

DEVELOPMENT OF AN ACTIVITY RECOGNITION SYSTEM USING ACCELEROMETERS

By

DANDY LAU JING HUI

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

© Copyright 2014

by

Dandy Lau Jing Hui, 2014

CERTIFICATION OF APPROVAL

DEVELOPMENT OF AN ACTIVITY RECOGNITION SYSTEM USING ACCELEROMETERS

by

Dandy Lau Jing Hui

13782

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL & ELECTRONICS ENGINEERING)

Approved:

AP. DR. TANG TONG BOON
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

August 2014

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

DANDY LAU JING HUI

ABSTRACT

Body Sensor Network (BSN) is a great area of interest in nowadays research due to the increase in demand of quality in healthcare. There are many applications and systems developed which are used to aid in monitoring health status of the people. With the increasing number of elderly people that live alone at most of the time or all the time, activity recognition system based on BSN can help in monitoring their activities at home. These systems could be further developed to integrate with telecommunication system so that it can alert the related parties should there be any emergency or potential threat of the home alone elders which would cost life, for instance falling down. This project is focused on developing an activity recognition system which is economic and lightweight for the use of elderly people such that it can accurately recognize some basic activities which are performed by the wearer. Different activity recognition systems are reviewed. Important factors especially the cost and size of the designed system are the priority for choosing the hardware. The prototype is built after deliberate studies and outlining the specifications according to requirement. An analytical software was developed using MATLAB to further illustrate the functions of the system. With accelerometers that capture the acceleration rate of different activities and Decision Tree algorithm for classification, the system is able to predict accurately the activity performed by the wearer. With two accelerometers (at thigh and ankle each), the system achieved overall accuracy of 94.7% in terms of recognition, an improvement of 27.3% as compared to system with one accelerometer.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to my supervisor, Assoc. Professor Dr. Tang Tong Boon for giving me the opportunity to do this project on the topic of Development of an Activity Recognition System Using Accelerometers, which also led me to do lot of researches and came to know about many new things and the capacity of Electrical and Electronic Engineering. Through his guidance and encouragement, the project has been successfully accomplished.

Furthermore, I am sincerely grateful to those who have helped me in giving constructive criticism and friendly advice during the project work, especially my course-mates. I would also like to thank my peers especially those who had been my subjects for experiment to gather data for analysis. Nevertheless, I express my warm thanks to all the examiners and judges of Poster Presentation, ELECTREX and SEDEX for sharing their illuminating views on a number of issues related to the project.

Lastly, I would also like to thank my parents and friends who have encouraged and supported me throughout the process.

TABLE OF CONTENTS

Table of Contents

ABSTRACT.....	I
ACKNOWLEDGEMENTS.....	II
INTRODUCTION.....	1
1. Background	1
2. Problem Statement.....	2
3. Objectives.....	2
4. Scope of Study.....	2
LITERATURE REVIEW	3
METHODOLOGY	7
1. Conceptual Design of Activity Recognition System	7
Hardware Architecture Design.....	7
2. Prototype Hardware Design.....	9
Configuration of Xbee Wireless Module	12
Testing of Capability of Transmitter and Receiver.....	13
Summary of Configurations	14
3. Software Development	15
Training Module.....	15
Testing Module	17
Testing Module with GUI	17
4. Method of Data Classification.....	21
Description about Classification Decision Tree.....	21
Computation Technique and Feature Extraction.....	21
5. Justification on the Accuracy of Recognition	21
6. Key Milestones	22
7. Gantt Chart.....	23
RESULTS & DISCUSSION	24
1. One-Sensor-Node Activity Recognition System	24
2. Two-Sensor-Nodes Activity Recognition System	26

3. Comparison between two systems and justification.....	28
CHAPTER 5: CONCLUSION & FUTURE WORKS	29
REFERENCES.....	30
APPENDICES	32
1. Form to be filled by candidate of experiment	32
2. Certificate of Participation in 33 rd Science & Engineering Design Exhibition	33
3. Data Sheet of Accelerometers – ADXL335	34
4. Data Sheet of Xbee Transceiver	34

Table of Tables

Table 1: Summary of a representative sample of past work on activity recognition using acceleration.....	4
Table 2: Comparison of specifications between Bluetooth, UWB, ZigBee and Wi-Fi.....	6
Table 3: List of Components	11
Table 4: Key Milestones for FYP 1	22
Table 5: Key Milestones for FYP 2	22
Table 6: One-sensor-node System Result	25
Table 7: Two-sensor-nodes System Result	26

Table of Figures

Figure 2: Hardware Architecture Design of Activity Recognition System.....	7
Figure 3: Components for Prototype	9
Figure 4: Schematic Diagram of the Connection for Transmitter	10
Figure 5: Connections of Accelerometer to Microcontroller (left) and Complete Sensor Node (right)	10
Figure 6: SKXbee Receiver Module (left) and Sensor Node In Operation (right).....	11
Figure 7: X-CTU Software Interface.....	12
Figure 8: Terminal to Program the Xbee Module	13
Figure 9: Sample Program to test the functionality of Sensor Node	13
Figure 10: Initializing Training Module in MATLAB.....	16
Figure 11: Graph displaying real time data acquisition in Testing Module when the subject is sitting	16
Figure 12: Initializing Testing Module in MATLAB	18
Figure 13: Graphical model of Classification Decision Tree (end nodes labelled as '1' stands for walking, '2' for standing, '3' for sitting and '4' for lying down)	18
Figure 14: Real time activity recognition in Testing Module	19
Figure 15: MATLAB GUI for Testing Module	19
Figure 16: Human vector graphics to illustrate the predicted activity performs by subject in Test GUI software	20

Figure 17: Subject under test with two-sensor-node-system (thigh and ankle) for standing activity where sensors are secured in position with reusable bandage.....	20
Figure 18: Gantt chart for FYP 1.....	23
Figure 19: Gantt chart for FYP 2.....	23
Figure 20: Classification Tree for one-sensor-node system on thigh (end nodes labelled as '1' stands for walking, '2' for standing, '3' for sitting and '4' for lying down)	24
Figure 21: Classification Tree for two-sensor-nodes system on thigh and ankle (end nodes labelled as '1' stands for walking, '2' for standing, '3' for sitting and '4' for lying down)	27

CHAPTER 1

INTRODUCTION

1. Background

Accidental fall is one of the potential risks which can be fatal for the elderly people. Fall is ranked second in terms the cause of death due to accidental or unintentional injury. It is recorded that about 424,000 people died each year (or equivalent to 1,161 people every day) due to fall with most of them aged above 60 [1]. Many suffer with other severe injuries like fractural of bones or permanent disabled. Elderly people are defined as people that is 60 years old or more [2]. In United States, over the years (from 2000 to 2010), based on the study of U.S. Bureau of the Census, it shows that the change in percentage of the number of elderly people aged above 65 is 15.1% increased [3]. The risk of death due to fall is higher in the situation where elders are living alone. Study in Australia shows that one quarter of people aged 65 is living alone [4]. The number of elderly people living alone is increasing and these numbers did not include those who are being left alone for a certain period of time [5]. With the fast growing population of elderly people and increase in number of elderly people living alone, improvement on healthcare and security for elderly people has to be taken into consideration seriously.

With the advancement of sensing technology, pervasive computing and the affordability of healthcare, electronic system can be designed and implemented to monitor the human daily activities for the aim of monitoring health and thus improving it. There are many researches focusing on intelligent wearable interfaces and human-machine interfaces which use the information from the body itself as the data to feed into the system which can perform useful functions. The degree of implementation and the effectiveness of such system can be varied, depending on how sophisticated of the design. For instance, activity recognition system such as [6] which can sensed and analyzed the health status of a person who wore it and send the information through communication network. Such system enabled the healthcare personnel to monitor the health of the patient which stays at home and also the children which are far from home to know that their elders are fit at home. Thus, this could be the solution to the problem mentioned earlier, though not to be able to

prevent the happen of fall but to ensure immediate acknowledgement and shortest possible delay of medical assistance.

2. Problem Statement

Fall has a huge impact on the quality of life and this has been one of the major cause of death for elderly people. As more and more elderly people are living alone at home or unattended at most of the time, accident might happened to them and if helps cannot be offered right at the time, their life could be endangered. Others would not be acknowledge until it is too late. Concerning on the other sides, the trend of the society, the age of retirement is extending, living expenses are increasing and demand on the quality of life is increasing. Most of the children of the elders nowadays have their own life with busy schedule and would not have enough time to spend on taking care of their parents. Without personal attendant for the elders and children are not around with them at most of the time, the abovementioned situation would be a great issue to the society.

3. Objectives

The aim of the project is to design a lightweight and economic activity recognition system which is suitable to be used by elderly people. It has to be able to use in real ambient situation and accurately recognize the activities of the user. Correct algorithm(s) or techniques have to be used and suitable type of hardware should be used to design the system in order to optimize the performance and the cost of the system.

4. Scope of Study

- a) Review different types of body sensor networks for activity recognition
- b) Design and implement with accelerometers an activity recognition system
- c) Develop the software to demonstrate the system
- d) Gather samples to test the system and justify the accuracy of recognition

CHAPTER 2

LITERATURE REVIEW

There are various ways to implement activity recognition system. Activity recognition system is categorized mainly into two types: external sensing or wearable sensing [7]. For external sensing, sensors are installed in a building or within a premise. With only some small, low cost, simple and easy to install sensors and the right design, external sensing activity recognition system can be easily implemented at home and can achieved notable outcome [8]. Though it can recognized highly complex activities but with the drawback of line-of-sight limitation. On the other hand, for wearable sensing type, it utilizes small size, independent powered sensors and interfaces which can be wearable by the subject and thus resulted in higher portability as well as flexibility of the system. Sensor nodes which are wore by the user at different locations of the body will record accurate and extensive information from the body in order to predict and analyze the activity performed by the subject. With the advantages of portability and flexibility, the wearable sensing type activity recognition system is preferred to be the used for elderly people who are living at home setting.

Considering the wearable type activity recognition system, to design such a system, many factors and variables have to be carefully considered in order to obtain the desired outcome and to meet the specific requirement. Some important factors are the type of sensors to be used, number of sensors to be used, position of the sensor to be placed, technology to be used and method of classification for pattern recognition. Each factors mentioned will be further explored in the following sections.

At the lowest tier of the system, sensor is the basic input component of the system which collects and feeds data into the next level to be processed. There are many types of sensors which can be used as the inputs of activity recognition system. Different sensors have been combined and used in research studies and results differently. Sensors such as accelerator sensor, camera, microphone, humidity sensor, thermometers, and light sensors are being used as the data acquisition unit. Different design utilizes different type of sensors and many of them combined two or more sensors to acquire the input data. Yet,

there is no significant difference in terms of the accuracy to recognize the activity when different type of sensors is being deployed. For instance, [9] uses accelerometers only to perform their study while [10] uses combination of microphone and accelerometers, both yield high accuracy rate up to 84%. Thus, choosing the type of sensor to be used will be depending on other factors such as cost, size and system design. Due to various advantages of small size, low power consumption and low cost of accelerometers as mentioned in [11], it is being widely used in variety of applications.

Recognition Accuracy	Activities Recognized	No. Subj.	Data Type	No. Sensors	Sensor Placement
92.85% to 95.91%	ambulation	8	L	2	2 thigh
83% to 90%	ambulation, posture	6	L	6	3 left hip, 3 right hip
95.8%	ambulation, posture, typing, talking, bicycling	24	L	4	chest, thigh, wrist, forearm
66.7%	ambulation, posture, typing, talking, bicycling	24	N	4	chest, thigh, wrist, forearm
89.30%	ambulation, posture	5	L	2	chest, thigh
N/A	walking speed, incline	20	L	4	3 lower back 1 ankle
86% to 93%	ambulation, posture, play	1	N	3	2 waist, 1 thigh
$\approx 65\%$ to $\approx 95\%$	ambulation, typing, stairs shake hands, write on board	1	L	up to 36	all major joints
96.67%	3 Kung Fu arm movements	1	L	2	2 wrist
42% to 96%	ambulation, posture, bicycling	1	L	2	2 lower back
85% to 90%	ambulation, posture	10	L	2	2 knee

Note. Adapted from *Pervasive Computing*. vol. 3001, A. Ferscha and F. Mattern, Eds., by L. Bao and S. Intille, 2004, ed: Springer Berlin Heidelberg, pp. 1-17.

Table 1: Summary of a representative sample of past work on activity recognition using acceleration

Accelerometer is a device that measures the acceleration of a moving object of all directions (xyz directions). It converts the physical activity into electrical signal which then can be used to analyze by electronically. Accelerometer can detect the motion or vibration, and it also can be used to measure the gravity and thus determining the orientation of the object such as tilt and inclination. Table 1 shows the summary of past

work on activity recognition using data of accelerometer as the input, mostly yield high recognition accuracy except for few studies which involved bicycling which is not necessary and applicable for the design system of this project. With the advantage of small size and lightweight, MEMS based capacitive accelerometer is widely used in consumer electronics nowadays [12]. Therefore, further contribute to miniaturization of the acquisition system.

In order to recognize activity performed by the subject, a number of sensors have to be deployed at different locations of the body. Number of sensors used in the system and position that they are being placed will inevitably affect the accuracy of activity recognition. In [13], the accuracy of recognition of different basic activities which sensors are being placed at 7 different locations on the body are being assessed. It is demonstrated that data acquired from wrist, arm, thigh, waist and ankle are highly distinctive for different activities. With the proposed method including the suggested sensor locations, the accuracy of recognition yielded is as high as 87%. In [9], it concludes that the at least two sensors have to be used and sensors placed on thigh and dominant wrist will yield higher accuracy for various type of activities performed.

The technology used for designing the activity recognition system varies, depending on the requirement of the system. Type of interfaces used is different from one to another design. Most of researches focused on wireless interfaces such as Bluetooth [14], or Radio Frequency. Different network can also be used to transmit the data such as Wi-Fi or GPRS network. The information can be displayed on different devices such as mobile phone [14], smart phone, pocket PC [15] and other platforms [16]. Based on the comparison of Table 2, with all the concerns such as power consumption, cost and performance in this design, ZigBee protocol is favored over the other wireless protocols such as UWB, Wi-Fi and Bluetooth. With only -25dB to 0 dB nominal transmission power, nominal range of 10-100m and high number of maximum cell nodes as reviewed in [17], ZigBee stand out to be the right choice for the wireless protocol for activity recognition system. The only drawback of ZigBee protocol is the maximum signal rate which is only 250kb/s. Yet, since the required data transfer rate of the sensor nodes for activity recognition system is very

small (low sampling frequency to reduce power consumption), it would not affect the capability of the system.

Standard	Bluetooth	UWB	ZigBee	Wi-Fi
IEEE spec.	802.15.1	802.15.3a *	802.15.4	802.11a/b/g
Frequency band	2.4 GHz	3.1-10.6 GHz	868/915 MHz; 2.4 GHz	2.4 GHz; 5 GHz
Max signal rate	1 Mb/s	110 Mb/s	250 Kb/s	54 Mb/s
Nominal range	10 m	10 m	10 - 100 m	100 m
Nominal TX power	0 - 10 dBm	-41.3 dBm/MHz	(-25) - 0 dBm	15 - 20 dBm
Number of RF channels	79	(1-15)	1/10; 16	14 (2.4 GHz)
Channel bandwidth	1 MHz	500 MHz - 7.5 GHz	0.3/0.6 MHz; 2 MHz	22 MHz
Modulation type	GFSK	BPSK, QPSK	BPSK (+ ASK), O-QPSK	BPSK, QPSK COFDM, CCK, M-QAM
Spreading	FHSS	DS-UWB, MB-OFDM	DSSS	DSSS, CCK, OFDM
Coexistence mechanism	Adaptive freq. hopping	Adaptive freq. hopping	Dynamic freq. selection	Dynamic freq. selection, transmit power control (802.11h)
Basic cell	Piconet	Piconet	Star	BSS
Extension of the basic cell	Scatternet	Peer-to-peer	Cluster tree, Mesh	ESS
Max number of cell nodes	8	8	> 65000	2007
Encryption	E0 stream cipher	AES block cipher (CTR, counter mode)	AES block cipher (CTR, counter mode)	RC4 stream cipher (WEP), AES block cipher
Authentication	Shared secret	CBC-MAC (CCM)	CBC-MAC (ext. of CCM)	WPA2 (802.11i)
Data protection	16-bit CRC	32-bit CRC	16-bit CRC	32-bit CRC

Table 2: Comparison of specifications between Bluetooth, UWB, ZigBee and Wi-Fi

The last tier of the processing of data is the classification of activity. The method of data classification used is different from one to another research. Method of classification is required for identify the activity performed by analyzing the pattern of data gathered. All the methods are developed from statistical models for machine learning. Some of the commonly used methods which have been assessed are Hidden Markov Model (HMM), Decision Tree, Naïve-Bayesian and k-Nearest Neighbor (k-NN). Most studies used more than one method for activity classification such as in [9]. In the study of [16], it is suggested that combination of Decision Tree and Naïve-Bayesian method yield higher accuracy of recognition and it has a moderate computation complexity as compared to the other methods.

CHAPTER 3

METHODOLOGY

1. Conceptual Design of Activity Recognition System

Hardware Architecture Design

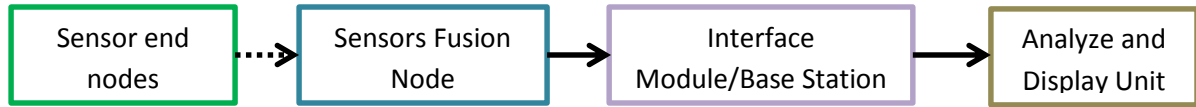


Figure 1: Hardware Architecture Design of Activity Recognition System

Figure shows the basic hardware architecture design of the activity recognition system which is to be deployed. Sensors to be used are accelerometer (Triple axis accelerometer ADXL335) only. XBee 802.15.0.4 (Series 1) wireless module will be interfacing with the sensors and transmit the data in packets to the receiver end wirelessly. Multiple sensor nodes will be used on different positions of the body to acquire data, similar to [9]. Sensors fusion node will be the main coordinator that gather the data from multiple sensors and to send to the base station. Base station with the specific type of interface will transmit the data to analyzer which is used to analyze and display the information. Analyze and display unit will be the computer. The sensors fusion node will receive the packets data sent by sensors and transfer to computer through interface module board.

Placement of Sensors

Accelerometers will be placed on two locations which are dominant wrist, dominant thigh to acquire accurate and distinctive data as suggested and assessed by different studies such as in [9, 13]. The accuracy of all the suggested studies yields more than 80% on activity recognition. We will also be examining accuracy of activity recognition by placing sensor on ankle and/or chest. Sensor nodes will be wrapped in reusable bandage which will then be tied to the abovementioned locations to acquire data from the subject.

Technology used

Based on review results (Chapter 2), it has come to the conclusion that ZIGBEE protocol should be used due to its advantages of low power consumption, low cost and adequate data rate for this system [17]. Xbee 802.15.0.4 (Series 1) wireless module will be deployed as the transmitter and the receiver for the communication. Accelerometer to be used is triple axis ADXL335 type which

can measure the acceleration in x, y and z directions (measuring $\pm 3g$). Xbee Explorer will be used as the microcontroller that perform analog-digital conversion of the acceleration data before feeding forward to the transmitter to be sent. The interface module used on the receiver side is SKXbee. The receiver module together with the interface board will be communicating through UART serial communication (but physically USB cable interface) with the computer.

Network Topology

Due to the architecture of the system which is point-to-multipoint, star topology will be implemented. Therefore, Xbee Series 1 will need to be configured to use API mode instead of AT mode. One node (receiver) will be the coordinator which sinks all the data from other nodes (transmitters). During the communication, the flow of the data has to be controlled. Access control and flow control will be following the standard of IEEE 802.15.0.4.

Algorithm for Classification

Different type of algorithm will be used for the recognizing and classifying the activity. Algorithms such as C4.5 Decision Tree and Naïve-Bayan will be combined to analyze the data set acquired as suggested in [9, 16]. A definite time window of data will be used to compare with trained data set in order to make the correct justification of the activity performed. Statistical approach will be used to extract the feature from the time window. Suitability of algorithm will be further justified based on the result.

Activities to be recognized

Four basic human activities which will be examined in the classification are walking, standing, sitting and lying down. These activities are chosen because these are the basic daily activities which are most frequently performed by the people at home.

Materials and Tools to be used

For hardware, computer, microcontroller, wireless module, accelerometers and dry cell battery will be used to construct the system. Tools such as soldering tools, cutter, Digital Multimeter (DMM) would also be required. For software, MATLAB software will be used as analytical software for this project. Compiler software might also be necessary if programming of microcontroller or other parts of the system required coding. MATLAB classification functions will be used to analyze and classify the data as required for activity recognition.

2. Prototype Hardware Design

Figures below showed the progress of prototyping the sensor node design.

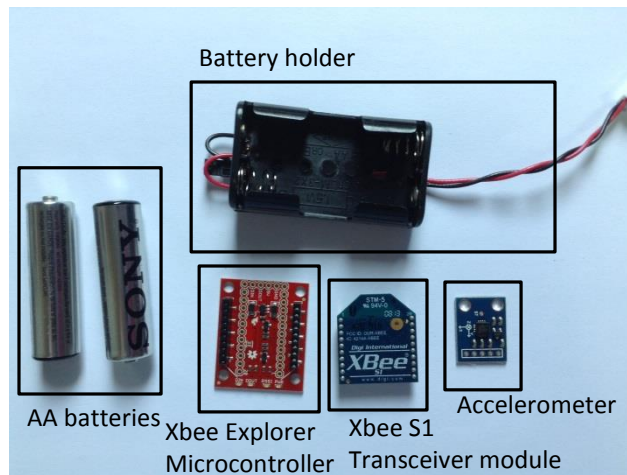


Figure 2: Components for Prototype

Two AA batteries and a battery holder were used as the power supply for each transmitter sensor node. Component in red is the Xbee Explorer Regulated which is the microcontroller that acts as the communication bridge for the accelerometer and Xbee wireless module.

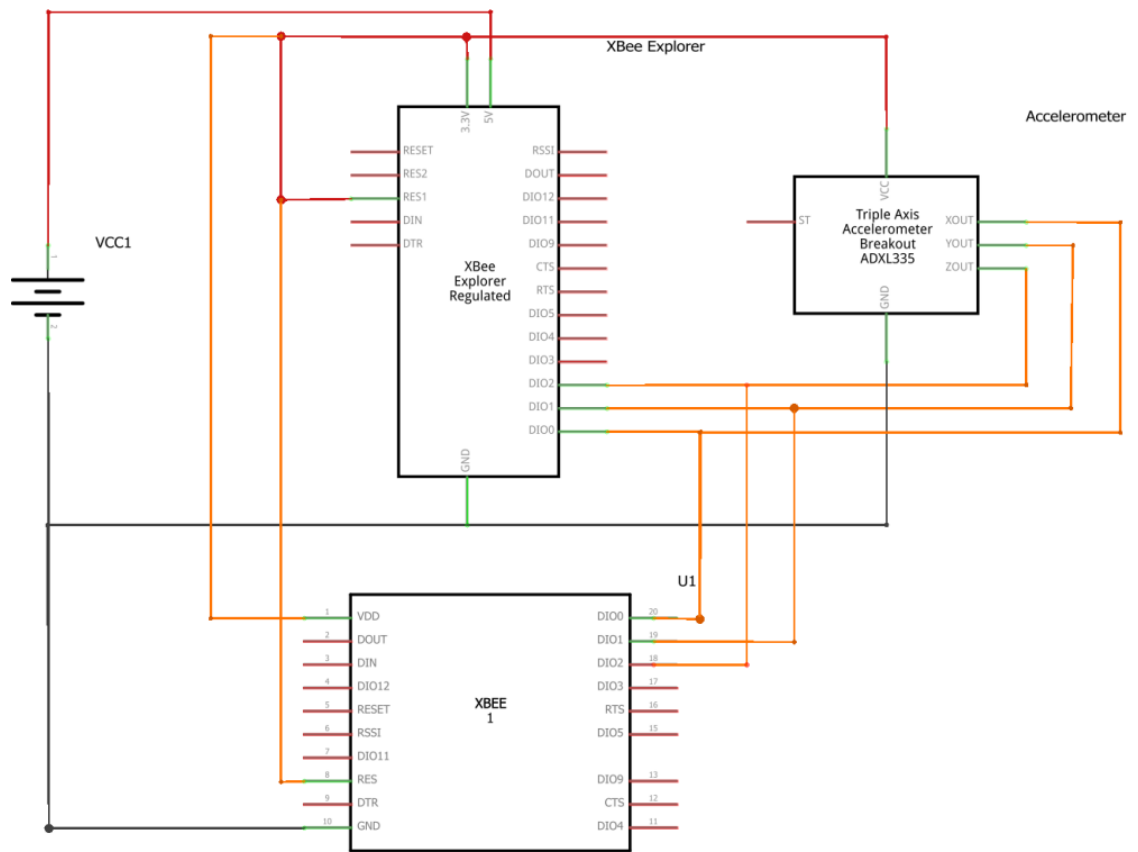


Figure 3: Schematic Diagram of the Connection for Transmitter

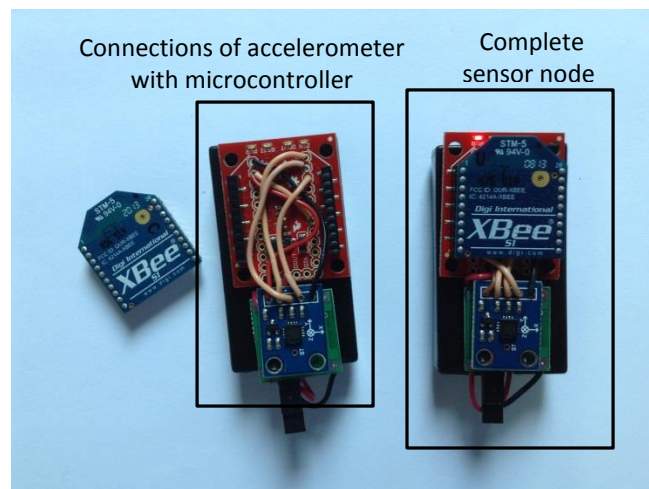


Figure 4: Connections of Accelerometer to Microcontroller (left) and Complete Sensor Node (right)

The schematic diagram for the transmitter was sketched. The sensor node was tested after the basic soldering works for the connections are done. For the current prototype, each sensor node is weighted about 55g and the dimension is 7.0cm x 3.0cm x 1.3cm, which is considered to be lightweight and small.

On the receiver side, SKXbee shield was used. It was coupled with another Xbee wireless module which was configured to be the receiver node using software provided by the manufacturer (Digi International, X-CTU software). The SKXbee has got micro-USB interface and it was connected physically to the laptop with USB interface. It is communicating in serial with the computer.

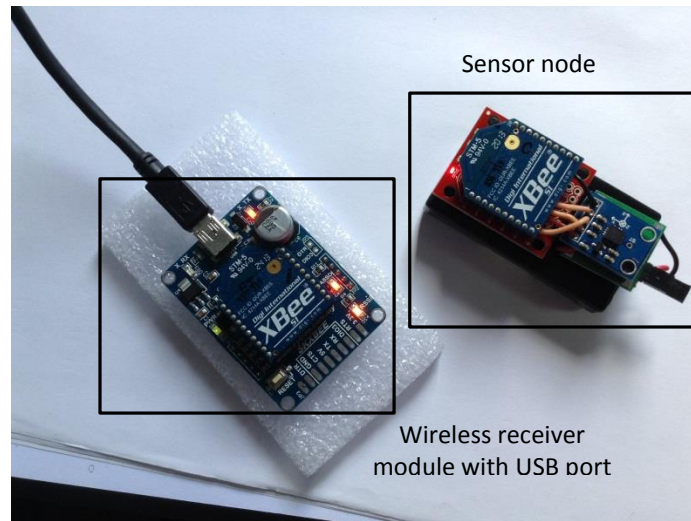


Figure 5: SKXbee Receiver Module (left) and Sensor Node In Operation (right)

Following is the list of components which are used for the prototype:

No.	Component	Quantity
1	ADXL335 Module 3-Axis Analog Output Accelerometer	2
2	XBee Explorer Regulated	2
3	Xbee 1mW Trace Antenna – Series 1 (802.15.4)	3
4	SKXbee Module (with USB cable)	1
5	5V DC-DC Step Up (2xAA) & Battery Holder	2
6	AA batteries	4

Table 3: List of Components

Configuration of Xbee Wireless Module

All the Xbee wireless modules were programmed as either receiver or transmitter using the software which was provided by the manufacturer (Digi International), named X-CTU.

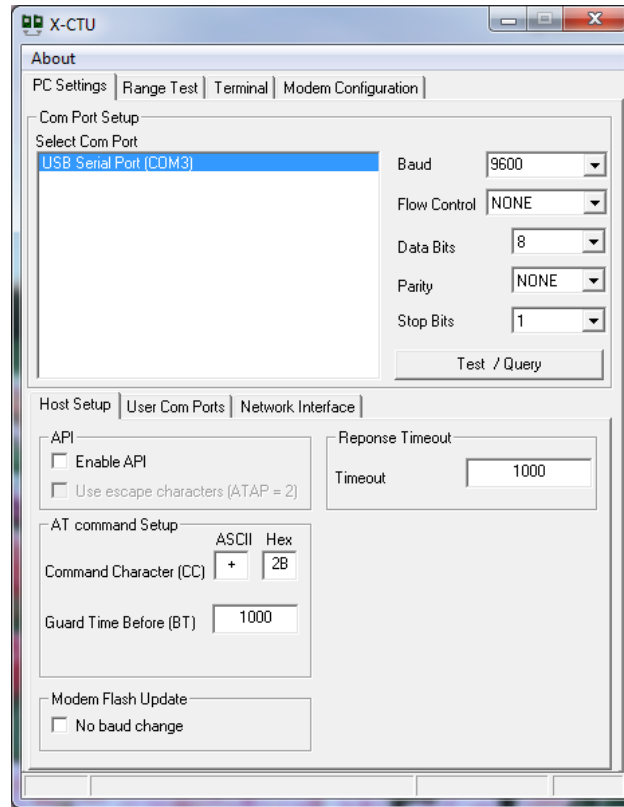


Figure 6: X-CTU Software Interface

Necessary parameters were defined based on the desired configuration by referring to the manual. The following figure shows the terminal of X-CTU which was used to program the Xbee wireless module. The syntax of the programming is referred from the manual provided by the manufacturer. Baud rate was set to be 115200, 8 data bits, no parity bit and 1 stop bit. The baud rate setting is calculated based on the requirement of sampling frequency which is set to be 100Hz, referring to [9]. The rest of the parameters are just standard configurations for typical COM port.

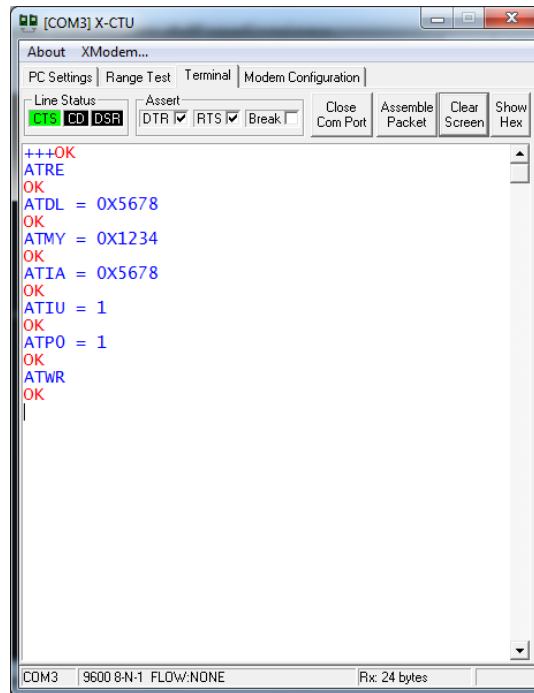


Figure 7: Terminal to Program the Xbee Module

All the configurations related to communication of the wireless modules are done here. The network topology and the address of each node has to be defined and correctly programmed. The instructions to program can be referred from the manual provided online by the manufacturer.

Testing of Capability of Transmitter and Receiver

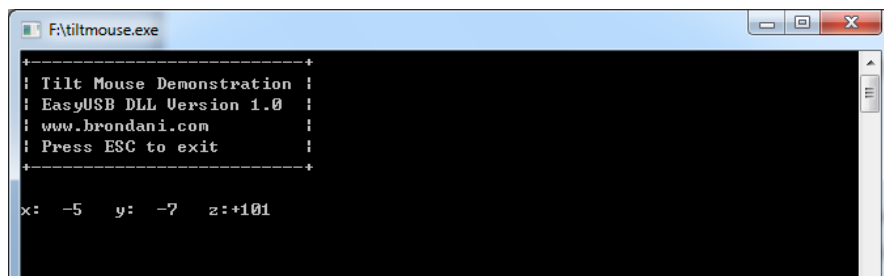


Figure 8: Sample Program to test the functionality of Sensor Node

A sample program which is written in C programming and is compatible with the hardware configuration of the prototype was used to test whether the data acquisition of the system is working correctly or not. The program verified that the sensor node is working as required. The x, y and z values which were sampled at 10ms in this program indicated the accelerations of the sensor node due to movement.

Summary of Configurations

Configuration	Properties
Serial communication	Baud rate = 115200, 8 data bits, no parity bit, 1 stop bit
Sampling rate / frequency	10ms / 100Hz
Max ADC level	512
Number of receiver	1
Number of sensor node (transmitter)	2
Network topology	Star, many-to-one, unidirectional, IEEE 802.15.0.4
Addressing	Receiver - 0x1234, Transmitter 1 - 0x5678, Transmitter 2 – 0x5778

3. Software Development

A program is developed using MATLAB in order to perform activity recognition. The program is divided into two major parts, (1) Training Module which is used to record sample data and to train the algorithm and (2) Testing Module to sample new data and classify data base on the trained algorithm. Database platform used is Microsoft Excel.

Training Module

Following are the major sections and processes in the Training Module:

- 1) Configurations and parameters setting
- 2) UART serial communication setup
- 3) Stripe off overhead and extract data
- 4) Data processing and features extraction
- 5) Save data to database

Parameters such as number of sensor node used, training or testing duration, activity to be performed (Training Module) is pre-defined by the user during the startup of the program as in Figure 9. The program will attempt to setup the serial communication through virtual COM port with the hardware. Once the connection is established the program will start to capture the acceleration data by extracting it from each packet of data. The raw acceleration data will then be stored and displayed at the same time on a figure, updating in real time as shown in Figure 11.

Each window of data contained 512 samples of acceleration data and is overlapped 50% by the previous window. The first window is discarded to exclude glitches occur in the first few seconds. The features for each window, for instance mean and standard deviation will be calculated. Excel database will then be dynamically generated to store the data (features of raw data). If the database is existed, data will be appended. The computation of features will be done for each window of acceleration data and the process of storing of data will only be done at the end of each training session in order to save the time for processing.

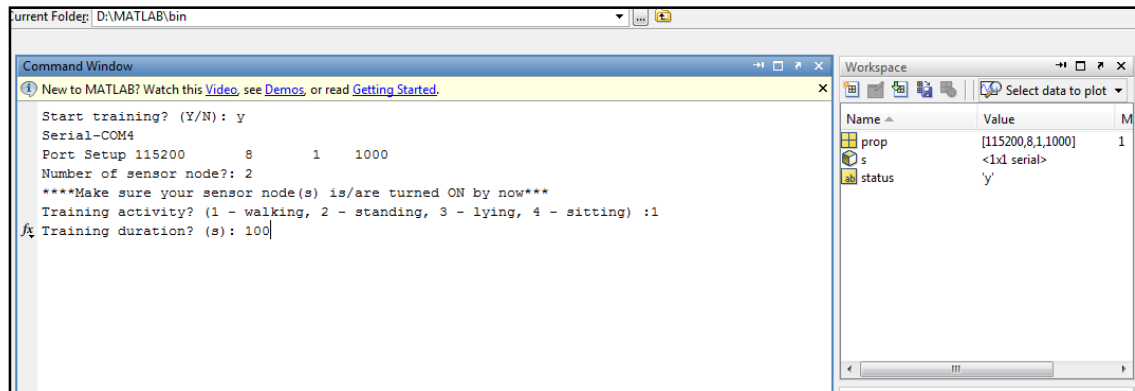


Figure 9: Initializing Training Module in MATLAB

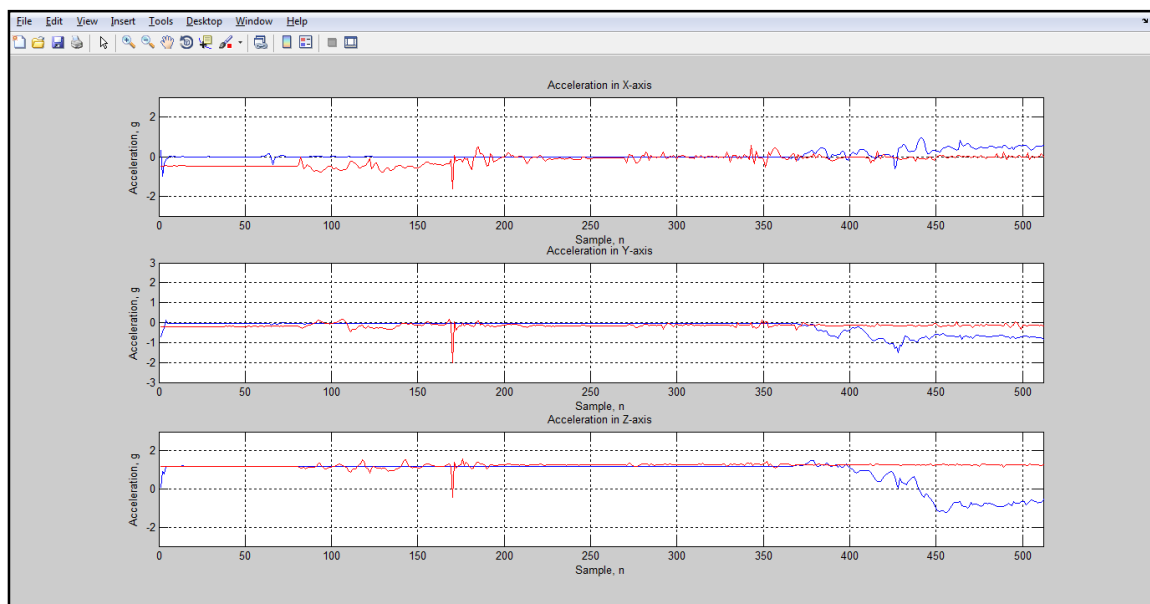


Figure 10: Graph displaying real time data acquisition in Testing Module when the subject is sitting

Testing Module

Following are the major sections and processes in the Testing Module:

- 1) Configurations and parameters setting
- 2) UART serial communication setup
- 3) Import data and algorithm
- 4) Stripe off overhead and extract data
- 5) Data processing and features extraction
- 6) Prediction of activity performed by subject

The initialization for Testing Module is similar to Training Module but with fewer parameters. The system has to be told the number of sensor node under operation and the time for the test like in Figure 11. The program will initially established the connection with the hardware. Once successful, it will then import the data from the database and generate the decision tree using Classification Decision Tree function. The decision tree can be shown in graphical model as in Figure 12. After that, the system will start to analyze the data gathered. For each window of data, the program will predict the activity and prompt the predicted results on the command window of MATLAB as shown in Figure 13. The result is time stamped and recorded.

Testing Module with GUI

The Testing Module is further developed using MATLAB GUI developing tools in order to ease the configuration and navigation of the software. The interactive interface of the module which has been developed is shown in Figure 14. With only two parameters to be configured, number of sensor node and duration of test, the Testing Module can be started. A human vector graphic will be showing the predicted activity based on the data acquired in real time. The Test GUI module works the same as the Testing Module.

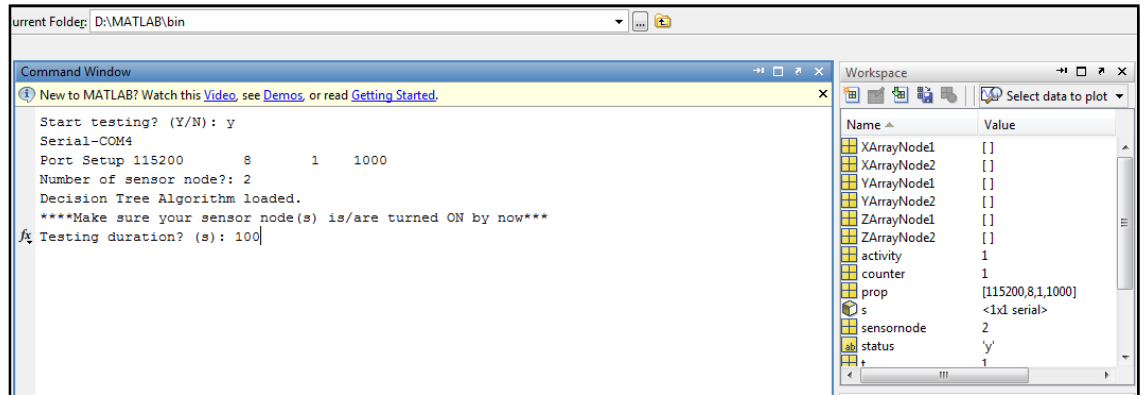


Figure 11: Initializing Testing Module in MATLAB

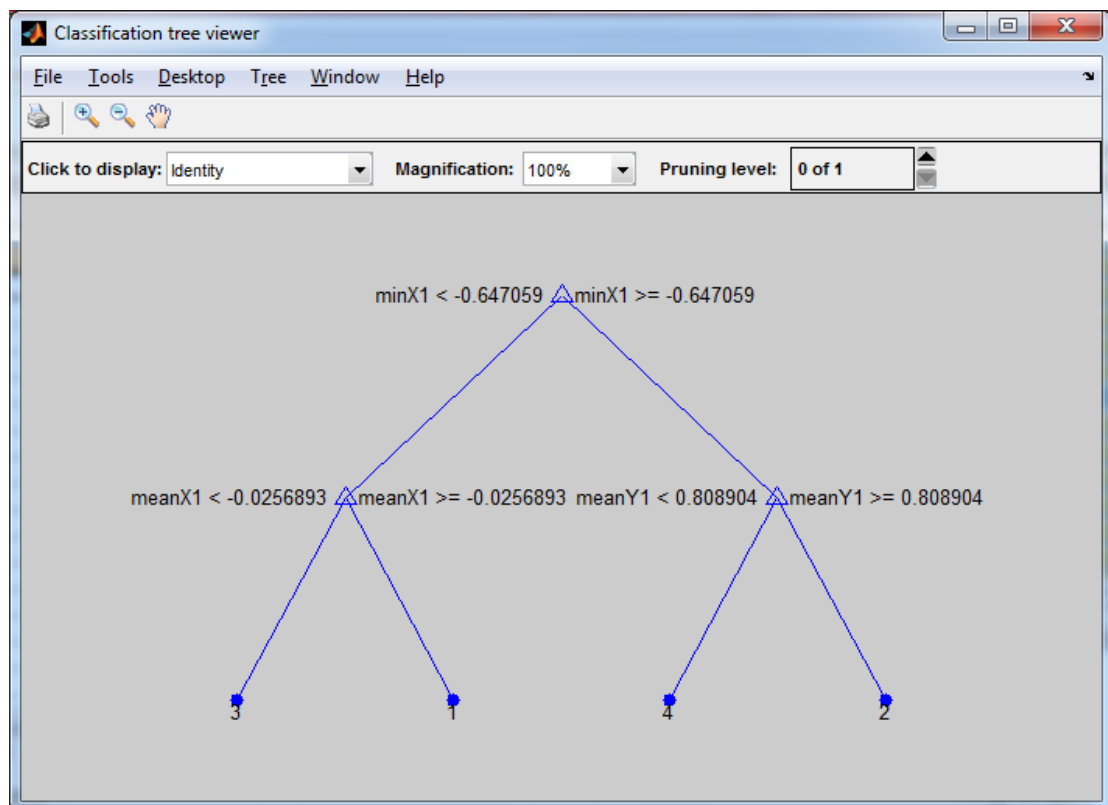


Figure 12: Graphical model of Classification Decision Tree (end nodes labelled as '1' stands for walking, '2' for standing, '3' for sitting and '4' for lying down)

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Start testing? (Y/N): y
Serial-COM4
Port Setup 115200      8      1      1000
Number of sensor node?: 1
Decision Tree Algorithm loaded.
****Make sure your sensor node(s) is/are turned ON by now****
Testing duration? (s): 100
Running....
Subject is walking
Subject is walking
Subject is walking
Subject is walking
Subject is walking
Subject is lying down
Subject is lying down
```

Figure 13: Real time activity recognition in Testing Module

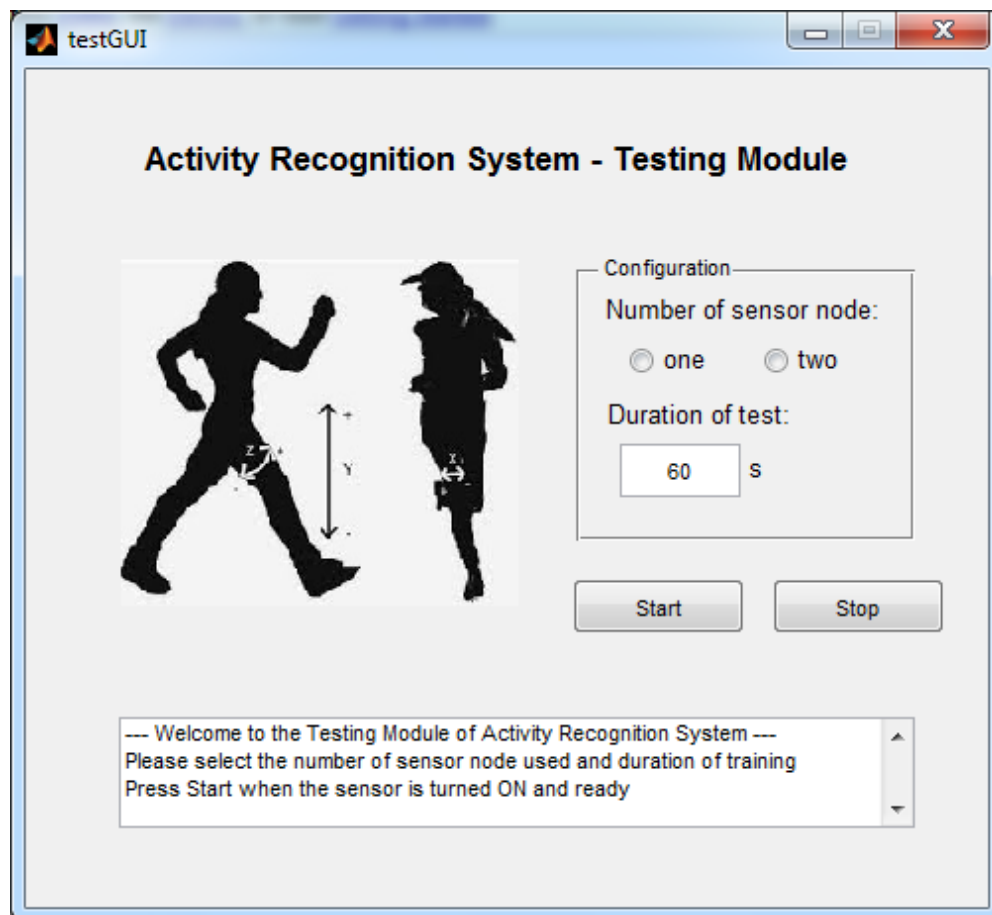


Figure 14: MATLAB GUI for Testing Module

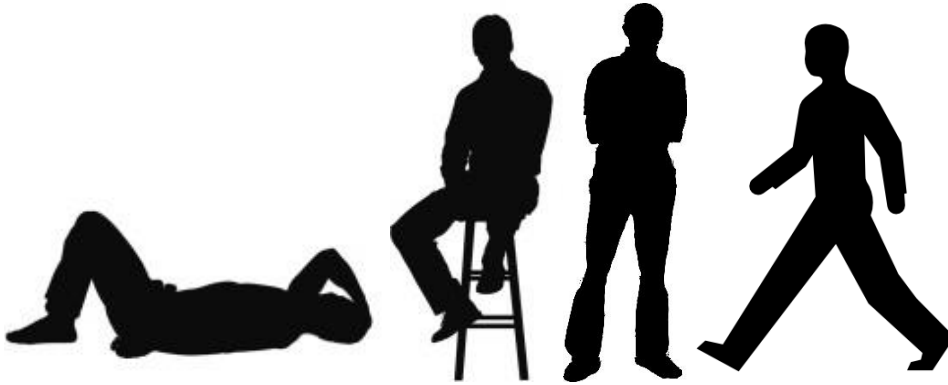


Figure 15: Human vector graphics to illustrate the predicted activity performs by subject in Test GUI software



Figure 16: Subject under test with two-sensor-node-system (thigh and ankle) for standing activity where sensors are secured in position with reusable bandage

4. Method of Data Classification

Description about Classification Decision Tree

Classification Decision Tree algorithm is used to perform classification of data [18]. The function of the algorithm is to produce a model which is able to determine the class of a data based on historical data or database which is pre-acquired. The Decision Tree can be illustrated as a flow-chart-like structure. The topmost node is called the root node, while each internal node represents a test on an attribute and a branch represents the outcome of a test. Each leaf node holds a class label. Decision Tree method is proved to be highly accurate in classifying the activities perform by subject as suggested in [9]. Decision Tree is created for predicting the response as a function of the predictors. Base on the values in the columns of predictors, each of the node of the binary tree is branched accordingly. In the actual implementation of the software, the algorithm to classify the data, Decision Tree function available in MATLAB is used where the model will be dynamically generated with the properly provided database.

Computation Technique and Feature Extraction

Data obtained from each observation is calculated from the raw data of one window (512 samples per window). Each window is overlapped 50% by the previous window, similar to the technique used in [9]. The first window is discarded to exclude glitches occur in the first few seconds. Instead of storing the raw data of the accelerations, features of each window of acceleration data are determined and be used as the attributes for the decision tree. Mean, standard deviation, maximum, minimum, correlation between axes and energy of the signal are the features extracted from each window of raw data. These features are then be used as the attributes for the node of decision tree.

5. Justification on the Accuracy of Recognition

The accuracy of the recognition system is defined by the following:

$$Accuracy (\%) = \frac{total\ no.\ of\ prediction - no.\ of\ incorrect\ prediction}{total\ no.\ of\ prediction} \times 100\%$$

where the number of prediction is determined by the duration of the testing session divide by the system respond interval due to processing.

6. Key Milestones

Following is the key milestones for the project, separated into two sections – Final Year Project I and Final Year Project II:

Number	Milestone Objective	Milestone Date
A1	Finalization of Topic and submission of proposal	31 Jan 2014
A2	Submission of Extended Proposal	21 Feb 2014
A3	Proposal Defense	14 Mar 2014
A4	Construction and testing of the prototype	28 Mar 2014
A5	Finalization of system architecture design and requirement	4 Apr 2014
A6	Submission of Interim Report draft	9 Apr 2014
A7	Submission of Interim Report	18 Apr 2014

Table 4: Key Milestones for FYP 1

Number	Milestone Objective	Milestone Date
B1	Software program development	13 Jun 2014
B2	Software feasibility test	27 Jun 2014
B3	Progress Report submission	07 Jul 2014
B4	Completion of data and result analysis	14 Jul 2014
B5	ELECTREX presentation	23 Jul 2014
B6	SEDEX presentation	6 Aug 2014
B7	Draft of Final Report	11 Aug 2014
B8	Submission of Final Report	18 Aug 2014
B9	Viva	27 Aug 2014

Table 5: Key Milestones for FYP 2

7. Gantt Chart

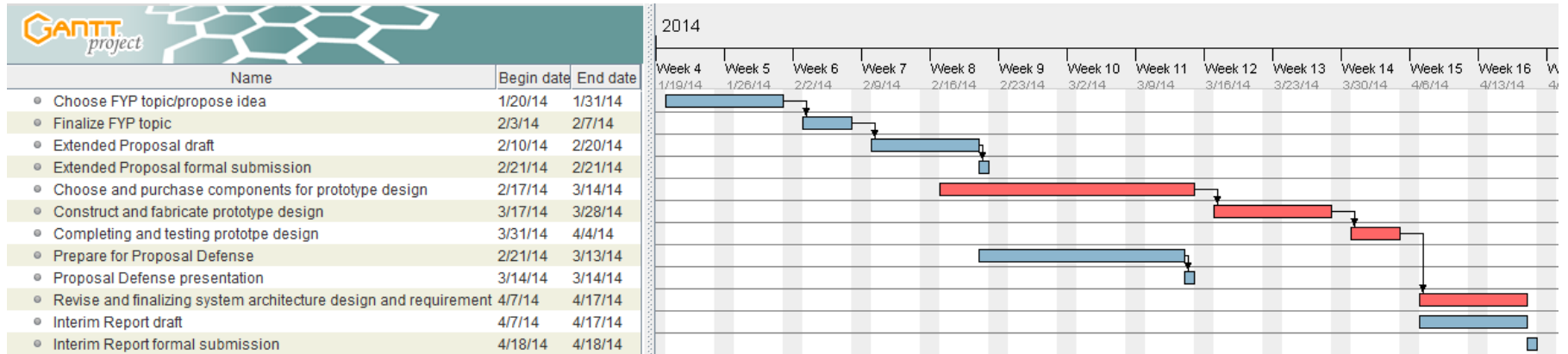


Figure 17: Gantt chart for FYP 1

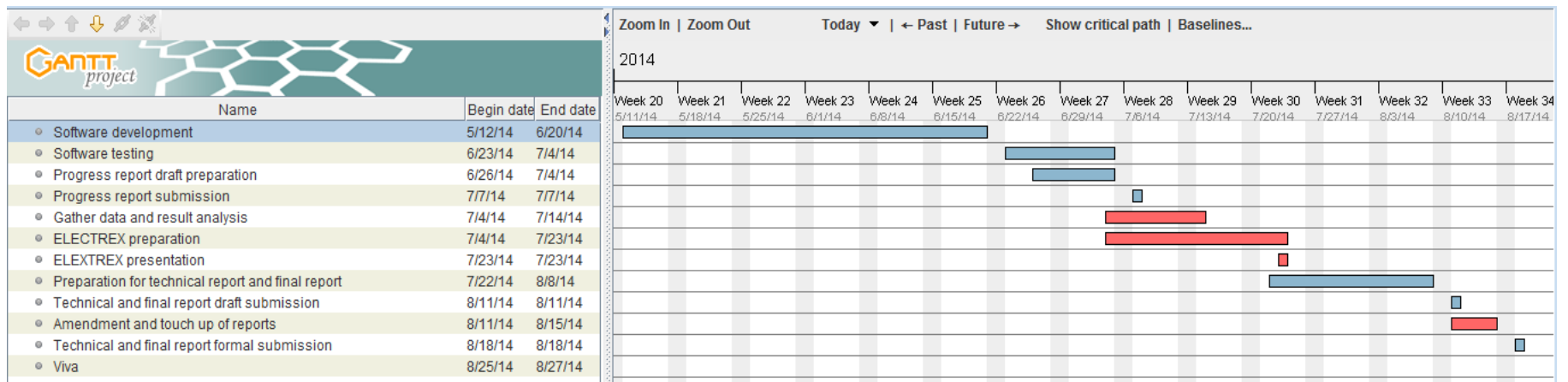


Figure 18: Gantt chart for FYP 2

CHAPTER 4

RESULTS & DISCUSSION

1. One-Sensor-Node Activity Recognition System

The accuracy of activity recognition system with only one sensor node is being examined. The sensor node is placed at different positions of the body in order to compare the difference in accuracy. Each testing session is 100 seconds. Following are the positions which have been experimented:

- 1) Chest
- 2) Left / right wrist
- 3) Left / right thigh

There is no significant variation of accuracy when the sensor node is placed on left side or right side of the examined position (for example accuracy of recognition on left wrist compared to accuracy of recognition on right wrist), therefore, they are categorized into the same class.

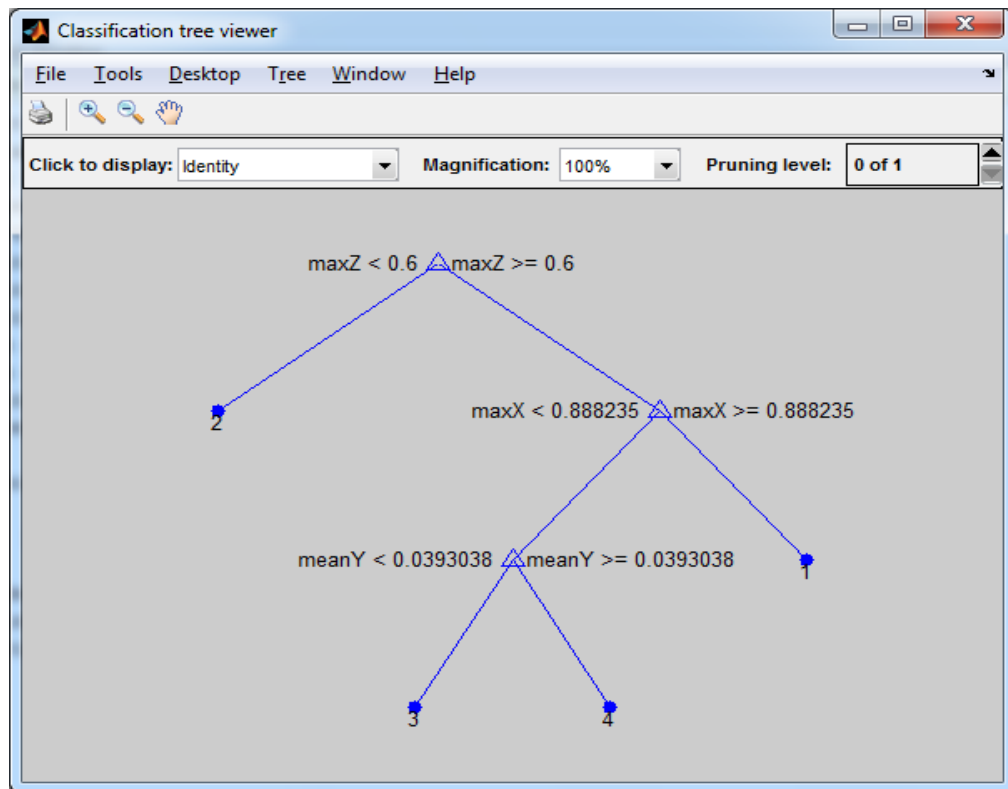


Figure 19: Classification Tree for one-sensor-node system on thigh (end nodes labelled as '1' stands for walking, '2' for standing, '3' for sitting and '4' for lying down)

In Figure 16, ‘1’ is walking activity, ‘2’ is standing activity, ‘3’ is sitting activity and ‘4’ is lying down. As shown in the classification tree diagram, the main features that classified the activities are maximum acceleration in Z-axis, maximum acceleration in X-axis and mean acceleration in Y-axis. Standing activity is distinguished by the acceleration in Z-axis ($\max Z < 0.6$), while walking activity is determined from the second level of the tree and by the maximum value of X-axis ($\max Z \geq 0.6$ and $\max X \geq 0.888235$). The other two activities are determined at the third level, mean value of acceleration in Y-axis ($\text{mean} Y < 0.0393038$ for sitting and $\text{mean} Y \geq 0.0393038$ for lying down).

Following are the results of overall accuracy:

Position of Sensor	Accuracy for each activity	Overall Accuracy	Remarks
Chest	Walking – 100% Standing – 18.18% Sitting – 12.12% Lying down – 100%	57.58%	System confused between Standing and Sitting
Left / right wrist	Walking – 100% Standing – 100% Sitting – 21.21% Lying down – 27.27%	62.12%	System confused between Sitting and Lying down
Left / right thigh	Walking – 100% Standing – 100% Sitting – 36.36% Lying down – 33.33%	67.42%	System confused between Sitting and Lying down

Table 6: One-sensor-node System Result

The overall accuracy is calculated by averaging the accuracy of recognition for all four activities.

2. Two-Sensor-Nodes Activity Recognition System

Features extraction between one-sensor-node system and two-sensor-nodes system are similar except that more features are being extracted. With two sensors, there are twelve correlations between axes and double of each feature like mean, standard deviation, maximum, minimum and energy. Each activity is performed for 100 seconds in consecutive (33 observations). The rest of the system is the similar with the previous system.

The accuracy of activity recognition system with two sensor nodes is being examined. The sensor nodes are placed at different positions of the body in order to compare the difference in accuracy. Each testing session is 100 seconds. Following are the positions which have been experimented up to date:

- 1) Thigh & ankle
- 2) Thigh & chest

Following are the results of accuracy:

Position of Sensor	Accuracy for each activity	Overall Accuracy	Remarks
Thigh & ankle	Walking – 100% Standing – 93.94% Sitting – 84.85% Lying down – 100%	94.70%	Minor error for Standing and Sitting activity.
Thigh & chest	Walking – 100% Standing – 84.85% Sitting – 90.91% Lying down – 87.88%	90.91%	Minor error for Standing, Sitting and Lying down activity.

Table 7: Two-sensor-nodes System Result

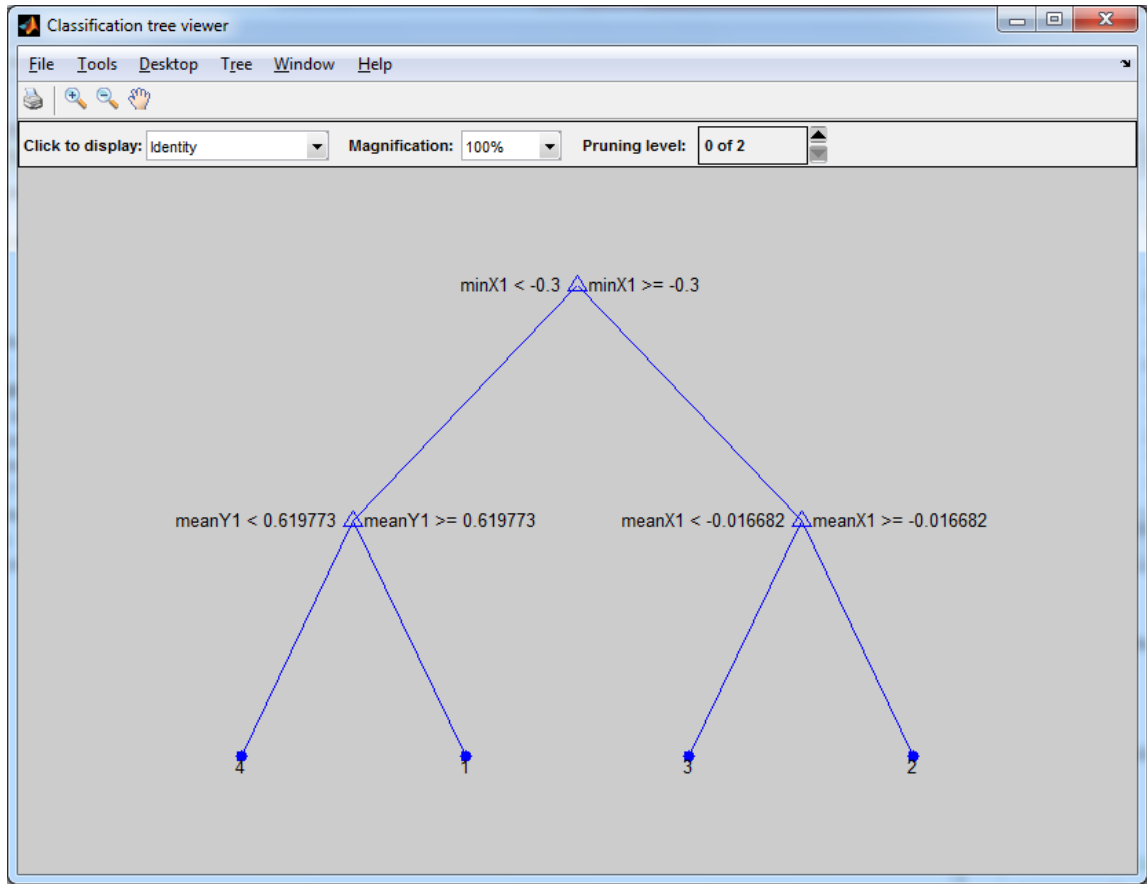


Figure 20: Classification Tree for two-sensor-nodes system on thigh and ankle (end nodes labelled as ‘1’ stands for walking, ‘2’ for standing, ‘3’ for sitting and ‘4’ for lying down)

In Figure 17, ‘1’ is walking activity, ‘2’ is standing activity, ‘3’ is sitting activity and ‘4’ is lying down. As shown in the classification tree diagram, the main features that classified the activities are minimum acceleration in X-axis of sensor node 1, mean acceleration in Y-axis of sensor node 1 and mean acceleration in X-axis of sensor node 1. Walking and lying down activity are distinguished by the minimum acceleration in X-axis of sensor node 1 and mean acceleration in Y-axis of sensor node 1 (walking - $\text{minX1} < -0.3$ and $\text{meanY1} \geq 0.619773$; lying down - $\text{minX1} < -0.3$ and $\text{meanY1} < 0.619773$), while standing activity is determined by the minimum acceleration in X-axis of sensor node 1 ($\text{minX1} > -0.3$) and mean acceleration in X-axis of sensor node 1 (standing - $\text{minX1} > -0.3$ and $\text{meanY1} \geq -0.016682$; sitting - $\text{minX1} > -0.3$ and $\text{meanY1} < -0.016682$).

3. Comparison between two systems and justification

Based on the results, one sensor node is evidently has flaws especially in detection static activity such as sitting, standing and lying. Since the variation in terms of the overall acceleration and energy of the signal between one activity and another for the abovementioned activities is small, therefore, many of the features are not sufficiently distinctive to be used by the Decision Tree node to predict the activity. As the acceleration towards gravity is not offset, the acceleration data can be interpreted to represent the orientation of sensor node. Therefore, the main and distinct features for the Decision Tree model for most of the static activities are mean, maximum and minimum of the acceleration in one of the direction (x, y or z), showing the acceleration towards ground. Yet, the accuracy of the system is still lower due to limited relevant features. For example when the sensor node is placed on chest, standing and sitting might be confused by the system if the subject's sitting position is up straight, very similar to his/her standing pose. The rest of the inaccuracy of the one-sensor-node-system is due to the similar issues as noted in the remark section of Table 6.

For system with two sensor nodes is more reliable because there are 46 features extracted from the data of two sensors, as compared to the previous one which has got only 20 features. The features include correlation between the acceleration data of one sensor node to another. The energy of the signal at different location of the sensor is also one of the distinct features which contribute to accurate classification of activity. Although there are still some minor inaccuracy of activity classification due to human error and other unaccountable errors, the overall accuracy is as high as 94.70% for sensors on thigh-ankle combination.

CHAPTER 5: CONCLUSION & FUTURE WORKS

The activity recognition system prototype together with the software to demonstrate the system had been successfully developed. With one-sensor-node system, the system achieved accuracy of recognition of only about 67.42%. The system is then improved by adding one more sensor node, which proved to improve the accuracy but another 27.28%. The weight of each sensor node is less than 55g each and the cost of the prototype which uses only off-the-shelf products cost less than RM500. In conclusion, the objective of the project which is to design a lightweight and activity recognition system with high accuracy and is suitable to be used by elderly people, is achieved.

Nevertheless, there are numerous areas of improvement which are recommended to be future works for this project. Area such as health related concerns is yet to be explored. Implication on health such as ergonomic problem has to be carefully studied so that it does not caused any health problem in oppose to the aim to help the elders. Choice of power supply can also be studied can considered as the current design uses standard AA battery as power supply for simplicity. With better choice of power supply, the sensor node can be further miniaturized and weight can also be reduced. Furthermore, power consumption can also be further optimized by manipulating the refresh time of the system. As current design predict the activity at 3 seconds, the power consumption can be further reduced if the interval for each observation is increased without compromising the overall accuracy of the system with careful research and study. The sensor node can also be programmed to go into sleep mode if necessary.

REFERENCES

- [1] (2012, 02/02/2014). *Falls*. Available: <http://www.who.int/mediacentre/factsheets/fs344/en/>
- [2] (03/02/2014). *Definition of an older or elderly person*. Available: <http://www.who.int/healthinfo/survey/ageingdefnolder/en/>
- [3] U. S. C. Bureau, "The Older Population: 2010," U.S. Department of Commerce C2010BR-09, 2011.
- [4] H. I. S. C. Adelaide. (2013, 11/02/2014). *Australian Seniors: The Facts & Statistics*. Available: <http://zee.homeinstead.com.au/adelaideeast/australian-seniors-facts-statistics>
- [5] J. Tai and L. L. K, "Seniors living alone may rise to 83,000 by 2030," in *Straits Times*, ed. Singapore: Singapore Press Holdings.
- [6] C. Otto, A. Milenkovi, #263, C. Sanders, and E. Jovanov, "System architecture of a wireless body area sensor network for ubiquitous health monitoring," *J. Mob. Multimed.*, vol. 1, pp. 307-326, 2005.
- [7] O. s. D. Lara and M. A. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors."
- [8] E. Tapia, S. Intille, and K. Larson, "Activity Recognition in the Home Using Simple and Ubiquitous Sensors," in *Pervasive Computing*. vol. 3001, A. Ferscha and F. Mattern, Eds., ed: Springer Berlin Heidelberg, 2004, pp. 158-175.
- [9] L. Bao and S. Intille, "Activity Recognition from User-Annotated Acceleration Data," in *Pervasive Computing*. vol. 3001, A. Ferscha and F. Mattern, Eds., ed: Springer Berlin Heidelberg, 2004, pp. 1-17.
- [10] P. Lukowicz, J. Ward, H. Junker, M. Stäger, G. Tröster, A. Atrash, *et al.*, "Recognizing Workshop Activity Using Body Worn Microphones and Accelerometers," in *Pervasive Computing*. vol. 3001, A. Ferscha and F. Mattern, Eds., ed: Springer Berlin Heidelberg, 2004, pp. 18-32.
- [11] G. Roos. (2010, 14/02/2014). *Accelerometers get smaller, less power hungry, and cheaper*. Available: <http://www.digikey.com/en-US/articles/techzone/2010/may/accelerometers-get-smaller-less-power-hungry-and-cheaper>
- [12] D. Yamane, T. Konishi, T. Matsushima, K. Machida, H. Toshiyoshi, and K. Masu, "Design of sub-1g microelectromechanical systems accelerometers)," *Applied Physics Letters*, vol. 104, pp. -, 2014.
- [13] X. Wenyao, Z. Mi, A. A. Sawchuk, and M. Sarrafzadeh, "Co-recognition of Human Activity and Sensor Location via Compressed Sensing in Wearable Body Sensor Networks," in *Wearable and Implantable Body Sensor Networks (BSN), 2012 Ninth International Conference on*, 2012, pp. 124-129.
- [14] Z. Lin, M. Sinclair, and R. Bittner, "A phone-centered body sensor network platform cost, energy efficiency & user interface," in *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, 2006, pp. 4 pp.-182.
- [15] R. Jafari, A. Encarnacao, A. Zahoory, F. Dabiri, H. Noshadi, and M. Sarrafzadeh, "Wireless sensor networks for health monitoring," in *Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on*, 2005, pp. 479-481.

- [16] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, 2006, pp. 4 pp.-116.
- [17] L. Jin-Shyan, S. Yu-Wei, and S. Chung-Chou, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, 2007, pp. 46-51.
- [18] Y. Yohannes and J. Hoddinott, "Classification and regression trees: an introduction," *International Food Policy Research Institute*, vol. 2033, 1999.

APPENDICES

1. Form to be filled by candidate of experiment

ACTIVITY RECOGNITION SYSTEM USING ACCELEROMETER

This form is to be filled up by candidate who has carried out the experiment of Activity Recognition System Using Accelerometer.

Basic information of candidate:

Name : _____

Age : ____

Gender: M / F

Weight: _____ kg

Height : _____ cm

Experiment carried out:

1. System used:

- ☐ One-sensor-node system
☐ Two-sensor-nodes system

2. Placement of sensor(s):

For one-sensor-node system: -

- ☐ Ankle ☐ Chest ☐ Thigh ☐ Waist

For two-sensor-nodes system: -

- ☐ Ankle & thigh ☐ Waist & thigh
☐ Waist & ankle ☐ Chest & Thigh

3. Training duration:

- ☐ < 1 min ☐ 1 to 3 mins ☐ 3 to 5 mins ☐ > 5 mins

4. Testing duration:

- ☐ < 1 min ☐ 1 to 3 mins ☐ 3 to 5 mins ☐ > 5 mins

5. Setting of experiment:

- ☐ indoor ☐ outdoor Specify: _____

6. Comment/suggestion/feedback:

—

—

—

Prepared by: Dandy (2014)

2. Certificate of Participation in 33rd Science & Engineering Design Exhibition



3. Data Sheet of Accelerometers – ADXL335

4. Data Sheet of Xbee Transceiver

(Refer to the next page)

FEATURES

3-axis sensing
Small, low profile package
 4 mm \times 4 mm \times 1.45 mm LFCSP
Low power : 350 μ A (typical)
Single-supply operation: 1.8 V to 3.6 V
10,000 g shock survival
Excellent temperature stability
BW adjustment with a single capacitor per axis
RoHS/WEEE lead-free compliant

APPLICATIONS

Cost sensitive, low power, motion- and tilt-sensing applications
Mobile devices
Gaming systems
Disk drive protection
Image stabilization
Sports and health devices

GENERAL DESCRIPTION

The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of $\pm 3 g$. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

The user selects the bandwidth of the accelerometer using the C_X , C_Y , and C_Z capacitors at the X_{OUT} , Y_{OUT} , and Z_{OUT} pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for the X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.

The ADXL335 is available in a small, low profile, 4 mm \times 4 mm \times 1.45 mm, 16-lead, plastic lead frame chip scale package (LFCSP_LQ).

FUNCTIONAL BLOCK DIAGRAM

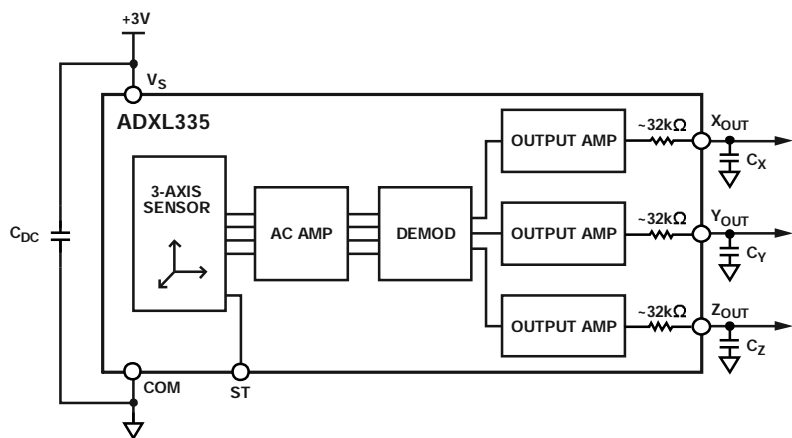


Figure 1.

Rev. 0

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
 Tel: 781.329.4700
 Fax: 781.461.3113
www.analog.com
 ©2009 Analog Devices, Inc. All rights reserved.

SPECIFICATIONS

$T_A = 25^\circ\text{C}$, $V_S = 3\text{ V}$, $C_X = C_Y = C_Z = 0.1\text{ }\mu\text{F}$, acceleration = 0 g, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range		± 3	± 3.6		g
Nonlinearity	% of full scale		± 0.3		%
Package Alignment Error			± 1		Degrees
Interaxis Alignment Error			± 0.1		Degrees
Cross-Axis Sensitivity ¹			± 1		%
SENSITIVITY (RATIOMETRIC) ²	Each axis				
Sensitivity at X_{OUT} , Y_{OUT} , Z_{OUT}	$V_S = 3\text{ V}$	270	300	330	mV/g
Sensitivity Change Due to Temperature ³	$V_S = 3\text{ V}$		± 0.01		%/ $^\circ\text{C}$
ZERO g BIAS LEVEL (RATIOMETRIC)					
0 g Voltage at X_{OUT} , Y_{OUT}	$V_S = 3\text{ V}$	1.35	1.5	1.65	V
0 g Voltage at Z_{OUT}	$V_S = 3\text{ V}$	1.2	1.5	1.8	V
0 g Offset vs. Temperature			± 1		mg/ $^\circ\text{C}$
NOISE PERFORMANCE					
Noise Density X_{OUT} , Y_{OUT}			150		$\mu\text{g}/\sqrt{\text{Hz}}$ rms
Noise Density Z_{OUT}			300		$\mu\text{g}/\sqrt{\text{Hz}}$ rms
FREQUENCY RESPONSE ⁴					
Bandwidth X_{OUT} , Y_{OUT} ⁵	No external filter		1600		Hz
Bandwidth Z_{OUT} ⁵	No external filter		550		Hz
R_{FILT} Tolerance			$32 \pm 15\%$		k Ω
Sensor Resonant Frequency			5.5		kHz
SELF-TEST ⁶					
Logic Input Low			+0.6		V
Logic Input High			+2.4		V
ST Actuation Current			+60		μA
Output Change at X_{OUT}	Self-Test 0 to Self-Test 1	-150	-325	-600	mV
Output Change at Y_{OUT}	Self-Test 0 to Self-Test 1	+150	+325	+600	mV
Output Change at Z_{OUT}	Self-Test 0 to Self-Test 1	+150	+550	+1000	mV
OUTPUT AMPLIFIER					
Output Swing Low	No load		0.1		V
Output Swing High	No load		2.8		V
POWER SUPPLY					
Operating Voltage Range		1.8		3.6	V
Supply Current	$V_S = 3\text{ V}$		350		μA
Turn-On Time ⁷	No external filter		1		ms
TEMPERATURE					
Operating Temperature Range		-40		+85	$^\circ\text{C}$

¹ Defined as coupling between any two axes.

² Sensitivity is essentially ratiometric to V_S .

³ Defined as the output change from ambient-to-maximum temperature or ambient-to-minimum temperature.

⁴ Actual frequency response controlled by user-supplied external filter capacitors (C_X , C_Y , C_Z).

⁵ Bandwidth with external capacitors = $1/(2 \times \pi \times 32\text{ k}\Omega \times C)$. For C_X , $C_Y = 0.003\text{ }\mu\text{F}$, bandwidth = 1.6 kHz. For $C_Z = 0.01\text{ }\mu\text{F}$, bandwidth = 500 Hz. For C_X , C_Y , $C_Z = 10\text{ }\mu\text{F}$, bandwidth = 0.5 Hz.

⁶ Self-test response changes cubically with V_S .

⁷ Turn-on time is dependent on C_X , C_Y , C_Z and is approximately $160 \times C_X$ or C_Y or $C_Z + 1\text{ ms}$, where C_X , C_Y , C_Z are in microfarads (μF).

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
Acceleration (Any Axis, Unpowered)	10,000 <i>g</i>
Acceleration (Any Axis, Powered)	10,000 <i>g</i>
<i>V</i> _S	−0.3 V to +3.6 V
All Other Pins	(COM − 0.3 V) to (<i>V</i> _S + 0.3 V)
Output Short-Circuit Duration (Any Pin to Common)	Indefinite
Temperature Range (Powered)	−55°C to +125°C
Temperature Range (Storage)	−65°C to +150°C

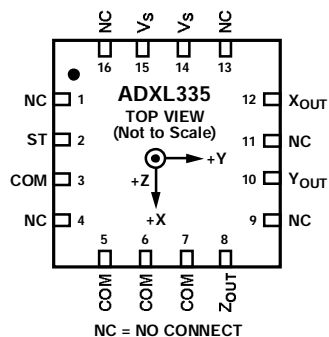
Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ESD CAUTION



ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS



NOTES
1. EXPOSED PAD IS NOT INTERNALLY CONNECTED BUT SHOULD BE SOLDERED FOR MECHANICAL INTEGRITY.

07808-003

Figure 2. Pin Configuration

Table 3. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	NC	No Connect ¹ .
2	ST	Self-Test.
3	COM	Common.
4	NC	No Connect ¹ .
5	COM	Common.
6	COM	Common.
7	COM	Common.
8	Z _{OUT}	Z Channel Output.
9	NC	No Connect ¹ .
10	Y _{OUT}	Y Channel Output.
11	NC	No Connect ¹ .
12	X _{OUT}	X Channel Output.
13	NC	No Connect ¹ .
14	V _S	Supply Voltage (1.8 V to 3.6 V).
15	V _S	Supply Voltage (1.8 V to 3.6 V).
16	NC	No Connect ¹ .
EP	Exposed Pad	Not internally connected. Solder for mechanical integrity.

¹NC pins are not internally connected and can be tied to COM pins, unless otherwise noted.

THEORY OF OPERATION

The ADXL335 is a complete 3-axis acceleration measurement system. The ADXL335 has a measurement range of ± 3 g minimum. It contains a polysilicon surface-micromachined sensor and signal conditioning circuitry to implement an open-loop acceleration measurement architecture. The output signals are analog voltages that are proportional to acceleration. The accelerometer can measure the static acceleration of gravity in tilt-sensing applications as well as dynamic acceleration resulting from motion, shock, or vibration.

The sensor is a polysilicon surface-micromachined structure built on top of a silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and plates attached to the moving mass. The fixed plates are driven by 180° out-of-phase square waves. Acceleration deflects the moving mass and unbalances the differential capacitor resulting in a sensor output whose amplitude is proportional to acceleration. Phase-sensitive demodulation techniques are then used to determine the magnitude and direction of the acceleration.

The demodulator output is amplified and brought off-chip through a $32\text{ k}\Omega$ resistor. The user then sets the signal bandwidth of the device by adding a capacitor. This filtering improves measurement resolution and helps prevent aliasing.

MECHANICAL SENSOR

The ADXL335 uses a single structure for sensing the X, Y, and Z axes. As a result, the three axes' sense directions are highly orthogonal and have little cross-axis sensitivity. Mechanical misalignment of the sensor die to the package is the chief source of cross-axis sensitivity. Mechanical misalignment can, of course, be calibrated out at the system level.

PERFORMANCE

Rather than using additional temperature compensation circuitry, innovative design techniques ensure that high performance is built in to the ADXL335. As a result, there is no quantization error or nonmonotonic behavior, and temperature hysteresis is very low (typically less than 3 mg over the -25°C to $+70^\circ\text{C}$ temperature range).

ADXL335

The ADXL335 output is ratiometric, therefore, the output sensitivity (or scale factor) varies proportionally to the supply voltage. At $V_s = 3.6$ V, the output sensitivity is typically 360 mV/g. At $V_s = 2$ V, the output sensitivity is typically 195 mV/g.

The zero g bias output is also ratiometric, thus the zero g output is nominally equal to $V_s/2$ at all supply voltages.

The output noise is not ratiometric but is absolute in volts; therefore, the noise density decreases as the supply voltage increases. This is because the scale factor (mV/g) increases while the noise voltage remains constant. At $V_s = 3.6$ V, the X-axis and Y-axis noise density is typically $120 \mu\text{g}/\sqrt{\text{Hz}}$, whereas at $V_s = 2$ V, the X-axis and Y-axis noise density is typically $270 \mu\text{g}/\sqrt{\text{Hz}}$.

Self-test response in g is roughly proportional to the square of the supply voltage. However, when ratiometricity of sensitivity is factored in with supply voltage, the self-test response in volts is roughly proportional to the cube of the supply voltage. For example, at $V_s = 3.6$ V, the self-test response for the ADXL335 is approximately -560 mV for the X-axis, +560 mV for the Y-axis, and +950 mV for the Z-axis.

At $V_s = 2$ V, the self-test response is approximately -96 mV for the X-axis, +96 mV for the Y-axis, and -163 mV for the Z-axis.

The supply current decreases as the supply voltage decreases. Typical current consumption at $V_s = 3.6$ V is 375 μA , and typical current consumption at $V_s = 2$ V is 200 μA .

AXES OF ACCELERATION SENSITIVITY

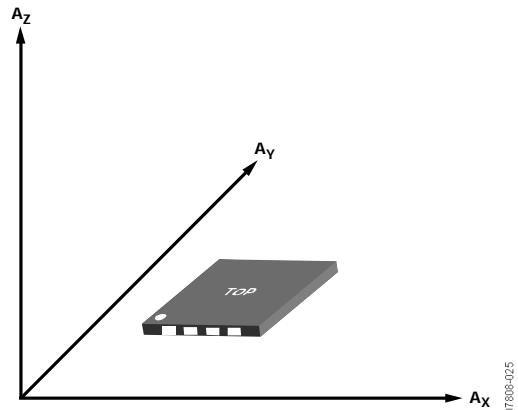


Figure 23. Axes of Acceleration Sensitivity; Corresponding Output Voltage Increases When Accelerated Along the Sensitive Axis.

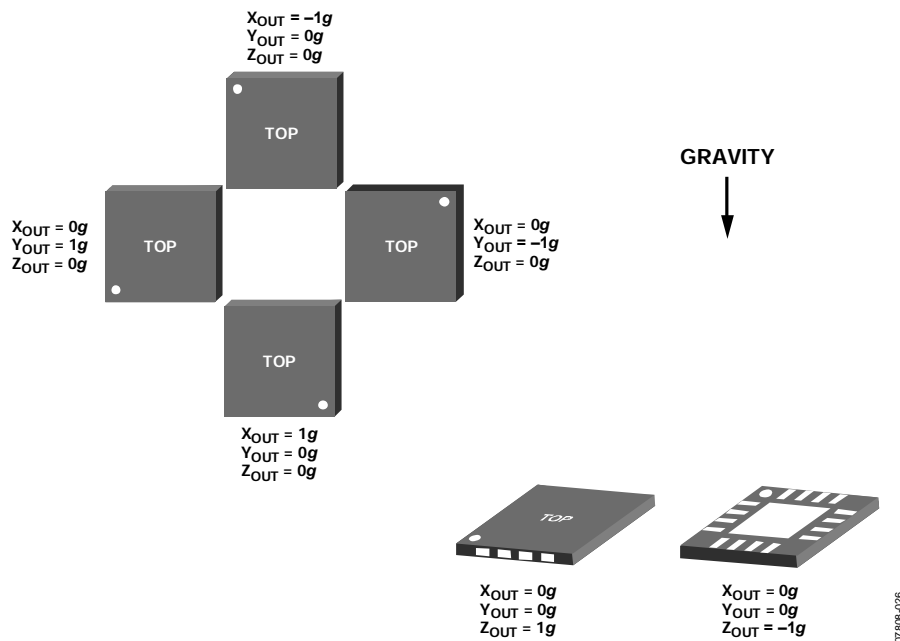


Figure 24. Output Response vs. Orientation to Gravity

ADXL335

OUTLINE DIMENSIONS

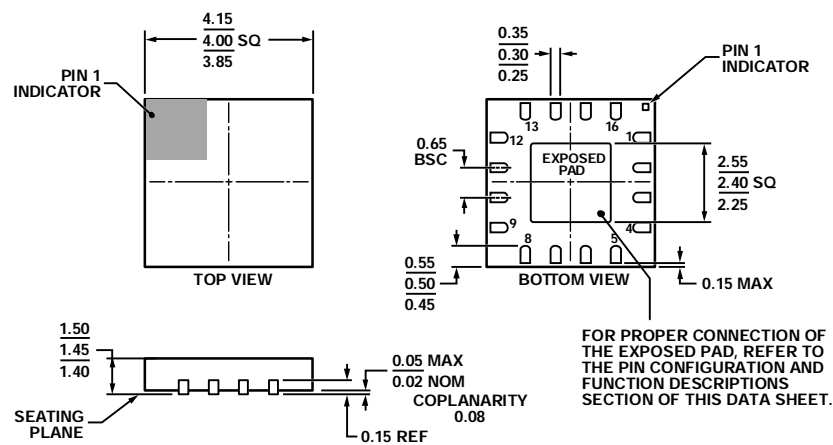


Figure 27. 16-Lead Lead Frame Chip Scale Package [LFCSP_LQ]
4 mm × 4 mm Body, 1.45 mm Thick Quad
(CP-16-14)
Dimensions shown in millimeters

ORDERING GUIDE

Model	Measurement Range	Specified Voltage	Temperature Range	Package Description	Package Option
ADXL335BCPZ ¹	±3 g	3 V	−40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
ADXL335BCPZ-RL ¹	±3 g	3 V	−40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
ADXL335BCPZ-RL7 ¹	±3 g	3 V	−40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
EVAL-ADXL335Z ¹				Evaluation Board	

¹ Z = RoHS Compliant Part.

1. XBee®/XBee-PRO® RF Modules

The XBee and XBee-PRO RF Modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices.

The modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other.



Key Features

Long Range Data Integrity

XBee

- Indoor/Urban: up to 100' (30 m)
- Outdoor line-of-sight: up to 300' (90 m)
- Transmit Power: 1 mW (0 dBm)
- Receiver Sensitivity: -92 dBm

XBee-PRO

- Indoor/Urban: up to 300' (90 m), 200' (60 m) for International variant
- Outdoor line-of-sight: up to 1 mile (1600 m), 2500' (750 m) for International variant
- Transmit Power: 63mW (18dBm), 10mW (10dBm) for International variant
- Receiver Sensitivity: -100 dBm

RF Data Rate: 250,000 bps

Advanced Networking & Security

Retries and Acknowledgements
DSSS (Direct Sequence Spread Spectrum)
Each direct sequence channels has over 65,000 unique network addresses available
Source/Destination Addressing
Unicast & Broadcast Communications
Point-to-point, point-to-multipoint and peer-to-peer topologies supported

Low Power

XBee

- TX Peak Current: 45 mA (@3.3 V)
- RX Current: 50 mA (@3.3 V)
- Power-down Current: < 10 μ A

XBee-PRO

- TX Peak Current: 250mA (150mA for international variant)
- TX Peak Current (RPSMA module only): 340mA (180mA for international variant)
- RX Current: 55 mA (@3.3 V)
- Power-down Current: < 10 μ A

ADC and I/O line support

Analog-to-digital conversion, Digital I/O
I/O Line Passing

Easy-to-Use

No configuration necessary for out-of box RF communications
Free X-CTU Software
(Testing and configuration software)
AT and API Command Modes for configuring module parameters
Extensive command set
Small form factor

Worldwide Acceptance

FCC Approval (USA) Refer to Appendix A [p64] for FCC Requirements.
Systems that contain XBee®/XBee-PRO® RF Modules inherit Digi Certifications.

ISM (Industrial, Scientific & Medical) **2.4 GHz frequency band**

Manufactured under **ISO 9001:2000** registered standards

XBee®/XBee-PRO® RF Modules are optimized for use in the United States, Canada, Australia, Japan, and Europe. Contact Digi for complete list of government agency approvals.



Specifications

Table 1-01. Specifications of the XBee®/XBee-PRO® RF Modules

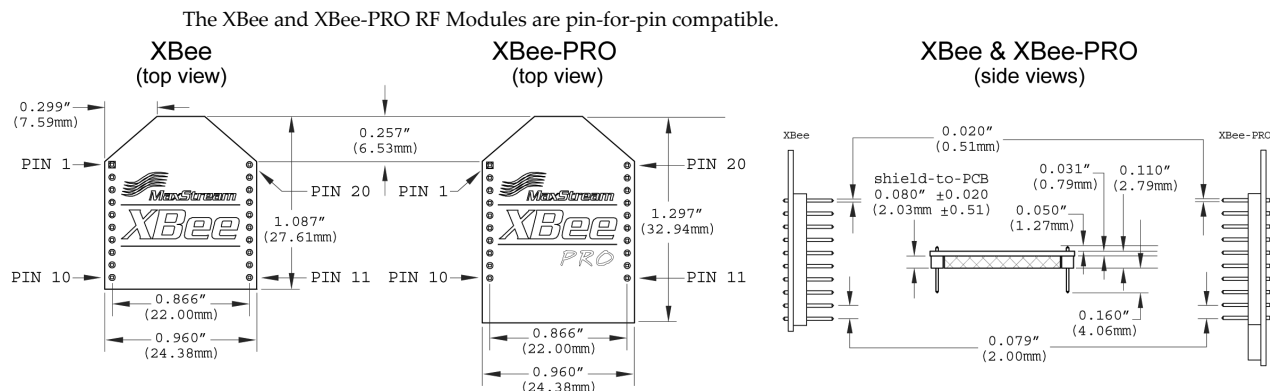
Specification	XBee	XBee-PRO
Performance		
Indoor/Urban Range	Up to 100 ft (30 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) International variant
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	1mW (0 dBm)	63mW (18dBm)* 10mW (10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 µA	< 10 µA
General		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RPSMA Connector	Integrated Whip, Chip or U.FL Connector, RPSMA Connector
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses
Agency Approvals		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)*
Japan	R201WW07215214	R201WW08215111 (Max. 10 dBm transmit power output)*
Australia	C-Tick	C-Tick

* See Appendix A for region-specific certification requirements.

Antenna Options: The ranges specified are typical when using the integrated Whip (1.5 dBi) and Dipole (2.1 dBi) antennas. The Chip antenna option provides advantages in its form factor; however, it typically yields shorter range than the Whip and Dipole antenna options when transmitting outdoors. For more information, refer to the "XBee Antennas" Knowledgebase Article located on Digi's Support Web site

Mechanical Drawings

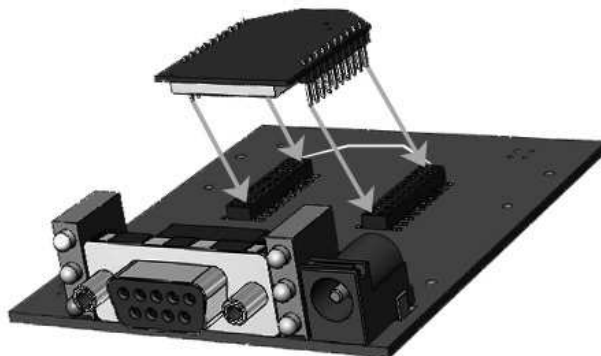
Figure 1-01. Mechanical drawings of the XBee®/XBee-PRO® RF Modules (antenna options not shown)



Mounting Considerations

The XBee®/XBee-PRO® RF Module was designed to mount into a receptacle (socket) and therefore does not require any soldering when mounting it to a board. The XBee Development Kits contain RS-232 and USB interface boards which use two 20-pin receptacles to receive modules.

Figure 1-02. XBee Module Mounting to an RS-232 Interface Board.



The receptacles used on Digi development boards are manufactured by Century Interconnect. Several other manufacturers provide comparable mounting solutions; however, Digi currently uses the following receptacles:

- Through-hole single-row receptacles -
Samtec P/N: MMS-110-01-L-SV (or equivalent)
- Surface-mount double-row receptacles -
Century Interconnect P/N: CPRMSL20-D-0-1 (or equivalent)
- Surface-mount single-row receptacles -
Samtec P/N: SMM-110-02-SM-S

Digi also recommends printing an outline of the module on the board to indicate the orientation the module should be mounted.

Pin Signals

Figure 1-03. XBee®/XBee-PRO® RF Module Pin Numbers

(top sides shown - shields on bottom)

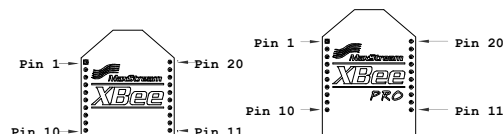


Table 1-02. Pin Assignments for the XBee and XBee-PRO Modules

(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / <u>CONFIG</u>	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	<u>RESET</u>	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	<u>CTS</u> / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / <u>SLEEP</u>	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	<u>RTS</u> / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

* Function is not supported at the time of this release

Design Notes:

- Minimum connections: VCC, GND, DOUT & DIN
- Minimum connections for updating firmware: VCC, GND, DIN, DOUT, RTS & DTR
- Signal Direction is specified with respect to the module
- Module includes a 50k Ω pull-up resistor attached to RESET
- Several of the input pull-ups can be configured using the PR command
- Unused pins should be left disconnected

Electrical Characteristics

Table 1-03. DC Characteristics (VCC = 2.8 - 3.4 VDC)

Symbol	Characteristic	Condition	Min	Typical	Max	Unit
V _{IL}	Input Low Voltage	All Digital Inputs	-	-	0.35 * VCC	V
V _{IH}	Input High Voltage	All Digital Inputs	0.7 * VCC	-	-	V
V _{OL}	Output Low Voltage	I _{OL} = 2 mA, VCC ≥ 2.7 V	-	-	0.5	V
V _{OH}	Output High Voltage	I _{OH} = -2 mA, VCC ≥ 2.7 V	VCC - 0.5	-	-	V
I _{IIN}	Input Leakage Current	V _{IN} = VCC or GND, all inputs, per pin	-	0.025	1	μA
I _{IOZ}	High Impedance Leakage Current	V _{IN} = VCC or GND, all I/O High-Z, per pin	-	0.025	1	μA
TX	Transmit Current	VCC = 3.3 V	-	45 (XBee) 215, 140 (PRO, Int)	-	mA
RX	Receive Current	VCC = 3.3 V	-	50 (XBee) 55 (PRO)	-	mA
PWR-DWN	Power-down Current	SM parameter = 1	-	< 10	-	μA

Table 1-04. ADC Characteristics (Operating)

Symbol	Characteristic	Condition	Min	Typical	Max	Unit
V _{REFH}	VREF - Analog-to-Digital converter reference range		2.08	-	V _{DDAD} *	V
I _{REF}	VREF - Reference Supply Current	Enabled	-	200	-	μA
		Disabled or Sleep Mode	-	< 0.01	0.02	μA
V _{INDC}	Analog Input Voltage ¹		V _{SSAD} - 0.3	-	V _{DDAD} + 0.3	V

1. Maximum electrical operating range, not valid conversion range.

* V_{DDAD} is connected to VCC.

Table 1-05. ADC Timing/Performance Characteristics¹

Symbol	Characteristic	Condition	Min	Typical	Max	Unit
R _{AS}	Source Impedance at Input ²		-	-	10	kΩ
V _{AIN}	Analog Input Voltage ³		V _{REFL}		V _{REFH}	V
RES	Ideal Resolution (1 LSB) ⁴	2.08V ≤ V _{DDAD} ≤ 3.6V	2.031	-	3.516	mV
DNL	Differential Non-linearity ⁵		-	±0.5	±1.0	LSB
INL	Integral Non-linearity ⁶		-	±0.5	±1.0	LSB
E _{ZS}	Zero-scale Error ⁷		-	±0.4	±1.0	LSB
F _{FS}	Full-scale Error ⁸		-	±0.4	±1.0	LSB
E _{IL}	Input Leakage Error ⁹		-	±0.05	±5.0	LSB
E _{TU}	Total Unadjusted Error ¹⁰		-	±1.1	±2.5	LSB

1. All ACCURACY numbers are based on processor and system being in WAIT state (very little activity and no IO switching) and that adequate low-pass filtering is present on analog input pins (filter with 0.01 μF to 0.1 μF capacitor between analog input and VREFL). Failure to observe these guidelines may result in system or microcontroller noise causing accuracy errors which will vary based on board layout and the type and magnitude of the activity.

Data transmission and reception during data conversion may cause some degradation of these specifications, depending on the number and timing of packets. It is advisable to test the ADCs in your installation if best accuracy is required.

2. R_{AS} is the real portion of the impedance of the network driving the analog input pin. Values greater than this amount may not fully charge the input circuitry of the ATD resulting in accuracy error.

3. Analog input must be between V_{REFL} and V_{REFH} for valid conversion. Values greater than V_{REFH} will convert to \$3FF.

4. The resolution is the ideal step size or 1LSB = (V_{REFH} - V_{REFL}) / 1024

5. Differential non-linearity is the difference between the current code width and the ideal code width (1LSB). The current code width is the difference in the transition voltages to and from the current code.

6. Integral non-linearity is the difference between the transition voltage to the current code and the adjusted ideal transition voltage for the current code. The adjusted ideal transition voltage is (Current Code - 1/2) * (1 / ((V_{REFH} + E_{FS}) - (V_{REFL} + E_{ZS}))).

7. Zero-scale error is the difference between the transition to the first valid code and the ideal transition to that code. The Ideal transition voltage to a given code is (Code - 1/2) * (1 / (V_{REFH} - V_{REFL})).

8. Full-scale error is the difference between the transition to the last valid code and the ideal transition to that code. The ideal transition voltage to a given code is (Code - 1/2) * (1 / (V_{REFH} - V_{REFL})).

9. Input leakage error is error due to input leakage across the real portion of the impedance of the network driving the analog pin. Reducing the impedance of the network reduces this error.

10. Total unadjusted error is the difference between the transition voltage to the current code and the ideal straight-line transfer function. This measure of error includes inherent quantization error (1/2LSB) and circuit error (differential, integral, zero-scale, and full-scale) error. The specified value of E_{TU} assumes zero E_{IL} (no leakage or zero real source impedance).

2. RF Module Operation

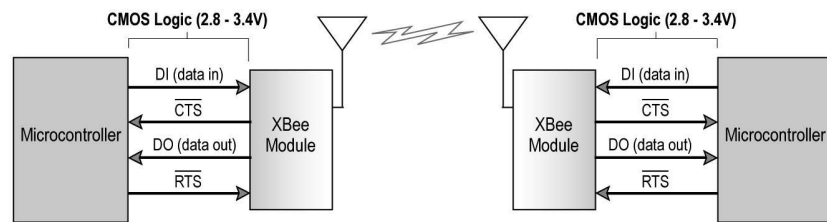
Serial Communications

The XBee®/XBee-PRO® RF Modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device (For example: Through a Digi proprietary RS-232 or USB interface board).

UART Data Flow

Devices that have a UART interface can connect directly to the pins of the RF module as shown in the figure below.

Figure 2-01. System Data Flow Diagram in a UART-interfaced environment
(Low-asserted signals distinguished with horizontal line over signal name.)

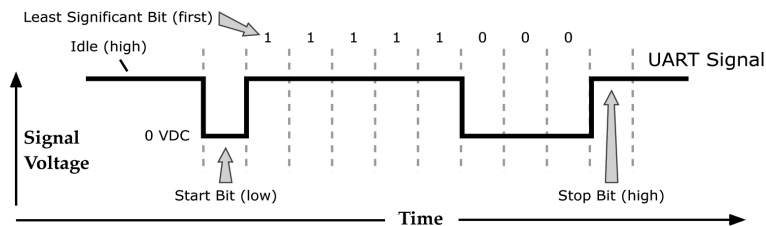


Serial Data

Data enters the module UART through the DI pin (pin 3) as an asynchronous serial signal. The signal should idle high when no data is being transmitted.

Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). The following figure illustrates the serial bit pattern of data passing through the module.

Figure 2-02. UART data packet 0x1F (decimal number "31") as transmitted through the RF module
Example Data Format is 8-N-1 (bits - parity - # of stop bits)



Serial communications depend on the two UARTs (the microcontroller's and the RF module's) to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits).

The UART baud rate and parity settings on the XBee module can be configured with the BD and SB commands, respectively. See the command table in Chapter 3 for details.

Transparent Operation

By default, XBee®/XBee-PRO® RF Modules operate in Transparent Mode. When operating in this mode, the modules act as a serial line replacement - all UART data received through the DI pin is queued up for RF transmission. When RF data is received, the data is sent out the DO pin.

Serial-to-RF Packetization

Data is buffered in the DI buffer until one of the following causes the data to be packetized and transmitted:

1. No serial characters are received for the amount of time determined by the RO (Packetization Timeout) parameter. If RO = 0, packetization begins when a character is received.
2. The maximum number of characters that will fit in an RF packet (100) is received.
3. The Command Mode Sequence (GT + CC + GT) is received. Any character buffered in the DI buffer before the sequence is transmitted.

If the module cannot immediately transmit (for instance, if it is already receiving RF data), the serial data is stored in the DI Buffer. The data is packetized and sent at any RO timeout or when 100 bytes (maximum packet size) are received.

If the DI buffer becomes full, hardware or software flow control must be implemented in order to prevent overflow (loss of data between the host and module).

API Operation

API (Application Programming Interface) Operation is an alternative to the default Transparent Operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module.

When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the module.

Transmit Data Frames (received through the DI pin (pin 3)) include:

- RF Transmit Data Frame
- Command Frame (equivalent to AT commands)

Receive Data Frames (sent out the DO pin (pin 2)) include:

- RF-received data frame
- Command response
- Event notifications such as reset, associate, disassociate, etc.

The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the module that contain address and payload information instead of using command mode to modify addresses. The module will send data frames to the application containing status packets; as well as source, RSSI and payload information from received data packets.

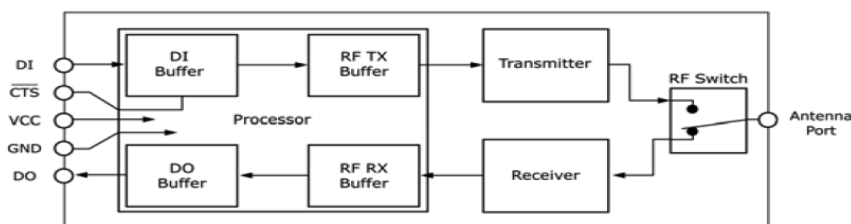
The API operation option facilitates many operations such as the examples cited below:

- > Transmitting data to multiple destinations without entering Command Mode
- > Receive success/failure status of each transmitted RF packet
- > Identify the source address of each received packet

To implement API operations, refer to API sections [p57].

Flow Control

Figure 2-03. Internal Data Flow Diagram



DI (Data In) Buffer

When serial data enters the RF module through the DI pin (pin 3), the data is stored in the DI Buffer until it can be processed.

Hardware Flow Control ($\overline{\text{CTS}}$). When the DI buffer is 17 bytes away from being full; by default, the module de-asserts $\overline{\text{CTS}}$ (high) to signal to the host device to stop sending data [refer to D7 (DIO7 Configuration) parameter]. $\overline{\text{CTS}}$ is re-asserted after the DI Buffer has 34 bytes of memory available.

How to eliminate the need for flow control:

1. Send messages that are smaller than the DI buffer size (202 bytes).
2. Interface at a lower baud rate [BD (Interface Data Rate) parameter] than the throughput data rate.

Case in which the DI Buffer may become full and possibly overflow:

If the module is receiving a continuous stream of RF data, any serial data that arrives on the DI pin is placed in the DI Buffer. The data in the DI buffer will be transmitted over-the-air when the module is no longer receiving RF data in the network.

Refer to the RO (Packetization Timeout), BD (Interface Data Rate) and D7 (DIO7 Configuration) command descriptions for more information.

DO (Data Out) Buffer

When RF data is received, the data enters the DO buffer and is sent out the serial port to a host device. Once the DO Buffer reaches capacity, any additional incoming RF data is lost.

Hardware Flow Control ($\overline{\text{RTS}}$). If $\overline{\text{RTS}}$ is enabled for flow control (D6 (DIO6 Configuration) Parameter = 1), data will not be sent out the DO Buffer as long as $\overline{\text{RTS}}$ (pin 16) is de-asserted.

Two cases in which the DO Buffer may become full and possibly overflow:

1. If the RF data rate is set higher than the interface data rate of the module, the module will receive data from the transmitting module faster than it can send the data to the host.
2. If the host does not allow the module to transmit data out from the DO buffer because of being held off by hardware or software flow control.

Refer to the D6 (DIO6 Configuration) command description for more information.

ADC and Digital I/O Line Support

The XBee®/XBee-PRO® RF Modules support ADC (Analog-to-digital conversion) and digital I/O line passing. The following pins support multiple functions:

Table 2-01. Pin functions and their associated pin numbers and commands

AD = Analog-to-Digital Converter, DIO = Digital Input/Output
Pin functions not applicable to this section are denoted within (parenthesis).

Pin Function	Pin#	AT Command
AD0 / DIO0	20	D0
AD1 / DIO1	19	D1
AD2 / DIO2	18	D2
AD3 / DIO3 / (COORD_SEL)	17	D3
AD4 / DIO4	11	D4
AD5 / DIO5 / (ASSOCIATE)	15	D5
DIO6 / (RTS)	16	D6
DIO7 / (CTS)	12	D7
DI8 / (DTR) / (Sleep_RQ)	9	D8

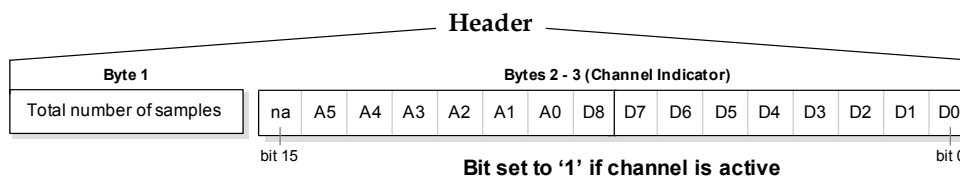
To enable ADC and DIO pin functions:

For ADC Support:	Set ATDn = 2
For Digital Input support:	Set ATDn = 3
For Digital Output Low support:	Set ATDn = 4
For Digital Output High support:	Set ATDn = 5

I/O Data Format

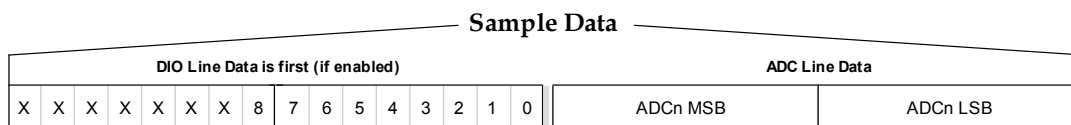
I/O data begins with a header. The first byte of the header defines the number of samples forthcoming. The last 2 bytes of the header (Channel Indicator) define which inputs are active. Each bit represents either a DIO line or ADC channel.

Figure 2-04. Header



Sample data follows the header and the channel indicator frame is used to determine how to read the sample data. If any of the DIO lines are enabled, the first 2 bytes are the DIO sample. The ADC data follows. ADC channel data is represented as an unsigned 10-bit value right-justified on a 16-bit boundary.

Figure 2-05. Sample Data



API Support

I/O data is sent out the UART using an API frame. All other data can be sent and received using Transparent Operation [refer to p11] or API framing if API mode is enabled (AP > 0).

API Operations support two RX (Receive) frame identifiers for I/O data (set 16-bit address to 0xFFFE and the module will do 64-bit addressing):

- 0x82 for RX (Receive) Packet: 64-bit address I/O
- 0x83 for RX (Receive) Packet: 16-bit address I/O

The API command header is the same as shown in the "RX (Receive) Packet: 64-bit Address" and "RX (Receive) Packet: 64-bit Address" API types [refer to p63]. RX data follows the format described in the I/O Data Format section [p13].

Applicable Commands: AP (API Enable)

Sleep Support

Automatic wakeup sampling can be suppressed by setting SO bit 1. When an RF module wakes, it will always do a sample based on any active ADC or DIO lines. This allows sampling based on the sleep cycle whether it be Cyclic Sleep (SM parameter = 4 or 5) or Pin Sleep (SM = 1 or 2). To gather more samples when awake, set the IR (Sample Rate) parameter.

For Cyclic Sleep modes: If the IR parameter is set, the module will stay awake until the IT (Samples before TX) parameter is met. The module will stay awake for ST (Time before Sleep) time.

Applicable Commands: IR (Sample Rate), IT (Samples before TX), SM (Sleep Mode), IC (DIO Change Detect), SO (Sleep Options)

DIO Pin Change Detect

When "DIO Change Detect" is enabled (using the IC command), DIO lines 0-7 are monitored. When a change is detected on a DIO line, the following will occur:

1. An RF packet is sent with the updated DIO pin levels. This packet will not contain any ADC samples.
2. Any queued samples are transmitted before the change detect data. This may result in receiving a packet with less than IT (Samples before TX) samples.

Note: Change detect will not affect Pin Sleep wake-up. The D8 pin (DTR/Sleep_RQ/DI8) is the only line that will wake a module from Pin Sleep. If not all samples are collected, the module will still enter Sleep Mode after a change detect packet is sent.

Applicable Commands: IC (DIO Change Detect), IT (Samples before TX)

NOTE: Change detect is only supported when the Dx (DIOx Configuration) parameter equals 3, 4 or 5.

Sample Rate (Interval)

The Sample Rate (Interval) feature allows enabled ADC and DIO pins to be read periodically on modules that are not configured to operate in Sleep Mode. When one of the Sleep Modes is enabled and the IR (Sample Rate) parameter is set, the module will stay awake until IT (Samples before TX) samples have been collected.

Once a particular pin is enabled, the appropriate sample rate must be chosen. The maximum sample rate that can be achieved while using one A/D line is 1 sample/ms or 1 KHz (Note that the modem will not be able to keep up with transmission when IR & IT are equal to "1" and that configuring the modem to sample at rates greater than once every 20ms is not recommended).

Applicable Commands: IR (Sample Rate), IT (Samples before TX), SM (Sleep Mode)

I/O Line Passing

Virtual wires can be set up between XBee®/XBee-PRO® Modules. When an RF data packet is received that contains I/O data, the receiving module can be setup to update any enabled outputs (PWM and DIO) based on the data it receives.

Note that I/O lines are mapped in pairs. For example: AD0 can only update PWM0 and DI5 can only update DO5. The default setup is for outputs not to be updated, which results in the I/O data being sent out the UART (refer to the IU (Enable I/O Output) command). To enable the outputs to be updated, the IA (I/O Input Address) parameter must be setup with the address of the module that has the appropriate inputs enabled. This effectively binds the outputs to a particular module's input. This does not affect the ability of the module to receive I/O line data from other modules - only its ability to update enabled outputs. The IA parameter can also be setup to accept I/O data for output changes from any module by setting the IA parameter to 0xFFFF.

When outputs are changed from their non-active state, the module can be setup to return the output level to its non-active state. The timers are set using the Tn (Dn Output Timer) and PT (PWM Output Timeout) commands. The timers are reset every time a valid I/O packet (passed IA check) is received. The IC (Change Detect) and IR (Sample Rate) parameters can be setup to keep the output set to their active output if the system needs more time than the timers can handle.

Note: DI8 cannot be used for I/O line passing.

Applicable Commands: IA (I/O Input Address), Tn (Dn Output Timeout), P0 (PWM0 Configuration), P1 (PWM1 Configuration), M0 (PWM0 Output Level), M1 (PWM1 Output Level), PT (PWM Output Timeout), RP (RSSI PWM Timer)

Configuration Example

As an example for a simple A/D link, a pair of RF modules could be set as follows:

Remote Configuration	Base Configuration
DL = 0x1234	DL = 0x5678
MY = 0x5678	MY = 0x1234
D0 = 2	P0 = 2
D1 = 2	P1 = 2
IR = 0x14	IU = 1
IT = 5	IA = 0x5678 (or 0xFFFF)

These settings configure the remote module to sample AD0 and AD1 once each every 20 ms. It then buffers 5 samples each before sending them back to the base module. The base should then receive a 32-Byte transmission (20 Bytes data and 12 Bytes framing) every 100 ms.

XBee®/XBee-PRO® Networks

The following terms will be used to explicate the network operations:

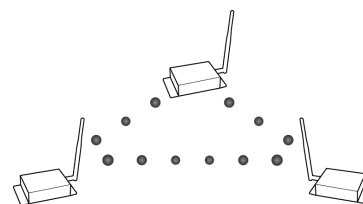
Table 2-02. Terms and definitions

Term	Definition
PAN	Personal Area Network - A data communication network that includes one or more End Devices and optionally a Coordinator.
Coordinator	A Full-function device (FFD) that provides network synchronization by polling nodes [NonBeacon (w/ Coordinator) networks only]
End Device	<i>When in the same network as a Coordinator</i> - RF modules that rely on a Coordinator for synchronization and can be put into states of sleep for low-power applications.
Association	The establishment of membership between End Devices and a Coordinator. Association is only applicable in NonBeacon (w/Coordinator) networks.

Peer-to-Peer

By default, XBee®/XBee-PRO RF Modules are configured to operate within a Peer-to-Peer network topology and therefore are not dependent upon Master/Slave relationships. NonBeacon systems operate within a Peer-to-Peer network topology and therefore are not dependent upon Master/Slave relationships. This means that modules remain synchronized without use of master/server configurations and each module in the network shares both roles of master and slave. Digi's peer-to-peer architecture features fast synchronization times and fast cold start times. This default configuration accommodates a wide range of RF data applications.

Figure 2-06. Peer-to-Peer Architecture



A peer-to-peer network can be established by configuring each module to operate as an End Device (CE = 0), disabling End Device Association on all modules (A1 = 0) and setting ID and CH parameters to be identical across the network.

NonBeacon (w/ Coordinator)

A device is configured as a Coordinator by setting the CE (Coordinator Enable) parameter to "1". Coordinator power-up is governed by the A2 (Coordinator Association) parameter.

In a Coordinator system, the Coordinator can be configured to use direct or indirect transmissions. If the SP (Cyclic Sleep Period) parameter is set to "0", the Coordinator will send data immediately. Otherwise, the SP parameter determines the length of time the Coordinator will retain the data before discarding it. Generally, SP (Cyclic Sleep Period) and ST (Time before Sleep) parameters should be set to match the SP and ST settings of the End Devices.

Association

Association is the establishment of membership between End Devices and a Coordinator. The establishment of membership is useful in scenarios that require a central unit (Coordinator) to relay messages to or gather data from several remote units (End Devices), assign channels or assign PAN IDs.

An RF data network that consists of one Coordinator and one or more End Devices forms a PAN (Personal Area Network). Each device in a PAN has a PAN Identifier [ID (PAN ID) parameter]. PAN IDs must be unique to prevent miscommunication between PANs. The Coordinator PAN ID is set using the ID (PAN ID) and A2 (Coordinator Association) commands.

An End Device can associate to a Coordinator without knowing the address, PAN ID or channel of the Coordinator. The A1 (End Device Association) parameter bit fields determine the flexibility of an End Device during association. The A1 parameter can be used for an End Device to dynamically set its destination address, PAN ID and/or channel.

For example: If the PAN ID of a Coordinator is known, but the operating channel is not; the A1 command on the End Device should be set to enable the 'Auto_Associate' and 'Reassign_Channel' bits. Additionally, the ID parameter should be set to match the PAN ID of the associated Coordinator.

Coordinator / End Device Setup and Operation

To configure a module to operate as a Coordinator, set the CE (Coordinator Enable) parameter to '1'. Set the CE parameter of End Devices to '0' (default). Coordinator and End Devices should contain matching firmware versions.

NonBeacon (w/ Coordinator) Systems

The Coordinator can be configured to use direct or indirect transmissions. If the SP (Cyclic Sleep Period) parameter is set to '0', the Coordinator will send data immediately. Otherwise, the SP parameter determines the length of time the Coordinator will retain the data before discarding it. Generally, SP (Cyclic Sleep Period) and ST (Time before Sleep) parameters should be set to match the SP and ST settings of the End Devices.

Coordinator Start-up

Coordinator power-up is governed by the A2 (Coordinator Association) command. On power-up, the Coordinator undergoes the following sequence of events:

1. Check A2 parameter- Reassign_PANID Flag

Set (bit 0 = 1) - The Coordinator issues an Active Scan. The Active Scan selects one channel and transmits a request to the broadcast address (0xFFFF) and broadcast PAN ID (0xFFFF). It then listens on that channel for beacons from any Coordinator operating on that channel. The listen time on each channel is determined by the SD (Scan Duration) parameter value.

Once the time expires on that channel, the Active Scan selects another channel and again transmits the BeaconRequest as before. This process continues until all channels have been scanned, or until 5 PANs have been discovered. When the Active Scan is complete, the results include a list of PAN IDs and Channels that are being used by other PANs. This list is used to assign an unique PAN ID to the new Coordinator. The ID parameter will be retained if it is not found in the Active Scan results. Otherwise, the ID (PAN ID) parameter setting will be updated to a PAN ID that was not detected.

Not Set (bit 0 = 0) - The Coordinator retains its ID setting. No Active Scan is performed.

2. Check A2 parameter - Reassign_Channel Flag (bit 1)

Set (bit 1 = 1) - The Coordinator issues an Energy Scan. The Energy Scan selects one channel and scans for energy on that channel. The duration of the scan is specified by the SD (Scan Duration) parameter. Once the scan is completed on a channel, the Energy Scan selects the next channel and begins a new scan on that channel. This process continues until all channels have been scanned.

When the Energy Scan is complete, the results include the maximal energy values detected on each channel. This list is used to determine a channel where the least energy was detected. If an Active Scan was performed (Reassign_PANID Flag set), the channels used by the detected PANs are eliminated as possible channels. Thus, the results of the Energy Scan and the Active Scan (if performed) are used to find the best channel (channel with the least energy that is not used by any detected PAN). Once the best channel has been selected, the CH (Channel) parameter value is updated to that channel.

Not Set (bit 1 = 0) - The Coordinator retains its CH setting. An Energy Scan is not performed.

3. Start Coordinator

The Coordinator starts on the specified channel (CH parameter) and PAN ID (ID parameter). Note, these may be selected in steps 1 and/or 2 above. The Coordinator will only allow End Devices to associate to it if the A2 parameter "AllowAssociation" flag is set. Once the Coordinator has successfully started, the Associate LED will blink 1 time per second. (The LED is solid if the Coordinator has not started.)

4. Coordinator Modifications

Once a Coordinator has started:

Modifying the A2 (Reassign_Channel or Reassign_PANID bits), ID, CH or MY parameters will cause the Coordinator's MAC to reset (The Coordinator RF module (including volatile RAM) is not reset). Changing the A2 AllowAssociation bit will not reset the Coordinator's MAC. In a non-beaconing system, End Devices that associated to the Coordinator prior to a MAC reset will have knowledge of the new settings on the Coordinator. Thus, if the Coordinator were to change its ID, CH or MY settings, the End Devices would no longer be able to communicate with the non-beacon Coordinator. Once a Coordinator has started, the ID, CH, MY or A2 (Reassign_Channel or Reassign_PANID bits) should not be changed.

End Device Start-up

End Device power-up is governed by the A1 (End Device Association) command. On power-up, the End Device undergoes the following sequence of events:

1. Check A1 parameter - AutoAssociate Bit

Set (bit 2 = 1) - End Device will attempt to associate to a Coordinator. (refer to steps 2-3).

Not Set (bit 2 = 0) - End Device will not attempt to associate to a Coordinator. The End Device will operate as specified by its ID, CH and MY parameters. Association is considered complete and the Associate LED will blink quickly (5 times per second). When the AutoAssociate bit is not set, the remaining steps (2-3) do not apply.

2. Discover Coordinator (if Auto-Associate Bit Set)

The End Device issues an Active Scan. The Active Scan selects one channel and transmits a BeaconRequest command to the broadcast address (0xFFFF) and broadcast PAN ID (0xFFFF). It then listens on that channel for beacons from any Coordinator operating on that channel. The listen time on each channel is determined by the SD parameter.

Once the time expires on that channel, the Active Scan selects another channel and again transmits the BeaconRequest command as before. This process continues until all channels have been scanned, or until 5 PANs have been discovered. When the Active Scan is complete, the results include a list of PAN IDs and Channels that are being used by detected PANs.

The End Device selects a Coordinator to associate with according to the A1 parameter "Reassign_PANID" and "Reassign_Channel" flags:

Reassign_PANID Bit Set (bit 0 = 1)- End Device can associate with a PAN with any ID value.

Reassign_PANID Bit Not Set (bit 0 = 0) - End Device will only associate with a PAN whose ID setting matches the ID setting of the End Device.

Reassign_Channel Bit Set (bit 1 = 1) - End Device can associate with a PAN with any CH value.

Reassign_Channel Bit Not Set (bit 1 = 0)- End Device will only associate with a PAN whose CH setting matches the CH setting of the End Device.

After applying these filters to the discovered Coordinators, if multiple candidate PANs exist, the End Device will select the PAN whose transmission link quality is the strongest. If no valid Coordinator is found, the End Device will either go to sleep (as dictated by its SM (Sleep Mode) parameter) or retry Association.

Note - An End Device will also disqualify Coordinators if they are not allowing association (A2 - AllowAssociation bit); or, if the Coordinator is not using the same NonBeacon scheme as the End Device. (They must both be programmed with NonBeacon code.)

3. Associate to Valid Coordinator

Once a valid Coordinator is found (step 2), the End Device sends an AssociationRequest message to the Coordinator. It then waits for an AssociationConfirmation to be sent from the Coordinator. Once the Confirmation is received, the End Device is Associated and the Associate LED will blink rapidly (2 times per second). The LED is solid if the End Device has not associated.

4. End Device Changes once an End Device has associated

Changing A1, ID or CH parameters will cause the End Device to disassociate and restart the Association procedure.

If the End Device fails to associate, the AI command can give some indication of the failure.

XBee®/XBee-PRO® Addressing

Every RF data packet sent over-the-air contains a Source Address and Destination Address field in its header. The RF module conforms to the 802.15.4 specification and supports both short 16-bit addresses and long 64-bit addresses. A unique 64-bit IEEE source address is assigned at the factory and can be read with the SL (Serial Number Low) and SH (Serial Number High) commands. Short addressing must be configured manually. A module will use its unique 64-bit address as its Source Address if its MY (16-bit Source Address) value is "0xFFFF" or "0xFFFE".

To send a packet to a specific module using 64-bit addressing: Set the Destination Address (DL + DH) of the sender to match the Source Address (SL + SH) of the intended destination module.

To send a packet to a specific module using 16-bit addressing: Set DL (Destination Address Low) parameter to equal the MY parameter of the intended destination module and set the DH (Destination Address High) parameter to '0'.

Unicast Mode

By default, the RF module operates in Unicast Mode. Unicast Mode is the only mode that supports retries. While in this mode, receiving modules send an ACK (acknowledgement) of RF packet reception to the transmitter. If the transmitting module does not receive the ACK, it will re-send the packet up to three times or until the ACK is received.

Short 16-bit addresses. The module can be configured to use short 16-bit addresses as the Source Address by setting (MY < 0xFFFE). Setting the DH parameter (DH = 0) will configure the Destination Address to be a short 16-bit address (if DL < 0xFFFE). For two modules to communicate using short addressing, the Destination Address of the transmitter module must match the MY parameter of the receiver.

The following table shows a sample network configuration that would enable Unicast Mode communications using short 16-bit addresses.

Table 2-03. Sample Unicast Network Configuration (using 16-bit addressing)

Parameter	RF Module 1	RF Module 2
MY (Source Address)	0x01	0x02
DH (Destination Address High)	0	0
DL (Destination Address Low)	0x02	0x01

Long 64-bit addresses. The RF module's serial number (SL parameter concatenated to the SH parameter) can be used as a 64-bit source address when the MY (16-bit Source Address) parameter is disabled. When the MY parameter is disabled (MY = 0xFFFF or 0xFFFE), the module's source address is set to the 64-bit IEEE address stored in the SH and SL parameters.

When an End Device associates to a Coordinator, its MY parameter is set to 0xFFFE to enable 64-bit addressing. The 64-bit address of the module is stored as SH and SL parameters. To send a packet to a specific module, the Destination Address (DL + DH) on the sender must match the Source Address (SL + SH) of the desired receiver.

Broadcast Mode

Any RF module within range will accept a packet that contains a broadcast address. When configured to operate in Broadcast Mode, receiving modules do not send ACKs (Acknowledgements) and transmitting modules do not automatically re-send packets as is the case in Unicast Mode.

To send a broadcast packet to all modules regardless of 16-bit or 64-bit addressing, set the destination addresses of all the modules as shown below.

Sample Network Configuration (All modules in the network):

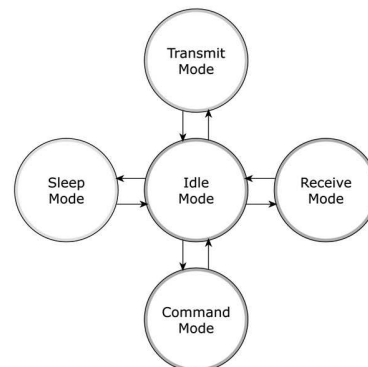
- DL (Destination Low Address) = 0x0000FFFF
- DH (Destination High Address) = 0x00000000 (default value)

NOTE: When programming the module, parameters are entered in hexadecimal notation (without the "0x" prefix). Leading zeros may be omitted.

Modes of Operation

XBee®/XBee-PRO® RF Modules operate in five modes.

Figure 2-07. Modes of Operation



Idle Mode

When not receiving or transmitting data, the RF module is in Idle Mode. The module shifts into the other modes of operation under the following conditions:

- Transmit Mode (Serial data is received in the DI Buffer)
- Receive Mode (Valid RF data is received through the antenna)
- Sleep Mode (Sleep Mode condition is met)
- Command Mode (Command Mode Sequence is issued)

Transmit/Receive Modes

RF Data Packets

Each transmitted data packet contains a Source Address and Destination Address field. The Source Address matches the address of the transmitting module as specified by the MY (Source Address) parameter (if MY \geq 0xFFFE), the SH (Serial Number High) parameter or the SL (Serial Number Low) parameter. The <Destination Address> field is created from the DH (Destination Address High) and DL (Destination Address Low) parameter values. The Source Address and/or Destination Address fields will either contain a 16-bit short or long 64-bit long address.

The RF data packet structure follows the 802.15.4 specification.

[Refer to the XBee/XBee-PRO Addressing section for more information]

Direct and Indirect Transmission

There are two methods to transmit data:

- Direct Transmission - data is transmitted immediately to the Destination Address
- Indirect Transmission - A packet is retained for a period of time and is only transmitted after the destination module (Source Address = Destination Address) requests the data.

Indirect Transmissions can only occur on a Coordinator. Thus, if all nodes in a network are End Devices, only Direct Transmissions will occur. Indirect Transmissions are useful to ensure packet delivery to a sleeping node. The Coordinator currently is able to retain up to 2 indirect messages.

Direct Transmission

A Coordinator can be configured to use only Direct Transmission by setting the SP (Cyclic Sleep Period) parameter to "0". Also, a Coordinator using indirect transmissions will revert to direct transmission if it knows the destination module is awake.

To enable this behavior, the ST (Time before Sleep) value of the Coordinator must be set to match the ST value of the End Device. Once the End Device either transmits data to the Coordinator or polls the Coordinator for data, the Coordinator will use direct transmission for all subsequent data transmissions to that module address until ST time occurs with no activity (at which point it will revert to using indirect transmissions for that module address). "No activity" means no transmission or reception of messages with a specific address. Global messages will not reset the ST timer.

Indirect Transmission

To configure Indirect Transmissions in a PAN (Personal Area Network), the SP (Cyclic Sleep Period) parameter value on the Coordinator must be set to match the longest sleep value of any End Device. The sleep period value on the Coordinator determines how long (time or number of beacons) the Coordinator will retain an indirect message before discarding it.

An End Device must poll the Coordinator once it wakes from Sleep to determine if the Coordinator has an indirect message for it. For Cyclic Sleep Modes, this is done automatically every time the module wakes (after SP time). For Pin Sleep Modes, the A1 (End Device Association) parameter value must be set to enable Coordinator polling on pin wake-up. Alternatively, an End Device can use the FP (Force Poll) command to poll the Coordinator as needed.

CCA (Clear Channel Assessment)

Prior to transmitting a packet, a CCA (Clear Channel Assessment) is performed on the channel to determine if the channel is available for transmission. The detected energy on the channel is compared with the CA (Clear Channel Assessment) parameter value. If the detected energy exceeds the CA parameter value, the packet is not transmitted.

Also, a delay is inserted before a transmission takes place. This delay is settable using the RN (Backoff Exponent) parameter. If RN is set to "0", then there is no delay before the first CCA is performed. The RN parameter value is the equivalent of the "minBE" parameter in the 802.15.4 specification. The transmit sequence follows the 802.15.4 specification.

By default, the MM (MAC Mode) parameter = 0. On a CCA failure, the module will attempt to re-send the packet up to two additional times.

When in Unicast packets with RR (Retries) = 0, the module will execute two CCA retries. Broadcast packets always get two CCA retries.

Acknowledgement

If the transmission is not a broadcast message, the module will expect to receive an acknowledgement from the destination node. If an acknowledgement is not received, the packet will be resent up to 3 more times. If the acknowledgement is not received after all transmissions, an ACK failure is recorded.

Sleep Mode

Sleep Modes enable the RF module to enter states of low-power consumption when not in use. In order to enter Sleep Mode, one of the following conditions must be met (in addition to the module having a non-zero SM parameter value):

- Sleep_RQ (pin 9) is asserted and the module is in a pin sleep mode (SM = 1, 2, or 5)
- The module is idle (no data transmission or reception) for the amount of time defined by the ST (Time before Sleep) parameter. [NOTE: ST is only active when SM = 4-5.]

Table 2-04. Sleep Mode Configurations

Sleep Mode Setting	Transition into Sleep Mode	Transition out of Sleep Mode (wake)	Characteristics	Related Commands	Power Consumption
Pin Hibernate (SM = 1)	Assert (high) Sleep_RQ (pin 9)	De-assert (low) Sleep_RQ	Pin/Host-controlled / NonBeacon systems only / Lowest Power	(SM)	< 10 μ A (@3.0 VCC)
Pin Doze (SM = 2)	Assert (high) Sleep_RQ (pin 9)	De-assert (low) Sleep_RQ	Pin/Host-controlled / NonBeacon systems only / Fastest wake-up	(SM)	< 50 μ A
Cyclic Sleep (SM = 4)	Automatic transition to Sleep Mode as defined by the SM (Sleep Mode) and ST (Time before Sleep) parameters.	Transition occurs after the cyclic sleep time interval elapses. The time interval is defined by the SP (Cyclic Sleep Period) parameter.	RF module wakes in pre-determined time intervals to detect if RF data is present / When SM = 5	(SM), SP, ST	< 50 μ A when sleeping
Cyclic Sleep (SM = 5)	Automatic transition to Sleep Mode as defined by the SM (Sleep Mode) and ST (Time before Sleep) parameters or on a falling edge transition of the SLEEP_RQ pin.	Transition occurs after the cyclic sleep time interval elapses. The time interval is defined by the SP (Cyclic Sleep Period) parameter.	RF module wakes in pre-determined time intervals to detect if RF data is present. Module also wakes on a falling edge of SLEEP_RQ	(SM), SP, ST	< 50 μ A when sleeping

The SM command is central to setting Sleep Mode configurations. By default, Sleep Modes are disabled (SM = 0) and the module remains in Idle/Receive Mode. When in this state, the module is constantly ready to respond to serial or RF activity.

Pin/Host-controlled Sleep Modes

The transient current when waking from pin sleep (SM = 1 or 2) does not exceed the idle current of the module. The current ramps up exponentially to its idle current.

Pin Hibernate (SM = 1)

- Pin/Host-controlled
- Typical power-down current: < 10 μ A (@3.0 VCC)
- Wake-up time: 13.2 msec

Pin Hibernate Mode minimizes quiescent power (power consumed when in a state of rest or inactivity). This mode is voltage level-activated; when Sleep_RQ (pin 9) is asserted, the module will finish any transmit, receive or association activities, enter Idle Mode, and then enter a state of sleep. The module will not respond to either serial or RF activity while in pin sleep.

To wake a sleeping module operating in Pin Hibernate Mode, de-assert Sleep_RQ (pin 9). The module will wake when Sleep_RQ is de-asserted and is ready to transmit or receive when the CTS line is low. When waking the module, the pin must be de-asserted at least two 'byte times' after CTS goes low. This assures that there is time for the data to enter the DI buffer.

Pin Doze (SM = 2)

- Pin/Host-controlled
- Typical power-down current: < 50 μ A
- Wake-up time: 2 msec

Pin Doze Mode functions as does Pin Hibernate Mode; however, Pin Doze features faster wake-up time and higher power consumption.

To wake a sleeping module operating in Pin Doze Mode, de-assert Sleep_RQ (pin 9). The module will wake when Sleep_RQ is de-asserted and is ready to transmit or receive when the CTS line is

low. When waking the module, the pin must be de-asserted at least two 'byte times' after CTS goes low. This assures that there is time for the data to enter the DI buffer.

Cyclic Sleep Modes

Cyclic Sleep Remote (SM = 4)

- Typical Power-down Current: < 50 μ A (when asleep)
- Wake-up time: 2 msec

The Cyclic Sleep Modes allow modules to periodically check for RF data. When the SM parameter is set to '4', the module is configured to sleep, then wakes once a cycle to check for data from a module configured as a Cyclic Sleep Coordinator (SM = 0, CE = 1). The Cyclic Sleep Remote sends a poll request to the coordinator at a specific interval set by the SP (Cyclic Sleep Period) parameter. The coordinator will transmit any queued data addressed to that specific remote upon receiving the poll request.

If no data is queued for the remote, the coordinator will not transmit and the remote will return to sleep for another cycle. If queued data is transmitted back to the remote, it will stay awake to allow for back and forth communication until the ST (Time before Sleep) timer expires.

Also note that $\overline{\text{CTS}}$ will go low each time the remote wakes, allowing for communication initiated by the remote host if desired.

Cyclic Sleep Remote with Pin Wake-up (SM = 5)

Use this mode to wake a sleeping remote module through either the RF interface or by the de-assertion of Sleep_RQ for event-driven communications. The cyclic sleep mode works as described above (Cyclic Sleep Remote) with the addition of a pin-controlled wake-up at the remote module. The Sleep_RQ pin is edge-triggered, not level-triggered. The module will wake when a low is detected then set $\overline{\text{CTS}}$ low as soon as it is ready to transmit or receive.

Any activity will reset the ST (Time before Sleep) timer so the module will go back to sleep only after there is no activity for the duration of the timer. Once the module wakes (pin-controlled), further pin activity is ignored. The module transitions back into sleep according to the ST time regardless of the state of the pin.

[Cyclic Sleep Coordinator (SM = 6)]

- Typical current = Receive current
- Always awake

NOTE: The SM=6 parameter value exists solely for backwards compatibility with firmware version 1.x60. If backwards compatibility with the older firmware version is not required, always use the CE (Coordinator Enable) command to configure a module as a Coordinator.

This mode configures a module to wake cyclic sleeping remotes through RF interfacing. The Coordinator will accept a message addressed to a specific remote 16 or 64-bit address and hold it in a buffer until the remote wakes and sends a poll request. Messages not sent directly (buffered and requested) are called "Indirect messages". The Coordinator only queues one indirect message at a time. The Coordinator will hold the indirect message for a period 2.5 times the sleeping period indicated by the SP (Cyclic Sleep Period) parameter. The Coordinator's SP parameter should be set to match the value used by the remotes.

Command Mode

To modify or read RF Module parameters, the module must first enter into Command Mode - a state in which incoming characters are interpreted as commands. Two Command Mode options are supported: AT Command Mode [refer to section below] and API Command Mode [p57].

AT Command Mode

To Enter AT Command Mode:

Send the 3-character command sequence “+++” and observe guard times before and after the command characters. [Refer to the “Default AT Command Mode Sequence” below.]

Default AT Command Mode Sequence (for transition to Command Mode):

- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]
- Input three plus characters (“+++”) within one second [CC (Command Sequence Character) parameter = 0x2B.]
- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]

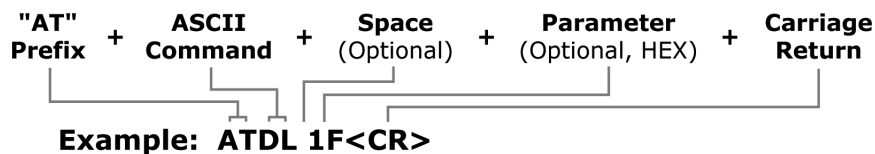
All of the parameter values in the sequence can be modified to reflect user preferences.

NOTE: Failure to enter AT Command Mode is most commonly due to baud rate mismatch. Ensure the ‘Baud’ setting on the “PC Settings” tab matches the interface data rate of the RF module. By default, the BD parameter = 3 (9600 bps).

To Send AT Commands:

Send AT commands and parameters using the syntax shown below.

Figure 2-08. Syntax for sending AT Commands



To read a parameter value stored in the RF module's register, omit the parameter field.

The preceding example would change the RF module Destination Address (Low) to “0x1F”. To store the new value to non-volatile (long term) memory, subsequently send the WR (Write) command.

For modified parameter values to persist in the module's registry after a reset, changes must be saved to non-volatile memory using the WR (Write) Command. Otherwise, parameters are restored to previously saved values after the module is reset.

System Response. When a command is sent to the module, the module will parse and execute the command. Upon successful execution of a command, the module returns an “OK” message. If execution of a command results in an error, the module returns an “ERROR” message.

To Exit AT Command Mode:

1. Send the ATCN (Exit Command Mode) command (followed by a carriage return).
[OR]
2. If no valid AT Commands are received within the time specified by CT (Command Mode Timeout) Command, the RF module automatically returns to Idle Mode.

For an example of programming the RF module using AT Commands and descriptions of each configurable parameter, refer to the RF Module Configuration chapter [p26].

3. RF Module Configuration

Programming the RF Module

Refer to the Command Mode section [p25] for more information about entering Command Mode, sending AT commands and exiting Command Mode. For information regarding module programming using API Mode, refer to the API Operation sections [p57].

Programming Examples

Setup

The programming examples in this section require the installation of Digi's X-CTU Software and a serial connection to a PC. (Digi stocks RS-232 and USB boards to facilitate interfacing with a PC.)

1. Install Digi's X-CTU Software to a PC by double-clicking the "setup_X-CTU.exe" file. (The file is located on the Digi CD and www.digi.com/xctu.)
2. Mount the RF module to an interface board, then connect the module assembly to a PC.
3. Launch the X-CTU Software and select the 'PC Settings' tab. Verify the baud and parity settings of the Com Port match those of the RF module.

NOTE: Failure to enter AT Command Mode is most commonly due to baud rate mismatch. Ensure the 'Baud' setting on the 'PC Settings' tab matches the interface data rate of the RF module. By default, the BD parameter = 3 (which corresponds to 9600 bps).

Sample Configuration: Modify RF Module Destination Address

Example: Utilize the X-CTU "Terminal" tab to change the RF module's DL (Destination Address Low) parameter and save the new address to non-volatile memory.

After establishing a serial connection between the RF module and a PC [refer to the 'Setup' section above], select the "Terminal" tab of the X-CTU Software and enter the following command lines ('CR' stands for carriage return):

Method 1 (One line per command)

Send AT Command	System Response
+++	OK <CR> (Enter into Command Mode)
ATDL <Enter>	{current value} <CR> (Read Destination Address Low)
ATDL1A0D <Enter>	OK <CR> (Modify Destination Address Low)
ATWR <Enter>	OK <CR> (Write to non-volatile memory)
ATCN <Enter>	OK <CR> (Exit Command Mode)

Method 2 (Multiple commands on one line)

Send AT Command	System Response
+++	OK <CR> (Enter into Command Mode)
ATDL <Enter>	{current value} <CR> (Read Destination Address Low)
ATDL1A0D,WR,CN <Enter>	OK<CR> OK<CR> OK<CR>

Sample Configuration: Restore RF Module Defaults

Example: Utilize the X-CTU "Modem Configuration" tab to restore default parameter values.

After establishing a connection between the module and a PC [refer to the 'Setup' section above], select the "Modem Configuration" tab of the X-CTU Software.

1. Select the 'Read' button.
2. Select the 'Restore' button.

Remote Configuration Commands

The API firmware has provisions to send configuration commands to remote devices using the Remote Command Request API frame (see API Operation). This API frame can be used to send commands to a remote module to read or set command parameters.

The API firmware has provisions to send configuration commands (set or read) to a remote module using the Remote Command Request API frame (see API Operations). Remote commands can be issued to read or set command parameters on a remote device.

Sending a Remote Command

To send a remote command, the Remote Command Request frame should be populated with values for the 64 bit and 16 bit addresses. If 64 bit addressing is desired then the 16 bit address field should be filled with 0xFFFE. If any value other than 0xFFFE is used in the 16 bit address field then the 64 bit address field will be ignored and 16 bit addressing will be used. If a command response is desired, the Frame ID should be set to a non-zero value.

Applying Changes on Remote

When remote commands are used to change command parameter settings on a remote device, parameter changes do not take effect until the changes are applied. For example, changing the BD parameter will not change the actual serial interface rate on the remote until the changes are applied. Changes can be applied using remote commands in one of three ways:

Set the apply changes option bit in the API frame

Issue an AC command to the remote device

Issue a WR + FR command to the remote device to save changes and reset the device.

Remote Command Responses

If the remote device receives a remote command request transmission, and the API frame ID is non-zero, the remote will send a remote command response transmission back to the device that sent the remote command. When a remote command response transmission is received, a device sends a remote command response API frame out its UART. The remote command response indicates the status of the command (success, or reason for failure), and in the case of a command query, it will include the register value.

The device that sends a remote command will not receive a remote command response frame if:

The destination device could not be reached

The frame ID in the remote command request is set to 0.

Command Reference Tables

XBee®/XBee-PRO® RF Modules expect numerical values in hexadecimal. Hexadecimal values are designated by a "0x" prefix. Decimal equivalents are designated by a "d" suffix. Commands are contained within the following command categories (listed in the order that their tables appear):

- Special
- Networking & Security
- RF Interfacing
- Sleep (Low Power)
- Serial Interfacing
- I/O Settings
- Diagnostics
- AT Command Options

All modules within a PAN should operate using the same firmware version.

Special**Table 3-01. XBee-PRO Commands - Special**

AT Command	Command Category	Name and Description	Parameter Range	Default
WR	Special	Write. Write parameter values to non-volatile memory so that parameter modifications persist through subsequent power-up or reset. Note: Once WR is issued, no additional characters should be sent to the module until after the response "OK\r" is received.	-	-
RE	Special	Restore Defaults. Restore module parameters to factory defaults.	-	-
FR (v1.x80*)	Special	Software Reset. Responds immediately with an OK then performs a hard reset ~100ms later.	-	-

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

Networking & Security**Table 3-02. XBee®/XBee-PRO® Commands - Networking & Security** (Sub-categories designated within [brackets])

AT Command	Command Category	Name and Description	Parameter Range	Default
CH	Networking {Addressing}	Channel. Set/Read the channel number used for transmitting and receiving data between RF modules (uses 802.15.4 protocol channel numbers).	0x0B - 0x1A (XBee) 0x0C - 0x17 (XBee-PRO)	0x0C (12d)
ID	Networking {Addressing}	PAN ID. Set/Read the PAN (Personal Area Network) ID. Use 0xFFFF to broadcast messages to all PANs.	0 - 0xFFFF	0x3332 (13106d)
DH	Networking {Addressing}	Destination Address High. Set/Read the upper 32 bits of the 64-bit destination address. When combined with DL, it defines the destination address used for transmission. To transmit using a 16-bit address, set DH parameter to zero and DL less than 0xFFFF. 0x000000000000FFFF is the broadcast address for the PAN.	0 - 0xFFFFFFFF	0
DL	Networking {Addressing}	Destination Address Low. Set/Read the lower 32 bits of the 64-bit destination address. When combined with DH, DL defines the destination address used for transmission. To transmit using a 16-bit address, set DH parameter to zero and DL less than 0xFFFF. 0x000000000000FFFF is the broadcast address for the PAN.	0 - 0xFFFFFFFF	0
MY	Networking {Addressing}	16-bit Source Address. Set/Read the RF module 16-bit source address. Set MY = 0xFFFF to disable reception of packets with 16-bit addresses. 64-bit source address (serial number) and broadcast address (0x000000000000FFFF) is always enabled.	0 - 0xFFFF	0
SH	Networking {Addressing}	Serial Number High. Read high 32 bits of the RF module's unique IEEE 64-bit address. 64-bit source address is always enabled.	0 - 0xFFFFFFFF [read-only]	Factory-set
SL	Networking {Addressing}	Serial Number Low. Read low 32 bits of the RF module's unique IEEE 64-bit address. 64-bit source address is always enabled.	0 - 0xFFFFFFFF [read-only]	Factory-set
RR (v1.xA0*)	Networking {Addressing}	XBee Retries. Set/Read the maximum number of retries the module will execute in addition to the 3 retries provided by the 802.15.4 MAC. For each XBee retry, the 802.15.4 MAC can execute up to 3 retries.	0 - 6	0
RN	Networking {Addressing}	Random Delay Slots. Set/Read the minimum value of the back-off exponent in the CSMA-CA algorithm that is used for collision avoidance. If RN = 0, collision avoidance is disabled during the first iteration of the algorithm (802.15.4 - macMinBE).	0 - 3 [exponent]	0
MM (v1.x80*)	Networking {Addressing}	MAC Mode. MAC Mode. Set/Read MAC Mode value. MAC Mode enables/disables the use of a Digi header in the 802.15.4 RF packet. When Modes 0 or 3 are enabled (MM=0,3), duplicate packet detection is enabled as well as certain AT commands. Please see the detailed MM description on page 47 for additional information.	0 - 3 0 = Digi Mode 1 = 802.15.4 (no ACKs) 2 = 802.15.4 (with ACKs) 3 = Digi Mode (no ACKs)	0
NI (v1.x80*)	Networking {Identification}	Node Identifier. Stores a string identifier. The register only accepts printable ASCII data. A string can not start with a space. Carriage return ends command. Command will automatically end when maximum bytes for the string have been entered. This string is returned as part of the ND (Node Discover) command. This identifier is also used with the DN (Destination Node) command.	20-character ASCII string	-
ND (v1.x80*)	Networking {Identification}	Node Discover. Discovers and reports all RF modules found. The following information is reported for each module discovered (the example cites use of Transparent operation (AT command format) - refer to the long ND command description regarding differences between Transparent and API operation). MY<CR> SH<CR> SL<CR> DB<CR> NI<CR><CR> The amount of time the module allows for responses is determined by the NT parameter. In Transparent operation, command completion is designated by a <CR> (carriage return). ND also accepts a Node Identifier as a parameter. In this case, only a module matching the supplied identifier will respond. If ND self-response is enabled (NO=1) the module initiating the node discover will also output a response for itself.	optional 20-character NI value	
NT (v1.xA0*)	Networking {Identification}	Node Discover Time. Set/Read the amount of time a node will wait for responses from other nodes when using the ND (Node Discover) command.	0x01 - 0xFC [x 100 ms]	0x19

Table 3-02. XBee®/XBee-PRO® Commands - Networking & Security (Sub-categories designated within {brackets})

AT Command	Command Category	Name and Description	Parameter Range	Default
NO (v1xC5)	Networking {Identification}	Node Discover Options. Enables node discover self-response on the module.	0-1	0
DN (v1.x80*)	Networking {Identification}	Destination Node. Resolves an NI (Node Identifier) string to a physical address. The following events occur upon successful command execution: 1. DL and DH are set to the address of the module with the matching Node Identifier. 2. "OK" is returned. 3. RF module automatically exits AT Command Mode If there is no response from a module within 200 msec or a parameter is not specified (left blank), the command is terminated and an "ERROR" message is returned.	20-character ASCII string	-
CE (v1.x80*)	Networking {Association}	Coordinator Enable. Set/Read the coordinator setting.	0 - 1 0 = End Device 1 = Coordinator	0
SC (v1.x80*)	Networking {Association}	Scan Channels. Set/Read list of channels to scan for all Active and Energy Scans as a bitfield. This affects scans initiated in command mode (AS, ED) and during End Device Association and Coordinator startup: bit 0 - 0x0B bit 4 - 0x0F bit 8 - 0x13 bit12 - 0x17 bit 1 - 0x0C bit 5 - 0x10 bit 9 - 0x14 bit13 - 0x18 bit 2 - 0x0D bit 6 - 0x11 bit 10 - 0x15 bit14 - 0x19 bit 3 - 0x0E bit 7 - 0x12 bit 11 - 0x16 bit 15 - 0x1A	0 - 0xFFFF [bitfield] (bits 0, 14, 15 not allowed on the XBee-PRO)	0x1FFE (all XBee-PRO Channels)
SD (v1.x80*)	Networking {Association}	Scan Duration. Set/Read the scan duration exponent. End Device - Duration of Active Scan during Association. Coordinator - If 'ReassignPANID' option is set on Coordinator [refer to A2 parameter], SD determines the length of time the Coordinator will scan channels to locate existing PANs. If 'ReassignChannel' option is set, SD determines how long the Coordinator will perform an Energy Scan to determine which channel it will operate on. 'Scan Time' is measured as (# of channels to scan) * (2 ^ SD) * 15.36ms). The number of channels to scan is set by the SC command. The XBee can scan up to 16 channels (SC = 0xFFFF). The XBee PRO can scan up to 13 channels (SC = 0x3FFE). Example: The values below show results for a 13 channel scan: If SD = 0, time = 0.18 sec SD = 8, time = 47.19 sec SD = 2, time = 0.74 sec SD = 10, time = 3.15 min SD = 4, time = 2.95 sec SD = 12, time = 12.58 min SD = 6, time = 11.80 sec SD = 14, time = 50.33 min	0-0x0F [exponent]	4
A1 (v1.x80*)	Networking {Association}	End Device Association. Set/Read End Device association options. bit 0 - ReassignPanID 0 - Will only associate with Coordinator operating on PAN ID that matches module ID 1 - May associate with Coordinator operating on any PAN ID bit 1 - ReassignChannel 0 - Will only associate with Coordinator operating on matching CH Channel setting 1 - May associate with Coordinator operating on any Channel bit 2 - AutoAssociate 0 - Device will not attempt Association 1 - Device attempts Association until success Note: This bit is used only for Non-Beacon systems. End Devices in Beacon-enabled system must always associate to a Coordinator bit 3 - PollCoordOnPinWake 0 - Pin Wake will not poll the Coordinator for indirect (pending) data 1 - Pin Wake will send Poll Request to Coordinator to extract any pending data bits 4 - 7 are reserved	0 - 0x0F [bitfield]	0
A2 (v1.x80*)	Networking {Association}	Coordinator Association. Set/Read Coordinator association options. bit 0 - ReassignPanID 0 - Coordinator will not perform Active Scan to locate available PAN ID. It will operate on ID (PAN ID). 1 - Coordinator will perform Active Scan to determine an available ID (PAN ID). If a PAN ID conflict is found, the ID parameter will change. bit 1 - ReassignChannel - 0 - Coordinator will not perform Energy Scan to determine free channel. It will operate on the channel determined by the CH parameter. 1 - Coordinator will perform Energy Scan to find a free channel, then operate on that channel. bit 2 - AllowAssociation - 0 - Coordinator will not allow any devices to associate to it. 1 - Coordinator will allow devices to associate to it. bits 3 - 7 are reserved	0 - 7 [bitfield]	0

Table 3-02. XBee®/XBee-PRO® Commands - Networking & Security (Sub-categories designated within [brackets])

AT Command	Command Category	Name and Description	Parameter Range	Default
AI (v1.x80*)	Networking {Association}	Association Indication. Read errors with the last association request: 0x00 - Successful Completion - Coordinator successfully started or End Device association complete 0x01 - Active Scan Timeout 0x02 - Active Scan found no PANs 0x03 - Active Scan found PAN, but the CoordinatorAllowAssociation bit is not set 0x04 - Active Scan found PAN, but Coordinator and End Device are not configured to support beacons 0x05 - Active Scan found PAN, but the Coordinator ID parameter does not match the ID parameter of the End Device 0x06 - Active Scan found PAN, but the Coordinator CH parameter does not match the CH parameter of the End Device 0x07 - Energy Scan Timeout 0x08 - Coordinator start request failed 0x09 - Coordinator could not start due to invalid parameter 0x0A - Coordinator Realignment is in progress 0x0B - Association Request not sent 0x0C - Association Request timed out - no reply was received 0x0D - Association Request had an Invalid Parameter 0x0E - Association Request Channel Access Failure. Request was not transmitted - CCA failure 0x0F - Remote Coordinator did not send an ACK after Association Request was sent 0x10 - Remote Coordinator did not reply to the Association Request, but an ACK was received after sending the request 0x11 - [reserved] 0x12 - Sync-Loss - Lost synchronization with a Beaconing Coordinator 0x13 - Disassociated - No longer associated to Coordinator 0xFF - RF Module is attempting to associate	0 - 0x13 [read-only]	-
DA (v1.x80*)	Networking {Association}	Force Disassociation. End Device will immediately disassociate from a Coordinator (if associated) and reattempt to associate.	-	-
FP (v1.x80*)	Networking {Association}	Force Poll. Request indirect messages being held by a coordinator.	-	-
AS (v1.x80*)	Networking {Association}	Active Scan. Send Beacon Request to Broadcast Address (0xFFFF) and Broadcast PAN (0xFFFF) on every channel. The parameter determines the time the radio will listen for Beacons on each channel. A PanDescriptor is created and returned for every Beacon received from the scan. Each PanDescriptor contains the following information: CoordAddress (SH, SL)<CR> CoordPanID (ID)<CR> CoordAddrMode <CR> 0x02 = 16-bit Short Address 0x03 = 64-bit Long Address Channel (CH parameter) <CR> SecurityUse<CR> ACLEntry<CR> SecurityFailure<CR> SuperFrameSpec<CR> (2 bytes): bit 15 - Association Permitted (MSB) bit 14 - PAN Coordinator bit 13 - Reserved bit 12 - Battery Life Extension bits 8-11 - Final CAP Slot bits 4-7 - Superframe Order bits 0-3 - Beacon Order GtsPermit<CR> RSSI<CR> (RSSI is returned as -dBm) TimeStamp<CR> (3 bytes) <CR> A carriage return <CR> is sent at the end of the AS command. The Active Scan is capable of returning up to 5 PanDescriptors in a scan. The actual scan time on each channel is measured as Time = [(2 ^SD PARAM) * 15.36] ms. Note the total scan time is this time multiplied by the number of channels to be scanned (16 for the XBee and 13 for the XBee-PRO). Also refer to SD command description.	0 - 6	-
ED (v1.x80*)	Networking {Association}	Energy Scan. Send an Energy Detect Scan. This parameter determines the length of scan on each channel. The maximal energy on each channel is returned & each value is followed by a carriage return. An additional carriage return is sent at the end of the command. The values returned represent the detected energy level in units of -dBm. The actual scan time on each channel is measured as Time = [(2 ^ED) * 15.36] ms. Note the total scan time is this time multiplied by the number of channels to be scanned (refer to SD parameter).	0 - 6	-
EE (v1.xA0*)	Networking {Security}	AES Encryption Enable. Disable/Enable 128-bit AES encryption support. Use in conjunction with the KY command.	0 - 1	0 (disabled)
KY (v1.xA0*)	Networking {Security}	AES Encryption Key. Set the 128-bit AES (Advanced Encryption Standard) key for encrypting/decrypting data. The KY register cannot be read.	0 - (any 16-Byte value)	-

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

RF Interfacing**Table 3-03. XBee/XBee-PRO Commands - RF Interfacing**

AT Command	Command Category	Name and Description	Parameter Range	Default
PL	RF Interfacing	Power Level. Select/Read the power level at which the RF module transmits conducted power.	0 - 4 (XBee / XBee-PRO) 0 = -10 / 10 dBm 1 = -6 / 12 dBm 2 = -4 / 14 dBm 3 = -2 / 16 dBm 4 = 0 / 18 dBm XBee-PRO International variant: PL=4: 10 dBm PL=3: 8 dBm PL=2: 2 dBm PL=1: -3 dBm PL=0: -3 dBm	4
CA (v1.x80*)	RF Interfacing	CCA Threshold. Set/read the CCA (Clear Channel Assessment) threshold. Prior to transmitting a packet, a CCA is performed to detect energy on the channel. If the detected energy is above the CCA Threshold, the module will not transmit the packet.	0x24 - 0x50 [-dBm]	0x2C (-44d dBm)

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

Sleep (Low Power)**Table 3-04. XBee®/XBee-PRO® Commands - Sleep (Low Power)**

AT Command	Command Category	Name and Description	Parameter Range	Default
SM	Sleep (Low Power)	Sleep Mode. Set/Read Sleep Mode configurations.	0 - 5 0 = No Sleep 1 = Pin Hibernate 2 = Pin Doze 3 = Reserved 4 = Cyclic sleep remote 5 = Cyclic sleep remote w/ pin wake-up 6 = [Sleep Coordinator] for backwards compatibility w/ v1.x6 only; otherwise, use CE command.	0
SO	Sleep (Low Power)	Sleep Options Set/Read the sleep mode options. Bit 0 - Poll wakeup disable 0 - Normal operations. A module configured for cyclic sleep will poll for data on waking. 1 - Disable wakeup poll. A module configured for cyclic sleep will not poll for data on waking. Bit 1 - ADC/DIO wakeup sampling disable. 0 - Normal operations. A module configured in a sleep mode with ADC/DIO sampling enabled will automatically perform a sampling on wakeup. 1 - Suppress sample on wakeup. A module configured in a sleep mode with ADC/DIO sampling enabled will not automatically sample on wakeup.	0-4	0
ST	Sleep (Low Power)	Time before Sleep. <NonBeacon firmware> Set/Read time period of inactivity (no serial or RF data is sent or received) before activating Sleep Mode. ST parameter is only valid with Cyclic Sleep Settings (SM = 4 - 5). Coordinator and End Device ST values must be equal. Also note, the GT parameter value must always be less than the ST value. (If GT > ST, the configuration will render the module unable to enter into command mode.) If the ST parameter is modified, also modify the GT parameter accordingly.	1 - 0xFFFF [x 1 ms]	0x1388 (5000d)
SP	Sleep (Low Power)	Cyclic Sleep Period. <NonBeacon firmware> Set/Read sleep period for cyclic sleeping remotes. Coordinator and End Device SP values should always be equal. To send Direct Messages, set SP = 0. End Device - SP determines the sleep period for cyclic sleeping remotes. Maximum sleep period is 268 seconds (0x68B0). Coordinator - If non-zero, SP determines the time to hold an indirect message before discarding it. A Coordinator will discard indirect messages after a period of (2.5 * SP).	0 - 0x68B0 [x 10 ms]	0
DP (1.x80*)	Sleep (Low Power)	Disassociated Cyclic Sleep Period. <NonBeacon firmware> End Device - Set/Read time period of sleep for cyclic sleeping remotes that are configured for Association but are not associated to a Coordinator. (i.e. If a device is configured to associate, configured as a Cyclic Sleep remote, but does not find a Coordinator, it will sleep for DP time before reattempting association.) Maximum sleep period is 268 seconds (0x68B0). DP should be > 0 for NonBeacon systems.	1 - 0x68B0 [x 10 ms]	0x3E8 (1000d)

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

Serial Interfacing

Table 3-05. XBee-PRO Commands - Serial Interfacing

AT Command	Command Category	Name and Description	Parameter Range	Default
BD	Serial Interfacing	Interface Data Rate. Set/Read the serial interface data rate for communications between the RF module serial port and host. Request non-standard baud rates with values above 0x80 using a terminal window. Read the BD register to find actual baud rate achieved.	0 - 7 (standard baud rates) 0 = 1200 bps 1 = 2400 2 = 4800 3 = 9600 4 = 19200 5 = 38400 6 = 57600 7 = 115200 0x80 - 0x3D090 (non-standard baud rates up to 250 Kbps)	3
RO	Serial Interfacing	Packetization Timeout. Set/Read number of character times of inter-character delay required before transmission. Set to zero to transmit characters as they arrive instead of buffering them into one RF packet.	0 - 0xFF [x character times]	3
AP (v1.x80*)	Serial Interfacing	API Enable. Disable/Enable API Mode.	0 - 2 0 = Disabled 1 = API enabled 2 = API enabled (w/escaped control characters)	0
NB	Serial Interfacing	Parity. Set/Read parity settings.	0 - 4 0 = 8-bit no parity 1 = 8-bit even 2 = 8-bit odd 3 = 8-bit mark 4 = 8-bit space	0
PR (v1.x80*)	Serial Interfacing	Pull-up Resistor Enable. Set/Read bitfield to configure internal pull-up resistor status for I/O lines Bitfield Map: bit 0 - AD4/DIO4 (pin11) bit 1 - AD3 / DIO3 (pin17) bit 2 - AD2/DIO2 (pin18) bit 3 - AD1/DIO1 (pin19) bit 4 - AD0 / DIO0 (pin20) bit 5 - RTS / AD6 / DIO6 (pin16) bit 6 - DTR / SLEEP_RQ / DI8 (pin9) bit 7 - DIN/CONFIG (pin3) Bit set to "1" specifies pull-up enabled; "0" specifies no pull-up	0 - 0xFF	0xFF

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

I/O Settings

Table 3-06. XBee-PRO Commands - I/O Settings (sub-category designated within {brackets})

AT Command	Command Category	Name and Description	Parameter Range	Default
D8	I/O Settings	DI8 Configuration. Select/Read options for the DI8 line (pin 9) of the RF module.	0 - 1 0 = Disabled 3 = DI (1,2,4 & 5 n/a)	0
D7 (v1.x80*)	I/O Settings	DIO7 Configuration. Select/Read settings for the DIO7 line (pin 12) of the RF module. Options include CTS flow control and I/O line settings.	0 - 1 0 = Disabled 1 = CTS Flow Control 2 = (n/a) 3 = DI 4 = DO low 5 = DO high 6 = RS485 Tx Enable Low 7 = RS485 Tx Enable High	1
D6 (v1.x80*)	I/O Settings	DIO6 Configuration. Select/Read settings for the DIO6 line (pin 16) of the RF module. Options include RTS flow control and I/O line settings.	0 - 1 0 = Disabled 1 = RTS flow control 2 = (n/a) 3 = DI 4 = DO low 5 = DO high	0

Table 3-06. XBee-PRO Commands - I/O Settings (sub-category designated within [brackets])

AT Command	Command Category	Name and Description	Parameter Range	Default
D5 (v1.x80*)	I/O Settings	DIO5 Configuration. Configure settings for the DIO5 line (pin 15) of the RF module. Options include Associated LED indicator (blinks when associated) and I/O line settings.	0 - 1 0 = Disabled 1 = Associated indicator 2 = ADC 3 = DI 4 = DO low 5 = DO high	1
D0 - D4 (v1.xA0*)	I/O Settings	(DIO4 - DIO4) Configuration. Select/Read settings for the following lines: AD0/DIO0 (pin 20), AD1/DIO1 (pin 19), AD2/DIO2 (pin 18), AD3/DIO3 (pin 17), AD4/DIO4 (pin 11). Options include: Analog-to-digital converter, Digital Input and Digital Output.	0 - 1 0 = Disabled 1 = (n/a) 2 = ADC 3 = DI 4 = DO low 5 = DO high	0
IU (v1.xA0*)	I/O Settings	I/O Output Enable. Disables/Enables I/O data received to be sent out UART. The data is sent using an API frame regardless of the current AP parameter value.	0 - 1 0 = Disabled 1 = Enabled	1
IT (v1.xA0*)	I/O Settings	Samples before TX. Set/Read the number of samples to collect before transmitting data. Maximum number of samples is dependent upon the number of enabled inputs.	1 - 0xFF	1
IS (v1.xA0*)	I/O Settings	Force Sample. Force a read of all enabled inputs (DI or ADC). Data is returned through the UART. If no inputs are defined (DI or ADC), this command will return error.	8-bit bitmap (each bit represents the level of an I/O line setup as an output)	-
IO (v1.xA0*)	I/O Settings	Digital Output Level. Set digital output level to allow DIO lines that are setup as outputs to be changed through Command Mode.	-	-
IC (v1.xA0*)	I/O Settings	DIO Change Detect. Set/Read bitfield values for change detect monitoring. Each bit enables monitoring of DIO0 - DIO7 for changes. If detected, data is transmitted with DIO data only. Any samples queued waiting for transmission will be sent first.	0 - 0xFF [bitfield]	0 (disabled)
IR (v1.xA0*)	I/O Settings	Sample Rate. Set/Read sample rate. When set, this parameter causes the module to sample all enabled inputs at a specified interval.	0 - 0xFFFF [x 1 msec]	0
IA (v1.xA0*)	I/O Settings {I/O Line Passing}	I/O Input Address. Set/Read addresses of module to which outputs are bound. Setting all bytes to 0xFF will not allow any received I/O packet to change outputs. Setting address to 0xFFFF will allow any received I/O packet to change outputs.	0 - 0xFFFFFFFFFFFFFFFF	0xFFFFFFFFFFFFFFFF
T0 - T7 (v1.xA0*)	I/O Settings {I/O Line Passing}	(D0 - D7) Output Timeout. Set/Read Output timeout values for lines that correspond with the D0 - D7 parameters. When output is set (due to I/O line passing) to a non-default level, a timer is started which when expired will set the output to its default level. The timer is reset when a valid I/O packet is received.	0 - 0xFF [x 100 ms]	0xFF
P0	I/O Settings {I/O Line Passing}	PWM0 Configuration. Select/Read function for PWM0 pin.	0 - 2 0 = Disabled 1 = RSSI 2 = PWM Output	1
P1 (v1.xA0*)	I/O Settings {I/O Line Passing}	PWM1 Configuration. Select/Read function for PWM1 pin.	0 - 2 0 = Disabled 1 = RSSI 2 = PWM Output	0
M0 (v1.xA0*)	I/O Settings {I/O Line Passing}	PWM0 Output Level. Set/Read the PWM0 output level.	0 - 0x03FF	-
M1 (v1.xA0*)	I/O Settings {I/O Line Passing}	PWM1 Output Level. Set/Read the PWM1 output level.	0 - 0x03FF	-
PT (v1.xA0*)	I/O Settings {I/O Line Passing}	PWM Output Timeout. Set/Read output timeout value for both PWM outputs. When PWM is set to a non-zero value: Due to I/O line passing, a time is started which when expired will set the PWM output to zero. The timer is reset when a valid I/O packet is received.]	0 - 0xFF [x 100 ms]	0xFF
RP	I/O Settings {I/O Line Passing}	RSSI PWM Timer. Set/Read PWM timer register. Set the duration of PWM (pulse width modulation) signal output on the RSSI pin. The signal duty cycle is updated with each received packet and is shut off when the timer expires.]	0 - 0xFF [x 100 ms]	0x28 (40d)

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

Diagnostics**Table 3-07. XBee®/XBee-PRO® Commands - Diagnostics**

AT Command	Command Category	Name and Description	Parameter Range	Default
VR	Diagnostics	Firmware Version. Read firmware version of the RF module.	0 - 0xFFFF [read-only]	Factory-set
VL (v1.x80*)	Diagnostics	Firmware Version - Verbose. Read detailed version information (including application build date, MAC, PHY and bootloader versions). The VL command has been deprecated in version 10C9. It is not supported in firmware versions after 10C8	-	-

Table 3-07. XBee®/XBee-PRO® Commands - Diagnostics

AT Command	Command Category	Name and Description	Parameter Range	Default
HV (v1.x80*)	Diagnostics	Hardware Version. Read hardware version of the RF module.	0 - 0xFFFF [read-only]	Factory-set
DB	Diagnostics	Received Signal Strength. Read signal level [in dB] of last good packet received (RSSI). Absolute value is reported. (For example: 0x58 = -88 dBm) Reported value is accurate between -40 dBm and RX sensitivity.	0x17-0x5C (XBee) 0x24-0x64 (XBee-PRO) [read-only]	-
EC (v1.x80*)	Diagnostics	CCA Failures. Reset/Read count of CCA (Clear Channel Assessment) failures. This parameter value increments when the module does not transmit a packet because it detected energy above the CCA threshold level set with CA command. This count saturates at its maximum value. Set count to "0" to reset count.	0 - 0xFFFF	-
EA (v1.x80*)	Diagnostics	ACK Failures. Reset/Read count of acknowledgment failures. This parameter value increments when the module expires its transmission retries without receiving an ACK on a packet transmission. This count saturates at its maximum value. Set the parameter to "0" to reset count.	0 - 0xFFFF	-
ED (v1.x80*)	Diagnostics	Energy Scan. Send 'Energy Detect Scan'. ED parameter determines the length of scan on each channel. The maximal energy on each channel is returned and each value is followed by a carriage return. Values returned represent detected energy levels in units of -dBm. Actual scan time on each channel is measured as Time = [(2 ^ SD) * 15.36] ms. Total scan time is this time multiplied by the number of channels to be scanned.	0 - 6	-

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

AT Command Options**Table 3-08. XBee®/XBee-PRO® Commands - AT Command Options**

AT Command	Command Category	Name and Description	Parameter Range	Default
CT	AT Command Mode Options	Command Mode Timeout. Set/Read the period of inactivity (no valid commands received) after which the RF module automatically exits AT Command Mode and returns to Idle Mode.	2 - 0xFFFF [x 100 ms]	0x64 (100d)
CN	AT Command Mode Options	Exit Command Mode. Explicitly exit the module from AT Command Mode.	--	--
AC (v1.xA0*)	AT Command Mode Options	Apply Changes. Explicitly apply changes to queued parameter value(s) and re-initialize module.	--	--
GT	AT Command Mode Options	Guard Times. Set required period of silence before and after the Command Sequence Characters of the AT Command Mode Sequence (GT+ CC + GT). The period of silence is used to prevent inadvertent entrance into AT Command Mode.	2 - 0x0CE4 [x 1 ms]	0x3E8 (1000d)
CC	AT Command Mode Options	Command Sequence Character. Set/Read the ASCII character value to be used between Guard Times of the AT Command Mode Sequence (GT+CC+GT). The AT Command Mode Sequence enters the RF module into AT Command Mode.	0 - 0xFF	0x2B ('+' ASCII)

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

API Operation

By default, XBee®/XBee-PRO® RF Modules act as a serial line replacement (Transparent Operation) - all UART data received through the DI pin is queued up for RF transmission. When the module receives an RF packet, the data is sent out the DO pin with no additional information.

Inherent to Transparent Operation are the following behaviors:

- If module parameter registers are to be set or queried, a special operation is required for transitioning the module into Command Mode.
- In point-to-multipoint systems, the application must send extra information so that the receiving module(s) can distinguish between data coming from different remotes.

As an alternative to the default Transparent Operation, API (Application Programming Interface) Operations are available. API operation requires that communication with the module be done through a structured interface (data is communicated in frames in a defined order). The API specifies how commands, command responses and module status messages are sent and received from the module using a UART Data Frame.

API Frame Specifications

Two API modes are supported and both can be enabled using the AP (API Enable) command. Use the following AP parameter values to configure the module to operate in a particular mode:

- AP = 0 (default): Transparent Operation (UART Serial line replacement)
API modes are disabled.
- AP = 1: API Operation
- AP = 2: API Operation (with escaped characters)

Any data received prior to the start delimiter is silently discarded. If the frame is not received correctly or if the checksum fails, the data is silently discarded.

API Operation (AP parameter = 1)

When this API mode is enabled (AP = 1), the UART data frame structure is defined as follows:

Figure 3-01. UART Data Frame Structure:



MSB = Most Significant Byte, LSB = Least Significant Byte

API Operation - with Escape Characters (AP parameter = 2)

When this API mode is enabled (AP = 2), the UART data frame structure is defined as follows:

Figure 3-02. UART Data Frame Structure - with escape control characters:



MSB = Most Significant Byte, LSB = Least Significant Byte

Escape characters. When sending or receiving a UART data frame, specific data values must be escaped (flagged) so they do not interfere with the UART or UART data frame operation. To escape an interfering data byte, insert 0x7D and follow it with the byte to be escaped XOR'd with 0x20.

Data bytes that need to be escaped:

- 0x7E – Frame Delimiter
- 0x7D – Escape
- 0x11 – XON
- 0x13 – XOFF

Example - Raw UART Data Frame (before escaping interfering bytes):

0x7E 0x00 0x02 0x23 0x11 0xCB

0x11 needs to be escaped which results in the following frame:

0x7E 0x00 0x02 0x23 0x7D 0x31 0xCB

Note: In the above example, the length of the raw data (excluding the checksum) is 0x0002 and the checksum of the non-escaped data (excluding frame delimiter and length) is calculated as:
 $0xFF - (0x23 + 0x11) = (0xFF - 0x34) = 0xCB$.

Checksum

To test data integrity, a checksum is calculated and verified on non-escaped data.

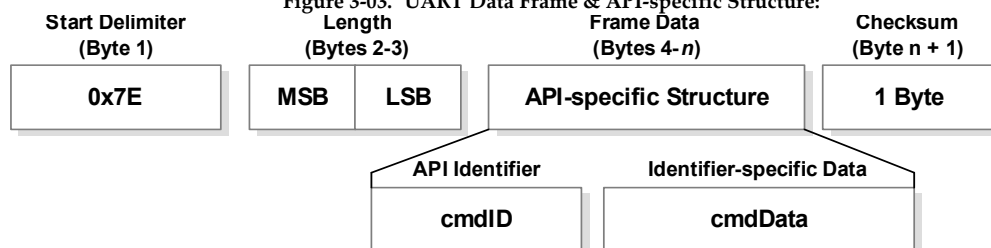
To calculate: Not including frame delimiters and length, add all bytes keeping only the lowest 8 bits of the result and subtract from 0xFF.

To verify: Add all bytes (include checksum, but not the delimiter and length). If the checksum is correct, the sum will equal 0xFF.

API Types

Frame data of the UART data frame forms an API-specific structure as follows:

Figure 3-03. UART Data Frame & API-specific Structure:



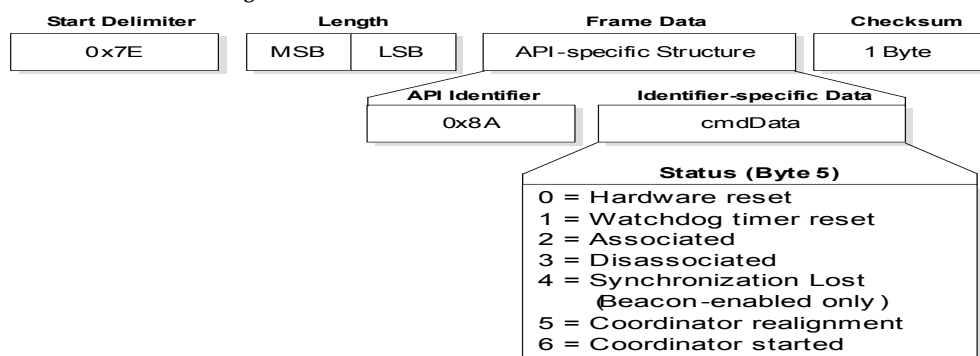
The cmdID frame (API-identifier) indicates which API messages will be contained in the cmdData frame (Identifier-specific data). Refer to the sections that follow for more information regarding the supported API types. Note that multi-byte values are sent big endian.

Modem Status

API Identifier: 0x8A

RF module status messages are sent from the module in response to specific conditions.

Figure 3-04. Modem Status Frames



AT Command

API Identifier Value: 0x08

The "AT Command" API type allows for module parameters to be queried or set. When using this command ID, new parameter values are applied immediately. This includes any register set with the "AT Command - Queue Parameter Value" (0x09) API type.

Figure 3-05. AT Command Frames

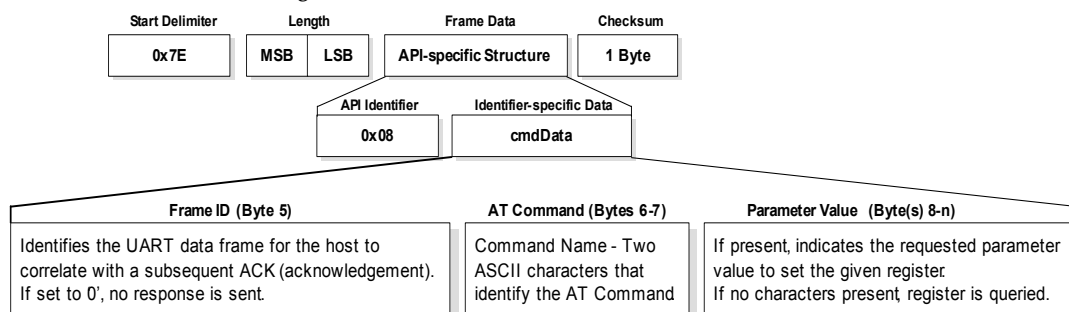
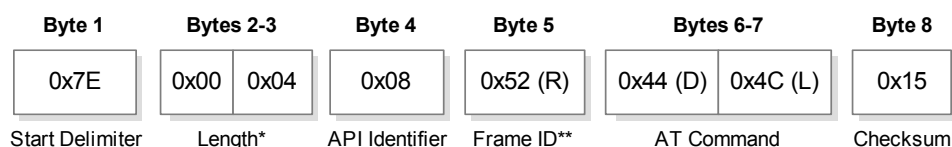


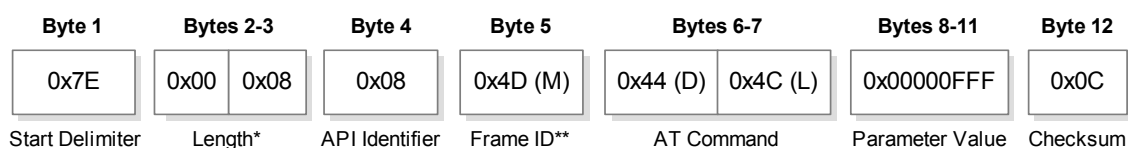
Figure 3-06. Example: API frames when reading the DL parameter value of the module.



* Length [Bytes] = API Identifier + Frame ID + AT Command

** "R" value was arbitrarily selected.

Figure 3-07. Example: API frames when modifying the DL parameter value of the module.



* Length [Bytes] = API Identifier + Frame ID + AT Command + Parameter Value

** "M" value was arbitrarily selected.

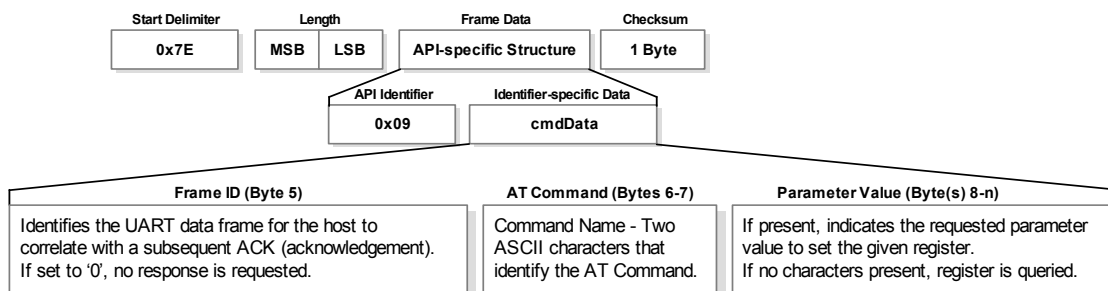
AT Command - Queue Parameter Value

API Identifier Value: 0x09

This API type allows module parameters to be queried or set. In contrast to the "AT Command" API type, new parameter values are queued and not applied until either the "AT Command" (0x08) API type or the AC (Apply Changes) command is issued. Register queries (reading parameter values) are returned immediately.

Figure 3-08. AT Command Frames

(Note that frames are identical to the "AT Command" API type except for the API identifier.)



AT Command Response

API Identifier Value: 0x88

Response to previous command.

In response to an AT Command message, the module will send an AT Command Response message. Some commands will send back multiple frames (for example, the ND (Node Discover) and AS (Active Scan) commands). These commands will end by sending a frame with a status of ATCMD_OK and no cmdData.

Figure 3-09. AT Command Response Frames.

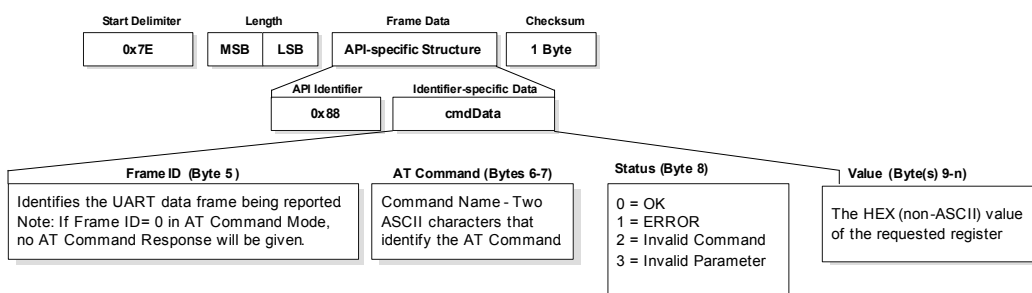
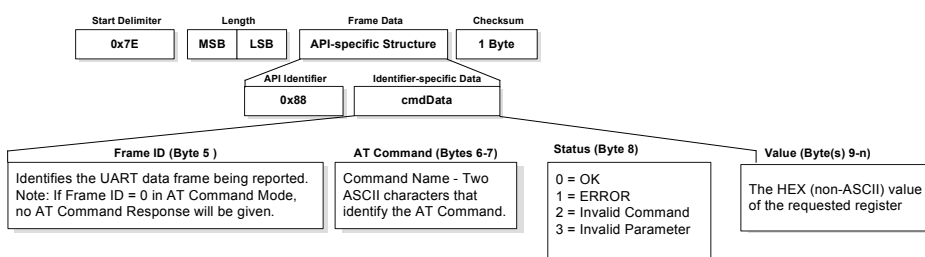


Figure 3-10. AT Command Response Frames.

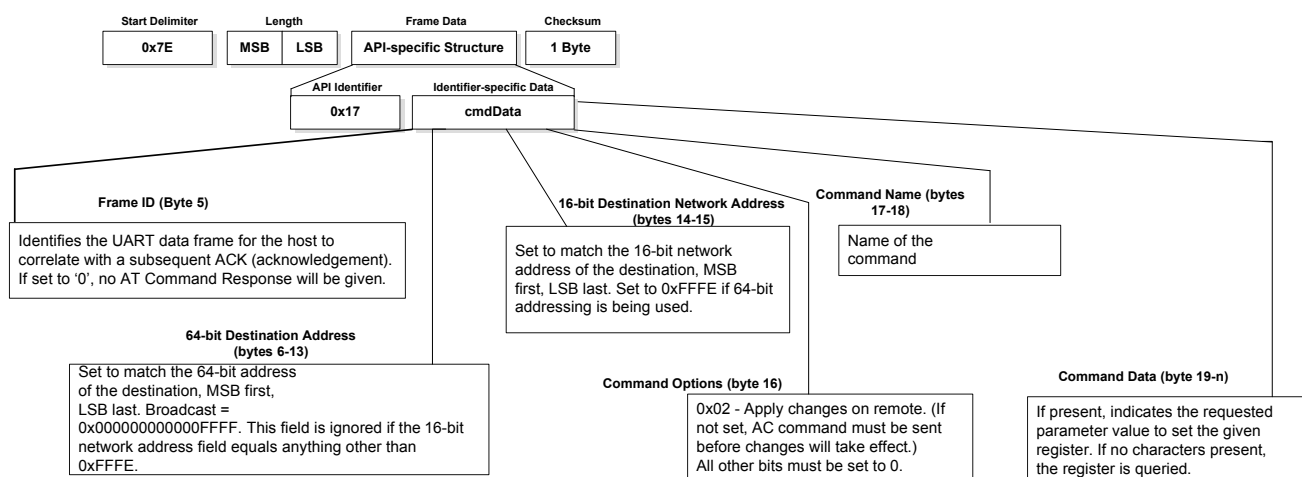


Remote AT Command Request

API Identifier Value: 0x17

Allows for module parameter registers on a remote device to be queried or set

Figure 3-11. Remote AT Command Request

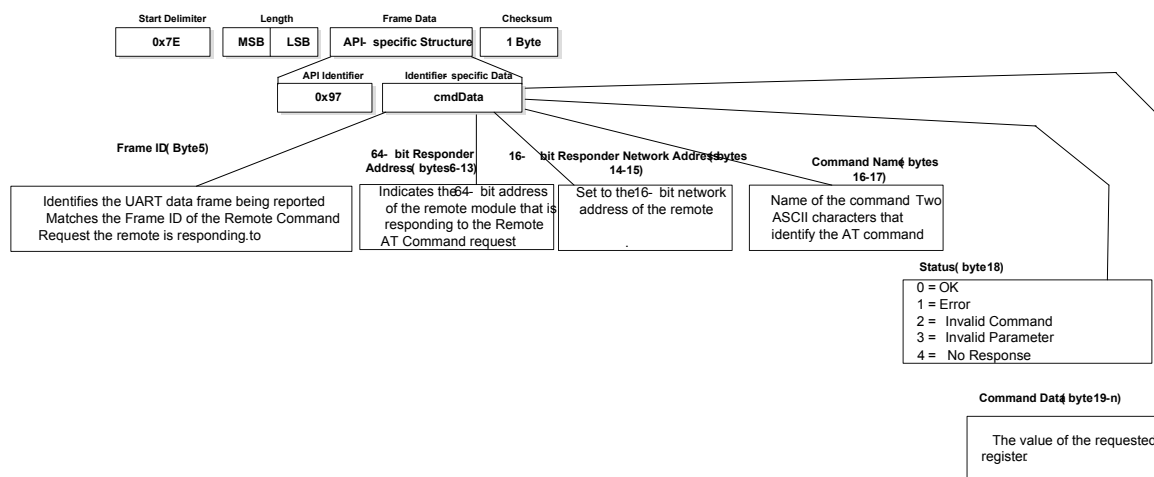


Remote Command Response

API Identifier Value: 0x97

If a module receives a remote command response RF data frame in response to a Remote AT Command Request, the module will send a Remote AT Command Response message out the UART. Some commands may send back multiple frames--for example, Node Discover (ND) command.

Figure 3-12. Remote AT Command Response.

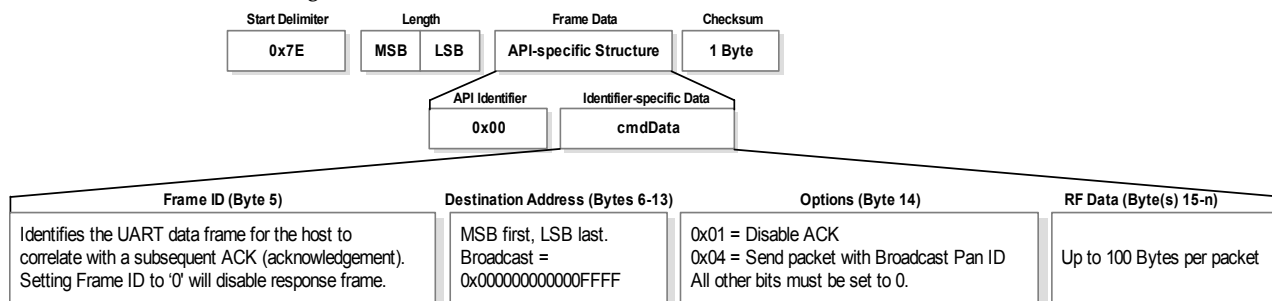


TX (Transmit) Request: 64-bit address

API Identifier Value: 0x00

A TX Request message will cause the module to send RF Data as an RF Packet.

Figure 3-13. TX Packet (64-bit address) Frames

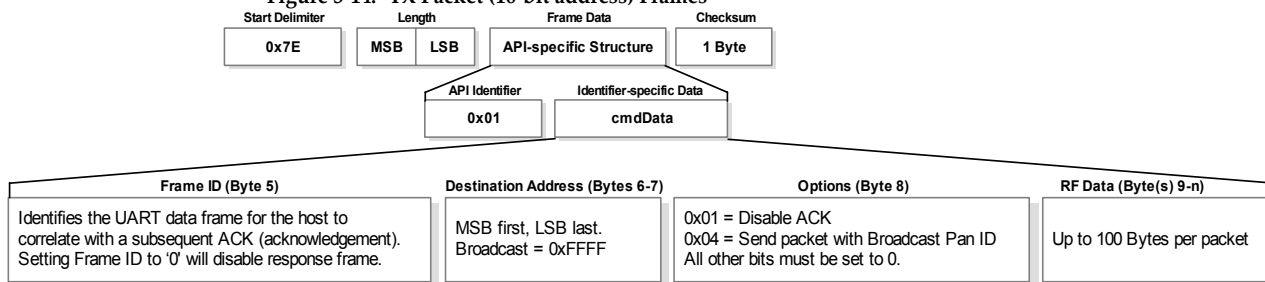


TX (Transmit) Request: 16-bit address

API Identifier Value: 0x01

A TX Request message will cause the module to send RF Data as an RF Packet.

Figure 3-14. TX Packet (16-bit address) Frames

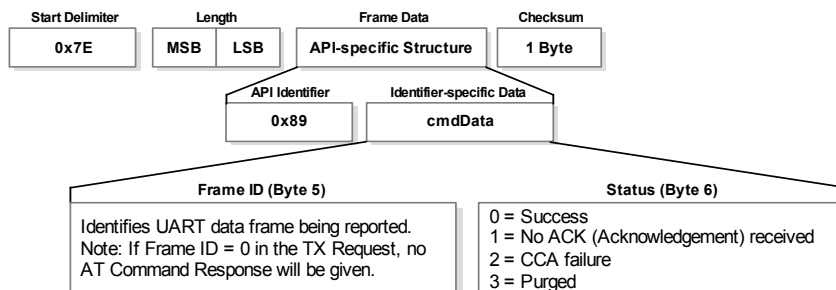


TX (Transmit) Status

API Identifier Value: 0x89

When a TX Request is completed, the module sends a TX Status message. This message will indicate if the packet was transmitted successfully or if there was a failure.

Figure 3-15. TX Status Frames



NOTES:

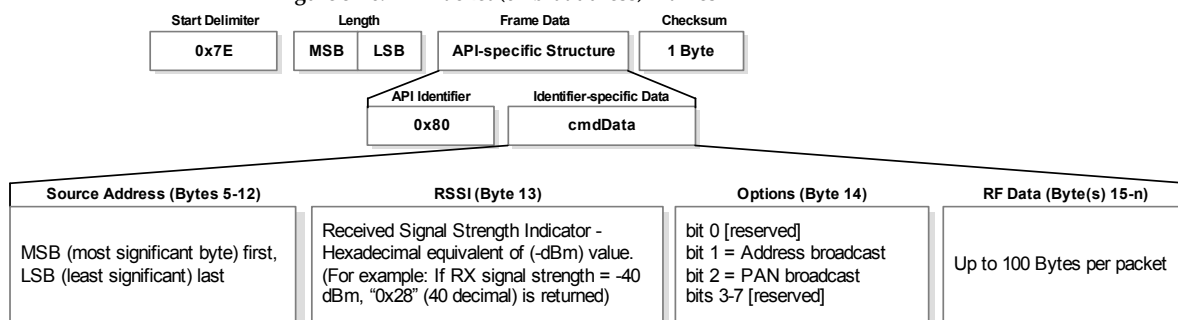
- "STATUS = 1" occurs when all retries are expired and no ACK is received.
- If transmitter broadcasts (destination address = 0x000000000000FFFF), only "STATUS = 0 or 2" will be returned.
- "STATUS = 3" occurs when Coordinator times out of an indirect transmission. Timeout is defined as (2.5 x SP (Cyclic Sleep Period) parameter value).

RX (Receive) Packet: 64-bit Address

API Identifier Value: 0x80

When the module receives an RF packet, it is sent out the UART using this message type.

Figure 3-16. RX Packet (64-bit address) Frames



RX (Receive) Packet: 16-bit Address

API Identifier Value: 0x81

When the module receives an RF packet, it is sent out the UART using this message type.

Figure 3-17. RX Packet (16-bit address) Frames

