भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology
Hyderabad

UART Interfacing through RaspberryPi and Arduino

Teaching Learning Center
IIT Hyderabad

Prashanth Kumar Aleshvaram and G V V Sharma*

Contents

***Abstract*—The process of serial communication between Arduino and RaspberryPi is explained by calculating the value of an unkown resistance in Arduino and sending the value to RaspberryPi through UART pins.**

1 COMPONENTS

The component details are available in Table I.

Component	Value	Quantity
Resistor	220 Ohm	1
Resistor	1K	1
Arduino	Uno	1
Jumper Wires	M to M	20
Jumper Wires	M to F	5
Bread board		1
Pi	Model B, Rev 3	1

TABLE I

2 Enabling UART in RaspberryPi

Problem 2.1. Connect the RaspberryPi to the desktop and power it on.

Problem 2.2. Open

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

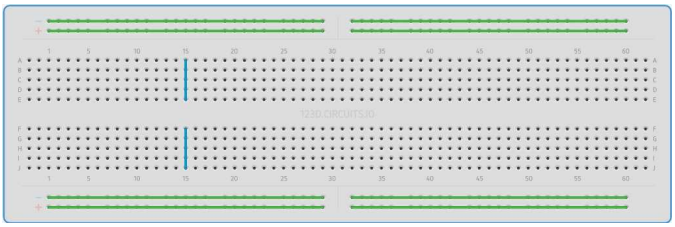

Fig. 3.1: Breadboard

```
sudo nano /boot/cmdline.txt
```

and remove **console=serial0,115200** or **console=ttyAMA0,115200**. Don't delete anything else in the file. **Ctrl+X** and exit.

Problem 2.3. In order to enable GPIO serial port,

```
sudo nano /boot/config.txt
```

and add

```
enable_uart=1
```

at the end of the file. Reboot.

Use **/dev/ttyS0** in any code which accesses the Serial Port of RaspberryPi.

3 Measuring the Resistance

Problem 3.1. Connect the 5V pin of the Arduino to an extreme pin of the Breadboard shown in Fig. 3.1. Let this pin be Vcc.

Problem 3.2. Connect the GND pin of the Arduino to the opposite extreme pin of the Breadboard.

Problem 3.3. Let R_1 be the known resistor and R_2 be the unknown resistor. Connect R_1 and R_2 in series such that R_1 is connected to GND and R_2 is connected to Vcc as in Fig. 3.3

Problem 3.4. Connect the junction between the two resistors to the A0 pin on the Arduino.

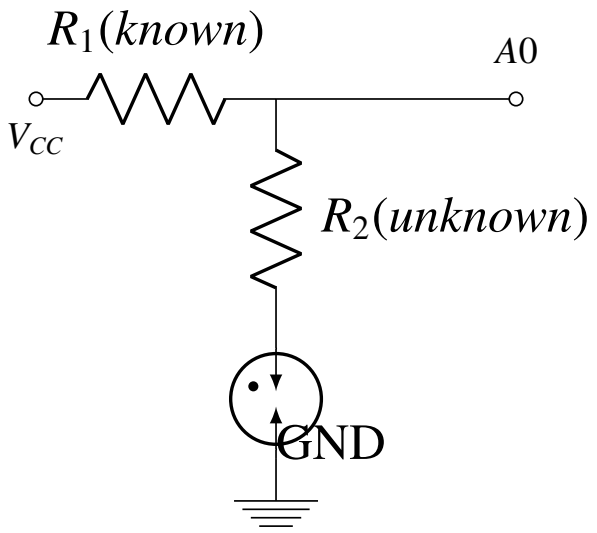

Fig. 3.3

Problem 3.5. Connect the arduino to the RaspberryPi so that it is powered.

Problem 3.6. Open the Arduino IDE and type the following code. Open the serial monitor to view the resistance.

Solution:

```
//Remove the following line if you
    are using the Arduino IDE
//#include "Arduino.h"

//Declaration or Initialization
int Vq=0;
float Vcc=5,V0;
float R1=220;
//R1 is known resistance

void setup()
{
  Serial.begin(9600);
}
void loop()
{
  int R2=0;
  //R2 is unknown resistance
  Vq=analogRead(0);
  //Reading from A0 and Vq is an
     integer between 0 to 1023
  V0=Vcc*Vq/1024;
  //V0 is actual output voltage
     across Unknown resistance R2
  R2=R1*((Vcc)/(V0)-1.0);
  //Formula from voltage divider
     circuit
  delay(3000);
  Serial.write(R2);
  //Writing the measured
     resistance to serial port of
     arduino
  Serial.println(R2);
  //To see the value in serial
     monitor of Arduino
}
```

4 UART Communication

The Arduino transmitting the resistance through it's serial port.

Problem 4.1. Connect the Pin 6 (not GPIO 6) of Pi to the GND pin of the Arduino.

Problem 4.2. Connect GPIO 16 pin (UART R_x) of Pi to the T_x pin of the Arduino.

Problem 4.3. Install the python serial package

```
sudo apt-get install python-serial
```

Problem 4.4. Open a text editor and type the following code. Save it as **filename.py**.

Solution:

```
#imporing serial package
import serial

#setting baudrate and giving the
   port
ser = serial.Serial(port = "/dev/
   ttyS0", baudrate=9600)

print "Serial is open!"
while 1:
        data=ser.read(1)
        #read the GPIO 16 pin
        print data
```

Problem 4.5. Run the above code using the following command. You should see the resistance value displayed on the terminal.

```
python filename.py
```