Masters Project Final Report
March 2018

Project Title	Wi-Fi Tomographic intruder detection security alarm system		
Student Name	Damian P. Muller		
Registration No. & Index No.	2015MIS016 15770166		
Supervisor's Name	Mr. Gihan Seneviratne		
Please Circle the appropriate	**Masters Program**	**Type**	
	MIS	**Research**	**Implementation**
For Office Use Only			

Wi-Fi Tomographic intruder detection security alarm system

Damian P. Muller

2018

Wi-Fi Tomographic intruder detection security alarm system

A dissertation submitted for the Degree of Master of Science in Information Security

Damian P. Muller

University of Colombo School of Computing
2018

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Students Name:

Signature: Date:

This is to certify that this thesis is based on the work of Mr. Damian P. Muller under my supervision.

The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name:

Signature: Date:

Abstract

Tomographic Motion Detection (TMD) is a technology that provides complete coverage of an area using radio waves and detects any movement with the ability to be completely hidden from view as it is not a line of sight technology, using signals that penetrate through objects. The monitored area is surrounded by nodes that communicates with each other and detects movements by the disturbance to the Receive Signal Strength (RSS) of each node measured by a processing and controlling unit.

2.4 GHz signals are heavily attenuated by anything containing water such as the human body. Hence using Wi-Fi (Wireless Fidelity) signals which operate in the 2.4 GHz band, it is possible to identify the significant disturbances in the Receive Signal Strength Indicator (RSSI) and identify human movements without carrying additional transmitting devices.

Today, protecting our valuable assets, confidential data, human life in a better way and avoiding unauthorised physical access to devices, in the field of Information Security, are demands in need. This is an eye opener for the use of TMD when compared with the limitations with commonly used Passive Infrared (PIR) motion detectors.
This research is an attempt to use non-overlapping Wi-Fi channels, simultaneously to generate better TMD for human detection within the Wi-Fi frequency range.

Initially, as the required hardware to capture the RSSI, ESP 8266, embedded Wi-Fi modules controlled by Arduino, connected to a computer via Universal Serial Bus (USB) were setup. The Wi-Fi modules can be configured for the required Wi-Fi channel, transmitting or receiving mode and capable of more configurations.
Arduino, Integrated Development Environment (IDE) which is open source and MS Comm Control a freeware control are used for developing, uploading the developed programs into the Wi-Fi module microcontrollers and reading RSSI using a computer.

Thereafter, identified the significant difference in the RSSI measured for humans, compared with other objects. For the analysis, data from RSSI readings were gathered in several tests placing objects and humans in between the Wi-Fi transmitters and receivers in a straight line obstructing the signals from the transmitters and graphs were generated.

Finally, determined how accurately a human movement can be detected, using a probability based approach. Monitored the Wi-Fi, RSSI, with statistical calculations, detected an intruder entering into the defined boundaries and triggered an alarm, constructing a security alarm.

Acknowledgements

Firstly, I would like to be very thankful to my supervisor Mr. Gihan Seneviratne, for the advice, support and technical knowledge rendered to me throughout this project work.

Also, I thank Dr. Chamath Keppetiyagama who was of a great help assisting and guiding me on the correct path to proceed with this topic of interest.

I am thankful to Dr. Kasun De Zoysa, coordinator of the Master of Information Security programme, Dr. Manjusri Wickramasinghe co-ordinator of this individual project and the thesis committee members for their ideas and recommendations.

I also thank the University of Colombo, School of Computing, its founders, late and present directors, co-ordinators, dedicated lecturers, demonstrators and all staff involved in the Master of Information Security program for giving us this opportunity to enrich our knowledge in the area of Information Security.

I would like to thank my work place, EDM Systems, the directors and the management for giving the opportunity of employment and training me on the programming languages Microsoft Visual FoxPro9, Visual Studio C# .net, automation of Microsoft applications and other techniques where I was able to practice them in programming, in this project.

Also, I thank my parents, brother and wife for motivating and encouraging me to complete this project. Not forgetting my friends and colleagues who were also with me, inspiring me to complete this degree program.

Finally, I thank you all for the help rendered to me in various ways for the completion of this project.

Table of Contents

Declaration ii
Abstract iii
Acknowledgements iv
Table of Contents v
List of Figures viii
List of Tables x
List of Abbreviations xi
Chapter 1 – Introduction 1
1.1 Overview 1
1.2 Problem domain 1
1.3 Motivation for the project 3
1.4 Research Questions 3
1.5 Aims and objectives 3
1.6 Scope and limitations 4
1.7 Methodology, testing and evaluation 4
1.8 Structure of this thesis 5
Chapter 2 – Literature Review and Methodology 6
2.1 Radio Tomography 6
2.1.1 Radio Tomographic Imaging 6
2.1.2 Tomographic Motion Detection 7
2.2 Wireless technologies 7
2.2.1 Comparison of wireless technologies in 2.4 GHz frequency 7
2.2.2 Wi-Fi frequencies and channels 9
2.3 Usage of TMD 10
2.3.1 Advantages of Wi-Fi TMD technology 10
2.3.2 Areas of TMD usage 11
2.3.3 Privacy concerns 11
2.3.4 Disadvantages of TMD usage 11
2.3.5 Research areas 11
2.3.6 Related commercial products 12
2.4 Methodology, Methods & Approaches 12
2.4.1 Hardware and software requirements 12
2.4.2 Configuration, programming and testing 15
2.4.3 Analysis of results and evaluation 16

Chapter 3 – Design & Implementation 17
3.1 Setting up and connecting devices 17
3.1.1 Connecting hardware devices 17
3.1.2 Installing the ESP8266 platform 17
3.1.3 Configuring Arduino software 18
3.1.4 Communicating with the Wi-Fi module 18
3.1.5 Reading neighbouring Wi-Fi access points 19
3.2 GUI for serial communication and generating graphs 19
3.3 Reading Wi-Fi signal strengths with channel scanning 20
3.3.1 Measuring neighbouring Wi-Fi RSSI 20
3.3.2 Observations and limitations 21
3.4 Reading Wi-Fi RSS by connecting to the network 21
3.4.1 Developing the program for the ESP8266-01 Wi-Fi module 21
3.4.2 Writing the code into the ESP8266 program memory 22
3.4.3 Developing a program for Arduino to control the Wi-Fi modules 23
3.4.4 Writing the code into the Arduino Mega program memory 23
3.4.5 Connecting the components 24
3.4.6 Testing the developed prototype 26
3.5 Determining deviations in RSS with statistical calculations 27
Chapter 4 – Results & Evaluation 29
4.1 Evaluating the system 29
4.2 Test results 30
4.2.1 Experiments with different objects 30
4.2.2 Experiments with multiple Wi-Fi channels 34
4.2.3 Experiments with Human movements 36
4.2.4 Experiments with RSSI through walls 39
Chapter 5 – Conclusion & Future Work 40
5.1 Achievements and contribution 40
5.2 Challenges faced 41
5.3 Improvements and future work 41
5.4 Lessons learnt 42
Appendix A – Hardware and software specifications 43
Appendix B – Hardware components and software 49
Appendix C – Source code listing 53
Appendix D – Data gathered 59

Appendix E – Graphs generated 62
References 67

List of Figures

Figure 1.1: Passive Infrared sensors 2

Figure 1.2: Beam sensors with different positions of transmitters and receivers 2

Figure 2.1: Wireless nodes monitoring an area to image an obstacle 6

Figure 2.2: Disturbance to RSS due to an obstacle 6

Figure 2.3: Overview of comparing wireless technologies in the 2.4 GHz band 7

Figure 2.4: Wi-Fi channels in the 2.4 GHz band 9

Figure 2.5: None overlapping Wi-Fi channels 9

Figure 2.6: ESP8266-01 Wi-Fi module with adapter 13

Figure 2.7: Pin connections of ESP8266-01 13

Figure 2.8: Block Diagram of ESP8266EX chip used by ESP8266-01 Wi-Fi module 13

Figure 2.9: View of Arduino Mega 2560 board 14

Figure 2.10: View of Arduino software IDE 15

Figure 2.11: Expected Wi-Fi RSSI when a human is detected 16

Figure 3.1: Pin connections of the ESP8266-01 Wi-Fi module 17

Figure 3.2: Installing the ESP8266 platform 18

Figure 3.3: GUI in development environment 20

Figure 3.4: Developed GUI monitoring the serial communication capturing RSS 20

Figure 3.5: Main function of the program written into the Wi-Fi program memory 22

Figure 3.6: Source code extract from the program written into Arduino memory 23

Figure 3.7: Diagram of Wi-Fi modules for simultaneous transmission and RSSI 24

Figure 3.8: Hardware prototype to capture Wi-Fi RSSI on non-overlapping channels 26

Figure 3.9: Testing the developed prototype 26

Figure 3.10: Formula of Standard Deviation used for calculations 27

Figure 3.11: Distribution of the probability in the Bell Curve 28

Figure 4.1: View of placing built hardware prototypes to monitor movement 29

Figure 4.2: No disturbance in RSSI 31

Figure 4.3: Cardboard sheet moved across the Wi-Fi signal path 31

Figure 4.4: Glass of water moved across the Wi-Fi signal path 31

Figure 4.5: Damp towel moved across the Wi-Fi signal path 32

Figure 4.6: Metal tray moved across the Wi-Fi signal path 32

Figure 4.7: Wooden board moved across the Wi-Fi signal path 32

Figure 4.8: Wooden log moved across the Wi-Fi signal path 33

Figure 4.9: Leather bag moved across the Wi-Fi signal path 33

Figure 4.10: Plastic cover moved across the Wi-Fi signal path 33

Figure 4.11: Human moved across the Wi-Fi signal path 34
Figure 4.12: Multi channel RSSI without disturbance 35
Figure 4.13: Multi channel RSSI deviation with human in the transmission path 35
Figure 4.14: Human movement across the transmitter and the receiver 37
Figure 4.15: Human movement across the transmitter and the receiver, diagonally 37
Figure 4.16: Human movement from the transmitter towards the receiver 38
Figure 4.17: Human movement from the receiver towards the transmitter 38
Figure 4.18: RSSI through a wall without disturbance 39
Figure 4.19: RSSI through a wall with human crossing the signal path 39
Figure A.1: Pin connections of the Arduino Mega 2560 board 45
Figure A.2: Important and functions of the ESP8266-01 Wi-Fi module 48
Figure B.1: ESP8266-07 Wi-Fi transceiver module 49
Figure B.2: Connecting the Wi-Fi module and Arduino board 49
Figure B.3: Prototype built using ESP8266-07 and Arduino Mega 49
Figure B.4: Diagram to program the Wi-Fi module with Arduino 50
Figure B.5: Hardware developed to upload a program into Wi-Fi module 50
Figure C.1: Source code reading text from serial communication 53
Figure C.2: Source code for generating graphs 53
Figure C.3: Source code extracts form the developed GUI to update the graph 55
Figure C.4: Source code extract to set the Wi-Fi mode 56
Figure C.5: Source code extract to connect to an AP 56
Figure D.1: Sample raw RSSI data gathered with channel scanning 59
Figure E.1: No disturbance between the Wi-Fi transmitter and the receiver 62
Figure E.2: Disturbance due to human between the Wi-Fi transmitter and receiver 62
Figure E.3: Reading RSSI without disturbance from center of a defined boundary 63
Figure E.4: Reading RSSI without disturbance from a corner in a defined boundary 63
Figure E.5: Transmitter, receiver and human, placed at 90 degrees 64
Figure E.6: Transmitter, receiver and human, placed 45 degrees 64
Figure E.7: Transmitter, receiver and human, placed at about 30 degrees 65
Figure E.8: Transmitter, receiver and human, placed at about 15 degrees 65
Figure E.9: Transmitter, receiver and human, placed at about 22 degrees 66
Figure E.10: Transmitter, receiver and human, placed at 90 degrees 66

List of Tables

Table 2.1: Comparison of commonly used wireless technologies in 2.4 GHz band ... 8
Table 2.2: Wi-Fi channel frequencies ... 10
Table 3.1: Pin connections of the ESP8266-01 Wi-Fi module ... 17
Table A.1: Specifications of ESP8266EX chip used in ESP8266-01 Wi-Fi module ... 43
Table A.2: Specifications of the Arduino Mega 2560 board ... 44
Table A.3: Basic AT Commands ... 46
Table A.4: Wi-Fi AT Commands ... 47
Table A.5: TCP/IP-Related AT Commands ... 48
Table D.1: Sample Wi-Fi RSSI data gathered ... 60
Table D.2: Processed Wi-Fi RSSI data for graphs ... 61
Table D.3: Statistical calculations on Wi-Fi RSSI data gathered ... 61

List of Abbreviations

3G	-	Third generation of wireless mobile telecommunication technology
ADC	-	Analog to Digital Conversion
AES	-	Advanced Encryption Standard
AP	-	Access Point
API	-	Application Program Interface
(A)RC4	-	(Alleged) Rivest Cipher 4 also known as ARCFOUR
(A/GF)SK	-	(Amplitude/Gaussian Frequency) Shift Keying
(B/Q)PSK	-	(Binary/Quadrature) Phase Shift Keying
C/C++	-	General purpose high-level/object oriented programming language
CBC-MAC	-	Cipher Block Chaining Message Authentication Code
CCK	-	Complementary Code Keying
CCM	-	Counter with CBC-MAC
CE	-	European Commission
CH-PD	-	Chip Power Down
COFDM	-	Coded Orthogonal Frequency Division Multiplexing
CPU	-	Central Processing Unit
CR	-	Carriage Return
CRC	-	Cyclic Redundancy Check
CSAIL	-	Computer Science and Artificial Intelligence Laboratory, USA
CT	-	Computed Tomography
CTR	-	Counter
(C/R)TS	-	(Clear/Request) To Send
dBm	-	decibel-milliwatts
E0	-	Stream cipher used in the Bluetooth protocol
EEPROM	-	Electrically Erasable Programmable Read-Only Memory
EN	-	Enable
ESP8266(EX)	-	Wi-Fi module from Espressif Systems, Shanghai, Chinese manufacturer (microcontroller)
FCC	-	Federal Communications Commission
FTP	-	File Transfer Protocol
g	-	gram
GND	-	Ground
(GP/SD)IO	-	(Secure Digital/General Purpose) Input Output
GSM	-	Global System for Mobile communication

HTTP	-	HyperText Transfer Protocol
I2C	-	Inter-Integrated Circuit
I2S	-	Inter-IC Sound
ICSP	-	In-Circuit Serial Programming
IDE	-	Integrated Development Environment
IEEE	-	Institute of Electrical and Electronics Engineers
IOT	-	Internet Of Things
IPEX	-	miniature RF connector for high-frequency signals
IPv4	-	Internet Protocol Version 4
IR	-	Infrared Radiation
KB	-	KiloByte
(k/M)bps	-	(Kilo/Mega) bits per second
LAN	-	Local Area Network
LED	-	Light Emitting Diode
LLC	-	Limited Liability Company
LOS	-	Line Of Sight
m	-	meter
mA	-	milli Ampere
MAC	-	Media Access Control
(M/G)Hz	-	(Mega/Giga) Hertz
MIT	-	Massachusetts Institute of Technology, USA
mm	-	millimeter
MOSI	-	Master Out Slave In
(M)QAM	-	(Multiple) Quadrature Amplitude Modulation
ms	-	miliseconds
MS Comm.	-	Microsoft Communication Control
NFC	-	Near Field Communication
NL	-	New Line
OCX	-	OLE Custom control
OLE	-	Object Linking and Embedding
OTA	-	Over The Air
PC	-	Personal Computer
PCB	-	Printed Circuit Board
PIR	-	Passive Infrared
PWM	-	Pulse Width Modulation

RF	-	Radio Frequency
RSS(I)	-	Receive Signal Strength (Indicator)
RST	-	Reset
RTI	-	Radio Tomographic Imaging
(RX/TX)(D)	-	(Receive/Transmit) (Data)
S(CL/DA)	-	Serial (Clock/Data)
SDK	-	Software Development Kit
SIG	-	Special Interest Group
SOC	-	System On a Chip
SPI	-	Serial Peripheral Interface
SRAM	-	Static Random Access Memory
SRRC	-	State Radio Regulatory Commission
SSID	-	Service Set Identifier
TCP	-	Transmission Control Protocol
TELEC	-	Telecom Engineering Center
TKIP	-	Temporal Key Integrity Protocol
TMD	-	Tomographic Motion Detection
UART	-	Universal Asynchronous Receiver-Transmitter
UDP	-	User Datagram Protocol
URL	-	Uniform Resource Locator
USB	-	Universal Serial Bus
V	-	Volts
Vcc	-	Voltage Common Collector
VFP	-	Visual FoxPro
WEP	-	Wired Equivalent Privacy
Wi-Fi	-	Wireless Fidelity
WiMax	-	Worldwide Interoperability for Microwave Access
WPA2_PSK	-	Wi-Fi Protected Access 2 - Pre Shared Key

Chapter 1
Introduction

1.1 Overview

Today, wireless technologies have become fast in communication and power efficient, thereby the usage of wireless communications are more widespread than ever before. Mainly the impractical usage, area of use and limitations of wired solutions also contribute to this improvement.

Embedded wireless systems which consist of an embedded processer or microcontroller with a wireless interface for communicating is increasing in its processing power, decreasing in cost, decreasing in size and being more portable.

The development of efficient and flexible wireless technology has created a wide variety of new wireless systems resulting from Bluetooth headsets to industrial process automations and still motivates, urges for new trends in wireless solutions for controlling and connecting to differentiate and improve products.

The goal of this research is an attempt, to use embedded Wi-Fi (Wireless Fidelity) modules, communicate among each other simultaneously in non-overlapping channels, measure their signal strengths, detect if an intruder has entered into the defined boundaries and trigger an alarm. This is also an attempt to identify movements of human without carrying any device and without invading their privacy unlike with camera systems.

1.2 Problem domain

Today, protecting assets, confidential data and human life is a great concern. Within our homes, in high security areas and in industries it is a need in demand. To minimise the threat, various alarm systems are developed and are in use which uses technologies such as Passive Infrared (PIR), microwave, ultrasound detectors for motion detection and laser break beams, infrared break beams for boundary crossing detection [4]. PIR are far most common type of detectors which work by detecting changes in infrared emitted by warm bodies. They are cheap and quite reliable [1].

Passive infrared sensors [2] and beam sensors with different positions [3] are shown below in Figure 1.1 and Figure 1.2.

These sensors have their own limitations as listed below and is challenging to identify different approaches to minimise or overcome these limitations.

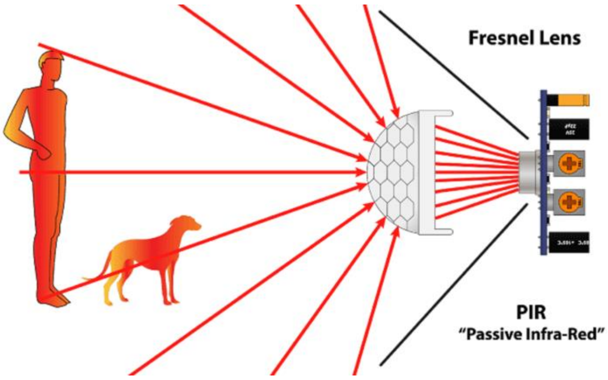


Figure 1.1: Passive Infrared sensors

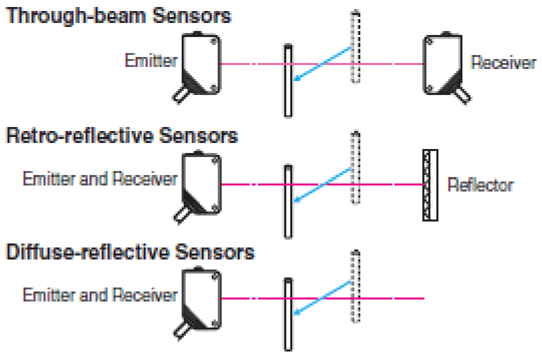


Figure 1.2: Beam sensors with different positions of transmitters and receivers

Some reasons for the need of a new technology are as follows.

- All these sensors cannot see through objects such as walls or other solids. Especially PIR detectors can be tricked by covering them cautiously [1].
- Many detectors and sensors are needed to monitor an area can be complexed or cluttered. Especially when the area is separated with walls, furniture or other obstacles or continually changing spaces needs more sensors to adequately monitor [1].
- Since PIR detectors function on infrared radiation emitted by warm bodies, in high temperature environments they may malfunction.
- These sensors may trigger false alarms due to insects, pets and natural movements such as wind, etc...

In terms of information security physical access to resources are critical. The goals of information security confidentiality, integrity and availability can be destroyed with physical access to unauthorized resources. Although, hardware resources such as servers, routers, data storages are well secured by its network, protection against malware, viruses, worms, spam mails, phishing attacks and are encrypted, an unauthorized user gaining physical access, could steal valuable information, alter information or at worst destroy the systems or resources resulting great loss and bad reputation.

This research is an attempt to identify unauthorized physical access to valuable resources by monitoring a predefined area for human movement and alert authorities. To evaluate and test, a prototype is built using electronic components which communicates using Wi-Fi channels, are programmed to monitor an area, gather data, compute, analyze and present results in graphs.

1.3 Motivation for the project

The below factors have contributed for the research and implementation of this project.

- Security of assets and human life is a great concern. An intruder may enter at any time and take advantage of a slight negligence such as not closing of a window or enter unknowingly and steal valuables or even threaten the residents.
- By following post graduate studies in Information Security and its contents in Wi-Fi security, mobile security, cryptography and Internet Of Things (IOT), it was enthusiastic to research in such a project where it can be practically applied.
- Also when compared with available commercial products they are either in research state, they communicate via Bluetooth [1], [9] which requires more nodes to monitor an area or is highly expensive [6].

1.4 Research Questions

By targeting the problem to be solved the following research areas are addressed.

- Can RSSI be used to detect an object or human placed in between a Wi-Fi transmitter (access point) and a receiver (client) in a straight line?
- Can non overlapping Wi-Fi channels be used, for simultaneous transmission, monitor the RSSI of each for better results and improve accuracy of detecting human presence?

1.5 Aims and objectives

To achieve the goal and research questions of this project, the research and implementation are mainly carried out in the following phases.

- Establish simple communication between a Wi-Fi module and a computer, read surrounding Wi-Fi access points and their Receive Signal Strength Indicator (RSSI) and present the captured information in a graph.
- Using all possible non-overlapping Wi-Fi channels for simultaneous transmitting and receiving, and setting up Wi-Fi transmitters on each channel selected. Again capture the RSSI and identify patterns in the signal drops when an object, human is placed in line between the transmitters and the receivers.

  As the RSSI is measured on non-overlapping channels simultaneously, the accuracy is higher than monitoring the RSSI of a single channel.
- Monitor the Wi-Fi RSSI, perform statistical calculations and detect an intruder entering into the area, by constructing a security alarm system.

1.6 Scope and limitations

The following limitations may cause malfunctions and may not produce expected results.

- These Wi-Fi modules ESP 8266, for the research are purchased locally or imported and are fairly cheap. They may malfunction or not provide proper results depending on the temperature and humidity. Therefore resulting graphs may not be very accurate.
- The Wi-Fi modules and communication among them may be slow and cause a delay, resulting undetected slow or fast movements within the Wi-Fi monitored range.
- It may be possible to distort Wi-Fi signals with attacks such as Evil twin, etc… although its encryption method is set as WPA2_PSK and cause false alarms.
- Also, external equipment operating in the Wi-Fi frequency range such as microwave ovens may interfere with the transmitting Wi-Fi signals. Therefore a Wi-Fi transceiver (node) may not function as expected and triggers a false alarm.

1.7 Methodology, testing and evaluation

From the research and implementation, the following are gathered and presented.

- Data gathered from Wi-Fi modules, based on the RSSI, presented in charts describing the variations of the signal strengths in the following situations,
    - Preliminary tests done when objects and human were placed within a straight line between the transmitter and receiver.
    - Transmitting and receiving signal strengths from Wi-Fi modules captured from all non-overlapping channels simultaneously, to generate better Radio Tomographic Imaging (RTI), for human detection within the Wi-Fi frequency range.
- A working prototype is developed that consists of the following,
    - Wi-Fi modules are placed in a vertical line for each non overlapping channel which are of 3 Wi-Fi modules each.
    - These Wi-Fi modules are controlled by microcontrollers, which monitor each other Wi-Fi module’s signal strengths, detect human movement within the coverage area and trigger an alarm.
- The source code and programs developed for the Wi-Fi module microcontrollers, the data so captured and the thesis are presented.

1.8 Structure of this thesis

Five main chapters are described in detail. Brief introductions on each are stated below.

Chapter 1: Introduction - Includes the main goal of the research and an overview of other sections including the problem domain, motivation, research questions, aims and objectives, scope and limitations, testing and evaluation.

Chapter 2: Literature Review - Describes the study of the project and research areas. Also includes similar technologies implemented.

Chapter 3: Design & Implementation - Implementation and development of the project including its hardware, software and source code with required tools and techniques used.

Chapter 4: Results & Evaluation - Captured results are analysed with the required calculations and formulas.

Chapter 5: Conclusion & Future Work - Emphasizes on knowledge gathered and lessons learnt. Also includes improvements and future developments for the project.

Other sections of the thesis - Once the main chapters are complete the Appendixes include the documents referred, test results and the source code.

Finally, the Reference section lists all the external material referred in this thesis.

Chapter 2
Literature Review and Methodology

2.1 Radio Tomography

Radio tomography is a transmission based imaging method which measures received signals on many different paths through a medium, using radio frequencies.

Radio tomography operates in two well-known and widely used types of imaging systems [7].

- Radar systems transmit RF signals and receive echoes caused by the objects in an environment. A delay between transmission and reception indicates a distance to the object.
- Computed Tomography (CT) methods in medical and geophysical imaging systems send signals along many different paths through a medium and measure the magnitude and phase of the transmitted signal. The measurements on many paths are used to compute an estimate the spatial field of the transmission parameters throughout the medium.

2.1.1 Radio Tomographic Imaging (RTI)

Tomography is an attempt which aims to obtain cross-sectional images of an object, originally used for medical analysis. An industrial Tomography System is a computerized system that can provide cross sectional images of an investigated object for industrial purposes.

RTI is an attempt which offers a new way to image objects in buildings and outdoor environments by transmitting signals and using Received Signal Strengths (RSS) [8]. [7].

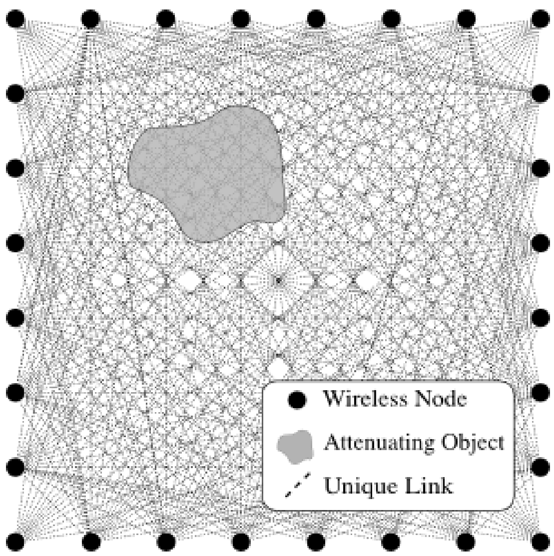


Figure 2.1: Wireless nodes monitoring an area to image an obstacle

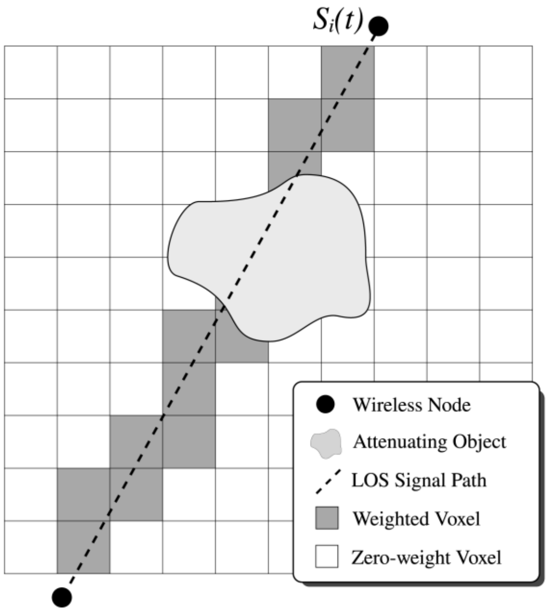


Figure 2.2: Disturbance to RSS due to an obstacle

2.1.2 Tomographic Motion Detection (TMD)

This is a new form of motion detection technology where it provides complete coverage of an area with the ability to be completely hidden from view as it is not a line of sight technology [5]. It works by surrounding an area with nodes that communicate with each other via radio waves creating a mesh network. It relies on disturbances of the mesh network to detect movement within an area. Each node may have a processing unit that controls the communication with each other and has the ability to join the network easily.

Each communication line between two nodes represents a radio path. The system mainly uses 2.4 GHz signals, same as with Wi-Fi or Bluetooth as the signals are heavily attenuated by anything containing water such as the human body. A human body placed in the radio path of any two nodes will noticeably reduce the received signal strength [1].

By carefully measuring the RSS from each node to each other and doing some clever processing, it is possible to build up an image of what the area looks like. Any significant disturbance could be identified and the motion of a human can be detected.

These signals penetrate through objects such as walls, furniture etc.., and as long as the signal strength is not attenuated too much, such signals can be detected by the receiver [1].

2.2 Wireless technologies

As the main objective of this project is to use wireless technology to identify human movements and as researches identify 2.4 GHz signals strengths are highly diminished when travelling through substances containing water [1], following is an analysis validating the suitability of Wi-Fi technology for this implementation from the commonly available technologies.

2.2.1 Comparison of wireless technologies in 2.4 GHz frequency

Below comparisons identifies the best transmission method for this research [10], [11], [12].

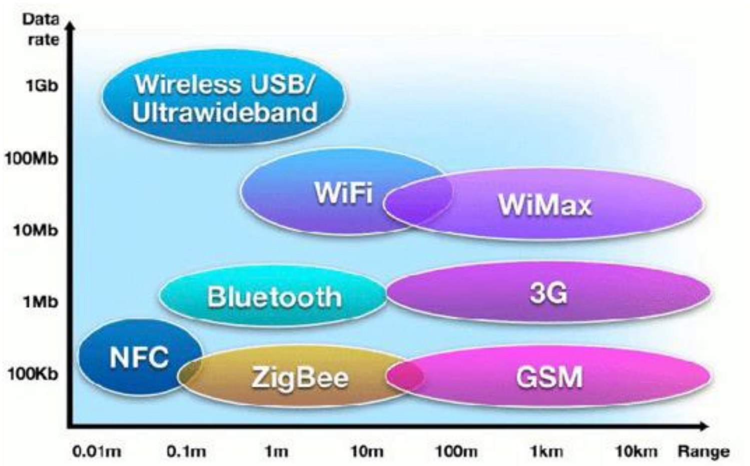


Figure 2.3: Overview of comparing wireless technologies in the 2.4 GHz band

Comparison attribute	ZigBee	Bluetooth	Wi-Fi
Wireless standard	IEEE 802.15.4	IEEE 802.15.1	IEEE 802.11
Year of development	1998	1994	1991
Organization	ZigBee Alliance	Bluetooth SIG	Wi-Fi Alliance
Networking topology	Mesh, Star, Tree, Ad-hoc, Point to point	Star, Ad-hoc, very small networks	Star, Point to point
Operating frequency	868 MHz in Europe 915 MHz in USA and Australia 2.4 GHz worldwide	2.4 GHz	2.4 GHz, 5 GHz
Channel bandwidth	0.3, 0.6, 2 MHz	1 MHz	22 MHz
Number of transmission channels	1, 10, 16	79	14 (2.4 GHz)
Data transfer rate	20, 40, and 250 kbps	1 Mbps	11, 54 Mbps
Nominal transmission power	(-25) - 0 dBm	0 - 10 dBm	15 - 20 dBm
Transmission range	10-100 m	10 m	50-100 m
Power Consumption - Battery life	Very low, low power is a design goal	Medium	High
Complexity Device, application impact	Low	High	High
Modulation type	BPSK (+ASK), O-QPSK	GFSK	BPSK, QPSK COFDM, CCK, M-QAM
Coexistence mechanism	Dynamic frequency selection	Adaptive frequency hopping	Dynamic frequency selection, Transmit power control (802.11h)
Encryption	AES block cipher (CTR, counter mode)	E0 stream cipher	RC4 stream cipher(WEP), AES block cipher
Authentication	CBC-MAC (ext. of CCM)	Shared secret	WPA2 (802.11i)
Data protection	16-bit CRC	16-bit CRC	32-bit CRC
Typical Applications	Industrial control and monitoring, Sensor networks, Building automation, Home control and automation	Wireless connectivity between devices such as phones, PDA, laptops, headsets	Wireless LAN connectivity, Broadband Internet access
Other Information Time to join an existing network	Device connection is under 30ms	Device connection requires up to 10s	Device connection requires 3-5s
Cost	Medium	Low	High

Table 2.1: Comparison of commonly used wireless technologies in 2.4 GHz band

According to the above it is ideal to use Wi-Fi for this research as it has a high transmission range and a high transmission data rate. Also Wi-Fi modules of different capacities and small sizes are freely available for purchase.

2.2.2 Wi-Fi frequencies and channels

Wi-Fi mainly operates in 2.4 GHz and 5 GHz frequency ranges [13] and for this research, 2.4 GHz band is used which is the desired frequency for the implementation.

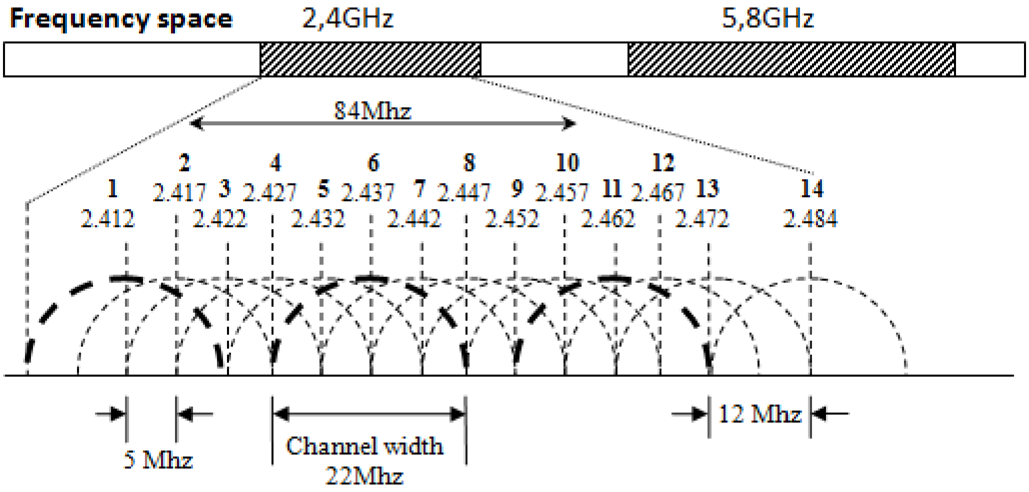


Figure 2.4: Wi-Fi channels in the 2.4 GHz band

As per Wi-Fi frequency design, 14 channels exists in the 2.4 GHz frequency band from which a maximum of three non over lapping channels could be selected for simultaneous transmission without distortion [13] of the transmitted data for better TMD.

Figure 2.4, 2.5 and Table 2.2 illustrates this.

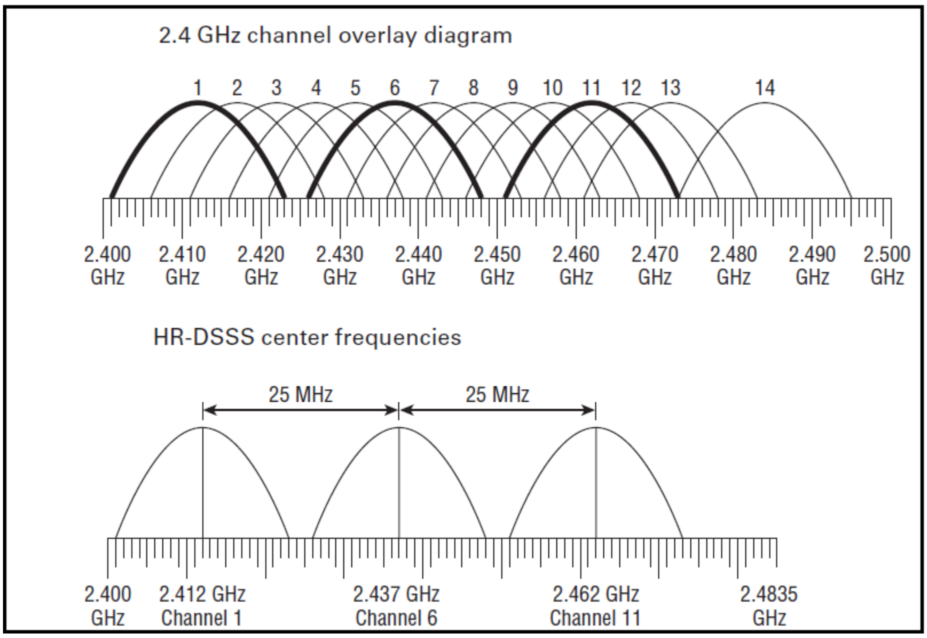


Figure 2.5: None overlapping Wi-Fi channels

Channel Number	Lower Frequency MHz	Center Frequency MHz	Upper Frequency MHz
1	2401	2412	2423
2	2406	2417	2428
3	2411	2422	2433
4	2416	2427	2438
5	2421	2432	2443
6	2426	2437	2448
7	2431	2442	2453
8	2436	2447	2458
9	2441	2452	2463
10	2446	2457	2468
11	2451	2462	2473
12	2456	2467	2478
13	2461	2472	2483
14	2473	2484	2495

Table 2.2: Wi-Fi channel frequencies

2.3 Usage of TMD

2.3.1 Advantages of Wi-Fi TMD technology

The following are a few advantages of Wi-Fi TMD compared to others sensors and detectors [5].

- Wi-Fi TMD uses radio waves in the 2.4 GHz band, which most wireless networks use today and travels through obstacles and even through concrete walls. The only exception is metal, which the radio waves will travel around it.
- Wi-Fi TMD does not rely on line of site so it cannot be blocked by objects moved in front of the nodes in the network and can be completely hidden from view.
- Wi-Fi TMD is immune to the leading causes of false alarms. Insects will not be detected, even if they crawl on the nodes. Birds do not have enough body mass to trigger the motion detection. However, it may detect large pets such as big dogs.
- Dirt built up is not an issue. When the detectors are properly placed within a case and it is completely covered in dirt, still it will function.
- Air conditioning, heating and other thermal energy has no effect on the system.
- Area of coverage is larger than a normal surveillance system.
- No need for different types of sensors, detectors such as door, motion detection and break beams, etc… and many of them.
- Cheaper and less equipment used when compared with a surveillance system.
- Can be used indoors, outdoors, and day or night is immaterial.

2.3.2 Areas of TMD usage

Environments of Tomographic Motion Detection can be listed as follows [5].

- Aesthetic-sensitive environments.
  Many sensors installed for security in homes and places looks indecent.
- High value assets.
  Security to protect banks, industries, companies and civilians that need reliable security to protect their assets namely cash, jewellery, stores, etc…
- Dirty and cluttered environments.
  Security in dirty and dynamic environments like warehouses, machine shops, etc., where traditional sensors are blocked either by equipment or dirt.

2.3.3 Privacy concerns

TMD can be used for disabled persons, handicapped, patients and elderly care, to monitor their movements or falls and alert care givers without invading their privacy unlike security cameras, etc…

2.3.4 Disadvantages of TMD usage

The main drawbacks of Tomographic Motion Detection are as follows.

- Each node needs to be powered and more power is required, as area covered is larger.
- Disturbances to the signals due to sabotage or other equipment such as microwave ovens that operate in the same frequency may result in false detections.

2.3.5 Research areas

Wi-Fi signals, its signal strengths are measured and researches are carried out in many areas, out of which the following are more adjacent to this research and are listed as follows.

- MIT's Computer Science and Artificial Intelligence Lab(CSAIL) have developed a Wi-Fi technology that is capable of seeing people through walls and other obstacles. This is further developed to capture movements across a house and also helps firefighters to determine if there are living people in a burning building [14].
- Computer scientists of the University of Washington have developed gesture-recognition technology that leverage Wi-Fi signals around to detect specific movements without needing sensors on the human body or cameras [15].
  Using an adapted Wi-Fi router and few wireless devices in the living room, it is possible to control electronic devices and household appliances from any room in the home with a simple gesture.

2.3.6 Related commercial products

- Xandem Technology LLC, a USA based company has invented a detector that uses 2.4 GHz radio signals to detect motion in a space and has introduced the product to the market [16].

  This Tomographic Motion Detection system is an individual unit and the number of units to be installed can be decided by the user. It is capable of monitoring a space from 50 to 500 square metres, not in a line of sight and can be discretely installed behind walls. This product uses a microcontroller CC2540 RF SOC from Texas Instruments, a cost-effective, low-power, true system-on-chip for Bluetooth low energy applications [9].
- Xandem Technology has also obtained a patent for the concept, research, methods on “Systems and method of device-free motion detection and presence detection” [17].
- Cognitive Systems, has created, a device Aura, that detects intruders using Wi-Fi, but still at research stage [38].

  It needs two devices placed in opposite directions to function. It can detect, indicate a movement via a smartphone app and the amount of movement within the monitored area. Aura can automatically notify a person leaving the area, who has registered his/her phone with the device, by detecting the presence of the phone. Thereafter it could automatically arm itself, although it can be done manually. When a movement is detected, Aura sends an alert to the registered phone or sets off the alarm.

2.4 Methodology, Methods & Approaches

This research focuses on analysing patterns in the Wi-Fi, RSS and identifies a human movement crossing the straight line between the Wi-Fi transmitter and receiver using simultaneous transmission of non-overlapping Wi-Fi channels. Data is gathered using hardware modules, analysed and a security alarm system is built.

2.4.1 Hardware and software requirements

To start, it is required to establish simple communication between a Wi-Fi module and a computer and to read the surrounding Wi-Fi access points and their signal strengths.

- This could be done with ESP8266-01 which is a commonly used basic Wi-Fi module that has serial interfacing with a desktop computer, microcontroller or laptop with USB to virtual Serial interface [31].

  An adapter is required to power the module with 3.3V, as the normal power source is 5V which is used for other logical operations.

A view of ESP8266-01 module and its adapter are shown below in Figure 2.6, Figure 2.7 and more details in Figure A.2.

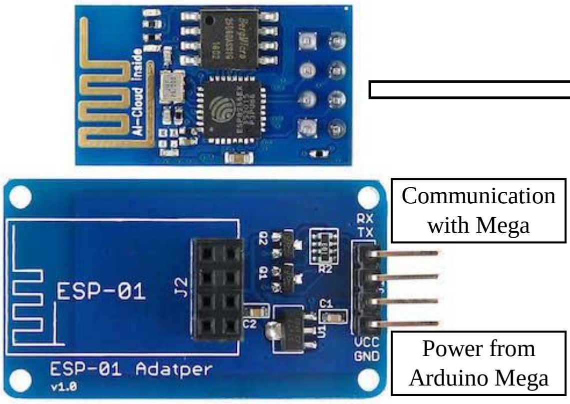


Figure 2.6: ESP8266-01 Wi-Fi module with adapter

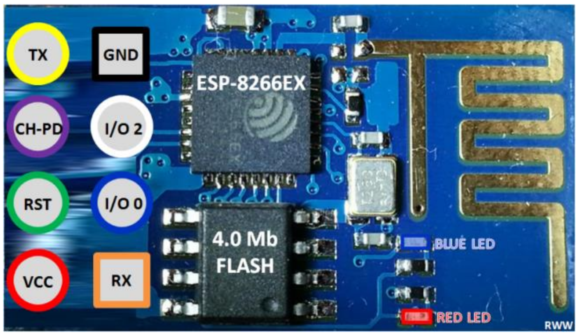


Figure 2.7: Pin connections of ESP8266-01

Show below in Figure 2.8 is the block diagram architecture of the ESP8266EX the chip used by the ESP8266-01 Wi-Fi module [18], [45]. Also specifications from the technical documentation [19], [20] of ESP8266EX are available in Appendix A. ESP8266-07 also was used for some tests which is similar to ESP8266-01 and its adaptor, together. ESP8266-07 is shown in Figure B.1.

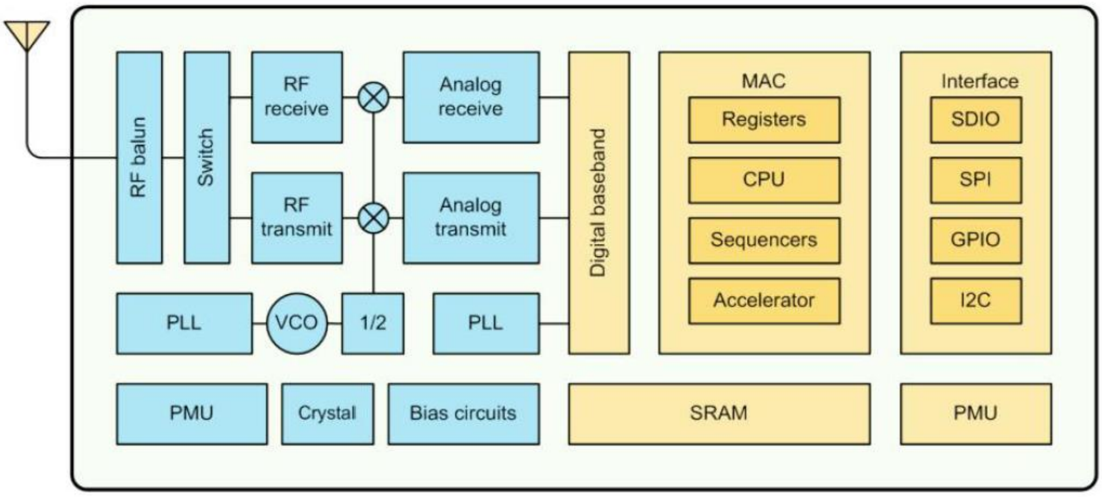


Figure 2.8: Block Diagram of ESP8266EX chip used by ESP8266-01 Wi-Fi module

- The ESP8266-01 operates with a predefined set of commands from its firmware, which need to be studied [23].

  Also, rather than using the default predefined set of commands, it is possible to save a program into the ESP8266-01 chip and enable it to operate to capture the RSSI when an event triggers based on the device input pin.

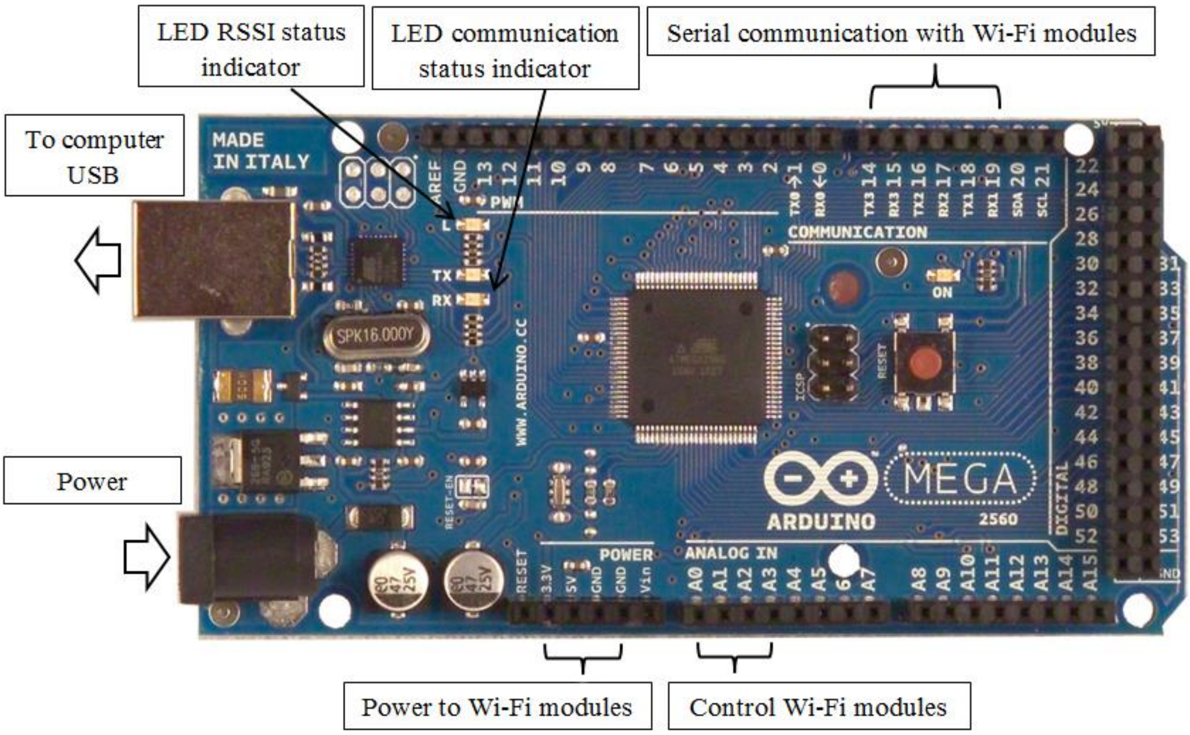


Figure 2.9: View of Arduino Mega 2560 board

- The use of Arduino Mega 2560 development board (version 3) and the relevant Arduino software (version 1.8.2) need to be installed in the computer or laptop.
  An Arduino Mega board is used as it is designed with three serial communication ports which could support three ESP8266-01 modules which are intended for this research to simultaneously communicate on three Wi-Fi channels. A view of Arduino Mega 2560 is shown in Figure 2.9 and specifications, technical documentation [24] are available in Appendix A.
  Arduino Uno similar to Arduino Mega but with less capacity was used for certain tests and writing the program into the Wi-Fi module. Figure B.5 shows the hardware built to upload a program into the Wi-Fi module.

- Arduino software is open source, uses a C/C++ compiler and is maintained by the Arduino Community worldwide [25].
  The Arduino IDE consists of many supportive features such as Arduino board selection, program verification, examples and online forum etc…

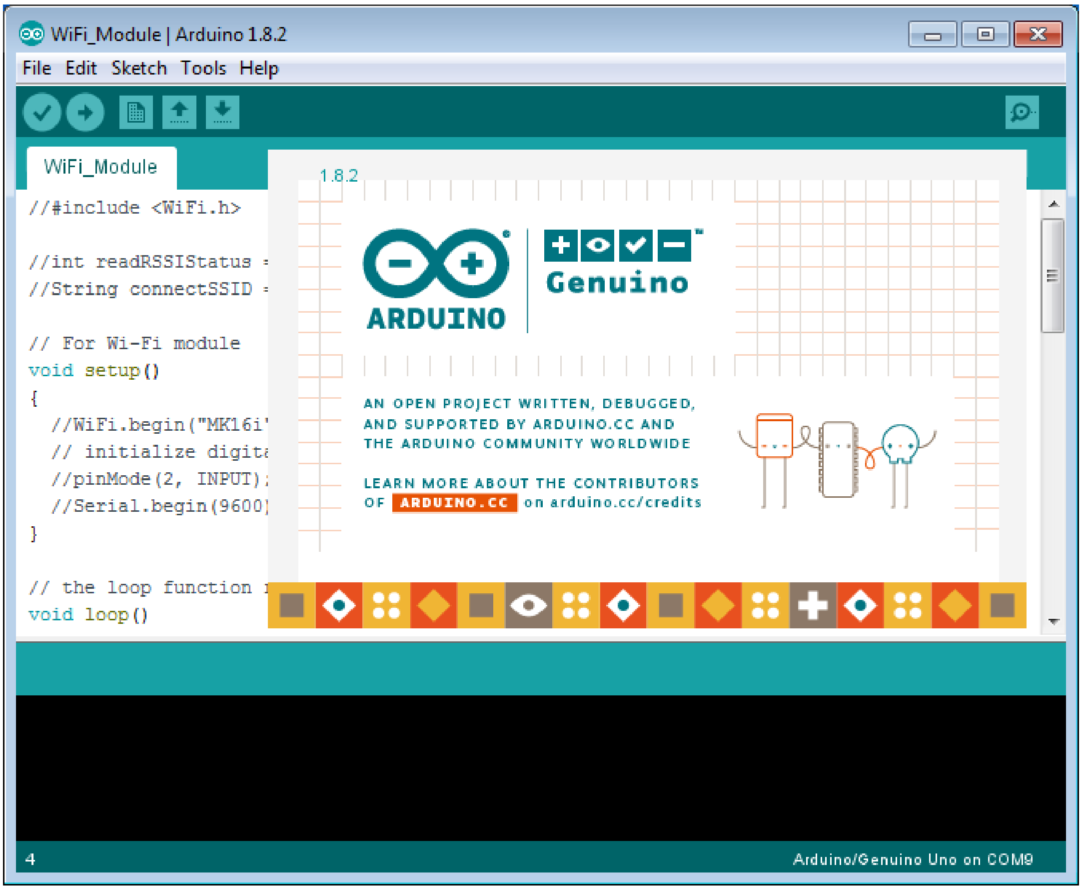


Figure 2.10: View of Arduino software IDE

2.4.2 Configuration, programming and testing

- As serial communication is used to receive data from the Arduino board to the computer, in a Windows environment it is required to use an Application Program Interface (API) or Object Linking and Embedding (OLE, OCX) control [26].

  To generate graphs on the RSSI, it captures data from the Wi-Fi module via serial communication and a developed program automatically gathers required information in suitable intervals.

  As this research is being done in a Windows environment, Visual FoxPro 9 is used as the programming language to generate required graphs.

  Also, the developed programs are saved into the ESP8266-01 Wi-Fi module and the Arduino Mega 2560 board microcontrollers.

  The expected graph when a human is detected with the RSSI vs. Time is shown in Figure 2.11 [27].

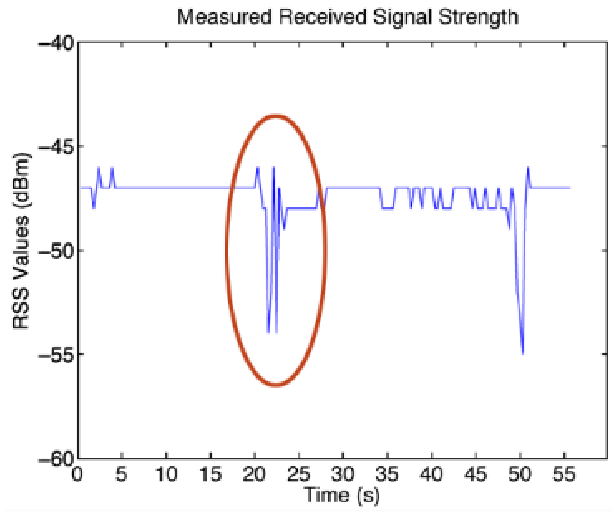


Figure 2.11: Expected Wi-Fi RSSI when a human is detected

- As the second phase, the focus is to use all possible non over lapping Wi-Fi channels for transmitting, receiving and reading RSSI values simultaneously for RTI.
  - For each non overlapping channel, Wi-Fi modules will be placed in a vertical line and each is configured to transmit in a single channel.
  - The receivers are placed in an opposite direction gathers these signal strengths in a timely manner and presents the data for graphs.

    At a given time all the transmitting Wi-Fi channels RSSI will be captured.

- This is further enhanced to monitor the Wi-Fi field within them and detect an intruder entering into the area, constructing a security alarm system.

2.4.3 Analysis of results and evaluation

The main goal of this research is to analyse Wi-Fi RSSI, on simultaneous transmission channels and detect human movements. This research could be evaluated and validated in the following ways,

- The graphs generated using the RSSI captured from the Wi-Fi modules will emphasize a human movement within the defined and monitored Wi-Fi range.
- The graphs generated using the RSSI captured from the Wi-Fi modules, of simultaneous transmission on different channels and statistical calculations using the standard deviation method could be observed.
- The practical usage, functionality and its test results of the developed prototype itself will evaluate the system and confirm that the targeted goals were achieved.

Chapter 3
Design & Implementation

3.1 Setting up and connecting devices

3.1.1 Connecting hardware devices

The two main components, the ESP8266-01 Wi-Fi module and the Arduino Mega 2560 which were discussed in the previous chapters are configured as follows.

The hardware specifications for these ESP8266-01 Wi-Fi modules and pin connections for the Arduino Mega 2560 could be referred from Appendix A.

Pin connections of the Wi-Fi module are shown in Figure 3.1 and Table 3.1.

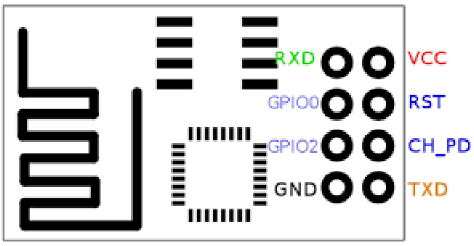


Figure 3.1: Pin connections of the ESP8266-01 Wi-Fi module

Pin	Description	Purpose
RXD	Receive Data	Data received into the module
GPIO0	General Purpose Input/Output no. 0	Can be used to pass an input into the module or get an output from the module
GPIO2	General Purpose Input/Output no. 2	
GND	0V	0 Voltage
VCC	3.3V	Power for the module
RST	Reset	Low restarts the module
CH_PD	Chip Power Down	Need to be high to enable the chip
TXD	Transmit Data	Data sent from the module

Table 3.1: Pin connections of the ESP8266-01 Wi-Fi module

ESP8266-01 Wi-Fi module and the Arduino Mega 2560 were set up as the diagram [28], in Figure B.2. Also, the Arduino Mega needs to be connected to the computer via USB and the Arduino software needs to be installed.

3.1.2 Installing the ESP8266 platform

The following steps are required to install the ESP8266 platform as it is not included in the Arduino IDE but needs to be downloaded and installed when required [29]. The ESP8266 version 2.3.0 was installed as in Figure 3.2, below.

- Open the preferences window from the Arduino IDE. Go to File > Preferences.
- Enter reference "http://arduino.esp8266.com/stable/package_esp8266com_index.json" into Additional Board Manager URLs field and click on "OK".

- Open Boards Manager by: Tools > Board > Boards Manager and select the ESP8266 board menu and install the "esp8266 platform".

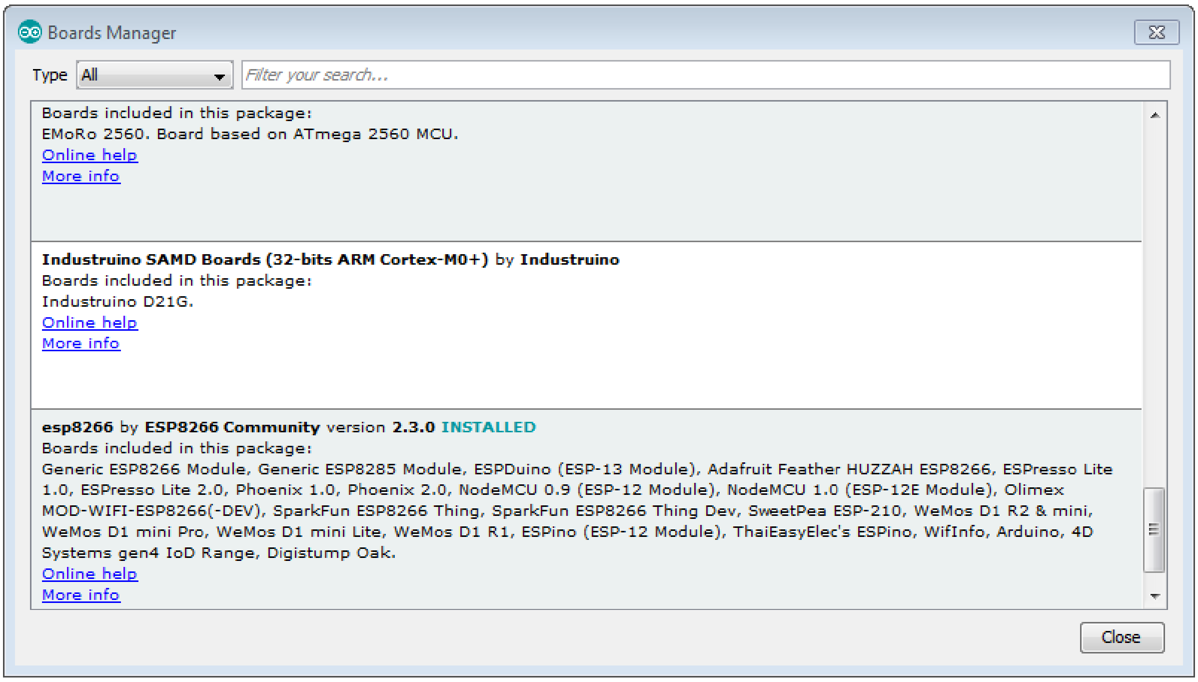


Figure 3.2: Installing the ESP8266 platform

3.1.3 Configuring Arduino software

The following need to be done in the Arduino software for the ESP8266-01 module to respond to the requests from the PC via USB to serial connection.

- Choose the Arduino Mega board from Tools > Board option.
- Set the baud rate as 115200bps which is recommended by the Wi-Fi module and should enable NL (new line) and CR (carriage return).

3.1.4 Communicating with the Wi-Fi module

The ESP8266-01, Wi-Fi module has a predefined set of firmware commands from which the following commands can be transmitted from the serial monitor terminal of the Arduino IDE to receive a positive response [28]. More commands are listed in Appendix A.

- Check if the Wi-Fi module is setup correctly and is functioning.

  Command sent: AT
  Response: OK

- Read the Wi-Fi mode connection mode stored in flash memory.

  Command sent: AT+CWMODE_DEF?
  Response:
  +CWMODE_DEF:<*mode*>
  OK
  <*mode*>: Connection mode returned, 1: Station, 2: Access point, 3: Both

- Set the Wi-Fi module as a station and save in flash memory.

  Command sent: AT+CWMODE_DEF=3
  Response: OK

- Read communication configuration, saved in flash memory.

  Command sent: AT+UART_DEF?
  Response:
  +UART_DEF=<*baudrate*>,<*databits*>,<*stopbits*>,<*parity*>,<*flow control*>
  OK

<*baudrate*>	: UART baud rate
<*databits*>	: data bits 5: 5-bit data, 6: 6-bit data, 7: 7-bit data, 8: 8-bit data
<*stopbits*>	: stop bits, 1: 1-bit stop bit, 2: 1.5-bit stop bit, 3: 2-bit stop bit
<*parity*>	: parity bit, 0: None, 1: Odd, 2: Even
<*flow control*>	: flow control, 0: flow control is not enabled, 1: enable RTS, 2: enable CTS, 3: enable both RTS and CTS

- Sets the communication configuration and save in flash.

  Command sent: AT+UART_DEF=9600,8,1,0,0
  Response: OK

3.1.5 Reading neighbouring Wi-Fi access points

When the firmware command is sent, the Wi-Fi module responds with a list of available access points, as follows.

Command sent: AT+CWLAP
Response:
+CWLAP:(3,"ZTE",-92,"cc:1a:fa:96:82:94",1)
+CWLAP:(4,"Dialog 4G",-89,"90:4e:2b:23:c0:e6",6)
+CWLAP:(3,"PROLINK_H5004NK_A5ED5",-63,"90:61:0c:1a:5e:d5",11)
OK

In general the result is,
+CWLAP: <*ecn*>,<*ssid*>,<*rssi*>,<*mac*>,<*ch*>

<*ecn*>	: encryption method. 0: OPEN, 1: WEP, 2: WPA_PSK, 3: WPA2_PSK, 4: WPA_WPA2_PSK
<*ssid*>	: SSID of the AP
<*rssi*>	: Signal strength
<*mac*>	: MAC address of the AP
<*ch*>	: Transmission channel

3.2 GUI for serial communication and generating graphs

A program with a graphical interface was developed in Visual FoxPro (VFP), which is a programing language by Microsoft, to read text from the virtual serial port which is created by the Arduino software, using the Windows communication control [26]. This program sends predefined ESP8266-01 commands in a timely manner, reads the response from the communication, extracts necessary information, saves the information in VFP databases, generates MS Excel graphs [30] and displays them on screen.

Sample data gathered and stored in databases are listed in Figure D1, Table D.1, Table D.2 and Table D.3. Screen images of the program developed is shown below in Figure 3.3 and Figure 3.4. Source code extracts are shown in Figure C.1, Figure C.2 and Figure C.3.

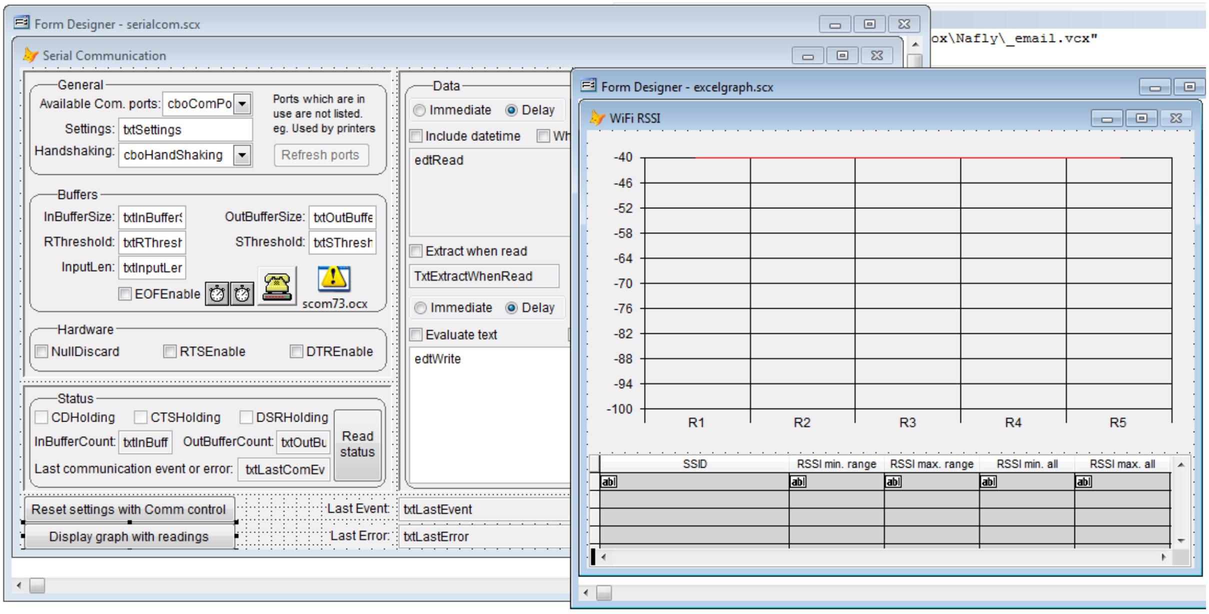


Figure 3.3: GUI in development environment

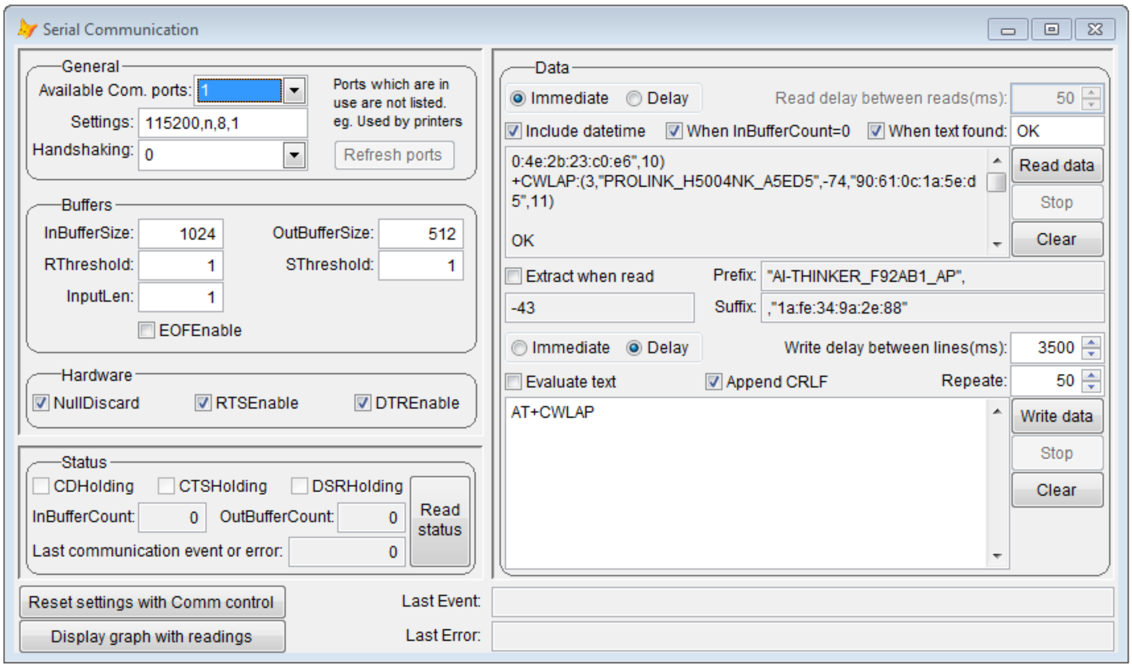


Figure 3.4: Developed GUI monitoring the serial communication capturing RSS

3.3 Reading Wi-Fi signal strengths with channel scanning

3.3.1 Measuring neighbouring Wi-Fi RSSI

The Wi-Fi Receive Signal Strength was measured from the neighbouring Wi-Fi access points as follows, with the program written to continuously send commands to the module and receiving a response. Collected data were used for graphs and samples are listed in Figure E.1 and Figure E.2.

- The distance between the Wi-Fi transmitter and receiver was about 2.5m.
- The receiver and the transmitter have not established a connection, but the receiver reads the available Wi-Fi SSIDs by scanning channels and their signal strengths.
- The disturbance to the RSSI was measured placing a human in almost the middle of the transmitter and the receiver.
- A Wi-Fi module ESP8266-01 was connected through an Arduino Mega with an USB connection to the PC as the receiver and another ESP8266-01 was setup on the same channel as the transmitter.
- A Microsoft excel graph is displayed by the developed program, which is updated with the RSSI every 3.5 seconds.

  Shorter intervals cannot be used as the Wi-Fi module takes some time to respond to requested commands.
- The serial communication baud rate is 115200bps, which is recommended by the Wi-Fi module.

3.3.2 Observations and limitations

The following were observed by the channel scanning implementation.

- There is no connection between the access point (transmitter) and client (receiver) but scans to identify the APs.
- When the firmware command is sent to read the available access points it takes two to three seconds to get the results with three access points in the list. This delay may be too long to capture fast movements within the monitored area.
- When there are more surrounding access points, the returned result list is long, gets delayed and the next cycle may not return results.

Therefore it is required to follow another approach which is programing the microcontroller of the ESP8266-01 Wi-Fi module to achieve required objectives.

3.4 Reading Wi-Fi RSS by connecting to the network

3.4.1 Developing the program for the ESP8266-01 Wi-Fi module

Arduino provides a library which could be used for this purpose [36] with documentation and samples [37]. The following is an extract of the main method of the source code saved into the ESP8266-01 program memory. More extracts are listed in Figure C.4 and Figure C.5.

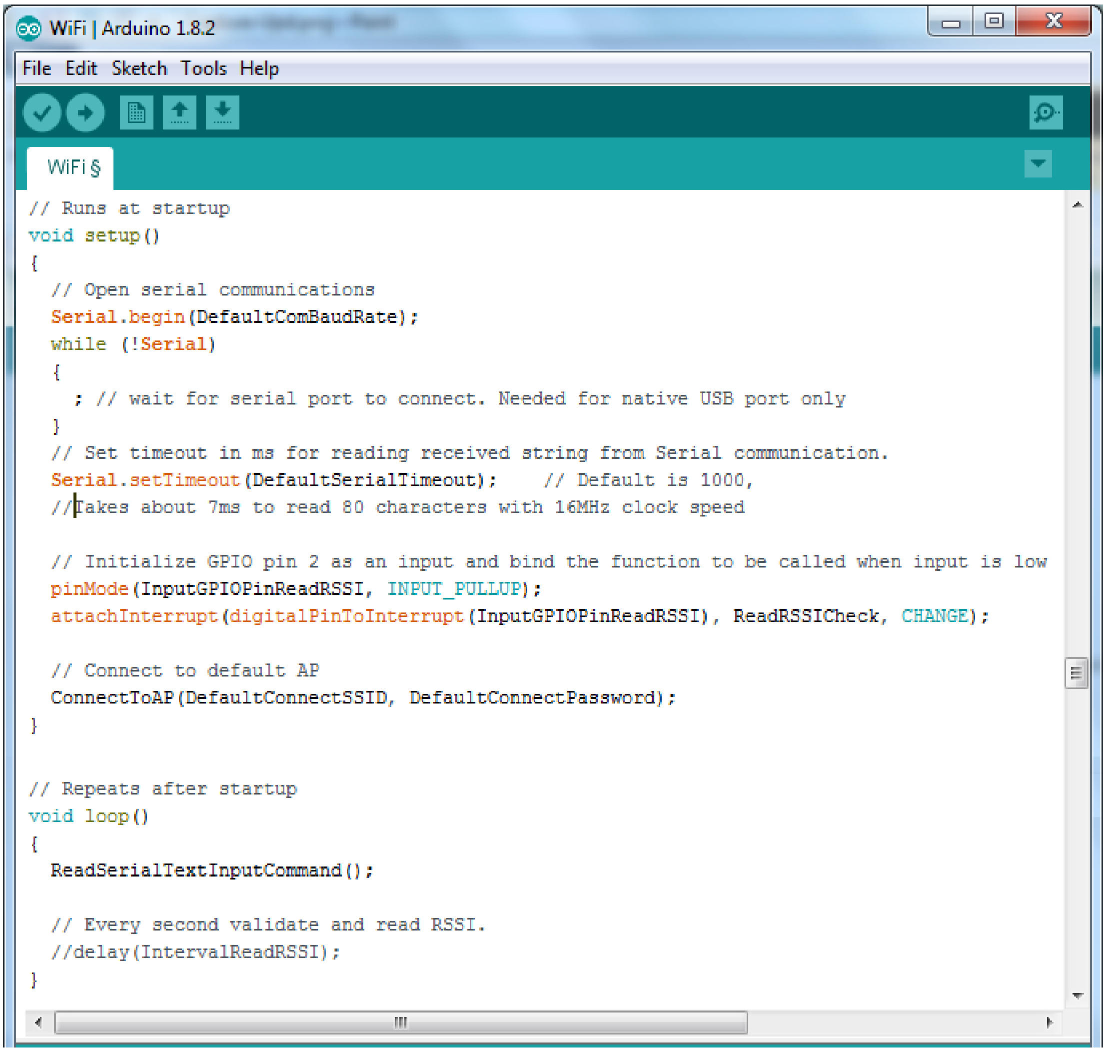


Figure 3.5: Main function of the program written into the Wi-Fi program memory

3.4.2 Writing the code into the ESP8266 program memory

In order to program the ESP8266-01 module, the connections between the Wi-Fi module and Arduino Mega board need to be setup as in Figure B.2 with the following steps.

- Choose the Generic ESP8266 Module from Tools > Board option.
- Set the baud rate as 115200bps which is recommended by the Wi-Fi module and should enable NL (new line) and CR (carriage return).

The diagram of the hardware developed to upload a program into the ESP8266-01 module is as shown in Figure B.4. Hardware developed to program the ESP8266-01, Wi-Fi module with an Arduino Uno board is shown in Figure B.5. External power need to be supplied as USB power is insufficient to program the microcontroller.

The program, esptool.exe used by Arduino IDE to write to the ESP8266-01 program memory, does not function as expected and causes an error [43] as in Figure B.6. As a solution a Python script, esptool.py can be installed and configured as in Figure B.7 to Figure B.9.

Once a program is uploaded into the ESP8266-01 module the functionality of the default inbuilt AT commands are overwritten. To program the ESP8266-01 with the default program, a tool [41] and instructions [42] are available.

3.4.3 Developing a program for Arduino to control the Wi-Fi modules

The purpose of this program is to control the three Wi-Fi modules configured to operate on non-overlapping W-Fi channels, 1, 6 & 11 and to read their RSSI in regular intervals every second. The program triggers the Wi-Fi modules to perform a RSSI read simultaneously via a configured output pin, to input pins of the Wi-Fi modules. Figure 3.7 illustrates this function.

A source code extract of the repeating function which is the loop method is shown, below in Figure 3.6. More source code extracts are in Appendix C and Figure 3.9 additionally shows the captured RSSI results.

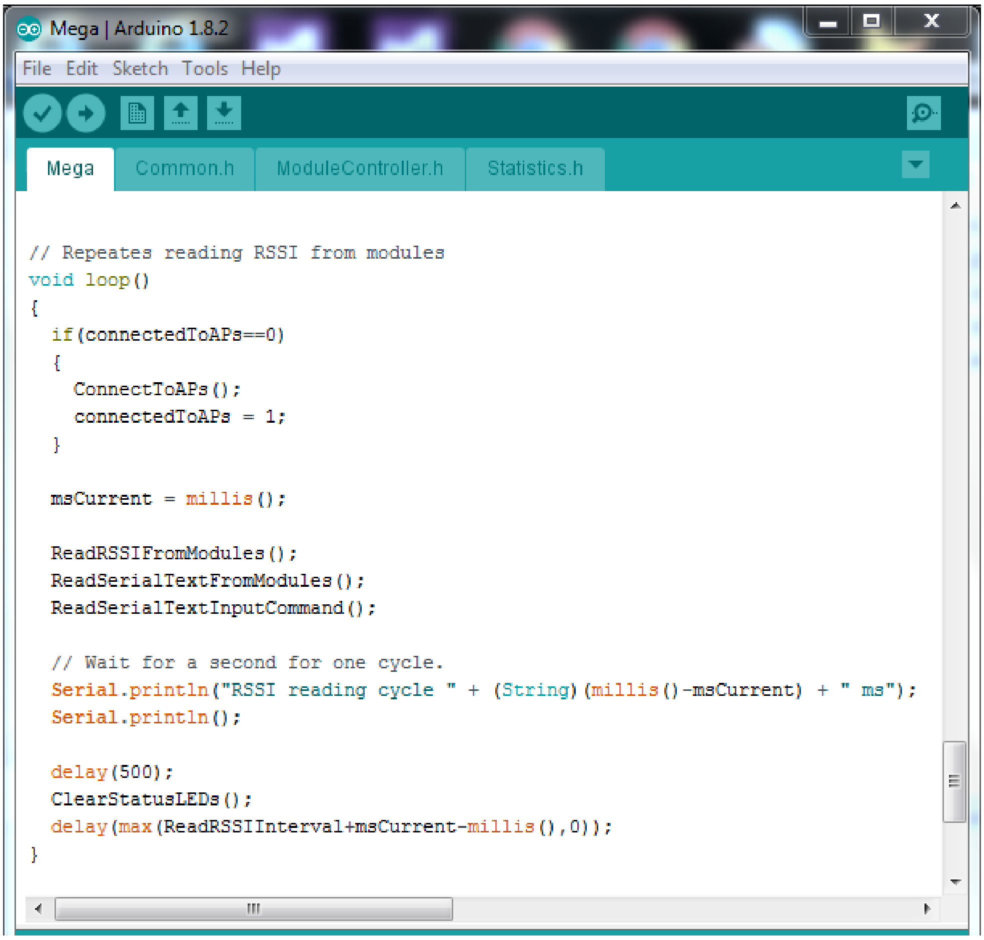


Figure 3.6: Source code extract from the program written into Arduino memory

3.4.4 Writing the code into the Arduino Mega program memory

Configurations need to be done as in section 3.1.3 and the program mentioned in section 3.4.3 is uploaded into the Arduino Mega microcontroller. It indicates how much of program memory is used and remaining as in Figure 3.9.

3.4.5 Connecting the components

In summary, a node consists three Wi-Fi modules configured to simultaneously transmit and receive RSSI, on non-overlapping Wi-Fi channels, controlled by the an Arduino Mega which does necessary calculations based on the received RSSI, detects unusual changes in the RSSI and transmits the information to a computer to display graphs. Such nodes are placed in an area to monitor and detect human movements. The diagram used to build the hardware prototype for this is illustrated in Figure 3.7, specifications of the prototype are shown in Table 3.2 and the hardware prototype is shown in Figure 3.8. Similar prototype built with ESP8266-07 is shown in Figure B.3.

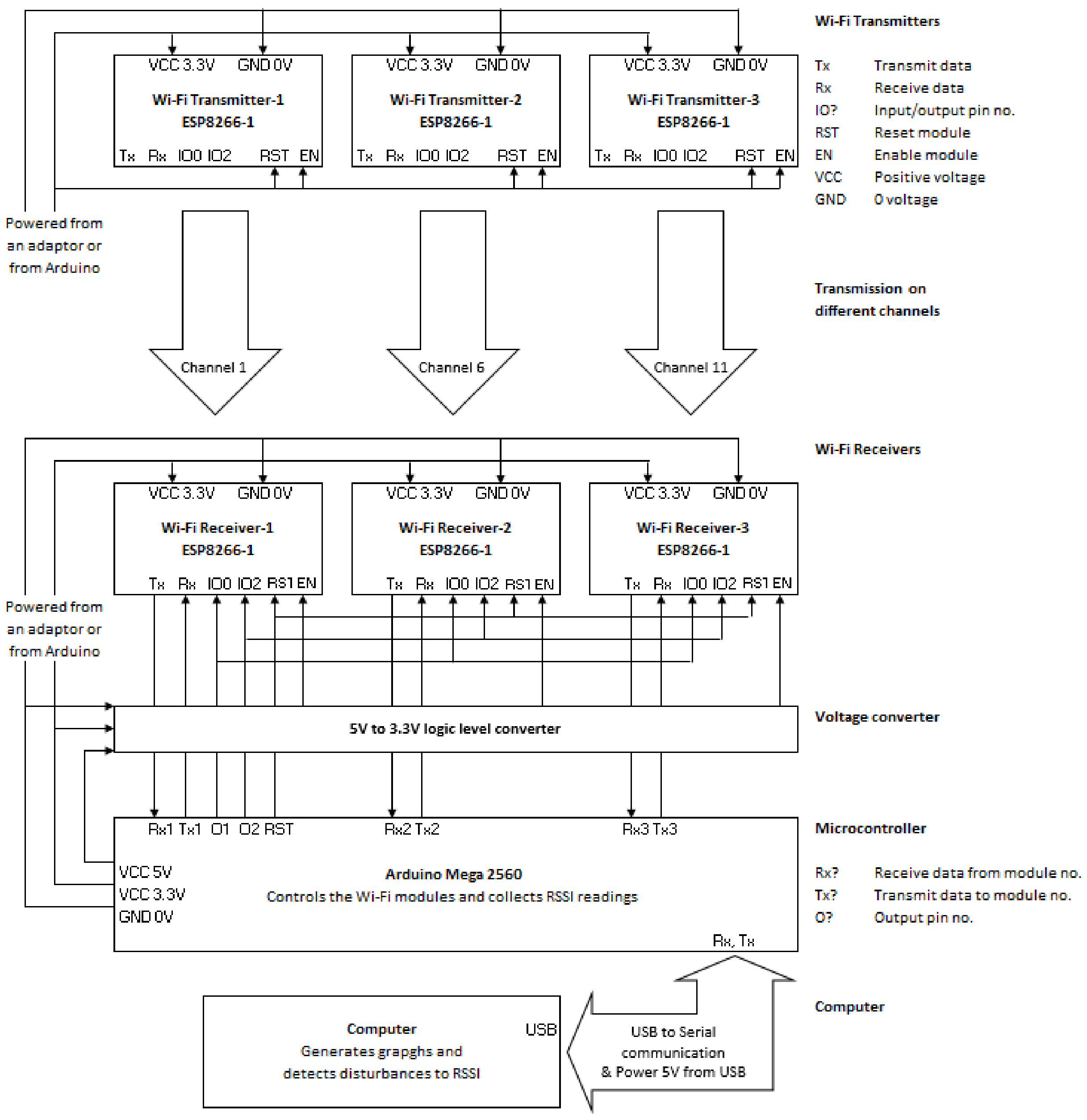


Figure 3.7: Diagram of Wi-Fi modules for simultaneous transmission and RSSI

Category	Attributes	Parameters
Physical characteristics, testing	Positions of the prototypes	4 modules, each at a corner forming a square (testing was done with 2 modules)
	Distance between prototypes	Length and breadth, 4 x 4 m
	Distance from ground level to a prototype	1m
	No. of Arduino Mega boards included for a prototype	1
	No. of Wi-Fi modules included for a prototype	3
	Distance between each Wi-Fi module within a prototype	52 mm
	Prototype container	Plastic box, white
	Prototype size	Length x width x height, 174 x 88 x 47 mm
	Prototype weight	230g
Technical (for a prototype)	RSSI reading frequency	Every 1s from the connected 3 Wi-Fi modules
	Method triggering RSSI readings	Via input pin on Wi-Fi modules connected to an output pin of Arduino Mega
	Used Wi-Fi channels	Each Wi-Fi module is set to a non-overlapping Wi-Fi channel. Channels no.s 1, 6 and 11 are used.
	Connection time of Wi-Fi module to an AP	About 7s
	Calibration time of the Wi-Fi modules after start up (no. of samples read prior to human detection)	60s
	No. of samples used for standard deviation	Last 60 samples read
	Communication between Arduino and Wi-Fi modules	Serial via TX and RX pins
	Communication between the computer and Arduino boards	Via USB to Serial(TX and RX)
	Input voltage	7-12 V
	Operating voltage	Arduino Mega 5V, Wi-Fi module 3.3V
	Operating current	About 500 mA
	Power source	AC mains to DC 9V power adaptor as power from computer's USB is insufficient
Environment	Humidity and Temperature	Normal (about 28°C, 75%). Not raining and no air conditioning.
	Testing	No obstacles between transmitter and receiver. With obstacles such as humans, walls, etc…

Table 3.2: Specifications of the built hardware prototype

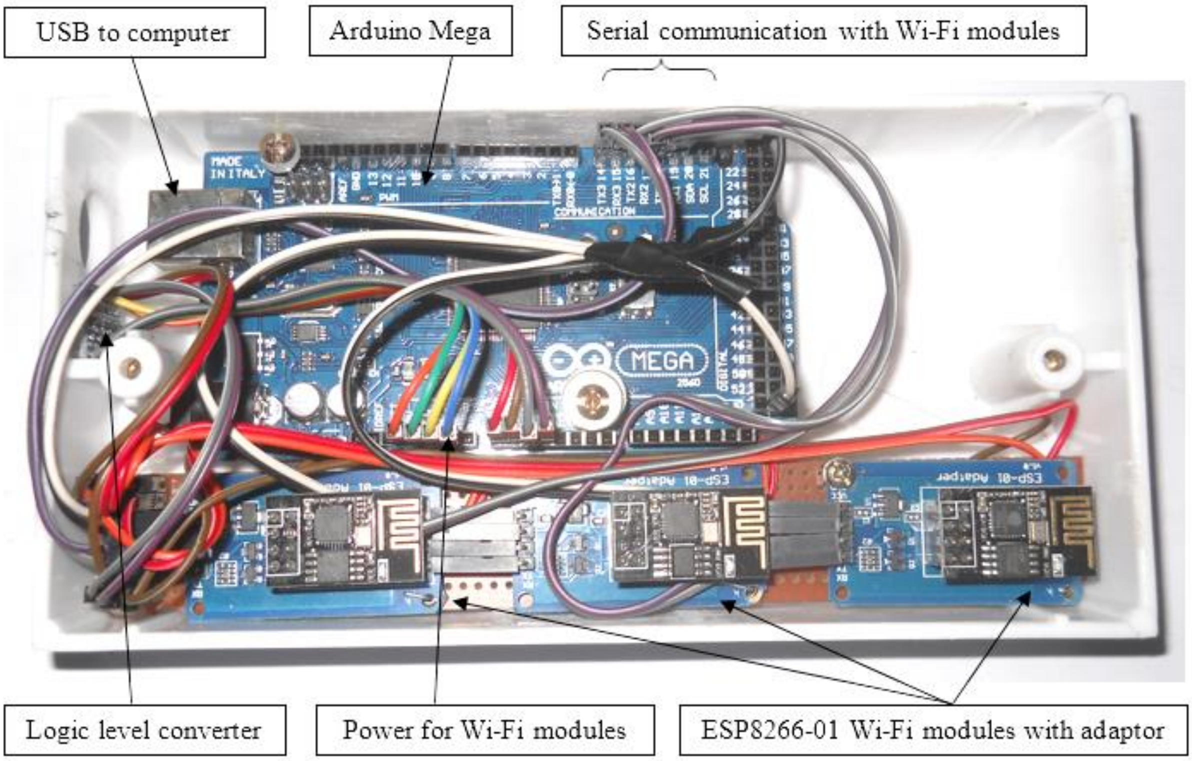


Figure 3.8: Hardware prototype to capture Wi-Fi RSSI on non-overlapping channels

3.4.6 Testing the developed prototype

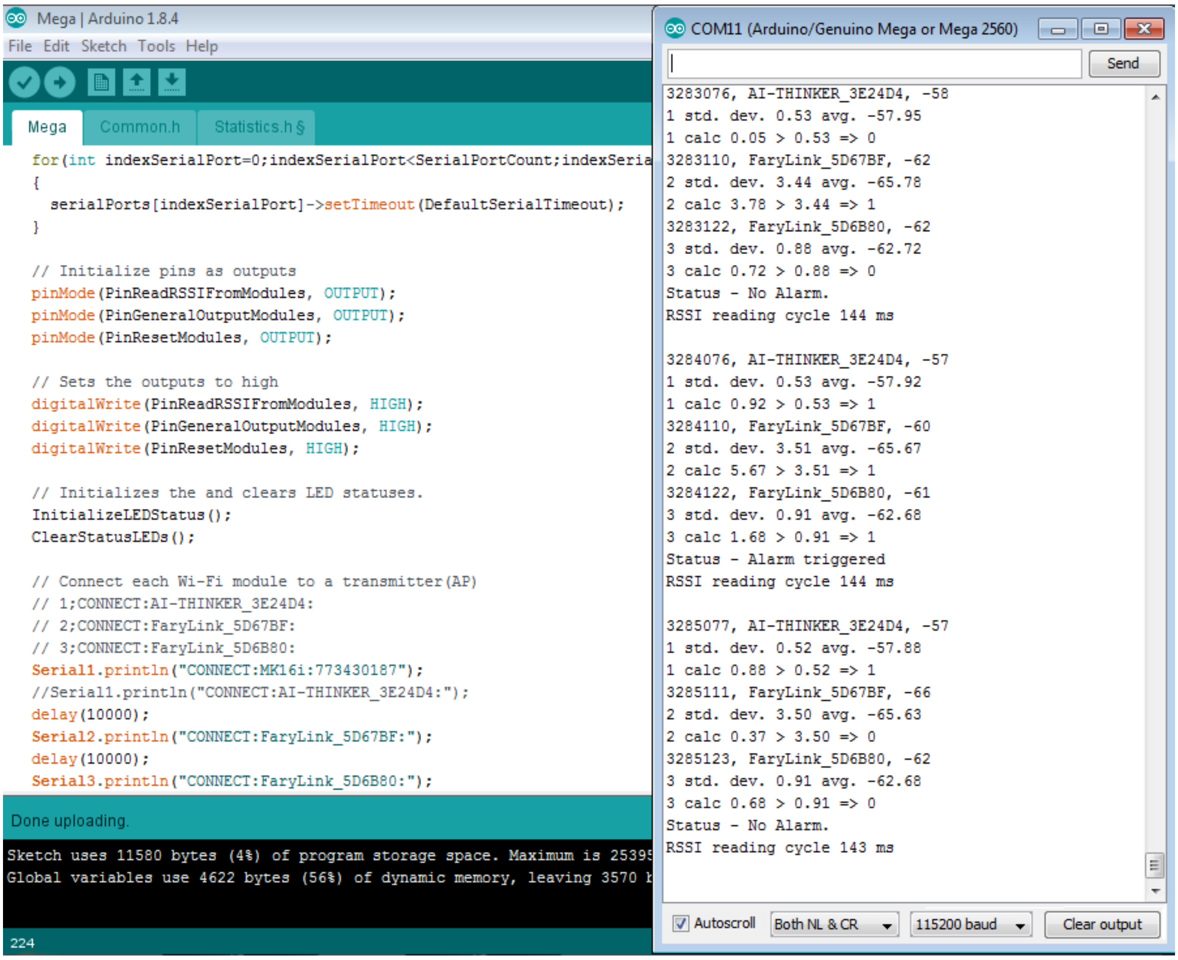


Figure 3.9: Testing the developed prototype

As shown in Figure 3.9 the prototype is tested. The below functions are performed.

- At start up, reads the SSIDs, passwords, channels stored in the Arduino Mega EEPROM.
- Sets the channel, connects each Wi-Fi module, to its relevant SSID network
- Reads RSSI from each Wi-Fi module, calculate the standard deviation as described in section 3.5 and waits until the reading are stable. Also sends the received RSSI to the computer for generating graphs.
- Starts to monitor the RSSI for deviations higher than the expected and triggers an alarm.

3.5 Determining deviations in RSS with statistical calculations

When Wi-Fi RSSI is monitored, it has slight fluctuations as seen in the test results with the graphs. For an example, Figure 4.18 could be observed. This may be due to the environment and the power of transmission by the device itself not been consistent. Therefore to smoothen such minor deviations and to identify major deviations, such as a movement within the monitored area, as in Figure 4.19, a statistical calculation method, the Standard Deviation [32] can be used as the formula shown in Figure 3.10 [35].

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

σ - Standard deviation of a sample
x_i - Each value in the dataset
$\bar{x}$ - Mean of all values in the dataset
n - No. of values in the dataset

Figure 3.10: Formula of Standard Deviation used for calculations

Further explaining, “Standard deviation is a widely used measurement of variability or diversity used in statistics and probability theory. It shows how much variation or ‘dispersion’ there is from the ‘average’ (mean, or expected value). A low standard deviation indicates that the data points tend to be very close to the mean, whereas high standard deviation indicates that the data is spread out over a large range of values.” [39].

Distribution of the probability can be viewed in the Bell Curve as in Figure 3.11 [34].

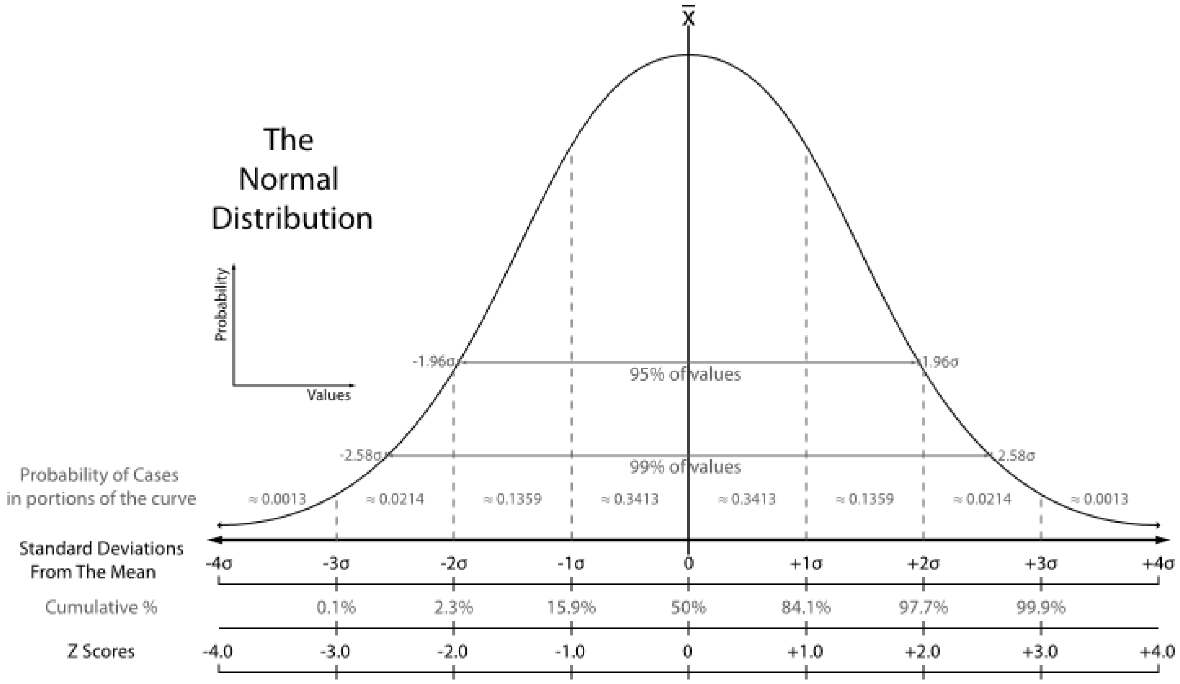


Figure 3.11: Distribution of the probability in the Bell Curve

Arduino provides a library which could be used for this purpose [33]. Therefore, the standard deviation is calculated for RSSI values read on each channel and compared as the following source code snippet to identify a human movement within the monitored signal path.

```
bool stdDevStatusChannel1 = (stdDeviationRSSIChannel1 < readRSSIChannel1 – averageRSSIChannel1);
bool stdDevStatusChannel2 = (stdDeviationRSSIChannel2 < readRSSIChannel2 – averageRSSIChannel2);
bool stdDevStatusChannel3 = (stdDeviationRSSIChannel3 < readRSSIChannel3 – averageRSSIChannel3);
if(stdDevStatusChannel1 and stdDevStatusChannel2 and stdDevStatusChannel3)
{
  // Alarm on, human detected.
}
```

Chapter 4
Results & Evaluation

4.1 Evaluating the system

The main goal of this research is to analyse Wi-Fi RSSI on multiple Wi-Fi transmission channels and detect human movement more accurately. This research could be evaluated and validated in the following ways,

- Graphs generated using the RSSI captured from Wi-Fi modules operating on a single channel for different interfering objects can be used to emphasize a human movement crossing the signal path are listed in section 4.2.1.
- Graphs generated using the RSSI captured from the Wi-Fi modules on multiple channels placing the Wi-Fi transmitter, receiver and human in multiple locations, could identify human presence with the RSSI variation by signal reflection and signal blockings are listed in section 4.2.2.
- Deviations in RSSI due to human movements are highlighted in section 4.2.3.
- Also, experiments were carried out to measure RSSI through walls are described in section 4.2.4.
- The practical usage, functionality and its test results of the developed prototype itself evaluates the system and confirms that the targeted goals were achieved.
  An overview of the built hardware prototypes placed and measuring RSSI is similar to Figure 4.1. Although the figure shows two nodes, it is expanded to four nodes so that the boundaries of the rectangular area are monitored.

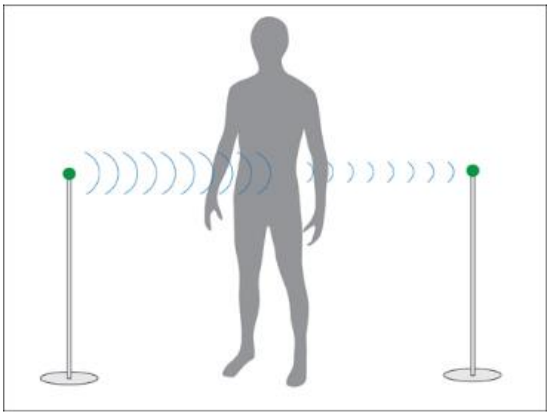


Figure 4.1: View of placing built hardware prototypes to monitor movement

The following sections emphasize on the testing and test results on each of the above.

4.2 Test results

4.2.1 Experiments with different objects

The following are some of the results of the tests that were carried out to identify the variance in RSSI, when different objects were placed in between a Wi-Fi transmitter and receiver, using a single channel with the below attributes.

- RSSI readings are taken every second from SSID, AI-THINKER_3E24D4.
- Distance between the transmitter and receiver is about 2.5m.
- Transmitter, ESP8266-01 module, configured with inbuilt commands (AT commands)
- Receiver, ESP8266-07 module, program uploaded to chip with Arduino IDE and Arduino Uno3 to connect with a Wi-Fi access point and read RSSI every second.
- Receiver is connected to Arduino Uno3 with serial communication (Rx, Tx)
- Arduino Uno3 is connected to a laptop via virtual USB to serial interfacing (Virtual COM) with a baud rate of 115200bps.
- Transmitter is on Wi-Fi channel 1(center frequency 2412 MHz), no encryption (Open), no password.
- ESP8266-01, ESP8266-07 and Arduino Uno3 are powered by the laptop's USB port.
- The GUI is developed in VFP9 using MS Excel graphs to plot the read RSSI.
- Calculations below the graphs show the minimum, maximum, average and standard deviation of the readings.

These Wi-Fi modules used for testing are cheap and suitable for constructing prototypes. Therefore due to its quality, the transmitted Wi-Fi signals are accessible within a short range. Similarly the Wi-Fi receivers are not much sensitive to identify weak signals. Also the readings gathered and visualized on graphs needs to show significant differences which could be observed and come to conclusions based on its results. These factors, limitations and testing determined the distance between the transmitter and receiver to 2.5m as noted above.

The below test results, from Figure 4.2 to Figure 4.11, reveal that Wi-Fi signals are attenuated by substances containing water and depends of the volume of the substance. Comparing the results of a moving wooden log as in Figure 4.8 and a human movement as in Figure 4.11, although they are similar, a human cause more disturbance to the Wi-Fi RSS. This research attempts to identify human movements and not positions of stationary objects. A log can be moved by a human covering himself and still would cause major disturbances in the RSS. Therefore a human movement can be identified comparing with other objects.

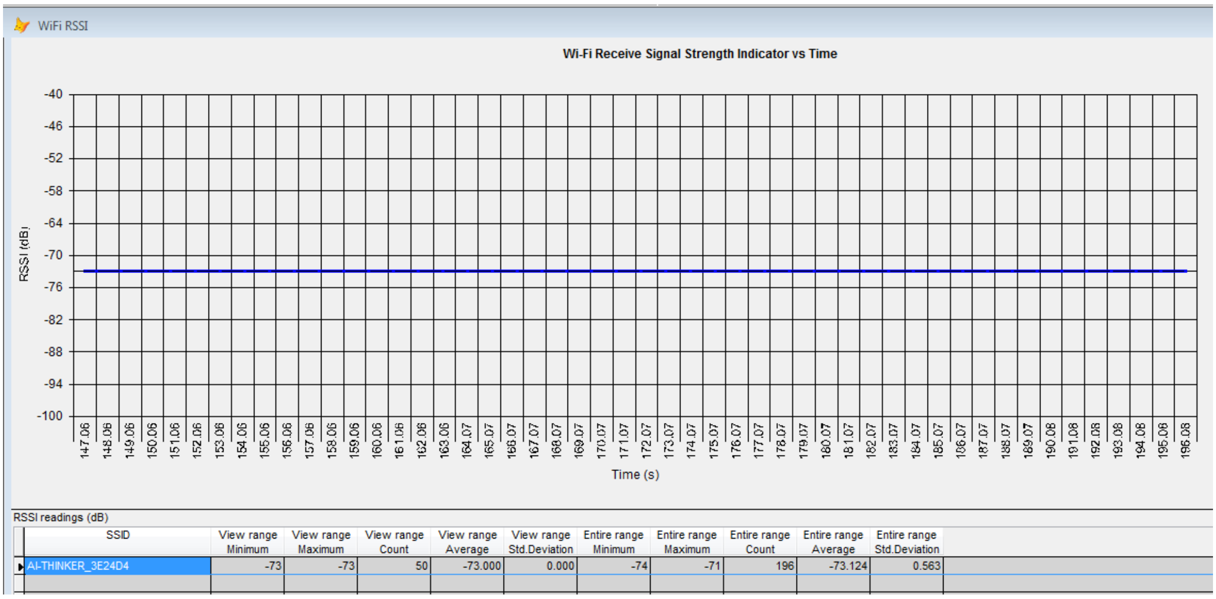


Figure 4.2: No disturbance in RSSI

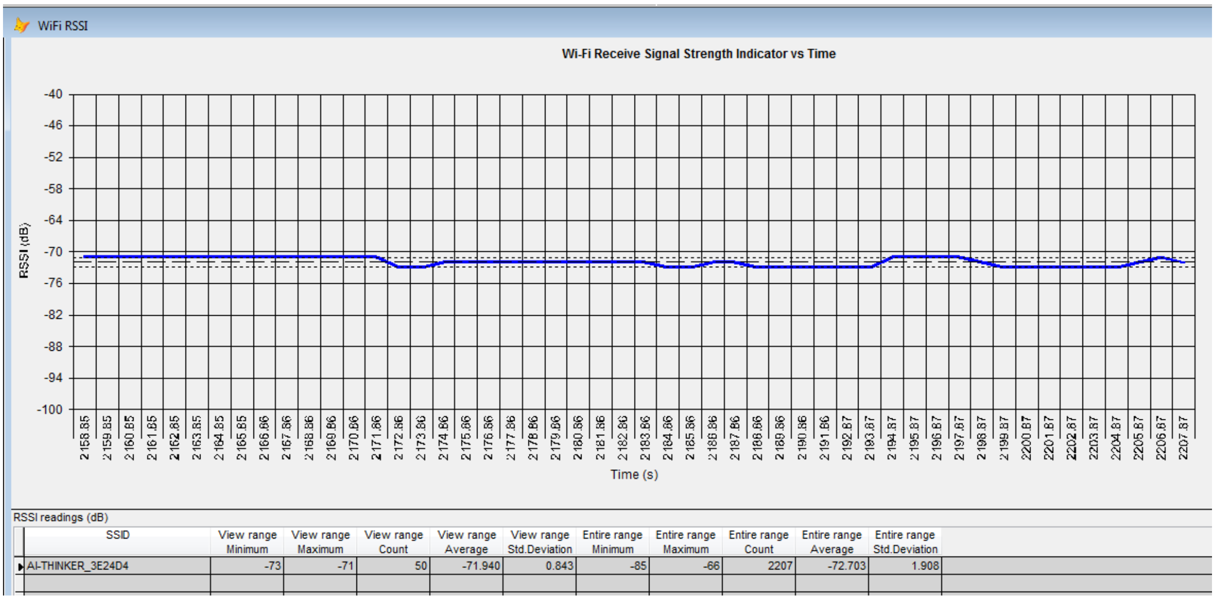


Figure 4.3: Cardboard sheet moved across the Wi-Fi signal path

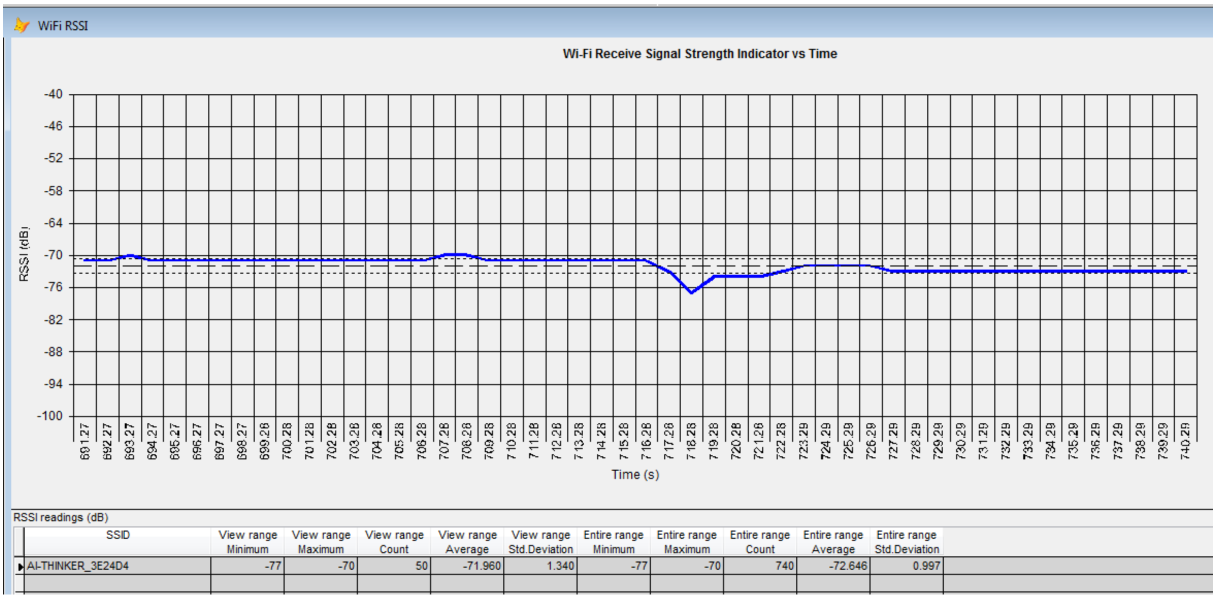
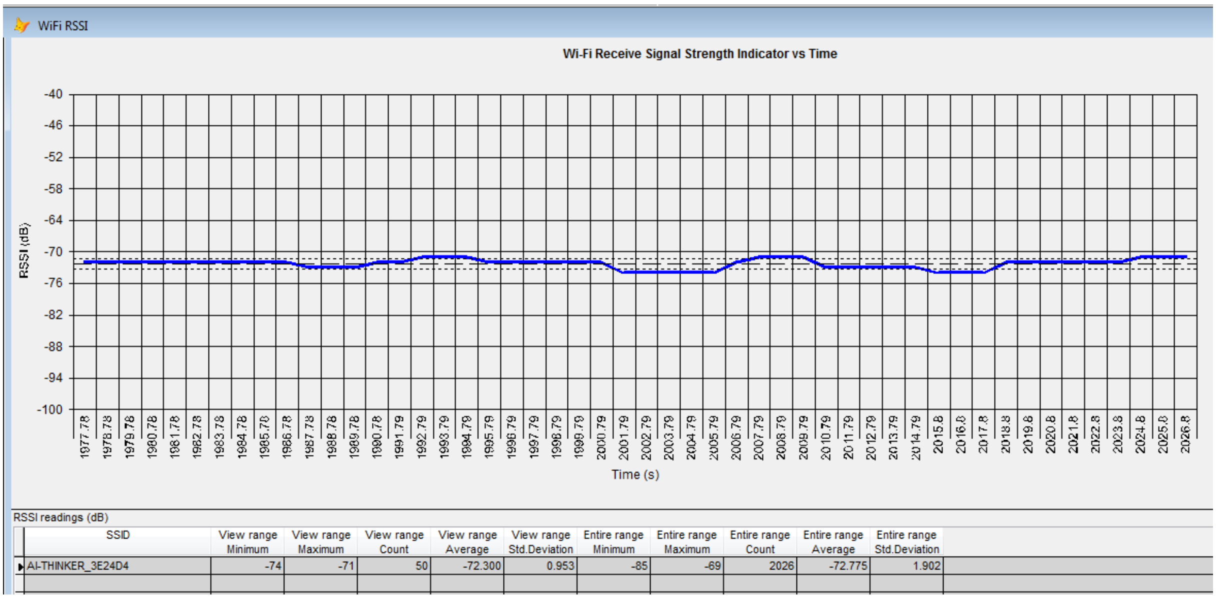
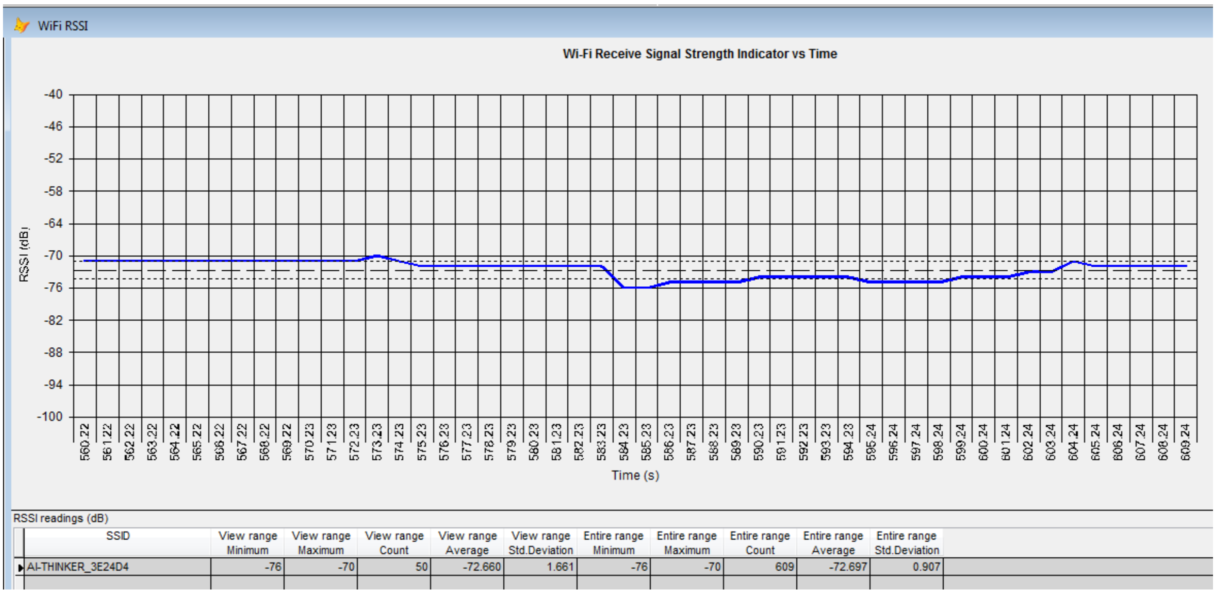


Figure 4.4: Glass of water moved across the Wi-Fi signal path



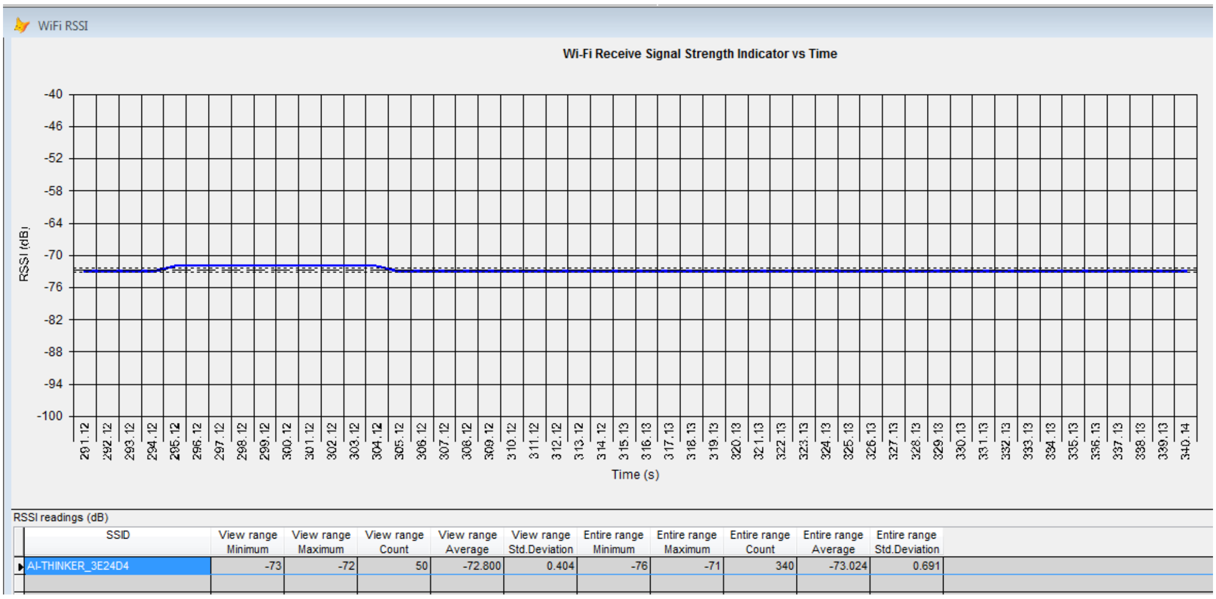
	SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
▸	AI-THINKER_3E24D4	-74	-71	50	-72.300	0.953	-85	-69	2026	-72.775	1.902

Figure 4.5: Damp towel moved across the Wi-Fi signal path



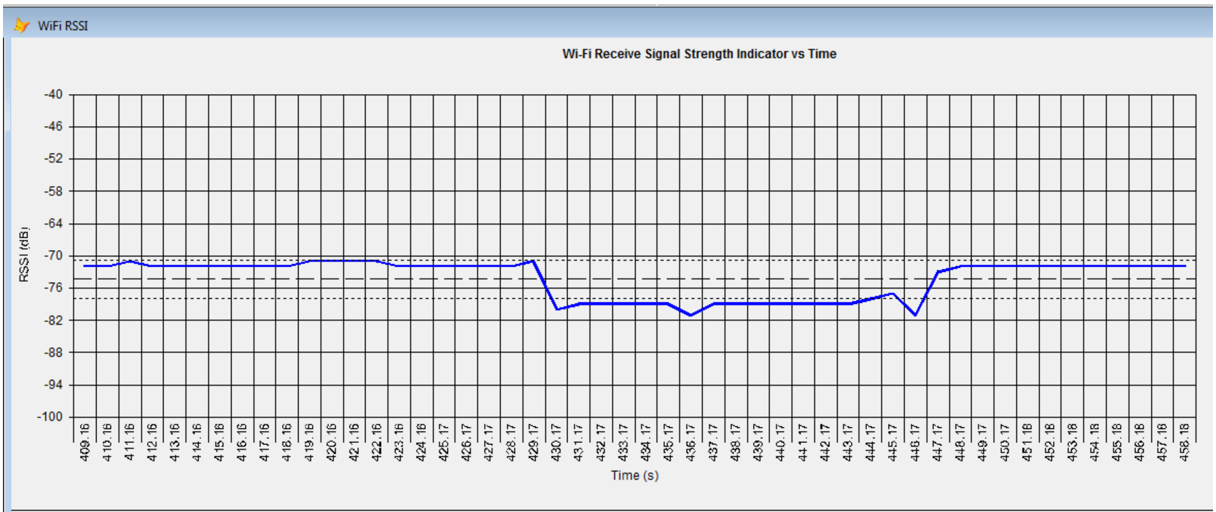
	SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
▸	AI-THINKER_3E24D4	-76	-70	50	-72.660	1.661	-76	-70	609	-72.697	0.907

Figure 4.6: Metal tray moved across the Wi-Fi signal path



	SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
▸	AI-THINKER_3E24D4	-73	-72	50	-72.800	0.404	-76	-71	340	-73.024	0.691

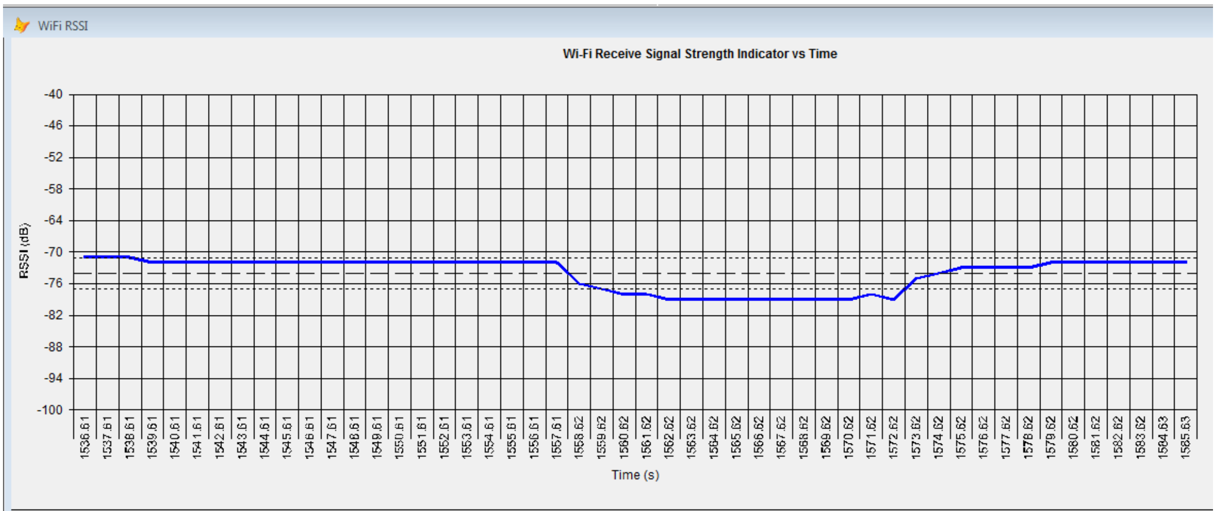
Figure 4.7: Wooden board moved across the Wi-Fi signal path



RSSI readings (dB)

SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-81	-71	50	-74.320	3.537	-82	-69	458	-74.336	3.229

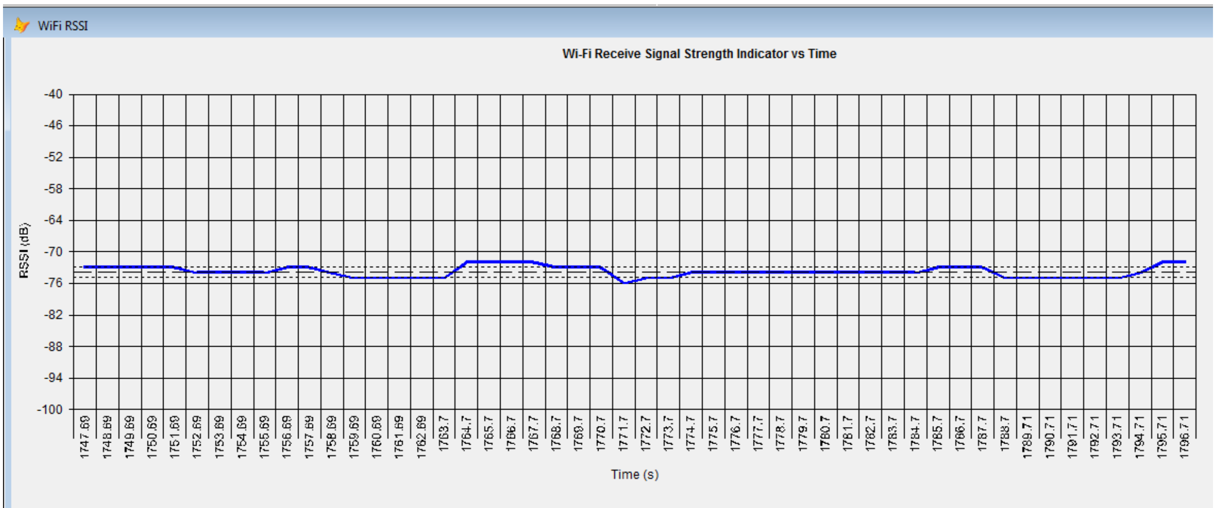
Figure 4.8: Wooden log moved across the Wi-Fi signal path



RSSI readings (dB)

SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-79	-71	50	-74.060	3.020	-85	-69	1585	-72.792	1.882

Figure 4.9: Leather bag moved across the Wi-Fi signal path



RSSI readings (dB)

SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-76	-72	50	-73.800	1.030	-85	-69	1796	-72.818	1.912

Figure 4.10: Plastic cover moved across the Wi-Fi signal path

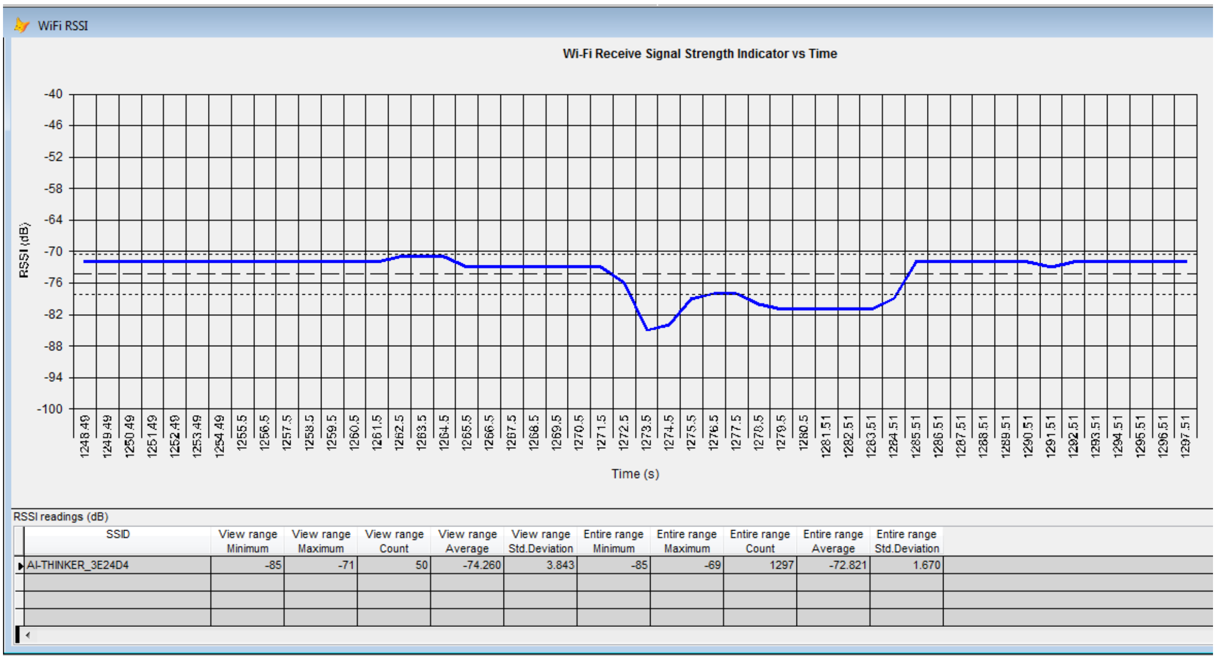


Figure 4.11: Human moved across the Wi-Fi signal path

4.2.2 Experiments with multiple Wi-Fi channels

Experiments were carried out with the following attributes. Some graphs are shown in Figure 4.12, Figure 4.13 and more are shown in Figure E.3 to Figure E.10 in Appendix E.

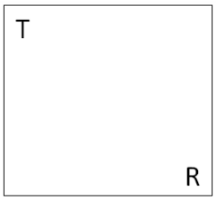
- Receivers are connected to Arduino Mega with serial communication for each module (Rx1, Tx1, Rx2, Tx2, Rx3, Tx3)
- Arduino Mega is connected to a laptop via virtual USB to serial interfacing (Virtual COM) with a baud rate of 115200bps.
- Transmitters are on Wi-Fi channel 1, 6 and 11 (center frequencies 2412, 2437, 2462 MHz) without encryption (Open) and password set.
- Transmitter ESP8266-01 modules are powered by an external power adapter and receiver ESP8266-01 modules are powered by Arduino Mega connected to the laptop’s USB port.
- The GUI developed in VFP9 using MS Excel graphs to plot read RSSI.
- The calculations below the graphs show the minimum, maximum, average and standard deviation of the readings.

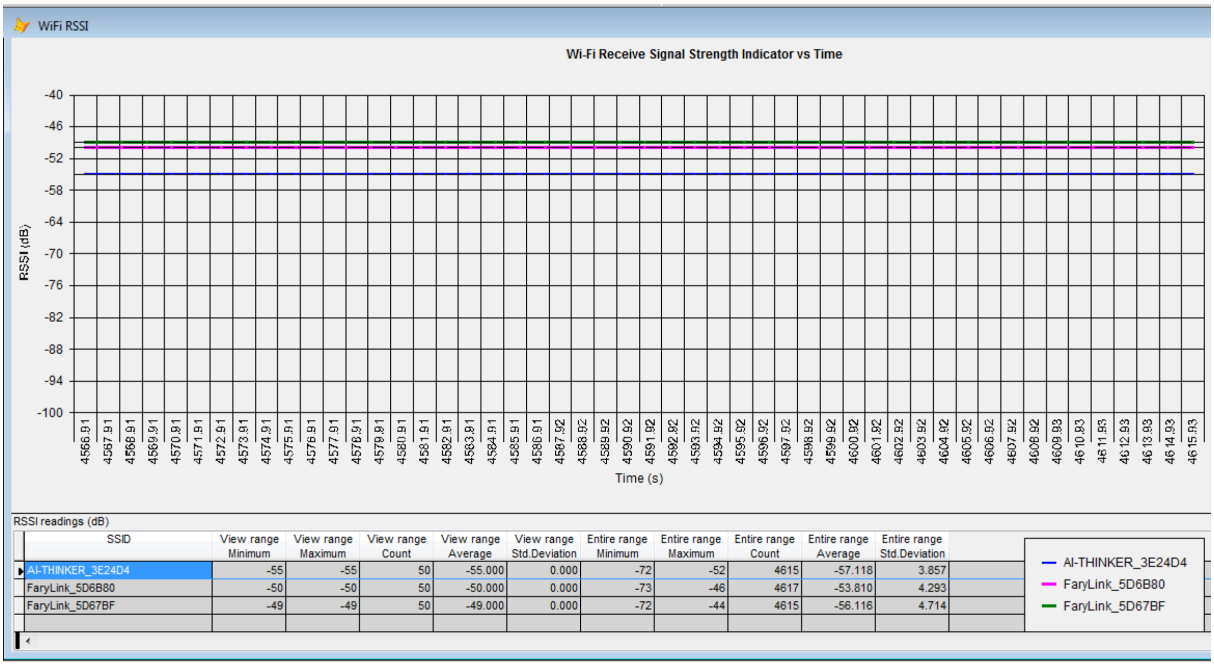
According to the results, all three Wi-Fi channels have a deviation and it could improve the accuracy. Also, by examining the graphs from Figure E.3 to Figure E.10, they show that, a human placed not in a line of sight between the transmitter and the receiver has a minor impact on the RSSI. This discloses that Wi-Fi signals reflect and cause these deviations.

As researches disclose, such signals could be monitored and an image of the object can be generated.

T – Transmitter
R – Receiver
H – Human

RSSI is not disturbed and clear reception

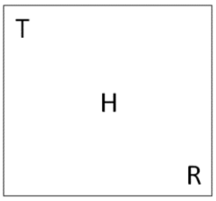


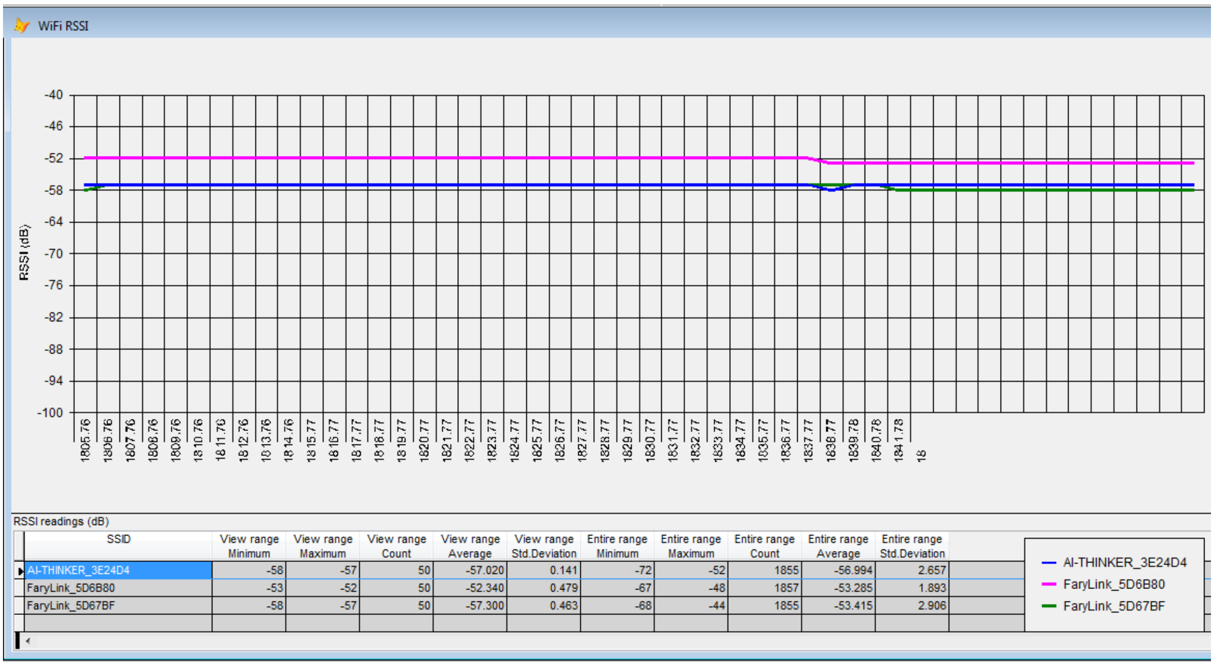


SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-55	-55	50	-55.000	0.000	-72	-52	4615	-57.118	3.857
FaryLink_5D6B80	-50	-50	50	-50.000	0.000	-73	-46	4617	-53.810	4.293
FaryLink_5D67BF	-49	-49	50	-49.000	0.000	-72	-44	4615	-56.116	4.714

Figure 4.12: Multi channel RSSI without disturbance

RSSI signal decrease can be noticed as human blocks the signal path.





SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-58	-57	50	-57.020	0.141	-72	-52	1855	-56.994	2.657
FaryLink_5D6B80	-53	-52	50	-52.340	0.479	-67	-48	1857	-53.285	1.893
FaryLink_5D67BF	-58	-57	50	-57.300	0.463	-68	-44	1855	-53.415	2.906

Figure 4.13: Multi channel RSSI deviation with human in the transmission path

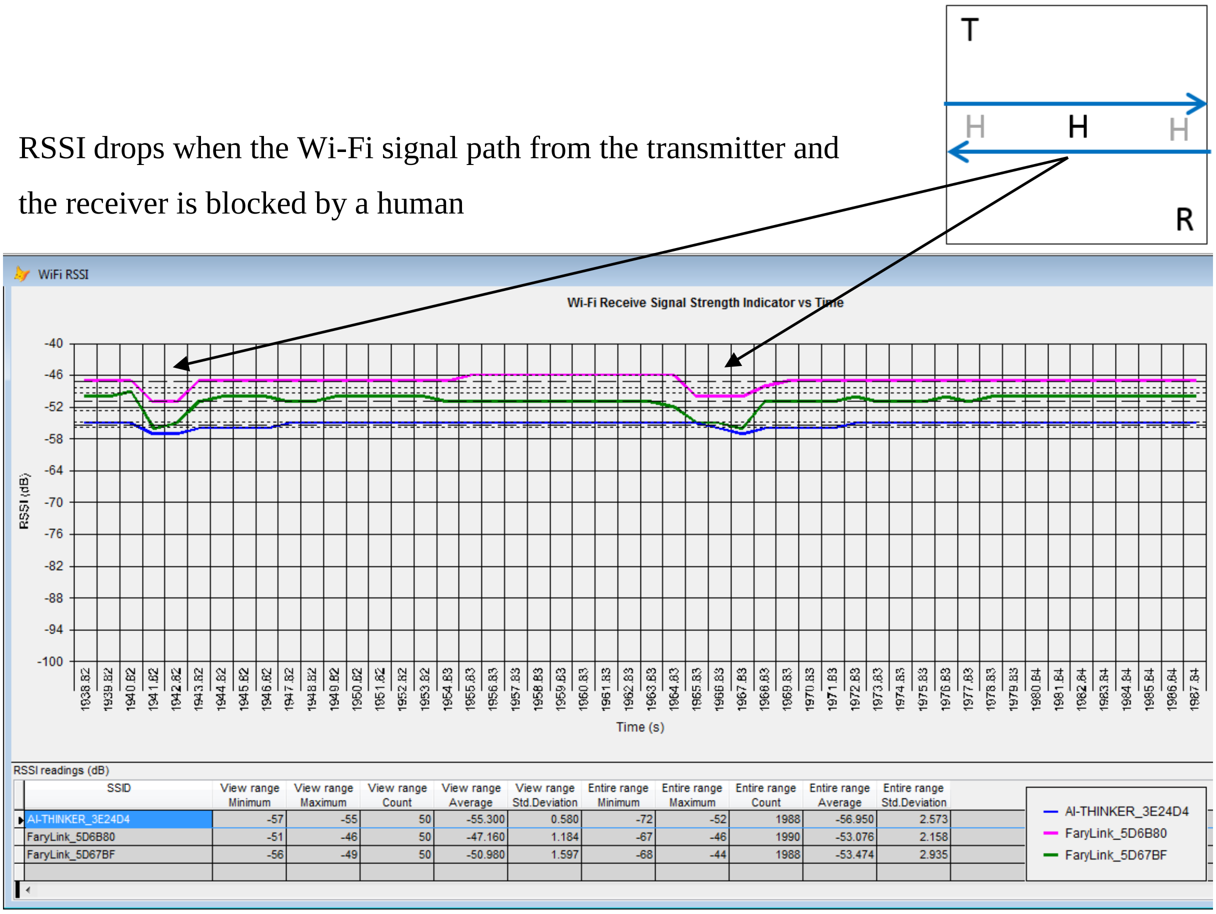
4.2.3 Experiments with Human movements

The following are some of the results of the tests that were carried out to identify the patterns in RSSI, when multiple Wi-Fi channels were used with different human movements with the below attributes and graphs are shown from Figure 4.14 to Figure 4.17.

- Tests were carried out in a square area with the distance of 4m in a side.
- Transmitter ESP8266-01 modules are configured with inbuilt commands (AT commands).
- Receiver ESP8266-01 modules are programed (program uploaded to chip) with Arduino IDE and Arduino Mega 2560, connects with Wi-Fi access points and read RSSI every second.
- RSSI monitored connecting to APs using three Wi-Fi channels namely 1, 6 and 11. Center frequencies 2412, 2437 and 2462 MHz, without encryption and passwords set.

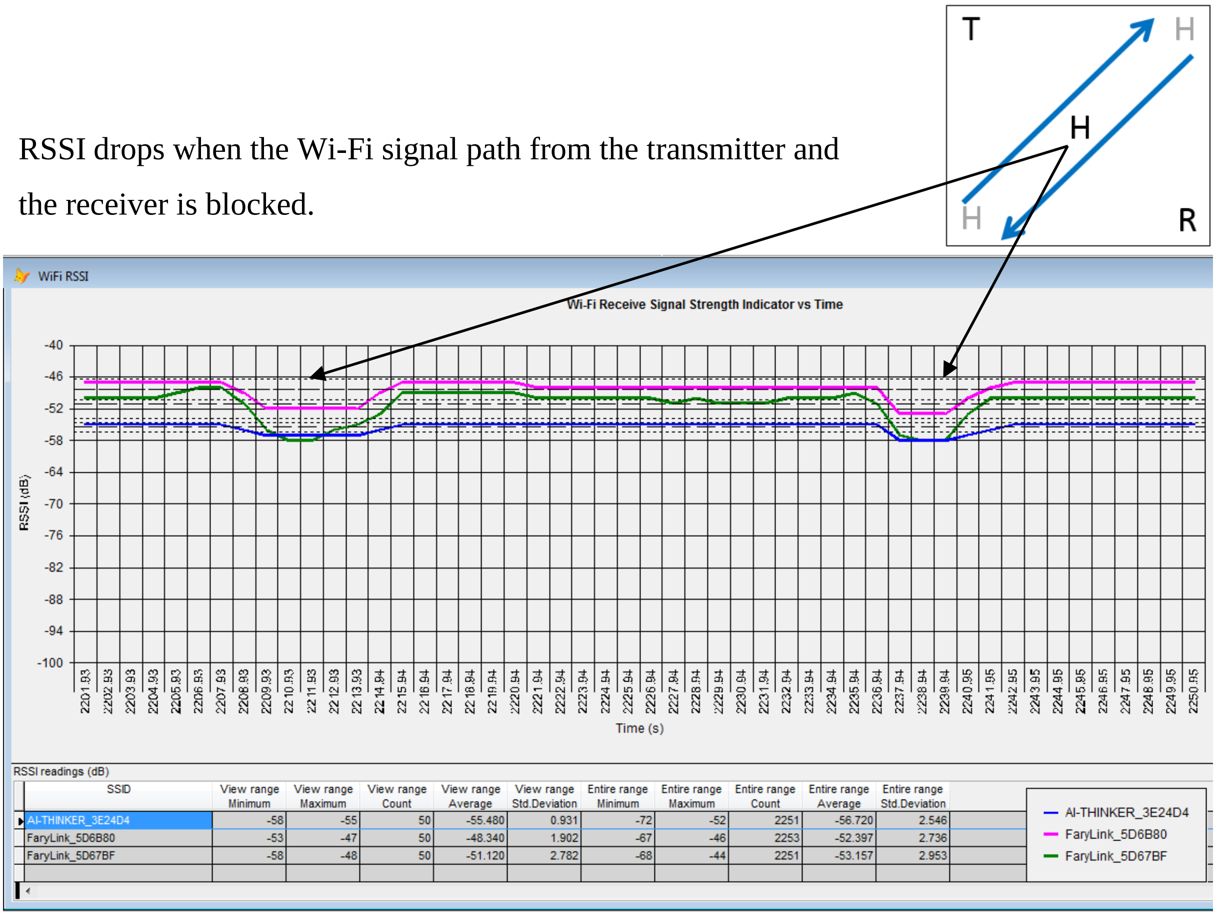
As mentioned above, testing were carried out in a square area where the length of a side is 4m. This was decided similar to as in section 4.2.1. In addition, to the hardware that was used which are for prototyping and its limitations, more space was required to monitor Wi-Fi signal strengths for human movements across a long distance. Also a square area was selected to easily position the transmitter, receiver and human in different locations.

Figure 4.16 shows a human moving from the Wi-Fi transmitter to the receiver. When the human is near the transmitter more Wi-Fi signals are attenuated as the signals are spread in a circular area from the transmitter resulting less signals reaching the receiver. Signals from the transmitter may reach the receiver indirectly, due to reflection caused by the surrounding walls and other smooth surfaces. These signals too are reduced by the human near and covering the transmitter. According to Figure 4.17 when the human is in the middle of the transmitter and the receiver, less signals attenuate, as the transmitter is open and more signals reach the receiver by signal reflection. Direct signals does not reach the receiver as the human is inline in between the transmitter and the receiver. But when the human is near the receiver more signals attenuate as it covers the direct signals and more of the reflected signals. Due to positioning of the transmitter, receiver and human, Figure 4.17 is not exactly the opposite of Figure 4.16 and may also have slight variations due to the quality of the hardware and all of the hardware modules not being from the same manufacturer. Observing Figure 4.16 and Figure 4.17 reveals that a human movement can be identified from and towards the transmitter and the receiver.



SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-57	-55	50	-55.300	0.580	-72	-52	1988	-56.950	2.573
FaryLink_5D6B80	-51	-46	50	-47.160	1.184	-67	-46	1990	-53.076	2.158
FaryLink_5D67BF	-56	-49	50	-50.980	1.597	-68	-44	1988	-53.474	2.935

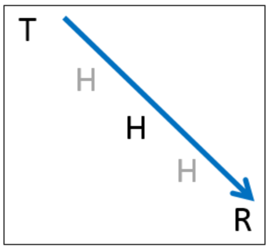
Figure 4.14: Human movement across the transmitter and the receiver

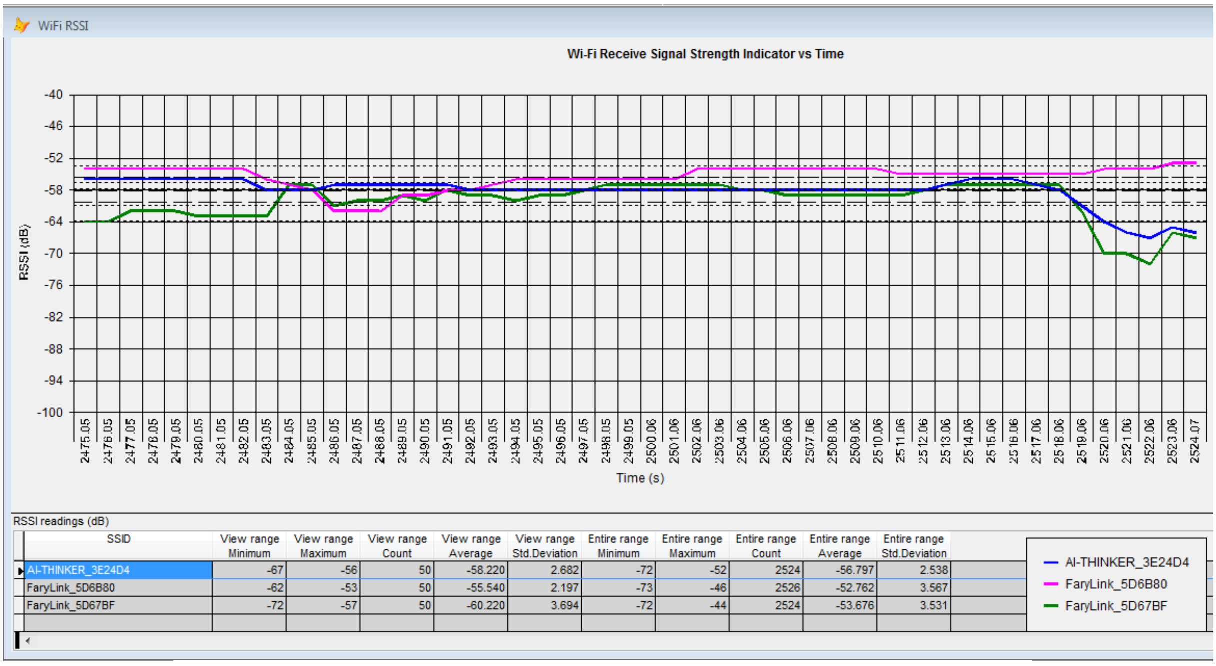


SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-58	-55	50	-55.480	0.931	-72	-52	2251	-56.720	2.546
FaryLink_5D6B80	-53	-47	50	-48.340	1.902	-67	-46	2253	-52.397	2.736
FaryLink_5D67BF	-58	-48	50	-51.120	2.782	-68	-44	2251	-53.157	2.953

Figure 4.15: Human movement across the transmitter and the receiver, diagonally

RSSI drop decreases when travelling from the transmitter to the receiver as less signals gets attenuated and transmitted signals reach the received from reflection as well.

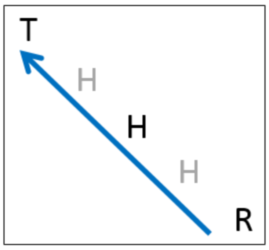


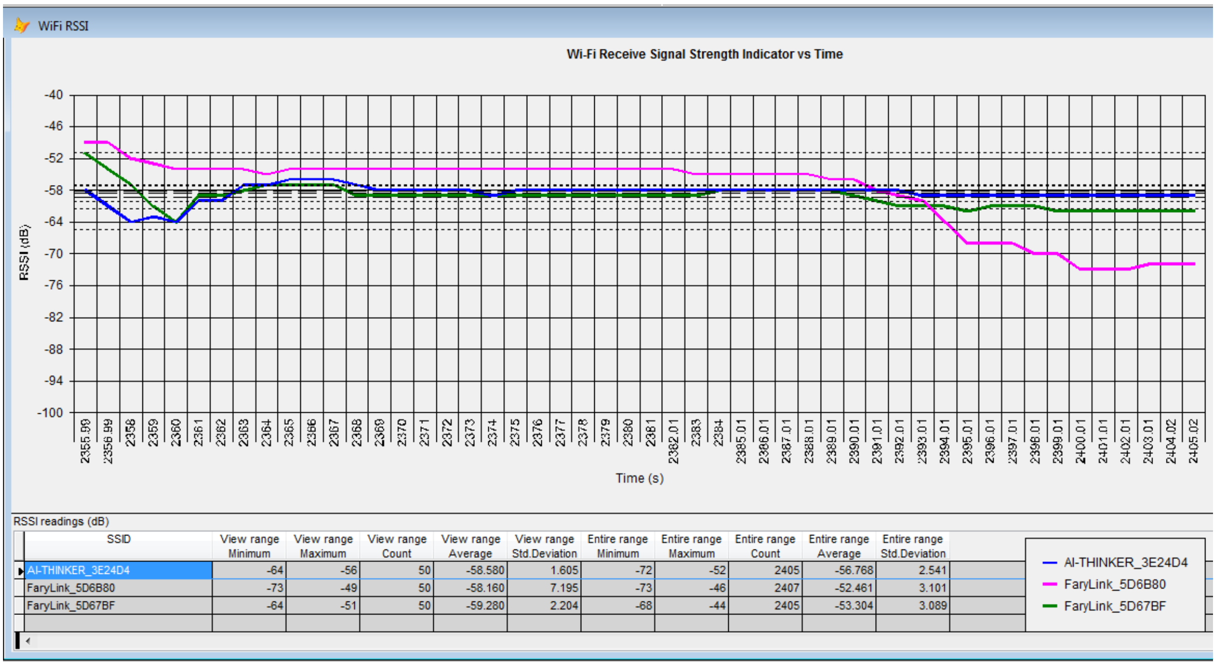


SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-67	-56	50	-58.220	2.682	-72	-52	2524	-56.797	2.538
FaryLink_5D6B80	-62	-53	50	-55.540	2.197	-73	-46	2526	-52.762	3.567
FaryLink_5D67BF	-72	-57	50	-60.220	3.694	-72	-44	2524	-53.676	3.531

Figure 4.16: Human movement from the transmitter towards the receiver

RSSI drop increases when travelling from the receiver to the transmitter as more signals gets attenuated and less of transmitted signals reach the receiver by reflection.





SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-64	-56	50	-58.580	1.605	-72	-52	2405	-56.768	2.541
FaryLink_5D6B80	-73	-49	50	-58.160	7.195	-73	-46	2407	-52.461	3.101
FaryLink_5D67BF	-64	-51	50	-59.280	2.204	-68	-44	2405	-53.304	3.089

Figure 4.17: Human movement from the receiver towards the transmitter

4.2.4 Experiments with RSSI through walls

Experiments were conducted similar to section 4.2 with relevant attributes. Some graphs are shown below in Figure 4.18 and Figure 4.19.

According to its results, it reveals that Wi-Fi signals penetrate through walls and RSS can be measured confirming the Wi-Fi signal path need not be in a line of sight.

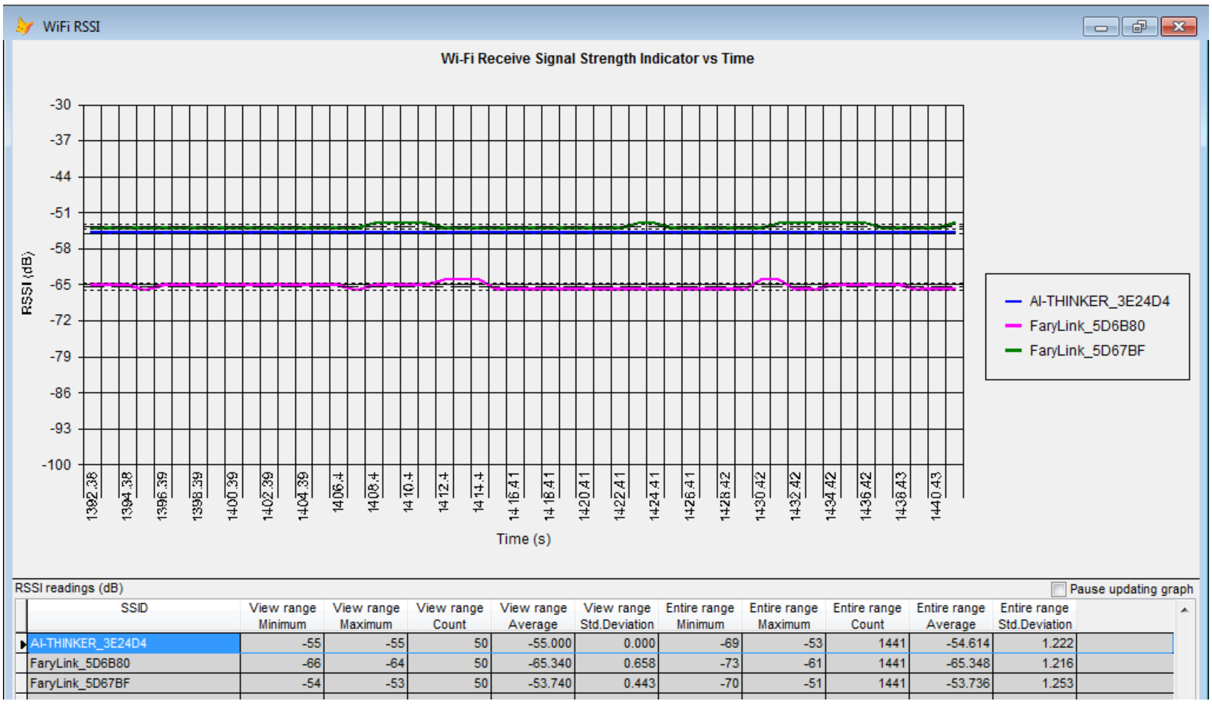


Figure 4.18: RSSI through a wall without disturbance

As per below Figure 4.19 a human crossed the signal path twice.

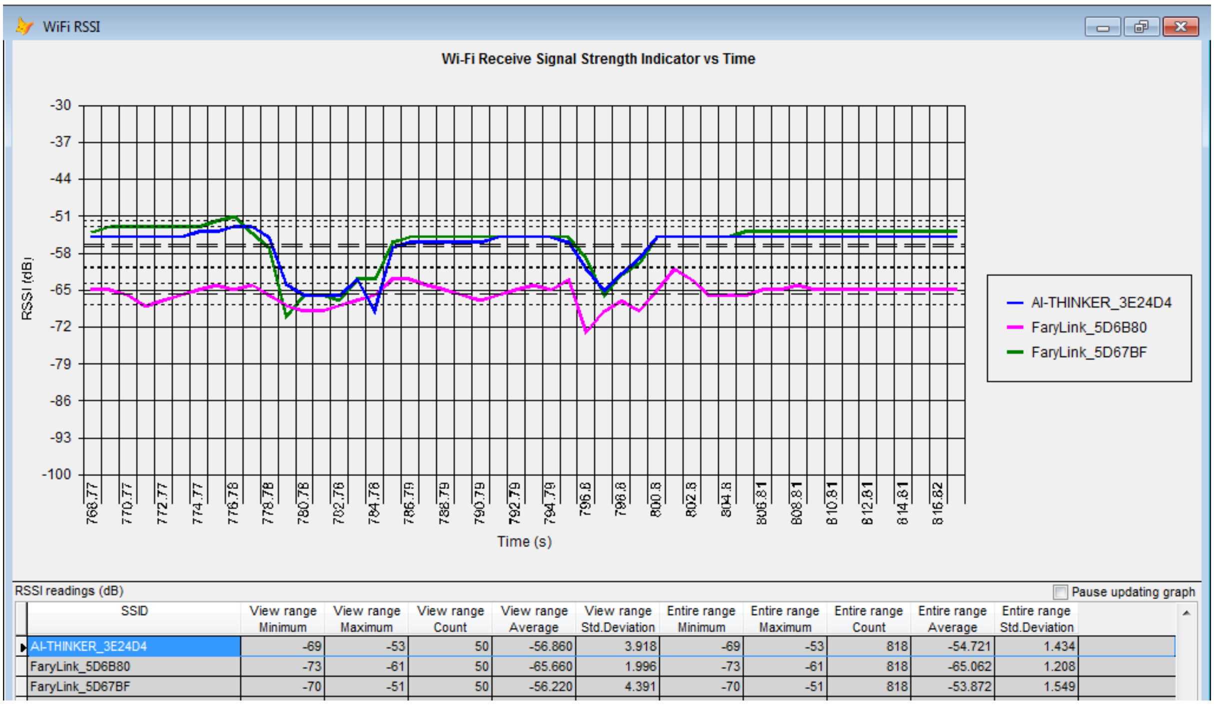


Figure 4.19: RSSI through a wall with human crossing the signal path

Chapter 5
Conclusion & Future Work

5.1 Achievements and contribution

It was with great effort and time that made it possible to complete this research and implementation with a working prototype to prove the expected results, within a period less than a year.

By re-visiting the research questions as defined in the introduction and by analysing the results obtained by the research, responses are as follows.

- Can RSSI be used to detect an object or human placed in between a Wi-Fi transmitter (access point) and a receiver (client) in a straight line?

  Yes it is possible.
  Placing a human cause RSSI to deviate more from the average, while other objects cause a slight deviation. The main reasons for the differences are,
    - Substances containing water highly absorb 2.4 GHz signals in which the Wi-Fi frequency range operates.
    - Objects with a higher mass volume attenuate the Wi-Fi signal strengths more.

- Can non overlapping Wi-Fi channels be used, for simultaneous transmission, monitor the RSSI of each for better results and improve accuracy of detecting human presence?

  Yes, it is possible.
  There could be disturbances to RSSI measurements due to environment conditions, instable operating power for the hardware and the low cost experimental hardware itself. These effects could be minimized by monitoring RSSI simultaneously on multiple Wi-Fi channels with calculations based on the standard deviation distribution. Three non-overlapping Wi-Fi channels were selected that do not affect the communication of each other, which improved the accuracy in RSSI and triggered an alarm identifying a human crossing the communication path.

5.2 Challenges faced

The challenges and difficulties encountered during the research and implementation of this project.

- Difficulty in finding the hardware components locally and delivery delays in purchasing them from overseas suppliers such as from eBay.
- Studying the functionality of Arduino hardware, IDE and the Wi-Fi modules.
- Finding relevant material and building prototypes to a certain quality were challenging, enthusiastic and time consuming.
- As these electronic modules are sensitive, a few of them malfunctioned as well.
- To interface simultaneously with multiple Wi-Fi modules, multiple communication ports were not available in Arduino Uno, which was used at the initial stage of the research. Although, Arduino SoftSerial library [40] was used it was unsuccessful as it internally uses normal IO digital pins for the serial communication and the maximum supported baud rate was 9600bps which is fairly show [44]. This also uses the microcontroller timers, caused overall delays for program functions and did not function properly, resulting inaccurate readings in RSSI.

  Therefore, it is ideal to address this issue in the hardware level and hence Arduino Mega 2560 which has three serial communication ports was used. Three serial ports are the maximum available in the Arduino hardware series.

5.3 Improvements and future work

- Use of Wi-Fi data transmission to communicate RSS readings, rather than using data cables.
- Notify the intruder alarm to registered mobile users by a call or SMS for immediate attention, interfacing a GSM module.
- Facilitate to ignore movements of humans carrying registered Wi-Fi devices such as mobiles and laptops, within the monitored area.
- Calculate and identify the positions of the obstacles within the monitored area which is ideal for areas separated with partitions or walls.
- Analyse and identify the differences in RSSI patterns for humans, pets and movements due to wind.

5.4 Lessons learnt

The knowledge gathered working with these hardware modules and software is an advantage for further developments. Also, completing this research project in a timely manner enforces planning, scheduling and working towards its finality.

Appendix A

Hardware and software specifications

Hardware specification of ESP8266-01 Wi-Fi module

Main specifications of ESP8266EX chip used in ESP8266-01 Wi-Fi module are as follows [18], [20], [45].

Categories	Items	Parameters
Wi-Fi	Standards	FCC/CE/TELEC/SRRC
	Protocols	802.11 b/g/n/e/i
	Frequency Range	2.4G ～ 2.5G (2400M ～ 2483.5M)
	Tx Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps)
		802.11 g: -75 dbm (54 Mbps)
		802.11 n: -72 dbm (MCS7)
	Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware	CPU	Tensilica L106 32-bit micro controller
	Peripheral Interface	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/ADC/PWM/LED Light & Button
	Operating Voltage	2.5V ~ 3.6V
	Operating Current	Average value: 80 mA
	Operating Temperature Range	-40°C ~ 125°C
	Storage Temperature Range	-40°C ~ 125°C
	Package Size	QFN32-pin (5 mm x 5 mm)
	External Interface	-
Software	Wi-Fi Mode	Station/SoftAP/SoftAP+Station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / Firmware and SDK for fast on-chip programming
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App

Table A.1: Specifications of ESP8266EX chip used in ESP8266-01 Wi-Fi module

Hardware specification of Arduino Mega 2560 board

Main specifications of Arduino Mega 2560 which is used to control and read Wi-Fi RSS from the ESP8266-01 module are as follows [24].

Items	Parameters
Brand	Arduino
Name	Mega 2560
Microcontroller (processor)	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 of which 15 provide PWM output
Analog Input Pins	16
Serial UART	4
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed, (CPU Speed)	16 MHz
LED built in pin	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g
USB socket	Regular

Table A.2: Specifications of the Arduino Mega 2560 board

Below is a diagram showing the pin connections of the Arduino Mega 2560 [21].

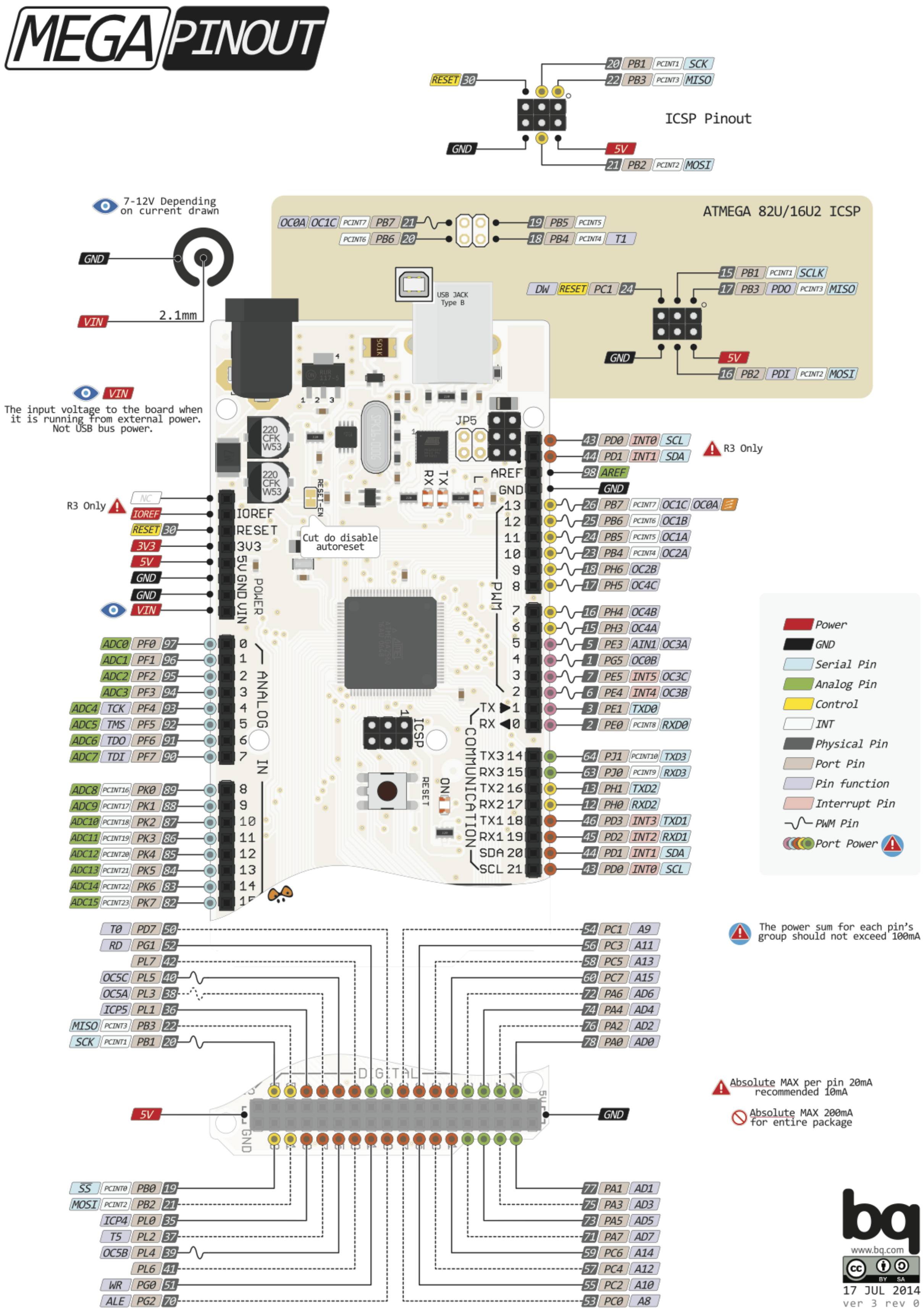


Figure A.1: Pin connections of the Arduino Mega 2560 board

AT commands of ESP8266-01 Wi-Fi module

Following are the commands which could be passed into the ESP8266-01 module [23]. All of these might not function as ESP8266-01 is the basic module selected for this research.

Commands	Description
AT	Tests AT startup.
AT+RST	Restarts the module.
AT+GMR	Checks version information.
AT+GSLP	Enters Deep-sleep mode.
ATE	Configures echoing of AT commands.
AT+RESTORE	Restores the factory default settings of the module.
AT+UART	UART configuration. [@deprecated]
AT+UART_CUR	The current UART configuration.
AT+UART_DEF	The default UART configuration, saved in flash.
AT+SLEEP	Configures the sleep modes.
AT+WAKEUPGPIO	Configures a GPIO to wake ESP8266 up from Light-sleep mode.
AT+RFPOWER	Sets the maximum value of the RF TX Power.
AT+RFVDD	Sets the RF TX Power according to VDD33.
AT+RFAUTOTRACE	Sets RF frequency offset trace.
AT+SYSRAM	Checks the available RAM size.
AT+SYSADC	Checks the ADC value.
AT+SYSIOSETCFG	Sets configuration of IO pins.
AT+SYSIOGETCFG	Gets configuration of IO pins.
AT+SYSGPIODIR	Configures the direction of GPIO.
AT+SYSGPIOWRITE	Configures the GPIO output level
AT+SYSGPIOREAD	Checks the GPIO input level.

Table A.3: Basic AT Commands

Commands	Description
AT+CWMODE	Sets the Wi-Fi mode (Station/AP/Station+AP). [@deprecated]
AT+CWMODE_CUR	Sets the Wi-Fi mode (Station/AP/Station+AP); configuration not saved in the flash.
AT+CWMODE_DEF	Sets the default Wi-Fi mode (Station/AP/Station+AP); configuration saved in the flash.
AT+CWJAP	Connect to an AP. [@deprecated]
AT+CWJAP_CUR	Connects to an AP; configuration not saved in the flash.
AT+CWJAP_DEF	Connects to an AP; configuration saved in the flash.
AT+CWLAPOPT	Sets the configuration of command AT+CWLAP.
AT+CWLAP	Lists available APs.
AT+CWQAP	Disconnects from an AP.
AT+CWSAP	Sets the configuration of the ESP8266 SoftAP. [@deprecated]
AT+CWSAP_CUR	Sets the current configuration of the ESP8266 SoftAP; configuration not saved in the flash.
AT+CWSAP_DEF	Sets the configuration of the ESP8266 SoftAP; configuration saved in the flash.
AT+CWLIF	Gets the Station IP to which the ESP8266 SoftAP is connected.

AT+CWDHCP	Enables/Disables DHCP. [@deprecated]
AT+CWDHCP_CUR	Enables/Disables DHCP; configuration not saved in the flash.
AT+CWDHCP_DEF	Enable/Disable DHCP; configuration saved in the flash.
AT+CWDHCPS_CUR	Sets the IP range of the DHCP server; configuration not saved in the flash.
AT+CWDHCPS_DEF	Sets the IP range of the DHCP server; configuration saved in the flash.
AT+CWAUTOCONN	Connects to an AP automatically on power-up.
AT+CIPSTAMAC	Sets the MAC address of the ESP8266 Station. [@deprecated]
AT+CIPSTAMAC_CUR	Sets the MAC address of the ESP8266 Station; configuration not saved in the flash.
AT+CIPSTAMAC_DEF	Sets the MAC address of ESP8266 station; configuration saved in the flash.
AT+CIPAPMAC	Sets the MAC address of the ESP8266 SoftAP. [@deprecated]
AT+CIPAPMAC_CUR	Sets the MAC address of the ESP8266 SoftAP; configuration not saved in the flash.
AT+CIPAPMAC_DEF	Sets the MAC address of the ESP8266 SoftAP; configuration saved in the flash.
AT+CIPSTA	Sets the IP address of the ESP8266 Station. [@deprecated]
AT+CIPSTA_CUR	Sets the IP address of the ESP8266 Station; configuration not saved in the flash.
AT+CIPSTA_DEF	Sets the IP address of the ESP8266 Station; configuration saved in the flash.
AT+CIPAP	Sets the IP address of ESP8266 SoftAP. [@deprecated]
AT+CIPAP_CUR	Sets the IP address of ESP8266 SoftAP; configuration not saved in the flash.
AT+CIPAP_DEF	Sets the IP address of ESP8266 SoftAP; configuration saved in the flash.
AT+CWSTARTSMART	Starts SmartConfig.
AT+CWSTOPSMART	Stops SmartConfig.
AT+CWSTARTDISCOVER	Enables the mode that ESP8266 can be found by WeChat.
AT+CWSTOPDISCOVER	Disables the mode that ESP8266 can be found by WeChat.
AT+WPS	Sets the WPS function.
AT+MDNS	Sets the MDNS function.
AT+CWHOSTNAME	Sets the host name of the ESP8266 Station.

Table A.4: Wi-Fi AT Commands

Command	Description
AT+CIPSTATUS	Gets the connection status
AT+CIPDOMAIN	DNS function
AT+CIPSTART	Establishes TCP connection, UDP transmission or SSL connection
AT+CIPSSLSIZE	Sets the size of SSL buffer
AT+CIPSEND	Sends data
AT+CIPSENDEX	Sends data when length of data is <length>, or when \0 appears in the data
AT+CIPSENDBUF	Writes data into TCP-send-buffer
AT+CIPBUFRESET	Resets the segment ID count
AT+CIPBUFSTATUS	Checks the status of TCP-send-buffer
AT+CIPCHECKSEQ	Checks if a specific segment is sent or not
AT+CIPCLOSE	Closes TCP/UDP/SSL connection
AT+CIFSR	Gets the local IP address
AT+CIPMUX	Configures the multiple connections mode
AT+CIPSERVER	Deletes/Creates a TCP server
AT+CIPMODE	Configures the transmission mode

AT+SAVETRANSLINK	Saves the transparent transmission link in the flash
AT+CIPSTO	Sets timeout when ESP8266 runs as TCP server
AT+PING	Ping packets
AT+CIUPDATE	Upgrades the software through network
AT+CIPDINFO	Shows remote IP and remote port with +IPD
AT+CIPSNTPCFG	Configures the time domain and SNTP server.
AT+CIPSNTPTIME	Queries the SNTP time.
AT+CIPDNS_CUR	Sets user-defined DNS servers; configuration not saved in the flash
AT+CIPDNS_DEF	Sets user-defined DNS servers; configuration saved in the flash

Table A.5: TCP/IP-Related AT Commands

Main functions and commonly used AT commands of ESP8266-01

Important and functions of the ESP8266-01 Wi-Fi module are as follows [22].

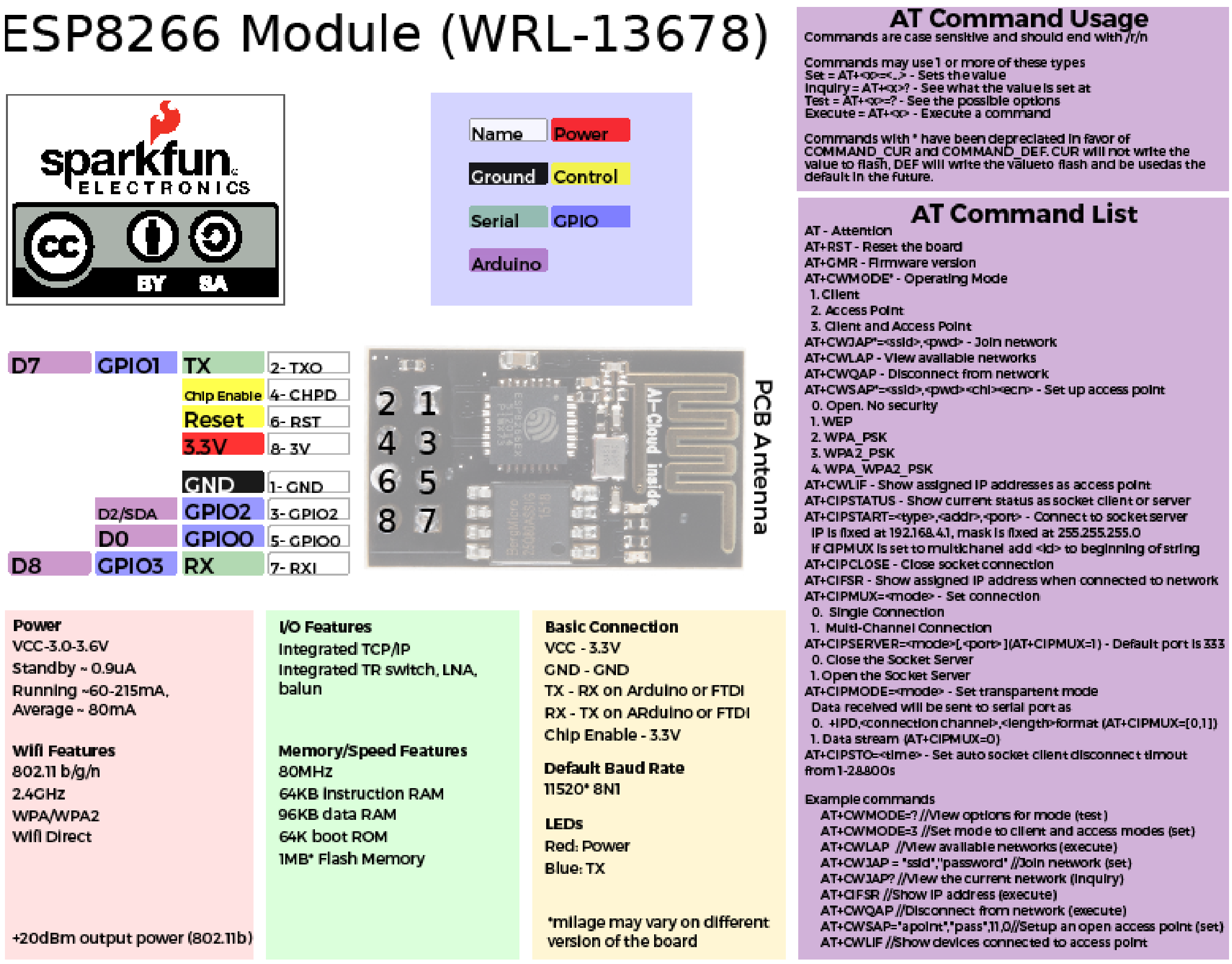


Figure A.2: Important and functions of the ESP8266-01 Wi-Fi module

Appendix B

Hardware components and software

Hardware components used to build the prototype

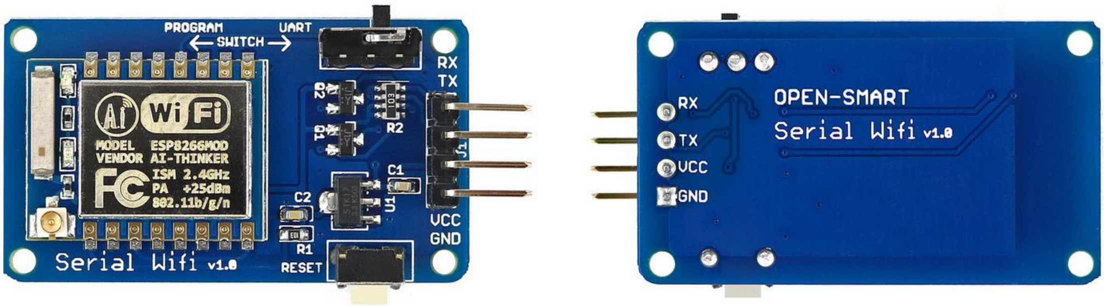


Figure B.1: ESP8266-07 Wi-Fi transceiver module

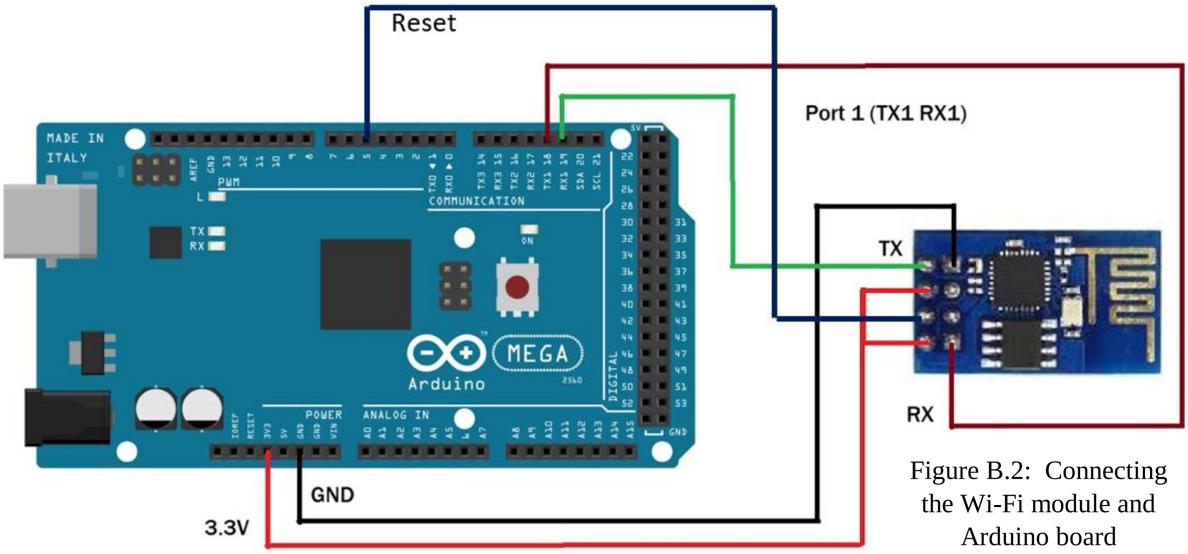


Figure B.2: Connecting the Wi-Fi module and Arduino board

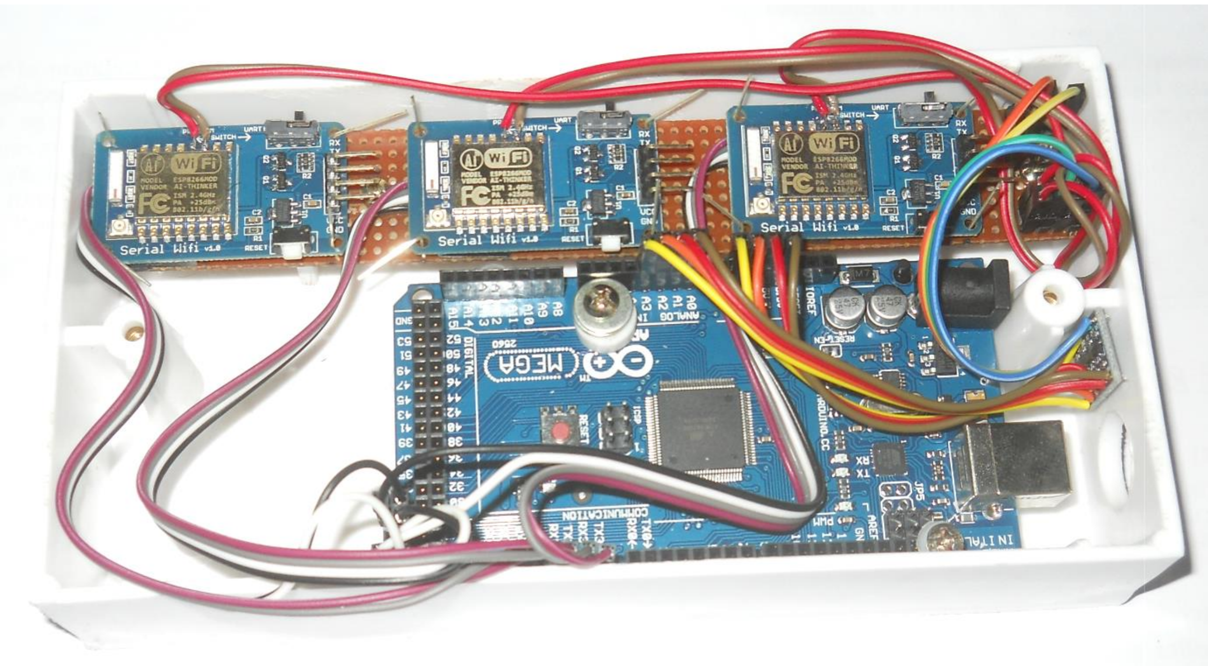


Figure B.3: Prototype built using ESP8266-07 and Arduino Mega

Hardware to write a program into ESP8266, 01 and 07

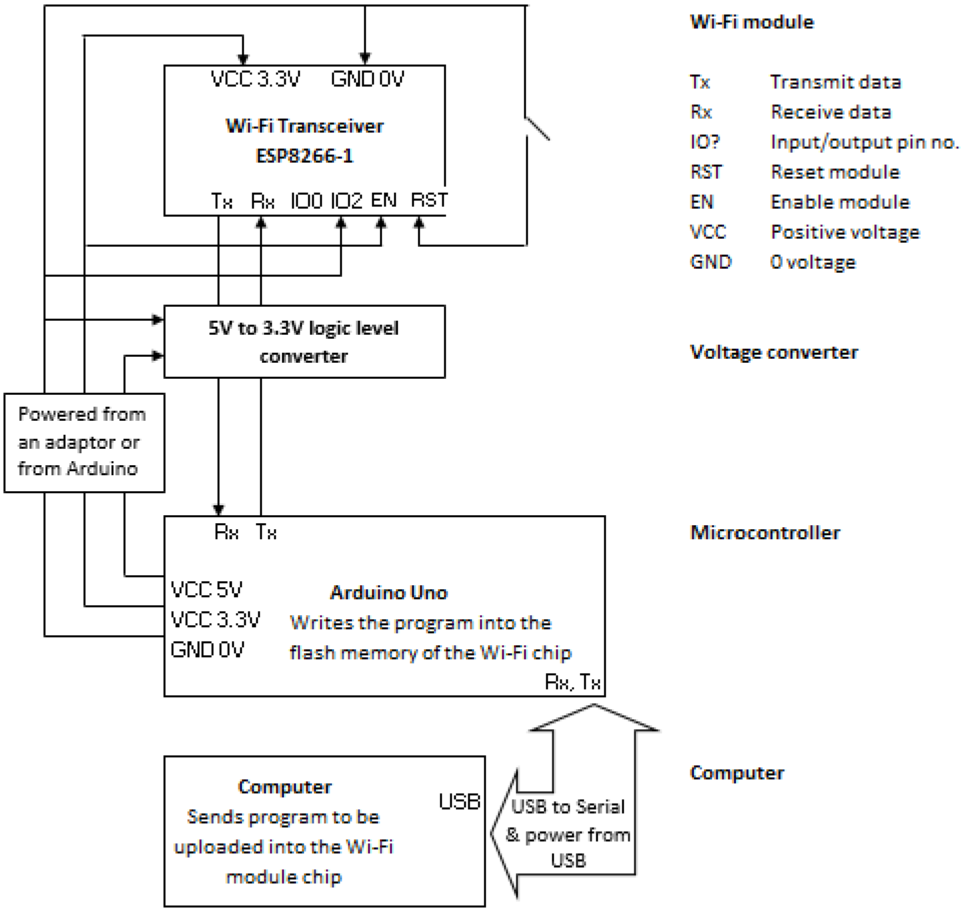


Figure B.4: Diagram to program the Wi-Fi module with Arduino

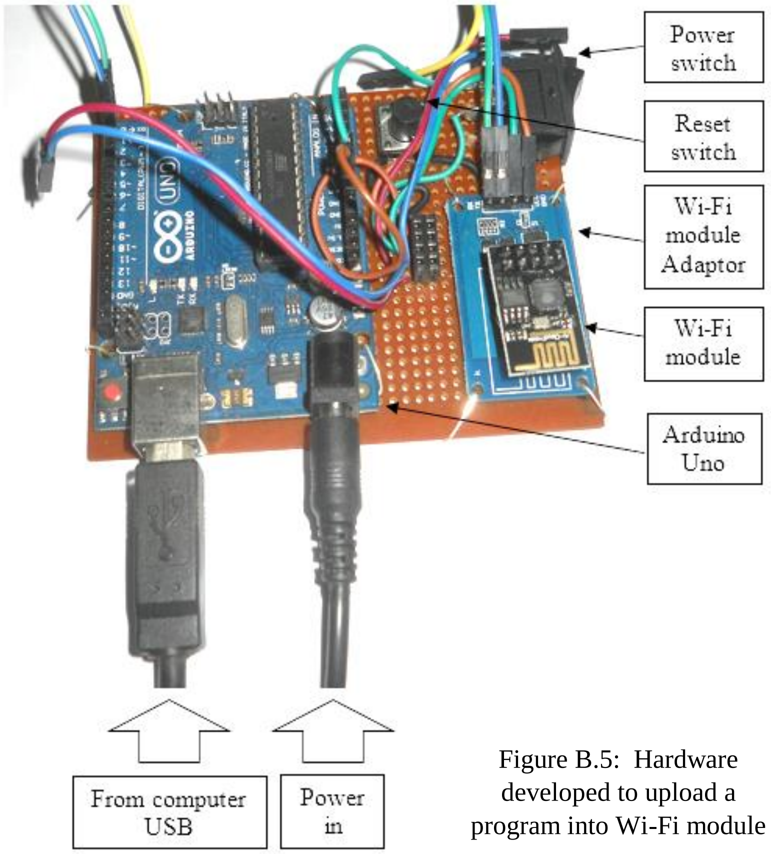


Figure B.5: Hardware developed to upload a program into Wi-Fi module

Programming the ESP8266-01 and ESP8266-07 modules

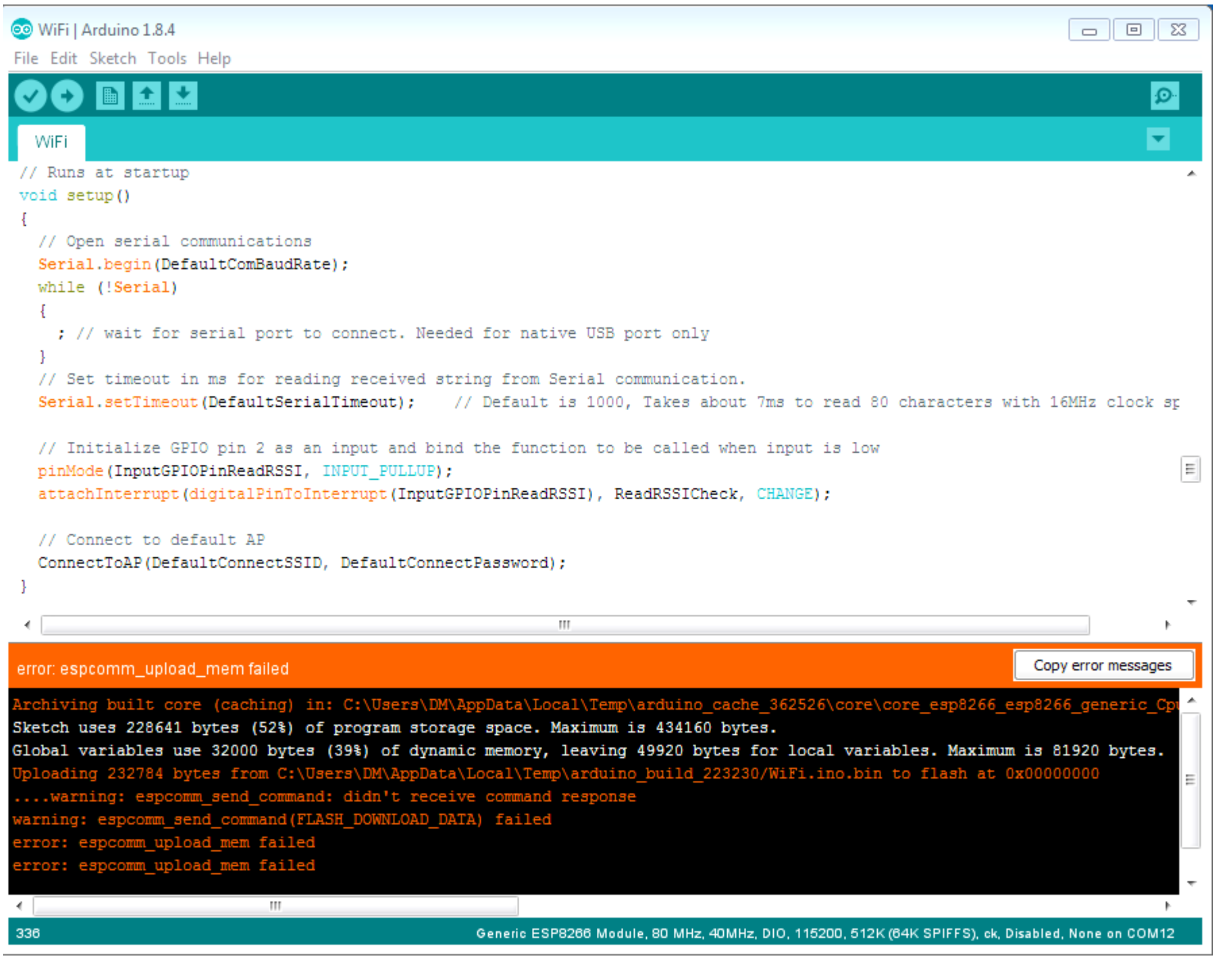


Figure B.6: Error writing into the ESP8266-01 program memory with Arduino ESP tool.exe

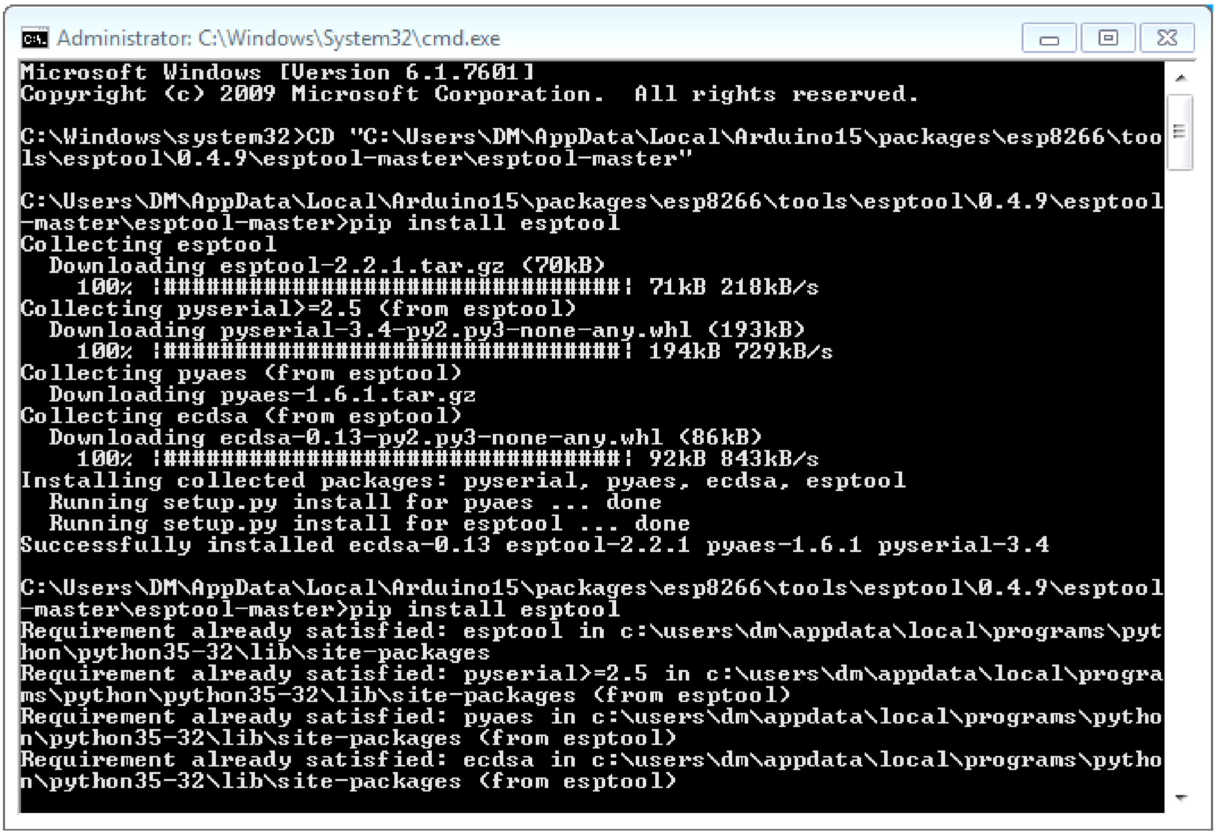


Figure B.7: Installing the Python script for programming the ESP8266-01

Specifying the esptool.py script as the uploading program rather than the Arduino default program, the esptool.exe

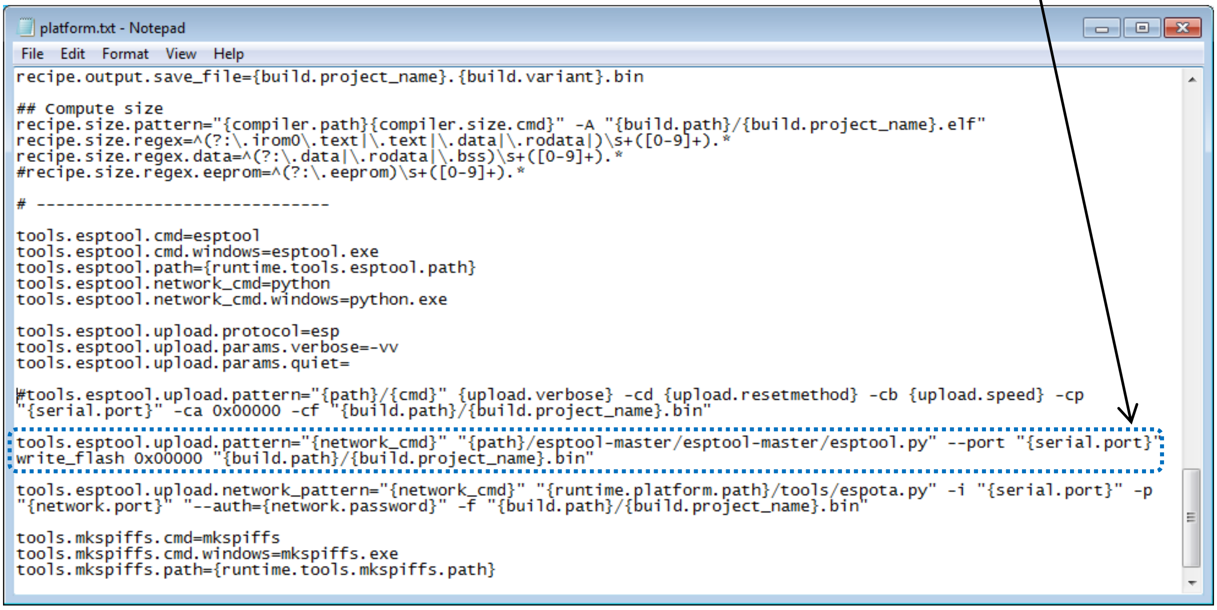


Figure B.8: Configuring Arduino IDE to use the Python script for programming ESP8266-01

Program successfully uploaded into the ESP8266 program memory with the esptool.py configured.

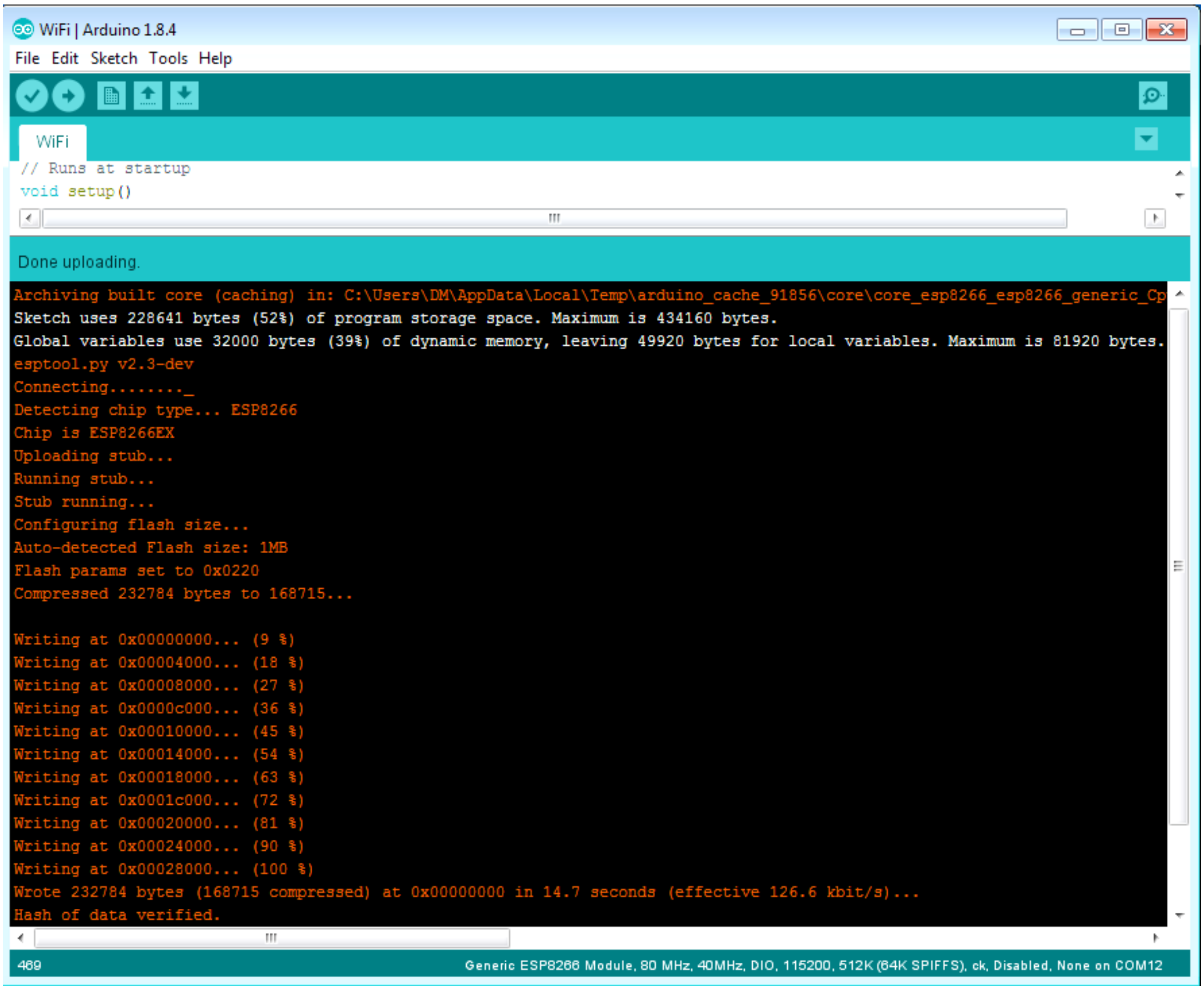


Figure B.9: Writing into the ESP8266-01 program memory with the python script

Appendix C

Source code listing

GUI for generating graphs

Some of the main coding areas for serial communication and generating graphs are as below.

- Reads the received data from the Wi-Fi module - ***oleCOM.OnComm()* event**

oleCOM.OnComm

Object: oleCOM Procedure: OnComm View Parent Code

```
*** ActiveX Control Event ***

LOCAL loControl, lnSearchIndex

IF THISFORM.nEnableReadData=2 AND THIS.CommEvent=2 AND THIS.InBufferCount>0
    THISFORM.cntReadWrite.cntRead.edtRead.Value = THISFORM.cntReadWrite.cntRead.edtRead.Value + THISFORM.Forn
    THISFORM.ExtractFromRead()
ENDIF

IF This.CommEvent > 0
    IF BETWEEN(This.CommEvent,comEvSend,comEvEOF)
        loControl = THISFORM.TxtLastEvent
    ELSE
        IF BETWEEN(This.CommEvent,comEventBreak,comEventDCB)
            loControl = THISFORM.TxtLastError
        ELSE
            loControl = null
        ENDIF
    ENDIF
    IF !ISNULL(loControl)
        lnSearchIndex = ASCAN(THISFORM.aCommEvents,TRANSFORM(This.CommEvent),1,-1,1,15)
        loControl.Value = THISFORM.aCommEvents[lnSearchIndex,2]
    ENDIF
ENDIF
```

Figure C.1: Source code reading text from serial communication

- Draws the graph with the two methods below ***FrmExcelGraph.DrawGraph()* method**

Object: FrmExcelGraph Procedure: drawgraph View Parent Code

```
** 14/07/2017

LOCAL lnCurrentWorkArea, lcLabels, lcRSSIFields, lcSSIDNo

lnCurrentWorkArea = SELECT(0)
lcLabels = ""
lcRSSIFields = ""
SELECT fSSIDInfoGraph
SCAN FOR Active=1 AND SSIDNo<=inMaxSSIDCount
    lcSSIDNo = PADL(fSSIDInfoGraph.SSIDNo,2,"0")
    lcLabels = lcLabels + RTRIM(fSSIDInfoGraph.SSID) + ","
    lcRSSIFields = lcRSSIFields + "SSID" + lcSSIDNo + ","
ENDSCAN
SELECT (lnCurrentWorkArea)

WITH THISFORM.oleSimpleChart
    .cAlias = "fRSSIInfoGraph"
    .cRowLabels = "ElapsedTm"
    .cLabels = RTRIM(lcLabels,1,",")
    .cData = RTRIM(lcRSSIFields,1,",")
    *.cFootnote = "Footnote"
    *.cTitle = "Title"
    *.BorderStyle = 0
    *.ChartType = 3
    *.lShowMarkers = .T.
    .ShowLegend = .T.
    .lIgnoreZero = .T.
    .cColours = "Blue, Magenta, Darkblue, Darkgreen, Brown"
    THIS.LockScreen = .T.
    .CreateChart()
    THIS.LockScreen = .F.
ENDWITH

GO TOP IN fSSIDInfoGraphGrid
```

Figure C.2: Source code for generating graphs

Updates the graph before it is displayed
***FrmExcelGraph.UpdateGraph()* method**

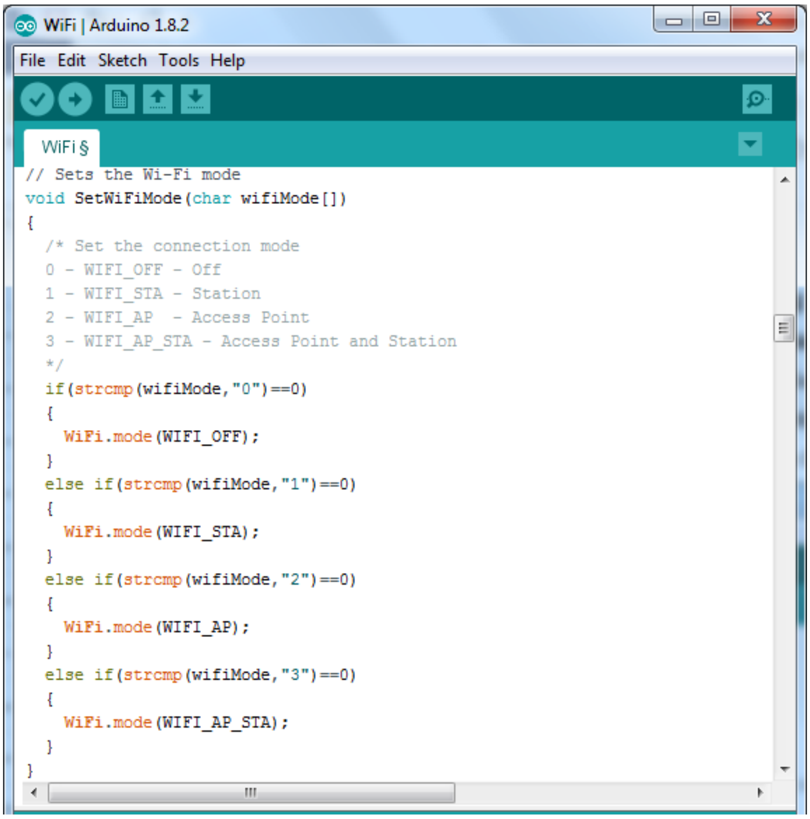
```
LPARAMETERS tcTextRead
* tcTextRead    -  Text read from the serial port      Compulsory

** 12/07/2017

** Note
** Since an entire block of text is not read at once and it may in parts,
** it is required to gather the text until a end block text is found.

LOCAL lnSeconds, llDrawGraph, lnBlockIncompleteTextIndex, lnBlockEndTextIndex, lcBockText,
lnSecondsToLog, lcBlockLinePrevious, ;
laBlockLines(1), lnBlockLineIndex, lcBlockLine, lcSeconds, lcRSSInfo, ;
lcEncription, lcSSID, lcRSSI, lcMac, lcChannel, lnRSSI, lnPreCount, lnNewCount, lnPreAvg, lnNewAvg,
lnPreStdDv, lnNewStdDv, lcSSIDNo, lnVisibleRecCount, ;
lnRSSIViewMax, lnRSSIViewMin, lnRSSIViewAvg, lnRSSIViewVar, lnViewStdDv, lcSecondsToLog, lnSeconds,
lnSecondsToLog, loCurrentInfo, lnCurrentWorkArea

*lnSeconds = SECONDS()

** Identifies the end of the block for the data read.
#DEFINE icBlockEndText      "OK"
#DEFINE icBlockIncompleteText  "busy p..."
#DEFINE inMaxAxisUnitis      20
#DEFINE inMaxSSIDCount      10

THIS.nViewXAxisUnits = inMaxAxisUnitis

llDrawGraph = .F.
THIS.cBlockText = THIS.cBlockText + tcTextRead

lnBlockIncompleteTextIndex = RAT(icBlockIncompleteText, THIS.cBlockText,1)
IF lnBlockIncompleteTextIndex > 0
  THIS.cBlockText = SUBSTR(THIS.cBlockText, lnBlockIncompleteTextIndex+LEN(icBlockIncompleteText))
ENDIF

lnBlockEndTextIndex = RAT(icBlockEndText, THIS.cBlockText,1)
IF lnBlockEndTextIndex > 0
  lcBockText = SUBSTR(THIS.cBlockText, 1, lnBlockEndTextIndex-1)
  THIS.cBlockText = SUBSTR(THIS.cBlockText, lnBlockEndTextIndex+LEN(icBlockEndText))
  IF !EMPTY(lcBockText)
    lnSecondsToLog = 0
    lcBlockLinePrevious = ""

    FOR lnBlockLineIndex = 1 TO ALINES(laBlockLines, lcBockText, 5)
      lcBlockLine = laBlockLines(lnBlockLineIndex)
      IF EMPTY(lcBlockLine)
        LOOP
      ENDIF

      lcSeconds = STREXTRACT(lcBlockLine, "Seconds", icCRLF, 1, 3)
      IF EMPTY(lcSeconds)
        lcBlockLine = lcBlockLinePrevious + lcBlockLine
        lcRSSInfo = STREXTRACT(lcBlockLine, "+CWLAP:(", ")", 1, 1)
        IF EMPTY(lcRSSInfo)
          lcBlockLinePrevious = lcBlockLine
        ELSE
          lcEncription = ALLTRIM(STREXTRACT(lcRSSInfo, "", ",", 1, 1))
          lcSSID = ALLTRIM(STREXTRACT(lcRSSInfo, ",", ",", 1, 1))
          lcRSSI = ALLTRIM(STREXTRACT(lcRSSInfo, ",", ",", 2, 1))
          lcMac = ALLTRIM(STREXTRACT(lcRSSInfo, ",", ",", 3, 1))
          lcChannel = ALLTRIM(STREXTRACT(lcRSSInfo, ",", ",", 4, 3))
          lcBlockLinePrevious = ""
          lnRSSI = VAL(lcRSSI)

          INSERT INTO fRSSIInfoExcel (DateNTime, ElapsedTm, SSID, MacAddress, RSSI, Channel,
EncryptMd);
          VALUES(DATETIME(), lnSecondsToLog, &lcSSID, &lcMac, lnRSSI, VAL(lcChannel),
VAL(lcEncription))

          IF INDEXSEEK(&lcSSID,.T.,"fSSIDInfoGraph", "SSID")
            lnPreCount = fSSIDInfoGraph.AllCount
            lnNewCount = lnPreCount + 1
            lnPreAvg = fSSIDInfoGraph.AllAvg
            lnNewAvg = (lnPreAvg * lnPreCount + lnRSSI) / lnNewCount
            lnPreStdDv = fSSIDInfoGraph.AllStdDv
            lnNewStdDv = SQRT(( (lnPreStdDv^2)*MAX(lnPreCount-1,0) + (((lnPreAvg-
lnNewAvg)^2)*lnPreCount) + (lnRSSI-lnNewAvg)^2) / MAX(lnPreCount,1))
            ** http://www.statisticshowto.com/what-is-standard-deviation/
            **
https://www.researchgate.net/publication/4364297_Using_RSSI_value_for_distance_estimation_in_wireles
s_sensor_networks_based_on_ZigBee
```

```
            ** New SD = [[(pre SD)^2 * (pre count-1) + (pre avg - cur avg)^2 * (pre count) + (cur
val - cur avg)^2] / (pre count)]
            REPLACE Active WITH 1, AllMax WITH MAX(AllMax,lnRSSI), AllMin WITH MIN(AllMin,lnRSSI),
AllAvg WITH lnNewAvg, AllStdDv WITH lnNewStdDv, AllCount WITH lnNewCount IN fSSIDInfoGraph
          ELSE
            INSERT INTO fSSIDInfoGraph (SSID, Active, AllMax, AllMin, AllAvg, AllStdDv, AllCount)
VALUES (&lcSSID, 1, lnRSSI, lnRSSI, lnRSSI, 0, 1)
          ENDIF

          IF fSSIDInfoGraph.SSIDNo <= inMaxSSIDCount
            lcSSIDNo = PADL(fSSIDInfoGraph.SSIDNo,2,"0")

            *IF !INDEXSEEK(STR(THIS.nCurrentSessionNo)+STR(lnSecondsToLog,10,3), .F.,
"fRSSIInfoGraph", "SesnElpdTm")
              *lnCurrentWorkArea = SELECT(0)
              *SELECT fRSSIInfoGraph
              *GO BOTTOM IN fRSSIInfoGraph
              *SCATTER FIELDS LIKE SSIS*, DateNTime, ElapsedTm, GrphActive, SessionNo MEMO NAME
loCurrentInfo
              *SELECT (lnCurrentWorkArea)
              loCurrentInfo = THIS.oRSSIInfoGraphEmpty

              IF !INDEXSEEK(STR(THIS.nCurrentSessionNo)+STR(lnSecondsToLog,10,3), .T.,
"fRSSIInfoGraph", "SesnElpdTm")
                loCurrentInfo.DateNTime = DATETIME()
                loCurrentInfo.ElapsedTm = lnSecondsToLog
                loCurrentInfo.GrphActive = 1
                loCurrentInfo.SessionNo = THIS.nCurrentSessionNo

                INSERT INTO fRSSIInfoGraph FROM NAME loCurrentInfo
                *INSERT INTO fRSSIInfoGraph (DateNTime, ElapsedTm, GrphActive, SessionNo) VALUES
(DATETIME(), lnSecondsToLog, 1, THIS.nCurrentSessionNo)

                **CALCULATE CNT() TO lnVisibleRecCount IN fRSSIInfoGraph
                **IF lnVisibleRecCount > inMaxAxisUnitis
                **  REPLACE UpdCount WITH 0 IN fTmpUpdate
                **  REPLACE ALL GrphActive WITH 0, fTmpUpdate.UpdCount WITH fTmpUpdate.UpdCount+1
WHILE fTmpUpdate.UpdCount<lnVisibleRecCount-inMaxAxisUnitis IN fRSSIInfoGraph
                **ENDIF
                **GO BOTTOM IN fRSSIInfoGraph

                ** Alternative for the above
                IF fRSSIInfoGraph.RSSINo - THIS.nCurrentRSSINo > inMaxAxisUnitis
                  GO TOP IN fRSSIInfoGraph
                  REPLACE GrphActive WITH 0 IN fRSSIInfoGraph
                  GO BOTTOM IN fRSSIInfoGraph
                ENDIF
              ENDIF
            *ENDIF

            REPLACE SSID&lcSSIDNo WITH lnRSSI IN fRSSIInfoGraph

            CALCULATE MAX(SSID&lcSSIDNo), MIN(SSID&lcSSIDNo), CNT(), AVG(SSID&lcSSIDNo),
VAR(SSID&lcSSIDNo) FOR GrphActive=1 TO lnRSSIViewMax, lnRSSIViewMin, lnVisibleRecCount,
lnRSSIViewAvg, lnRSSIViewVar IN fRSSIInfoGraph
            IF lnVisibleRecCount > 0
              lnViewStdDv = SQRT((lnRSSIViewVar*lnVisibleRecCount)/MAX(lnVisibleRecCount-1,1))
              REPLACE ViewMax WITH lnRSSIViewMax, ViewMin WITH lnRSSIViewMin, ViewAvg WITH
lnRSSIViewAvg, ViewStdDv WITH lnViewStdDv, ViewCount WITH lnVisibleRecCount IN fSSIDInfoGraph
            ENDIF

            llDrawGraph = .T.
          ENDIF
        ENDIF
      ELSE
        lcSecondsToLog = ALLTRIM(lcSeconds)
        lnSeconds = &lcSecondsToLog
        IF THIS.nStartSeconds = 0
          THIS.nStartSeconds = lnSeconds
        ENDIF
        lnSecondsToLog = lnSeconds - THIS.nStartSeconds
      ENDIF
    ENDFOR
  ENDIF
ENDIF

IF llDrawGraph
  THIS.DrawGraph()
ENDIF

*WAIT WINDOW "Update time: "+TRANSFORM(SECONDS()-lnSeconds)+" s" NOWAIT
```

Figure C.3: Source code extracts form the developed GUI to update the graph

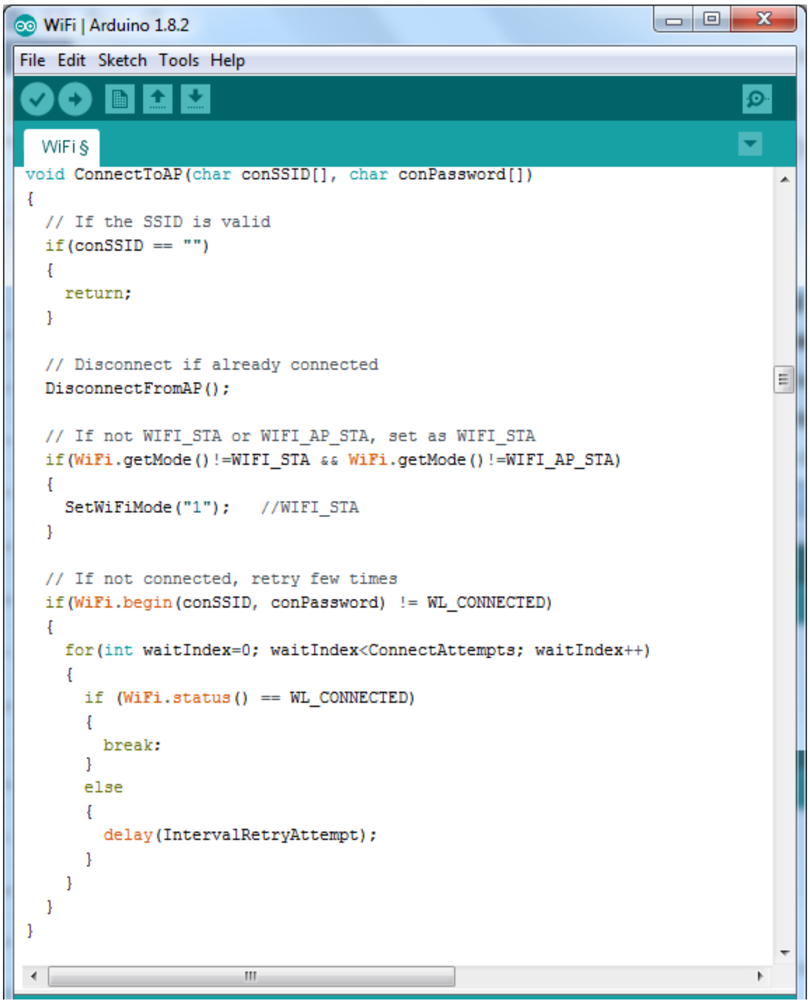
Program saved into the Wi-Fi module

Following in Figure C.4 and Figure C.5 are some extracts of its source code.




```
// Sets the Wi-Fi mode
void SetWiFiMode(char wifiMode[])
{
  /* Set the connection mode
  0 - WIFI_OFF - Off
  1 - WIFI_STA - Station
  2 - WIFI_AP  - Access Point
  3 - WIFI_AP_STA - Access Point and Station
  */
  if(strcmp(wifiMode,"0")==0)
  {
    WiFi.mode(WIFI_OFF);
  }
  else if(strcmp(wifiMode,"1")==0)
  {
    WiFi.mode(WIFI_STA);
  }
  else if(strcmp(wifiMode,"2")==0)
  {
    WiFi.mode(WIFI_AP);
  }
  else if(strcmp(wifiMode,"3")==0)
  {
    WiFi.mode(WIFI_AP_STA);
  }
}
```

Figure C.4: Source code extract to set the Wi-Fi mode




```
void ConnectToAP(char conSSID[], char conPassword[])
{
  // If the SSID is valid
  if(conSSID == "")
  {
    return;
  }

  // Disconnect if already connected
  DisconnectFromAP();

  // If not WIFI_STA or WIFI_AP_STA, set as WIFI_STA
  if(WiFi.getMode()!=WIFI_STA && WiFi.getMode()!=WIFI_AP_STA)
  {
    SetWiFiMode("1");   //WIFI_STA
  }

  // If not connected, retry few times
  if(WiFi.begin(conSSID, conPassword) != WL_CONNECTED)
  {
    for(int waitIndex=0; waitIndex<ConnectAttempts; waitIndex++)
    {
      if (WiFi.status() == WL_CONNECTED)
      {
        break;
      }
      else
      {
        delay(IntervalRetryAttempt);
      }
    }
  }
}
```

Figure C.5: Source code extract to connect to an AP

Program saved into Arduino Mega to control the Wi-Fi modules

Following are some source code extracts.

Main program

```
// Runs at startup to initializations
void setup()
{
  // Open serial communications
  for(int indexSerialPort=0;indexSerialPort<SerialPortCount;indexSerialPort++)
  {
    serialPorts[indexSerialPort]->begin(DefaultComBaudRate);
  }
  while (!Serial)
  {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // Set timeout in ms for reading received string from Serial communication.
  // Default is 1000, Takes about 7ms to read 80 characters with 16MHz clock speed
  for(int indexSerialPort=0;indexSerialPort<SerialPortCount;indexSerialPort++)
  {
    serialPorts[indexSerialPort]->setTimeout(DefaultSerialTimeout);
  }

  // Initialize pins as outputs
  pinMode(PinReadRSSIFromModules, OUTPUT);
  pinMode(PinGeneralOutputModules, OUTPUT);
  pinMode(PinResetModules, OUTPUT);

  // Sets the outputs to high
  digitalWrite(PinReadRSSIFromModules, HIGH);
  digitalWrite(PinGeneralOutputModules, HIGH);
  digitalWrite(PinResetModules, HIGH);

  // Connect each Wi-Fi module to a transmitter(AP)
  // 1;CONNECT:AI-THINKER_3E24D4:
  // 2;CONNECT:FaryLink_5D67BF:
  // 3;CONNECT:FaryLink_5D6B80:
  Serial1.println("CONNECT:MK16i:773430187");
  //Serial1.println("CONNECT:AI-THINKER_3E24D4:");
  delay(10000);
  Serial2.println("CONNECT:FaryLink_5D67BF:");
  delay(10000);
  Serial3.println("CONNECT:FaryLink_5D6B80:");
  delay(10000);

  UpdateDefaultRSSIQueue();
  InitializeLEDStatus();
}
```

Reading RSSI from each Wi-Fi module

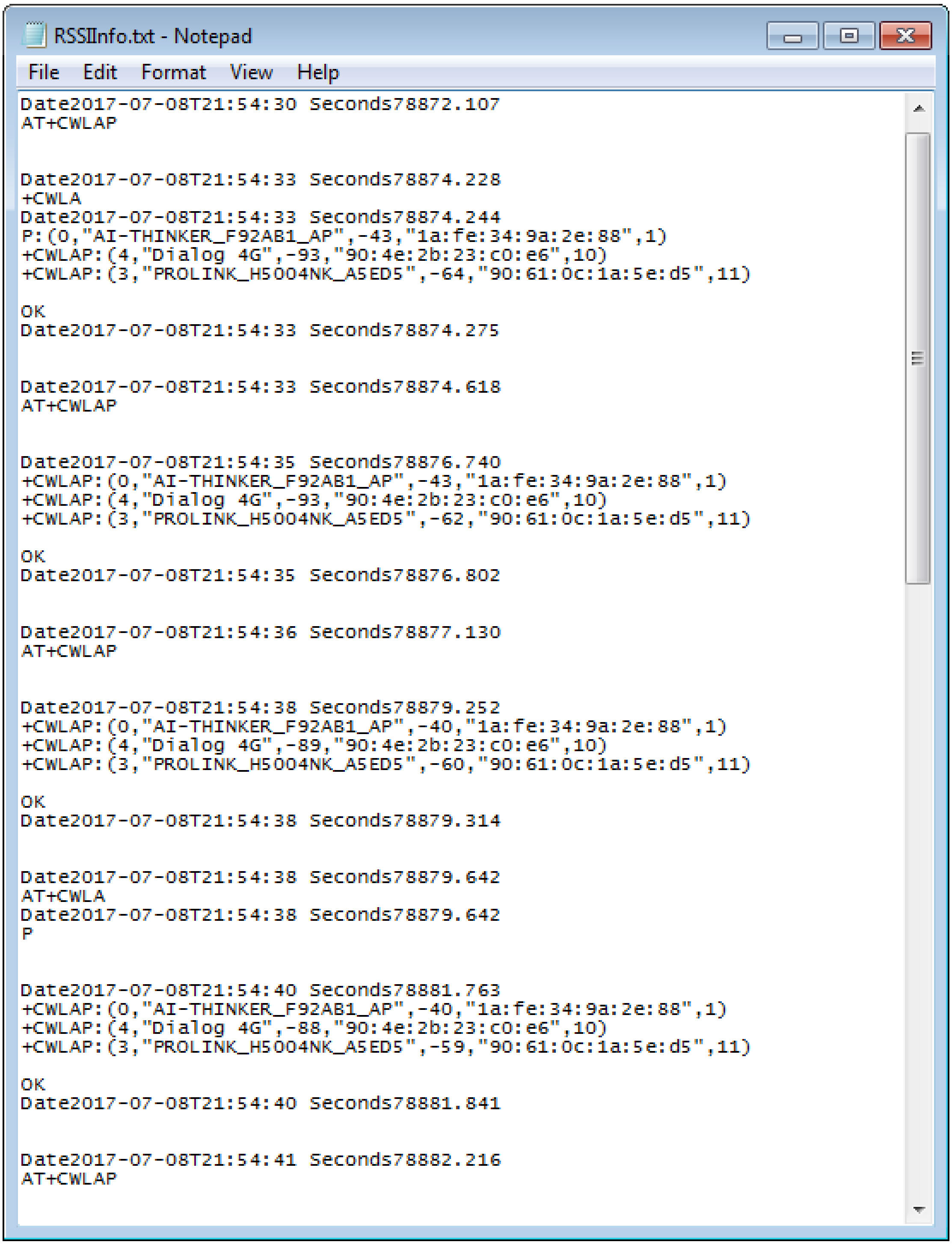
```
// Reads RSSI from modules and send into the virtual serial comm port.
void ReadSerialTextFromModules()
{
  for(int indexSerialPort=1;indexSerialPort<SerialPortCount;indexSerialPort++)
  {
    if (serialPorts[indexSerialPort]->available() > 0)
    {
      // Read input text
      char readBuffer[ReadBufferSize];
      memset(readBuffer, InputNewLine, ReadBufferSize);  // Clear contents of the buffer
      int readByteCount = serialPorts[indexSerialPort]->readBytesUntil(InputNewLine, readBuffer,
ReadBufferSize);
      if(readBuffer[readByteCount]==InputNewLine)        // Length is returned with the new line
      {
        readBuffer[readByteCount-2] = InputNewLine;
      }

      // If there is input text
      if(readByteCount > 0)
      {
          Serial.println(readBuffer);
          CalculateStatistics(readBuffer, indexSerialPort);
      }
    }
  }
  ValidateAndTriggerAlarm();
}
```

Appendix D

Data gathered

Raw text read from serial communication

Sample of the raw text gathered form the Wi-Fi module with the channel scanning method are as follows in Figure D.1.




```
Date2017-07-08T21:54:30 Seconds78872.107
AT+CWLAP


Date2017-07-08T21:54:33 Seconds78874.228
+CWLA
Date2017-07-08T21:54:33 Seconds78874.244
P:(0,"AI-THINKER_F92AB1_AP",-43,"1a:fe:34:9a:2e:88",1)
+CWLAP:(4,"Dialog 4G",-93,"90:4e:2b:23:c0:e6",10)
+CWLAP:(3,"PROLINK_H5004NK_A5ED5",-64,"90:61:0c:1a:5e:d5",11)

OK
Date2017-07-08T21:54:33 Seconds78874.275


Date2017-07-08T21:54:33 Seconds78874.618
AT+CWLAP


Date2017-07-08T21:54:35 Seconds78876.740
+CWLAP:(0,"AI-THINKER_F92AB1_AP",-43,"1a:fe:34:9a:2e:88",1)
+CWLAP:(4,"Dialog 4G",-93,"90:4e:2b:23:c0:e6",10)
+CWLAP:(3,"PROLINK_H5004NK_A5ED5",-62,"90:61:0c:1a:5e:d5",11)

OK
Date2017-07-08T21:54:35 Seconds78876.802


Date2017-07-08T21:54:36 Seconds78877.130
AT+CWLAP


Date2017-07-08T21:54:38 Seconds78879.252
+CWLAP:(0,"AI-THINKER_F92AB1_AP",-40,"1a:fe:34:9a:2e:88",1)
+CWLAP:(4,"Dialog 4G",-89,"90:4e:2b:23:c0:e6",10)
+CWLAP:(3,"PROLINK_H5004NK_A5ED5",-60,"90:61:0c:1a:5e:d5",11)

OK
Date2017-07-08T21:54:38 Seconds78879.314


Date2017-07-08T21:54:38 Seconds78879.642
AT+CWLA
Date2017-07-08T21:54:38 Seconds78879.642
P


Date2017-07-08T21:54:40 Seconds78881.763
+CWLAP:(0,"AI-THINKER_F92AB1_AP",-40,"1a:fe:34:9a:2e:88",1)
+CWLAP:(4,"Dialog 4G",-88,"90:4e:2b:23:c0:e6",10)
+CWLAP:(3,"PROLINK_H5004NK_A5ED5",-59,"90:61:0c:1a:5e:d5",11)

OK
Date2017-07-08T21:54:40 Seconds78881.841


Date2017-07-08T21:54:41 Seconds78882.216
AT+CWLAP
```

Figure D.1: Sample raw RSSI data gathered with channel scanning

Read data formatted and stored in a database

Sample of the raw data gathered and stored in a database to generate the graphs are as follows in Table D.1.

datentime	elapsedtm	ssid	macaddress	rssi	channel	encryptmd
16/09/2017 21:26:57	30.701	Dialog 4G	90:4e:2b:23:c0:e6	-85	6	4
16/09/2017 21:27:01	34.289	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-49	1	0
16/09/2017 21:27:01	34.289	ZTE	cc:1a:fa:96:82:94	-93	1	3
16/09/2017 21:27:01	34.289	Dialog 4G	90:4e:2b:23:c0:e6	-88	6	4
16/09/2017 21:27:04	37.892	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-50	1	0
16/09/2017 21:27:04	37.892	ZTE	cc:1a:fa:96:82:94	-94	1	3
16/09/2017 21:27:04	37.892	Dialog 4G	90:4e:2b:23:c0:e6	-86	6	4
16/09/2017 21:27:08	41.496	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-50	1	0
16/09/2017 21:27:08	41.496	ZTE	cc:1a:fa:96:82:94	-94	1	3
16/09/2017 21:27:08	41.496	Dialog 4G	90:4e:2b:23:c0:e6	-85	6	4
16/09/2017 21:27:11	45.084	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-50	1	0
16/09/2017 21:27:11	45.084	ZTE	cc:1a:fa:96:82:94	-91	1	3
16/09/2017 21:27:11	45.084	Dialog 4G	90:4e:2b:23:c0:e6	-86	6	4
16/09/2017 21:27:15	48.688	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-50	1	0
16/09/2017 21:27:15	48.688	ZTE	cc:1a:fa:96:82:94	-94	1	3
16/09/2017 21:27:15	48.688	Dialog 4G	90:4e:2b:23:c0:e6	-87	6	4
16/09/2017 21:27:19	52.26	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-50	1	0
16/09/2017 21:27:19	52.26	ZTE	cc:1a:fa:96:82:94	-93	1	3
16/09/2017 21:27:19	52.26	Dialog4G	90:4e:2b:23:c0:e6	-84	6	4
16/09/2017 21:27:22	55.864	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-50	1	0
16/09/2017 21:27:22	55.864	ZTE	cc:1a:fa:96:82:94	-94	1	3
16/09/2017 21:27:22	55.864	Dialog 4G	90:4e:2b:23:c0:e6	-86	6	4
16/09/2017 21:27:26	59.467	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-49	1	0
16/09/2017 21:27:26	59.467	ZTE	cc:1a:fa:96:82:94	-94	1	3
16/09/2017 21:27:26	59.467	Dialog 4G	90:4e:2b:23:c0:e6	-85	6	4
16/09/2017 21:27:33	63.071	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-50	1	0
16/09/2017 21:27:33	63.071	ZTE	cc:1a:fa:96:82:94	-92	1	3
16/09/2017 21:27:33	63.071	Dialog 4G	90:4e:2b:23:c0:e6	-85	6	4
16/09/2017 21:27:33	66.659	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-49	1	0
16/09/2017 21:27:33	66.659	ZTE	cc:1a:fa:96:82:94	-96	1	3
16/09/2017 21:27:33	66.659	Dialog 4G	90:4e:2b:23:c0:e6	-86	6	4
16/09/2017 21:27:37	70.263	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-49	1	0
16/09/2017 21:27:37	70.263	ZTE	cc:1a:fa:96:82:94	-93	1	3
16/09/2017 21:27:37	70.263	Dialog 4G	90:4e:2b:23:c0:e6	-84	6	4
16/09/2017 21:27:40	73.866	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-49	1	0
16/09/2017 21:27:40	73.866	ZTE	cc:1a:fa:96:82:94	-92	1	3
16/09/2017 21:27:40	73.866	Dialog 4G	90:4e:2b:23:c0:e6	-84	6	4
16/09/2017 21:27:44	77.439	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-50	1	0
16/09/2017 21:27:44	77.439	ZTE	cc:1a:fa:96:82:94	-93	1	3
16/09/2017 21:27:44	77.439	Dialog 4G	90:4e:2b:23:c0:e6	-84	6	4
16/09/2017 21:27:47	81.042	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-49	1	0
16/09/2017 21:27:47	81.042	ZTE	cc:1a:fa:96:82:94	-92	1	3
16/09/2017 21:27:47	81.042	Dialog 4G	90:4e:2b:23:c0:e6	-85	6	4
16/09/2017 21:27:51	84.646	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-49	1	0
16/09/2017 21:27:51	84.646	ZTE	cc:1a:fa:96:82:94	-92	1	3
16/09/2017 21:27:51	84.646	Dialog 4G	90:4e:2b:23:c0:e6	-85	6	4
16/09/2017 21:27:55	88.234	AI-THINKER_F92AB1_AP	1a:fe:34:9a:2e:88	-49	1	0

Table D.1: Sample Wi-Fi RSSI data gathered

Processed data to generate graphs

Sample data gathered, processed and stored in a database to plot the graph with RSSI vs time are as follows in Table D.2. Each SSID is a different column and time measured in a separate row. Data includes three SSIDs, which were scanned.

rssino	sessionno	datentime	elapsedtm	grphactive	ssid01	ssid02	ssid03	ssid04	ssid05
6625	7	10/01/2018 21:16:50	388.14	1	-54	-54	-47	0	0
6626	7	10/01/2018 21:16:51	389.14	1	-54	-54	-46	0	0
6627	7	10/01/2018 21:16:52	390.15	1	-54	-54	-46	0	0
6628	7	10/01/2018 21:16:53	391.15	1	-54	-54	-46	0	0
6629	7	10/01/2018 21:16:54	392.15	1	-54	-54	-46	0	0
6630	7	10/01/2018 21:16:55	393.15	1	-54	-52	-44	0	0
6631	7	10/01/2018 21:16:56	394.15	1	-54	-52	-44	0	0
6632	7	10/01/2018 21:16:57	395.15	1	-56	-55	-49	0	0
6633	7	10/01/2018 21:16:58	396.15	1	-54	-54	-46	0	0
6634	7	10/01/2018 21:16:59	397.15	1	-57	-53	-52	0	0
6635	7	10/01/2018 21:17:00	398.15	1	-54	-53	-48	0	0
6636	7	10/01/2018 21:17:01	399.15	1	-54	-54	-47	0	0
6637	7	10/01/2018 21:17:02	400.15	1	-54	-55	-47	0	0
6638	7	10/01/2018 21:17:03	401.15	1	-54	-55	-47	0	0
6639	7	10/01/2018 21:17:04	402.15	1	-54	-55	-47	0	0
6640	7	10/01/2018 21:17:05	403.15	1	-54	-55	-47	0	0
6641	8	14/01/2018 12:59:19	0.00	1	-58	-67	-59	0	0
6642	8	14/01/2018 12:59:20	1.00	1	-58	-70	-59	0	0
6643	8	14/01/2018 12:59:21	2.00	1	-58	-70	-59	0	0
6644	8	14/01/2018 12:59:22	3.00	1	-59	-64	-58	0	0
6645	8	14/01/2018 12:59:23	4.00	1	-59	-67	-60	0	0
6646	8	14/01/2018 12:59:24	5.00	1	-57	-68	-62	0	0
6647	8	14/01/2018 12:59:25	6.00	1	-57	-69	-61	0	0
6648	8	14/01/2018 12:59:26	7.00	1	-57	-64	-62	0	0
6649	8	14/01/2018 12:59:27	8.00	1	-57	-63	-60	0	0
6650	8	14/01/2018 12:59:28	9.00	1	-57	-64	-59	0	0
6651	8	14/01/2018 12:59:29	10.00	1	-57	-63	-59	0	0
6652	8	14/01/2018 12:59:30	11.00	1	-57	-63	-59	0	0
6653	8	14/01/2018 12:59:31	12.00	1	-57	-63	-59	0	0

Table D.2: Processed Wi-Fi RSSI data for graphs

Following Table D.3 shows statistical calculations, including the standard deviation on the sample data gathered stored in a database for comparison and identifying RSSI deviations from normal readings. The first is for a range of 50 reads and the other is the entire range.

ssidno	ssid	viewmax	viewmin	viewavg	viewstddv	viewcount
1	AI-THINKER_3E24D4	-54	-55	-54.72	0.454	50
2	FaryLink_5D6B80	-64	-66	-65.32	0.653	50
3	FaryLink_5D67BF	-53	-54	-53.60	0.495	50

ssidno	ssid	allmax	allmin	allavg	allstddv	allcount
1	AI-THINKER_3E24D4	-53	-69	-54.61	1.222	1456
2	FaryLink_5D6B80	-61	-73	-65.35	1.216	1456
3	FaryLink_5D67BF	-51	-70	-53.73	1.253	1456

Table D.3: Statistical calculations on Wi-Fi RSSI data gathered

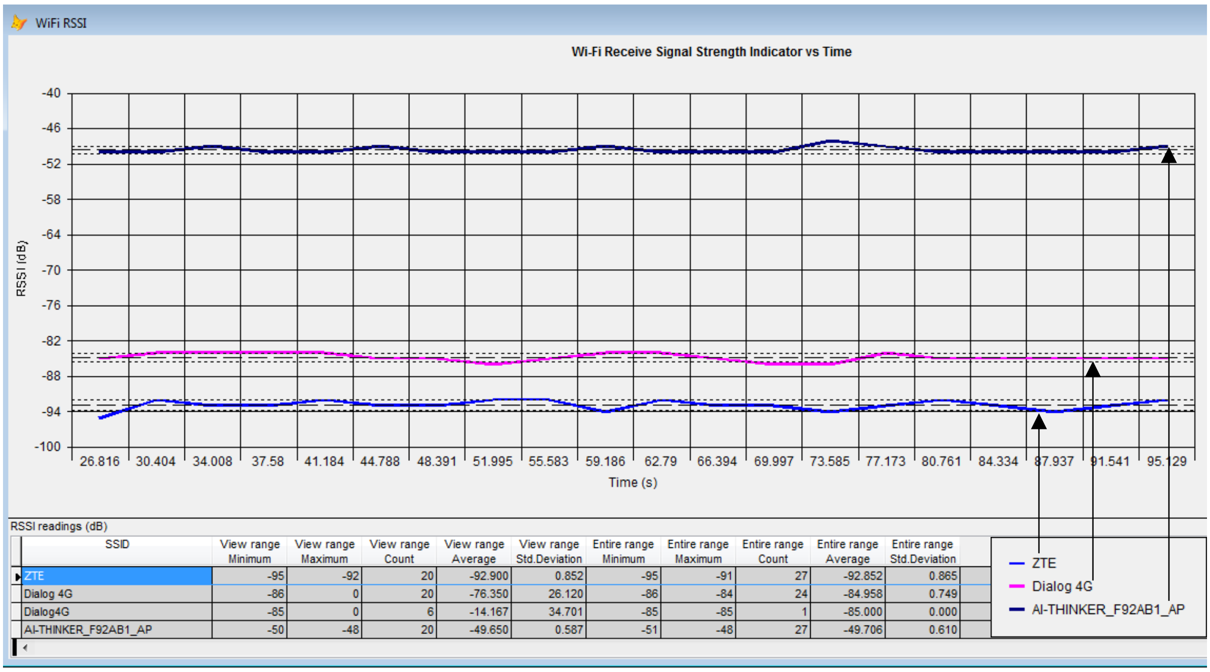
Appendix E

Graphs generated

Measuring RSS from surrounding Wi-Fi access points

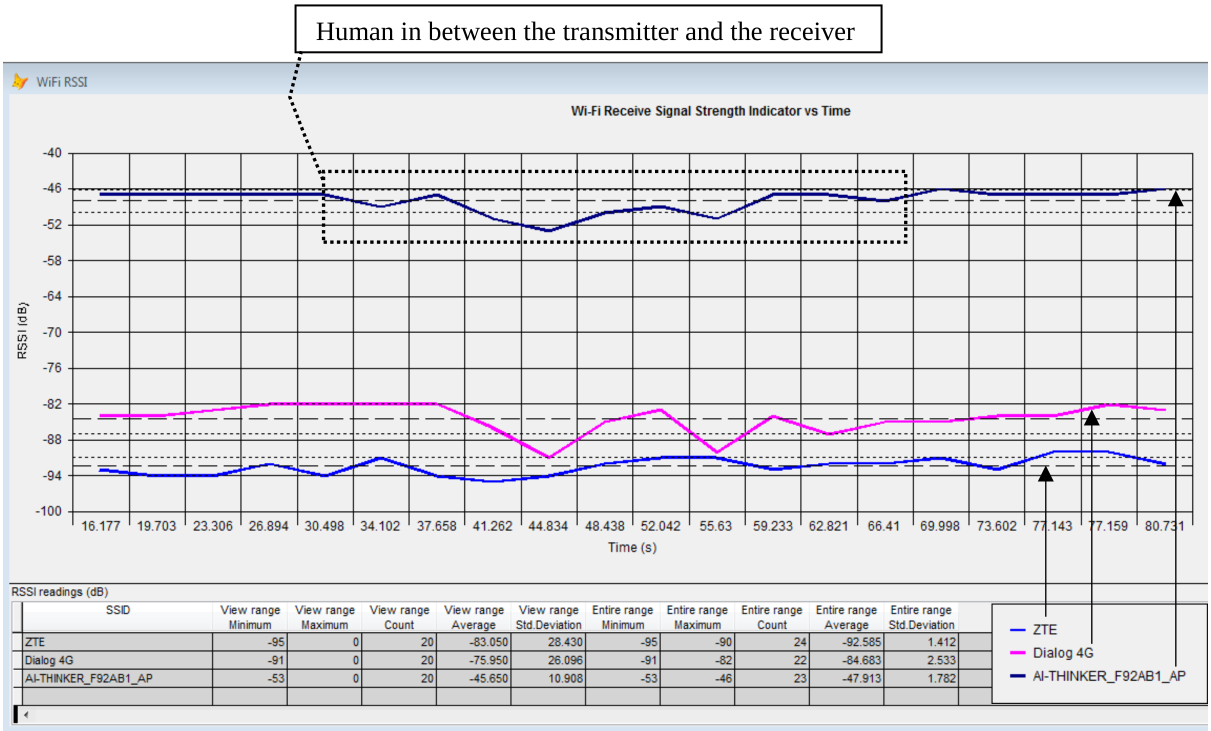
Following figures show reading RSS of surrounding Wi-Fi access points, with AT commands of ESP8266-01, which uses the default program saved in the microcontroller.

A human is placed to identify the difference in RSS.



SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
ZTE	-95	-92	20	-92.900	0.852	-95	-91	27	-92.852	0.865
Dialog 4G	-86	0	20	-76.350	26.120	-86	-84	24	-84.958	0.749
Dialog4G	-85	0	6	-14.167	34.701	-85	-85	1	-85.000	0.000
AI-THINKER_F92AB1_AP	-50	-48	20	-49.650	0.587	-51	-48	27	-49.706	0.610

Figure E.1: No disturbance between the Wi-Fi transmitter and the receiver



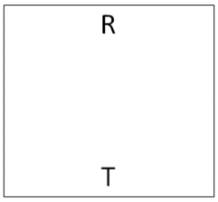
SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
ZTE	-95	0	20	-83.050	28.430	-95	-90	24	-92.585	1.412
Dialog 4G	-91	0	20	-75.950	26.096	-91	-82	22	-84.683	2.533
AI-THINKER_F92AB1_AP	-53	0	20	-45.650	10.908	-53	-46	23	-47.913	1.782

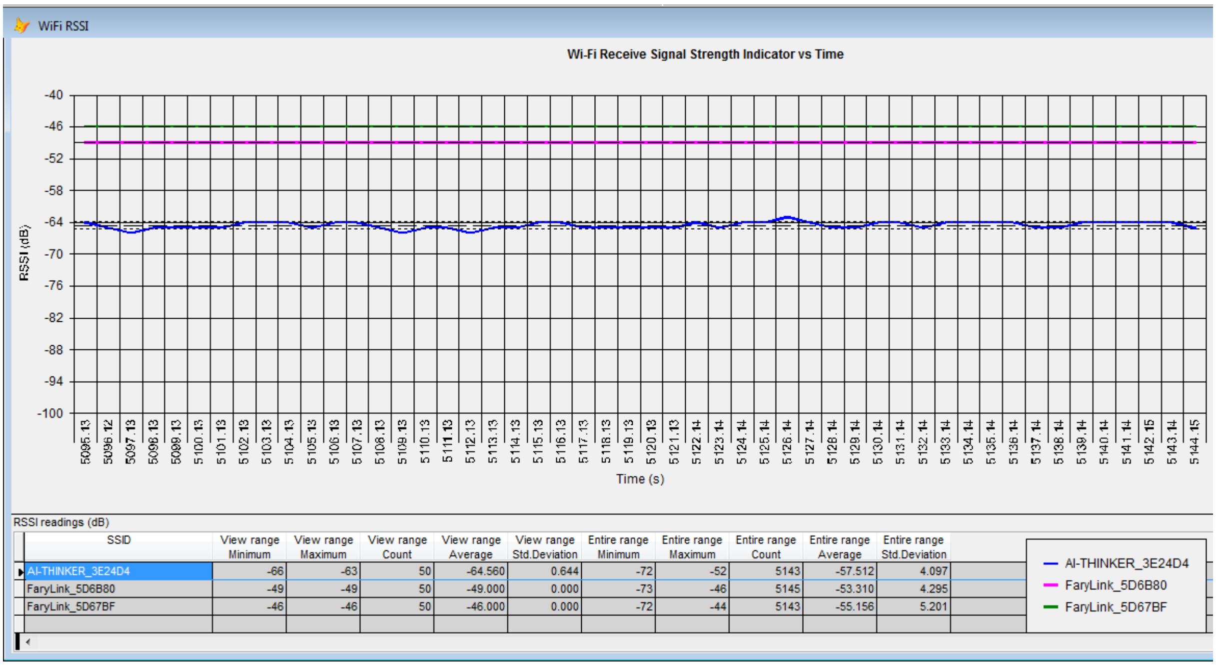
Figure E.2: Disturbance due to human between the Wi-Fi transmitter and receiver

Measuring RSS from different positions

Tests below were done placing the Wi-Fi transmitter and receiver in different positions within a square area of a side with 4m. Also the receiver was moved around a human, to identify patterens in RSS. More details could be referred at section 4.1.2.

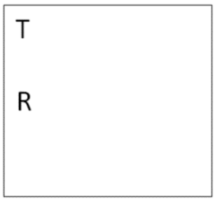
T – Transmitter
R – Receiver
H – Human

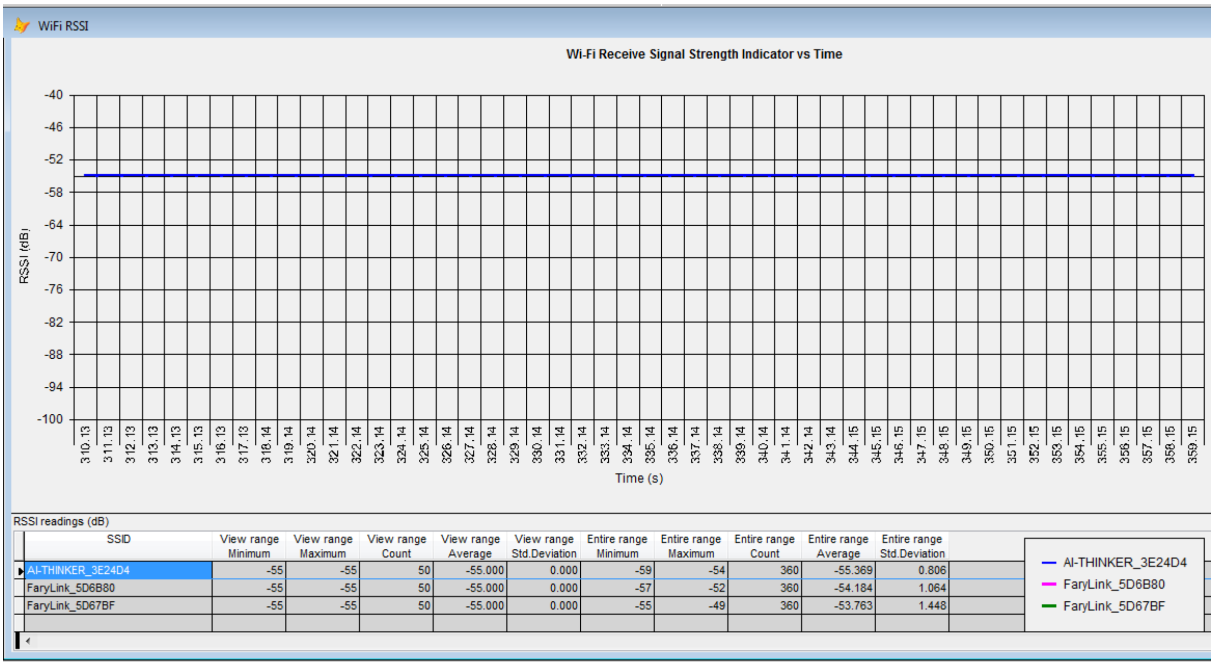




SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-66	-63	50	-64.560	0.644	-72	-52	5143	-57.512	4.097
FaryLink_5D6B80	-49	-49	50	-49.000	0.000	-73	-46	5145	-53.310	4.295
FaryLink_5D67BF	-46	-46	50	-46.000	0.000	-72	-44	5143	-55.156	5.201

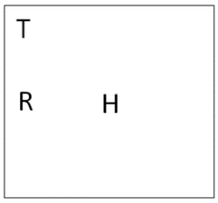
Figure E.3: Reading RSSI without disturbance from center of a defined boundary

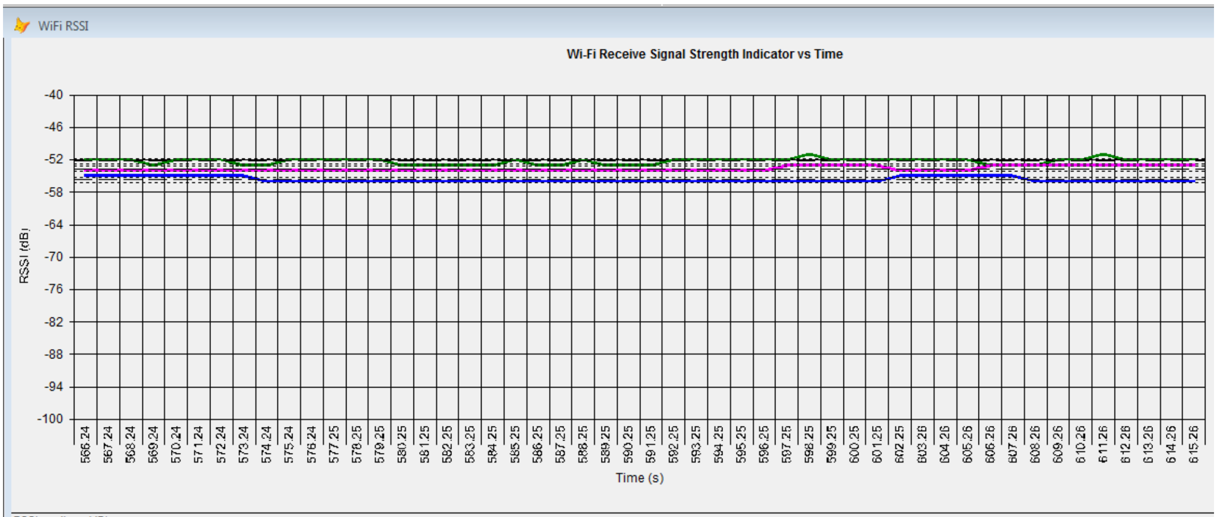




SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-55	-55	50	-55.000	0.000	-59	-54	360	-55.369	0.806
FaryLink_5D6B80	-55	-55	50	-55.000	0.000	-57	-52	360	-54.184	1.064
FaryLink_5D67BF	-55	-55	50	-55.000	0.000	-55	-49	360	-53.763	1.448

Figure E.4: Reading RSSI without disturbance from a corner in a defined boundary

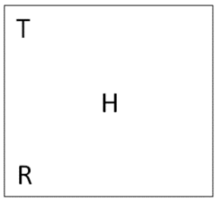


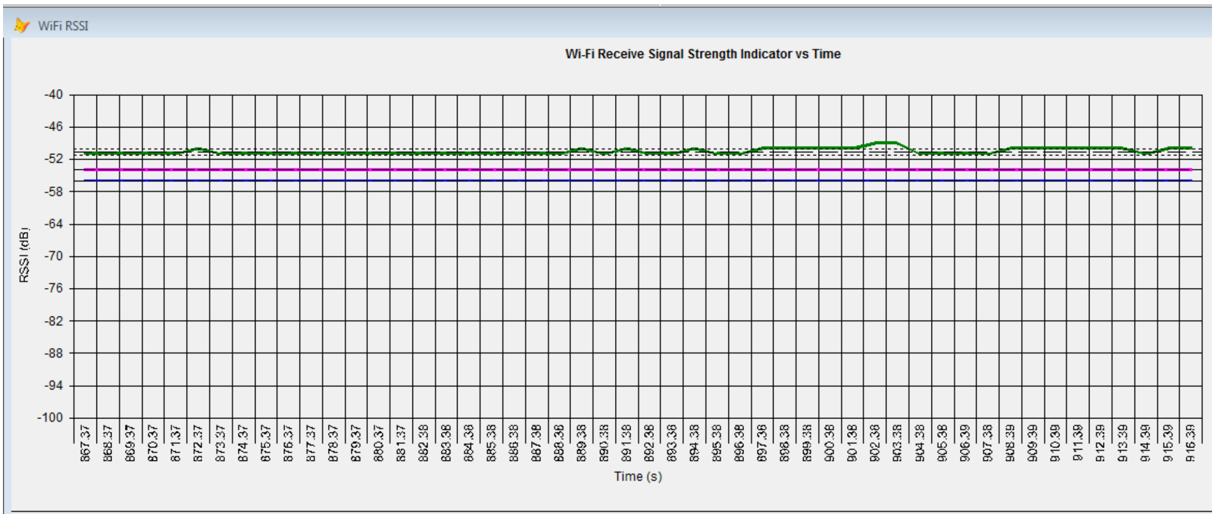


RSSI readings (dB)

SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-56	-55	50	-55.720	0.454	-59	-54	616	-55.343	0.734
FaryLink_5D6B80	-54	-53	50	-53.700	0.463	-57	-52	616	-54.205	0.934
FaryLink_5D67BF	-53	-51	50	-52.280	0.536	-55	-49	616	-53.468	1.335

Figure E.5: Transmitter, receiver and human, placed at 90 degrees

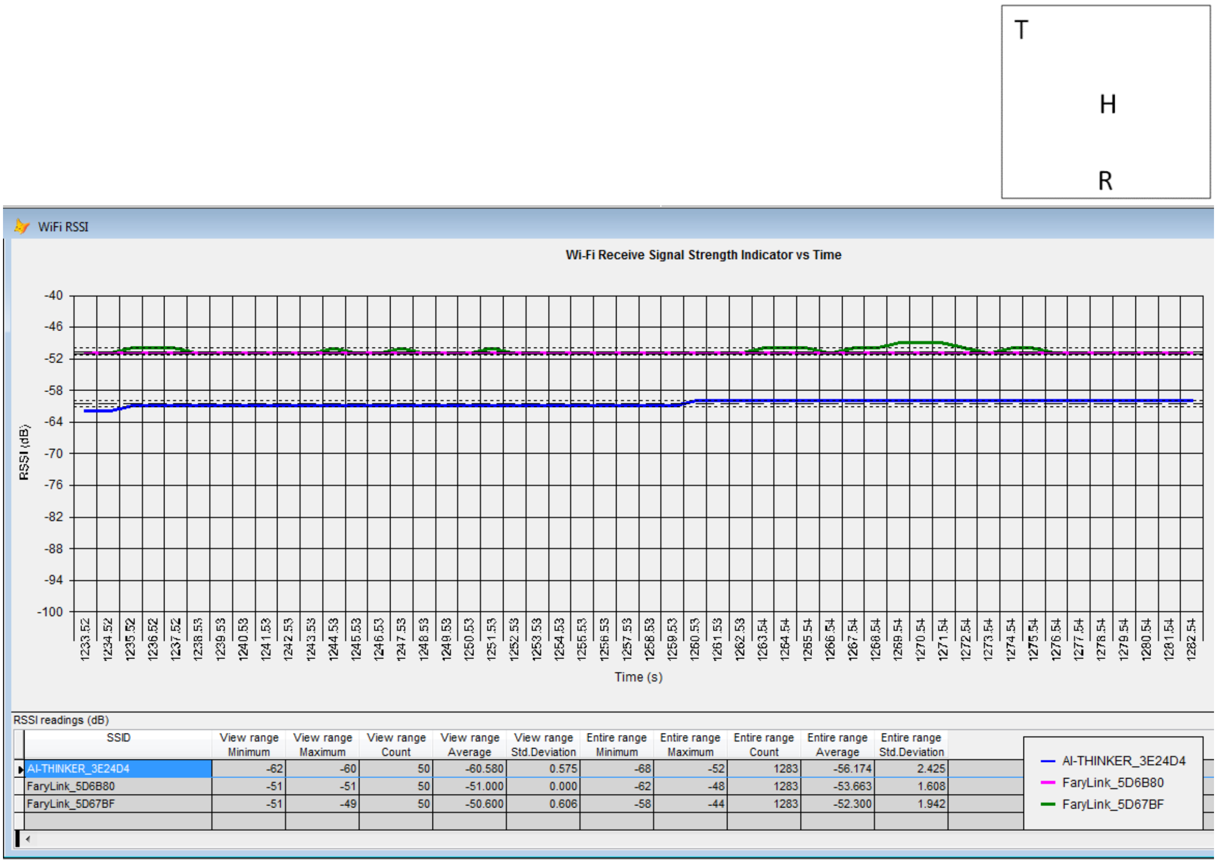




RSSI readings (dB)

SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-56	-56	50	-56.000	0.000	-59	-53	917	-55.376	0.879
FaryLink_5D6B80	-54	-54	50	-54.000	0.000	-62	-52	917	-54.127	0.929
FaryLink_5D67BF	-51	-49	50	-50.580	0.575	-57	-49	917	-52.905	1.654

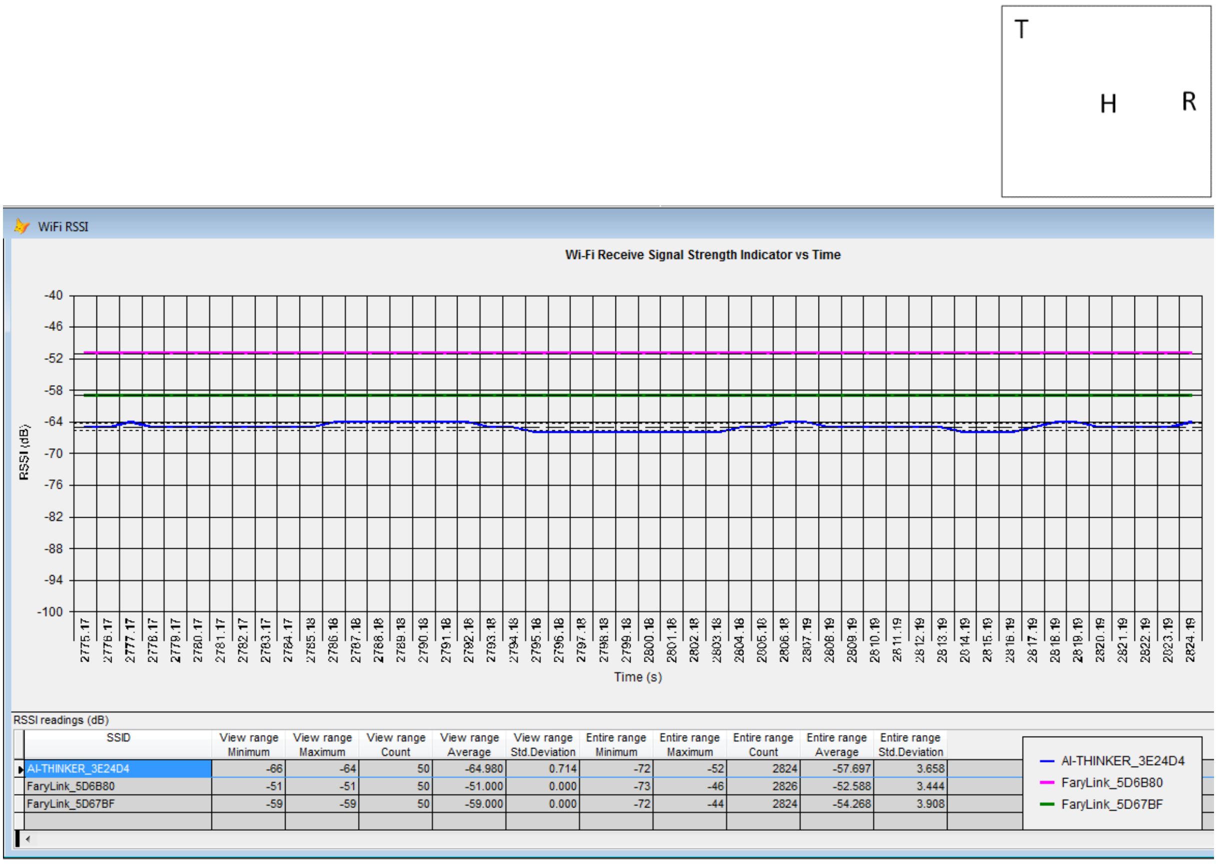
Figure E.6: Transmitter, receiver and human, placed 45 degrees



RSSI readings (dB)

SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-62	-60	50	-60.580	0.575	-68	-52	1283	-56.174	2.425
FaryLink_5D6B80	-51	-51	50	-51.000	0.000	-62	-48	1283	-53.663	1.608
FaryLink_5D67BF	-51	-49	50	-50.600	0.606	-58	-44	1283	-52.300	1.942

Figure E.7: Transmitter, receiver and human, placed at about 30 degrees



RSSI readings (dB)

SSID	View range Minimum	View range Maximum	View range Count	View range Average	View range Std.Deviation	Entire range Minimum	Entire range Maximum	Entire range Count	Entire range Average	Entire range Std.Deviation
AI-THINKER_3E24D4	-66	-64	50	-64.980	0.714	-72	-52	2824	-57.697	3.658
FaryLink_5D6B80	-51	-51	50	-51.000	0.000	-73	-46	2826	-52.588	3.444
FaryLink_5D67BF	-59	-59	50	-59.000	0.000	-72	-44	2824	-54.268	3.908

Figure E.8: Transmitter, receiver and human, placed at about 15 degrees

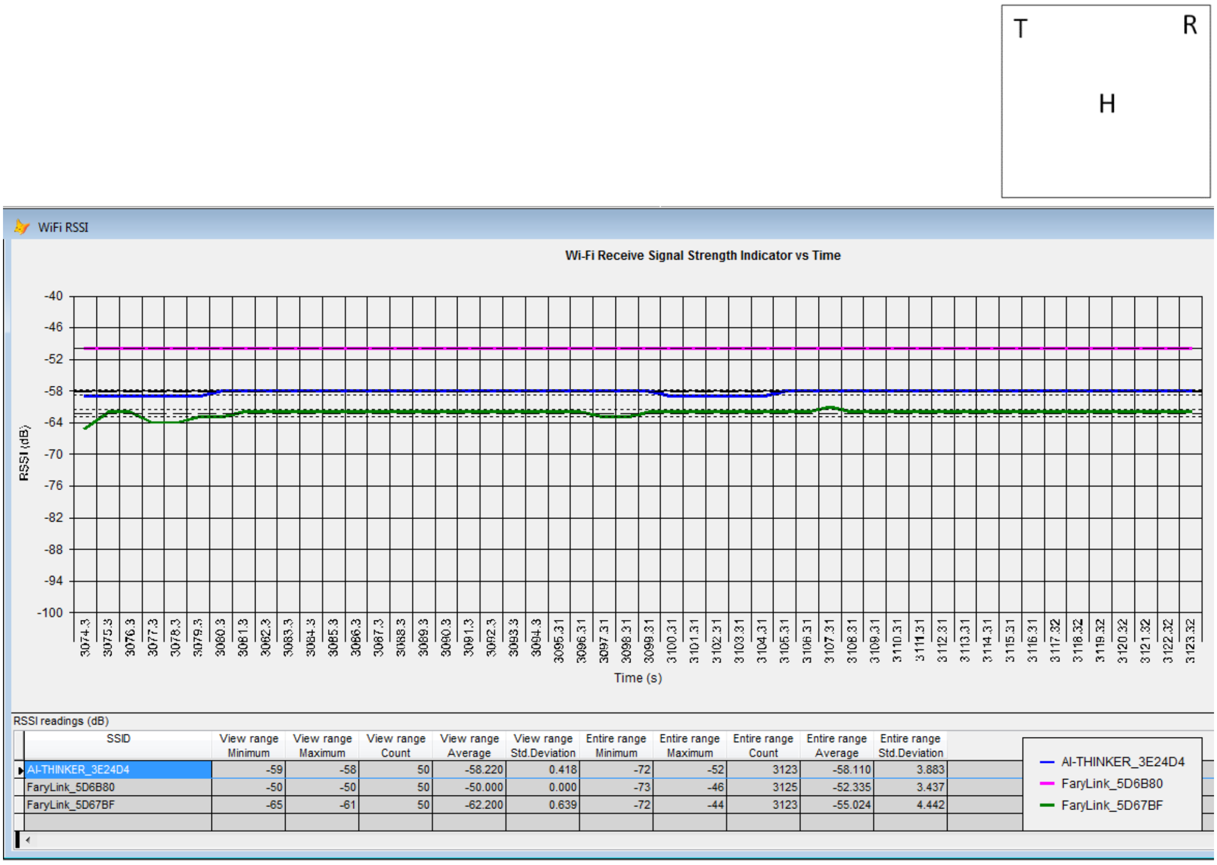


Figure E.9: Transmitter, receiver and human, placed at about 22 degrees

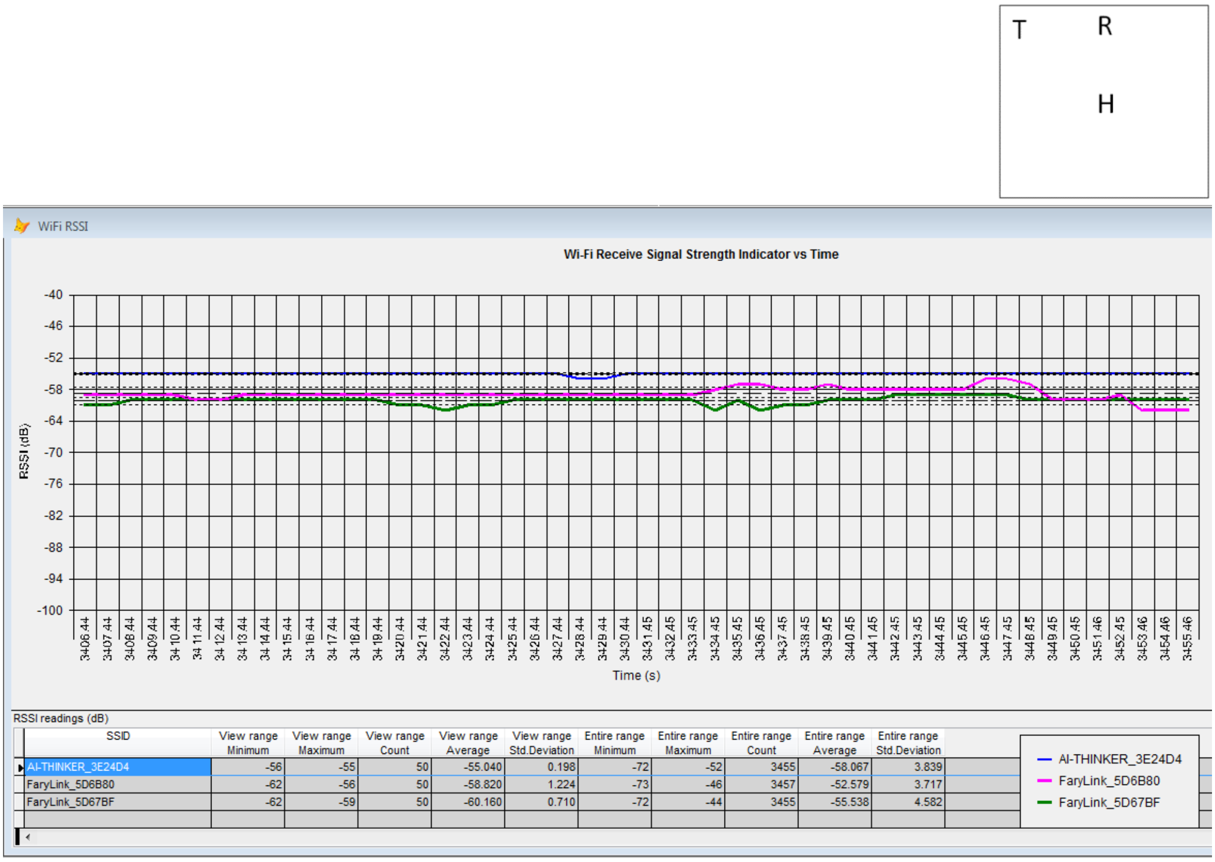


Figure E.10: Transmitter, receiver and human, placed at 90 degrees

References

[1] cybergibbons. (2013, May 9). Tomographic motion detection [Online]. Available: https://cybergibbons.com/alarms-2/tomographic-motion-detection/

[2] Dejan Nedelkovski. "How PIR Sensor Works and How To Use It with Arduino" [Online]. Available: https://howtomechatronics.com/tutorials/arduino/how-pir-sensor-works-and-how-to-use-it-with-arduino/

[3] OMRON. "Photoelectric Sensors | Technical Guide | Australia | Omron IA" [Online]. Available: http://www.omron.com.au/service_support/technical_guide/photoelectric_sensor/further_information.asp

[4] wikipedia. "Motion detector" [Online]. Available: https://en.wikipedia.org/wiki/Motion_detector#Sensors

[5] Matt Kankainen | Xandem Technology. (2013, September 13). "Tomographic Motion Detection | HomeToys" [Online]. Available: http://www.hometoys.com/content.php?post_type=2180

[6] XANDEM, "XANDEM HOME" [Online]. Available: http://xandem.com/xandem-home

[7] Joey Wilson, Neal Patwari, University of Utah. (2009, September 10). Radio Tomographic Imaging with Wireless Networks [Online]. Available: http://www.span.ece.utah.edu/uploads/RTI_version_3.pdf

[8] University of Utah. "Radio Tomographic Imaging | SPAN" [Online]. Available: http://span.ece.utah.edu/radio-tomographic-imaging

[9] Texas Instruments. "CC2540 Bluetooth low energy wireless MCU with USB | TI.com" [Online]. Available: http://www.ti.com/product/cc2540&DCMP=LowPowerRFICs+Other&HQS=Other+OT+cc2540

[10] Jin-Shyan Lee, Yu-Wei Su, Chung-Chou Shen, Information & Communications Research Labs, ITRI. A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi [Online]. Available: http://eee.guc.edu.eg/Announcements/Comparaitive_Wireless_Standards.pdf

[11] RF Wireless World."zigbee vs wifi | difference between zigbee and wifi" [Online]. Available: http://www.rfwireless-world.com/Terminology/zigbee-vs-wifi.html

[12] Comparison of Wireless Technologies(NFC - WIFI - Zigbee - Bluetooth - GSM) [Online]. Available: http://www.fut-electronics.com/wp-content/plugins/fe_downloads/Uploads/Comparison%20of%20Wireless%20Technologies.pdf.

[13] Ian Poole. Wi-Fi / WLAN Channels, Frequencies, Bands & Bandwidths [Online]. Available: http://www.radio-electronics.com/info/wireless/wi-fi/80211-channels-number-frequencies-bandwidth.php

[14] Jay Bennett, Popular Mechanics. (2015, Oct 28). "Researchers Use Wi-Fi To See You Through Walls" [Online]. Available: http://www.popularmechanics.com/technology/news/a17974/wireless-signals-let-you-see-people-through-walls/

[15] Michelle Ma, UW News. (2013, June 4). "Wi-Fi signals enable gesture recognition throughout entire home" [Online]. Available: http://www.washington.edu/news/2013/06/04/wi-fi-signals-enable-gesture-recognition-throughout-entire-home/

[16] Security Electronics and Networks. (2014, June 24). "Xandem TMD Tomographic Motion Detection Review" [Online]. Available: https://securityelectronicsandnetworks.com/articles/2014/06/24/xandem-tmd-tomographic-motion-detection-review/

[17] Anthony Joseph Wilson, Neal K. Patwari, "Systems and methods of device-free motion detection and presence detection", U.S. Patent US20120146788A1, June, 14, 2012

[18] Espressif Systems. (2013, Oct 12). ESPRESSIF SMART CONNECTIVITY PLATFORM: ESP8266 [Online]. Available: https://nurdspace.nl/images/e/e0/ESP8266_Specifications_English.pdf

[19] Espressif Systems. (2017, May 25). ESP8266 Technical Reference [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf

[20] ESP8266 Serial Esp-01 WIFI Wireless. (2015, June,15). [Online]. Available: https://www.optimusdigital.ro/index.php?controller=attachment&id_attachment=210

[21] Diwo. (2014, August 4). MEGA PINOUT (ver 3 rev 1) [Online]. Available: http://diwo.bq.com/wp-content/uploads/2014/11/mega.pdf

[22] ESP8266 Module [Online]. Available: https://cdn.sparkfun.com/datasheets/Wireless/WiFi/ESP8266ModuleV1.pdf

[23] Espressif. (2018, Feb 7). ESP8266 AT Instruction Set (Version 2.2) [Online]. Available: https://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_en.pdf

[24] Arduino. (2018). Arduino Mega 2560 Rev3 [Online]. Available: https://store.arduino.cc/arduino-mega-2560-rev3

[25] Arduino. (2018). "Arduino - Software" [Online]. Available: https://www.arduino.cc/en/Main/Software

[26] Microsoft. (2018). "Using the Communications Control" [Online]. Available: https://msdn.microsoft.com/en-us/library/aa231237(v=vs.60).aspx

[27] Brad Mager. (2014). Fall Detection Using RF Sensor Networks [Online]. Available: http://sensornetworksresearch.com/falldetection.html

[28] ESP8266 WiFi Module Quick Start Guide [Online]. Available: http://rancidbacon.com/files/kiwicon8/ESP8266_WiFi_Module_Quick_Start_Guide_v_1.0.4.pdf

[29] Robin Thomas. How to Program the ESP8266's Onboard GPIO Pins [Online]. Available: https://maker.pro/education/esp8266-tutorial-programming-the-onboard-gpio-pins

[30] Mike Lewis. Add graphs and charts to your Visual FoxPro applications [Online]. Available: http://www.ml-consult.co.uk/contact.htm

[31] ITEAD Wiki. (2016, June 21). ESP8266 Serial WIFI Module [Online]. Available: https://www.itead.cc/wiki/ESP8266_Serial_WIFI_Module

[32] Wikipedia. (2018, March 2). Standard deviation [Online]. Available: https://en.wikipedia.org/wiki/Standard_deviation

[33] Arduino. (2017, January 2). A quick statistics library for Arduino [Online]. Available: https://playground.arduino.cc/Main/QuickStats

[34] MathBitsNotebook. (2018). Understanding Z-Scores [Online]. Available: https://mathbitsnotebook.com/Algebra2/Statistics/STzScores.html

[35] K. Benkič, M. Malajner, P. Planinšič, Ž. Čučej. (2008, July). Using RSSI value for distance estimation in wireless sensor networks based on ZigBee [Online]. Available: https://www.researchgate.net/publication/4364297_Using_RSSI_value_for_distance_estimation_in_wireless_sensor_networks_based_on_ZigBee

[36] Arduino. (2018). WiFiRSSI [Online]. Available: https://www.arduino.cc/en/Reference/WiFiRSSI

[37] Ivan Grokhotkov. (2017). ESP8266WiFi library [Online]. Available: http://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html

[38] Sophie Weiner. (2017, Apr 29). This Box Detects Home Intruders By Only Using Wi-Fi [Online]. Available: https://www.popularmechanics.com/technology/security/a26312/aura-wi-fi-security/

[39] Math planet. Standard deviation and normal distribution [Online]. Available: https://www.mathplanet.com/education/algebra-2/quadratic-functions-and-inequalities/standard-deviation-and-normal-distribution

[40] Arduino. (2018). SoftwareSerial [Online]. Available: https://www.arduino.cc/en/Reference/SoftwareSerial

[41] Espressif Systems. Tools | Espressif Systems [Online]. Available: https://www.espressif.com/en/support/download/other-tools?keys=&field_type_tid%5B%5D=14

[42] DIY Projects. ESP-01: Reinstalling the Original NodeMCU Firmware (ESP8266) [Online]. Available: https://diyprojects.io/esp-01-reinstalling-original-firmware-esp8266/#.WqN5g3zYW01

[43] GitHub, Inc. (2016, Dec 27). Unable to load the code from arduino ide to esp8266 [Online]. Available: https://github.com/esp8266/Arduino/issues/2801

[44] The Mien Blog. Connecting ESP8266-01 With Arduino UNO via Software Serial [Online]. Available: http://www.mien.in/2017/07/21/arduino-uno-esp8266-software-serial/

[45] AI-Thinker. (2015, Sep 9). ESP-01 WiFi Module (Version1.0) [Online]. Available: http://www.electrodragon.com/w/images/f/fb/ESP-01_EN.pdf