

Group – 11

Soumava Paul 16EE10056

Yerramsetty Rohit 16EE10055

Experiment 10

Bluetooth Interfacing with Arduino and Smartphone

Aim:

- Interfacing a Bluetooth module (HC05) to your (Android) smartphone using Virtuino
- Controlling a LED using Bluetooth from Smartphone
- Interfacing an LDR and temperature sensor (LM35) with Smartphone using Bluetooth and Virtuino

Apparatus Required:

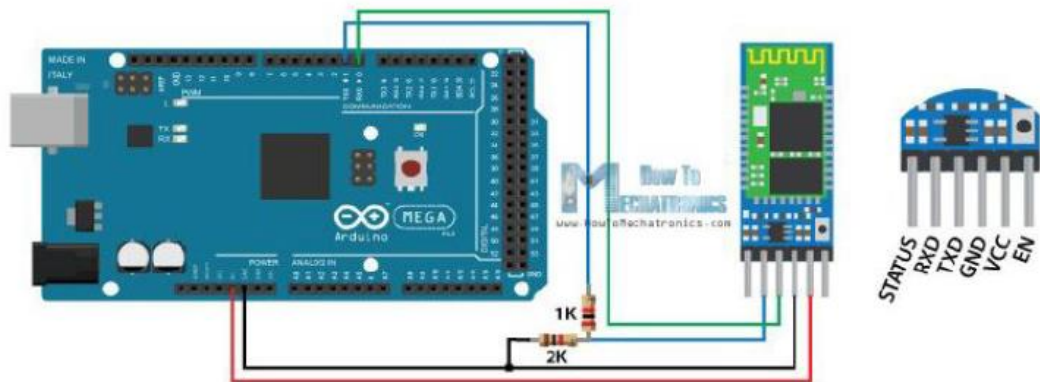
- Arduino Uno Board
- PC with Arduino installed
- Bluetooth module (HC-05)
- Android Smartphone with Virtuino app installed
- Light Dependent Resistor and LM35(Temperature Sensor)

Theory/Procedure:

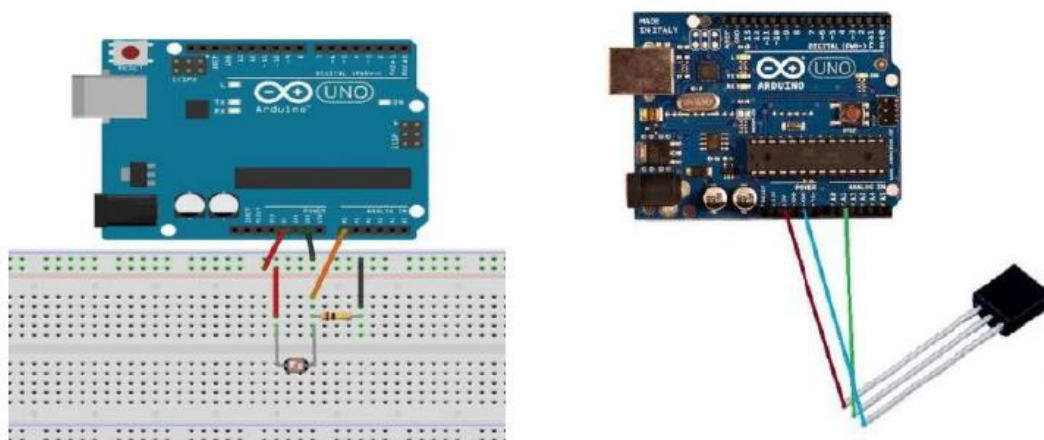
The HC-05 module is a Bluetooth SPP (Serial Port Protocol) module, which means it communicates with the Arduino via the Serial Communication. The circuit connection diagram is shown below.

The logic voltage level of the data pins is 3.3V. So, the line between the Arduino TX (Transmit Pin, which has 5V output) and the Bluetooth module RX (Receive Pin, which supports only 3.3V) needs to be connected through a voltage divider in order not to burn the module.

On the other hand, the line between the Bluetooth module TX pin and the Arduino RX pin can be connected directly because the 3.3V signal from the Bluetooth module is enough to be accepted as a high logic at the Arduino Board.



Interfacing Bluetooth module



Circuits for LDR and LM35

Codes

HC05_BlinkLED

```
/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial

#include <SoftwareSerial.h>
SoftwareSerial SwSerial(10, 11); // RX, TX

#include <BlynkSimpleSerialBLE.h>
#include <SoftwareSerial.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "e21b3eb0994a4754868f83bb8dd9197b";

SoftwareSerial SerialBLE(10, 11); // RX, TX

void setup()
{
  // Debug console
  Serial.begin(9600);

  SerialBLE.begin(9600);
  Blynk.begin(SerialBLE, auth);

  Serial.println("Waiting for connections...");
}

void loop()
{
  Blynk.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
  // to avoid delay() function!
}
```

Code for LDR and LM35

```
#define DHTPIN 6
#define DHTTYPE DHT11
//DHT dht(DHTPIN, DHTTYPE);
```

```

#include "VirtuinoBluetooth.h"           // virtuino library

// Code to use SoftwareSerial

#include <SoftwareSerial.h>               // Disable this line if you want to
use hardware serial

SoftwareSerial bluetoothSerial = SoftwareSerial(2,3); // arduino RX pin=2
arduino TX pin=3  connect the arduino RX pin to bluetooth module TX pin -
connect the arduino TX pin to bluetooth module RX pin. Disable this line if
you want to use hardware serial

VirtuinoBluetooth virtuino(bluetoothSerial,9600);    // Set SoftwareSerial
baud rate. - Disable this line if you want to use hardware serial


// Code to use HardwareSerial

// VirtuinoBluetooth virtuino(Serial1);           // enable this line and disable all
SoftwareSerial lines

// Open VirtuinoBluetooth.h file on the virtuino
library folder -> disable the line: #define
BLUETOOTH_USE_SOFTWARE_SERIAL

// Connect HC-05 module to Arduino (MEGA or
DUE) Serial1. (pins: 18,19)


//=====
===== setup

//=====
=====

int ldr_input = A0;

void setup(void){

  Serial.begin(9600);  // start monitor serial port

```

```
bluetoothSerial.begin(9600);    // Enable this line if you want to use
software serial (UNO, Nano etc.)

//Serial1.begin(9600);          // Enable this line if you want to use hardware
serial (Mega, DUE etc.)
```

```
virtuino.DEBUG=true;           // set this value TRUE to enable the serial
monitor status
```

```
// Start up the library Dallas Temperature IC Control Library
// dht.begin();
pinMode(ldr_input, INPUT);
Serial.println("Setup completed.");
}
```

```
//=====
===== setup
void readTemperatures(){
    Serial.println("Read sensors' values...");
    int ldrval = analogRead(ldr_input);
    Serial.println("LDRVAL="+String(ldrval)+" *ohm");
    virtuino.vMemoryWrite(0,ldrval); // write temperature 1 to virtual pin V0.
    On Virtuino panel add a value display or an analog instrument to pin V0
    // float temperature = dht.readTemperature();
    // float humidity = dht.readHumidity();
    // if (isnan(temperature) || isnan(humidity)) {
    //     Serial.println("Failed to read from DHT");
    // }
    // else {
    //     Serial.println("Temp="+String(temperature)+" *C");
    //     Serial.println("Humidity="+String(humidity)+" %");
```

```

//      virtuino.vMemoryWrite(0,temperature); // write temperature 1 to virtual
pin V0. On Virtuino panel add a value display or an analog instrument to pin V0
//      virtuino.vMemoryWrite(1,humidity); // write temperature 1 to virtual
pin V1. On Virtuino panel add a value display or an analog instrument to pin V1
//  }

}

//=====
===== setup
//=====
=====

void loop(void){

    virtuino.run();          // necessary command to communicate with Virtuino
    android app

    //----- enter your loop code below here

    //----- avoid to use delay() function in your code. Use the command
    virtuino.vDelay() instead of delay()

    //----- your code .....


    readTemperatures();      // Dont' read the sensors values every cicle
    virtuino.vDelay(1000);    // Add a delay at least 1 second long.

}

```

Discussions

In this experiment, we have dealt with the Bluetooth module to interface between our android phone and the Arduino Uno board. The communication between the module and the interface was carried out by the transmitter and receiver pins of the Arduino which was wirelessly sensed by the android phone through the bluetooth module (HC05). Using the bluetooth module, the arduino can be controlled by either PC or smartphone and various sensors and sophisticated wireless devices can be combined. One precaution that must be taken while uploading the code is to remove

all the connections from the Rx pin. After the code is uploaded, only then we should connect a wire to Rx, otherwise it will throw not in sync error.