Experiment No. – 7

Title: Interfacing Arduino to Bluetooth Module

Aim: Interfacing of Bluetooth Module HC 05 with Arduino

Hardware Requirements: Arduino Board **,** Bluetooth Module HC 05

Software Requirements: Arduino IDE

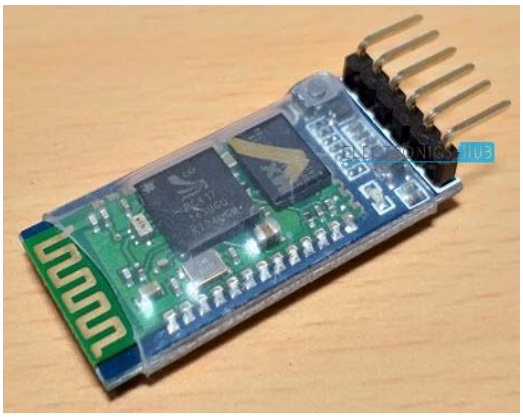
Theory:

Bluetooth Communication is a 2.4GHz frequency based RF Communication with a range of approximately 10 meters. It is one of the most popular and most frequently used low range communication for data transfer, audio systems, handsfree, computer peripherals etc.

HC-05 Bluetooth Module

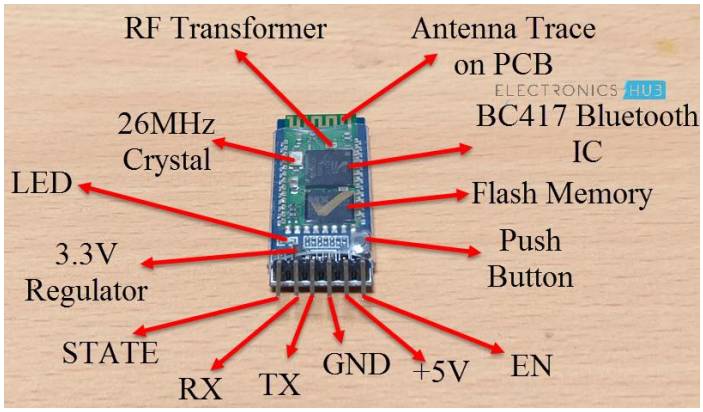
HC-05 Bluetooth Module is a simple Wireless Communication device based on the Bluetooth Protocol. This module is based on BC417 Single Chip Bluetooth IC that is compliant with Bluetooth v2.0 standard and with support for both UART and USB interfaces.

Generally, the HC-05 Bluetooth Module, or the HC-05 Sub Module, to be precise, comes with the BC417 IC along with a flash memory. Such Modules come as surface mount board and several third-party manufacturers use these board to build a more complete system with necessary pins and components.



HC-05 Bluetooth Module

Pins Diagram:



An important point to remember is the HC-05 Bluetooth Module works on a logic level of 3.3V. Hence, a 3.3V Regulator is used on the board.

Pin Description

EN: It is the enable pin. When this pin is floating or connected to 3.3V, the module is enabled. If this pin is connected to GND, the module is disabled.

+5V: This is the supply pin for connecting +5V. As the Module has on-board 3.3V regulator,

GND: It is the ground pin.

TX: It is the Transmitter pin of the UART Communication.

RX: It is the Receive Pin of UART.

STATE: This is a status indicator pin. This pin goes LOW when the module is not connected to any device. When the module is paired with any device, this pin goes HIGH.

Modes of Operation

The HC-05 Bluetooth Module can be configured in two modes of operation: **Command Mode and Data Mode.**

In Command Mode, you can communicate with the Bluetooth module through AT Commands for configuring various settings and parameters of the Module like get the firmware information, change UART Baud Rate, change module name, set it as either Master or slave etc.

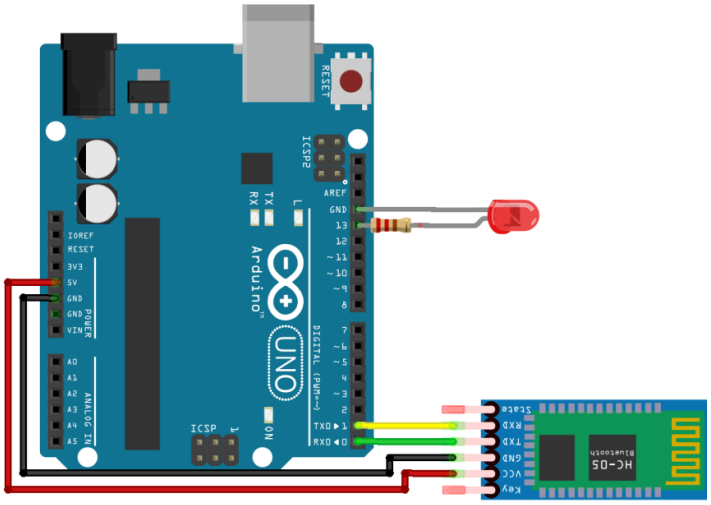
An important point about HC-05 Module is that it can be configured as Master or Slave in a communication pair. In order to select either of the modes, you need to activate the Command Mode and sent appropriate AT Commands.

Coming to the Data Mode, in this mode, the module is used for communicating with other Bluetooth device i.e. data transfer happens in this mode

Default Settings of HC-05 Bluetooth Module

The following is a list of few of the default settings of the HC-05 Bluetooth Module.
Name: HC-05
Password: 1234 (or 0000)
Type: Slave
Mode: Data
Baud Rate: 9600 with 8 data bits, no parity and 1 stop bit

The aim of this circuit is to connect the Bluetooth Module with Arduino, Pair the Bluetooth Module with an Android Phone, send data from Android Phone to the Bluetooth Module using a simple App, read the data from Bluetooth Module through Arduino and finally, display the data and control a device based on the data.



Description: First we need to define the pin to which our LED will be connected and a variable in which we will store the data coming from the smartphone. In the setup section we need to define the LED pin as output and set it low right away. As mention previously, we will use the serial communication so we need to begin the serial communication at 38400 baud rate, which is the default baud rate of the Bluetooth module. In the loop section with the Serial.available() function we will check whether there is available data in the serial port to be read. This means that when we will send data to the Bluetooth module this statement will be true so then using the Serial.read() function we will read that data and put it into the "state" variable. So if the Arduino receive the character '0' it will turn the LED off and using the Serial.println() function it will send back to the smartphone, via the serial port, the String "LED: OFF". Additionally we will reset the "state" variable to 0 so that the two above lines will be executed only once. Note here that the "state" variable is integer, so when we receive the character '0' that comes from smartphone, the actual value of the integer "state" variable is 48, which corresponds to character '0', according to the ASCII table.. That's why in the "if" statement we are comparing the "state" variable to a character '0'. On the other hand, if the received character is '1', the LED will light up and the String "LED: ON" will be sent back.

App for Bluetooth Communication

MIT App Inventor, Bluetooth controller, Arduino Bluetooth controller etc

Connecting the Smartphone to the HC-05 Bluetooth Module and the Arduino :

Connect the smartphone to the Bluetooth module and the Arduino. What we need to do here is to activate the Bluetooth and the smartphone will find the HC-05 Bluetooth module.

Procedure:

Step 1: Connect the Arduino board to the Micro-IoT Sensor board using the FRC cable provided with the board.

Step 2: Connect the Power supply adaptor and power on the circuit.

Step 3: Open Arduino IDE and create a new sketch (program) using the above pins.

Step 4: In the Arduino IDE go to tools→Port and select the appropriate COM port.

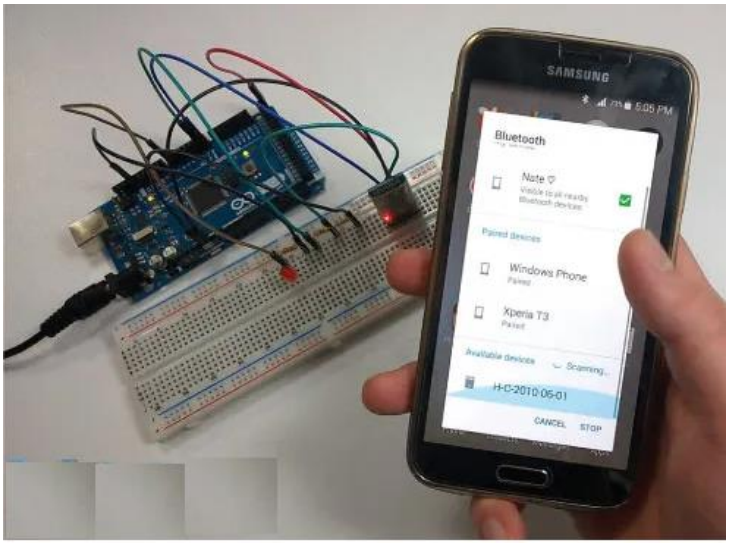
Step 5: In the Arduino IDE click on the upload button ( ) to compile and download the code into the Arduino UNO. When successfully downloaded the code will start running.

Step 6: This experiment uses the HC-05 Bluetooth module. To connect the module to the circuit slide the switch SW6 between 2-3 (BT_ON) . Now reset the Arduino board. Open the serial monitor @9600 baud rate and observe the output.

Step 7: On your Android phone install the Bluetooth Terminal application. Scan for Bluetooth devices from your android phone. Pair with the HC-05 (passkey is 1234). Once paired open the Bluetooth Terminal application and select the HC05 module. Send ascii data . Also if you send '0' or '1' the LEDs on the board can be switched On or OFF.

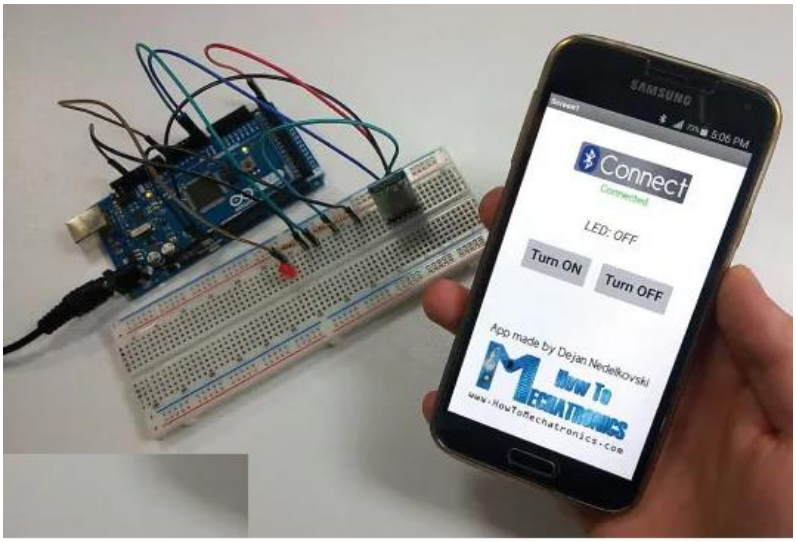
Observation:

Whatever Ascii data was send from the Bluetooth Terminal Application is received on the Arduino and displayed on the Serial Monitor. The LED's are also controlled through the data.



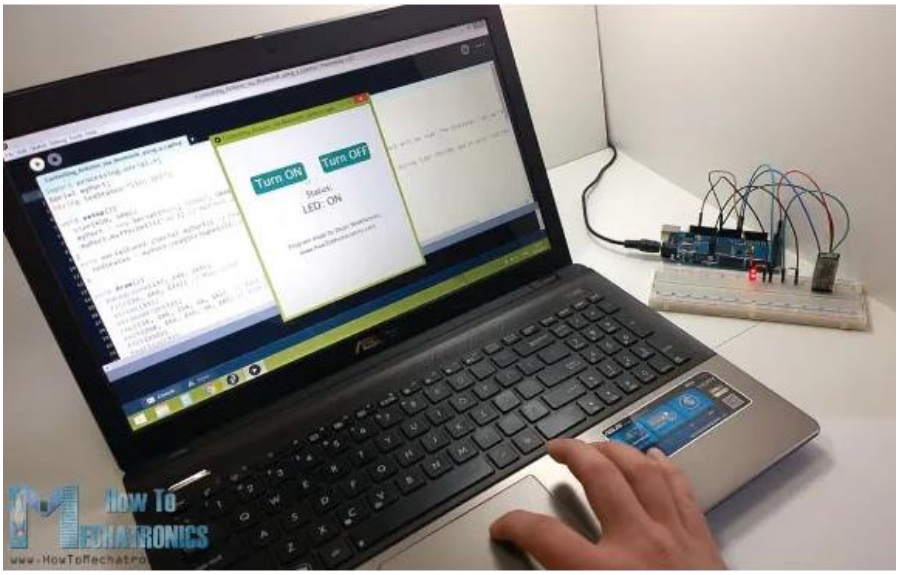
Then we need to pair the devices and the default password of the HC-05 module is 1234. After we have paired

the devices we need an application for controlling the Arduino.



With the connect button we will connect the smartphone to the Bluetooth module and the status text below the button will tell us whether we have successfully connected. Using the "Turn ON" and "Turn OFF" buttons we can turn on and off the LED. The text above the buttons is the one that the Arduino is sending back to the smartphone when a particular button is pressed.

Similarly, we can control the Arduino via Bluetooth using a Laptop or a PC.



Conclusion:

Code:

```
#include<SoftwareSerial.h>

#define LED1 0
#define LED2 1
#define LED3 2
#define LED4 4

/* Create object named bt of the class SoftwareSerial */
SoftwareSerial bt(8,7); /* (Rx,Tx) */

void setup() {
  bt.begin(9600); /* Define baud rate for software serial communication */
  Serial.begin(9600); /* Define baud rate for serial communication */
  Serial.println("Bluetooth ");
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
}

void loop() {
 char input;
 if (bt.available()) /* If data is available on serial port */
 {
   input = bt.read();
   Serial.write(input); /* Print character received on to the serial monitor */

  if(input == '1')
  {
  Serial.println("Device ON");
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, HIGH);
  digitalWrite(LED3, HIGH);
  digitalWrite(LED4, HIGH);
  }
  if(input == '0')
  {
  Serial.println("Device OFF");
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
  digitalWrite(LED4, LOW);
  }
  }
}
```