Controlling Robot using Long Distance Wireless Communication Technique

Uday Prakash Tiwari

Abstract:-

In this project we had controlled the robot to **1.2 km** distance. The Zigbee module used is S2C. We have router and coordinator module interaction in order to control the robot. The router module is used to send the signal and coordinator module is to receive the signal generated from router module. The signal is generated using UART serial communication. The microcontroller used for UART serial communication is PIC16F877A. **UART** communication is tested using **hyperterminal** software. The XCTU software is for router and coordinator interaction. We have controlled robot using **AT command**. Components used are two S2C modules, PIC16F877A development board, Zigbee adapter, Zigbee shield, Wheel mobile platform, UART adapter and Two 12V, 2A battery.

Interfacing UART with Hyperterminal:-

In this we can check the serial communication of PIC16F877A uC using universal serial bus with hyper terminal software.
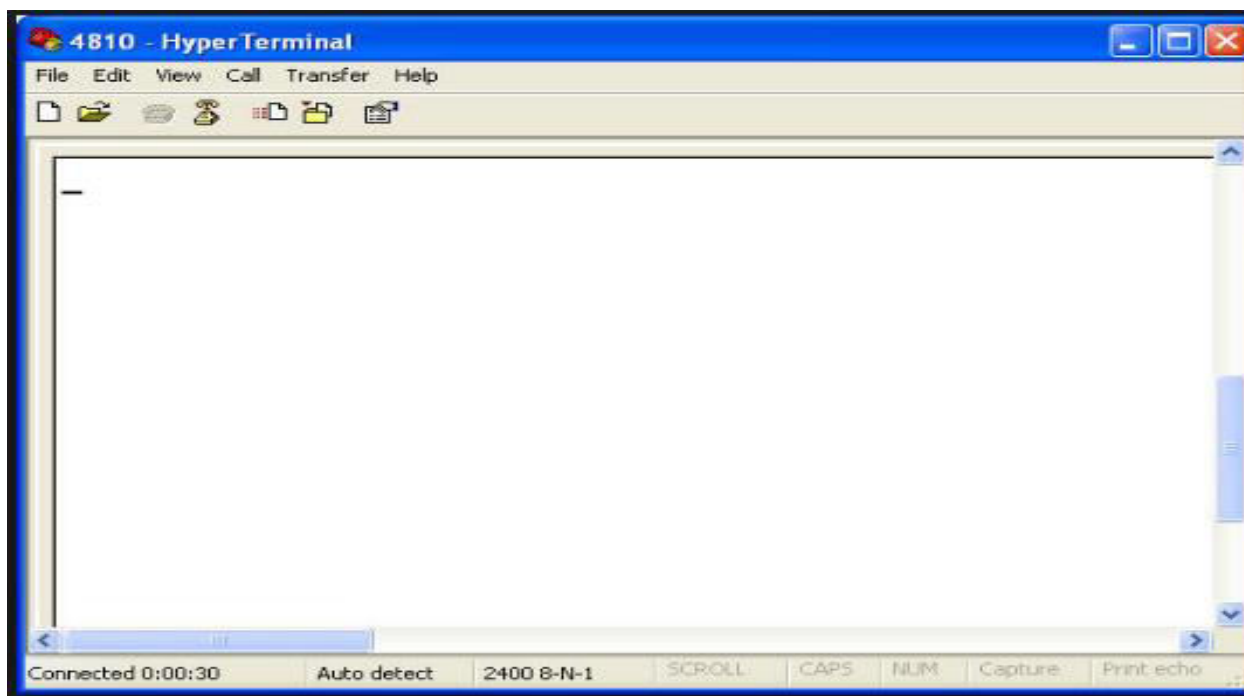

Proteus Simulation:-
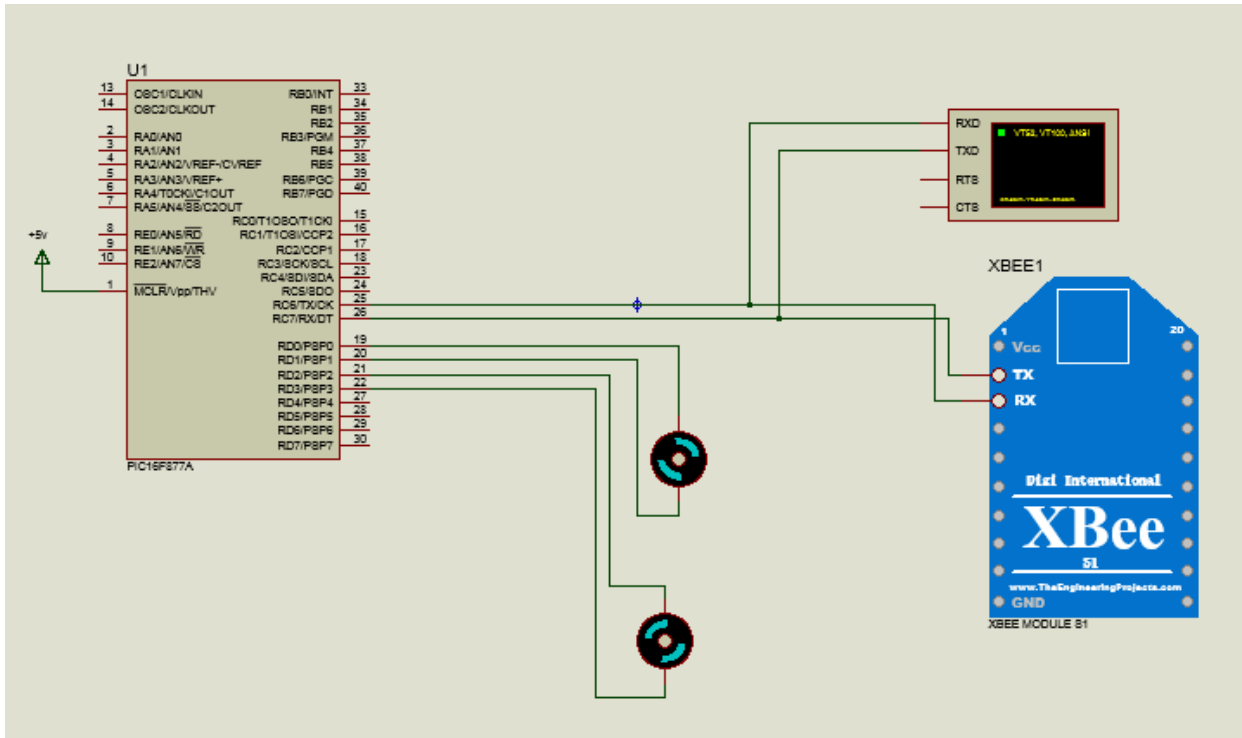
AT command using XCTU Software:-

Steps to be followed for giving AT command:-

(a) In the Consoles working mode of XCTU, click the **Open the serial connection with the radio module** button.

(b) Use **+++** to enter into command mode and wait for an OK response.

(c) To set a register, type an AT command followed by the value you want to set; for example, **ATID 2015**; followed by a Return.

(d) To read a register, type an AT command; for example, **ATID**; followed by a Return.

(e) Use the **ATWR** command to write the new configuration to the module's memory.

(f) Exit command mode with the **ATCN** command.
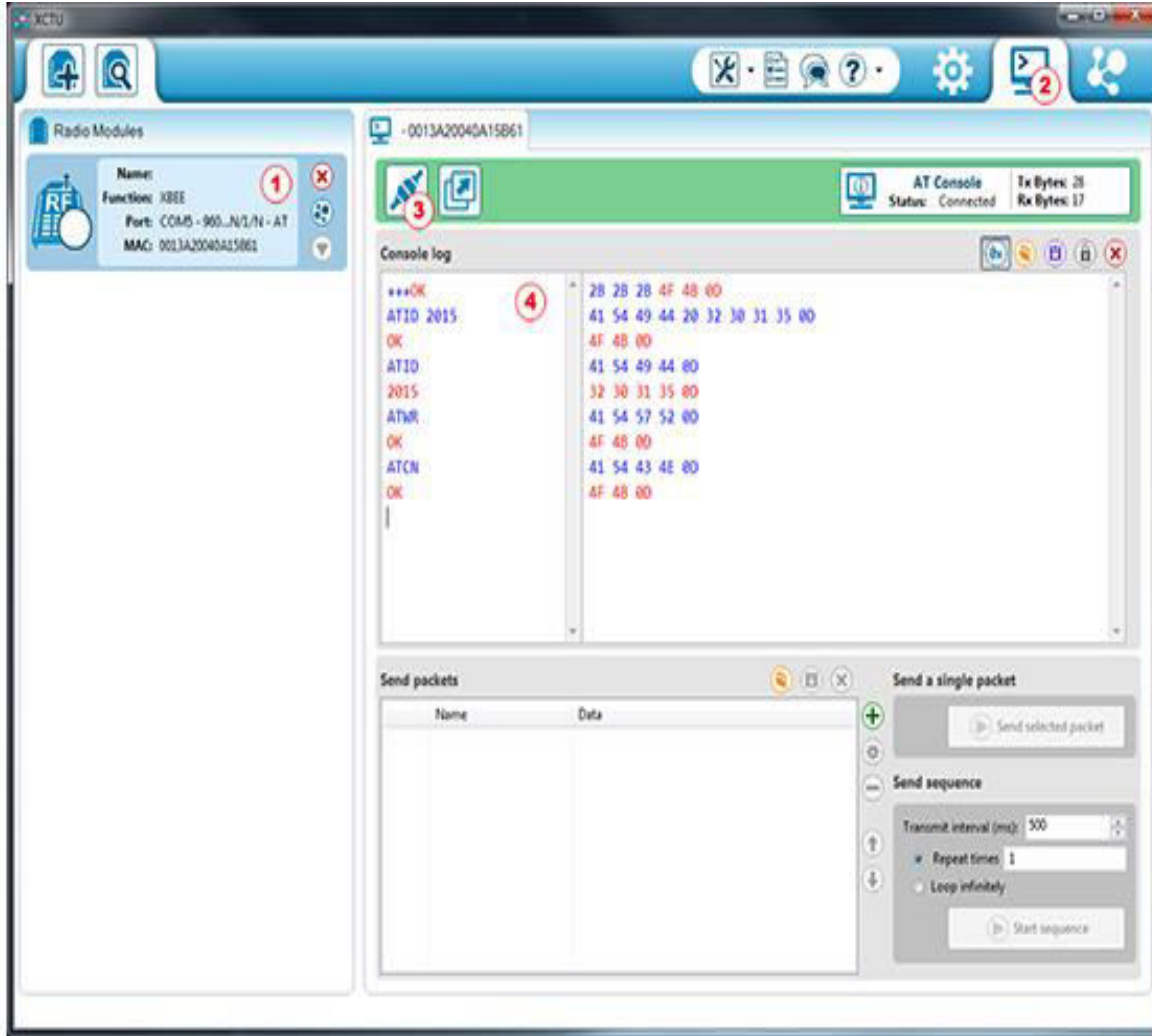
For controlling zigbee there is need for router and coordinator configuration. If the instruction is given by coordinator the router should perform the same action. This is based on master and slave condition.

Steps to be followed for router and coordinator configuration:-

(a) Selecting COM port for router and coordinator.
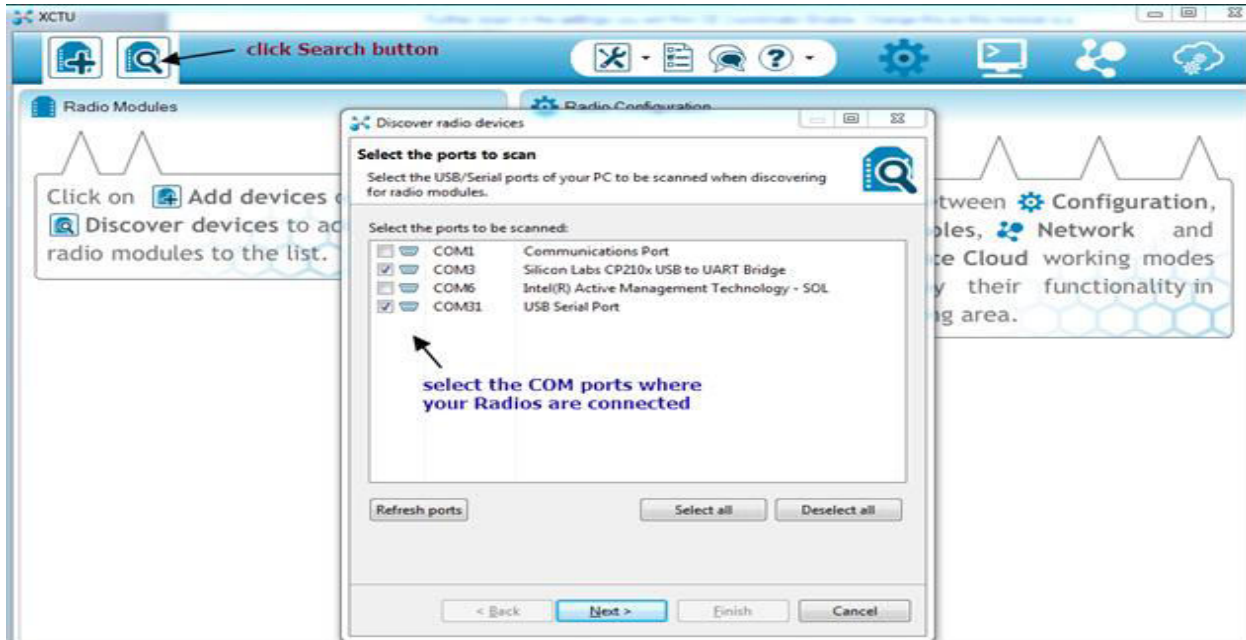
(b) Selecting port parameters.
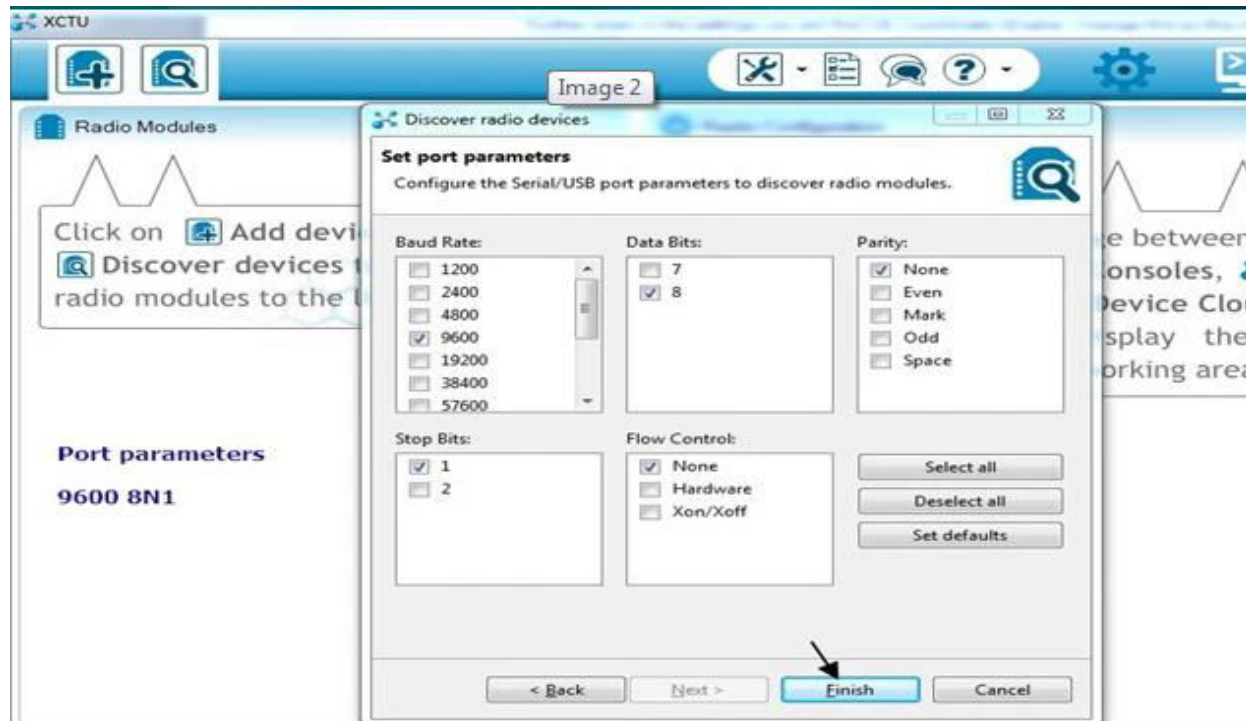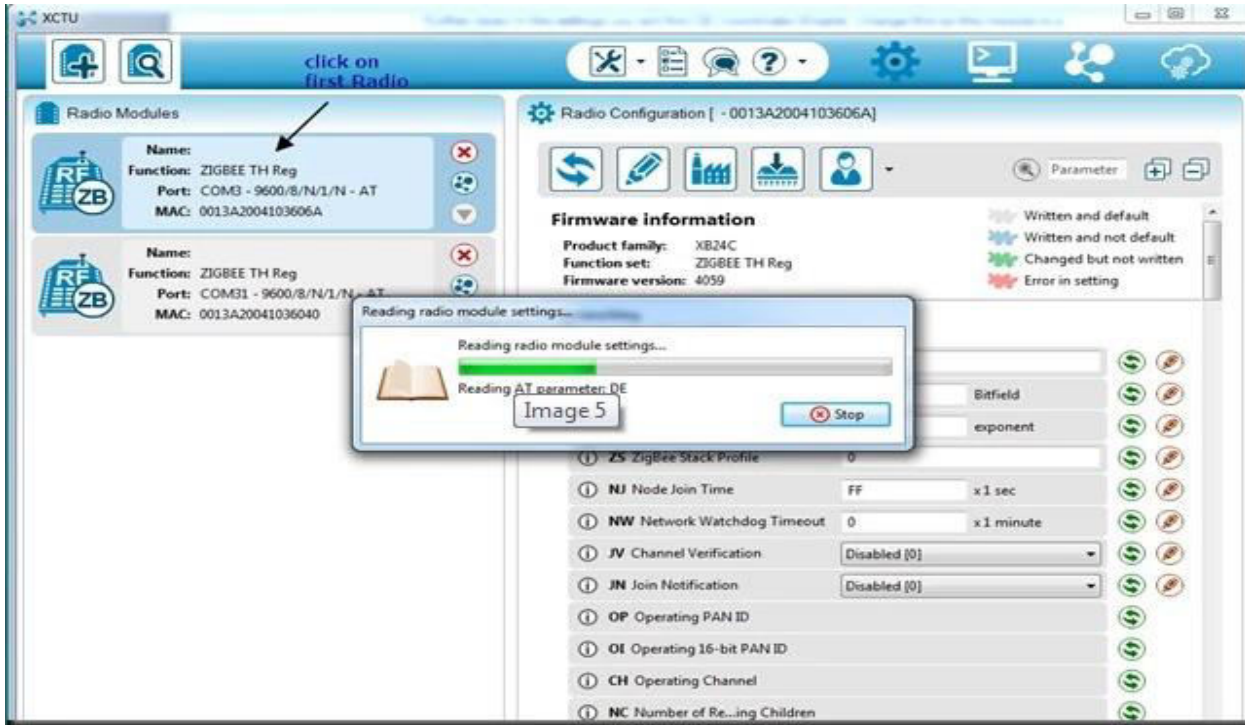
(c) **The XCTU Scans the USB ports selected and lists the RADIOs found with their unique 64 bit address.**
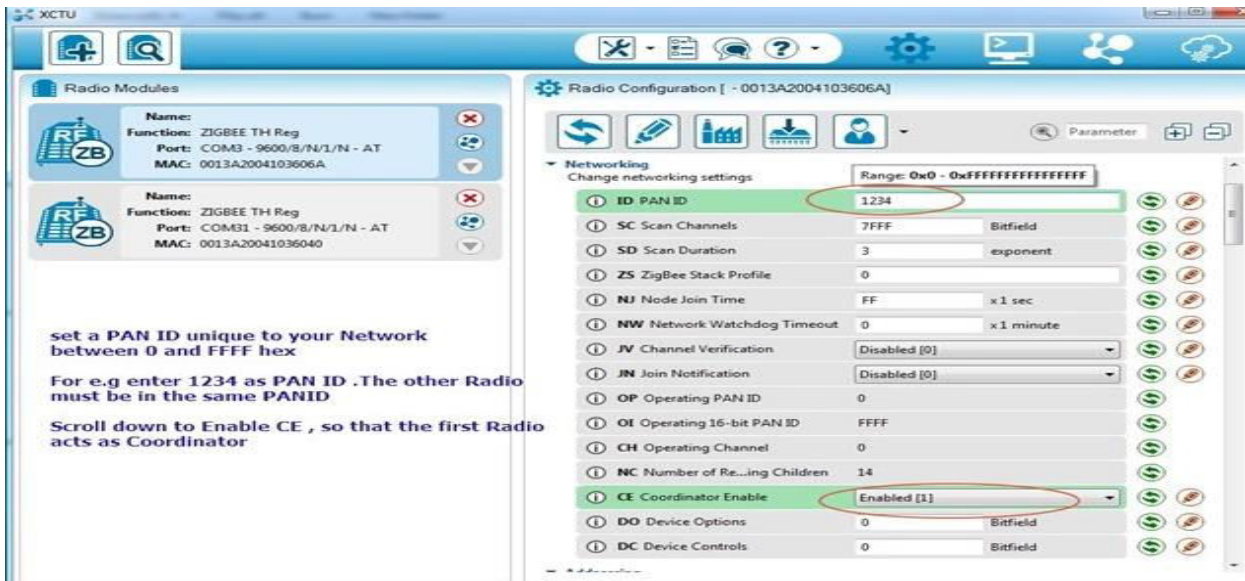

(d) **Now both the Radios appear on left plane.**
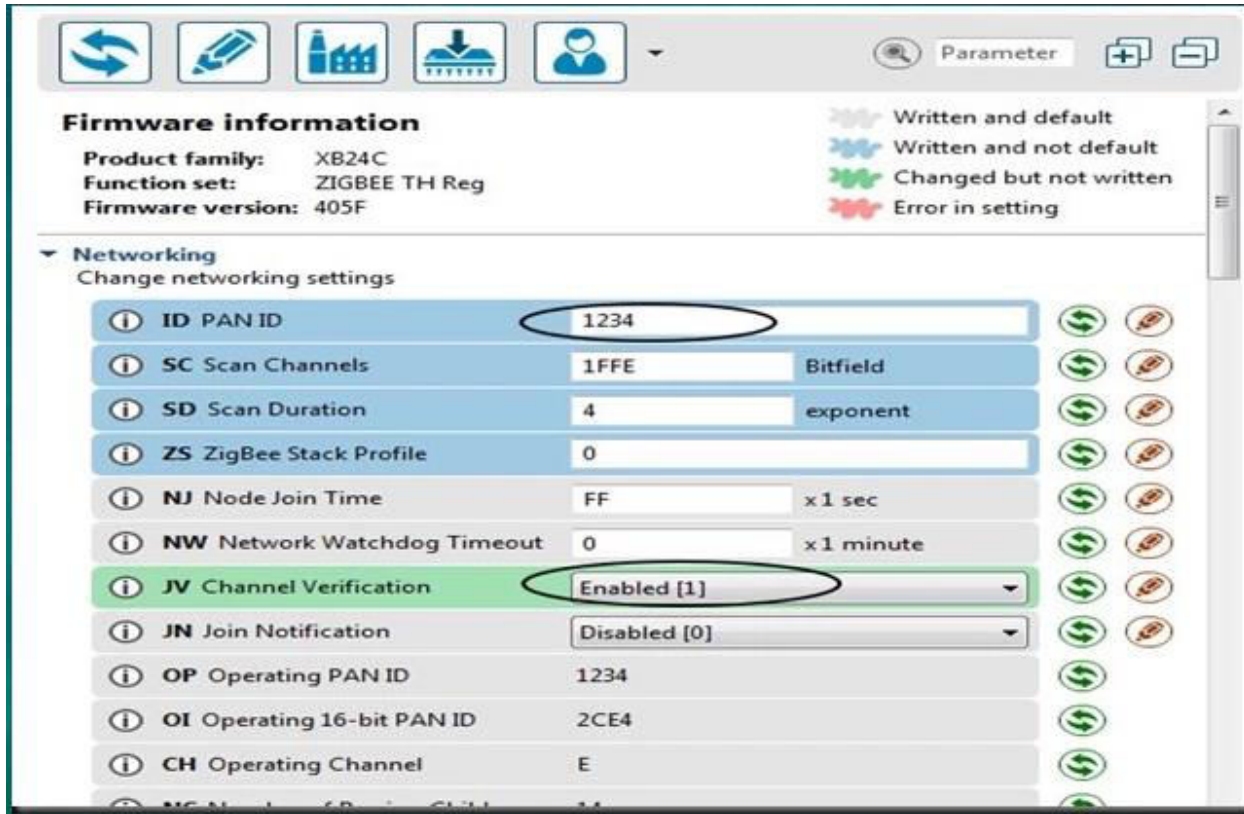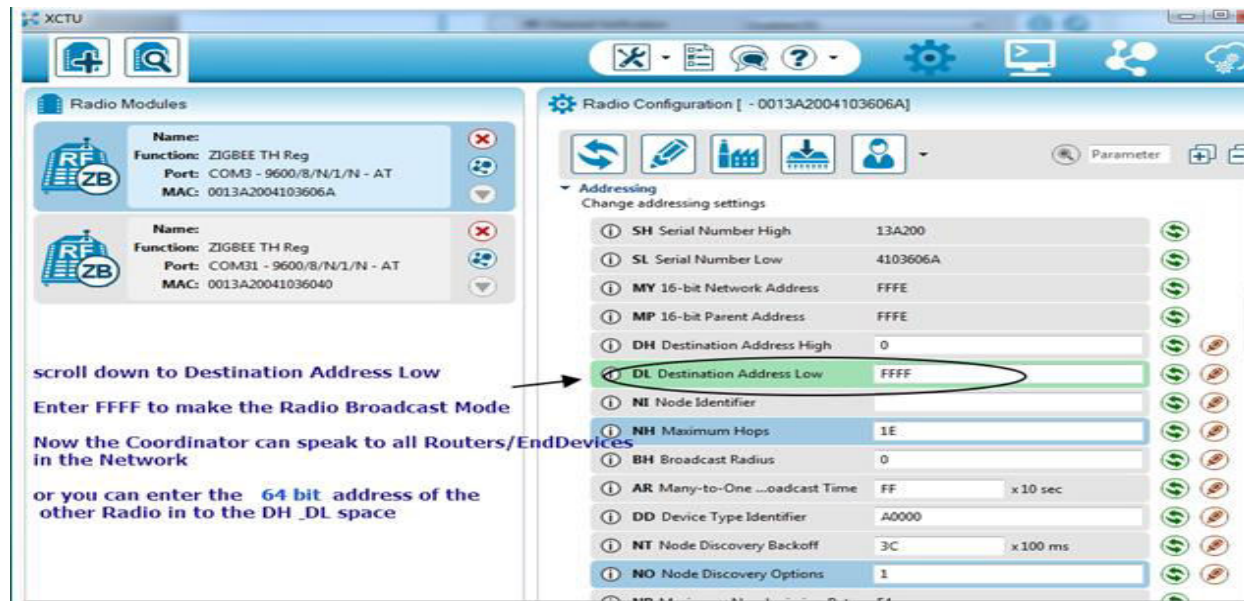
(e) Updating the firmware.

(f) Changing the coordinator configuration.

(g) Changing the router configuration.

(h) Changing the router and coordinator address.

This configuration helps coordinator to be connected with router. Number of routers can be connected to coordinator using this configuration. When both the zigbee led is blinking that means they are transmitting

the signal. But this doesn't means router and coordinator is connected. The Zigbee is always operated under **AT command mode**.

Robot code:-

```
 #include "htc.h"

//#include "stdio.h"

#ifndef _XTAL_FREQ

 // This definition is required to calibrate __delay_us() and __delay_ms ( )

#define _XTAL_FREQ 8000000

#endif

__CONFIG ( FOSC_HS & WDTE_OFF & PWRTE_OFF & CP_OFF & BOREN_ON & LVP_OFF &
CPD_OFF & WRT_OFF & DEBUG_OFF);


#include "uart1.h"

#define SBIT_TXEN    5

#define SBIT_SPEN    7

#define SBIT_CREN    4

void UART_Init(int baudRate)

{

   TRISC=0x80;          // Configure Rx pin as input and Tx as output

  TXSTA=(1<<SBIT_TXEN); // Asynchronous mode, 8-bit data & enable transmitter

   RCSTA=(1<<SBIT_SPEN)|(1<<SBIT_CREN);  // Enable Serial Port and 8-bit continuous receive

   SPBRG = (20000000UL/(long)(64UL*baudRate))-1;     // baud rate @20Mhz Clock

}

void UART_TxChar(char ch)

{

   while(TXIF==0);    // Wait till the transmitter register becomes empty
```

```
    TXIF=0;          // Clear transmitter flag

    TXREG=ch;          // load the char to be transmitted into transmit reg

}
char UART_RxChar ( )

{

   while (RCIF==0);   // Wait till the data is received

   RCIF=0;          // Clear receiver flag

   Return (RCREG);    // Return the received data to calling function

}
int main( )

{

   TRISD=0x00;


   char i,a[ ]={"a"};

   char ch;

   char x,y;

   init_comms ( );

   UART_Init (9600);          //Initialize the UART module with 9600   baud rate

 for (i=0;a[i]!=0;i++)

  {

     UART_TxChar (a[i]); // Transmit predefined string

  }

  while(1)

  {

  ch = UART_RxChar ( ); // Receive a char from serial port
```

```
    UART_TxChar (ch);

  if (ch=='1')

  {

  PORTD=0x00;    // Stop

  }

  if (ch=='2')

{

PORTD=0x09;     // Forward

}

if (ch=='3')

{

PORTD=0x05;      // Left

}

if (ch=='4')

{

PORTD=0x08;     // Right

}

   }

     }
```

Library file uart1.h:-

```
# ifndef _SERIAL_H_

# define _SERIAL_H_

# define BAUD 9600

# define FOSC 8000000L

# define NINE 0     /* Use 9bit communication? FALSE=8bit */
```

```
# define DIVIDER ((int)(FOSC/(16UL * BAUD) -1))

# define HIGH_SPEED 1

# if NINE == 1

# define NINE_BITS 0x40

# else

# define NINE_BITS 0

# endif

# if HIGH_SPEED == 1

# define SPEED 0x4

# else

# define SPEED 0

# endif

# if defined(_16F87) || defined(_16F88)

        #define RX_PIN TRISB2

        #define TX_PIN TRISB5

# else

        # define RX_PIN TRISC7

        # define TX_PIN TRISC6

# endif


/* Serial initialization */

# define init_comms ( )

        RX_PIN = 1;

        TX_PIN = 1;
```

SPBRG = DIVIDER;

RCSTA = (NINE_BITS|0x90);

TXSTA = (SPEED|NINE_BITS|0x20);

void putch (unsigned char);

unsigned char getch (void);

unsigned char getche (void);

#endif

Conclusion:-

Using zigbee S2C module we can control robot upto 1.2 Km distance. Through robot code we can control wheel movement of robot by pressing some characters or integer values from keyboard . With this technology any electronics components can be controlled like servo motor, stepper motor, dc motor etc at a given distance for a particular module. The various motions of robots are Forward, Reverse, Right, Left, Stop. There is wireless interaction between router and coordinator modules for movement of robot. The wireless interaction occurs at frequency (2.45 GHz). We can use Zigbee S2C Pro module for increasing control of robot upto 2 Km distance. The wireless interaction is based on UART serial communication. Thus, robot performs various motions successfully by pressing particular integers i.e. 1,2,3,4 from keyboard.