ARDUINO UNO – LED CONTROL USING BLUETOOTH

AIM: To Control LEDs via Bluetooth using Arduino UNO.

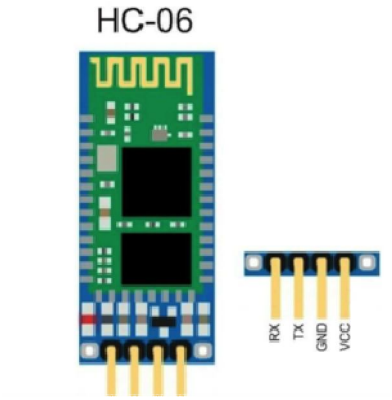
Apparatus:

1. Arduino UNO & programming cable
2. HC-06 Bluetooth module
3. 4 x LEDs
4. 4 x 220 ohms resistors
5. Breadboard
6. Connecting wires

Pinouts:

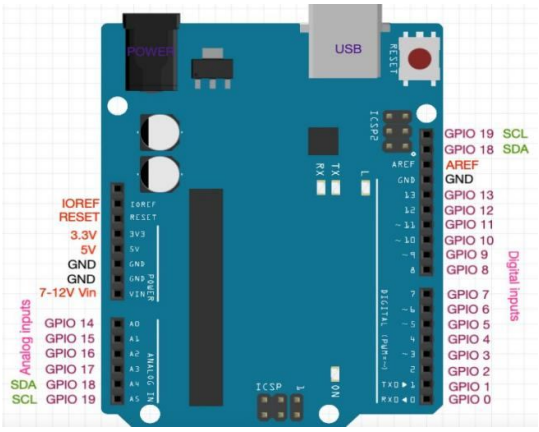
1. HC-05 Bluetooth Module

The HC-05 Bluetooth module is a slave Bluetooth module designed for wireless serial communication. It is a slave module meaning that it can receive serial data when serial data is sent out from a master Bluetooth device(device able to send serial data through the air: smart phones, PC).

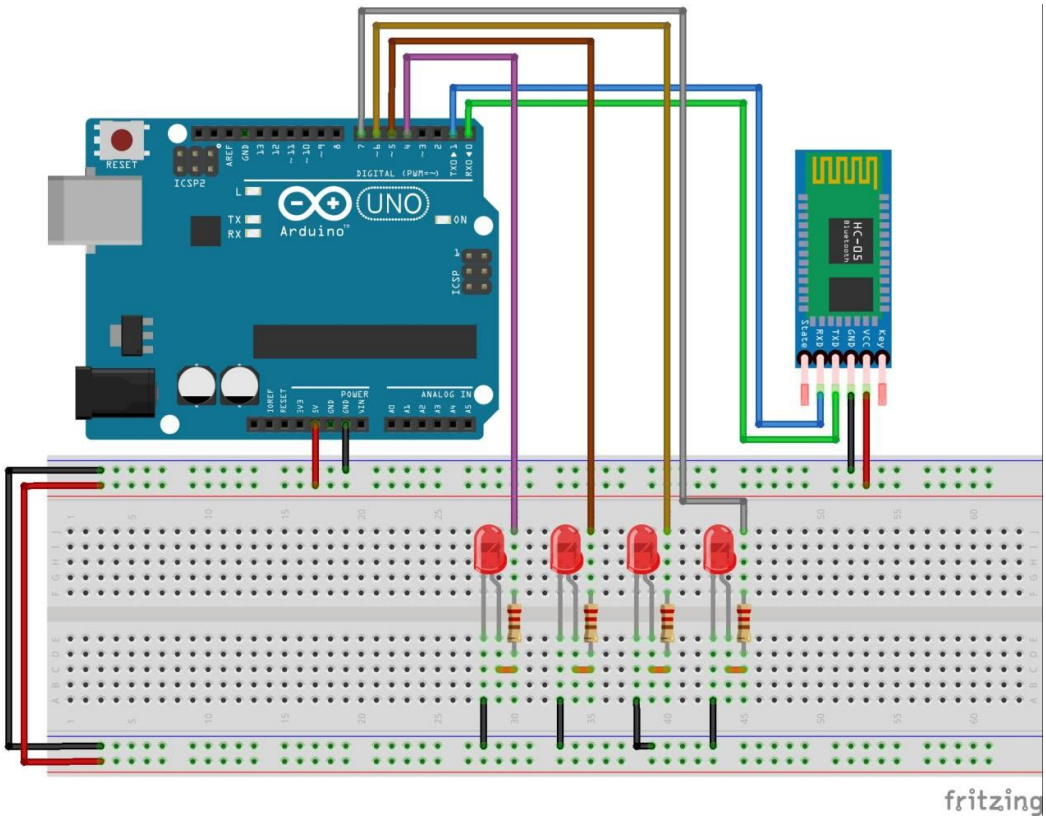


2. Arduino UNO

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.



Circuit:



Procedure:

a) Connect the circuit as shown.
We use the 5V power supply from Arduino
Connect the HC-05 to Arduino accordingly:

HC-06	--->	**Arduino UNO**
VCC	--->	+5V
GND	--->	GND
Tx	--->	Rx (pin 0)
Rx	--->	Tx (pin 1)

Connect the 1st LED anode to 220 ohm resistor and 220 ohm resistor to pin 4 of Arduino UNO. LED cathode should be connected to ground.
Similarly, connect 2nd LED to pin 5 of Arduino UNO, 3rd LED to pin 6 and 4th LED to pin 7 of Arduino UNO.

b) Program the Arduino.

Functions used in the program

1. Serial.begin(9600): Start serial communication with baud rate at 9600bps.
2. Serial.println("Hello"): Displays a message Hello or anything typed between "" over serial.
3. Serial.read(): Read the incoming data through serial.
4. digitalRead(*pin_number*): read the digital value of a pin, from digital I/Os on Arduino.
5. digitalWrite(*pin_number*, *value*): write the digital value (HIGH/LOW) to a pin, from digital I/Os on Arduino

Program:
Upload the following program to Arduino

```
//Naming Arduino pins 4, 5, 6 & 7 to their respective LEDs
#define LED1 4
#define LED2 5
#define LED3 6
#define LED4 7

//Character variable to store the command received from Bluetooth
char cmdIn;

void setup()
{
 Serial.begin(9600); //Start communication with Bluetooth module

 //Set LED pins as output
 pinMode(LED1, OUTPUT);
 pinMode(LED2, OUTPUT);
 pinMode(LED3, OUTPUT);
 pinMode(LED4, OUTPUT);

 //Just a text to confirm that Bluetooth communication has established
 Serial.println("Hello");
 //You should see this text 'Hello' on the mobile app
}

void loop()
{
 //If the incoming data is available from Bluetooth,
 //then store the data in the variable
 if (Serial.available()>0)
 {
  cmdIn = Serial.read();
 }

  /*
 *These if statements check that the received data is 'A', 'B', 'C' or 'D'.
 *when any of the command is received, the respective if statement gets executed.
 *Here, we toggle the respective LED. To do this, we first check if the LED
 *is ON or OFF and perform the opposite action, and display the status of LED.
 *Example: if LED is ON then turn the LED OFF.
 */

 if (cmdIn == 'A') //'A' is received
 {
  if (digitalRead(LED1) == LOW) //check if LED is OFF
  {
   digitalWrite(LED1, HIGH); //Turn LED ON
   Serial.println("LED1 is ON"); //Display that LED is ON
  }
  else if (digitalRead(LED1) == HIGH) //Check if LED is ON
  {
   digitalWrite(LED1, LOW); //Turn LED OFF
   Serial.println("LED1 is OFF"); //Display that LED is OFF
  }
 }
```

```
 if (cmdIn == 'B')
 {
  if (digitalRead(LED2) == LOW)
  {
    digitalWrite(LED2, HIGH);
    Serial.println("LED2 is ON");
  }
  else if (digitalRead(LED2) == HIGH)
  {
    digitalWrite(LED2, LOW);
    Serial.println("LED2 is OFF");
  }
 }

 /*As you can see the code in below if statements is different than the above ones.
  *Here, the same process is done as above, but in a different way
  *digitalWrite(LED3, !digitalRead(LED3)) to understand this function, here is the format
  *digitalWrite(_output_pin_, NOT of _output_pin_)
  *The exclamation mark '!' refers to NOT function.
  *Hence the function digitalWrite(LED3, !digitalRead(LED3)) means
  *turn LED3 ON or OFF, based on the compliment of previous state of that LED3
  *To display the message about status of LED, we join one string "LED3 is "
  *with the string value of current state of LED3 i.e. (String(digitalRead(LED3))
  *
 */

 if (cmdIn == 'C')
 {
  digitalWrite(LED3, !digitalRead(LED3));
  Serial.println("LED3 is " + (String(digitalRead(LED3))));
 }

 if (cmdIn == 'D')
 {
  digitalWrite(LED4, !digitalRead(LED4));
  Serial.println("LED4 is " + (String(digitalRead(LED4))));
 }

 delay(1000); //wait for 1 second
}
```

Mobile App:
Download and install the following mobile application from Google play store or scan the QR code below

Bluetooth Terminal HC-05 by mightyIT

When the circuit is on, the LED on the Bluetooth module starts to blink.

1. Go to Bluetooth settings on your phone.
2. Scan for new devices.
3. Find HC-05 or HC-06 under available devices.
4. Enter pairing key as 1234 or 0000
5. Wait till your phone adds the device under paired devices.
6. Once this is done, open the installed app.
7. Look for HC-05 or HC-06 under paired devices.
8. Wait for the device to get connected.
9. Once connected, the LED on the Bluetooth Module is stable & ON.

- When the application is connected to Bluetooth module, Then press the reset button on the Arduino board. You will receive a ‘‘Hello’’ message displayed on the black terminal.
- Then type the commands A, B, C or D in the textbox which says ‘Enter ASCII Command’ and click on ‘Send ASCII’ button just besides the textbox.
- You will see the respective LED connected to Arduino turning ON and a text will be displayed on the app.
- Now type and send the same command you sent earlier, now you will see the LED turn OFF and a text displayed
- Remember, command ‘A’ controls LED1, ‘B’ controls LED2, ‘C’ controls LED3 & ‘D’ controls LED4.
- Try sending other commands, also try sending multiple commands like ABC, BD, ABCD, AC, etc.