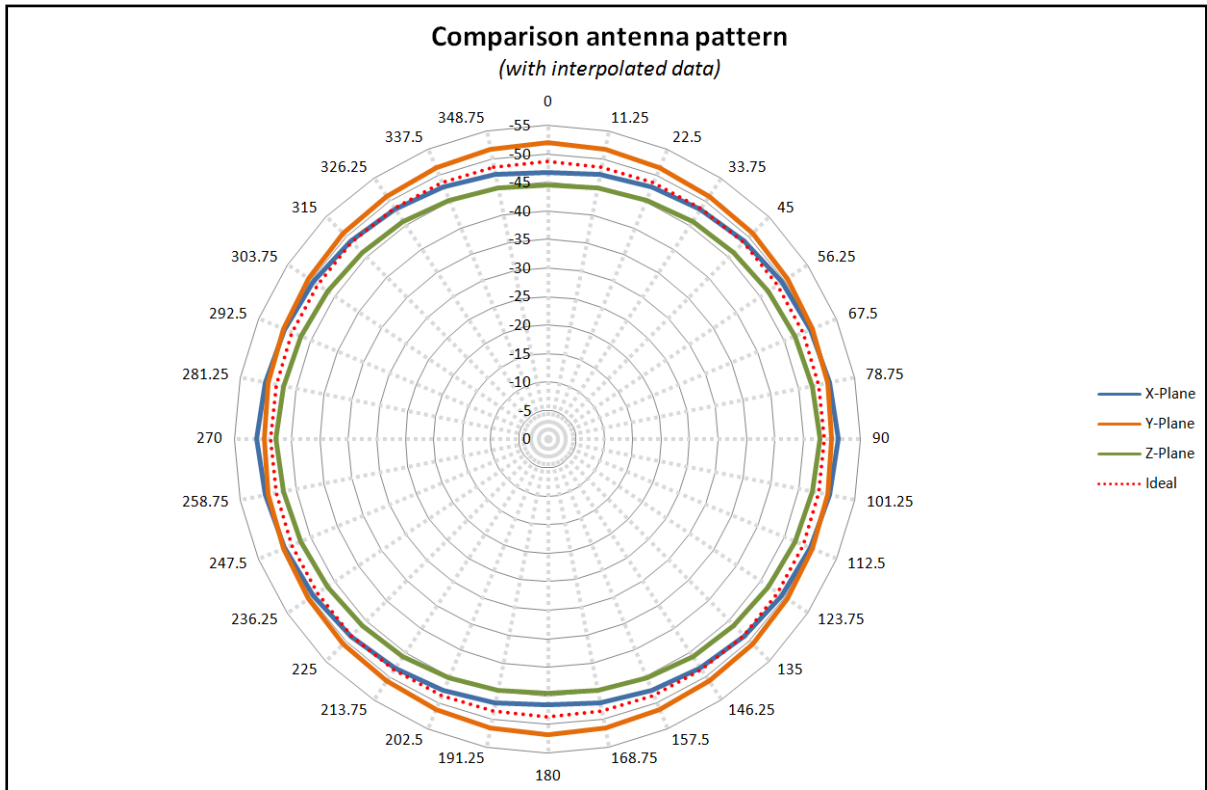


Figure 25: Polar plot demonstrating the directionality of the RZUSB Stick

Taking the omni-directional model for power at one meter (-48 dBm) and applying

the Friis Transmission Equation (Equation 3, presented in section 2.2) we arrive at the following:

$$[-48] = [3] + [Fade\ Margin] + 20\log\left(\frac{\lambda}{4\pi}\right)$$

Equation 5: Friis Transmission Equation applied to received power measured at 1 m in RF shielded laboratory

Thus, from rearranging Equation 5 we can obtain a value for the fade margin in this experiment as -10.67 dB to two significant figures.

Using Equation 5 **Error! Reference source not found.** and the transmitter / receiver specifications of the RZUSB Stick from its datasheets [101], [106] the following table was derived for transmission at a 1 m range:

	RZUSB Stick to RZUSB Stick
TX power (dBm)*	3
RX sensitivity (dBm)*	-101
TX gain (dB)*	0
RX gain (dB)*	0
Fade margin (dB)***	-10.67
Frequency of Channel 26 (MHz)**	2.48
Wavelength of Channel 26 (m)**	0.121
Path loss (dB)***	93.33
Range using Equation 2 (m)***	446.7
Range Friis Equation (m)***	446.3
* Provided by Atmel	
** Measured or controlled value	
*** Calculated value	

Table 7: Transmission characteristics of two RZUSB Sticks at 1 m separation

We can see from Table 7 that both range estimation models provide for very similar results albeit that Equation 2 is a somewhat simpler model to implement. Using these calculated values allows a relationship to distance to be approximated as shown in Figure 26:

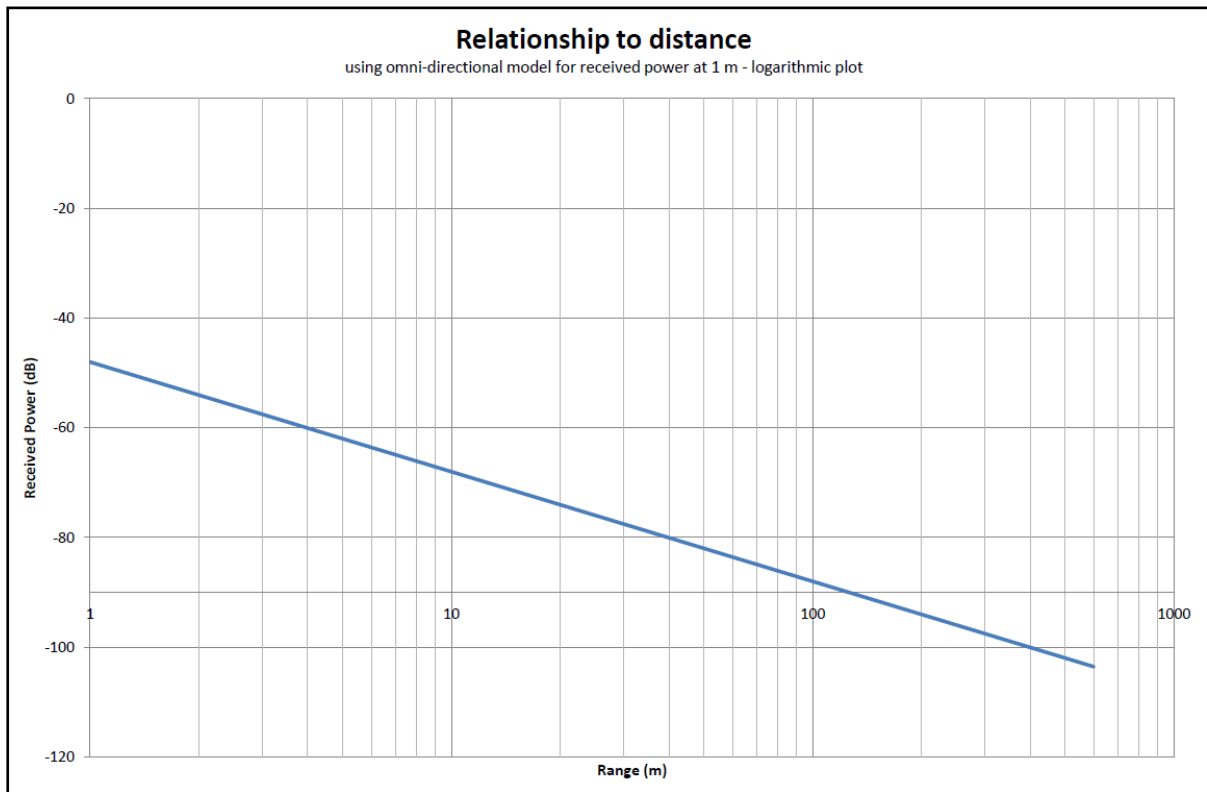


Figure 26: Received power vs. range relationship model using an omni-directional model of received power at 1 m

$$[Power Rx] = \frac{[Power Tx]}{4\pi[Distance]^2}$$

Equation 6: Inverse square law for electromagnetic radiation

This relationship appears to be a good logarithmic fit and can be checked against the inverse square law for radio propagation (Equation 6) by taking a few sample distances and calculating their returned power levels. As shown in

Range (m)	Calculated Received Power (dB)	Δ (dB)
1	-48.00	0.00
2	-54.02	-6.02
4	-60.04	-6.02
8	-66.06	-6.02
16	-72.08	-6.02
32	-78.10	-6.02
64	-84.12	-6.02
128	-90.14	-6.02

Table 8, the 6.02 dB delta per squared increase in distance accurately fits the inverse square law to which radio power propagation is assumed to adhere.

Range (m)	Calculated Received Power (dB)	Δ (dB)
1	-48.00	0.00

2	-54.02	-6.02
4	-60.04	-6.02
8	-66.06	-6.02
16	-72.08	-6.02
32	-78.10	-6.02
64	-84.12	-6.02
128	-90.14	-6.02

Table 8: Calculated Rx power at sample distances using the omni-directional model of received power at 1 m

4.2.5. Conclusions

This testing successfully concluded a distinct omni-directional distribution for received power at one meter. The antenna of an RZUSB Stick was shown to be broadly omni-directional but a best orientation for range measurement was also determined (ZY plane as discussed in the results).

Through doing this testing and resulting analysis a model relationship between range and received power has been calculated. This is fundamentally a model that could be used for performing a multilateration methodology to derive location based upon multiple received transmission powers.

A notable misrepresentation of the figures presented is that it would appear from Figure 26, and indeed the data used to calculate it, that an RZUSB Stick would be able to transmit at a power level of just 3 dB and receive messages over a distance greater than several hundred meters without really approaching the receiver sensitivity limit of -101 dB. The important point to remember in this instance is that this model was created as a representation of an antenna in free space; with obstacles, moisture and other factors contributing to poor quality, multipath environments this range will be significantly reduced. This is the reason why a Fade margin for this test set up was calculated to be approximately -4 dB as opposed to a more realistic -30 dB or so for a nominally good environment.

To improve this testing it would be ideal to undertake measurements in a manner more analogous to electromagnetic compatibility testing; utilising a well characterised and calibrated receiving antenna to measure the radiated emissions at the centre frequency relating to the channel chosen.

It would also be ideal to make use of automated turntables with small step sizes (perhaps just one degrees of rotation per thousand measurements) at both the transmitter and receiver end to fully and more precisely measure the directionality. This is again analogous to the radiated emissions testing for electromagnetic compatibility certification albeit at a single frequency. This approach was not undertaken here upon the merit of time; however it would be necessary for a truer representation of the polar response of the antenna.

Despite the suggested test improvements, it is expected that the results gathered are sufficiently detailed and accurate to inform the task of measuring signal strength against range in the next section (4.3). From this perspective the testing was a success and usefully demonstrated broadly omni-directional behaviour by the RZUSB Stick, meeting the requirements of the test.

The next section will attempt to reinforce and support this model through practical testing and comparison.

4.3. Measuring received signal strength with range

4.3.1. Purpose

Section 4.2 formulated a basic model to derive range from the radio frequency power measurement made with an RZUSB Stick. With knowledge of range from one or more networks more sophisticated location derivation methodologies than proximity can be used – for instance multilateration.

This section sets out to test the accuracy and effectiveness of the basic range model presented.

As discussed in the chapter introduction (section 4.1), there are two notable papers which argue that it is not possible to accurately determine range (and thus location via this measurement) between IEEE 802.15.4 network nodes:

- Benkic et al. [2] undertake range testing with three different radio modules, all of which however were discounted in Chapter 3 of this study.

Their testing took place in a highly multipath environment (an enclosed concrete corridor) and within the presence of multiple operating WiFi networks coexisting within the same band and upon uncontrolled channel allocations.

Benkic et al. took infrequent and low volume measurements during their testing – circa two hundred transmitted messages per two to five metre increments in range.

They additionally struggled with obtaining any reliable source of information for converting the received signal strength and link quality indicators into accurate power measurements.

For these reasons the author believes the testing undertaken was sufficiently compromised as to not support a reliable argument to the effect that received signal strength of IEEE 802.15.4 transmissions and range do not possess sufficient correlation to derive one from the other.

- Heurtefeux and Valois [3] similarly present that IEEE 802.15.4 networks do not provide sufficient received signal strength granularity for accurate range derivation. Instead of a measurement of signal strength between two nodes, their testing involved grids of two hundred and fifty distributed nodes each transmitting a message every thirty seconds, each of which will then record the received signal strength indicator of messages broadcast by neighbouring nodes.

As with Benkic et al., their test environment is indoors and they have additional wireless equipment operating in the vicinity (this time out of band however so a significant improvement).

Heurtefeux and Valois show that their test set up exposed issues with directionality and non-uniformity of individual links due in part to the hardware chosen. They utilised multiple different hardware platforms and transmission powers resulting in a deceptively large spread of returned average received signal strength indications.

Heurtefeux and Valois' models were based upon collaborative localisation which is not a transferable methodology to this thesis due to the third party aspect of smart meters and equivalent domestic networks. Regardless of this fact, they were still claiming error rates too great to be able to correlate signal strength to range.

The author intends to prove via testing that a basic inverse square law model presented in section 4.2 can loosely be applied to the measured radio frequency power of wireless sensor networks despite the assertions of these particular authors. The model demonstrated in the previous section correlates with widely accepted radio frequency theory and with the range estimation work undertaken with similar radio networks such as IEEE 802.11 WiFi networks.

Practical measurement of received radio power at several ranges would prove or disprove any fit to the assumed model. It should also prove or disprove both Benkic et al. and Heurtefeux and Valois' assertions that sufficiently granular range is not derivable from an IEEE 802.15.4 transmission received signal strength indication.

This testing represents the climax of the research study as successful range or location derivation based upon the signal strengths of third party wireless sensor networks has not been evidenced in the available literature.

4.3.2. Requirements

Successful testing would evidence a correlation (or lack thereof) between distance and signal strengths in IEEE 802.15.4 networks, where there is no connectivity association between the nodes.

The results of this testing should be comparable to the relationship model shown in Figure 26 where upon an inverse square law model was fitted to measurement data at one metre.

As before, this testing should strive to mitigate any risk of unintentional or collateral interception of unintended transmissions. In this instance this was doubly important so as not to skew the measurement of received signal strengths with that of measurements from uncontrolled and unreferenced network nodes.

Finally, the testing needed to be undertaken in a manner by which the short fallings of Benkic et al. and Heurtefeux and Valois' experiments are addressed. The methodology chosen drew from some of the synonymous studies undertaken with WiFi and BlueTooth technologies – both of which operate within the same frequency band.

4.3.3. Methodology

The testing methodology adopted was a conglomeration of three of the most applicable studies identified [2], [86], [87] and some alterations based upon standard industry practices for radio frequency emissions testing.

Ruiz et al. measured the received signal strength indications of several radio frequency identification (RFID) tags read by a reader at different ranges [45]. Kotanen et al. used calculated received signal strengths of Bluetooth to attempt a correlation with range [86]. Finally, and most akin to this study, Benkic et al. [2] used the received signal strength indication values to correlate with range using IEEE 802.15.4 networks. Regardless of the technology used, all three utilised the same base methodology to determine their respective correlations. The exact set up of each study's experiments differed however and it is from a comparison of these differences that the methodology for this experiment shall be chosen.

The following experiment parameters were chosen based upon a comparison of the methodologies:

4.3.3.1. Experimental parameters

4.3.3.1.1. Elevated height

Benkic et al. do not define node elevation in their paper but it is assumed that they at least kept this height constant. Ruiz et al. mounted their tags at a height of 2 m upon walls and Kotanen et al. had theirs mounted at 0.75 m high.

As opposed to the of 0.75 m elevation used by Kotanen et al., a delta from ground of 1 m was used in this study. As mentioned in section 4.2 is the default industrial standard height above ground when measuring radiated emissions and so is a suitable choice for this experiment. 1 m above ground ensures a sufficient separation from the natural ground plane and mitigates measurement fluctuations from varying moisture content within the ground's substrate.

In this testing the base of the RZUSB Stick was mounted 1 m above the ground by being taped to a measured wooden dowel. The construction of the dowel was chosen for its minimal interference with radio waves.

4.3.3.1.2. Surroundings

In all three of the studies discussed, measurements were taken indoors in a non-ideal environment. This study undertook these measurements outdoors in the

centroid of a large flat playing field to achieve as close to a free space environment as possible. This had the additional benefit in this instance of sufficiently distancing both of the nodes from any surrounding equipment that may have been operating in this band – an advantage both for improving the signal to noise of the band (and so experimental reliability) and mitigating any risk of collateral intrusion or denial of third party networking.

A mown grass area was chosen to maximise the anechoic characteristics of the surface and so reduce multipath reflections.

4.3.3.1.3. *Measurement samples per increment*

As mentioned earlier in section 4.3.1, Benkic et al. took very few samples. At each increment in range they measured 270 samples and used a statistical poison analysis to present 210 samples per range. This represented a total of just 1,260 to 1,890 measurements per hardware module investigated (each device tested was measured over increasingly fewer range increments).

In contrast, Kotanen et al. transmitted one thousand messages between two Bluetooth nodes at varied separation distances. The statistical robustness of the data collected negated the need for a statistical sub-sampling approach however would have been more time consuming to undertake. Kotanen et al. performed a total of 65,000 measurements.

Ruiz et al. were using 71 radio frequency identification tags each of which transmitted a burst every second for a minute. With all the tags spread across 32 separation distances and each range measured independently for a minute the total experiment represented a little over 136,000 transmissions. As a result of the time based approach and simultaneous transmissions of at least two tags per range, Ruiz et al. suffered somewhat from a high transmission error volume with a low volume of transmitted messages being received (just 46,687 – 34%).

This testing adopted the statistically strong one thousand measurements per separation distance and additionally the resolution of measurement was increased with comparison to any of the studies; totalling 100,000 measurements. With less than 0.01% of transmissions recorded as not detected this signifies a significant increase in data strength compared to other studies of similar technologies. The author believes that this was only possible as a result of the high levels of scripting and automation developed for the pursuit of this experiment.

4.3.3.1.4. *Delta range step size*

Ruiz et al. do not directly state the distance spacing used during their testing, however they do state that they took measurements at 32 positions along their main corridor which is on their map could appear to be approximately 32 m in length implying a separation distance interval of 1 m. Their radio hardware operated at 433 MHz as opposed to at 2.4 GHz as per the other two studies and this investigation; in

this sense, Ruiz et al.'s testing can be expected to span the greatest maximum range for similar power transmission strengths.

Benkic et al. start their experiments with a 1 m interval spacing however this then increases to 2.5 m and later 5 m separations. On their subsequent radio modules tested they begin with 2 m intervals and increase to 5 m intervals at ranges greater than 10 m. Their maximum distance measurements vary from 20 m to 25 m dependant on the radio module investigated. Although their experimental process was inconsistent and seemingly disorganized it did lend weight to the assumption that they were refining their process as they went along. The author assumes that they decided that a non-linear separation distance was suitable for an expected logarithmic fit and this principle was taken forward in this experiment.

In comparison to the other two papers, Kotanen et al. spaced their measurements at a much finer resolution of 20cm intervals from 0.2 m to 13.0 m. From their results they had a reasonable fit to their predicted propagation model above 4 m; however they had a very poor fit between 0.6 m and 4 m. At the smaller ranges most measurements returned the same value as the received power levels were all within the wide dynamic range of the Golden Receive Power Range which distorted the results. This is not something that would affect the RZUSB Stick hardware as this is a feature of the Bluetooth protocol.

This study blended the high resolution measurement attempt of Kotanen et al. with the non-linear principle used by Benkic et al. and then further improved upon both: From 0.1 m to 7.0 m a 10 cm increment was measured and from 7.0 m to 22.0 m a 50 cm increment was measured combing to the total of 100,000 measurement sets. In this manner 70% of the measurements should be concentrated upon the most dominantly changing part of a logarithmic fit.

4.3.3.1.5. *Radio module type*

Benkic et al. were the only team to try and mitigate the response of a single transceiver type by profiling three different sets of radio hardware. Unfortunately, doing so seemed to reduce the effort available for effective and robust testing of any single module. For this reason it had been decided to concentrate efforts purely upon the RZUSB Sticks identified in Chapter 3.

Where Kotanen et al. primarily struggled was in the resolution and precision of calculated received signal strength. For Bluetooth devices such as the ones used in their experiments the reported power values are a proportional representation of the Golden Receive Power Range. This is used as a part of the Bluetooth standard to dynamically adjust the transmission powers and conserve power consumption. As a result Kotanen et al. suggest an enhancement to their methodology would be to use a device that can measure received power level directly – the RZUSB Stick chosen for this study does just that.

4.3.3.1.6. *Antennae gain*

Likewise to Kotanen et al. (and by expectation Ruiz et al.) similar antennae gains were used in this experiment for the receive and transmit nodes (both being the same design of printed circuit board trace antenna and from the same batch of RZUSB Stick devices). The same specific unit was used as either transmitter or receiver throughout the testing to ensure consistency.

4.3.3.1.7. *Antennae orientation*

Ruiz et al. were the only team to make note of controlling or mitigating antennae polarity and directionality. To mitigate the effects they took measurements at each distance in four receiver orientations.

In this study the author took heed of the significance of this fact to direct the previous testing (section 4.2) such that the tightest defined response and most omnidirectional orientation could be chosen. As mentioned previously this orientation was with the RZUSB Stick in a vertical position.

4.3.3.2. **Test setup**

As alluded when discussing the surroundings and environment for the testing, this experimentation was undertaken in the centroid of a large playing field away from all buildings, objects and people. 1000 measurements of received radio power were taken at a series of 100 different separation distances from just 10 cm to a maximum of 22 m. The testing setup is shown in the diagram and photos below (Figure 27 through to Figure 31); during testing the laptops were both closed and spaced a minimum of 1 m laterally from the RZUSB Sticks, out of line with the communication.

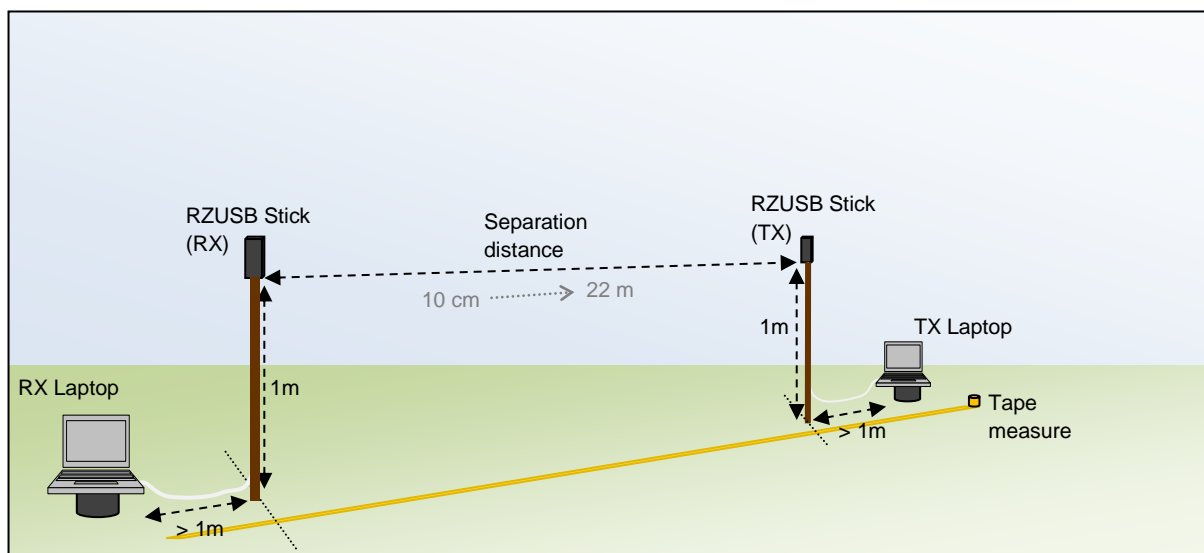


Figure 27: Test setup to determine the relationship between the received radio power of an IEEE 802.15.4 network and distance

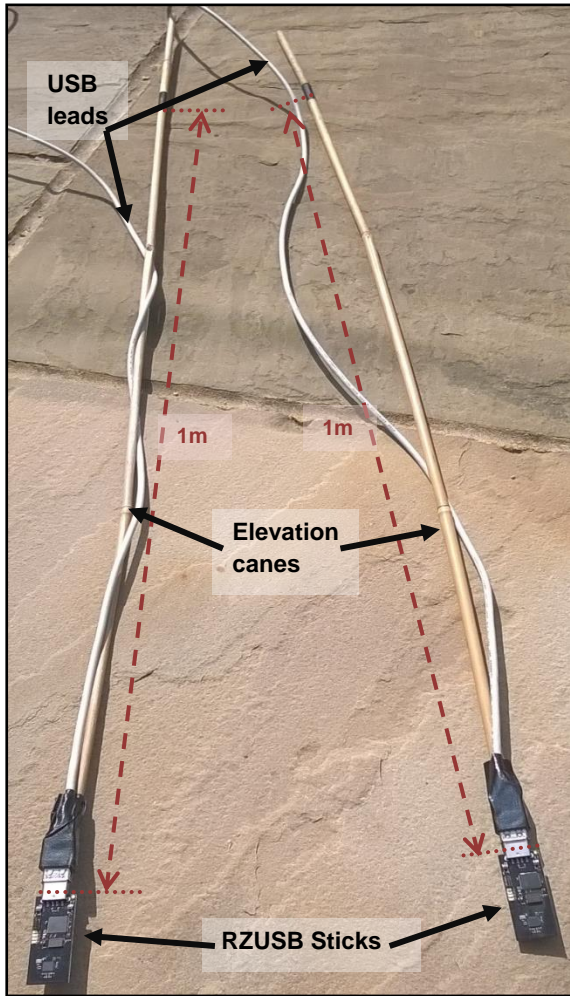


Figure 28: RZUSB Sticks mounted upon canes to elevate to 1 m above ground

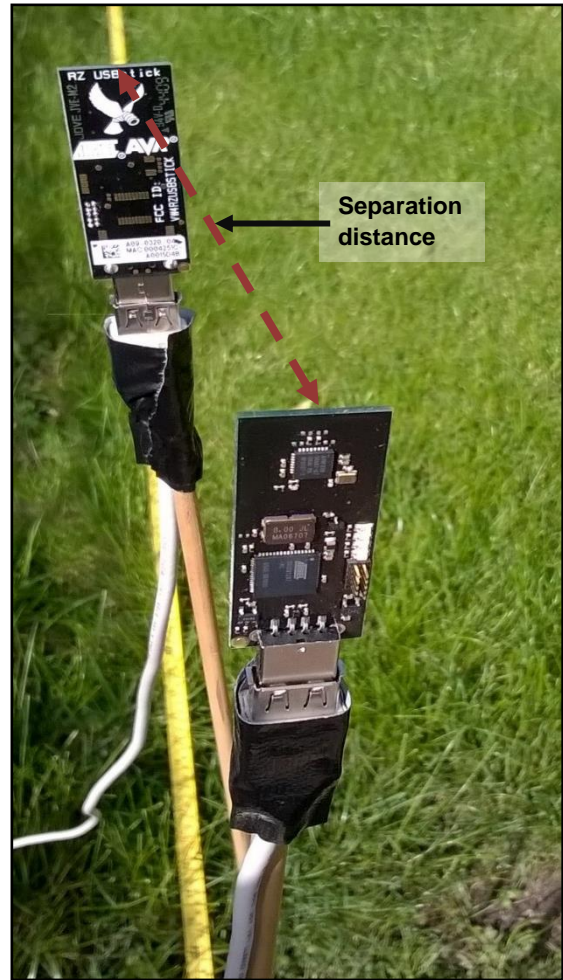


Figure 29: Orientations of the RZUSB Sticks during very close range measurement (RX in foreground, TX behind)

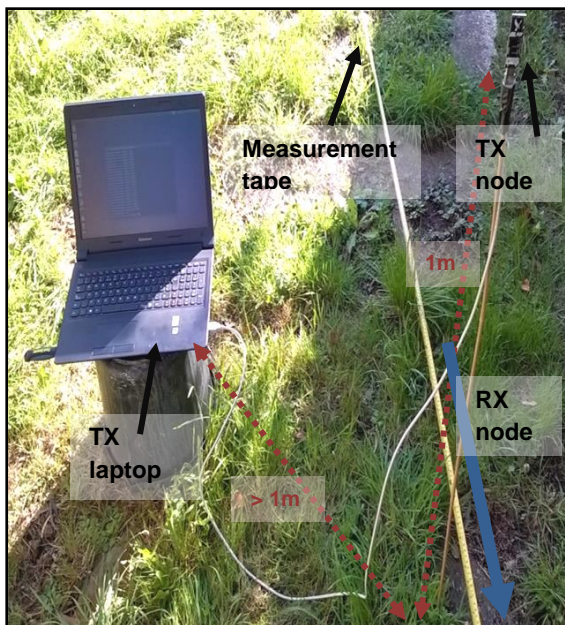


Figure 30: TX node during configuration

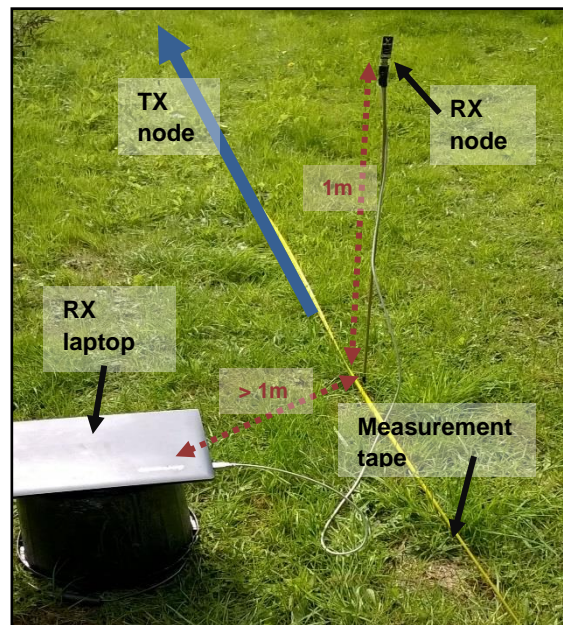


Figure 31: RX node being set up at 540 cm separation distance

The same script (TPRange, described in Appendix 7 section 12.2.2) and process was used as that in the preceding investigations in section 4.3). This script automates the transmission and logging of a defined number of messages and records a received signal strength indication for each message. As before, the script utilises formulae provided in the Atmel documentation [106] to convert the indication value into a power level.

The automated nature of the scripts and the designed in mechanisms for the operator to remove themselves from the area prior to measurement (a countdown before starting transmission and a loud audible tone on completion) allowed for the mass collection of data undertaken. Even so this testing spanned several days of effort and as such one variable that required a best efforts approach to maintaining constant was the atmospheric weather.

4.3.4. Results

A separate comma separated values file was made for each measurement set of 1000 transmissions at a given transmitter / receiver separation distance. Each of these files were imported into Microsoft Excel to create individual histograms and missing data statistics per separation distance.

The resulting data was conglomerated to allow statistical analysis of the data, potential errors and any relationships received power may have to distance.

4.3.4.1. Data presentation

Kotanen et al. presented their data in the simplest manner of the three studies by plotting mean received power levels (a derived metric) against range on a linear scale - see Figure 32.

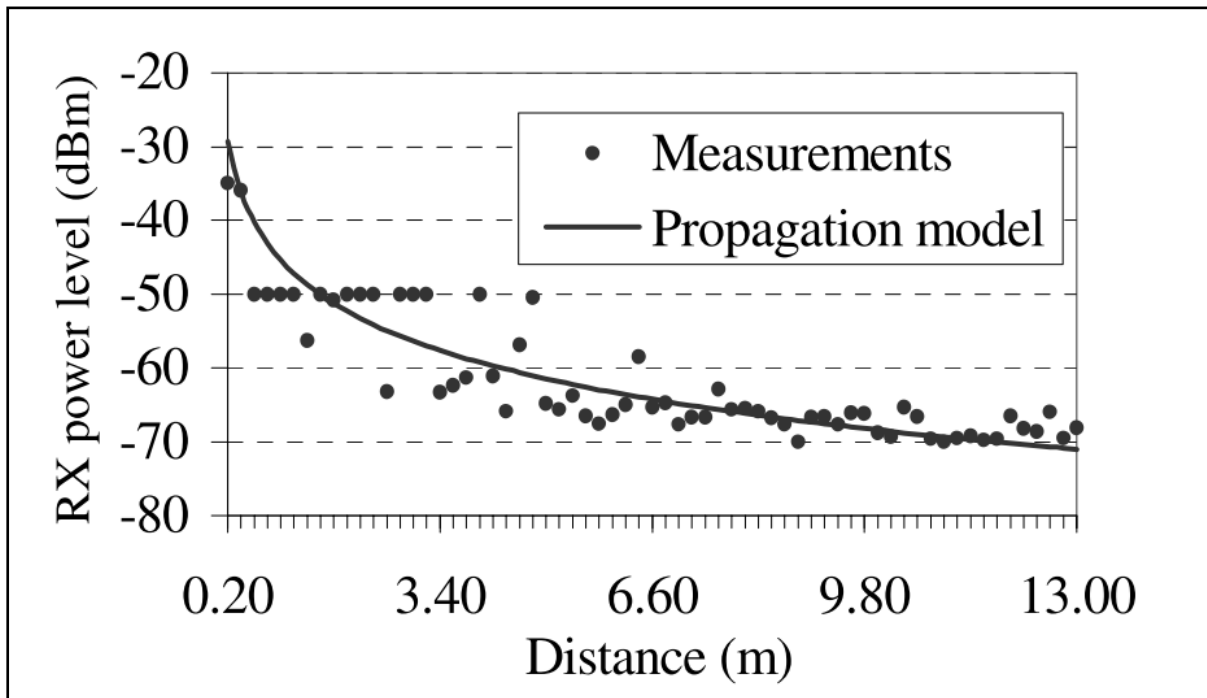


Figure 32: Kotanen et al. presented received Bluetooth power vs. range data in a straight forward manner [86]

By doing so an easy comparison could be made against their propagation model. It was additionally easy to identify their problematic where all the results at less than 4 m returned as -50 dBm (i.e. zero delta from the median of the Bluetooth Golden Receive Power Range).

In contrast, Benkic et al. abstracted their data representation from the correlation by plotting received signal strength indication against measurement number (from which range can be derived) – see Figure 33.

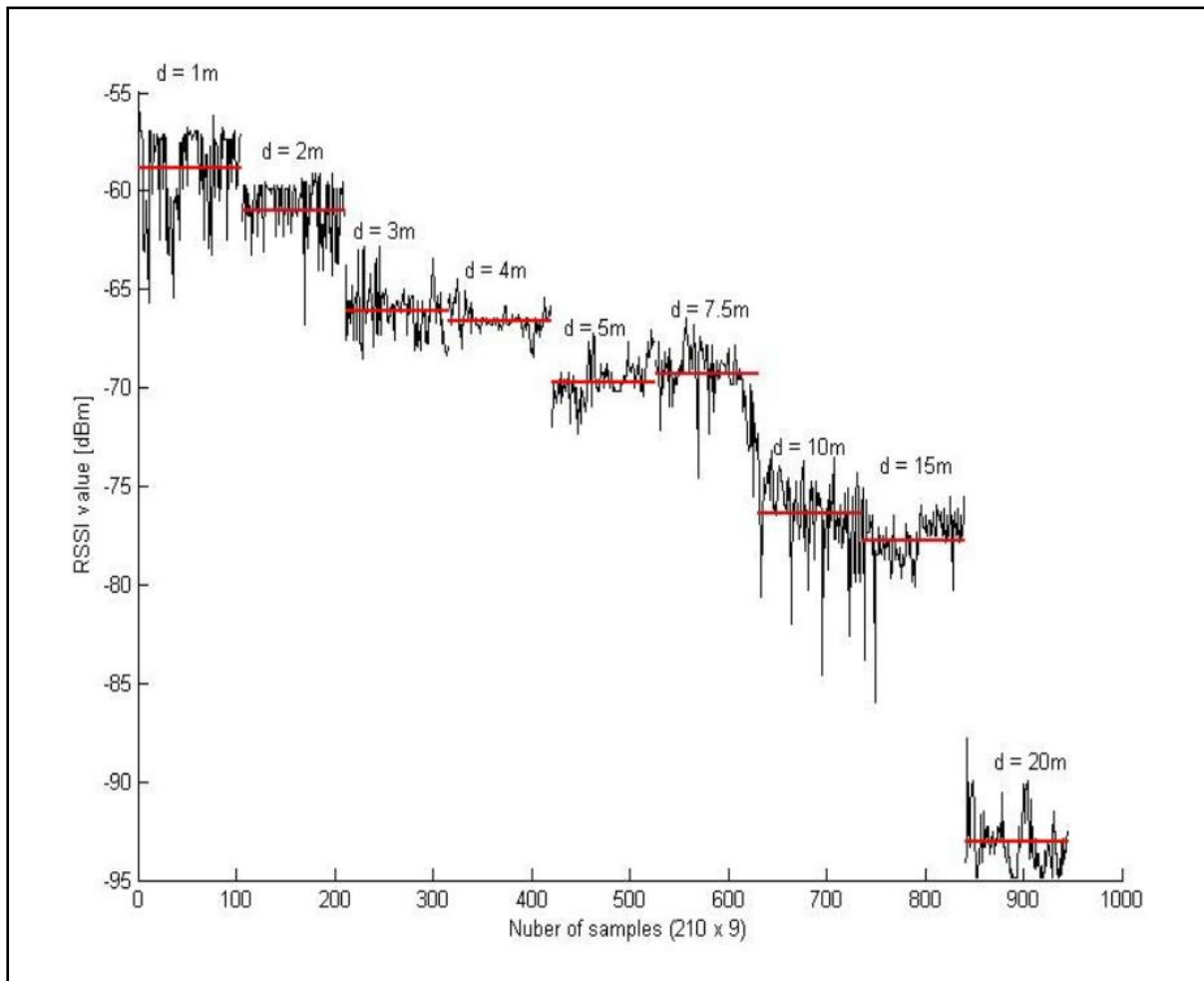


Figure 33: Benkic et al. present their wireless sensor network signal strength vs. range in an abstracted manner [2]

Although somewhat harder to comprehend, this format does provide for an appreciation of the spread of data collated at each measurement range.

Perhaps best, was the presentation of radio frequency identification tag received signal strength vs. range by Ruiz et al. – see Figure 34. They effectively plotted a series of histograms representing the distribution of received signal strength at each range. This was additionally plotted as a heat map for a different perspective of the same information.

The combined view provided by Ruiz et al. transfers a complex understanding of the correlation and spread of the data in a simplistic manner.

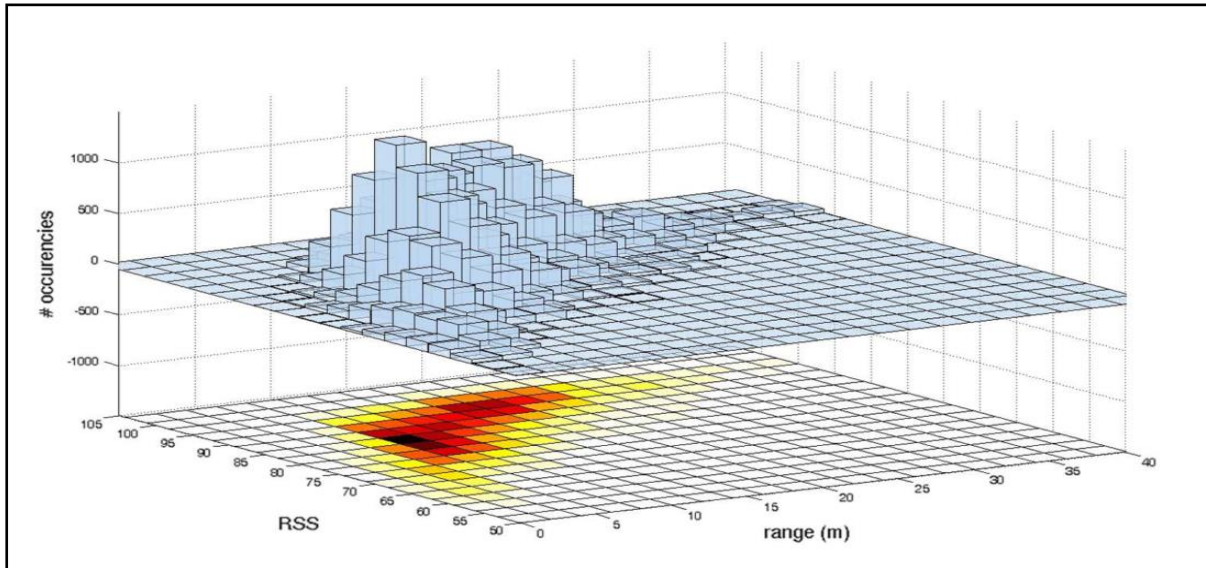


Figure 34: Ruiz et al. present their RFID signal strength vs range data in this format [45]

Attempting to achieve Ruiz et al.'s presentation style using the raw data obtained for this study was not as easy as there were too many discrete measurement intervals to comprehend. The heat map style of presentation did add benefit; however Microsoft Excel was not able to handle the full amount of data in this form. Instead, somewhat of a blend of the three approaches was taken and is represented in Figure 35.

This plot simultaneously represents the overall trend of the received radio power vs. separation distance and the spread of the received radio power received at each distance. Darker colours in this plot represent a greater concentration of measurements resulting at a single received power level.

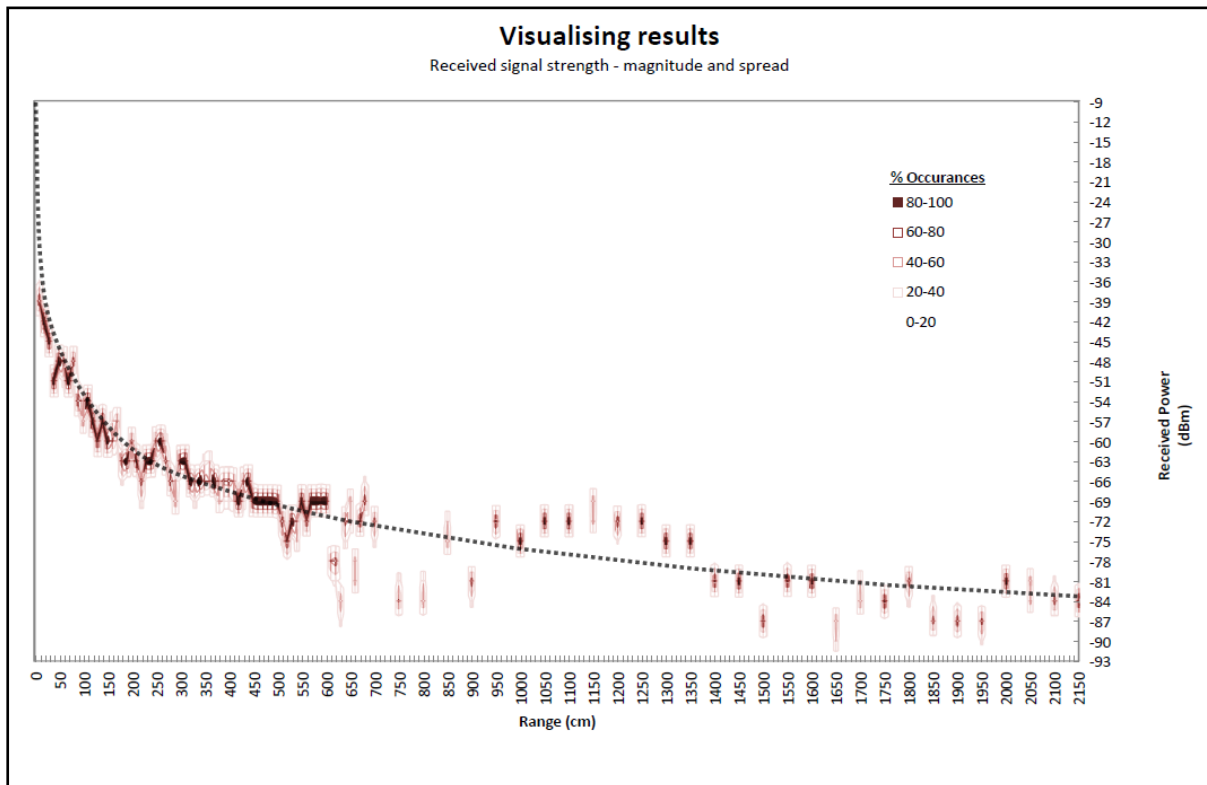


Figure 35: Plot showing the basic trend of received power vs. distance and also the spread of measurements at each range

From an initial consideration of Figure 35 the results appear to adhere to the expected logarithmic decay predicted in section 4.2. It also appears that below 6 m is a relatively tight fit to a trend, whereas measurements at greater than 6 m evidence a looser correlation. Both of these aspects shall now be explored in more detail.

4.3.4.2. Relationship between power and distance

Figure 36 shows a plot of the mean received power at each separation distance. This provides for a simpler representation of the data as per Kotanen et al.'s graph. Because of the increased simplicity of representation it was also possible to include metrics for expected error of measurement.

Figure 37 shows the exact same information but on a logarithmic scale for range. Given the good fit to a straight line on a logarithmic scale this reflects a good measured approximation to the inverse square law as anticipated.

In both Figure 36 and Figure 37 there are two red bounding lines ± 3 dB from the trend line; this represents the measurement uncertainty of received power based upon the RZUSB Stick received signal strength indication step size. For each measurement an integer received signal strength indication value is reported in step sizes of 3 dB.

In Figure 36 the horizontal axis scale is too granular to represent the horizontal error; Figure 37 however does represent an expected positional error bar (+/- 1 cm) where visible at the lower end of the scale.

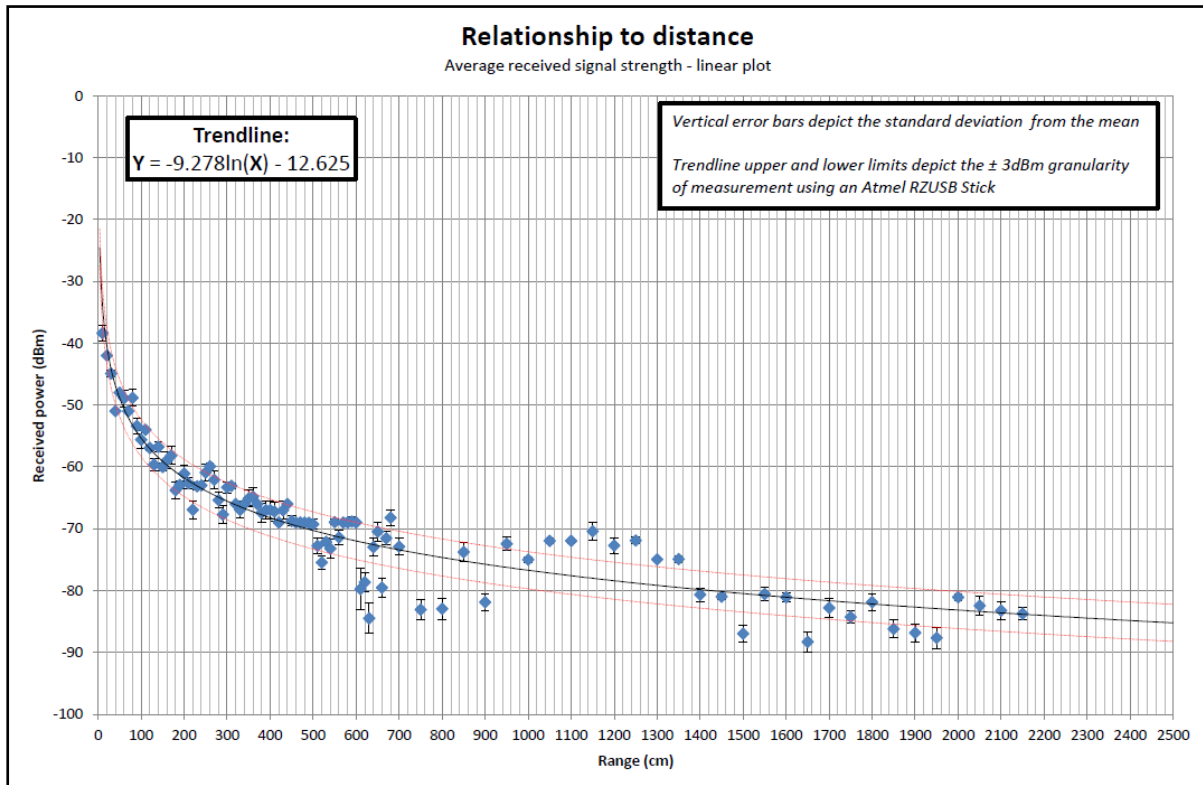


Figure 36: Received power vs. linear range with error metrics

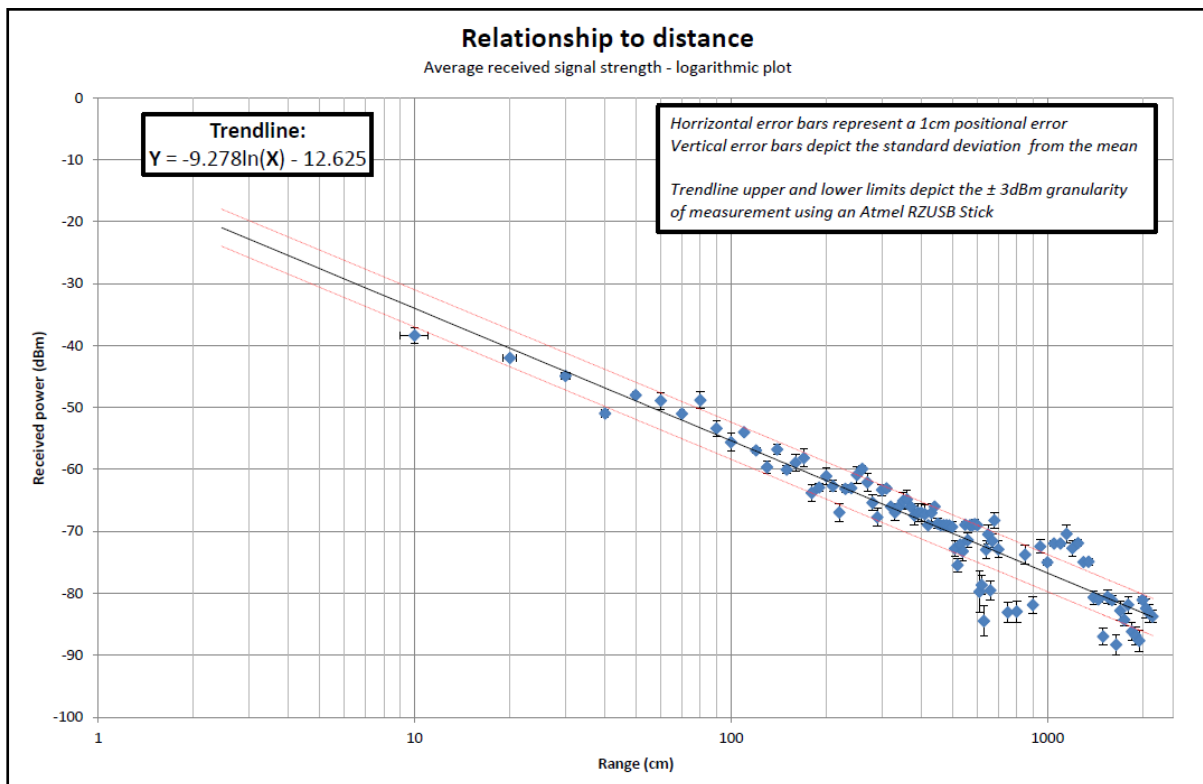


Figure 37: Received power vs. logarithmic range with error metrics

It is possible to see from both of these graphs (Figure 36 and Figure 37) that below 0.6 m all of the mean received power results would appear to lie within the bounding lines of the trend and so represent a strong logarithmic correlation between distance and received power.

Above 0.6 m however, although the trend appears to continue to match the expected inverse square law model, the data would appear to become less reliable. The majority of mean data points actually lie outside of the bounding lines, but with no clear direction or shift. The possible reasons for this increasing variance will be further examined in section 4.3.4.3.

In their experiments with Bluetooth, Kotanen et al. present the following relationship between received signal strength of a radio transmission and the distance between transmitter and receiver:

$$[Distance] = 10^{\left(\frac{[Tx Power] - [Rx Power] + [Tx Gain] + [Rx Gain] - [Random variance] + 20 \log(\lambda) + 20 \log(4\pi)}{10 \times [Environmental coefficient]}\right)}$$

Equation 7: Kotanen et al. present this model for relating received power and range [107]

Kotanen et al.'s relationship is derived from and hence similar to the Friis Transmission Equation presented in section 2.2, however they have notably accounted for the environment (i.e. number of walls and obstacles) and normally distributed random noise, the sources of which they do not elaborate upon and the value of which they ignore in their studies.

Figure 38 displays the measured results of this experiment set against the three different relationships: Friis Transmission Equation (Equation 4), the inverse square law (Equation 6), and Kotanen et al.'s relationship (Equation 7).

In Kotanen et al.'s relationship shown in Figure 38, the RZUSB Stick parameters (presented in Table 7) were used representing 0 dB gains and losses, additionally an environmental coefficient of 2.0 was applied for a perfect representation of free space [107] (values of 3.0 to 5.0 may typically represent an urban or semi-urban setting as defined in section 1.5).

A second version of Kotanen et al.'s relationship is also shown in Figure 38 where the parameters have been tweaked to best fit the results measured. The new parameters represent a value of 2.125 for the environmental coefficient (still a very close approximation to free space) and a value of -24.5 dB to account for gains, losses and the random variance.

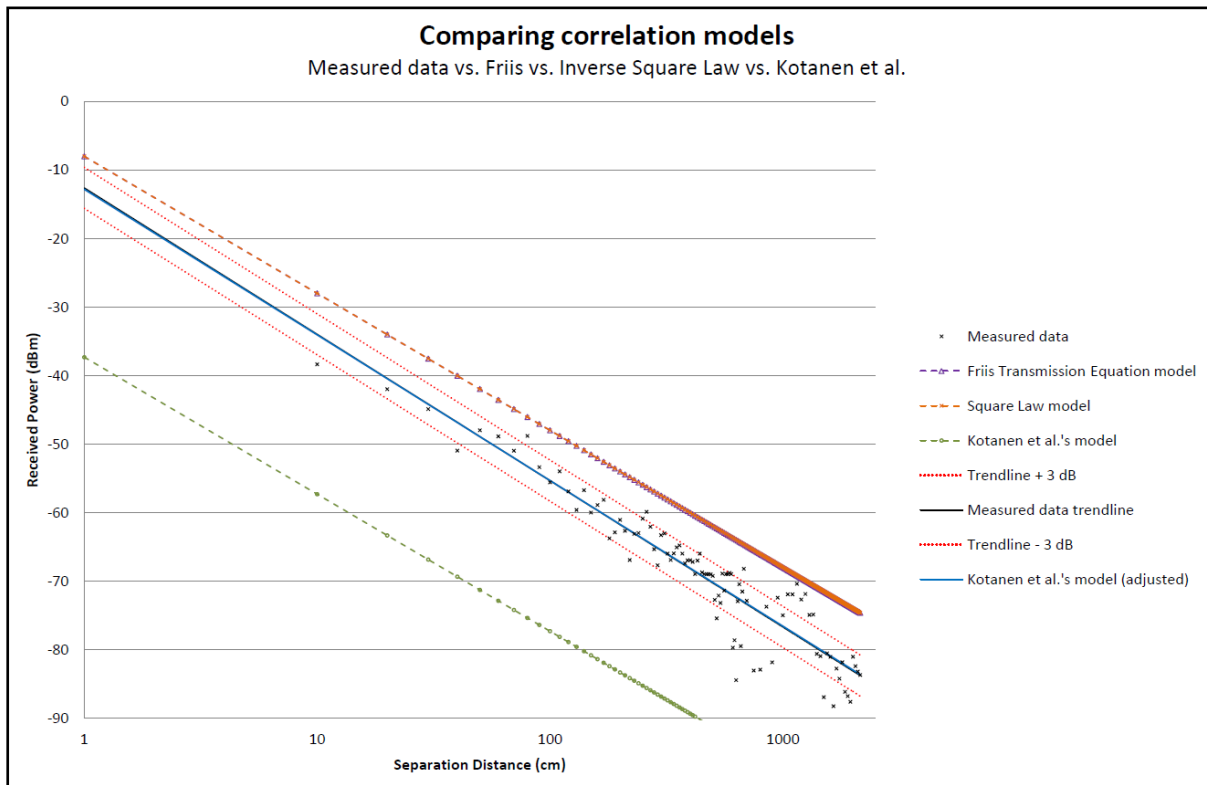


Figure 38: Comparing the measured results to predicted relationship models

It can be seen from Figure 38 that there is a clear similarity between the measured data and the previously presented relationship models. There is no scope for correction using the Friis Transmission Equation model or inverse square law model; an adjusted version of Kotanen et al.'s relationship represents an extremely good fit to the measured data however. Thus the relationship between received signal strength and separation distance of two RZUSB Sticks as determined by experimentation is:

$$[Distance] = 10^{\left(\frac{[Rx Power] - 21.5 + 20 \log(\lambda) + 20 \log(4\pi)}{21.25}\right)}$$

Equation 8: Relationship between received signal strength and distance for two RZUSB Sticks

Another aspect to consider as a relationship of the measured received power levels to distance, is the spread of received signal strength values returned at a single distance. It could be anticipated that measurement error and the logarithmic response of the relationship observed would compound to provide an increasing spread of measured values proportional to the separation distance. Figure 39 shows however that this is not the case and that there would appear to be no strong correlation between the standard deviation of received power at each separation distance and the magnitude of the separation distance at which the values were obtained.

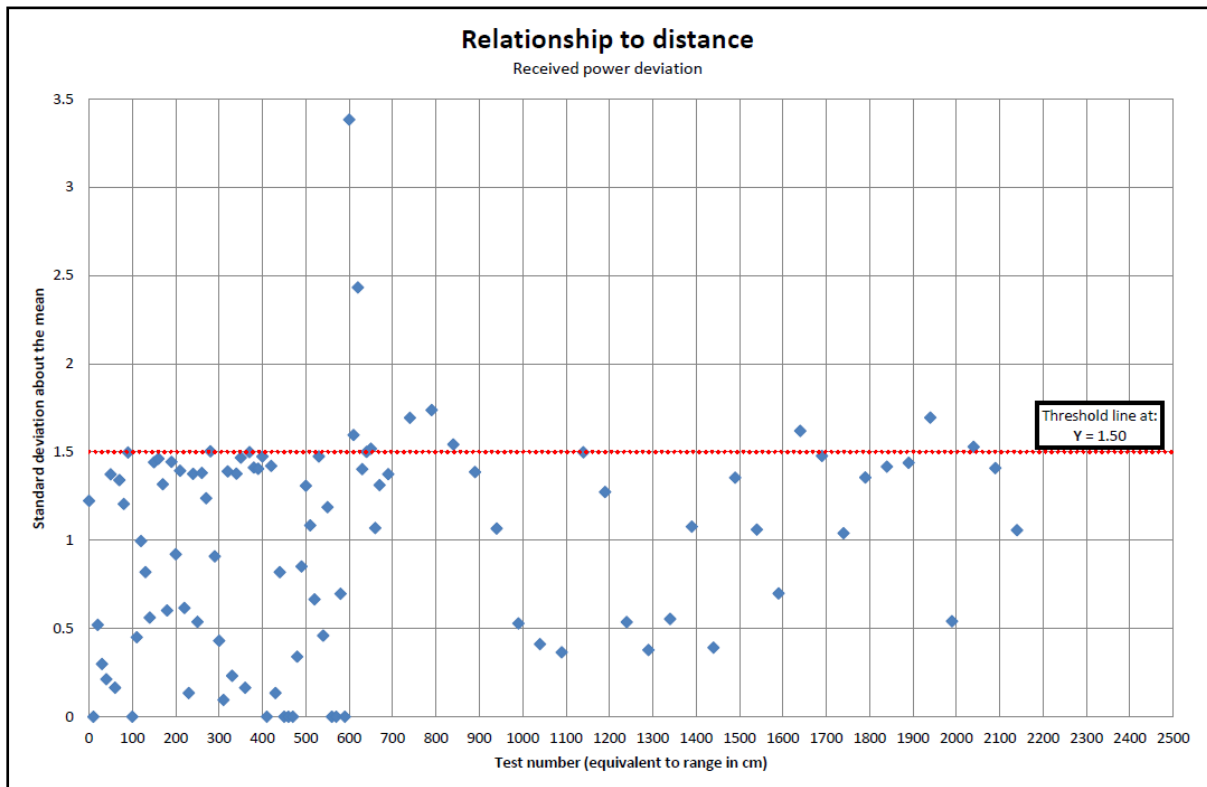


Figure 39: Standard deviation of received power about the mean vs. distance

4.3.4.3. Analysing sources of error

The Y-axis error bars shown in Figure 36 and Figure 37 represent the standard deviation of the measured data for that separation distance measurement set about the mean of that same data set. It is possible to observe that the greatest magnitude Y-axis error bars appear to coincide with the data points that fall outside of the bounding lines ± 3 dB about the trend. This correlation is supported by cross-referencing Figure 39 which more clearly depicts the greatest standard deviations and at which separation distances these occurred.

A suggested data validation would be to re-measure those measurement sets which experienced an abnormally high standard deviation to assess whether a random error had occurred distorting the results. These are those data sets seen in Figure 39 as sitting above the 1.5 standard deviations limit depicted – this limit was chosen as to represent the worst 15 % of data sets.

Alternatively, it could also be the case that

In both cases random error such as fluctuations in the test environment or observational / human error could have been a potential cause.

Further analysis has been undertaken to rule out more systematic errors (for example: consistently erroneous or uncalibrated equipment, poor experimental control or theoretical simplifications):

- Table 9 shows the measurement sets suspected worthy of retesting based upon either their magnitude of deviation from the mean trend line or the significance of their standard deviation. In total, this represents 31 % of the total number of data sets collated which represents a significant investment in time if they all were to be re-measured.
- Figure 40 shows two plots depicting the values from Table 9 plotted against range to try and determine any correlation between outliers and increased separation distance of the network nodes.
- Figure 41 shows the values from Table 9 plotted against one another to determine any correlation between magnitude of deviation from the mean trend line and the significance of their standard deviation.

Distance of measurement set(m)	0.1	0.4	0.8	2.2	2.6	2.9	5.2	6.1
Delta from mean RSS 3dB limits (dBm)	-1	-1	1	-1	1	0	-1	-5
Standard Deviation about RSS mean	1.22	0.30	1.34	1.39	0.54	1.50	1.08	3.38
Distance of measurement set(m)	6.2	6.3	6.6	6.8	7.5	8	8.5	9
Delta from mean RSS 3dB limits (dBm)	-4	-9	-4	2	-6	-5	0	-3
Standard Deviation about RSS mean	1.60	2.43	1.52	1.31	1.69	1.74	1.54	1.39
Distance of measurement set(m)	9.5	10.5	11	11.5	12	12.5	13	13.5
Delta from mean RSS 3dB limits (dBm)	1	2	3	5	2	4	1	1
Standard Deviation about RSS mean	1.07	0.41	0.37	1.50	1.27	0.54	0.38	0.55
Distance of measurement set(m)	15	16.5	18.5	19	19.5	20.5	22	
Delta from mean RSS 3dB limits (dBm)	-4	-4	-1	-1	-2	0	0	
Standard Deviation about RSS mean	1.35	1.62	1.42	1.44	1.69	1.53	3.50	

Table 9: Measurement datasets with mean received power values potentially worthy of re-measurement

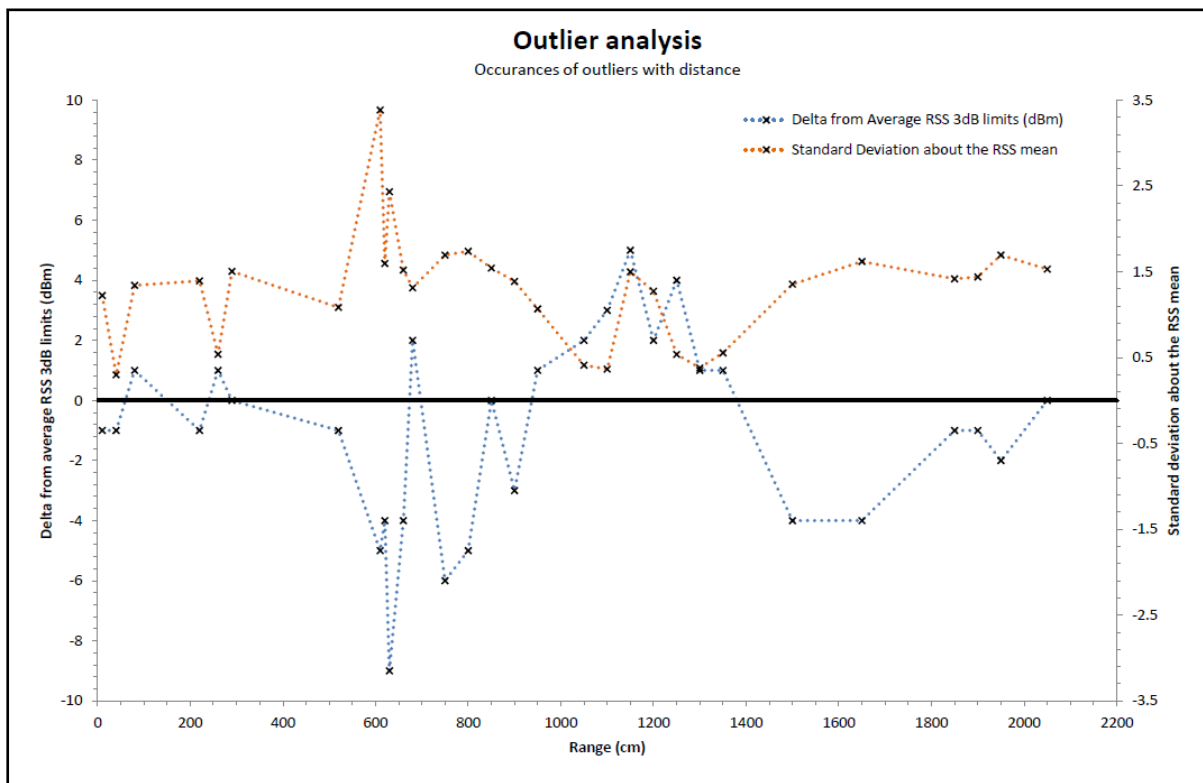


Figure 40: Assessing links between range and the delta from mean or standard deviation

The presence of random error appears to be supported by this evidence as there appears to be no obvious correlation between separation distance and neither the magnitude of departure from the trend, nor the spread of received power values measured.

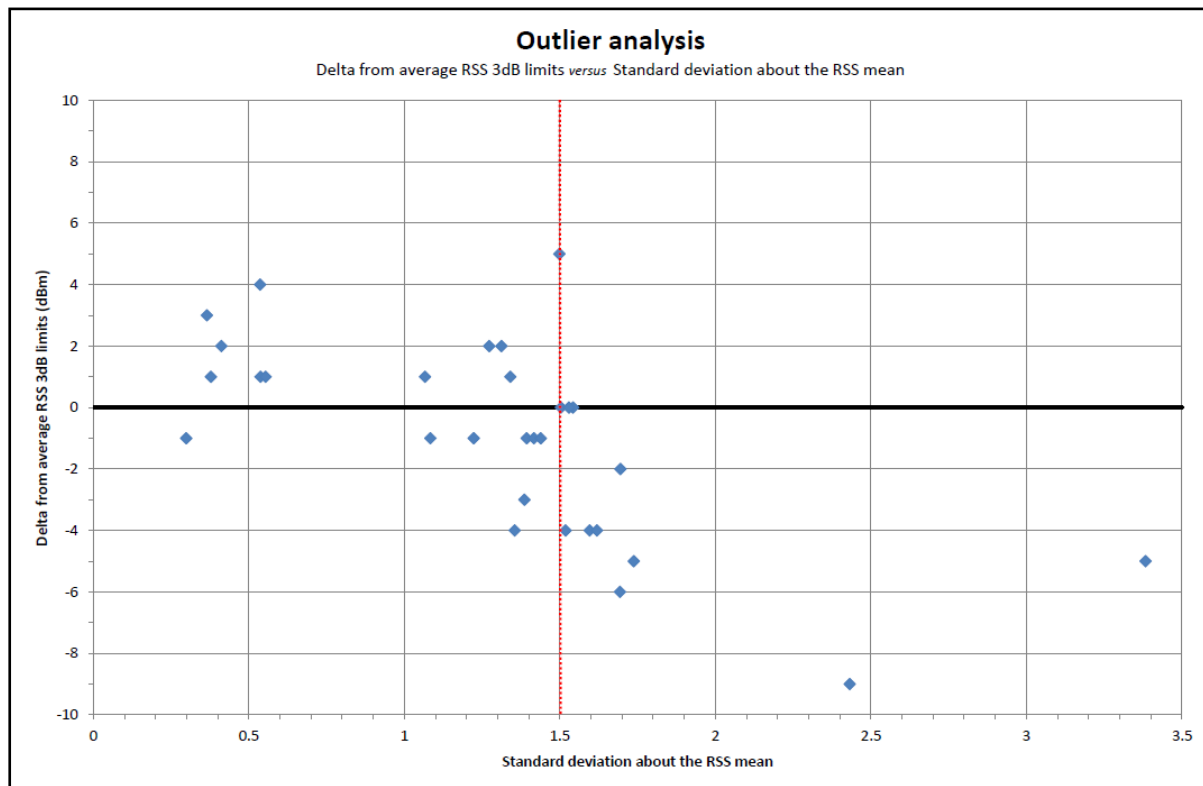


Figure 41: Assessing links between standard deviation and delta from mean

There also appears to be no firm link between the standard deviation about the mean received power at a distance and the magnitude of departure from the trend at that separation distance.

Further, there would appear to be an equal number of positively skewed data points in the graphs of receiver power vs. range (such as Figure 36) as there are negatively skewed data points. Based on this, random error is assumed to be the main influencing factor in the deviance of the results from the trend line indicating correlation between received signal strength and separation distance.

4.3.4.4. Interpreting the results against the thesis

Revisiting the thesis set out in section 1.1.1, it is desirable to define a link between distance and the received signal strength of smart meters and other domestic wireless sensor networks. This would allow for the use of sophisticated location derivation methodologies such as multilateration or weighted proximity detection.

By the use of summed averages to combine the measurements obtained at separation distances less than 7 m into 0.5 m separation distance buckets, it was possible to simplify the data collected such that a representation similar to Ruiz et al.'s could be used - Figure 42. This plot shows a series of histograms representing

the received signal strengths obtained for each 0.5 m separation distance between the network nodes. This is a representation of the probability of receiving any particular received signal strength when at any particular range.

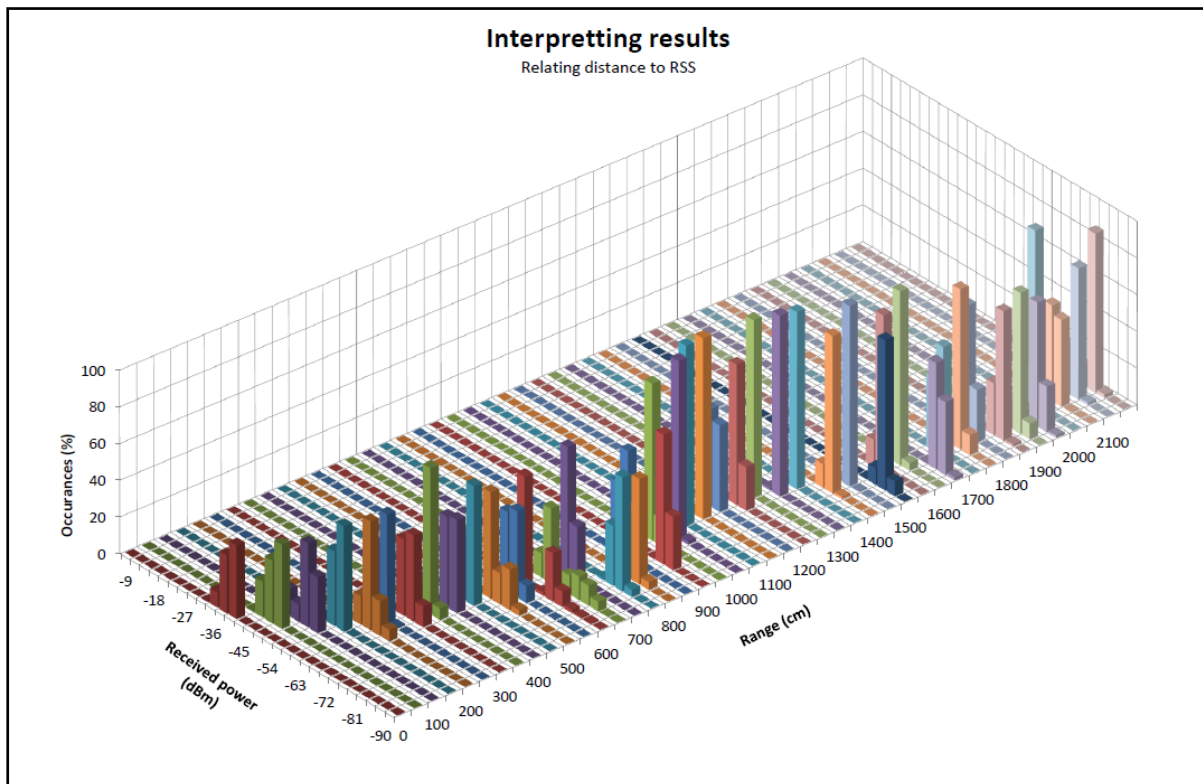


Figure 42: Histograms of received power at different ranges

It is fairly evident from Figure 42 that given any particular range (as a multiple of 0.5 m) it should be possible to predict the expected received signal strength that will be measured. This is possible due to the tight and broadly normally distributed histograms at each range interval. This strong correlation is promising, but to be useful for location purposes, it is necessary to invert the question such that one can predict the separation distance based upon the received signal strength measured.

Figure 43 shows the probability of being at any particular range based upon the measured received signal strength – this is the same data as previously, just represented from a different perspective.

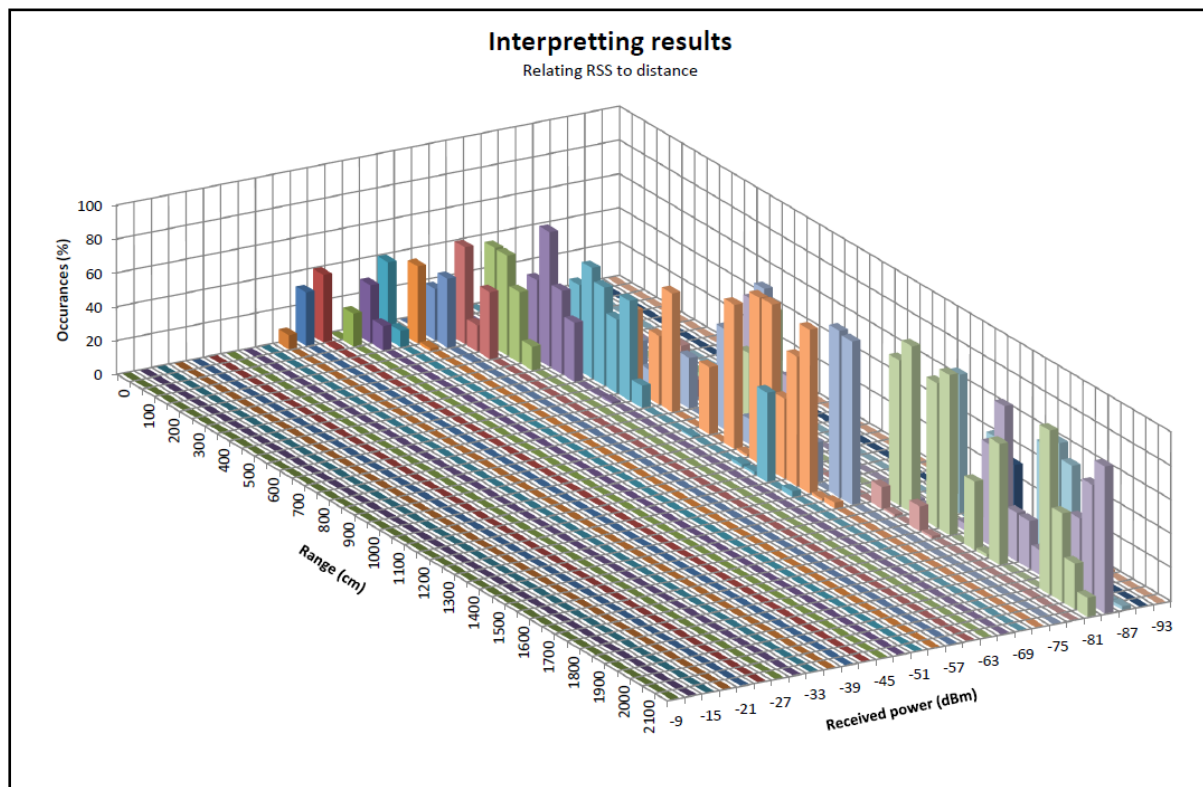


Figure 43: Histograms of range information separated by received power

As Figure 43 shows it is much more difficult to predict the range at which a received power measurement was taken. The histograms are much less normally distributed and span a wide array of possible ranges for any given received power. This is somewhat surprising and was not initially expected based upon the evidence of Figure 42 where the data was considered from the perspective of returned signal strengths at particular distances.

4.3.5. Conclusions

This experiment showed that there is a definite logarithmic correlation between received signal strength and distance. This is closely matched with the expected inverse square law model predicted and matched the assertions of other researchers in the field of radio frequency ranging. An equation has been presented which defines the correlation when using two RZUSB Sticks in as close as possible to free space (Equation 8).

As discussed during the introduction to this testing (section 4.3.1), Benkic et al. [2] and Heurtefeux and Valois [3] both claimed that a correlation between received signal strength and separation distance for IEEE 802.15.4 networks could not be shown. Although now satisfactorily disproved through his first-hand investigations, the author would caveat that their assertions still maintain some merit when considering a similar experimental methodology. It does appear from the author's own testing and the two research papers contended, that it is indeed difficult to derive a highly probable estimation of range based upon a measured received signal strength – this was highlighted in Figure 43.

From a reflection of the data obtained and the methodology undertaken the author proposes that the methodology used by himself and most others in the field was both a suitable and achievable approach for determining a correlation between the two parameters. However, the author proposes that this methodology was unsuitable for predicting range based upon the received signal strengths measured. For this purpose the data was collected from the wrong perspective.

To achieve such a prediction, the author concludes that a methodology similar to the following should be used:

- Using a statistically strong number of samples (1000 measurements per set is still deemed appropriate) measure the range at which each of a series of received power levels is returned. This is a transposed version of the approach taken, whereby instead of measuring the power level at particular distances, the distance is measured at particular power levels.
- Although an RZUSB Stick may still be appropriate for this testing, particularly on the basis of the existing scripts and experience, it may be worth investigating if there were any hardware that could report received signal strength in a finer than 3 dB resolution step size.
- Additionally, in order to reduce the potential for fluctuations in environment, climate or wave propagation, it would be desirable to experiment with utilising the link quality indication (LQI) figure detailed in the AT86RF230 datasheet [106]. This should provide a means for rejecting measurements involving a high likelihood of external interference or multipath signals, thus improving the reliability of the data.

Based upon using an RZUSB Stick this would be perhaps 1000 measurements at 29 possible received power levels so 29,000 total data points – more if a finer resolution of power level could be obtained. The author acknowledges that despite having a third as many measurements as the experiment undertaken for this study, the newly proposed methodology will involve significantly more human participation and thus time in order to undertake a sufficiently robust test approach.

It could alternatively be worth retesting the identified measurement sets in Table 9 that have been noted as potential outliers with a probable cause of random, non-systematic error. This would involve a further 31,000 measurements, but at a lower level of human involvement to the transposed methodology. Re-measured average signal strengths at these distances may present a better fit to the trend lines identified in Figure 36 and Figure 37 as well as a tighter returned spread of possible received power levels at those distances. This would be advantageous and would present a tighter correlation between received signal strength and range; however as Figure 43 demonstrates, even the “strong fit” measurement sets below 0.6 m separation represent a reasonably distributed spread of possible ranges. In any case, the relationship derived from the measured data (including the potential

random errors) was a good fit with predicted models and the work of previous researchers in other radio frequency bands and technologies.

For the reasons above, the author advocates that any future work should focus upon the transposed methodology in favour of increasing the robustness of the results already evaluated. A correlation has now been successfully proven demonstrating significant progress upon the research undertaken to date of using IEEE 802.15.4 networks for location; it is an important next step to prove that more than just a correlation, and contrary to existing research, it is possible to derive range from received signal strength.

4.4. Findings relating to IEEE 802.15.4 range determination

4.4.1. Achievements and impact

In this chapter several key achievements have been demonstrated:

- An omni-directional radiated emissions model of an RZUSB Stick has been created.
- From this a predicted relationship between received signal strength and separation distance was calculated using the Friis Transmission Equation.
- This relationship was shown to be an exact match to the inverse square law, verifying the applicability of the approach.
- The first hand testing and measurements undertaken took on some of the best practices demonstrated by others and achieved (by the aid of previous chapters) some of the finest resolution testing evidenced in correlating received signal strengths to separation distance - regardless of the technology used. Importantly, this testing was performed using two unassociated IEEE 802.15.4 nodes and beacon request frames so closely representing in all but antennae gains the interaction of a RZUSB Stick with a third party network or smart meter.
- The testing confirmed a similarity between the measured results and the previously predicted Friis Transmission Equation / Square Law model. It also showed a near exact fit to Kotanen et al.'s model and the parameters representing free space, gains, losses and random disturbance have been identified for the RZUSB Stick in these tests. The experiment successfully concluded the existence of a firm relationship between received signal strength and separation distance; an equation for this has been presented (Equation 8).
- A systematic error has been identified, concluding that the experimental approach taken by the author and prior researchers is not the most appropriate for subsequently predicting the separation distance of future measurements based upon the received signal strength returned.

On the basis of these achievements the author concludes that range based location finding methodologies (such as those presented in Chapter 2) would be appropriate for using with IEEE 802.15.4 wireless sensor networks.

There is a lack of published research where any of these achievements or conclusions has been evidenced in relation to smart meters or third party wireless sensor networks. In this respect the achievements set out in this chapter represent an original and significant contribution to knowledge.

4.4.2. Next steps

Aside from improving the reliability of the data as discussed in the test conclusions, the next logical progression would be to measure the accuracy to which this same hardware and findings can be used to return a physical location.

The conclusions and next steps of Chapter 3 discussed the merits of orchestrating a controlled grid of simulated smart meters, or alternatively arranging for access to a newly built and unpopulated housing development. If such a facility were available it would be highly desirable to use an RZUSB Stick and the model presented in this chapter (Equation 8) to derive a multilateration style location coordinate and compare this to known reference position values as derived from another source (e.g. GPS). One could go further as to test non-static location accuracy and the utilisation of particle filtering (e.g. Kalman or Bayesian) and map matching techniques with the use of multilateration of smart meter beacon requests. This would be a significant step towards proving a fully functioning solution.

Multilateration of radio frequency networks based upon received signal strength derived ranges has been widely covered in the literature, especially for WiFi networks operating at 2.4 GHz. Due to this, it is arguable that now a mechanism for obtaining a figure for range has been achieved with smart meters, it can be assumed the multilateration aspect is translatable to smart meters. There are still differences remaining which would affect the overall effectiveness however such as the strength and frequency of signals from a smart meter in a house to a hand held location device operated by a pedestrian on the path alongside.

It could be concluded on balance that to pursue a route to market of a new location technology the most appropriate next step may be to physically and experimentally investigate the likely returned accuracies of a smart meter based multilateration approach. Again, as mentioned in section 3.5.2, the roll out of smart meters is not yet due to be completed for a number of years so the useable horizon of any such technology is likely to be some time away.

In contrast, for the purposes of academic study and to more fully explore the possibilities for answering the thesis set out in section 1.1.1, the next chapter shall instead consider the applicability of a third methodology – scene analysis.

Chapter 5 – Preparing for Scene Analysis Using Smart Meters

Chapter Summary

In this chapter some initial work is undertaken to investigate the merits and strengths of fingerprinting methodologies when applied to low-rate wireless personal area networks.

This approach hasn't been fully explored in a practical data collection sense due to the sponsor's policy restrictions and the potential for collateral intrusion of personal information. However despite this, the suitability of the hardware and the requirements of the methodology have been tested.

5.1. Chapter introduction

The overall tangible success of a functioning geolocation system was impossible to directly test whilst also mitigating the risk of accidental data capture and maintaining an affordable research study both in terms of time and financial cost.

Instead, this chapter attempts to probe the viability of using the same hardware identified in Chapter 3 for the purposes of undertaking a fingerprinting methodology. This will explore the suitability of a third approach rather than applying a proximity technique (discussed in Chapter 3) or using multilateration (discussed in Chapter 4). It is notable that no other papers were identified that have considered this approach for IEEE 802.15.4 networks.

As explained in section 1.4, wardriving and other collection of data that could be linked to individuals, businesses or premises was not undertaken during any part of this study.

The chapter concludes with recommendations upon the use of scene analysis of wireless sensor networks for the purposes of location in urban and semi-urban environments.

5.2. Database creation

5.2.1. Purpose

Regardless of the location algorithms used, knowledge of the whereabouts of third party network nodes would be necessary for the purpose of implementing geolocation via IEEE 802.15.4 networks – much the same as it is necessary for geolocation via WiFi.

For proximity and multilateration techniques, it is conceivable that an agglomerate database obtained from energy supplier regarding the locations and network identifiers of their smart meters may in future be a possibility. For a scene analysis methodology however surveying the intended geographic zone prior to using live measurements for location derivation is a necessity. As described in section 2.3.1.4, scene analysis relies upon the existence of a database of historic network signal strength measurements at and around the location of the equipment being located.

By the nature of IEEE 802.15.4 networks, transmissions are infrequent, sporadic and unpredictable. It was shown in Chapter 3 however that it is possible to solicit a response to a beacon request upon demand. Regardless, transmissions are of very short duration and being intended for low power devices are unlikely to cover a significant range. Geographical coverage by these networks is also very low at present. In order to effectively create a database or lookup table of geolocational network information it is necessary to detect as many transmissions as possible of those that are within the proximity of the sensor during an area survey.

This chapter will show and develop a means of generating a database such as is required for scene analysis methodologies to be used.

5.2.2. Requirements

To be successful, these tests need to identify practical drawbacks or methodological enhancements for IEEE 802.15.4 geolocational database creation and pre-surveying.

Simultaneous and sequential channel sampling data collection techniques both required comparison to determine their respective effectiveness. Enough data needed to be collected from a transmission such that a uniquely identifiable network could be theoretically tagged with a corresponding GPS location. As such, it was necessary to use networks under the ownership of the author for this testing to enable live data to be sampled. It was also necessary to employ some of the network filtering techniques developed in previous tests (section 3.4) to ensure that unintentional capture of networks outside the target profile was not possible.

A suitable means of correlating logged data with GPS location information needed to be identified.

And finally, knowledge regarding dominant channel usages was required with a view to advising on the potential benefits of scanning all channels vs. a subset of dominant channels. It is necessary that this data is anonymised with no potential for privacy intrusion by collecting only channel number / spot frequency information whilst mobile but without knowledge of location or time.

5.2.3. Methodology

This testing began with the attaining and reprogramming of twenty RZUSB Sticks – see Appendix 5 for details of the reprogramming. One evidenced manufacturing faults and so was replaced, however as the replacement came from a separate production batch this was never used.



Figure 44: A large batch of RZUSB Sticks were purchased and reprogrammed with customised firmware

To be able to best survey the majority of local transmissions, the system must listen on all the available channels simultaneously using multiple receivers. This approach requires sixteen 2.4 GHz transceivers to cover the channel allocations - something likely to be prohibitive to any future practical implementation of such a system due to physical size, power drain and cost. However, the approach should be tried to determine the viability of the hypothesis despite the assumed drawbacks.

An alternative approach is to cycle through all the available channels at a rate fast enough to detect events on each channel before the survey system's location has changed. This methodology was used effectively in the author's previous work with WiFi (IEEE 802.11) networks [25], [36]. However with the infrequent and short transmissions of the IEEE 802.15.4 standard it is expected that this approach will not be successful at detecting all local network traffic unless substantial dwell periods are allocated per channel in each detection location. It is worth a short investigation to determine the merits of this technique.

Three surveying methodologies were tested with accompanying scripts to enable this and are listed below:

1. A single RZUSB Stick was used with the TPStumbler script (see Appendix 7, section 12.2.1) to automatically cycle through the channels whilst an active network moved through the detection area at a constant walking speed.

The RZUSB Stick scanned all channels at dwell times of 0.1 Seconds, 0.2 seconds, 0.5 seconds, 1 second and 2 seconds. Both active and passive modes were trialled, where "active mode" sent out a beacon request on the current channel at the start of each dwell period, and "passive mode" purely listened for identifiable information (see section 3.4) transmitted between the active network nodes without stimulation from the RZUSB Stick.

2. A single RZUSB Stick per channel (totalling sixteen detectors connected to the laptop via two eight-way USB hubs) was used with the TPStumbler script (see Appendix 7, section 12.2.1) whilst an active network moved through the detection area at a constant walking speed. A bash script (see Appendix 7, section 12.3) was used to automate the TPStumbler scripts as background processes and to allocate channels to each RZUSB Stick.

The RZUSB Sticks scanned their respective channels continuously but in active mode a beacon request was transmitted at dwell times of 0.1 Seconds, 0.2 seconds, 0.5 seconds, 1 second and 2 seconds. Both active and passive modes were trialled, where "active mode" sent out a beacon request on the current channel at the start of each dwell period, and "passive mode" purely listened for identifiable information (see section 3.4) transmitted between the active network nodes without stimulation from the RZUSB Stick.

3. Four RZUSB Sticks were connected to a four-way USB hub and configured as detectors, four more were connected to another four-way USB hub and

configured as transmitters both sets using the TPStumbler script (see Appendix 7, section 12.2.1) whilst an active network moved through the detection area at a constant walking speed. A bash script (see Appendix 7, section 12.3) was used to automate the TPStumbler scripts as background processes and to allocate channels to each RZUSB Stick. Only four channels were tested due to the difficulties faced (and discussed in the results) when attempting the second methodology.

The RZUSB Sticks scanned their respective channels continuously but the transmitting nodes sent a beacon request at intervals of 0.1 Seconds, 0.2 seconds, 0.5 seconds, 1 second and 2 seconds. Only active mode was attempted, sending out a beacon request on the current channel at the intervals defined.

Throughout all three methodologies it was decided to keep the RZUSB Stick static and move the transmitting network (consisting of two XBee nodes and an MBed development board simulating a smart meter network s at a 1 Hz transmission frequency – see section 3.4) through the detection zone. The testing was performed in the centre of a large playing field with the RZUSB Stick mounted at a height of 1 m whilst the simulation network was carried at approximately 1 m height. This mitigated many of the risks of collateral intrusion or disruption of external parties' networks and additionally by distancing the testing from all other sources of 2.4 GHz interference and reflections a more stable transmission range was used.

A proposed efficiency gain to the channel sampling technique was previously proposed by the author during his WiFi geolocation investigations [25]. A study was made of dominant WiFi channel usage by wireless routers and the survey system then cycled only the most dominant channels (1, 6, 11 and 13). Although some networks were inevitably missed, the overall effectiveness of the system actually increased as faster transitions and greater dwell times were possible. To identify the possibility of using a technique such as this a sample survey was made of channel dominance in the local area.

Strict measures were required during the channel dominance surveying to protect against any intrusion of privacy; active scanning only was undertaken with all messages not in response to a solicited beacon request being discarded. The only information recorded in this testing was the fact that a beacon response had been made, and which channel this was made upon. To anonymise the test results further, nothing was displayed upon the console window during testing and no timing or location information was recorded such that channel numbers and localities could not be associated. Additionally to compound this and provide anonymity via obfuscation, a sufficiently wide area (15 mile radius) was traversed such that singular results could not be identified to any particular areas.

To undertake the test ten survey runs were completed with eight RZUSB Sticks sequentially and actively scanning through the channels on a 0.5 second dwell time. To ensure maximum coverage, the script for each RZUSB Stick was started one

scan duration later than the previous channel so that once they had all been started there would be eight different channels scanned per interval and all sixteen channels would be scanned per second.

5.2.4. Results

Testing the three outlined surveying methodologies succeeded in identifying faults and difficulties with each process but did not provide any significantly measurable results.

Table 10 shows the number of messages received for each surveying format; in general there was a high number of missed transmissions not detected by the RZUSB Stick; those records demarked by an asterisk evidenced software malfunctions which are described later.

Methodology	Dwell Time (Seconds)	Scans Per Channel (32 Second Total Duration)	Non-Beacon Messages Transmitted (Channel 14)	Non-Beacon Messages Transmitted (Channel 20)	Total Messages Received	Missed Transmissions (%)
Sequential x1 Rx (Passive)	0.1	20	32	32	7	89.1
	0.2	10	32	32	3	95.3
	0.5	4	32	32	6	90.6
	1	2	32	32	3	95.3
	2	1	32	32	0	100.0
Sequential x1 Rx / Tx (Active)	0.1	20	32	32	11	86.9
	0.2	10	32	32	5	93.2
	0.5	4	32	32	9	86.8
	1	2	32	32	4	93.9
	2	1	32	32	1	98.5
Simultaneous x16 Rx (Passive)	-	-	32	32	11*	82.8
Simultaneous x16 Rx / Tx (Active)	0.1	-	32	32	0*	100.0
	0.2	-	32	32	9*	87.8
	0.5	-	32	32	18*	73.5
	1	-	32	32	24	63.6
	2	-	32	32	22	66.2
Simultaneous x4 Rx + x4 Tx (Active)	0.1	-	32	32	63	25.0
	0.2	-	32	32	57	53.0
	0.5	-	32	32	61	10.3
	1	-	32	32	46	30.3
	2	-	32	32	53	18.5

Table 10: Results of the surveying methodologies

The first method, sequentially scanning all channels at different dwell times, did manage to solicit and capture some transmissions by the passing network. A large number of transmissions were lost and there was very minimal measurable difference between the dwell periods as can be seen in Table 10. The two second

dwelling period showed no results, presumably because the network had already exited the receiving range before scanning on that channel could commence – this highlighted the trouble of requiring the surveying device to be on the right channel, at the right time and in the right place.

The testing of the second method did not perform at all well. It transpired that sixteen RZUSB Sticks cannot be supported by two eight-way USB hubs on a laptop offering just one USB bus. A large number of USB conflicts and unexpected script terminations occurred meaning that the results obtained are not reliable or even complete. The results demarked by an asterisk evidenced fatal script terminations however even the two results which did not force premature closure of the scripts do not, in the author’s opinion, represent reliable data. The only true result from these measurements was to show that the hardware was not capable of the task.

On the basis of the failure of the sixteen simultaneous test nodes, the final methodology was trialled with just four channels spanning the two target channels and additionally channels 16 and 18. This setup did not outwardly appear to evidence any software issues or USB conflict errors. There was a marked improvement in capture rate versus a sequential rate, but that is to be expected as each channel of interest was being monitored all of the time as well as sending beacon requests at regular intervals.

Test Run Number	Transmitted Beacon Requests	Channel 11	Channel 12	Channel 13	Channel 14	Channel 15	Channel 16	Channel 17	Channel 18
1	854	1	1	1					
2	538		3	2					
3	985					5			
4	875					10			
5	293			1					
6	798		1						
7	20								
8	?	4		9	18	3	2	1	
9	578	1							
10	?	9					3		
Total	4941 + ?	15	5	13	18	18	5	1	0

Test Run Number	Transmitted Beacon Requests	Channel 19	Channel 20	Channel 21	Channel 22	Channel 23	Channel 24	Channel 25	Channel 26
1	854						1		
2	538								
3	985	1							
4	875	2							
5	293		4						
6	798								
7	20		4						
8	?								
9	578				1				
10	?								
Total	4941 + ?	3	8	0	1	0	1	0	0

Table 11: Channel dominance surveying results

Test Run Number	Transmitted Beacon Requests	Total Received Responses
1	854	4
2	538	5
3	985	6
4	875	12
5	293	5
6	798	1
7	20	4
8	?	37
9	578	2
10	?	12
Total	4941 + ?	88

Table 12: Summary channel dominance survey results

From Table 11 it is possible to see that pre-surveying the target geographical zone to determine channel dominance may be a useful technique. In this instance channels 11, 13 and 15 consistently evidenced the most consistently high number of responses out of the ten test runs of the locality surveyed. It is acknowledged that not a particularly large sample of responses was possible due to the only partially completed smart meter roll out.

5.2.5. Conclusions

In general, and excluding the software malfunctions, active scanning appeared to perform a little better than passive, but with seemingly little margin. On the merit of this experiment, the type of scanning performed on a live system is more likely to be determined by policy than performance advantages.

The use of multiple simultaneous channels to detect transmissions and measure their received signal strengths did prove to be a successful methodology. The hardware implementation chosen however was not capable of performing this task adequately or reliably. It is the author's opinion that to successfully monitor all sixteen channels simultaneously it would be necessary to design a bespoke hardware solution. A field programmable gate array with software defined radio implementations may present the best parallel approach to capturing multichannel data. It is beyond the scope of this project to investigate this avenue further.

Cycling through all the channels in sequence did successfully capture identifiable data sufficient for locating oneself from a database of measurements. However, it is unlikely to yield sufficiently comprehensive data for the purposes of pre-surveying and database creation. As can be expected a large amount of transmissions were missed compared to the multichannel system due to the 1/16th dwell time per channel (or worse due to the time taken to switch between channels).

The testing undertaken for channel dominance suggests that the channel cycling methodology could be improved by as much as four or five times by concentrating efforts on only the most dominant three or four channels. Although this is a significant improvement, the multichannel system still outperformed this expected return by considerably more again. Perhaps the most beneficial utilisation of this channel dominance data would be to simplify the multichannel system to a manageable number of simultaneous channels and thus create an effective surveying tool for geotagging IEEE 802.15.4 fingerprints.

The channel dominance test methodology did have several weaknesses and a more robust methodology could have been taken. For instance, it would have been by far preferable to conduct both passive and active scanning simultaneously across multiple channels. It would also be ideal to record network identifiers such that responses from the same network are not counted twice – or at the very least maintain constant speeds without any stationary periods to ensure all networks respond a similar number of times. In a practical sense however these improvements were both difficult to achieve and in some cases would not be within the bounds of the employer's policy and intrusion mitigation requirements. An ideal alternative solution may be to request channel usage statistics from the energy suppliers to the local area (if this data is held).

5.3. Scene analysis findings

5.3.1. Achievements and impact

This testing conclusively showed that the chosen hardware solution was not suitable for simultaneously scanning a large number of channels.

The experiment also showed that for an area of interest, pre-surveying the area for channel dominance is likely to be an effective means of reducing sequential scanning efforts. This is important as it means that it is not necessary to develop a device with multiple simultaneously scanning channels to be able to obtain some multichannel measurements of use to a scene analysis methodology. This means devices can be smaller, lower power and also presumably cheaper – all important traits of handheld consumer location aids.

5.3.2. Next steps

The testing in this chapter only began to investigate the effectiveness of using a scene analysis methodology. As with proximity and multilateration methodology investigations, the next step to continue this research would be to ascertain location accuracy figures through the use of a large area distributed network (either simulated or as discussed previously by accessing a newly constructed housing development). This would facilitate the large scale collection of received signal strengths geocoded to systematic survey of a geographical area.

A location accuracy comparison between a device operating as a sequential scanner across only the most dominant channels, and a device simultaneously scanning all channels would be exceedingly useful for focusing any future commercial developments. It is anticipated that the simultaneous scanner would evidence greater accuracy, but that there would be a cost and size trade-off at which the lesser accuracy of the simultaneous scanner may be preferable.

As discussed, a laptop proved not to be the most appropriate platform for this experiment. To meaningfully progress investigations into scene analysis it would be necessary to either carry out the surveying above on several occasions using a small number (say four) of receiving nodes, or to produce a custom hardware solution capable of operating the full sixteen nodes.

Chapter 6 – Discussions and Conclusions

Chapter Summary

In this chapter the research questions supporting the original thesis are reevaluated and the overall successes of the proposed methodology are considered against the hypothesis.

This final chapter considers the merits of the research undertaken and the strengths of the conclusions reached. It also proposes the next steps for continuing this line of study and potential means for improving the success of using IEEE 802.15.4 for deriving a location.

6.1. Using IEEE 802.15.4 for Location

From the experiments undertaken in this study and the prior research of others in the wider field of radio frequency based location (particularly those at 2.4 GHz such as WiFi and Bluetooth) it has been observed that determining location from unrelated third party IEEE 802.15.4 wireless personal area networks is a definite possibility. This has been a significant academic achievement as the only researchers to previously attempt similar studies (although they did not tackle the use of third party networks let alone smart meters) have reported that the resolution of measurement possible was not sufficient for measuring the separation distance between transceivers with an aim to deriving location.

It has further been evidenced in this research that, with appropriate hardware, it is not necessary to be connected to or associated with the networks being measured for the purposes of deriving location. This means that it is possible to identify one's location from energy smart meters and other domestic wireless sensor networks as per the opening thesis.

From the evidence gathered, it is apparent that transmissions from smart meters may be significantly better for location purposes than would otherwise be suggested by the standards. Specifically, as opposed to a once per ten second transmission frequency based upon the standard (striving towards once per five seconds) it would appear that actual smart meters exhibit a once per second transmit frequency. This will ensure a vastly greater chance of detection whilst moving or significantly more detections to base a reliable position upon whilst static.

The testing undertaken was built upon the knowledge gained from a thorough literature review of the field. For instance, many of the principles employed by others, and previously by the author, in WiFi (IEEE 802.11) location systems investigation could be directly ported to this application. Despite significant differences in networking topologies and communication formats and rates, the physics principles behind the radio transmissions at 2.4 GHz of both protocols remain identical.

6.1.1. Suitable methodologies for geolocation

This study has shown that there are several plausible and more importantly achievable means of deriving location from low-rate wireless personal area networks such as smart meters. Within the experiments undertaken for this research it has been evidenced that proximity and scene analysis methodologies are both achievable and the manner with which this can be undertaken is explored.

More significantly, this research proved that a direct correlation exists between received signal strengths of IEEE 802.15.4 networks and their range. It has been shown in prior literature how range can be used to implement multilateration to several transmitters of known position and thus deduce the range of the detecting node to a high accuracy.

Although not experimentally discussed, this study also showed through an absence of directional and well documented IEEE 802.15.4 radio modules that triangulation and direction finding style methodologies are not currently suitable unless bespoke hardware is designed. The same can be said of time based range derivation techniques (e.g. time of arrival or time difference of arrival); however these are not as significant given a means for range based location has been identified.

Combining the relationship arrived at in Chapter 4 between received signal strength and separation distance between two RZUSB Sticks, with the transmission characteristics of a smart meter (discussed in section 2.2) results in the relationship shown in

Equation 9 for transmissions from a smart meter to an RZUSB Stick:

$$[Distance] = 10^{\left(\frac{[Rx Power] - 27.5 + 20 \log(\lambda) + 20 \log(4\pi)}{10n}\right)}$$

Equation 9: Relationship between received signal strength and distance for RZUSB Stick measuring smart meter transmissions

Where the environmental coefficient n is typically expected to be in the region of 3.0 to 5.0 to represent an urban or semi-urban environment.

Achieving to determine a corroborated relationship of received signal strength and range means that range based location methodologies (primarily multilateration) are possible using smart meters and appropriate detection hardware.

With this relationship it is possible to revisit the anticipated ranges (Table 13) and resulting possible speeds of travel at which a proximity type location derivation would be plausible (Table 14).

	RZUSB Stick to RZUSB Stick	Smart meter to RZUSB Stick
TX power (dBm)	3	-3
RX sensitivity (dBm)	-101	-101
TX gain (dB)	0	-
RX gain (dB)	0	-
Fade margin (dB)	-10.67	-10.67
Frequency (MHz)	2.48	2.48
Wavelength (m)	0.121	0.121
Range (m)	45.2	30.5
Range Fiss Equation (m)	446.3	223.7

Table 13: Revised transmission range to a smart meter

Table 13 shows that a much greater detection range than anticipated is achievable between a smart meter and an RZUSB Stick. A coefficient for n of 3.5 has been used here to account for a singular wall and some furniture existing between a smart meter and the detection node.

6.1.1. Practical implications

In comparison to in section 2.2, Table 14 additionally shows update frequencies of one transmission per one or two seconds (1 and 0.5 Hz) based upon the earlier discussed increased rates of transmissions evidenced.

Transmission Frequency (S)	1	2	5	10	1800	Speed (MPH)
Distance Travelled (m)	0.4	0.9	2.2	4.5	804.7	1
	2.2	4.5	11.2	22.4	4023.4	5
	4.5	8.9	22.4	44.7	8046.7	10
	6.7	13.4	33.5	67.1	12070.1	15
	8.9	17.9	44.7	89.4	16093.4	20
	11.2	22.4	55.9	111.8	20116.8	25
	13.4	26.8	67.1	134.1	24140.2	30
	15.6	31.3	78.2	156.5	28163.5	35
	17.9	35.8	89.4	178.8	32186.9	40
	20.1	40.2	100.6	201.2	36210.2	45
	22.4	44.7	111.8	223.5	40233.6	50
	24.6	49.2	122.9	245.9	44257.0	55
	26.8	53.6	134.1	268.2	48280.3	60
	29.1	58.1	145.3	290.6	52303.7	65
	31.3	62.6	156.5	312.9	56327.0	70
	33.5	67.1	167.6	335.3	60350.4	75
	35.8	71.5	178.8	357.6	64373.8	80
	38.0	76.0	190.0	380.0	68397.1	85
	40.2	80.5	201.2	402.3	72420.5	90
	42.5	84.9	212.3	424.7	76443.8	95
	44.7	89.4	223.5	447.0	80467.2	100
	46.9	93.9	234.7	469.4	84490.6	105
	49.2	98.3	245.9	491.7	88513.9	110
	51.4	102.8	257.0	514.1	92537.3	115
	53.6	107.3	268.2	536.4	96560.6	120
	55.9	111.8	279.4	558.8	100584.0	125
	58.1	116.2	290.6	581.2	104607.4	130
	60.4	120.7	301.8	603.5	108630.7	135
Key:						
TX range < 12.5 m radius	62.6	125.2	312.9	625.9	112654.1	140
TX range < 30.5 m radius	64.8	129.6	324.1	648.2	116677.4	145
Distance travelled > than TX range	67.1	134.1	335.3	670.6	120700.8	150

Table 14: Revised estimation of ability to attain location fix based upon speed

The green cells in Table 14 indicate those speeds at which the smart meter

transmissions would definitely occur whilst within the originally anticipated transmission range of a smart meter (based on the calculations in section 2.2).

The orange cells indicate the speeds at which based on the testing and evidence obtained in this study, a smart meter transmission could be detected and utilised for proximity or scene analysis based location derivation.

The red cells are those speeds at which a transmission from a smart meter may be detected if it were to occur at the point when passing, but it would not be guaranteed that the detection device would be in range for the entire transmission window.

Comparing Table 2 and Table 14, this study has shown that the practical effectiveness of obtaining measurable data whilst in motion is better in reality than might be supposed from an analysis of the standards – from both perspectives of frequency of transmission and detectable range.

As was identified early on (section 1.6) the use of smart meters and low-rate wireless personal area networks for location derivation has many benefits commercially such as size, power consumption, firmware complexity etc. These advantages mean that the significance of the research findings presented is greater than just the notion of academic novelty. King et al. make a strong case for the agglomeration of multiple sensors when determining location [51]. Commercially, being able to utilise the same antennae hardware within mobile technology aids as is already utilised for WiFi and Bluetooth location methods would harbour significant advantages and increase the resolution, accuracy and reliability of the technical solution in urban and semi urban environments.

6.2. Research Limitations

6.2.1. Suitability of methodology

The literature review and progressive suite of tests undertaken did successfully answer and explore the research questions set and ultimately upheld the thesis.

Ideally, each of the three localisation methodologies explored would be pursued to a metric of their locational accuracies – i.e. a definition of uncertainty in +/- metres for a static or moving measurement device. To be comparable to GPS technologies this should be within approximately +/- 2 m and < +/- 1m for WiFi if within a densely populated urban environment. The key aspect would be the ability to compare the accuracy of the state of the art in multilateration or scene analysis for WiFi and to compare the accuracies and uncertainties when performed with smart meters and domestic wireless sensor networks. It is currently only feasible to simulate the above comparison given that the smart meter programme has not been completed. Finding and procuring access to a site large enough but still isolated from collateral wireless intrusion and unintended disruption was not a possibility within the timescales and resources of this project, nor would it be easy to justify against the achievements made without such provisions.

Having considered the possibilities and reflected upon the findings of the experiments the author firmly believes that the methodology taken was appropriate and suitable even if there were still scope for further improvement. The technology readiness of a smart meter dependent location system is not yet established sufficiently to justify the suggested measurement uncertainty of location comparisons.

Although not considered to be of significance to alter the overall findings of this study, there are some areas where the methodology adopted could be enhanced upon for any future studies:

- In generating the directionality model of the RZUSB Stick it would be ideal to approach this from a closer perspective to radiated emissions testing for new products. Utilising anechoic chambers, turn tables, spectrum analysers and high sensitivity selective receivers a much finer resolution, 720 ° polar response plot could be formed allowing far greater insight into the directionality characteristics of the transmitting antenna.
- When performing range testing it would be ideal to have undertaken measurements for a range of environments to be able to formulate a further relationship to account for the type and construction of buildings.

- It would additionally have been ideal to have taken the range measurements with respect to a sample of actual smart meters rather than a second RZUSB Stick given the antennae and transmission powers will differ significantly.
- To obtain a tighter distribution of received signal strengths at particular ranges the range testing should have been performed differently as explained in the conclusions to the experiment (section 4.3.5). Measuring the ranges at which particular received signal strengths were obtained would have provided better distributed test results for the questions being asked. That said, it did not have an impact upon the ability to formulate a mathematical correlation validating the assertion that the range and thus location can be determined.

6.2.2. Suitability of equipment

For the approach taken the RZUSB Stick was seemingly more than capable of the task and achieved greater successes than the radio modules used previously by other researchers. It would be worth investigating the new Californian Eastern Laboratories' EM357 USB Stick as discussed in section 3.2.5 as this may prove capable of finer resolution measurements of received signal strength than an RZUSB Stick. However unless the firmware stack were similarly reprogrammable then there is unlikely to be another commercial-off-the-shelf module capable of interacting with third party networks so beacon requests would be the only measurement mechanism.

If a directional methodology (e.g. triangulation) were to be attempted in the future then the RZUSB Stick as stands would not be suitable. That said, there does not appear to exist any directional radio modules (let alone those which may prove to be reprogrammable) so for such a purpose a custom solution would need to be designed – this was beyond the scope of this research study.

The equipment set up chosen proved incapable of simultaneous multichannel capture due to the fact that only a single USB bus existed on the laptop motherboard. The only conceivable means to accomplish simultaneous capture on up to sixteen channels would be to operate several computing platforms. The alternative of creating a bespoke hardware solution / software define radio (discussed in the test conclusions) would be preferable, but at too great a resource requirement for the study undertaken.

6.2.3. Validity of data

Several issues were identified and discussed during the scene analysis applicability experiments in Chapter 5. Primarily hardware inadequacies prevented the collection of wholly reliable or accurate data. An additional consideration if this were to be repeated would be to automate the movement of the transmitting networks through the detection zone such that a constant acceleration and measured velocity were used.

Seemingly there was an opportunity for random error in the measurement of received signal strength when looking for a relationship to range – these were discussed in the results and conclusions of Chapter 4. Although extensively analysed it would appear that this was not particularly a result of the methodology or equipment, and retesting the dubious data points may prove useful (a worst case of 31 measurement sets in total). The net effect upon the trend and therefore derived model appeared to be negligible and the final results tied closely with the work of other researchers.

No issues or questionable results can be mentioned with regards to the proximity methodology testing (Chapter 3).

6.2.4. Value of research versus peers

There are no other identified studies which investigate the possibility of locating oneself based upon smart meters or any other third party IEEE 802.15.4 networks. In this sense alone there is much value in having shown that such location derivation is plausible. To the best of the author's knowledge this research constitutes an original contribution to knowledge and credits this work as the first to conceptualise the use of smart meters for aiding location.

Given the plethora of research into WiFi location techniques that have been published now that WiFi routers have been present in the majority of domestic and commercial properties for some years, it is anticipated that once the smart meter roll out programme is complete there will be a similar surge in academic interest. In this sense, the author has quite effectively initiated some of the first of a potentially large field of research into this emerging scene; identifying several potential paths for successive studies to come.

As discussed in detail in Chapter 4, there are only two studies which relate to deriving range from the received signal strengths of low-rate wireless personal area networks for the purpose of multilateration. To argue and successfully conclude that these two studies are erroneous in their assertions that a clear relationship cannot be derived is a notable achievement – particularly when the findings are in accord with, and thus corroborated by, the remainder of the field of radio frequency based localisation research.

Compared to studies in related fields (WiFi, Bluetooth and RFID), the investigations undertaken and presented of the relationship between received signal strength and range were the largest and most tightly controlled evidenced in literature. The firmware alteration, script automation and the author's own experience of industry product certification testing helped to create one of the most robust experiments of this type ever undertaken – even despite the identified data questionability above six metres separation distance.

6.3. Further Avenues for Research

6.3.1. Including link quality indicator

Regardless of whether multilateration or scene analysis methodologies are employed, Chapter 4 showed that there may be merit in utilising the link quality indication values provided by IEEE 802.15.4 hardware. In any measurement of received signal strength the link quality indication ought to provide a level of confidence that the measured data received has not been compromised by multipath effects. This should improve the reliability of the data, presumably making a marked performance increase when considering positional accuracies.

6.3.2. Alternative logging equipment

On the basis of the investigations undertaken in Chapter 3, and the hardware used by others in their research, it appears that there does not exist any radio module capable of directional measurement of received signal strengths to a fine resolution. This is especially true if considering the case of measuring non-beacon messages from third party networks.

Similarly, it would be beneficial to pursue a development of some bespoke hardware specifically for an academic purpose. It is the author's opinion that an open source design would be beneficial to allow for continued evolution of the design as technology and knowledge develops.

If such a development were undertaken it would be useful to additionally consider the ability to operate numerous simultaneous channels. Perhaps 32 to account for a separate transmitter and receiver per channel in the 2.4 GHz band; or better still additionally incorporating the 868 and 915 MHz channels for global coverage of the ISM band.

6.3.3. Live data capture

The most significant avenue for continuation (and hence most discussed previously within this dissertation) would be the large scale capture of real, measurable smart meter transmissions. With different legal and policy restraints it would be conceivably possible to await the completed roll out of domestic smart meters and then undertake a wardriving style surveys to examine the real world practical accuracies of different approaches and algorithms.

With enough funding and planning, similar results could conceivably be achieved by creating a private collection of operational smart meters and measuring the positional accuracies achieved. With this approach aspects such as the spacing between smart meter systems, models and channels used could all be controlled and varied to investigate the parameter space.

6.3.1. Simulated locational testing

Using the evidence from this study, it ought to be possible to formulate the beginnings of an entirely computer simulated scenario through which altering parameters such as building constructions, physical distances, weather conditions, antenna responses, smart meter densities etc can all be investigated.

This would potentially be the most viable and arguably academically rewarding path for continued study as this would strive towards answers regarding the most effective methodologies and the creation of generic / dynamic locational models accounting for the perceived environment.

References

- [1] HM Government, "GOV.UK - Smart meters," 2014. [Online]. Available: <https://www.gov.uk/smart-meters>. [Accessed: 03-Oct-2015].
- [2] K. Benkic, M. Malajner, P. Planinsic, and Z. Cucej, "Using RSSI value for distance estimation in wireless sensor networks based on ZigBee," *2008 15th International Conference on Systems, Signals and Image Processing*, 2008.
- [3] K. Heurtefeux and F. Valois, "Is RSSI a Good Choice for Localization in Wireless Sensor Network?," in *2012 IEEE 26th International Conference on Advanced Information Networking and Applications*, 2012, pp. 732–739.
- [4] Crossbow Technologies, "TelosB Mote Datasheet," 2005. [Online]. Available: http://www.willow.co.uk/TelosB_Datasheet.pdf. [Accessed: 15-Oct-2015].
- [5] Legislation.gov.uk, *Wireless Telegraphy Act 2006 c.36*. UK: Her Majesty's Stationery Office and Queen's Printer of Acts of Parliament, 2006.
- [6] O. Survey, "A guide to coordinate systems in Great Britain," *Ordnance Survey Southampton httpwww gps gov*, vol. v2.3, p. 43, 2015.
- [7] I. Constandache, R. R. Choudhury, and I. Rhee, "Towards mobile phone localization without war-driving," in *Proceedings - IEEE INFOCOM*, 2010, pp. 1–9.
- [8] (IEEE), "802.15.4g-2012 - IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks." IEEE Computer Society, pp. 1–252, 2012.
- [9] G. Jose, "IEEE Std. 802.15.4 - Enabling Pervasive Wireless Networks," *University of California, Berkeley*, 2005. [Online]. Available: <http://www.cs.berkeley.edu/~prabal/teaching/cs294-11-f05/slides/day21.pdf>. [Accessed: 08-Oct-2015].
- [10] HM Government, "Industry's Draft Specifications," 2011. [Online]. Available: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/39368/2393-smart-metering-industrys-draft-tech.pdf.
- [11] HM Government, "GOV.UK - Smart meters: information for industry and other

- stakeholders,” *Climate change and energy – guidance*, 2013. [Online]. Available: <https://www.gov.uk/smart-meters-information-for-industry-and-other-stakeholders>. [Accessed: 08-Oct-2015].
- [12] K. Al Nuaimi and H. Kamel, “A survey of indoor positioning systems and algorithms,” in *2011 International Conference on Innovations in Information Technology, IIT 2011*, 2011, pp. 185–190.
- [13] J. A. Besada, A. M. Bernardos, P. Tarrío, and J. R. Casar, “Analysis of tracking methods for wireless indoor localization,” in *2007 2nd International Symposium on Wireless Pervasive Computing*, 2007, pp. 492–497.
- [14] P. A. Zandbergen, “Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning,” *Transactions in GIS*, vol. 13, no. Supplement s1, pp. 5–25, 2009.
- [15] Y. Qi, H. Kobayashi, and H. Suda, “Analysis of wireless geolocation in a non-line-of-sight environment,” *Wireless Communications, IEEE Transactions on*, vol. 5, no. 3, pp. 672–681, 2006.
- [16] Y. Gu, A. Lo, and I. Niemegeers, “A survey of indoor positioning systems for wireless personal networks,” *IEEE Communications Surveys and Tutorials*, vol. 11, no. 1, pp. 13–32, 2009.
- [17] R. Mautz and S. Tilch, “Survey of optical indoor positioning systems,” in *2011 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2011*, 2011, pp. 1–7.
- [18] M. Bouet and A. L. Dos Santos, “RFID tags: Positioning principles and localization techniques,” in *2008 1st IFIP Wireless Days, WD 2008*, 2008, pp. 1–5.
- [19] H. Liu, H. Darabi, P. Banerjee, and J. Liu, “Survey of wireless indoor positioning techniques and systems,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, 2007.
- [20] R. Harle, “A survey of indoor inertial positioning systems for pedestrians,” *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1281–1293, 2013.
- [21] C. Wu, Z. Yang, Y. Liu, and W. Xi, “WILL: Wireless indoor localization without site survey,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, pp. 839–848, 2013.
- [22] R. D. Hill, “Theory of Geolocation by Light Levels,” in *Elephant Seals:*

- Population Ecology, Behavior, and Physiology*, Illustrate., B. J. L. Boeuf and R. M. Laws, Eds. Berkeley: University of California Press, 1994, pp. 227–236.
- [23] G. M. Djuknic and R. E. Richton, “Geolocation and assisted GPS,” *Computer*, vol. 34, no. 2, pp. 123–125, 2001.
- [24] HM Government, “Smart Metering Equipment Technical Specifications,” *Smart Metering Implementation Programme*, 2014. [Online]. Available: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/381535/SMIP_E2E_SMETS2.pdf. [Accessed: 15-Oct-2015].
- [25] HM Government Home Office Centre for Applied Science and Technology, *FLAME Test Documentation*. Home Office, 2013.
- [26] G. Mao, B. Fidan, and B. D. O. Anderson, “Wireless sensor network localization techniques,” *Computer Networks*, vol. 51, no. 10. pp. 2529–2553, 2007.
- [27] S. Farahani, “Location Estimation Methods,” in *ZigBee Wireless Networks and Transceivers: the complete guide for RF/wireless engineers*, Burlington: Elsevier Science, 2011, pp. 225–246.
- [28] K. Sjö, D. Gálvez López, C. Paul, P. Jensfelt, and D. Kragic, “Object Search and Localization for an Indoor Mobile Robot,” *CIT. Journal of Computing and Information Technology*, vol. 17, no. 1. pp. 67–80, 2004.
- [29] Z. Farid, R. Nordin, and M. Ismail, “Recent advances in wireless indoor localization techniques and system,” *Journal of Computer Networks and Communications*, vol. 2013. 2013.
- [30] R. D. Hill and M. J. Braun, “Geolocation by light level—The next step: Latitude.,” in *Electronic Tagging and Tracking in Marine Fisheries*, 2001, pp. 315–330.
- [31] Sonitor RTLS Technologies, “Sonitor Sense RTLS.” [Online]. Available: <http://www.sonitor.com/>. [Accessed: 20-May-2007].
- [32] Z. Weissman and Tadlys Wireless Communications Ltd., “TOPAZ - Indoor Location Systems,” 2004. [Online]. Available: www.tadlys.co.il/pages/Product_content.asp?iGlobalId=2. [Accessed: 20-May-2005].
- [33] L. M. Ni and A. P. Patil, “LANDMARC: indoor location sensing using active RFID,” in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).*, 2003, pp. 407–415.

- [34] A. Schwaighofer, M. Grigoraş, V. Tresp, and C. Hoffmann, "GPPS: A Gaussian Process Positioning System for Cellular Networks," *Advances in Neural Information Processing Systems*, pp. 579–586, 2003.
- [35] Y. Zhao and L. M. Ni, "VIRE: Virtual reference elimination for active RFID-based localization," *Ad-Hoc and Sensor Wireless Networks*, vol. 17, no. 1c, pp. 169–191, 2013.
- [36] HM Government Home Office Centre for Applied Science and Technology, *FLAME User Manual*. Home Office, 2013.
- [37] Y. Fukuju, M. Minami, H. Morikawa, and T. Aoyama, "DOLPHIN: an autonomous indoor positioning system in ubiquitous computing environment," *Proceedings IEEE Workshop on Software Technologies for Future Embedded Systems. WSTFES 2003*, 2003.
- [38] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system," *Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom '00*, pp. 32–43, 2000.
- [39] A. Ward, A. Jones, and A. Hopper, "A new location technique for the active office," *Personal Communications*, vol. 4, no. 5, pp. 42–47, 1997.
- [40] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, 1992.
- [41] M. K. Hoang, S. Schmitz, C. Drueke, D. H. T. Vu, J. Schmalenstroeer, and R. Haeb-Umbach, "Server based indoor navigation using RSSI and inertial sensor information," in *2013 10th Workshop on Positioning, Navigation and Communication (WPNC)*, 2013, pp. 1–6.
- [42] P. Robertson, M. Angermann, and B. Krach, "Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors," *Proceedings of the 11th international conference on Ubiquitous computing Ubicomp 09*, p. 93, 2009.
- [43] M. Popa, J. Ansari, J. Riihijarvi, and P. Mahonen, "Combining Cricket System and Inertial Navigation for Indoor Human Tracking," *2008 IEEE Wireless Communications and Networking Conference*, 2008.
- [44] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proceedings of the 10th international conference on Ubiquitous computing*, 2008, pp. 114–123.

- [45] A. R. Jiménez Ruiz, F. Seco Granja, J. C. Prieto Honorato, and J. I. Guevara Rosas, "Accurate pedestrian indoor navigation by tightly coupling foot-mounted IMU and RFID measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 1, pp. 178–189, 2012.
- [46] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-Effort Crowdsourcing for Indoor Localization," in *Proceedings of the 18th annual international conference on Mobile computing and networking - Mobicom '12*, 2012, pp. 293–304.
- [47] C. Lukianto, C. Hönniger, and H. Sternberg, "Pedestrian smartphone-based indoor navigation using ultra portable sensory equipment," in *2010 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2010 - Conference Proceedings*, 2010, pp. 1–5.
- [48] C. Fischer, K. Muthukrishnan, M. Hazas, and H. Gellersen, "Ultrasound-aided pedestrian dead reckoning for indoor navigation," in *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, 2008, pp. 31–36.
- [49] S. Beauregard, "A helmet-mounted pedestrian dead reckoning system," in *Applied Wearable Computing*, 2006, pp. 1–11.
- [50] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, vol. 2, 2000.
- [51] T. King, S. Kopf, T. Haenselmann, C. Lubberger, and W. Effelsberg, "COMPASS: A probabilistic indoor positioning system based on 802.11 and digital compasses," *Online*, no. September, pp. 34–40, 2006.
- [52] J. Hightower, G. Borriello, and R. Want, "SpotON: An indoor 3D location sensing technology based on RF signal strength," *UW CSE 00-02-02, University of*, p. 16, 2000.
- [53] C. Brignone and T. Connors, "SmartLOCUS: An autonomous, self-assembling sensor network for indoor asset and systems management," *Mobile Media Syst. Lab., ...*, 2003.
- [54] R. Battiti, N. T. Le, and A. Villani, "Location-aware computing: a neural network model for determining location in wireless LANs," 2002.
- [55] a. M. Ladd, K. E. Bekris, G. Marceau, A. Rudys, D. S. Wallach, and L. E.

- Kavraki, "Using wireless Ethernet for localization," in *IEEE/RSJ International Conference on Intelligent Robots and System*, 2002, vol. 1, pp. 402–408.
- [56] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," *Proceedings of the 10th annual international conference on Mobile computing and networking - MobiCom '04*, p. 70, 2004.
- [57] Z. Xiang, S. Song, J. Chen, H. Wang, J. Huang, and X. Gao, "A Wireless LAN-based Indoor Positioning Technology," *IBM Journal of Research and Development*, vol. 48, no. 5/6, pp. 617–626, 2004.
- [58] P. Prasithsangaree, P. Krishnamurthy, and P. K. Chrysanthis, "On indoor position location with wireless lans," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, 2002, vol. 2, pp. 720–724.
- [59] Y. Gwon and R. Jain, "Error Characteristics and Calibration-free Techniques for Wireless LAN-based Location Estimation," *Performance Evaluation*, pp. 2–9, 2004.
- [60] X. Huang, R. Janaswamy, and A. Ganz, "Scout: Outdoor Localization Using Active RFID Technology," in *2006 3rd International Conference on Broadband Communications, Networks and Systems*, 2006, pp. 1–10.
- [61] E. Aitenbichler and M. Muhlhauser, "An IR local positioning system for smart items and devices," *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.*, 2003.
- [62] Y. Zhang, M. G. Amin, and S. Kaushik, "Localization and Tracking of Passive RFID Tags Based on Direction Estimation," *International Journal of Antennas and Propagation*, vol. 2007, pp. 1–9, 2007.
- [63] Multispectral Solutions, "Sapphire DART UWB-based Real-Time Location Systems," 2005. [Online]. Available: <https://www.zebra.com/gb/en/products/location-solutions/dart-uwb.html>.
- [64] Ubisense Group plc., "Ubisense RTLS," 2005. [Online]. Available: <http://www.ubisense.net>. [Accessed: 20-May-2005].
- [65] Ekahau Inc., "Ekahau," 2008. [Online]. Available: <http://www.ekahau.com/>. [Accessed: 01-May-2015].
- [66] C. Wang, H. Wu, and N. F. Tzeng, "RFID-based 3-D positioning schemes," *Proceedings - IEEE INFOCOM*, pp. 1235–1243, 2007.

- [67] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services - MobiSys '05*, 2005, pp. 205–218.
- [68] V. Otsason, A. Varshavsky, A. Lamarca, and E. De Lara, "Accurate GSM Indoor Localization," *Pervasive and Mobile Computing*, vol. 3, no. 6, pp. 698–720, 2007.
- [69] S. Tilch and R. Mautz, "Current investigations at the ETH zurich in optical indoor positioning," in *Proceedings of the 2010 7th Workshop on Positioning, Navigation and Communication, WPNC'10*, 2010, pp. 174–178.
- [70] S. Tilch and R. Mautz, "Development of a new laser-based, optical indoor positioning system," *ISPRS Commission*, vol. XXXVIII, pp. 575–580, 2010.
- [71] iRobot®, "NorthStar® Navigation Cube – Channel 2," 2015. [Online]. Available: <http://www.irobot.co.uk/Store/accessories/northstar-navigation-cube-channel-2>. [Accessed: 20-May-2005].
- [72] Y. Netzer and T. Wang, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop on deep learning and unsupervised feature learning*, 2011, vol. 2011, pp. 1–9.
- [73] H. Hile and G. Borriello, "Positioning and orientation in indoor environments using camera phones," *IEEE Computer Graphics and Applications*, vol. 28, no. 4, pp. 32–39, 2008.
- [74] A. Kitanov, S. Bisevac, and I. Petrovic, "Mobile robot self-localization in complex indoor environments using monocular vision and 3D model," in *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, 2007, pp. 1–6.
- [75] J. Ido, Y. Shimizu, Y. Matsumoto, and T. Ogasawara, "Indoor Navigation for a Humanoid Robot Using a View Sequence," *The International Journal of Robotics Research*, vol. 28, no. 2, pp. 315–325, 2009.
- [76] T. Inc., "Sky-Trax," 2010. [Online]. Available: <http://www.totaltraxinc.com/index.php/smart-forklift-solutions/forklift-tracking/sky-trax>. [Accessed: 01-May-2015].
- [77] HagiSonic, "Localization system StarGazer™ for Intelligent Robots." HagiSonic, 2008.

- [78] S. Lee and J. Song, "Mobile robot localization using infrared light reflecting landmarks," in *2007 International Conference on Control, Automation and Systems*, 2007, pp. 674–677.
- [79] D. Ashbrook and T. Starner, "Using GPS to learn significant locations and predict movement across multiple users," *Personal and Ubiquitous Computing*, vol. 7, no. 5, pp. 275–286, 2003.
- [80] D. Ashbrook and T. Starner, "Learning significant locations and predicting user movement with GPS," in *Proceedings. Sixth International Symposium on Wearable Computers*, 2002, pp. 101–108.
- [81] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell, "NextPlace: A spatio-temporal prediction framework for pervasive systems," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6696 LNCS, 2011, pp. 152–169.
- [82] T. F. Bechteler and H. Yenigün, "2-D localization and identification based on SAW ID-tags at 2.5 GHz," *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, no. 5, pp. 1584–1590, 2003.
- [83] C. V. Lopes, A. Haghghat, A. Mandal, T. Givargis, and P. Baldi, "Localization of off-the-shelf mobile devices using audible sound," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 10, p. 38, 2006.
- [84] I. Güvenç and C. C. Chong, "A survey on TOA based wireless localization and NLOS mitigation techniques," *IEEE Communications Surveys and Tutorials*, vol. 11, no. 3, pp. 107–124, 2009.
- [85] a. Stelzer, "Concept and application of LPM—a novel 3-D local position measurement system," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 12, pp. 2664–2669, 2004.
- [86] A. Kotanen, M. Hannikainen, H. Leppakoski, and T. D. Hamalainen, "Experiments on local positioning with Bluetooth," *Proceedings ITCC 2003. International Conference on Information Technology: Coding and Computing*, 2003.
- [87] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *16th Annual International Conference on Mobile Computing and Networking - (MobiCom '10)*, 2010, p. 173.
- [88] K. Jones, L. Liu, and F. Alizadeh-Shabdiz, "Improving wireless positioning with look-ahead map-matching," in *Proceedings of the 4th Annual International*

Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MobiQuitous 2007, 2007, pp. 1–8.

- [89] J. A. Bitsch Link, F. Gerdsmeyer, P. Smith, and K. Wehrle, “Indoor navigation on wheels (and on foot) using smartphones,” in *2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012 - Conference Proceedings*, 2012, pp. 1–10.
- [90] F. Tappero, “Low-cost optical-based indoor tracking device for detection and mitigation of NLOS effects,” in *Procedia Chemistry*, 2009, vol. 1, no. 1, pp. 497–500.
- [91] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, “Place Lab: Device Positioning Using Radio Beacons in the Wild,” in *Pervasive Computing*, vol. 3468, 2005, pp. 116–133.
- [92] C. O. Savage, R. L. Cramer, and H. A. Schmitt, “TDOA Geolocation with the Unscented Kalman Filter,” in *2006 IEEE International Conference on Networking, Sensing and Control*, 2006, pp. 602 – 606.
- [93] P. A. Ekstrom, “An advance in geolocation by light,” *Memoirs of the National Institute of Polar Research, Special Issue*, vol. Special, no. 58, pp. 210–226, 2004.
- [94] D. Han, D. G. D. G. Andersen, M. Kaminsky, K. Papagiannaki, and S. Seshan, “Access Point Localization Using Local Signal Strength Gradient,” in *Passive and Active Network Measurement*, vol. 5448, 2009, pp. 99–108.
- [95] J. Yang, A. Varshavsky, H. Liu, Y. Chen, and M. Gruteser, “Accuracy characterization of cell tower localization,” in *Proceedings of the 12th ACM international conference on Ubiquitous computing*, 2010, pp. 223–226.
- [96] Skyhook Wireless, “Skyhook Wireless Technology Used in Revolutionary iPhone and iPod touch,” 2008. [Online]. Available: <http://www.skyhookwireless.com/press/skyhookapple.php>. [Accessed: 01-Nov-2013].
- [97] A. J. Durrant and M. J. Hill, “Technical assessment of the DL2s GPS data quality ,” 2005.
- [98] Riverloop Security, “KillerBee.” GitHub, 2014.
- [99] M. Maróti, P. Völgyesi, S. Dóra, B. Kusý, A. Nádas, Á. Lédeczi, G. Balogh, and K. Molnár, “Radio Interferometric Geolocation,” in *Proceedings of the 3rd*

- International Conference on Embedded Networked Sensor Systems*, 2005, pp. 1–12.
- [100] X. An, J. Wang, R. V. Prasad, and I. Niemegeers, “OPT: online person tracking system for context-awareness in wireless personal network,” in *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, 2006, pp. 47–54.
- [101] Atmel, “RZUSB Stick,” *Products, Wireless Connectivity, 802.15.4, Transceivers*. [Online]. Available: <http://www.atmel.com/tools/RZUSBSTICK.aspx>. [Accessed: 15-Oct-2015].
- [102] University of Hertfordshire Smart Systems Lab, “MultiSensor.” Google Play Store, 2013.
- [103] California Eastern Laboratories, “MeshConnect™ EM35x USB Sticks Datasheet,” 2015. [Online]. Available: http://meshconnect.cel.com/docs/default-source/data-sheets/em357_usb_sticks_ds.pdf?sfvrsn=34. [Accessed: 15-Oct-2015].
- [104] (IEEE), *IEEE Standard for Local and metropolitan area networks, Part 15.4: Low-Rate Wireless Personal Area Networks*. 2011.
- [105] (IEEE), “IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer,” *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*. pp. 1–225, 2012.
- [106] Atmel, “AT86RF230 Preliminary Datasheet,” 2007. .
- [107] A. Kotanen, M. Hannikainen, H. Leppakoski, and T. D. Hamalainen, “Experiments on local positioning with Bluetooth,” *Information Technology: Coding and Computing [Computers and Communications], 2003. Proceedings. ITCC 2003. International Conference on*. pp. 297–303, 2003.
- [108] T. K. Kohoutek, R. Mautz, and A. Donaubauer, “Real-time indoor positioning using range imaging sensors,” *Proceedings of SPIE*, vol. 7724, no. May, p. 77240K–77240K–8, 2010.
- [109] C. Schlaile, O. Meister, N. Frietsch, C. Keßler, J. Wendel, and G. F. Trommer, “Using natural features for vision based navigation of an indoor-VTOL MAV,” *Aerospace Science and Technology*, vol. 13, no. 7, pp. 349–357, 2009.
- [110] O. Maye, J. Schaffner, and M. Maaser, “An Optical Indoor Positioning System for the Mass Market,” in *WPNC’06*, 2006, no. Proc. of the 3rd Workshop on

Positioning, Navigation and Communication, pp. pp. 111–116.

- [111] M. Köhler, S. N. Patel, J. W. Summet, E. P. Stuntebeck, and G. D. Abowd, “TrackSense : Infrastructure Free Precise Indoor Positioning Using Projected Patterns,” *Pervasive Computing 5th International Conference, PERVASIVE 2007, Toronto, Canada, May 13-16, 2007. Proceedings*, pp. 334–350, 2007.
- [112] A. Soloviev and D. Venable, “Integration of GPS and vision measurements for navigation in GPS challenged environments,” in *Record - IEEE PLANS, Position Location and Navigation Symposium*, 2010, pp. 826–833.
- [113] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, “LANDMARC: Indoor Location Sensing Using Active RFID,” *Wireless Networks*, vol. 10, no. 6, pp. 701–710, 2004.
- [114] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, “EasyLiving: Technologies for Intelligent Environments,” *Lecture Notes in Computer Science*, vol. 1927, no. Chapter 2, pp. 1–18, 2002.
- [115] Now Wireless Limited, “MESH4G Metropolitan Wireless,” 2007. [Online]. Available: <http://mesh.nowwireless.com/technology.htm>. [Accessed: 20-May-2005].
- [116] J. Werb and C. Lanzl, “Designing a positioning system for finding things and people indoors,” *IEEE Spectrum*, vol. 35, no. 9, pp. 71–78, 1998.
- [117] B. Krach and P. Robertson, “Integration of foot-mounted inertial sensors into a bayesian location estimation framework,” *5th Workshop on Positioning, Navigation and Communication 2008, WPNC’08*, no. 2, pp. 55–61, 2008.
- [118] A. Bekkali, H. Sanson, and M. Matsumoto, “RFID indoor positioning based on probabilistic RFID map and Kalman Filtering,” *3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2007*, no. WiMob, pp. 1–3, 2007.
- [119] M. Bouet and G. Pujolle, “A range-free 3-D localization method for RFID tags based on virtual landmarks,” in *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2008, pp. 1–5.

Appendix 1

7.1. Location System Comparison

System / Lead Author	Technologies Discussed	Methodologies Used	Reported Accuracy	Coverage / Range	Additional Comments	Collated by	Document Reference
Kohoutek	Camera	Optical time of flight ranging and 3D model comparison	< 1 m	Scalable	-	[17]	[108]
Hile	Camera	Image database matching and recognition	30 cm	Scalable	-	[17]	[73]
Kitanov	Camera	Optical feature tracking	< 1 m	Scalable	-	[17]	[74]
Schlaile	Camera & Accelerometer / Gyro	Optical feature tracking & inertial measurement	10 cm per minute	Scalable	-	[17]	[109]
Ido	Camera	Image database matching and recognition	30 cm	Scalable	-	[17]	[75]
Maye	Camera & Compass	Image translation speed and direction & rotation measurement	1 %	Scalable	-	[17]	[110]
Sky-Trax	Camera	Optical fiducial recognition	2 - 30 cm	Scalable	-	[17]	[76]
StarGazer	Camera	Optical fiducial recognition	< 1 m	Scalable	-	[17]	[77]
Lee	Camera	Optical fiducial recognition and measurement	< 1 m	36 m ²	-	[17]	[78]
TrackSense	Camera	Projected grid measurement	4 cm	25 m ²	Cited accuracy is for range from a single wall. 2D position accuracy is reported at < 17cm	[17]	[111]
CLIPS	Camera	Projected grid measurement	0.5mm	36 m ²	-	[17]	[70]
NorthStar	Camera	Projected grid measurement	< 1 m	36 m ²	Formerly by Evolution Robotics, now acquired by iRobot	[17]	[71]
Tappero	Camera	Optical change measurement	< 1 m	30 m ²	-	[17]	[90]
Soloviev	Camera, GPS & Accelerometer	Optical feature tracking, GPS & inertial measurement	< 1 m	Scalable	-	[17]	[112]
Active Badge	Infrared beacons	Time Of Arrival trilateration	7 cm	5 m	Uncertain how Al Nuaimi and Kamel arrived at the stated accuracy and range figures as system reports rooms or features that the system is at	[12], [16]	[40]
Active Bats	Ultrasonic beacons	Time Of Arrival trilateration	9 cm	50 m		[12], [16]	[39]

Cricket	Ultrasonic beacons & Radio frequency beacons	Time Difference Of Arrival trilateration	2 cm	10 m		[12], [16]	[38]
Dolphin	Ultrasonic beacons & Radio frequency beacons	Time Difference Of Arrival trilateration	2 cm	Room		[12]	[37]
RADAR	Radio frequency networks	Received Signal Strength trilateration	< 3 m	Room		[12], [16], [19]	[50]
Wave LAN	Radio frequency networks	Scene analysis	3 m	Room		[12]	[58]
LANDMARC	Radio frequency beacons	Proximity analysis	< 2 m	50 m		[12], [18], [19]	[113]
Horus	Radio frequency networks	Scene analysis	2 m	10 m	Later papers present a refined Horus system with 0.6m accuracy	[12], [19]	[67]
COMPASS	Radio frequency networks & Compass	Received Signal Strength trilateration & orientation filtering	< 2 m	15 m		[12], [16]	[51]
Beaugard	Accelerometer & GPS	Inertial measurement	10 m	Room		[12], [20]	[49]
FootSLAM	Accelerometer	Inertial measurement	< 3 m	2 m		[12], [20]	[42]
Fischer	Ultrasonics & Accelerometer	Ultrasonic ranging and inertial measurement	< 1 m	3 m		[12]	[48]
Woodman	Radio frequency networks & Accelerometer	Received Signal Strength trilateration & inertial measurement	50 cm	Building	Al Nuaimi and Kamel incorrectly label this as the [Active] Bat System as proposed by Priyantha, Chakaraborty and Balakrishnan	[12], [20]	[44]
IRIS_LPS	Stereo camera, Infrared beacons	Angle of Arrival triangulation	< 16 mm	< 135 m ²		[16]	40
Sonitor	Ultrasonic beacons	Proximity analysis	"Room Level"	Building		[16]	46
Ekahau	Radio frequency networks	Scene analysis	< 1 m	Scalable	Option to increase accuracy with "infrared location beacons"	[16], [19]	58
Topaz	Bluetooth & Infrared	Received Signal Strength trilateration & Proximity analysis	< 3 m	Scalable		[16], [19]	[32]
OPT	IEEE 802.15.4	Collaborative Received Signal Strength Proximity analysis	< 4 m	Scalable	Taken from the perspective of tracking one node within the network	[16]	[100]
Ubisense	Radio frequency beacons	Time Difference Of Arrival trilateration & Angle of Arrival trilateration	15 cm	100 m - 1000 m		[16], [19]	[64]
Easy Living	Stereo camera, radar	Optical feature tracking	Variable	Room	This is more predominantly a system for utilising tracking data	[16]	[114]
Beep	Audio beacon and / or radio	Time of Flight trilateration	< 0.5 cm	Room		[16]	[83]

	frequency beacons						
DIT	Radio frequency networks	Received Signal Strength trilateration	3 m	< 625 m ²		[19]	[54]
SnapTrack	GPS / Assisted-GPS	Time Difference Of Arrival trilateration	5 m - 50 m	?	Details now unavailable following acquisition	[19]	-
Sappire Dart	UWB Radio frequency beacons	Time Difference of Arrival trilateration & orientation filtering	< 0.3 m	< 200 m	Now owned by Zebra Technologies and called Dart UWB	[19]	[63]
SmartLOCUS	Radio frequency networks & Ultrasonic beacons	Received Signal Strength trilateration & Time Of Flight trilateration	< 15 cm	Scalable		[19]	[53]
EIRIS	Infrared beacons & Radio frequency beacons	?	< 1 m	?	Details unavailable and references invalid	[19]	-
SpotON	Radio frequency beacons	Received Signal Strength trilateration	< 1 m	Scalable	Accuracies are theoretical only	[18], [19]	[52]
MPS	Quad Channel Military Radio	Time of Flight & triangulation	+/- 10 m	Scalable		[19]	[115]
GPPS	DECT cellular system	Proximity analysis	< 10 m	National		[19]	[34]
Robot-based (Ladd)	Radio frequency networks	Received Signal Strength trilateration	1.5 m	Building		[19]	[55]
Robot-based (Haerberlen)	Radio frequency networks	Received Signal Strength trilateration	5.5 m	Building		[19]	[56]
Robot-based (Xiang)	Radio frequency networks	Received Signal Strength trilateration	2 m (static) 5 m (dynamic)	Building		[19]	[57]
MultiLoc	Radio frequency networks	Received Signal Strength trilateration	2.7 m	Room		[19]	[58]
TIX	Radio frequency networks	Received Signal Strength trilateration	5.4 m	1020 m ²		[19]	[59]
PinPoint 3D-ID	Radio frequency beacons	Time Of Flight trilateration	1 m	Building		[19]	[116]
GSM Fingerprinting	GSM cellular network	Scene analysis	5 m	Building		[19]	[68]
Klepal	Accelerometer	Dead reckoning and map matching	"Corridor width"	Building		[20]	[117]
SAW ID-tags	Surface Acoustic Wave	Time Of Arrival trilateration	+/- 0.2 m	< 100 m ²		[18]	[82]
LPM	Radio frequency beacons	Time Difference of Arrival	< 10 cm	Race track (< 500 m ²)		[18]	[85]
RSP	Radio frequency identification	Angle of Arrival triangulation	6 cm	Conveyor belt		[18]	[62]
VIRE	Radio frequency identification	Proximity analysis	< 0.47 m	Room		[18]	[35]
Simplex	Radio frequency identification	Scene analysis	< 1 m (passive) < 0.15 m (active)	Room		[18]	[66]
Bekkali	Radio frequency identification	Multilateration & Kalman Filtering	Unstated	Unstated	No reference to accuracy metrics	[18]	[118]
Scout	Radio frequency identification	Received Signal Strength trilateration	< 10 m	"large, outdoor"		[18]	[60]
3-D Constraints	Radio frequency identification	Proximity analysis	< 1.2 m	Scalable	Also estimates uncertainty	[18]	[119]

Table 15: Comprehensive comparison of indoor location systems as contrasted in summary literature works

Appendix 2

8.1. Prototyping platform versus development from scratch

The author had access to facilities for rapid prototyping and PCB development. This included Mentor Graphics and Altium design and simulation software packages, RF modelling tools, component stores (and trade accounts with suppliers), PCB milling, etching and tin plating facilities, soldering and reflow work stations and various machining or 3D plastic / metal printing capabilities. Using the sponsor's facilities, and with a well established design, it would have been possible to turn out an unpopulated PCB in a matter of hours and a fully enclosed and working product in a day or perhaps two.

In contrast to designing bespoke PCBs, there are numerous and varied prototyping platforms and modules developed for quickly and easily experimenting with project concepts. These are typically reasonably priced and adaptable enough to meet most needs out of the box. They are designed to provide a convenient and resource effective way to design a product or solution which can then be transformed into a production version at a later stage. The silicon manufacturers' who principally design and retail the prototyping boards intend that the product designers will go on to use the same components and processors in the final versions of their products. They anticipate large volume productions of these products thus demanding purchases of high quantities of the silicon manufacturer's goods.

There are several considerations, both positive and negative, to producing a custom development environment versus using a prebuilt prototyping module, the main of which are discussed below.

- A custom design provides complete flexibility and scope for expansion with a high degree of knowledge of the system and component interoperability. The flipside of this however is that the finesse, quality and features of the concept design are potentially limited by the skill and experience of the designer. These are skills that can be developed through training, research and experimentation but each of these takes additional time and could prolong the project if aspects of the design require a degree of personal development.
- Battery saving and power reduction can be much easier to achieve with a custom circuit; typically prototyping platforms and development boards are built for wide audiences with a large scope of possible usage scenarios. Because of this they normally provide a large number of peripherals, oscillators and over-specified processors that are not utilised by the project at hand. By using a custom circuit the power required by these

unused features can be saved and other options such as low power modes, hardware separation and control of modular sections and carefully specified components can be used to further reduce power demands.

- One of the biggest disadvantages to designing a bespoke circuit for developing a solution to an IEEE 802.15.4 location tool is the time required to develop and test the hardware. This would be needed before any investigation into the effectiveness of the IEEE 802.15.4 protocol and devices as a location tool can begin. This is the barrier that prototyping boards are designed to remove (or at the very least reduce) and is their greatest selling factor.
- Cost is a difficult attribute to compare between the two approaches; the component bill for a custom development can be several orders of magnitude lower versus a prototyping board. However the component cost accumulated following multiple redesigns, amendments, improvements and expansions can soon compound to a far greater figure. Additionally the cost in labour can be far higher for a bespoke design unless a poor choice of prototyping board is made, in which case manipulating the environment to suit the task can also take a considerable time in some cases.
- Unless external manufacture is tendered and procured, the build quality and finish of an in-house production will not match that of a commercial off the shelf prototyping board. This has implications for longevity of use and for resilience to environmental conditions.

On balance for this project, it was chosen to utilise prototyping boards and commercial off the shelf equipment in so far as possible. Wherever feasible, practicable and suitable, external circuitry was also to be commercial off the shelf modules or existing and proven designs. Time was at a premium to complete a project of this scale and the ability to remove the time taken in designing and testing a hardware platform before a usable system can be implemented for testing was a key advantage.

Appendix 3

9.1. Background scan of 2.4 GHz band

This report contains an amount of automated text content by the WiSpy Channelizer software. Only approximately the last fifteen seconds (from fifteen minutes) of data was output for the waterfall graph in order to reduce the report length.

Background Scan



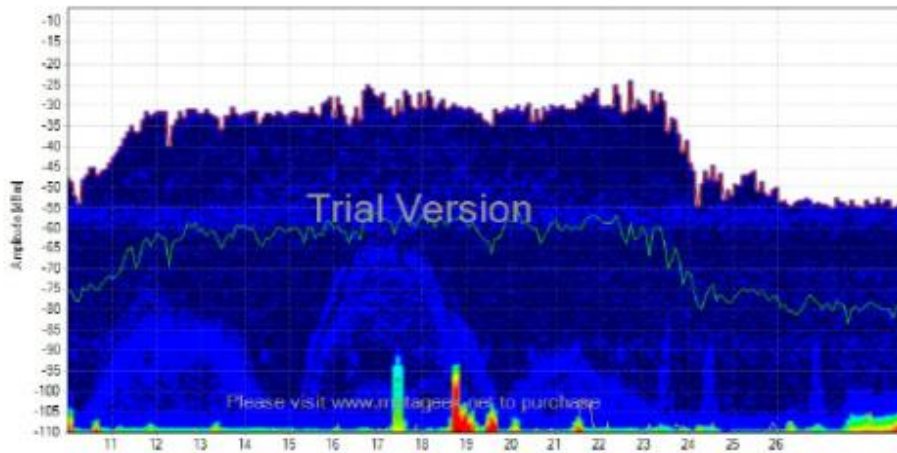
Site Info: **Home Office Centre for Applied Science and Technology**

Prepared By: **T Perrin**

Prepared For: **CAST EEE Functional Home**

Date: **09/10/2014 15:53:07**

Density Graph



The **Density View** maps and displays what is currently happening in the spectrum, so you can identify devices, see how loud they are, and see how often they are transmitting.

With **Color by Utilization** enabled, the height of the graph shows how loud devices are (amplitude), and the intensity of the color shows how often signals are occurring. The more intense the color, the more often the frequency is in use. This is called **utilization**, which is similar to **duty cycle** and **airtime usage**. For example, if a frequency has 40 percent utilization, it is only free for use by other transmitters for 60 percent of the time.

A blue spike or shape indicates a short signal, like a clap. A red spike or shape indicates a long, continuous signal, like an air horn.

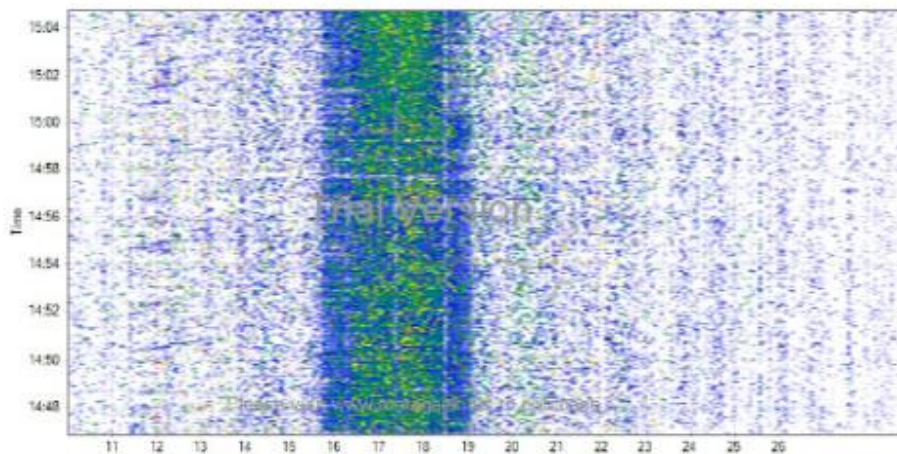
Blue - Less than 10 percent utilization

Green - 20 percent utilization

Yellow - 40 percent utilization

Red - Over 50 percent utilization

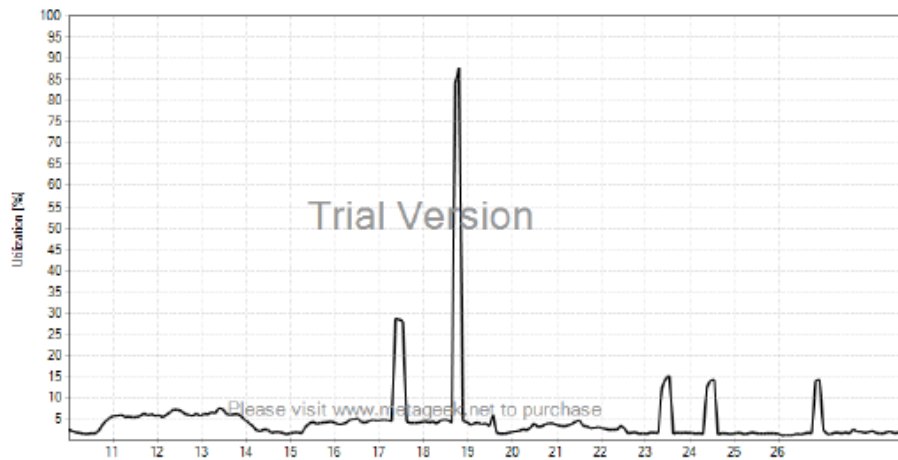
Waterfall Graph



The **Waterfall View** graphs amplitude over time for all frequencies in the selected band, much like a seismometer graphs earthquakes. This view is useful for watching the spectrum over time.

Unlike the Density View which uses Color by Utilization, the intensity of the color in the Waterfall View indicates amplitude. Blue indicates low-amplitude signals, while red indicates high-amplitude signals.

Utilization Graph



Utilization measures the percentage of activity above a defined amplitude threshold. Utilization is similar to **airtime usage** and **duty cycle**. The **Utilization Graph** gives more exact representations of utilization in the spectrum than the Density View's approximations.

Spectrum Analysis Overview

Most wireless networks and devices today use radio frequency (RF) technology to transmit data and certain types of devices use different sections or "bands" for transmission. Wi-Fi equipment has been allocated by international governing bodies to use certain unlicensed sections of the RF spectrum - specifically at 2.4 and at 5 GHz - for its operation. Being unlicensed, these bands are shared between many, many different kinds of devices and are the only section of spectrum where they are legally allowed to transmit RF signals. In environments like offices, warehouses or high-tech residences, where several wireless devices vie for the same spectrum space to communicate, interference can occur and networks become slow, drop connection or crash.

Why spectrum analysis?

Since RF signals are invisible to the naked eye, a spectrum analyzer (like Wi-Spy) is necessary to see into the wireless landscape to observe what is transmitting and where in the spectrum the "noise" is occurring. Sometimes the solution is to change the channel of the Wi-Fi network to avoid the other signals, and sometimes eliminating the offending wireless devices that "don't play well with others" is the answer. Occasionally, in situations where interference cannot be avoided or eliminated, the only solution is to switch Wi-Fi bands completely. Without spectrum analysis, implementing the proper solution is an expensive and time-consuming game of trial and error.

2.4 GHz Overview

The 2.4 GHz band contains eleven channels (and up to 13 or 14 in Europe and Japan, respectively), but only channels 1, 6, and 11 do not overlap. The 2.4 GHz band offers the widest compatibility with Wi-Fi devices through 802.11b/g/n, but with only three channels to chose from, the 2.4 GHz band is often very limited in the amount of traffic that it can sustain. While the 2.4 GHz band offers good range, it also suffers from non-Wi-Fi interference caused by electronic devices like cordless phones, security systems, microwave ovens, wireless audio-visual systems, Bluetooth devices, and more.

Channels Table

Channel	Grade	Utilization	Average (dBm)	Current (dBm)	Max (dBm)	Noise Floor (dBm)	Access Points
1	96.6	4.9%	-62.5	-110.5	-33.5	-109.5	0
2	96.4	4.7%	-61.0	-110.5	-32.5	-109.5	0
3	96.0	4.4%	-60.5	-110.5	-31.5	-109.5	0
4	95.2	5.3%	-59.5	-109.5	-30.5	-109.0	0
5	94.5	5.0%	-58.5	-109.5	-29.5	-109.0	0
6	94.3	8.6%	-58.5	-107.0	-29.5	-108.0	0
7	94.7	8.3%	-58.0	-107.0	-29.5	-108.0	0
8	95.8	8.0%	-58.5	-107.0	-30.0	-108.0	0
9	96.5	6.3%	-58.5	-106.5	-30.0	-108.5	0
10	97.0	3.4%	-59.5	-108.5	-29.5	-109.5	0
11	97.2	3.8%	-60.0	-109.0	-30.0	-109.5	0

12	97.3	3.4%	-61.5	-109.0	-31.0	-110.0	0
13	97.6	3.0%	-65.0	-110.0	-33.0	-110.0	0
14	0.0	0.0%	0.0	0.0	0.0	0.0	0

The **Channels Table** grades each Wi-Fi channel based on the RF activity within the defined time span. This table is useful for making channel deployment decisions, because it considers all activity in each channel, and gives each one a relative grade of usability.

9.2. Detection scan of 2.4 GHz band

This report contains an amount of automated text content by the WiSpy Channelizer software. Only approximately the last fifteen seconds (from fifteen minutes) of data was output for the waterfall graph in order to reduce the report length.

ZigBee Scan



Site Info: **Home Office Centre for Applied Science and Technology**

Prepared By: **T Perrin**

Prepared For: **CAST EEE Functional Home**

Date: **09/10/2014 15:51:56**

Density Graph



The **Density View** maps and displays what is currently happening in the spectrum, so you can identify devices, see how loud they are, and see how often they are transmitting.

With **Color by Utilization** enabled, the height of the graph shows how loud devices are (amplitude), and the intensity of the color shows how often signals are occurring. The more intense the color, the more often the frequency is in use. This is called **utilization**, which is similar to **duty cycle** and **airtime usage**. For example, if a frequency has 40 percent utilization, it is only free for use by other transmitters for 60 percent of the time.

A blue spike or shape indicates a short signal, like a clap. A red spike or shape indicates a long, continuous signal, like an air horn.

Blue - Less than 10 percent utilization

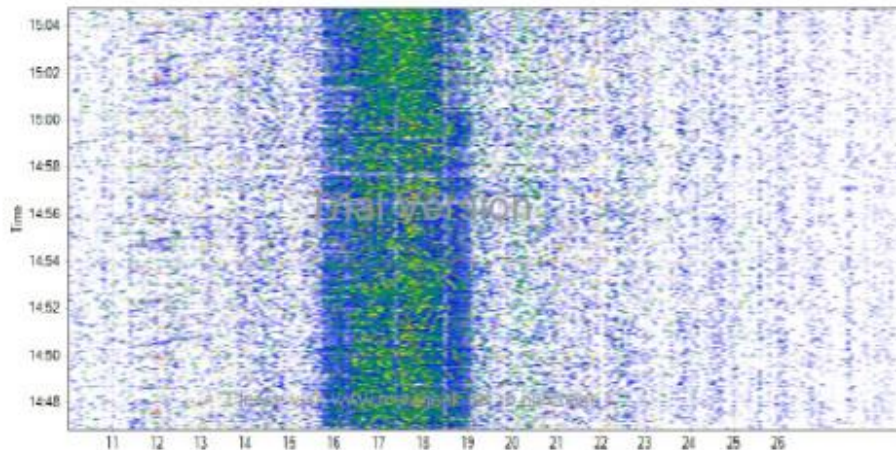
Green - 20 percent utilization

Yellow - 40 percent utilization

Red - Over 50 percent utilization

The peak at Channel 12 show the ZigBee network set up and operating during the test.

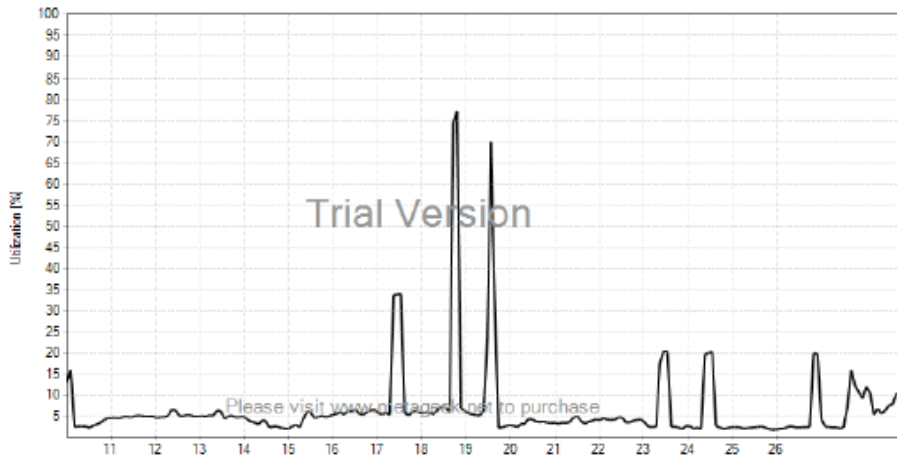
Waterfall Graph



The **Waterfall View** graphs amplitude over time for all frequencies in the selected band, much like a seismometer graphs earthquakes. This view is useful for watching the spectrum over time.

Unlike the Density View which uses Color by Utilization, the intensity of the color in the Waterfall View indicates amplitude. Blue indicates low-amplitude signals, while red indicates high-amplitude signals.

Utilization Graph



Utilization measures the percentage of activity above a defined amplitude threshold. Utilization is similar to **airtime usage** and **duty cycle**. The **Utilization Graph** gives more exact representations of utilization in the spectrum than the Density View's approximations.

Spectrum Analysis Overview

Most wireless networks and devices today use radio frequency (RF) technology to transmit data and certain types of devices use different sections or "bands" for transmission. Wi-Fi equipment has been allocated by international governing bodies to use certain unlicensed sections of the RF spectrum - specifically at 2.4 and 5 GHz - for its operation. Being unlicensed, these bands are shared between many, many different kinds of devices and are the only section of spectrum where they are legally allowed to transmit RF signals. In environments like offices, warehouses or high-tech residences, where several wireless devices vie for the same spectrum space to communicate, interference can occur and networks become slow, drop connection or crash.

Why spectrum analysis?

Since RF signals are invisible to the naked eye, a spectrum analyzer (like Wi-Spy) is necessary to see into the wireless landscape to observe what is transmitting and where in the spectrum the "noise" is occurring. Sometimes the solution is to change the channel of the Wi-Fi network to avoid the other signals, and sometimes eliminating the offending wireless devices that "don't play well with others" is the answer. Occasionally, in situations where interference cannot be avoided or eliminated, the only solution is to switch Wi-Fi bands completely. Without spectrum analysis, implementing the proper solution is an expensive and time-consuming game of trial and error.

2.4 GHz Overview

The 2.4 GHz band contains eleven channels (and up to 13 or 14 in Europe and Japan, respectively), but only channels 1, 6, and 11 do not overlap. The 2.4 GHz band offers the widest compatibility with Wi-Fi devices through 802.11b/g/n, but with only three channels to choose from, the 2.4 GHz band is often very limited in the amount of traffic that it can sustain. While the 2.4 GHz band offers good range, it also suffers from non-Wi-Fi interference caused by electronic devices like cordless phones, security systems, microwave ovens, wireless audio-visual systems, Bluetooth devices, and more.

Channels Table

Channel	Grade	Utilization	Average (dBm)	Current (dBm)	Max (dBm)	Noise Floor (dBm)	Access Points
1	94.9	4.4%	-56.0	-106.5	-27.0	-106.0	0
2	94.8	4.5%	-55.5	-109.0	-27.0	-106.0	0
3	94.5	4.6%	-58.5	-109.0	-30.5	-106.0	0
4	93.5	6.3%	-58.0	-109.0	-30.0	-106.0	0
5	92.3	6.6%	-57.5	-109.5	-29.0	-106.0	0
6	91.8	11.4%	-57.0	-107.0	-29.0	-106.5	0
7	92.3	11.6%	-57.0	-106.5	-29.0	-106.5	0
8	93.6	11.1%	-57.5	-106.5	-30.0	-106.5	0
9	94.8	9.0%	-57.5	-106.5	-29.5	-107.0	0

10	95.6	6.7%	-58.0	-107.0	-29.0	-107.5	0
11	95.9	5.3%	-59.0	-109.0	-29.0	-108.0	0
12	96.2	5.0%	-60.0	-109.0	-30.0	-108.5	0
13	96.5	4.6%	-62.5	-109.5	-31.5	-109.0	0
14	0.0	0.0%	0.0	0.0	0.0	0.0	0

The **Channels Table** grades each Wi-Fi channel based on the RF activity within the defined time span. This table is useful for making channel deployment decisions, because it considers all activity in each channel, and gives each one a relative grade of usability.

Appendix 4

10.1. Installing the KillerBee Environment to a Raspberry Pi

The following steps outline the steps and procedures that were undertaken during the installation of the KillerBee libraries and scripts to a Raspberry Pi. Although the Raspberry Pi platform was only used during an initial investigation into its merits, the process outlined below is portable to other linux platforms. Ultimately a very similar approach to this was used during the communications and range testing using a laptop.

The latest installation information and source files can be found at the KillerBee GitHub repository: <https://github.com/riverloopsec/killerbee> and is maintained by R Speers of Riverloop Security.

10.1.1. Console export

```
pi@Voyager ~ $ sudo apt-get update

Hit http://raspberrypi.collabora.com wheezy Release.gpg
Hit http://archive.raspberrypi.org wheezy Release.gpg
Get:1 http://mirrordirector.raspbian.org wheezy Release.gpg [490 B]
Hit http://raspberrypi.collabora.com wheezy Release
Get:2 http://mirrordirector.raspbian.org wheezy Release [14.4 kB]
Hit http://archive.raspberrypi.org wheezy Release
Hit http://raspberrypi.collabora.com wheezy/rpi armhf Packages
Get:3 http://mirrordirector.raspbian.org wheezy/main armhf Packages [6,894 kB]
Hit http://archive.raspberrypi.org wheezy/main armhf Packages
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en_GB
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en
Ign http://archive.raspberrypi.org wheezy/main Translation-en_GB
Ign http://archive.raspberrypi.org wheezy/main Translation-en
Hit http://mirrordirector.raspbian.org wheezy/contrib armhf Packages
Hit http://mirrordirector.raspbian.org wheezy/non-free armhf Packages
Hit http://mirrordirector.raspbian.org wheezy/rpi armhf Packages
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en
Fetched 6,908 kB in 50s (136 kB/s)
Reading package lists... Done
```

```
pi@Voyager ~ $ sudo apt-get upgrade

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  base-files curl file libc-bin libc-dev-bin libc6 libc6-dev libcurl3
  libcurl3-gnutls libmagic1 locales multiarch-support tzdata
13 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 15.5 MB of archives.
After this operation, 1,024 B of additional disk space will be used.
Do you want to continue [Y/n]? Y
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main base-files armhf 7.1wheezy8+rp1 [67.7 kB]
Get:2 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libc6-dev armhf 2.13-38+rp12+deb7u6 [2,428 kB]
Get:3 http://mirrordirector.raspbian.org/raspbian/ wheezy/main file armhf 5.11-2+deb7u7 [53.0 kB]
Get:4 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libmagic1 armhf 5.11-2+deb7u7 [201 kB]
Get:5 http://mirrordirector.raspbian.org/raspbian/ wheezy/main curl armhf 7.26.0-1+wheezy12 [268 kB]
Get:6 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libcurl3 armhf 7.26.0-1+wheezy12 [316 kB]
Get:7 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libc6 armhf 2.13-38+rp12+deb7u6 [4,116 kB]
Get:8 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libcurl3-gnutls armhf 7.26.0-1+wheezy12 [307 kB]
Get:9 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libc-dev-bin armhf 2.13-38+rp12+deb7u6 [223 kB]
Get:10 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libc-bin armhf 2.13-38+rp12+deb7u6 [1,205 kB]
Get:11 http://mirrordirector.raspbian.org/raspbian/ wheezy/main locales all 2.13-38+rp12+deb7u6 [5,711 kB]
```

```

Get:12 http://mirrordirector.raspbian.org/raspbian/ wheezy/main multiarch-support armhf 2.13-38+rpi2+deb7u6 [151 kB]
Get:13 http://mirrordirector.raspbian.org/raspbian/ wheezy/main tzdata all 2014j-0wheezy1 [442 kB]
Fetched 15.5 MB in 29s (532 kB/s)
Preconfiguring packages ...
(Reading database ... 74979 files and directories currently installed.)
Preparing to replace base-files 7.1wheezy6+rpi1 (using .../base-files_7.1wheezy8+rpi1_armhf.deb) ...
Unpacking replacement base-files ...
Processing triggers for install-info ...
Processing triggers for man-db ...
Setting up base-files (7.1wheezy8+rpi1) ...
Installing new version of config file /etc/debian_version ...
(Reading database ... 74979 files and directories currently installed.)
Preparing to replace libc6-dev:armhf 2.13-38+rpi2+deb7u3 (using .../libc6-dev_2.13-38+rpi2+deb7u6_armhf.deb) ...
Unpacking replacement libc6-dev:armhf ...
Preparing to replace libc6-dev-bin 2.13-38+rpi2+deb7u3 (using .../libc6-dev-bin_2.13-38+rpi2+deb7u6_armhf.deb) ...
Unpacking replacement libc6-dev-bin ...
Preparing to replace libc-bin 2.13-38+rpi2+deb7u3 (using .../libc-bin_2.13-38+rpi2+deb7u6_armhf.deb) ...
Unpacking replacement libc-bin ...
Processing triggers for man-db ...
Setting up libc-bin (2.13-38+rpi2+deb7u6) ...
(Reading database ... 74980 files and directories currently installed.)
Preparing to replace libc6:armhf 2.13-38+rpi2+deb7u3 (using .../libc6_2.13-38+rpi2+deb7u6_armhf.deb) ...
Unpacking replacement libc6:armhf ...
Setting up libc6:armhf (2.13-38+rpi2+deb7u6) ...
(Reading database ... 74980 files and directories currently installed.)
Preparing to replace file 5.11-2+deb7u6 (using .../file_5.11-2+deb7u7_armhf.deb) ...
Unpacking replacement file ...
Preparing to replace libmagic1:armhf 5.11-2+deb7u6 (using .../libmagic1_5.11-2+deb7u7_armhf.deb) ...
Unpacking replacement libmagic1:armhf ...
Preparing to replace curl 7.26.0-1+wheezy11 (using .../curl_7.26.0-1+wheezy12_armhf.deb) ...
Unpacking replacement curl ...
Preparing to replace libcurl3:armhf 7.26.0-1+wheezy11 (using .../libcurl3_7.26.0-1+wheezy12_armhf.deb) ...
Unpacking replacement libcurl3:armhf ...
Preparing to replace libcurl3-gnutls:armhf 7.26.0-1+wheezy11 (using .../libcurl3-gnutls_7.26.0-1+wheezy12_armhf.deb) ...
Unpacking replacement libcurl3-gnutls:armhf ...
Preparing to replace multiarch-support 2.13-38+rpi2+deb7u3 (using .../multiarch-support_2.13-38+rpi2+deb7u6_armhf.deb) ...
Unpacking replacement multiarch-support ...
Processing triggers for man-db ...
Setting up multiarch-support (2.13-38+rpi2+deb7u6) ...
(Reading database ... 74980 files and directories currently installed.)
Preparing to replace tzdata 2014h-0wheezy1 (using .../tzdata_2014j-0wheezy1_all.deb) ...
Unpacking replacement tzdata ...
Setting up tzdata (2014j-0wheezy1) ...

```

```

Current default time zone: 'Etc/UTC'
Local time is now: Sun Jan 11 16:05:45 UTC 2015.
Universal Time is now: Sun Jan 11 16:05:45 UTC 2015.
Run 'dpkg-reconfigure tzdata' if you wish to change it.

```

```

(Reading database ... 74983 files and directories currently installed.)
Preparing to replace locales 2.13-38+rpi2+deb7u3 (using .../locales_2.13-38+rpi2+deb7u6_all.deb) ...
Unpacking replacement locales ...
Processing triggers for man-db ...
Setting up libc-dev-bin (2.13-38+rpi2+deb7u6) ...
Setting up libc6-dev:armhf (2.13-38+rpi2+deb7u6) ...
Setting up libmagic1:armhf (5.11-2+deb7u7) ...
Setting up file (5.11-2+deb7u7) ...
Setting up libcurl3:armhf (7.26.0-1+wheezy12) ...
Setting up curl (7.26.0-1+wheezy12) ...
Setting up libcurl3-gnutls:armhf (7.26.0-1+wheezy12) ...
Setting up locales (2.13-38+rpi2+deb7u6) ...
Generating locales (this might take a while)...
en_GB.UTF-8... done
Generation complete.

```

pi@Voyager ~ \$ sudo apt-get install svn

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package svn

```

pi@Voyager ~ \$ sudo apt-get install subversion

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libapr1 libaprutil1 libneon27-gnutls libserf1 libsvn1
Suggested packages:
  subversion-tools db5.1-util
The following NEW packages will be installed:
  libapr1 libaprutil1 libneon27-gnutls libserf1 libsvn1 subversion
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,633 kB of archives.
After this operation, 6,941 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libapr1 armhf 1.4.6-3+deb7u1 [90.9 kB]
Get:2 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libaprutil1 armhf 1.4.1-3 [77.1 kB]
Get:3 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libserf1 armhf 1.1.0-2 [41.4 kB]
Get:4 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libneon27-gnutls armhf 0.29.6-3 [128 kB]
Get:5 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libsvn1 armhf 1.7.5-1+rpi1 [1,010 kB]
Get:6 http://mirrordirector.raspbian.org/raspbian/ wheezy/main subversion armhf 1.7.5-1+rpi1 [1,286 kB]
Fetched 2,633 kB in 6s (404 kB/s)
Selecting previously unselected package libapr1.
(Reading database ... 74983 files and directories currently installed.)
Unpacking libapr1 (from .../libapr1_1.4.6-3+deb7u1_armhf.deb) ...
Selecting previously unselected package libaprutil1.
Unpacking libaprutil1 (from .../libaprutil1_1.4.1-3_armhf.deb) ...
Selecting previously unselected package libserf1:armhf.
Unpacking libserf1:armhf (from .../libserf1_1.1.0-2_armhf.deb) ...
Selecting previously unselected package libneon27-gnutls.

```

```
Unpacking libneon27-gnutls (from .../libneon27-gnutls_0.29.6-3_armhf.deb) ...
Selecting previously unselected package libsvn1:armhf.
Unpacking libsvn1:armhf (from .../libsvn1_1.7.5-1+rpi1_armhf.deb) ...
Selecting previously unselected package subversion.
Unpacking subversion (from .../subversion_1.7.5-1+rpi1_armhf.deb) ...
Processing triggers for man-db ...
Setting up libapr1 (1.4.6-3+deb7u1) ...
Setting up libaprutil1 (1.4.1-3) ...
Setting up libserf1:armhf (1.1.0-2) ...
Setting up libneon27-gnutls (0.29.6-3) ...
Setting up libsvn1:armhf (1.7.5-1+rpi1) ...
Setting up subversion (1.7.5-1+rpi1) ...
```

pi@Voyager ~ \$ **sudo svn checkout http://killerbee.googlecode.com/svn/trunk/ killerbee-read-only**

```
A killerbee-read-only/killerbee
A killerbee-read-only/killerbee/LICENSE.txt
A killerbee-read-only/killerbee/tools
A killerbee-read-only/killerbee/tools/zboardrive
A killerbee-read-only/killerbee/tools/zbdump
A killerbee-read-only/killerbee/tools/zbkey
A killerbee-read-only/killerbee/tools/zbdnsniff
A killerbee-read-only/killerbee/tools/zbopenear
A killerbee-read-only/killerbee/tools/zbgoodfind
A killerbee-read-only/killerbee/tools/zbassocflood
A killerbee-read-only/killerbee/tools/zbreplay
A killerbee-read-only/killerbee/tools/zbid
A killerbee-read-only/killerbee/tools/zbstumbler
A killerbee-read-only/killerbee/tools/zbscapy
A killerbee-read-only/killerbee/tools/zbwiresniff
A killerbee-read-only/killerbee/tools/zbconvert
A killerbee-read-only/killerbee/tools/zbfind
A killerbee-read-only/killerbee/sample
A killerbee-read-only/killerbee/sample/control4-sample.txt
A killerbee-read-only/killerbee/sample/control4-sample.pcap
A killerbee-read-only/killerbee/sample/802154_encr_sample.dcf
A killerbee-read-only/killerbee/sample/zigbee-network-key-ota.dcf
A killerbee-read-only/killerbee/doc
A killerbee-read-only/killerbee/doc/toc-killerbee.GoodFETCCSPI-module.html
A killerbee-read-only/killerbee/doc/killerbee.GoodFET.GoodFETbtser-class.html
A killerbee-read-only/killerbee/doc/killerbee.openear.gps.gps'.gpsfix-class.html
A killerbee-read-only/killerbee/doc/killerbee.openear.gps.client.dictwrapper-class.html
A killerbee-read-only/killerbee/doc/killerbee.GoodFET-pysrc.html
A killerbee-read-only/killerbee/doc/killerbee.GoodFETatmel128.GoodFETatmel128rfa1-class.html
A killerbee-read-only/killerbee/doc/killerbee.openear.gps.misc-module.html
A killerbee-read-only/killerbee/doc/killerbee.GoodFETA VR.GoodFETA VR-class.html
A killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps-pysrc.html
A killerbee-read-only/killerbee/doc/killerbee.zboardrive.testGPS-pysrc.html
A killerbee-read-only/killerbee/doc/killerbee.daintree.DainTreeReader-class.html
A killerbee-read-only/killerbee/doc/killerbee.zboardrive-pysrc.html
A killerbee-read-only/killerbee/doc/killerbee.openear.scanner.CaptureThread-class.html
A killerbee-read-only/killerbee/doc/killerbee.kbutils.findFromList-class.html
A killerbee-read-only/killerbee/doc/killerbee.zboardrive.capture-pysrc.html
A killerbee-read-only/killerbee/doc/killerbee.GoodFET.GoodFET-class.html
A killerbee-read-only/killerbee/doc/killerbee.dev_rzusbstick-module.html
A killerbee-read-only/killerbee/doc/killerbee.dev_telosb-pysrc.html
A killerbee-read-only/killerbee/doc/toc-killerbee.zboardrive.scanning-module.html
A killerbee-read-only/killerbee/doc/toc-killerbee.openear.gps.gps'-module.html
A killerbee-read-only/killerbee/doc/killerbee.zboardrive.db-module.html
A killerbee-read-only/killerbee/doc/killerbee.kbutils.findFromListAndBusDevId -class.html
A killerbee-read-only/killerbee/doc/killerbee.config-module.html
A killerbee-read-only/killerbee/doc/killerbee.GoodFETCCSPI-module.html
A killerbee-read-only/killerbee/doc/toc-killerbee.GoodFET-module.html
A killerbee-read-only/killerbee/doc/killerbee.GoodFETA VR-pysrc.html
A killerbee-read-only/killerbee/doc/epydoc.js
A killerbee-read-only/killerbee/doc/killerbee-pysrc.html
A killerbee-read-only/killerbee/doc/killerbee.KillerBee-class.html
A killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps.gps'.gps-class.html
A killerbee-read-only/killerbee/doc/killerbee.dev_wislab-module.html
A killerbee-read-only/killerbee/doc/killerbee.dev_wislab.WISLAB-class.html
A killerbee-read-only/killerbee/doc/toc-killerbee.dev_apimote-module.html
A killerbee-read-only/killerbee/doc/killerbee.config-pysrc.html
A killerbee-read-only/killerbee/doc/killerbee.kbutils.KBInterfaceError-class.html
A killerbee-read-only/killerbee/doc/toc-killerbee.zboardrive.gps.misc-module.html
A killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps.gps'.gpsfix-class.html
A killerbee-read-only/killerbee/doc/killerbee.dev_telosb-module.html
A killerbee-read-only/killerbee/doc/killerbee.GoodFET-module.html
A killerbee-read-only/killerbee/doc/killerbee.openear.gps.gps'.gps-class.html
A killerbee-read-only/killerbee/doc/killerbee.openear.scanner-pysrc.html
A killerbee-read-only/killerbee/doc/toc-killerbee.zigbeedecode-module.html
A killerbee-read-only/killerbee/doc/killerbee.zboardrive.zboardrive-module.html
A killerbee-read-only/killerbee/doc/toc-killerbee.kbutils-module.html
A killerbee-read-only/killerbee/doc/killerbee.dblog-pysrc.html
A killerbee-read-only/killerbee/doc/toc-killerbee.dev_wislab-module.html
A killerbee-read-only/killerbee/doc/killerbee.kbutils.KBCapabilities-class.html
A killerbee-read-only/killerbee/doc/killerbee.openear.capture.CaptureThread-class.html
A killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps.client.gpscommon-class.html
A killerbee-read-only/killerbee/doc/killerbee.zboardrive.testGPS-module.html
A killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps-module.html
A killerbee-read-only/killerbee/doc/toc-killerbee.dot154decode-module.html
A killerbee-read-only/killerbee/doc/killerbee.zboardrive.zboardrive-pysrc.html
A killerbee-read-only/killerbee/doc/killerbee.openear.gps.client-pysrc.html
A killerbee-read-only/killerbee/doc/killerbee.pcapdump-module.html
A killerbee-read-only/killerbee/doc/toc-killerbee.dev_telosb-module.html
A killerbee-read-only/killerbee/doc/crarr.png
A killerbee-read-only/killerbee/doc/toc-killerbee.openear-module.html
A killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps.misc-pysrc.html
A killerbee-read-only/killerbee/doc/killerbee.kbutils-module.html
A killerbee-read-only/killerbee/doc/killerbee.zboardrive-module.html
A killerbee-read-only/killerbee/doc/killerbee.openear.gps.client.gpscommon-class.html
A killerbee-read-only/killerbee/doc/killerbee.openear.gps-module.html
A killerbee-read-only/killerbee/doc/toc-killerbee.zboardrive.db-module.html
```

A [killerbee-read-only/killerbee/doc/killerbee.openear.gps.misc-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps.client.gpsjson-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.kbutils-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.GoodFETatmel128-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.pcapdump.PcapDumper-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.openear-module.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.zboardrive.testGPS-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.openear.gps.client-module.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.openear.gps.misc-module.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.zboardrive-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps.gps.satellite-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.GoodFETatmel128-module.html](#)
A [killerbee-read-only/killerbee/doc/toc.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.dev_freakduino-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zigbeedecode-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zboardrive.db-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.zboardrive.zboardrive-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.GoodFETatmel128-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.openear.gps.client-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps.gps'-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zigbeedecode.ZigBeeNWKPacketParser-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dot154decode-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps.client.dictwrapper-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dblog-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.daintree-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.openear.gps.gps'-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.GoodFETAVR-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.GoodFETCCSPI.GoodFETCCSPI-class.html](#)
A [killerbee-read-only/killerbee/doc/module-tree.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps.gps'-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.kbutils.KBException-class.html](#)
A [killerbee-read-only/killerbee/doc/help.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zboardrive.db.ZBScanDB-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.scapy_extensions-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dblog.DBReader-class.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.openear.capture-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zboardrive.capture-module.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.GoodFETAVR-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps.client-module.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zboardrive.scanning-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.openear.capture-module.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.pcapdump-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dev_telosb.TELOB-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dev_zigduino.ZIGDUINO-class.html](#)
A [killerbee-read-only/killerbee/doc/api.pdf](#)
A [killerbee-read-only/killerbee/doc/killerbee.scapy_extensions-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dot154decode-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.openear.scanner.LocationThread-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.pcapdlt-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.openear.scanner-module.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.zboardrive.capture-module.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.dblog-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dev_rzusbstick-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.openear.gps.gps.satellite-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zboardrive.scanning-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.openear.scanner-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.daintree.DainTreeDumper-class.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.dev_rzusbstick-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.GoodFETCCSPI-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/epydoc.css](#)
A [killerbee-read-only/killerbee/doc/toc-everything.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.config-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dev_freakduino.FREAKDUINO-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dblog.DBLogger-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dev_apimote-module.html](#)
A [killerbee-read-only/killerbee/doc/frames.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.pcapdump.PcapReader-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps.client-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps.misc-module.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.openear.gps-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zigbeedecode.ZigBeeAPSPacketParser-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.daintree-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/identifier-index.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.zboardrive.gps.client-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zboardrive.gps.gps'.gpsdata-class.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.dev_zigduino-module.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.pcapdlt-module.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.daintree-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.openear.gps.gps'-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.openear.gps.gps'.gpsdata-class.html](#)
A [killerbee-read-only/killerbee/doc/class-tree.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dev_rzusbstick.RZUSBSTICK-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.GoodFET.SymbolTable-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.pcapdump-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dev_freakduino-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.pcapdlt-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dot154decode.Dot154PacketParserclass.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.zboardrive.gps-module.html](#)
A [killerbee-read-only/killerbee/doc/api-objects.txt](#)
A [killerbee-read-only/killerbee/doc/killerbee.openear.gps-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dev_freakduino-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.openear.gps.client.gpsjson-class.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dev_zigduino-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.openear-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.openear.capture-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dev_apimote-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dev_apimote.APIMOTE-class.html](#)
A [killerbee-read-only/killerbee/doc/redirect.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.scapy_extensions-module.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.zigbeedecode-pysrc.html](#)
A [killerbee-read-only/killerbee/doc/killerbee.dev_zigduino-module.html](#)
A [killerbee-read-only/killerbee/doc/index.html](#)
A [killerbee-read-only/killerbee/doc/toc-killerbee.zboardrive.gps.gps'-module.html](#)

```

A killerbee-read-only/killerbee/doc/killerbee_dev_wislab_pysrc.html
A killerbee-read-only/killerbee/doc/killerbee_zboardrive_capture.CaptureThread-class.html
A killerbee-read-only/killerbee/zigbee_crypt
A killerbee-read-only/killerbee/zigbee_crypt/zigbee_crypt.c
A killerbee-read-only/killerbee/zigbee_crypt/zigbee_crypt.h
A killerbee-read-only/killerbee/setup.py
A killerbee-read-only/killerbee/scripts
A killerbee-read-only/killerbee/scripts/zbfixupz
A killerbee-read-only/killerbee/scripts/sfuzzex.py
A killerbee-read-only/killerbee/scripts/create_db.sql
A killerbee-read-only/killerbee/scripts/zbsendone
A killerbee-read-only/killerbee/scripts/bootloader_test
A killerbee-read-only/killerbee/scripts/qbp.py
A killerbee-read-only/killerbee/scripts/configure_wislab.py
A killerbee-read-only/killerbee/scripts/update-dft.sh
A killerbee-read-only/killerbee/scripts/zbtstpkts
A killerbee-read-only/killerbee/README.txt
A killerbee-read-only/killerbee/firmware
A killerbee-read-only/killerbee/firmware/gf-telosb-001.hex
A killerbee-read-only/killerbee/firmware/flash_apimote.sh
A killerbee-read-only/killerbee/firmware/gf-zigduino.hex
A killerbee-read-only/killerbee/firmware/goodfet.bsl
A killerbee-read-only/killerbee/firmware/apimotev4_gf.hex
A killerbee-read-only/killerbee/firmware/kb-rzusbstick-001.hex
A killerbee-read-only/killerbee/firmware/flash_telosb.sh
A killerbee-read-only/killerbee/firmware/flash_zigduino.sh
A killerbee-read-only/killerbee/killerbee
A killerbee-read-only/killerbee/killerbee/zboardrive
A killerbee-read-only/killerbee/killerbee/zboardrive/testGPS.py
A killerbee-read-only/killerbee/killerbee/zboardrive/__init__.py
A killerbee-read-only/killerbee/killerbee/zboardrive/zboardrive.py
A killerbee-read-only/killerbee/killerbee/zboardrive/scanning.py
A killerbee-read-only/killerbee/killerbee/zboardrive/gps
A killerbee-read-only/killerbee/killerbee/zboardrive/gps/gps.py
A killerbee-read-only/killerbee/killerbee/zboardrive/gps/misc.py
A killerbee-read-only/killerbee/killerbee/zboardrive/gps/client.py
A killerbee-read-only/killerbee/killerbee/zboardrive/gps/__init__.py
A killerbee-read-only/killerbee/killerbee/zboardrive/capture.py
A killerbee-read-only/killerbee/killerbee/zboardrive/README.txt
A killerbee-read-only/killerbee/killerbee/zboardrive/db.py
A killerbee-read-only/killerbee/killerbee/zboardrive/Makefile
A killerbee-read-only/killerbee/killerbee/daintree.py
A killerbee-read-only/killerbee/killerbee/dev_apimote.py
A killerbee-read-only/killerbee/killerbee/scapy_extensions.py
A killerbee-read-only/killerbee/killerbee/__init__.py
A killerbee-read-only/killerbee/killerbee/GoodFETAVR.py
A killerbee-read-only/killerbee/killerbee/dev_rzusbstick.py
A killerbee-read-only/killerbee/killerbee/openear
A killerbee-read-only/killerbee/killerbee/openear/gps.py
A killerbee-read-only/killerbee/killerbee/openear/scanner.py
A killerbee-read-only/killerbee/killerbee/openear/__init__.py
A killerbee-read-only/killerbee/killerbee/openear/gps
A killerbee-read-only/killerbee/killerbee/openear/gps/gps.py
A killerbee-read-only/killerbee/killerbee/openear/gps/misc.py
A killerbee-read-only/killerbee/killerbee/openear/gps/client.py
A killerbee-read-only/killerbee/killerbee/openear/gps/__init__.py
A killerbee-read-only/killerbee/killerbee/openear/capture.py
A killerbee-read-only/killerbee/killerbee/openear/README
A killerbee-read-only/killerbee/killerbee/openear/Makefile
A killerbee-read-only/killerbee/killerbee/config.py
A killerbee-read-only/killerbee/killerbee/GoodFETatmel128.py
A killerbee-read-only/killerbee/killerbee/zigbeedecode.py
A killerbee-read-only/killerbee/killerbee/pcapdump.py
A killerbee-read-only/killerbee/killerbee/GoodFETCCSPI.py
A killerbee-read-only/killerbee/killerbee/kbutils.py
A killerbee-read-only/killerbee/killerbee/dot154decode.py
A killerbee-read-only/killerbee/killerbee/dev_sewio.py
A killerbee-read-only/killerbee/killerbee/dev_treakduino.py
A killerbee-read-only/killerbee/killerbee/dev_telosb.py
A killerbee-read-only/killerbee/killerbee/dev_zigduino.py
A killerbee-read-only/killerbee/killerbee/dblog.py
A killerbee-read-only/killerbee/killerbee/GoodFET.py
A killerbee-read-only/killerbee/killerbee/pcapdlt.py
Checked out revision 96.

```

pi@Voyager ~ \$ **sudo apt-get install python-gtk2 python-cairo python-usb python-crypto python-serial python-dev libgcrypt-dev**

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libgrypt11-dev' instead of 'libgrypt-dev'
python-serial is already the newest version.
The following extra packages will be installed:
  libexpat1-dev libgpg-error-dev libssl-dev libssl-doc python-gobject-2 python2.7-dev
Suggested packages:
  libgrypt11-doc python-crypto-dbg python-crypto-doc python-gobject-2-dbg python-gtk2-doc
The following NEW packages will be installed:
  libexpat1-dev libgrypt11-dev libgpg-error-dev libssl-dev libssl-doc python-cairo python-crypto python-dev python-gobject-2 python-gtk2 python-usb
python2.7-dev
0 upgraded, 12 newly installed, 0 to remove and 0 not upgraded.
Need to get 34.6 MB of archives.
After this operation, 52.2 MB of additional disk space will be used.

```

Do you want to continue [Y/n]?

```

Y
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libexpat1-dev armhf 2.1.0-1+deb7u1 [210 kB]
Get:2 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libgpg-error-dev armhf 1.10-3.1 [40.0 kB]
Get:3 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libgrypt11-dev armhf 1.5.0-5+deb7u2 [398 kB]
Get:4 http://mirrordirector.raspbian.org/raspbian/ wheezy/main python-dev all 2.7.3-4+deb7u1 [920 B]
Get:5 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libssl-dev armhf 1.0.1e-2+rvt+deb7u13 [1,504 kB]
Get:6 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libssl-doc all 1.0.1e-2+rvt+deb7u13 [1,205 kB]
Get:7 http://mirrordirector.raspbian.org/raspbian/ wheezy/main python-cairo armhf 1.8.8-1 [68.6 kB]
Get:8 http://mirrordirector.raspbian.org/raspbian/ wheezy/main python-crypto armhf 2.6.4-4+deb7u3 [522 kB]
Get:9 http://mirrordirector.raspbian.org/raspbian/ wheezy/main python2.7-dev armhf 2.7.3-6+deb7u2 [28.7 MB]

```



```

Get:10 http://mirrordirector.raspbian.org/raspbian/ wheezy/main python-gobject-2 armhf 2.28.6-10 [475 kB]
Get:11 http://mirrordirector.raspbian.org/raspbian/ wheezy/main python-gtk2 armhf 2.24.0-3 [1,450 kB]
Get:12 http://mirrordirector.raspbian.org/raspbian/ wheezy/main python-usb armhf 0.4.3-1 [17.7 kB]
Fetched 34.6 MB in 1min 18s (439 kB/s)
Selecting previously unselected package libexpat1-dev.
(Reading database ... 75105 files and directories currently installed.)
Unpacking libexpat1-dev (from .../libexpat1-dev_2.1.0-1+deb7u1_armhf.deb) ...
Selecting previously unselected package libpgp-error-dev.
Unpacking libpgp-error-dev (from .../libpgp-error-dev_1.10-3.1_armhf.deb) ...
Selecting previously unselected package libgpg-error-dev.
Unpacking libgpg-error-dev (from .../libgpg-error-dev_1.10-3.1_armhf.deb) ...
Selecting previously unselected package libgcrypt11-dev.
Unpacking libgcrypt11-dev (from .../libgcrypt11-dev_1.5.0-5+deb7u2_armhf.deb) ...
Selecting previously unselected package libssl-dev.
Unpacking libssl-dev (from .../libssl-dev_1.0.1e-2+rvt+deb7u13_armhf.deb) ...
Selecting previously unselected package libssl-doc.
Unpacking libssl-doc (from .../libssl-doc_1.0.1e-2+rvt+deb7u13_all.deb) ...
Selecting previously unselected package python-cairo.
Unpacking python-cairo (from .../python-cairo_1.8.8-1_armhf.deb) ...
Selecting previously unselected package python-crypto.
Unpacking python-crypto (from .../python-crypto_2.6-4+deb7u3_armhf.deb) ...
Selecting previously unselected package python2.7-dev.
Unpacking python2.7-dev (from .../python2.7-dev_2.7.3-6+deb7u2_armhf.deb) ...
Selecting previously unselected package python-dev.
Unpacking python-dev (from .../python-dev_2.7.3-4+deb7u1_all.deb) ...
Selecting previously unselected package python-gobject-2.
Unpacking python-gobject-2 (from .../python-gobject-2_2.28.6-10_armhf.deb) ...
Selecting previously unselected package python-gtk2.
Unpacking python-gtk2 (from .../python-gtk2_2.24.0-3_armhf.deb) ...
Selecting previously unselected package python-usb.
Unpacking python-usb (from .../python-usb_0.4.3-1_armhf.deb) ...
Processing triggers for man-db ...
Setting up libexpat1-dev (2.1.0-1+deb7u1) ...
Setting up libpgp-error-dev (1.10-3.1) ...
Setting up libgcrypt11-dev (1.5.0-5+deb7u2) ...
Setting up libssl-dev (1.0.1e-2+rvt+deb7u13) ...
Setting up libssl-doc (1.0.1e-2+rvt+deb7u13) ...
Setting up python-cairo (1.8.8-1) ...
Setting up python-crypto (2.6-4+deb7u3) ...
Setting up python2.7-dev (2.7.3-6+deb7u2) ...
Setting up python-dev (2.7.3-4+deb7u1) ...
Setting up python-gobject-2 (2.28.6-10) ...
Setting up python-gtk2 (2.24.0-3) ...
Setting up python-usb (0.4.3-1) ...
Processing triggers for python-support ...

```

pi@Voyager ~ \$ **dir**

Desktop killerbee-read-only python_games ustream

pi@Voyager ~ \$ **cd killerbee-read-only/killerbee/**

pi@Voyager ~/killerbee-read-only/killerbee \$ **dir**

doc firmware killerbee LICENSE.txt README.txt sample scripts setup.py tools zigbee_crypt

pi@Voyager ~/killerbee-read-only/killerbee \$ **sudo python setup.py install**

```

No setupools found, attempting to use distutils instead.
/usr/lib/python2.7/dist-packages/gtk-2.0/gtk/_init__.py:57: GtkWarning: could not open display
  warnings.warn(str(e), _gtk.Warning)
Note: You are using pyUSB 0.x. Consider upgrading to pyUSB 1.x.
/usr/lib/python2.7/distutils/dist.py:267: UserWarning: Unknown distribution option: 'install_requires'
  warnings.warn(msg)
running install
running build
running build_py
creating build
creating build/lib.linux-armv6l-2.7
creating build/lib.linux-armv6l-2.7/killerbee
copying killerbee/zigbeedecode.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/pcapdlt.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/kbutils.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/dev_apimote.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/dev_sewio.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/GoodFETatmel128.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/dev_zigduino.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/scapy_extensions.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/dev_rzusbstick.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/dot154decode.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/_init__.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/GoodFET.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/GoodFETCCSPI.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/dblog.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/pcapdump.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/daintree.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/dev_telosb.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/GoodFETAVR.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/dev_freakduino.py -> build/lib.linux-armv6l-2.7/killerbee
copying killerbee/config.py -> build/lib.linux-armv6l-2.7/killerbee
creating build/lib.linux-armv6l-2.7/killerbee/opencv
copying killerbee/opencv/gps.py -> build/lib.linux-armv6l-2.7/killerbee/opencv
copying killerbee/opencv/_init__.py -> build/lib.linux-armv6l-2.7/killerbee/opencv
copying killerbee/opencv/scanner.py -> build/lib.linux-armv6l-2.7/killerbee/opencv
copying killerbee/opencv/capture.py -> build/lib.linux-armv6l-2.7/killerbee/opencv
creating build/lib.linux-armv6l-2.7/killerbee/zboardrive
copying killerbee/zboardrive/testGPS.py -> build/lib.linux-armv6l-2.7/killerbee/zboardrive
copying killerbee/zboardrive/db.py -> build/lib.linux-armv6l-2.7/killerbee/zboardrive
copying killerbee/zboardrive/zboardrive.py -> build/lib.linux-armv6l-2.7/killerbee/zboardrive
copying killerbee/zboardrive/scanning.py -> build/lib.linux-armv6l-2.7/killerbee/zboardrive
copying killerbee/zboardrive/_init__.py -> build/lib.linux-armv6l-2.7/killerbee/zboardrive
copying killerbee/zboardrive/capture.py -> build/lib.linux-armv6l-2.7/killerbee/zboardrive

```

```
running build_ext
building 'zigbee_crypt' extension
creating build/temp.linux-armv6l-2.7
creating build/temp.linux-armv6l-2.7/zigbee_crypt
gcc -pthread -fno-strict-aliasing -DNDEBUG -g -fwrapv -O2 -Wall -Wstrict-prototypes -fPIC -I/usr/local/include -I/usr/include -I/sw/include/ -Izigbee_crypt -
I/usr/include/python2.7 -c zigbee_crypt/zigbee_crypt.c -o build/temp.linux-armv6l-2.7/zigbee_crypt/zigbee_crypt.o
gcc -pthread -shared -Wl,-O1 -Wl,-Bsymbolic-functions -Wl,-z,relro build/temp.linux-armv6l-2.7/zigbee_crypt/zigbee_crypt.o -L/usr/local/lib -L/sw/var/lib/ -
lgcrypt -o build/lib.linux-armv6l-2.7/zigbee_crypt.so
running build_scripts
creating build/scripts-2.7
copying and adjusting tools/zbdump -> build/scripts-2.7
copying and adjusting tools/zbgoodfind -> build/scripts-2.7
copying and adjusting tools/zbid -> build/scripts-2.7
copying and adjusting tools/zbreplay -> build/scripts-2.7
copying and adjusting tools/zbconvert -> build/scripts-2.7
copying and adjusting tools/zbsniff -> build/scripts-2.7
copying and adjusting tools/zbstumbler -> build/scripts-2.7
copying tools/zbassocflood -> build/scripts-2.7
copying and adjusting tools/zbfind -> build/scripts-2.7
copying and adjusting tools/zbscapy -> build/scripts-2.7
copying and adjusting tools/zbwireshark -> build/scripts-2.7
copying and adjusting tools/zbkey -> build/scripts-2.7
copying and adjusting tools/zboardrive -> build/scripts-2.7
copying and adjusting tools/zbopenear -> build/scripts-2.7
changing mode of build/scripts-2.7/zbdump from 644 to 755
changing mode of build/scripts-2.7/zbgoodfind from 644 to 755
changing mode of build/scripts-2.7/zbid from 644 to 755
changing mode of build/scripts-2.7/zbreplay from 644 to 755
changing mode of build/scripts-2.7/zbconvert from 644 to 755
changing mode of build/scripts-2.7/zbsniff from 644 to 755
changing mode of build/scripts-2.7/zbstumbler from 644 to 755
changing mode of build/scripts-2.7/zbassocflood from 644 to 755
changing mode of build/scripts-2.7/zbfind from 644 to 755
changing mode of build/scripts-2.7/zbscapy from 644 to 755
changing mode of build/scripts-2.7/zbwireshark from 644 to 755
changing mode of build/scripts-2.7/zbkey from 644 to 755
changing mode of build/scripts-2.7/zboardrive from 644 to 755
changing mode of build/scripts-2.7/zbopenear from 644 to 755
running install_lib
copying build/lib.linux-armv6l-2.7/zigbee_crypt.so -> /usr/local/lib/python2.7/dist-packages
creating /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/zigbeedecode.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/pcapdlt.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/kbutils.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/dev_apimote.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/dev_sewio.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/GoodFETatmel128.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/dev_zigduino.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/scapy_extensions.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/dev_rzusbstick.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/dot154decode.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/_init_.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/GoodFET.py -> /usr/local/lib/python2.7/dist-packages/killerbee
creating /usr/local/lib/python2.7/dist-packages/killerbee/openear
copying build/lib.linux-armv6l-2.7/killerbee/openear/gps.py -> /usr/local/lib/python2.7/dist-packages/killerbee/openear
copying build/lib.linux-armv6l-2.7/killerbee/openear/_init_.py -> /usr/local/lib/python2.7/dist-packages/killerbee/openear
copying build/lib.linux-armv6l-2.7/killerbee/openear/scanner.py -> /usr/local/lib/python2.7/dist-packages/killerbee/openear
copying build/lib.linux-armv6l-2.7/killerbee/openear/capture.py -> /usr/local/lib/python2.7/dist-packages/killerbee/openear
copying build/lib.linux-armv6l-2.7/killerbee/GoodFETCCSPI.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/dblog.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/pcapdump.py -> /usr/local/lib/python2.7/dist-packages/killerbee
creating /usr/local/lib/python2.7/dist-packages/killerbee/zboardrive
copying build/lib.linux-armv6l-2.7/killerbee/zboardrive/testGPS.py -> /usr/local/lib/python2.7/dist-packages/killerbee/zboardrive
copying build/lib.linux-armv6l-2.7/killerbee/zboardrive/db.py -> /usr/local/lib/python2.7/dist-packages/killerbee/zboardrive
copying build/lib.linux-armv6l-2.7/killerbee/zboardrive/zboardrive.py -> /usr/local/lib/python2.7/dist-packages/killerbee/zboardrive
copying build/lib.linux-armv6l-2.7/killerbee/zboardrive/scanning.py -> /usr/local/lib/python2.7/dist-packages/killerbee/zboardrive
copying build/lib.linux-armv6l-2.7/killerbee/zboardrive/_init_.py -> /usr/local/lib/python2.7/dist-packages/killerbee/zboardrive
copying build/lib.linux-armv6l-2.7/killerbee/zboardrive/capture.py -> /usr/local/lib/python2.7/dist-packages/killerbee/zboardrive
copying build/lib.linux-armv6l-2.7/killerbee/daintree.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/dev_telosb.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/GoodFETAVR.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/dev_freakduino.py -> /usr/local/lib/python2.7/dist-packages/killerbee
copying build/lib.linux-armv6l-2.7/killerbee/config.py -> /usr/local/lib/python2.7/dist-packages/killerbee
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/zigbeedecode.py to zigbeedecode.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/pcapdlt.py to pcapdlt.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/kbutils.py to kbutils.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/dev_apimote.py to dev_apimote.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/dev_sewio.py to dev_sewio.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/GoodFETatmel128.py to GoodFETatmel128.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/dev_zigduino.py to dev_zigduino.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/scapy_extensions.py to scapy_extensions.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/dev_rzusbstick.py to dev_rzusbstick.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/dot154decode.py to dot154decode.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/_init_.py to _init_.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/GoodFET.py to GoodFET.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/openear/gps.py to gps.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/openear/_init_.py to _init_.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/openear/scanner.py to scanner.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/openear/capture.py to capture.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/GoodFETCCSPI.py to GoodFETCCSPI.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/dblog.py to dblog.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/pcapdump.py to pcapdump.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/zboardrive/testGPS.py to testGPS.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/zboardrive/db.py to db.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/zboardrive/zboardrive.py to zboardrive.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/zboardrive/scanning.py to scanning.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/zboardrive/_init_.py to _init_.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/zboardrive/capture.py to capture.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/daintree.py to daintree.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/dev_telosb.py to dev_telosb.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/GoodFETAVR.py to GoodFETAVR.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/dev_freakduino.py to dev_freakduino.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/killerbee/config.py to config.pyc
running install_scripts
copying build/scripts-2.7/zbreplay -> /usr/local/bin
```

```
copying build/scripts-2.7/zbid -> /usr/local/bin
copying build/scripts-2.7/zbassocflood -> /usr/local/bin
copying build/scripts-2.7/zbkey -> /usr/local/bin
copying build/scripts-2.7/zbwireshark -> /usr/local/bin
copying build/scripts-2.7/zbdump -> /usr/local/bin
copying build/scripts-2.7/zbscap -> /usr/local/bin
copying build/scripts-2.7/zbopenear -> /usr/local/bin
copying build/scripts-2.7/zbconvert -> /usr/local/bin
copying build/scripts-2.7/zbdnsniff -> /usr/local/bin
copying build/scripts-2.7/zbwardrive -> /usr/local/bin
copying build/scripts-2.7/zbfind -> /usr/local/bin
copying build/scripts-2.7/zbgoodfind -> /usr/local/bin
copying build/scripts-2.7/zbstumbler -> /usr/local/bin
changing mode of /usr/local/bin/zbreplay to 755
changing mode of /usr/local/bin/zbid to 755
changing mode of /usr/local/bin/zbassocflood to 755
changing mode of /usr/local/bin/zbkey to 755
changing mode of /usr/local/bin/zbwireshark to 755
changing mode of /usr/local/bin/zbdump to 755
changing mode of /usr/local/bin/zbscap to 755
changing mode of /usr/local/bin/zbopenear to 755
changing mode of /usr/local/bin/zbconvert to 755
changing mode of /usr/local/bin/zbdnsniff to 755
changing mode of /usr/local/bin/zbwardrive to 755
changing mode of /usr/local/bin/zbfind to 755
changing mode of /usr/local/bin/zbgoodfind to 755
changing mode of /usr/local/bin/zbstumbler to 755
running install_egg_info
Writing /usr/local/lib/python2.7/dist-packages/killerbee-2.5.0.egg-info
```

Appendix 5

11.1. Installing new firmware to the RZUSB

As in Appendix 5, the customised KillerBee firmware required can be found in a GitHub repository: <https://github.com/riverloopsec/killerbee>.

Once downloaded and extracted, the .hex file can be programmed to the RZUSB Stick using Atmel's integrated development environment and their JTAGICE Mk II programmer. It is necessary to solder on a 50mm JTAG header to the RZUSB Stick as shown in Figure 45 and to use a 100mm to 50mm header adapter with the JTAGICE Mk II programmer as shown in Figure 46.

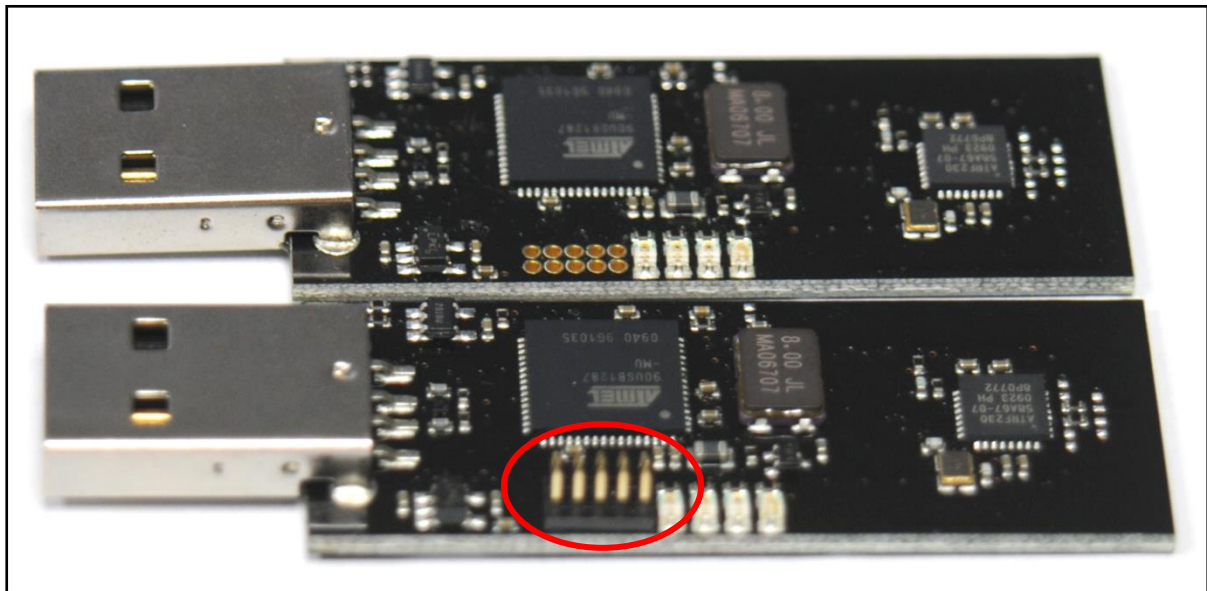


Figure 45: RZUSB Stick before (top) and after (bottom) fitting a 50mm, 5-pin x 2-row JTAG header



Figure 46: Atmel JTAGICE Mk II programmer with an additional 100mm to 50mm header adapter

Instructions for updating the firmware using their tools can be found with their source and project files zip file on Atmel's website at this address:

<http://www.atmel.com/tools/RZUSBSTICK.aspx?tab=overview>

Having attempted this route however the author found this to be a burdensome approach as opposed to using the Linux tool *AVRDude* and the Atmel JTAGICE Mk II as suggested by Riverloop Security and the KillerBee project (instructions are found on the "Read Me" section of their GitHub repository). Figure 47 shows the console output following a successful reprogramming operation using *AVRDude*.

```

cast@HP-625:~/Documents$ cd ~/killerbee/killerbee/firmware/
cast@HP-625:~/killerbee/killerbee/firmware$ sudo avrdude -c jtag2 -p AT90USB1287
-U flash:w:kb-rzusbstick-001.hex

avrdude: jtagmkII_initialize(): warning: OCDEN fuse not programmed, single-byte
EEPROM updates not possible
avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.01s

avrdude: Device signature = 0x1e9782
avrdude: NOTE: "flash" memory has been specified, an erase cycle will be perform
ed
To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: jtagmkII_initialize(): warning: OCDEN fuse not programmed, single-byte
EEPROM updates not possible
avrdude: reading input file "kb-rzusbstick-001.hex"
avrdude: input file kb-rzusbstick-001.hex auto detected as Intel Hex
avrdude: writing flash (26778 bytes):

Writing | ##### | 100% 2.41s

avrdude: 26778 bytes of flash written
avrdude: verifying flash memory against kb-rzusbstick-001.hex:
avrdude: load data flash data from input file kb-rzusbstick-001.hex:
avrdude: input file kb-rzusbstick-001.hex auto detected as Intel Hex
avrdude: input file kb-rzusbstick-001.hex contains 26778 bytes
avrdude: reading on-chip flash data:

Reading | ##### | 100% 2.94s

avrdude: verifying ...
avrdude: 26778 bytes of flash verified

avrdude: safemode: Fuses OK (H:FB, E:91, L:DF)
avrdude: jtagmkII_program_disable(): bad response to leave progmode command: RSP
_FAILED

avrdude done. Thank you.

```

Figure 47: Reprogramming an Atmel RZUSB Stick with the KillerBee firmware

After reprogramming with the KillerBee firmware it is possible to check for correct operation by observing the colour of the light emitting diode that is lit upon the application of power (Figure 48).

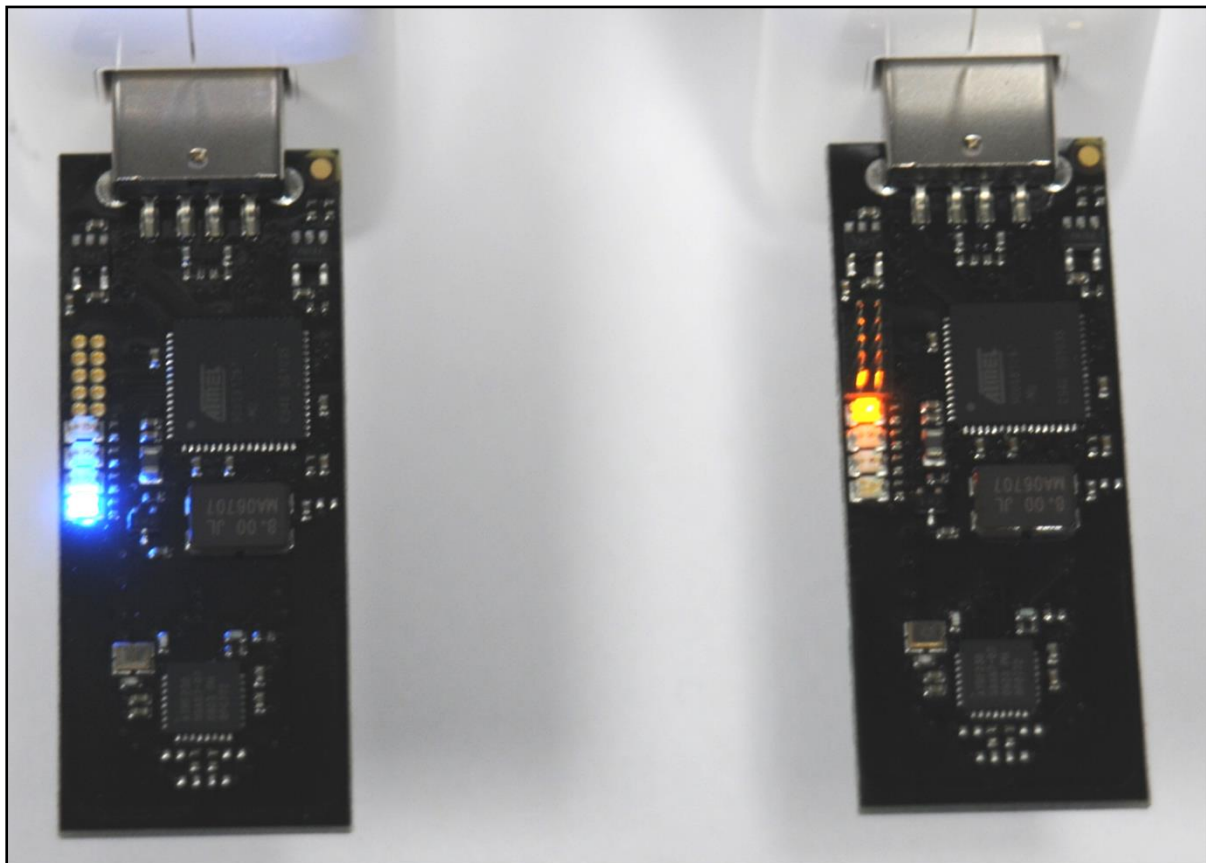


Figure 48: An off-the-shelf RZUSB Stick (left) and a modified one (right)

Also, checking the device product string when using one of the KillerBee scripts with a device identify parameter (-D) will return either RZUSBSTICK for an off-the-shelf Atmel board, or KILLERB001 for a modified board (Figure 49).

```
cast@HP-625:~$ sudo python /home/cast/Documents/Test\ runs/tpstumbler -v -w /home/cast/Documents/Test\ runs/testing.csv -D
[sudo] password for cast:
      Dev Product String      Serial Number
004:004 RZUSBSTICK           0004251CA001
cast@HP-625:~$ sudo python /home/cast/Documents/Test\ runs/tpstumbler -v -w /home/cast/Documents/Test\ runs/testing.csv -D
      Dev Product String      Serial Number
004:005 KILLERB001           0004251CA001
cast@HP-625:~$
```

Figure 49: Checking the device recognition before and after reprogramming with the KillerBee firmware

Once fully programmed with the KillerBee firmware files the board is ready for use with the KillerBee Linux scripts. In total the author purchased and modified sixteen of these boards in order to attempt simultaneous scanning on all channels (see section 5.1).

The author did also obtain a copy of the source of the modified firmware direct from Riverloop Securities; this was not made available on their GitHub repositories. Using some file comparison tools the author investigated the differences between Atmel's source and Riverloop Securities' source. There was not found to be any alterations directly related to received signal strength or otherwise useful for location and ranging. Most alterations were concerned with utilising the hardware for network penetration testing and renaming the device identifiers.

In the author's view, it would be quite feasible, given the remaining flash memory space, to further modify the firmware running on the RZUSB Stick such that the board plus a power supply could operate as a standalone investigative tool. In essence, place code equivalent to the python scripts running on Linux directly onto the RZUSB Stick and remove any requirement for a laptop. This would be an interesting route for further investigation.

Appendix 6

12.1. MBed LCP1768 source listing

The following simple code listing is that of the routine programmed to the flash memory of the MBed LCP1768 which hosted an XBee transceiver module. This code routine has a single function which continuously transmits a “Hello World” message every ten seconds (approximately).

The XBee Pro module itself is setup using the XBee XCTU software. This defines the channel, network information and stack profiles etc. In this manner a network can be set up to loosely simulate a smart meter by utilising similar stack profiles and transmitting data at similar intervals and power settings.

12.1.1. Code listing

```
#include "mbed.h" //required for all mbed builds to inform compiler about mbed core hardware
#include "C12832.h" //required for the lcd
```

```
C12832 lcd(p5, p7, p6, p8, p11); //set up the pins for the lcd
Serial xbee1(p9, p10); //creates a variable for serial communication through pin 9 and 10
```

```
//set up the pins for the tri-colour led
```

```
PwmOut r (p23);
PwmOut g (p24);
PwmOut b (p25);
```

```
DigitalOut rst1(p30); //digital reset for the xbee, 200ns for reset
DigitalOut myled(LED4); //create variable for led 3 on the mbed
```

```
int main() {
    //splash screen
    int j=0;
    lcd.cls();
    lcd.locate(0,3);
    lcd.printf("Setting up wireless comms");
```

```
    //initialise
```

```
    rst1 = 0; //set reset pin to 0
    myled = 0; //set led3 to 0
    wait_ms(1); //wait at least one millisecond
    rst1 = 1; //set reset pin to 1
    r.period(0.001); //init rgb led
    wait(1); //wait at least another millisecond, 1 second to give time for text to display
```

```
    //never ending loop
```

```
    while (1) {
        //create network traffic
        myled = 1;
        xbee1.printf("hello world\n");
        wait_ms(100);
        myled = 0;
```

```
    //display message number count to lcd to potentially account for missing data
```



```
lcd.locate(0,15);  
lcd.printf("Message ID : %d",j);  
j++;
```

//play rgb led pattern to generate a delay simulating operating another useful function

```
for(float i = 0.0; i < 1.0 ; i += 0.001) {  
    float p = 3 * i;  
    r = 1.0 - ((p < 1.0) ? 1.0 - p : (p > 2.0) ? p - 2.0 : 0.0);  
    g = 1.0 - ((p < 1.0) ? p : (p > 2.0) ? 0.0 : 2.0 - p);  
    b = 1.0 - ((p < 1.0) ? 0.0 : (p > 2.0) ? 3.0 - p : p - 1.0); ;  
    wait (0.03);  
}  
}  
}
```

Appendix 7

12.2. Adapted KillerBee python scripts

The following scripts were created by the author during the primary research phase of this research study. The code is based upon original scripts provided by Rlverloop Securities within their KillerBee frame work [98].

12.2.1. TPStumbler python script

This script is based upon the ZBStumbler script provided within the core KillerBee source. Adaptations have been made to the data recorded and layout of the CSV file and some initial message classification (such as recognising and ignoring acknowledge messages). Other minor tweaks have been made to better suit the author's testing processes.

An initial attempt was made at interfacing a USB GPS receiver in preparation for later investigations, however this was never completed and the investigations were not pursued.

12.2.1.1. Template CSV file

Column order:

message, test #, panid, source, ext_panid, stack profile, stack version, channel, msg_length, #tx, #rx, rssi, fix, lat, lon, utc,time, route

12.2.1.2. Source listing

```
#!/usr/bin/env python
```

```
'''
```

```
Using provided channel, or cycling through channels, scan for 802.15.4 packets  
log results to csv file and utilise active (default) or passive scanning.  
Active scanning involves transmitting beacon frame requests and awaiting a reponse.  
GPS lat/lon/time logging not yet fully implemented.
```

```
'''
```

```
#library imports  
import sys  
import os  
import signal  
import time  
import argparse  
import gps  
from killerbee import *
```

```

#script parameters
parser = argparse.ArgumentParser(description=__doc__)
parser.add_argument('-i', '--iface', '--dev', action='store', dest='devstring')
parser.add_argument('-g', '--gps', action='store_true')
parser.add_argument('-s', '--delay', action='store', type=float, dest='delay', default=2.0)
parser.add_argument('-v', '--verbose', action='store_true')
parser.add_argument('-c', '--channel', action='store', type=int, default=None)
parser.add_argument('-w', '--file', action='store', dest='csvfile', default=None)
parser.add_argument('-p', '--passive', action='store_true')
parser.add_argument('-D', action='store_true', dest='showdev')
args = parser.parse_args()

#802.15.4 stats
txcount = 0
rxcount = 0
stumbled = {}

if args.gps:
    try:
        #gps setup
        session = gps.gps()
        session.poll()
        session.stream()
    except Exception, e:
        print("Issue initialising GPS {0}.".format(e))
        sys.exit(-1)

#*****
def display_details(routerdata):
    global args, csvfile
    stackprofile_map = {0:"Network Specific",
                        1:"ZigBee Standard",
                        2:"ZigBee Enterprise"}
    stackver_map = {0:"ZigBee Prototype",
                    1:"ZigBee 2004",
                    2:"ZigBee 2006/2007"}
    spanid, source, extpanid, stackprofilever, channel, packet, rssi = routerdata
    stackprofile = ord(stackprofilever) & 0x0f
    stackver = (ord(stackprofilever) & 0xf0) >>4

    if args.verbose:
        print "New Network: PANID 0x%02X%02X Source 0x%02X%02X"%(ord(spanid[0]),
ord(spanid[1]), ord(source[0]), ord(source[1]))

    try:
        extpanidstr=""
        for ind in range(0,7):
            extpanidstr += "%02x:"%ord(extpanid[ind])
        extpanidstr += "%02X"%ord(extpanid[-1])
        sys.stdout.write("\tExt PANID: " + extpanidstr)
    except IndexError:
        sys.stdout.write("\tExt PANID: Unknown")

    try:
        print "\tStack Profile: %s"%stackprofile_map[stackprofile]
        stackprofilestr = stackprofile_map[stackprofile]
    except KeyError:
        print "\tStack Profile: Unknown (%d)"%stackprofile
        stackprofilestr = "Unknown (%d)"%stackprofile

    try:
        print("\tStack Version: {0}.".format(stackver_map[stackver]))

```

```

    stackverstr = stackver_map[stackprofile]
except KeyError:
    print("\tStack Version: Unknown ({0})".format(stackver))
    stackverstr = "Unknown (%d)"%stackver

if args.verbose:
    print("\tChannel: {0}".format(channel))

if args.csvfile is not None:
    #TODO test csvfile.write case for GPS args = true
    if not args.gps:

csvfile.write("***BEACON***,X,0x%02X%02X,0x%02X%02X,%s,%s,%s,%d,%d,%d,\n"%(ord(spanid[
0]), ord(spanid[1]), ord(source[0]), ord(source[1]), extpanidstr, stackprofilestr, stackverstr, channel,
len(packet), rssi))
    else:

csvfile.write("***BEACON***,X,0x%02X%02X,0x%02X%02X,%s,%s,%s,%d,%d,%d,%d,%s,%s,%s,%
s\n"%(ord(spanid[0]), ord(spanid[1]), ord(source[0]), ord(source[1]), extpanidstr, stackprofilestr,
stackverstr, channel, len(packet), rssi, fix, lat, lon, utc, time))

#*****
#TODO test addition of gps variables when args.gps = false
def response_handler(stumbled, packet, rssi, channel):#, fix, lat, lon, utc, time):
    global args
    d154 = Dot154PacketParser()
    # Chop the packet up
    pktdecode = d154.pktchop(packet)

    # Byte-swap the frame control field
    fcf = struct.unpack("<H", pktdecode[0])[0]

    if args.verbose:
        #play alert sound
        os.system('play --no-show-progress --null --channels 1 synth %s sine %f' % ( 0.3, 2000))

    # Check if this is a beacon frame
    if (fcf & DOT154_FCF_TYPE_MASK) == DOT154_FCF_TYPE_BEACON:
        if args.verbose:
            print "Received frame is a beacon."

        # The 6th element offset in the Dot154PacketParser.pktchop() method
        # contains the beacon data in its own list. Extract the Ext PAN ID.
        spanid = pktdecode[4][::-1]
        source = pktdecode[5][::-1]
        beacondata = pktdecode[6]
        extpanid = beacondata[6][::-1]
        stackprofilever = beacondata[4]

        key = ".join([spanid, source])
        value = [spanid, source, extpanid, stackprofilever, channel, packet, rssi]
        if not key in stumbled:
            if args.verbose:
                print("Beacon represents new network.")
            stumbled[key] = value
            display_details(value)
        return value

#filter acknowledgements (FCF = 0x0002)
if (len(packet) < 8) and (fcf == 2):
    if args.verbose:
        print "***ACK message***"
    if args.csvfile is not None:

```

```

        #TODO test csvfile.write case for GPS args = true
        if not args.gps:
            csvfile.write("****ACK***,X,FCF={0},,,,,,%d,%d,%d,,\n".format(pktdecode[0].encode('hex'))
            % (channel, len(packet), rssi))
        else:
            csvfile.write("****ACK***,X,FCF={0},,,,,,%d,%d,%d,%d,%s,%s,%s,%s\n".format(pktdecode[0].e
            ncode('hex')) % (channel, len(packet), rssi, fix, lat, lon, utc, time))

    #handle uncategorised responses
    else:
        if args.verbose:
            print "Unrecognised message"
            print hexdump(packet)
            print ("Packet length, %d bytes." % len(packet))
        if args.csvfile is not None:
            #TODO test csvfile.write case for GPS args = true
            if not args.gps:
                csvfile.write("****Unrecognised
            message***,X,FCF={0},,,,,,%d,%d,%d,\n%s".format(pktdecode[0].encode('hex')) % (channel,
            len(packet), rssi, hexdump(packet)))
            else:
                csvfile.write("****Unrecognised
            message***,X,FCF={0},,,,,,%d,%d,%d,%d,%s,%s,%s,%s\n%s".format(pktdecode[0].encode('hex')) %
            (channel, len(packet), rssi, fix, lat, lon, utc, time, hexdump(packet)))

    return None

#*****
def interrupt(signum, frame):
    global txcount, rxcount
    global kb
    global args, csvfile
    if args.csvfile is not None:
        csvfile.write("%s,X,***,***,***,***,***,***,***,***,%d,%d\n"%(args.csvfile, txcount, rxcount))
        csvfile.close()
    kb.close()
    if args.verbose:
        print("\n{0} packets transmitted, {1} packets received.".format(txcount, rxcount))
    sys.exit(0)

#*****
if __name__ == '__main__':

    #list usb device in use
    if args.showdev:
        show_dev()
        sys.exit(0)

    if args.verbose:
        os.system('clear')

    if args.csvfile is not None:
        try:
            csvfile = open(args.csvfile, 'w')
        except Exception as e:
            print("Issue opening CSV output file: {0}.".format(e))
            #TODO test csvfile.write case for GPS args = true
        if not args.gps:

            csvfile.write("message,test#,panid,source,extpanid,stackprofile,stackversion,channel,msglength,rssi,#
            tx,#rx,route\n")
            else:

```

```

csvfile.write("message,test#,panid,source,extpanid,stackprofile,stackversion,channel,msglength,#tx,#r
x,rxssi,fix,lat,lon,utc,time,route\n")

# Beacon frame
beacon = "\x03\x08\x00\xff\xff\xff\xff\x07"
# Immutable strings - split beacon around sequence number field
beaconp1 = beacon[0:2]
beaconp2 = beacon[3:]

try:
    kb = KillerBee(device=args.devstring)
except KBIInterfaceError as e:
    print("Interface Error: {0}".format(e))
    sys.exit(-1)

signal.signal(signal.SIGINT, interrupt)
if args.verbose:
    if not args.passive:
        print("zbstumbler: Transmitting and receiving on interface \"{0}\"".format(kb.get_dev_info()[0]))
    else:
        print("zbstumbler: Receiving on interface \"{0}\"".format(kb.get_dev_info()[0]))

# Sequence number of beacon request frame
seqnum = 0
if args.channel:
    channel = args.channel
    kb.set_channel(channel)
else:
    channel = 11

# Loop injecting and receiving packets
while 1:

    if args.gps:
        #get latest gps
        session.poll()

        #catch variable overflows
    if channel > 26:
        channel = 11

    if seqnum > 255:
        seqnum = 0

        #change channel if not user defined
    if not args.channel:
        if args.verbose:
            print("Setting channel to {0}".format(channel))
        try:
            kb.set_channel(channel)
        except Exception, e:
            print("ERROR: Failed to set channel to {0}. {1}".format(channel, e))
            sys.exit(-1)

#transmit beacon request unless passive
    if not args.passive:
        if args.verbose:
            print("Transmitting beacon request.")

    beaconinj = ".join([beaconp1, "%c" % seqnum, beaconp2])

    # Process packets for arg_delay seconds looking for the beacon response frame.

```

```

start = time.time()

try:
    txcount+=1
    kb.inject(beaconinj)
except Exception, e:
    print("ERROR: Unable to inject packet: {0}".format(e))
    sys.exit(-1)
else:
    # Process packets for arg_delay seconds looking for the beacon response frame.
    start = time.time()

    #receive packet
while (start+args.delay > time.time()):
    # Does not block
    recvpkt = kb.pnext()
    # Check for empty packet (timeout) and valid FCS
    if recvpkt != None and recvpkt['validcrc']:
        rxcount += 1
        if args.verbose:
            print ""
            print("Received frame.")
            if args.gps:
                print 'fix      ', ("NO_FIX","FIX","DGPS_FIX")[session.fix.mode - 1]
                print 'latitude ', session.fix.latitude
                print 'longitude ', session.fix.longitude
                print 'time utc ', session.utc, session.fix.time
            #TODO test networkdata case for args.gps = true
            #if not args.gps:
            networkdata = response_handler(stumbled, recvpkt[0], recvpkt[2], channel)
            #else:
            # networkdata = response_handler(stumbled, recvpkt[0], recvpkt[2], channel,
            (("NO_FIX","FIX","DGPS_FIX")[session.fix.mode - 1]), session.fix.latitude, session.fix.longitude,
            session.utc, session.fix.time)
            if args.verbose:
                print ""

        #prepare for next loop
kb.sniffer_off()
seqnum += 1
if not args.channel:
    channel += 1

```

12.2.2. TPRange python script

This script is heavily adapted from the TPStumbler script in section 12.2.1 which was in turn based upon the ZBStumbler script provided within the core KillerBee source. The script is substantially different from the original KillerBee source and serves to operate in two modes; as a transmitting or receiving interface.

When operating as a transmitter, the script causes an attached RZUSB Stick to transmit a specified number of beacon requests with a specified period.

When operated in receiving mode, the script logs the number of messages received upon a specified channel until interrupted (if no channel is defined it will scan through all the channels at a defined dwell period). Only beacon frames and minimal data are recorded to limit collateral data collection.

12.2.2.1. Template CSV file

Column order:

rxcount, message, fcf, rssi, power (dbm)

At the end of the file a total RX count is displayed to quickly enable calculation of the number of lost messages.

12.2.2.2. Source listing

```
#!/usr/bin/env python

'''
with one RZUSB transmit on provided channel a set number of times and then notify user
with another RZUSB in receive mode (-r) log transmissions with RSSI and calculated power in dBm
do not log or display transmissions from other sources, however do indicate when they are received
to ensure missed data is accounted for
'''

#library imports
import sys
import os
import signal
import time
import argparse
from killerbee import *

#script parameters
parser = argparse.ArgumentParser(description=__doc__)
parser.add_argument('-i', '--iface', '--dev', action='store', dest='devstring')
parser.add_argument('-v', '--verbose', action='store_true')
parser.add_argument('-c', '--channel', action='store', type=int, default=None)
parser.add_argument('-n', '--numbertx', action='store', type=int, default=0)
parser.add_argument('-w', '--file', action='store', dest='csvfile', default=None)
parser.add_argument('-r', '--receive', action='store_true')
parser.add_argument('-s', '--delay', action='store', type=float, dest='delay', default=2.0)
parser.add_argument('-D', action='store_true', dest='showdev')
args = parser.parse_args()

#802.15.4 stats
txcount = 0
rxcount = 0
stumbled = {}

def response_handler(stumbled, packet, rssi, channel):
    global args, power, rxcount
    d154 = Dot154PacketParser()
    # Chop the packet up
    pktdecode = d154.pktchop(packet)

    # Byte-swap the frame control field
    fcf = struct.unpack("<H", pktdecode[0])[0]

    #this formula actually needs to be placed at each message filter else simultaneous messages
    result in incorrect power levels
    #power = -90+(3*(rssi-1))

    #if args.verbose:
        #play alert sound
        #os.system('play --no-show-progress --null --channels 1 synth %s sine %f % ( 0.3, 2000))
```



```

# Check if this is a beacon frame
if (fcf & DOT154_FCF_TYPE_MASK) == DOT154_FCF_TYPE_BEACON:
    power = -90+(3*(rssi-1))
    if args.verbose:
        print "****BEACON message****"
        print ("RSSI: %d, Power: %d dbm\n" % (rssi, power))
    #rxcount -= 1
    if args.csvfile is not None:
        csvfile.write("****BEACON****\n")

#filter acknowledgements (FCF = 0x0002)
if (len(packet) < 8) and (fcf == 2):
    power = -90+(3*(rssi-1))
    if args.verbose:
        print "****ACK message****"
        print ("RSSI: %d, Power: %d dbm\n" % (rssi, power))
    #rxcount -= 1
    if args.csvfile is not None:
        csvfile.write("****ACK****\n")

#filter beacon requests (FCF = 0x0803)
if (fcf == 2051):
    power = -90+(3*(rssi-1))
    rxcount += 1
    if args.verbose:
        print "****Beacon Request****"
        print ("RSSI: %d, Power: %d dbm      (x%d)\n" % (rssi, power, rxcount))
    if args.csvfile is not None:
        csvfile.write("%d, ****Beacon Request***,0x{0:04x},%d,%d\n".format(fcf) % (rxcount, rssi,
power))

#handle uncategoryed responses
else:
    power = -90+(3*(rssi-1))
    if args.verbose:
        print "Unrecognised message"
        print ("RSSI: %d, Power: %d dbm\n" % (rssi, power))
    #rxcount -= 1
    if args.csvfile is not None:
        csvfile.write("****Unrecognised message****\n")

return None

#*****
def interrupt(signum, frame):
    global txcount, rxcount
    global kb

#show summary results
if args.receive:
    print ""
    print ""
    print "****END****"
    print ("Total received: %d" %rxcount)
    print ""
    print ""
    if args.csvfile is not None:
        csvfile.write("Total received: %d" %rxcount)

kb.close()
sys.exit(0)

```

```

#*****
if __name__ == '__main__':

    #list usb device in use
    if args.showdev:
        show_dev()
        sys.exit(0)

    if args.verbose:
        os.system('clear')

    if args.csvfile is not None:
        try:
            csvfile = open(args.csvfile, 'w')
        except Exception as e:
            print("Issue opening CSV output file: {0}.".format(e))
            csvfile.write("rxcount,message,fcf,rssi,power (dbm)\n")

    # Beacon frame
    beacon = "\x03\x08\x00\xff\xff\xff\xff\x07"
    # Immutable strings - split beacon around sequence number field
    beaconp1 = beacon[0:2]
    beaconp2 = beacon[3:]

    try:
        kb = KillerBee(device=args.devstring)
    except KBIInterfaceError as e:
        print("Interface Error: {0}".format(e))
        sys.exit(-1)

    signal.signal(signal.SIGINT, interrupt)
    if args.verbose:
        if not args.receive:
            print("zbstumbler: Transmitting on interface \'{0}\' %d times".format(kb.get_dev_info()[0])
%args.numbertx)
        else:
            print("zbstumbler: Receiving on interface \'{0}\'".format(kb.get_dev_info()[0]))

    if not args.receive:
        #create delay to allow user time to evacuate
        if args.verbose:
            print "Waiting 10 seconds for area to clear"
        time.sleep(10)

    # Sequence number of beacon request frame
    seqnum = 0
    if args.channel:
        channel = args.channel
        kb.set_channel(channel)
    else:
        channel = 11

    # Loop injecting and receiving packets
    while 1:

        #catch variable overflows
        if channel > 26:
            channel = 11

        if seqnum > 255:
            seqnum = 0

        #change channel if not user defined

```

```

if not args.channel:
    if args.verbose:
        print("Setting channel to {0}.".format(channel))
    try:
        kb.set_channel(channel)
    except Exception, e:
        print("ERROR: Failed to set channel to {0}. ({1})".format(channel, e))
        sys.exit(-1)

#transmit beacon request unless passive
if not args.receive:
    if (txcount < args.numbertx):
        txcount+=1

        if args.verbose:
            print("Transmitting beacon request (x%d)" %txcount)

        beaconinj = ".join([beaconp1, "%c" % seqnum, beaconp2])

        # Process packets for arg_delay seconds looking for the beacon response frame.
        start = time.time()

        try:
            kb.inject(beaconinj)
        except Exception, e:
            print("ERROR: Unable to inject packet: {0}".format(e))
            sys.exit(-1)
    else:
        if args.verbose:
            #play alert sound
            os.system('play --no-show-progress --null --channels 1 synth %s sine %f % ( 1, 4000))
            print ("Transmitted %d times" %args.numbertx)
            sys.exit(0)

        time.sleep(args.delay)

else:
    # Process packets for arg_delay seconds looking for the beacon response frame.
    start = time.time()

    #receive packet
    while (start+args.delay > time.time()):
        # Does not block
        recvpkt = kb.pnext()
        # Check for empty packet (timeout) and valid FCS
        if recvpkt != None and recvpkt['validcrc']:
            #rxcount += 1
            #if args.verbose:
            #    print ""
            networkdata = response_handler(stumbled, recvpkt[0], recvpkt[2], channel)

        #prepare for next loop
        kb.sniffer_off()
        seqnum += 1

```

12.3. Multi-channel automation scripts

The following script was utilised to automate the operation of multiple simultaneous TPStumbler scripts (see section 12.2.1). Unlike the previous script listings in this section, this script was created independently of the KillerBee framework.

The user lists RZUSB Stick device ID's (obtained by running any KillerBee script with a `-D` parameter). The script then starts multiple background TPStumbler processes to individually log one channel per RZUSB Stick.

12.3.1.1. Source listing

```
#!/bin/bash

#CNTRL+C interrupt closes all processes by issuing ^C (SIGINT) to appropriate process id's
onINT() {
    echo "\nKilling Process IDs: $command11, $command12, $command13, $command14,
$command15, $command16, $command17" #, $command18, $command19, $command20,
$command21, $command22, $command23, $command24, $command25, $command26"
    kill -INT $command11 $command12 $command13 $command14 $command15 $command16
$command17 # $command18 $command19 $command20 $command21 $command22 $command23
$command24 $command25 $command26
    exit
}

#set interrupt on CNTRL+C
trap "onINT" 2

#get device IDs of all attached units and show to user
sudo python /home/cast/Documents/Testrums/tpstumbler -v -D

#prompt user for device channel assignments
read -p "Enter device numbers in channel order " CHAN11 CHAN12 CHAN13 CHAN14 CHAN15
CHAN16 CHAN17 #CHAN18 CHAN19 CHAN20 CHAN21 CHAN22 CHAN23 CHAN24 CHAN25
CHAN26

#start logging in background with assigned devices
sudo python /home/cast/Documents/Testrums/tpstumbler -w
/home/cast/Documents/Testrums/multi/testing11.csv -i $CHAN11 -c 11 &
command11="$!"
echo $command11

sudo python /home/cast/Documents/Testrums/tpstumbler -w
/home/cast/Documents/Testrums/multi/testing12.csv -i $CHAN12 -c 12 &
command12="$!"
echo $command12

sudo python /home/cast/Documents/Testrums/tpstumbler -w
/home/cast/Documents/Testrums/multi/testing13.csv -i $CHAN13 -c 13 &
command13="$!"
echo $command13

sudo python /home/cast/Documents/Testrums/tpstumbler -w
/home/cast/Documents/Testrums/multi/testing14.csv -i $CHAN14 -c 14 &
command14="$!"
```

```
echo $command14
```

```
sudo python /home/cast/Documents/Testruns/tpstumbler -w  
/home/cast/Documents/Testruns/multi/testing15.csv -i $CHAN15 -c 15 &  
command15="$!"  
echo $command15
```

```
sudo python /home/cast/Documents/Testruns/tpstumbler -w  
/home/cast/Documents/Testruns/multi/testing16.csv -i $CHAN16 -c 16 &  
command16="$!"  
echo $command16
```

```
sudo python /home/cast/Documents/Testruns/tpstumbler -w  
/home/cast/Documents/Testruns/multi/testing17.csv -i $CHAN17 -c 17 &  
command17="$!"  
echo $command17
```

```
#sudo python /home/cast/Documents/Testruns/tpstumbler -w  
/home/cast/Documents/Testruns/multi/testing18.csv -i $CHAN18 -c 18 &  
command18="$!"  
echo $command18
```

```
#sudo python /home/cast/Documents/Testruns/tpstumbler -w  
/home/cast/Documents/Testruns/multi/testing19.csv -i $CHAN19 -c 19 &  
command19="$!"  
echo $command19
```

```
#sudo python /home/cast/Documents/Testruns/tpstumbler -w  
/home/cast/Documents/Testruns/multi/testing20.csv -i $CHAN20 -c 20 &  
command20="$!"  
echo $command20
```

```
#sudo python /home/cast/Documents/Testruns/tpstumbler -w  
/home/cast/Documents/Testruns/multi/testing21.csv -i $CHAN21 -c 21 &  
command21="$!"  
echo $command21
```

```
#sudo python /home/cast/Documents/Testruns/tpstumbler -w  
/home/cast/Documents/Testruns/multi/testing22.csv -i $CHAN22 -c 22 &  
command22="$!"  
echo $command22
```

```
#sudo python /home/cast/Documents/Testruns/tpstumbler -w  
/home/cast/Documents/Testruns/multi/testing23.csv -i $CHAN23 -c 23 &  
command23="$!"  
echo $command23
```

```
#sudo python /home/cast/Documents/Testruns/tpstumbler -w  
/home/cast/Documents/Testruns/multi/testing24.csv -i $CHAN24 -c 24 &  
command24="$!"  
echo $command24
```

```
#sudo python /home/cast/Documents/Testruns/tpstumbler -w  
/home/cast/Documents/Testruns/multi/testing25.csv -i $CHAN25 -c 25 &  
command25="$!"  
echo $command25
```

```
#sudo python /home/cast/Documents/Testruns/tpstumbler -w  
/home/cast/Documents/Testruns/multi/testing26.csv -i $CHAN26 -c 26 &  
command26="$!"  
echo $command26
```

```
#while true; do  
echo "press CNTRL+C to close all scans nicely"
```

```
while true; do  
read ENTRY  
done
```

```
#001:010 001:009 001:008 001:007 001:006 001:005 001:004
```