COMMUNICATION PROTOCOL AND REMOTE DESIGN

June 1, 2020

Experiment:To control the speed of the DC motor wirelessly by varying potentiometer at other end.

The main aim of the experiment is to know about the various communication protocols(serial communication and wireless communication) it works upon.

Solution:-

To control things wirelessly using Xbee module, You have to configure the Xbee modules using software called XCTU. Here, comes the part of Serial communication protocol.

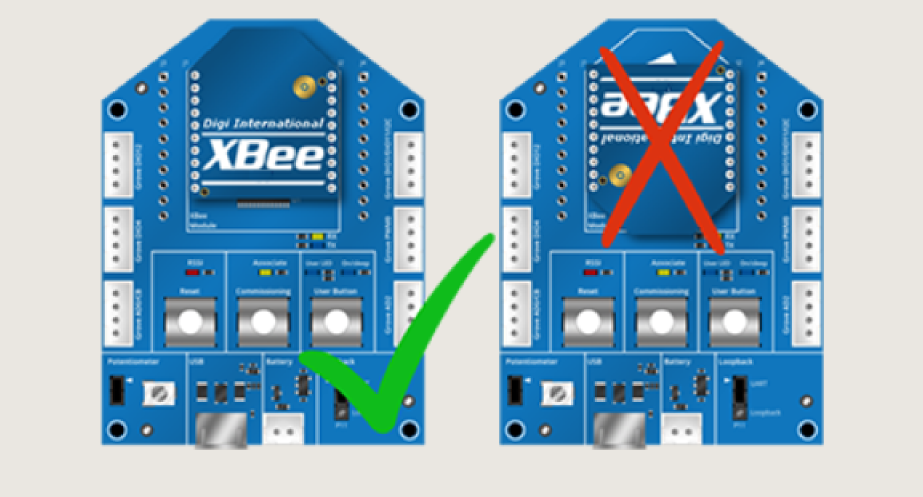
Serial communication consists of mainly three types:-

1. **SPI:-** Serial Peripheral Interface
   SPI (Serial Peripheral Interface) is a full duplex synchronous serial communication interface used for short distance communications at very high speed. SPI is a single-master communication protocol. This means that one central device initiates all the communications with the slaves. SPI protocol consists of four wires such as MISO, MOSI, CLK, SS used for master/slave communication.

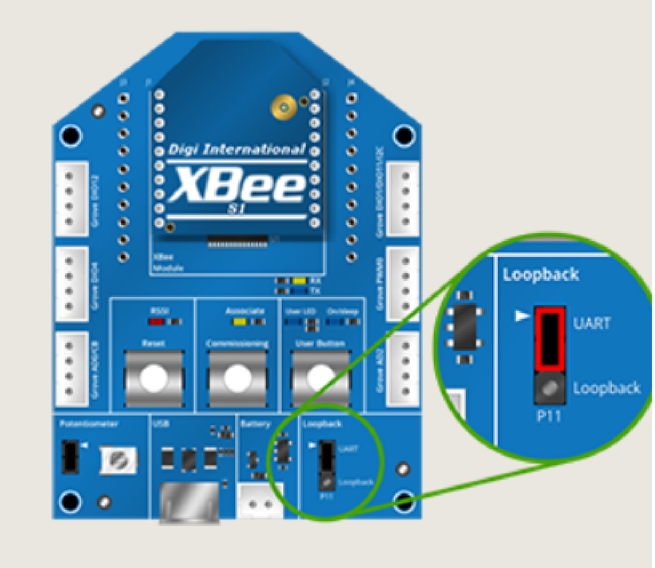
2. **I2C:-** Inter Integerated Circuit
   I2C is a serial communication protocol, so data is transferred bit by bit along a single wire (the SDA line). Like SPI, I2C is synchronous, so the output of bits is synchronized to the sampling of bits by a clock signal shared between the master and the slave. It can connect up to 128 devices to the main board while maintaining a clear communication pathway. I2C is a multi-master bus.I2C uses an address system and a shared bus = many different devices can be connected using the same wires and all data are transmitted on a single wire.

3. **UART:-** Universal Asynchronous Reception and Transmission
   It is a two wire data communication protocol with no clock signal, data format and transmission speed is configurable. UART supports bidrectional, asynchronous and serial data transmission. It has two data lines, one to transmit (TX) and another to receive (RX) which is used to communicate. TX and RX are connected between two devices. UART can also handle synchronization management issues between computers and external serial devices.

To configure Xbee module through XCTU software, we need Xbee development board because Xbee follows UART serial communication and we cannot directly attach our pc's/computers to Xbee module. So, we need a thing which communicates both the system and Xbee module, in our case it is Xbee development board which converts the signal into understandable signal for Xbee module. Mount Xbee on top of the development board as shown in figure...



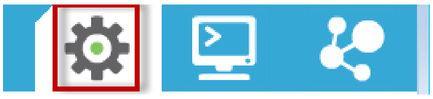
Ensure that the lookup point is in UART mode as shown in figure. . .



Connect the components

To get started, connect the components and start XCTU.

1. Plug the XBee modules into the XBee Grove Development Boards and connect them to your computer using the micro USB cable.
2. After connecting the modules to your computer, open XCTU.
3. Make sure you are in **Configuration working mode.**



Add the XBee modules to XCTU

Use XCTU to find your XBee modules and add them to the tool.

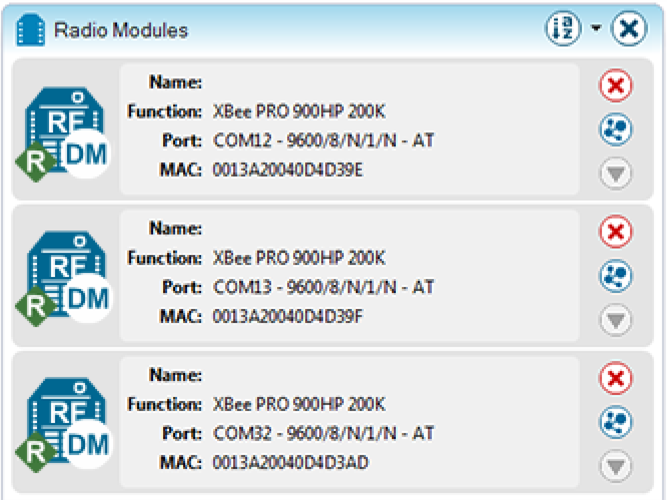
1. Click **Discover radio modules** from the toolbar.

2. In the **Discover radio modules** dialog, select the serial port(s) in which you want to look for radio modules. If you do not know the serial ports where your modules are attached, select all ports. Click Next.

3. In the **Set port parameters** window, maintain the default values and click Finish.

4. As XCTU locates radio modules, they appear in the **Discovering radio modules...** dialog box. Once the discovery process has finished, click **Add selected devices**.

5. At this point, assuming you have two modules connected to your computer, you should see something like this in the **Radio Modules** section on the left:



Name:
Function: XBee PRO 900HP 200K
Port: COM12 - 9600/8/N/1/N - AT
MAC: 0013A20040D4D39E

Name:
Function: XBee PRO 900HP 200K
Port: COM13 - 9600/8/N/1/N - AT
MAC: 0013A20040D4D39F

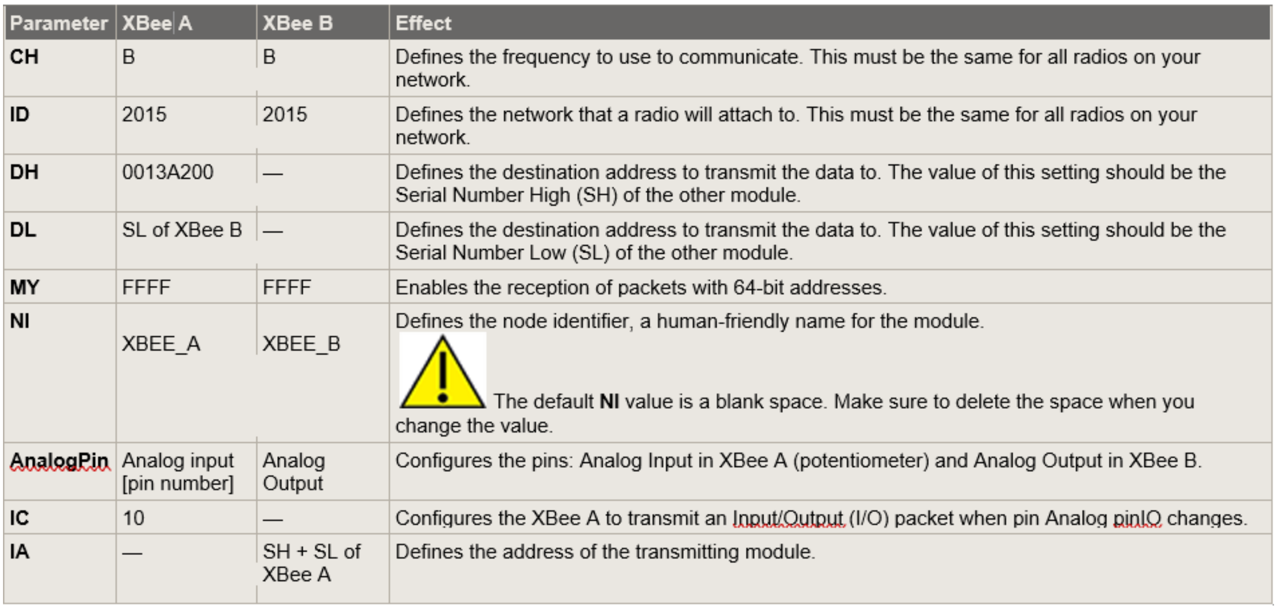
Name:
Function: XBee PRO 900HP 200K
Port: COM32 - 9600/8/N/1/N - AT
MAC: 0013A20040D4D3AD

Configure the XBee modules

In order to transmit data wirelessly between your XBees, set the DH and DL parameters on each module to match the SH and SL (SH + SL = MAC address) of the other module. For example, if the MAC of XBee A is 0013A20012345678, then the DH of XBee B should be 0013A200 and the DL 12345678.

1. Restore the default settings of all XBee modules with the **Load default firmware settings** button at the top of the Radio Configuration section.

2. Use XCTU to configure the following parameters:



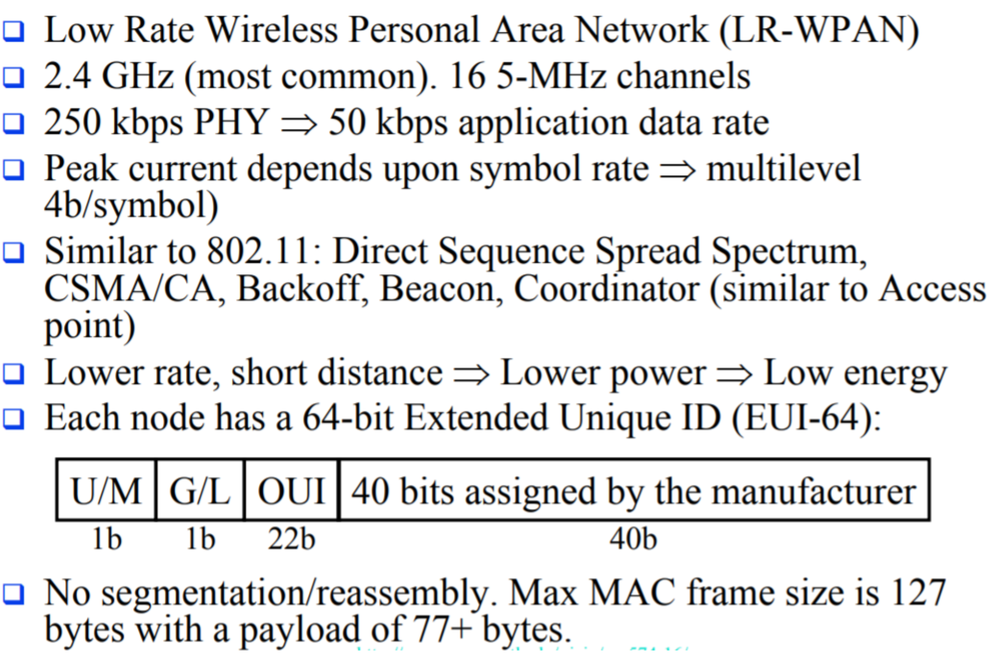
Parameter	XBee A	XBee B	Effect
CH	B	B	Defines the frequency to use to communicate. This must be the same for all radios on your network.
ID	2015	2015	Defines the network that a radio will attach to. This must be the same for all radios on your network.
DH	0013A200	—	Defines the destination address to transmit the data to. The value of this setting should be the Serial Number High (SH) of the other module.
DL	SL of XBee B	—	Defines the destination address to transmit the data to. The value of this setting should be the Serial Number Low (SL) of the other module.
MY	FFFF	FFFF	Enables the reception of packets with 64-bit addresses.
NI	XBEE_A	XBEE_B	Defines the node identifier, a human-friendly name for the module. ⚠ The default **NI** value is a blank space. Make sure to delete the space when you change the value.
AnalogPin	Analog input [pin number]	Analog Output	Configures the pins: Analog Input in XBee A (potentiometer) and Analog Output in XBee B.
IC	10	—	Configures the XBee A to transmit an Input/Output (I/O) packet when pin Analog pinIO changes.
IA	—	SH + SL of XBee A	Defines the address of the transmitting module.

Note:The dash (—) in the table means to keep the default value. Do not change the default value. CH and ID for both the xbees should be same.

3. Write the settings of all XBee modules with the Write radio settings button at the top of the Radio Configuration section.

4. Now, connect the output of the potentiometer at the Analog pin of the Xbee module which act as a transmitter. If we power up both the Xbee's the Transmission and reception process begins through wireless communication. Here, comes two point i.e

   (a) Wireless communication Protocol
   For wireless communication, there are many standards which one should while developing and generating signal. For Xbee wireless communication it follows IEEE 802.15.4 some section under this protocol given below.



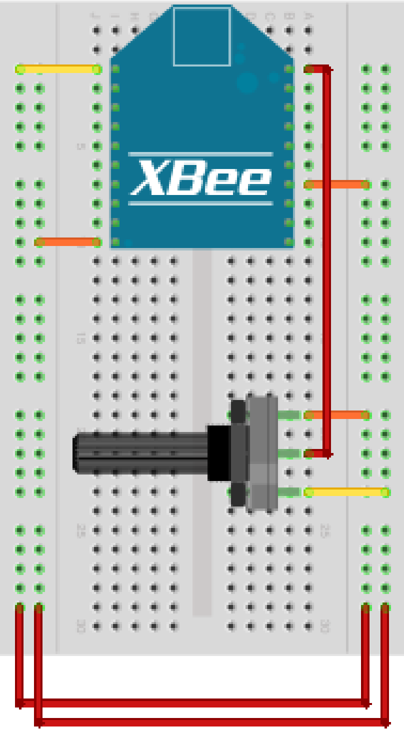
- ❑ Low Rate Wireless Personal Area Network (LR-WPAN)
- ❑ 2.4 GHz (most common). 16 5-MHz channels
- ❑ 250 kbps PHY ⇒ 50 kbps application data rate
- ❑ Peak current depends upon symbol rate ⇒ multilevel 4b/symbol)
- ❑ Similar to 802.11: Direct Sequence Spread Spectrum, CSMA/CA, Backoff, Beacon, Coordinator (similar to Access point)
- ❑ Lower rate, short distance ⇒ Lower power ⇒ Low energy
- ❑ Each node has a 64-bit Extended Unique ID (EUI-64):

U/M	G/L	OUI	40 bits assigned by the manufacturer
1b	1b	22b	40b

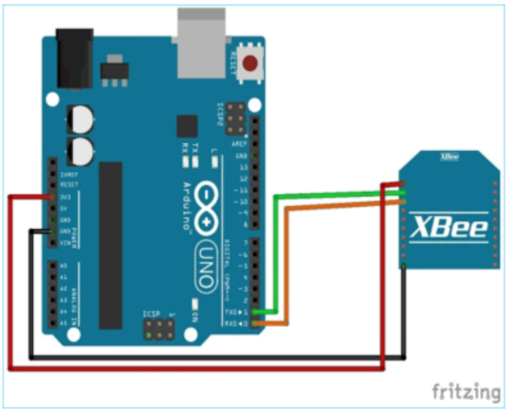
- ❑ No segmentation/reassembly. Max MAC frame size is 127 bytes with a payload of 77+ bytes.

(b) Receiver end receives data but it cannot process it and react to the data accordingly. For that we have to connect an intelligent device like arduino to get data from it using UART serial communication and map the data to generate PWM signal accordingly to control the speed of the DC motor by giving signal to motor driver.

Connection at transmitter end:-



Connection at reciever end:-



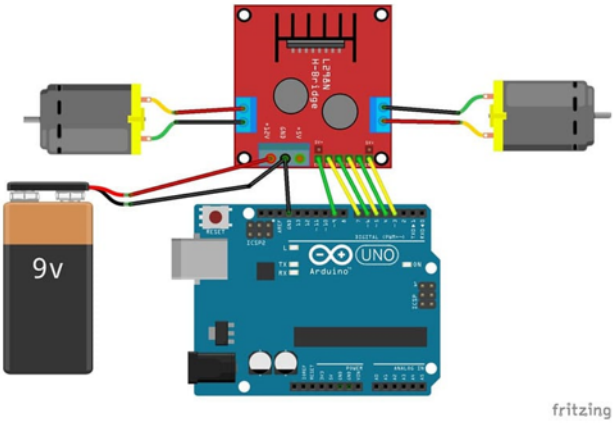
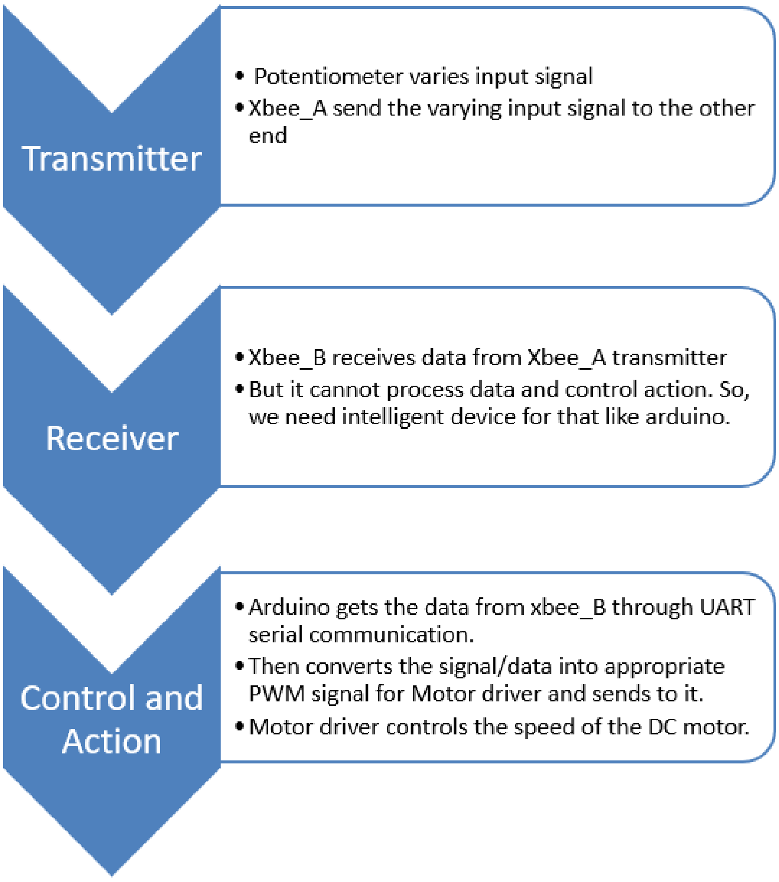


Image reference:-

Robu.in (image 2)Arduino Xbee Tutorial: XBee Module Interfacing with Arduino(image 1)

Block Diagram :-



Transmitter

- Potentiometer varies input signal
- Xbee_A send the varying input signal to the other end

Receiver

- Xbee_B receives data from Xbee_A transmitter
- But it cannot process data and control action. So, we need intelligent device for that like arduino.

Control and Action

- Arduino gets the data from xbee_B through UART serial communication.
- Then converts the signal/data into appropriate PWM signal for Motor driver and sends to it.
- Motor driver controls the speed of the DC motor.

Reference:- https://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#concepts/c_wc_kit.htm%3FTocPath%3D_____1

Remote design

Now, We know how to control things wirelessly. So, This time we have to design remote for the cycle bot keeping in mind about the remote optimization factor which plays a critical role in the smooth functioning of the things.

This depend upon the two factors i.e

1. Parameters need control
2. And cost(should be minimum).

In case of cycle bot we need to two factors i.e

1. Speed of the back wheel motor
2. And Controlling front servo motor for left right motion of the cycle bot.

For controlling this two parameters we need two potentiometer which is in built in one joy-stick. And one Xbee module for transmitting data, battery for power supply and toggle switch to on/off.(resistor and capacitors for enhancing the performance)

Components required:

1. Xbee module
2. One joystick
3. Power supply(battery)
4. Resistor,capacitor,perfboar and wires.

In this case, we use two joystick to control two parameters because generally we controls the remote manually by grab it using both hands and it is easy to control the parameter at the same time compared to the one joystick to control two parameter it needs practice for that. But the cost factor will increase a bit in this case. But the results in this case will be better.