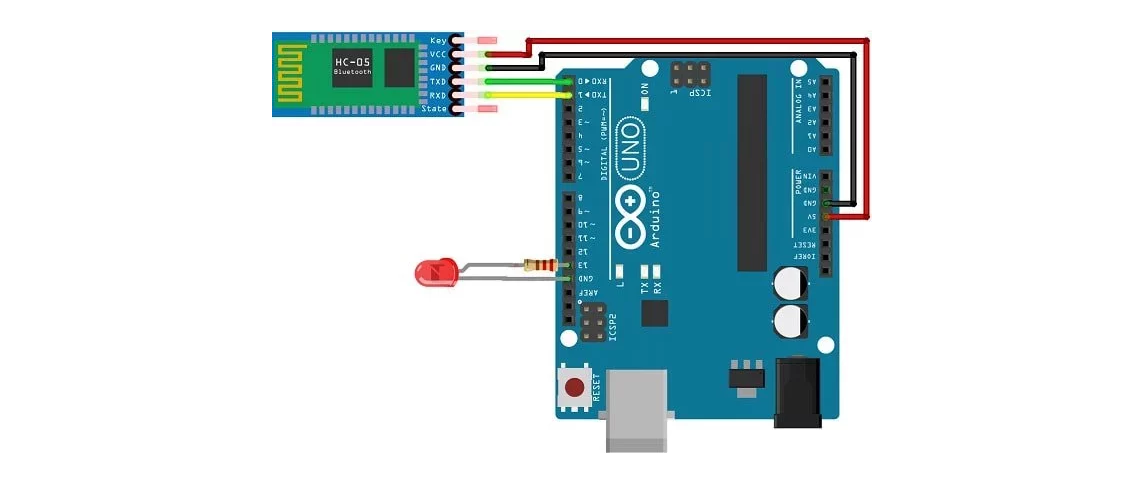
PROJECT

Bluetooth controlled led using Arduino

HC 05/06 works on serial communication. The Android app is designed to send serial data to the Arduino Bluetooth module when a button is pressed on the app. The Arduino Bluetooth module at the other end receives the data and sends it to the Arduino through the TX pin of the Bluetooth module (connected to RX pin of Arduino). The code uploaded to the Arduino checks the received data and compares it. If the received data is 1, the LED turns ON. The LED turns OFF when the received data is 0. You can open the serial monitor and watch the received data while connecting.

Circuit and Working



PIN CONNECTIONS

RX (Pin 0) ———> TX  
TX (Pin 1) ———> RX  
5V ———> VCC  
GND ———> GND

Connect an LED positive to pin 13 of the Arduino [through a resistance](https://eepower.com/textbook/vol-i-foundations-power-design/chapter-3-power-ac-systems/passive-components-ac-circuits) (valued between 220Ω–1KΩ). Connect its negative to GND, and you're done with the circuit!

WORKING

1. Turn ON the HC 05/06 Bluetooth module by powering the Arduino.
2. Scan your smartphone for available devices.
3. 3. Pair your smartphone to the HC 05/06 by entering default password 1234 OR 0000.

* Install the application on your Android device. <https://play.google.com/store/apps/details?id=com.circuitmagic.arduinobluetooth>
* Open the application.
* Press "paired devices".
* Select your Bluetooth module from the list (HC-05/06)
* After connecting successfully, press the ON button to turn the LED on and the OFF button to turn the LED off.
* Disconnect the button to disconnect the Bluetooth module.

CODE

#include "Wire.h" // This library allows you to communicate with I2C devices.

const int MPU\_ADDR = 0x68; // I2C address of the MPU-6050. If AD0 pin is set to HIGH, the I2C address will be 0x69.

int16\_t accelerometer\_x, accelerometer\_y, accelerometer\_z; // variables for accelerometer raw data

int16\_t gyro\_x, gyro\_y, gyro\_z; // variables for gyro raw data

int16\_t temperature; // variables for temperature data

char tmp\_str[7]; // temporary variable used in convert function

char\* convert\_int16\_to\_str(int16\_t i) { // converts int16 to string. Moreover, resulting strings will have the same length in the debug monitor.

sprintf(tmp\_str, "%6d", i);

return tmp\_str;

}

void setup() {

Serial.begin(9600);

Wire.begin();

Wire.beginTransmission(MPU\_ADDR); // Begins a transmission to the I2C slave (GY-521 board)

Wire.write(0x6B); // PWR\_MGMT\_1 register

Wire.write(0); // set to zero (wakes up the MPU-6050)

Wire.endTransmission(true);

}

void loop() {

Wire.beginTransmission(MPU\_ADDR);

Wire.write(0x3B); // starting with register 0x3B (ACCEL\_XOUT\_H) [MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.2, p.40]

Wire.endTransmission(false); // the parameter indicates that the Arduino will send a restart. As a result, the connection is kept active.

Wire.requestFrom(MPU\_ADDR, 7\*2, true); // request a total of 7\*2=14 registers

// "Wire.read()<<8 | Wire.read();" means two registers are read and stored in the same variable

accelerometer\_x = Wire.read()<<8 | Wire.read(); // reading registers: 0x3B (ACCEL\_XOUT\_H) and 0x3C (ACCEL\_XOUT\_L)

accelerometer\_y = Wire.read()<<8 | Wire.read(); // reading registers: 0x3D (ACCEL\_YOUT\_H) and 0x3E (ACCEL\_YOUT\_L)

accelerometer\_z = Wire.read()<<8 | Wire.read(); // reading registers: 0x3F (ACCEL\_ZOUT\_H) and 0x40 (ACCEL\_ZOUT\_L)

temperature = Wire.read()<<8 | Wire.read(); // reading registers: 0x41 (TEMP\_OUT\_H) and 0x42 (TEMP\_OUT\_L)

gyro\_x = Wire.read()<<8 | Wire.read(); // reading registers: 0x43 (GYRO\_XOUT\_H) and 0x44 (GYRO\_XOUT\_L)

gyro\_y = Wire.read()<<8 | Wire.read(); // reading registers: 0x45 (GYRO\_YOUT\_H) and 0x46 (GYRO\_YOUT\_L)

gyro\_z = Wire.read()<<8 | Wire.read(); // reading registers: 0x47 (GYRO\_ZOUT\_H) and 0x48 (GYRO\_ZOUT\_L)

// print out data

Serial.print("aX = "); Serial.print(convert\_int16\_to\_str(accelerometer\_x));

Serial.print(" | aY = "); Serial.print(convert\_int16\_to\_str(accelerometer\_y));

Serial.print(" | aZ = "); Serial.print(convert\_int16\_to\_str(accelerometer\_z));

// the following equation was taken from the documentation [MPU-6000/MPU-6050 Register Map and Description, p.30]

Serial.print(" | tmp = "); Serial.print(temperature/340.00+36.53);

Serial.print(" | gX = "); Serial.print(convert\_int16\_to\_str(gyro\_x));

Serial.print(" | gY = "); Serial.print(convert\_int16\_to\_str(gyro\_y));

Serial.print(" | gZ = "); Serial.print(convert\_int16\_to\_str(gyro\_z));

Serial.println();

// delay

delay(1000);

}

