



IJMIRD 2015; 2(3): 206-212
www.allsubjectjournal.com
Received: 25-02-2015
Accepted: 08-03-2015
E-ISSN: 2349-4182
P-ISSN: 2349-5979
Impact factor: 3.762

Shweta Maurya

Department of Electronics and
Communication Engineering,
World College of Technology and
Management, Maharshi
Dayanand University, Rohtak.

Ajay Dagar

Department of Electronics and
Communication Engineering,
World College of Technology and
Management, Maharshi
Dayanand University, Rohtak.

Lalit Sharma

Department of Electronics and
Communication Engineering,
World College of Technology and
Management, Maharshi
Dayanand University, Rohtak.

Correspondence:

Shweta Maurya

Department of Electronics and
Communication Engineering,
World College of Technology and
Management, Maharshi
Dayanand University, Rohtak.

Autonomous wheeled Robot Control in a coordinate system using Wireless Transmission

Shweta Maurya, Ajay Dagar, Lalit Sharma

Abstract

In this paper, an autonomous robot which is guided by coordinate system to perform some work at specified coordinate. The coordinate system is drawn manually on a sheet or using image processing a virtual navigation system can be created. By specifying the coordinates robot moves to the particular coordinates sequentially. The robot's chassis is constructed using perforated steel plates, two dc motors and a castor wheel and driven by differential drive mechanism. The entire system is controlled by PIC microcontroller 18f series (18f4550) and the dc motors are connected through L293D drivers. The sensors used for detection of obstacles and the coordinate lines are infra-red transmitter and detector. Other sensors are connected to the robot to measure temperature or pressure at particular coordinate or collect samples. The entire circuit is etched on a PCB (printed circuit board). Such systems are very useful at area where the place is hazardous or where a man can't go like mines, dangerous gases, or brittle areas where the land might collapse etc. One of the examples of such robot is land rover on mars created by NASA which is controlled through satellites from earth.

Keywords: PIC Microcontroller, Mobile Robot, L293D driver, Three Light sensor, C18 Microchip compiler, Emulator drivers, Wireless Transmission, XBee-PRO OEM RF Modules.

1. Introduction

Navigation is a major challenge for autonomous, mobile robots. The problem can basically be divided into positioning and path planning. In this paper we present an approach which we call grid-based navigation. Though we also propose a scheme for path finding, we focus on positioning. Our approach uses minimal environmental infrastructure and only two light sensors on the mobile device itself. Starting out from a predefined tile and orientation in the grid, the mobile robot can autonomously head for destination tiles in the grid, on its way it determines the current location in the grid using a finite state machine by picking up line crossing events with its sensors. A key ability needed by an autonomous, mobile robot is the possibility to navigate through the space. The problem can basically be decomposed into positioning and path planning. Though we also propose a scheme for the latter, we clearly focus on the detection of the current position by monitoring grid crossing events by using three light sensors. Especially if the robot is severely resource-constrained, simple schemes are favorable to elaborated algorithms. Rather simple sensors and actuators as well as a limited computing platform also demand simple, robust techniques due to inaccuracy and the lack of resources. In this, the following are structured: We give an overview of the construction of the chassis part of the robot and overview of the environment that the robot needs for navigation. We present the underlying algorithm and discuss various details. The software structure of the implementation is presented and the advancements in the paper are discussed.

The most important part of a robot system is its controller. Now days an obvious technical trend of robot system is to construct controller applying open PC software and hardware infrastructure, which can take full advantages of the relevant PC techniques to enhance robot controller's performance effectively.

2. Material and Methods

2.1 Chassis

The mechanical construction needs to be planned carefully. There has to be a balance between speed and torque. We used 100 rpm geared dc motors because they have high torque and its speed is considerable (for starters like us). In case higher rpm motors are selected the weight of the chassis has to be reduced. The relation between the torque and the speed is inverse. We have used wood of 3.4-4 mm thickness to prepare the chassis.

The chassis consisted of wood and steel perforated plates. The bottom one is 16x12cm and the upper one is 9x8cm. The two parts are held apart at a height of 5cm using nut and bolt of considerable strength. The dc motors are clamped to the bottom piece after drilling and screwed tightly

2.2 Differential drive

The differential drive is a two-wheeled drive system with independent actuators for each wheel. The name refers to the fact that the motion vector of the robot is sum of the

independent wheel motions, something that is also true of the mechanical differential (however, this drive system does not use a mechanical differential). The drive wheels are usually placed on each side of the robot and toward the front.

2.3 Hardware

The basic components for the microcontroller circuit to be active is the XTAL(crystal) circuit, reset circuit, battery source and 7805 and couples of led's to troubleshoot.

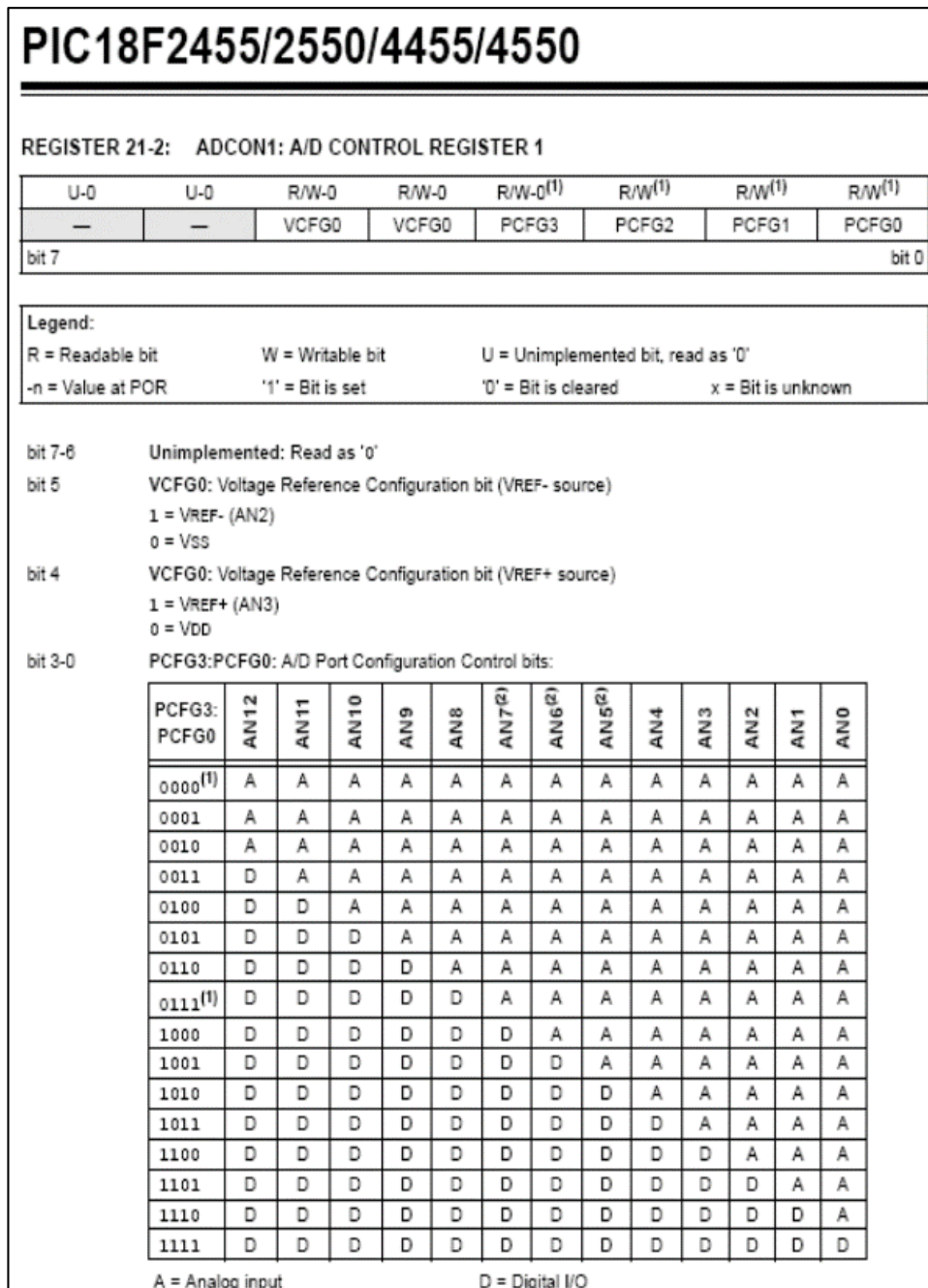


Fig 1: PIC (Peripheral Interface Controller) 18F2455 microcontroller and A/D Control Register.

The crystal being used is 11.0592 MHZ. Any crystal within the range 4MHZ and 48MHZ can be used. When the power is supplied the crystal oscillates at that frequency. The next circuit is the reset circuit. The reset pin in 18f4550 is MCLR and is active low. When logic 0 is sent to the pin the microcontroller is returned to the initial conditions. Then couple of led's are added to I/O pins. The circuit is ready and the controller can be checked if it is working properly.

The sensor pins are connected to the PORTA. The controller consists of 13 a dc pins which can be programmed either as digital or analog. They can be programmed by writing to the register ADCON [1]. The sensor's signal pin can be connected to any of these pins can be connected to any of these pins. We are using three IR sensors for this project connected to three I/O pins of the controller. For programming the controller while still embedded to the

circuit ICSP and USB can be used to dump the hex file into the controller.

Design, build, and demonstrate a robot which will traverse across a coordinate system autonomously, that is, without human intervention. The robot starts from origin (0,0) on the grid and then reach the destination coordinate. The inputs to the robot will be the destination point, i.e. x and y coordinates of that point. The path through which the robot has to travel can also be mentioned; otherwise the robot takes its own path to reach the end. The physical structure of the robot will be built using perforated steel plates and plywood. The weight of the chassis of the robot inclusive of all sensors, battery and boards should not exceed 1 Kg. The programming should be done using a PIC 18f4550 microcontroller. The code to generate the robot movement should be a generic code and universal so that the code can be changed simply in minutes rather than rewriting the code every time we need to change the coordinates. The inputs to the robot should be given using Wireless Transmission. The code for the serial port communication is generated using Lab VIEW software. Xbee modules should be used for the wireless transmission.

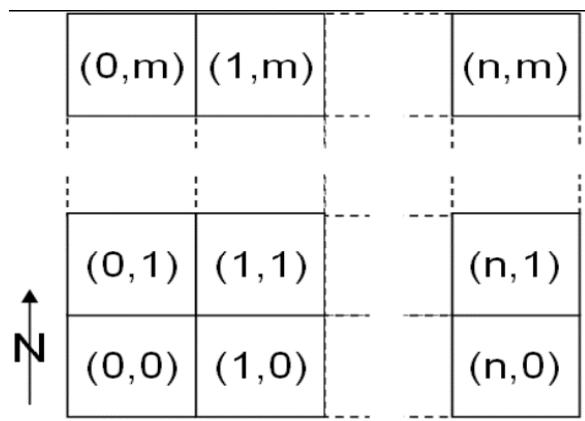


Fig 2: Manual Coordinate System representing.

(0,0) to (n,m)

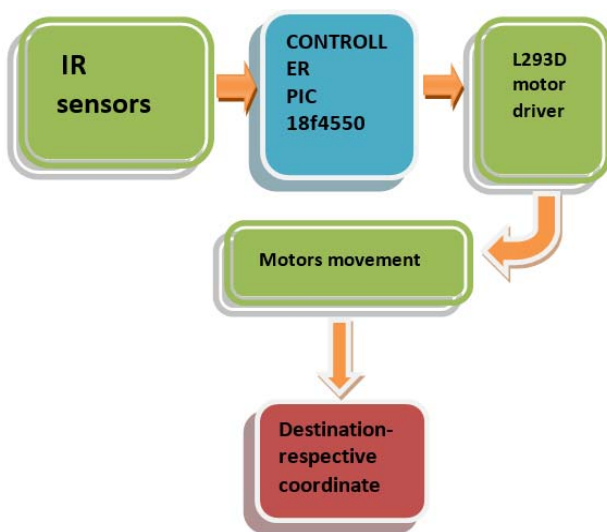


Fig 3: Block Diagram of Controller PIC 18f4550.

2.3.1 Voltage Regulator

This is most common voltage regulator that is still used in embedded designs. LM7805 voltage regulator is a linear regulator made by several manufacturers like Fairchild, or ST Microelectronics.

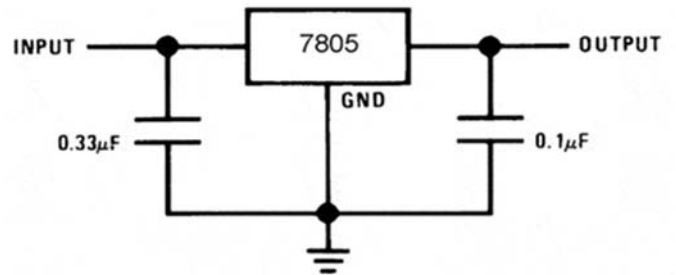


Fig 4: Voltage Regulator.

They can come in several types of packages. For output current up to 1A there may be two types of packages: TO-220 (vertical) and D-PAK (horizontal). With proper heat sink these LM78xx types can handle even more than 1A current. They also have Thermal overload protection, Short circuit protection.

2.3.2 Battery

The supply used is a rechargeable lead acid battery of 12V and 1.2A connected in series. The lead acid battery is powerful and bulky and proper care must be taken so that it does not get short circuited.

2.3.3 Infrared Sensor

It consists of just two components. The first is an Infra-Red (IR) transmitter (usually an LED), while the second is an Infra-Red receiver (usually a transistor). IR is transmitted out of the sensor unit. If the IR is reflected back, it is picked up by the IR receiver transistor. LM358n is a Low Power Dual Operational Amplifier.

2.3.4 L293D Driver

L293D is a dual H-Bridge motor driver with one IC we can interface two DC motors which can be controlled in both clockwise and counter clockwise direction and if you have motor with fix direction of motion, then you can make use of all the four I/Os to connect up to four DC motors. L293D has output current of 600mA and peak output current of 1.2A per channel.

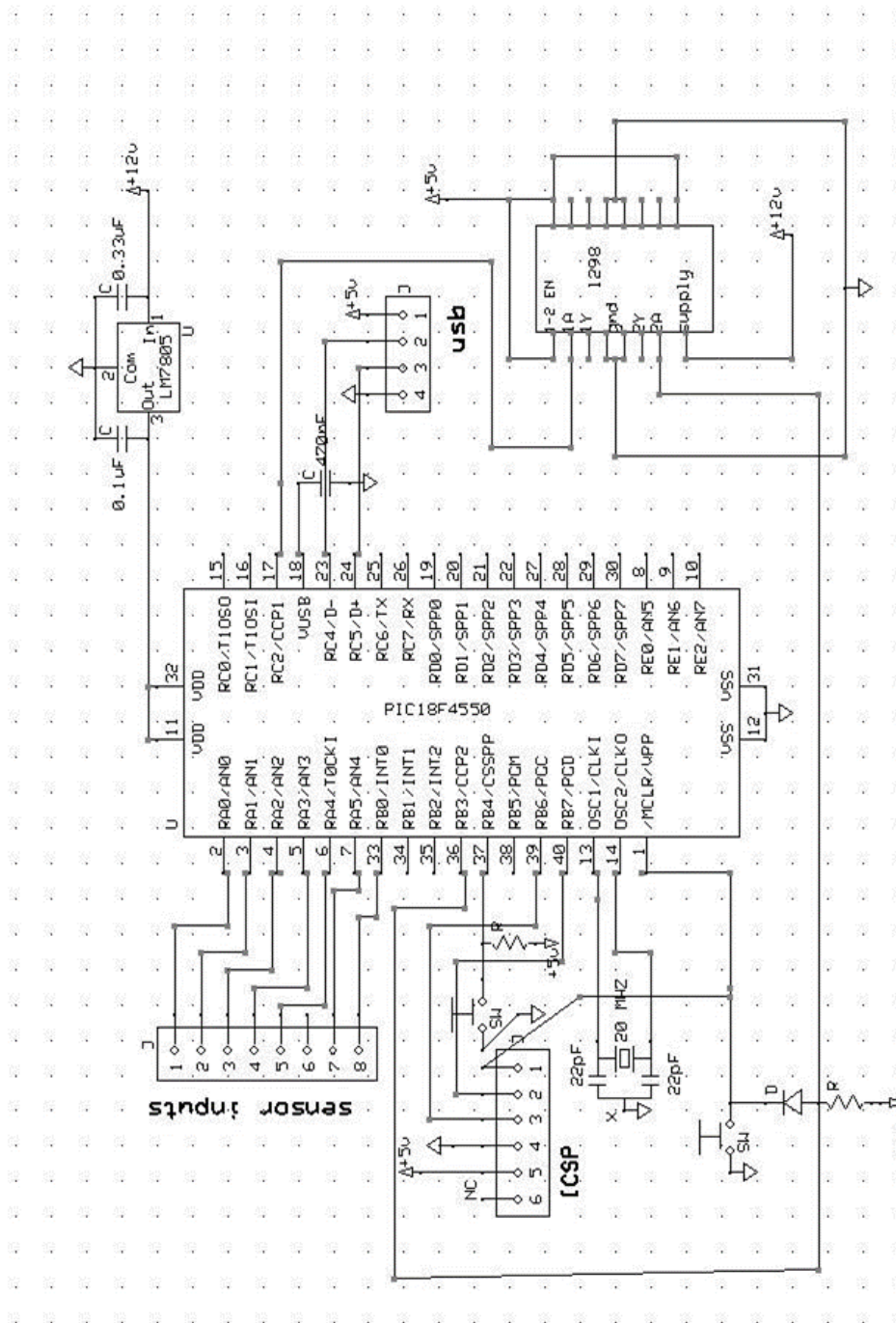


Fig 5: Circuit Diagram of Controller PIC 18f4550

3. Results and Discussion

3.1 Software

3.1.1 MP Lab IDE

MPLAB is an IDE that can be downloaded free from Microchip’s website. It contains all the software tools necessary to write a program in Assembler, assemble it, stimulate it and then download it to a programmer. The latter must be built or bought, or designed in to the target system. Further software tools can be bought and then integrated with MPLAB, both from Microchip and from other suppliers. This includes alternatives to what MPLAB already offers – e.g. Assemblers or simulators, as well as tools which offer much greater development power, like C compilers [1] or emulator drivers.

3.1.2 Elements of MPLAB

MPLAB is made up of a number of distinct elements, which work together to give the overall development environment.

These are

3.1.2.1 Text editor

This allows entry of the source code. It behaves to some extent like a simple text editor such as Notepad, but it can recognize the main elements of the programming language that is being used. Thus, in Assembler it colour-codes instructions in one colour, labels in another and comments in a third. In this way the programmer can immediately see if there is a misconception in his placing or use of text within the Assembler line.

3.1.2.2 Project manager

The preferred way of developing programs in MPLAB is by creating a project. An MPLAB project groups all the files together that relate to any one project, and ensures that they interact with each other in an appropriate way and are updated as needed.

3.1.2.3 Assembler and Linker

The function of the Assembler has already been discussed. So far we have assumed that there is a single source file. In advanced projects, however, the code may be created from a number of different files. The role of the Linker is to put these together, give each its correct location in memory, and ensure that branches and calls from one file to the other are correctly established.

3.1.2.4 Software simulator and debugger

A software simulator allows the program that has been developed to be tested, by running it on a simulated CPU in the host computer. Inputs can also be simulated, and outputs and memory values can be observed. The debugger contains the tools which allow program execution.

3.1.3 C18 compiler

The MPLAB C18 compiler is Microchip's own C compiler, written especially for the PIC 18 Series microcontroller. It follows the ANSI X3.159-1989 standard, except that it also contains a number of extensions, designed to optimise its use with the PIC microcontroller.

3.1.4 Creating Hex File

- 1) Install MPLab and c18 compiler
- 2) Go to project and open project wizard and select your pic 18 device
- 3) In the tool suite contents add the c18 compiler
- 4) Then create a file name for it and finish the process
- 5) Write the above code in the editor and save it in *.c format
- 6) Then go to view and open project
- 7) Right click on source files and add the c file you just saved
- 8) Right click on the linker files and add the corresponding PIC model linker file from the c18 folder (if you are using 18f4550 add 18f4550.lkr)
- 9) Then go to project-build options and click on project
- 10) For output directory direct it to where you want it to be saved
- 11) For include directory the path should direct to c18 folder 'h'
- 12) For library director 'lib' and linker 'lkr'
- 13) Apply the changes and then click on build all in the projects menu
- 14) The hex file is created in case there are no errors in the code

3.1.5 Sample code – to blink an LED

Connect an LED to pin PORTB7

```
#include<stdlib.h>
#include<p18f4550.h> //this header is added so that the code
to manipulate the 18f can be enabled
#include<delays.h>
void main(void)
{TRISBbits.TRISB7=0; //initialize PortB.7 as output}
While (1)
```

```
{PORTBbits.RB7 =! PORTBbits.RB7; //Toggle LE
Delay10KTCYx (200);}
```

TRIS is the direction command. It determines if the particular pin acts as input or output.

1 is for input and 0 is for output.

PORT command is used for latching the pin to read or write.

3.1.6 Grid algorithm

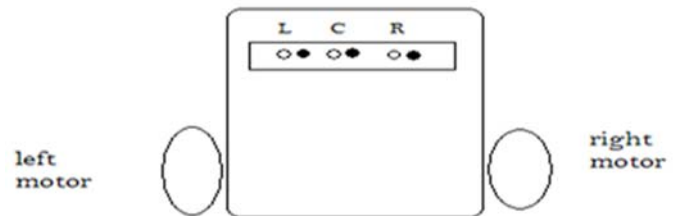


Fig 6: LCR Grid

The numbers of sensor used are 3 as shown in the figure, left centre and right.

Grid algorithm to move the bot from (0, 0) to (2, 3)

<Grid pic>

To move the bot to position (x, y)

1) The initial position of the bot is (0, 0) facing the positive x direction

2) Function of the important integers

- X1, Y1-the bots present location
- X2, Y2-the bots destination
- P- robot direction after the execution of the y part of the coordinate

If (p==1) positive y direction

If (p==0) negative y direction

- Q- robot direction after the execution of the x part of the coordinate

If (q==1) positive x direction

If (q==0) negative x direction

- N-the no. coordinates the robot has to pass through
- Z-the present coordinates number

3) If the robot has to pass through 3 coordinates then n = 3 etc.

4) Let the robot pass through (2, 3), then n = 1.

5) Initial coordinate (0, 0) =>x1=0, y1=0

6) Initialize the ports.

7) Fetch the destination coordinates through fetch coordinates function

8) x2 = 2,y2=3

9) The execution goes to gotox () function

m = 0;

m= x2 - x1=2-0=2

Since m >0 the bot has to move towards the positive x direction so q=1

Move bot gets executed

J is initialized to 0

Every time the bot comes across a junction j is incremented by 1.

So the crosses two junctions and stops.

Execution goes to gotoy ()

10) Execution of the y coordinate

m is initialized to 0.

m=y2-y1=3-0.

$m > 0$, so the bot is supposed to move in positive y direction; $p = 1$.
 Since the previous x position was towards right, the robot has to turn 90 left
 The execution now goes to movebot just as it was for x direction
 11) The robot has reached its destination (2, 3)

3.2 Wireless Transmission

Wireless communication is the transfer of information over a distance without the use of electrical conductors or "wires". The distances involved may be short (a few meters as in television remote control) or long (thousands or millions of kilometers for radio communications). When the context is clear, the term is often shortened to "wireless". XBee is one of the most efficient methods of wireless transmission of data.

3.2.1. XBee

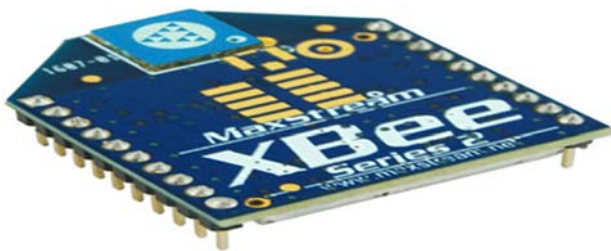


Fig 7: Pin of XBee-Pro OEM RF Modules.

The XBee and XBee-PRO OEM 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.

3.2.1.1 Applications:

- **Home Entertainment and Control** — Smart lighting, advanced temperature control, safety and security, movies and music
- **Home Awareness** — Water sensors, power sensors, smoke and fire detectors, smart appliances and access sensors
- **Mobile Services** — m-payment, m-monitoring and control, m-security and access control, m-healthcare and tele-assist

3.2.1.2 Modes of Operation

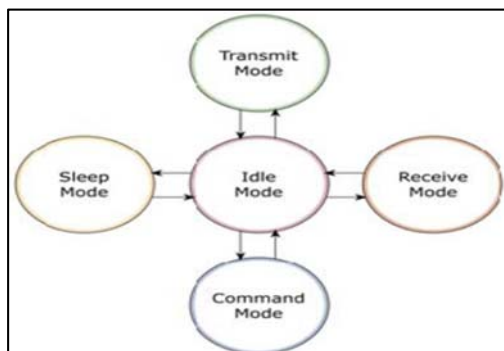


Fig 8: XBee/XBee-PRO RF Modules operate in five modes.

1. **Idle mode:** When not receiving or transmitting data, the RF module is in Idle Mode
2. **Transmit Mode:** Serial data is received in the DI Buffer
3. **Receive Mode:** Valid RF data is received through the antenna
4. **Sleep Mode:** Sleep Mode condition is met
5. **Command Mode:** Command Mode Sequence is issued

3.3 Sample USART program

```
#include <p18f4550.h>
#include <usart.h>
#include <delays.h>
void main(){
    TRISCbits.TRISC7 = 1;
    TRISCbits.TRISC6 = 1;
    while (1)
    {OpenUSART      (USART_TX_INT_OFF      &
    USART_RX_INT_OFF & USART_ASYNCH_MODE &
    USART_EIGHT_BIT  & USART_CONT_RX    &
    USART_BRGH_LOW, 77);
    puts USART ("VALUE");
    while (Busy USART());
    puts USART ("\n\r");
    while (Busy USART());
    Close USART ();
    Delay10KTCYx (100);
    break;}
    baud rate calculation
```

Asynchronous mode, high speed:

$$FOSC / (16 * (spbrg + 1))$$

Asynchronous mode, low speed:

$$FOSC / (64 * (spbrg + 1))$$

Where FOSC is the oscillator frequency

C18 consists of readymade funtions for USART or serial transmission. To setup serial transmission we have to open the usart by typing the function

OpenUSART (). to send a string we can use the function putsUSART ("value"), this function will send the string "value".

Thus, by specifying the destination coordinates and the path through which it has to travel, the robot has moved to that position autonomously.

4. Conclusion

Today we find most robots working for people in industries, factories, warehouses, and laboratories. Robots are useful in many ways. For instance, it boosts economy because businesses need to be efficient to keep up with the industry competition.

Therefore, having robots helps business owners to be competitive, because robots can do jobs better and faster than humans can, e.g. robot can built, assemble a car. Yet robots cannot perform every job; today robots roles include assisting research and industry. Finally, as the technology improves, there will be new ways to use robots which will bring new hopes and new potentials.

5. References

1. www.microchip.com
2. www.autopay.ws
3. www.myfirstbot.blogspot.com
4. www.ni.com
5. www.digi.com

6. www.eegeek.net
7. www.societyofrobots.com
8. 8.PIC Microcontrollers Principles and Applications by
Tim Willshurs