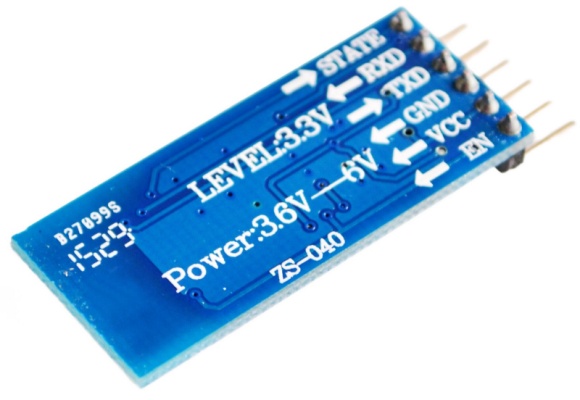
Bluetooth module HC09

**A communication interface is necessary for the robot to convey location information. We are using Bluetooth module hc-09 (having Bluetooth 4.1 technology).**



Feature:  
1, serial module operating voltage 3.3V.  
2, the baud rate can be set 2400,4800,9600,19200,38400,57600,115200,230400  
3 core module size: 28mm x 15 mm x 2.35mm.

The module has 6 pins

|  |  |
| --- | --- |
| PIN | FUNCTION |
| State | Gives status about connection |
| RXD | Reveive pin |
| TXD | Transmit pin |
| GND | Ground |
| VCC | +3.3 v input |
| EN | Enable pin |

A mobile phone will communicate to bot controller (atmega 328p) through Bluetooth module. Phone will send data over Bluetooth to the module. And module will send data serially to the controller.

Testing

* For testing purpose , we interfaced Bluetooth module to the arduino UNO board. And wrote a program to display the received data on computer screen
* We selected alphabets A to G as valid data and all other characters, signs, spaces & special characters as invalid data.
* We have pin0 and pin1 of UNO board as RX and TX respectively, that we are using for sending the received data on computer.
* So we used pins 8 and 9 to interface Bluetooth module. A function called SoftwareSerial
* Configures the normal GPIO pins as TX and RX for serial communication.
* Data transmitted through Mobile phone via app “serial Bluetooth terminal “ .
* Arduino has inbuild LED connected on pin 13. We used that LED to indicate that valid data is received.

Connection chart

|  |  |
| --- | --- |
| HC-09 | Arduino UNO |
| State | Nc |
| RXD | D9 |
| TXD | D8 |
| GND | Gnd |
| VCC | +3.3v |
| EN | +5v |

Program

#include <SoftwareSerial.h>

#define LED\_PIN 13

SoftwareSerial mySerial(8, 9); // RX, TX for arduino

// Connect HM10 Arduino Uno

// Pin 1/TXD Pin 8

// Pin 2/RXD Pin 9

void setup()

{ Serial.begin(9600); //9600 baudrate

pinMode(13, OUTPUT);

mySerial.begin(9600); //bluetooth module supports 9600 baudrate

mySerial.println("welcome "); //send message welcome to phone

}

void loop()

{char c;

if (mySerial.available()) //if there is data on the line

{

c = mySerial.read(); //copy data in variable c

if (c != '0' && c != '\n' && c != '\r' && c != ' ')

//to ignore non-character inputs

{ delay(1);

Serial.println("char received");

}

/\* Here onwards compare the received data to valid inputs (A-G) and transmit the same on serial monitor\*/

if (c == 'a' || c == 'A')

{

Serial.println("A received"); led();

}

else if (c == 'b' || c == 'B')

{

Serial.println("B received"); led();

}

else if (c == 'c' || c == 'C')

{

Serial.println("C received"); led();

}

else if (c == 'd' || c == 'D')

{

Serial.println("D received"); led();

}

else if (c == 'E' || c == 'e')

{

Serial.println("E received"); led();

}

else if (c == 'F' || c == 'f')

{

Serial.println("f received"); led();

}

else if (c == 'g' || c == 'G')

{

Serial.println("G received"); led();

}

/\* now if received data is invalid, then send message “invalid input !” to serial terminal and mobile phone. \*/

if ((c >= 14 && c <= 64) || (c >= 73 && c <= 96) || (c >= 104))

{

Serial.println("Invalid Input!");

mySerial.println("Invalid Input!");

}

}

}

/\* function to flash inbuilt LED (5 times) on receiving valid data \*/

void led()

{ int i;

for (i = 0; i < 5; i++)

{ digitalWrite(LED\_PIN, HIGH);

delay(50);

digitalWrite(LED\_PIN, LOW);

delay(50);

}

}

Reference

Testing <https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-05-bluetooth-module-tutorial/>