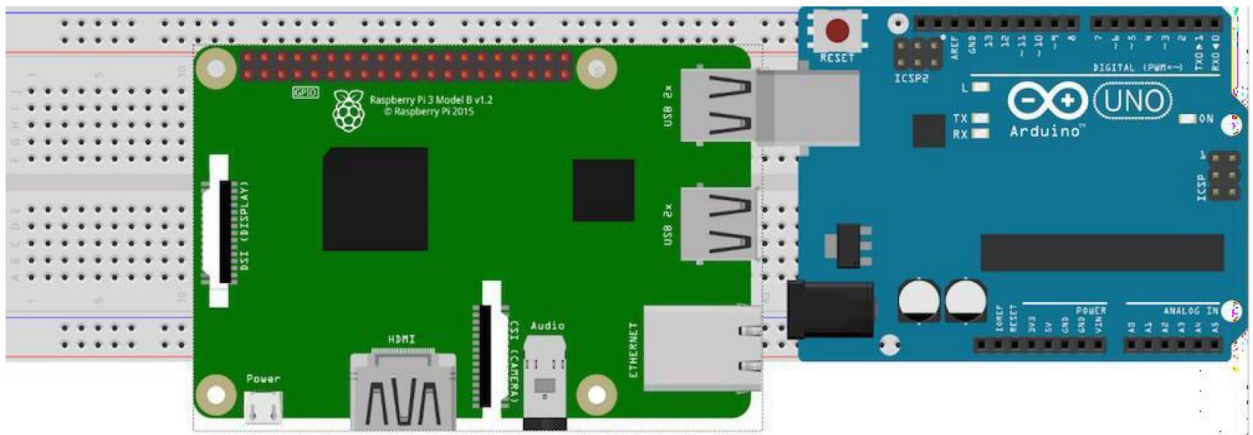


4.3.1 Interface a Raspberry Pi with an Arduino



Interface a Raspberry Pi with an Arduino so the two boards can communicate with one another. Sometimes you may need to connect an Arduino to a Raspberry Pi. For example, if you have sensors, motors, and actuators, you can connect these to the Arduino and make the Arduino send values to and from the Raspberry Pi. This way, we can separate the computing intensive tasks (done by the Raspberry Pi) and controlling tasks (done by the Arduino).

we will connect an Arduino to a Raspberry Pi and have the Arduino send "Hello from Arduino" to the Raspberry Pi, and the Raspberry Pi will blink an LED upon receiving the command from the Arduino.

For communication, we will use simple serial communication over USB cable. So, let's get started! Connect the LED to pin number 11 as shown in the picture below.



Turn on the Raspberry Pi and open Python 3 in a new window.

Write the following code in the new window and save it. (Save to your desktop so you don't lose it.)

```
import serial
import RPi.GPIO as GPIO
import time
```

```
ser=serial.Serial("/dev/ttyACM0",9600) #change ACM number as found from ls /dev/tty/ACM*
ser.baudrate=9600
def blink(pin):
    GPIO.output(pin,GPIO.HIGH)
    time.sleep(1)
```

```
GPIO.output(pin,GPIO.LOW)time.sleep(1)
```

```
return
```

```
GPIO.setmode(GPIO.BOARD)GPIO.setup(11, GPIO.OUT)
```

```
while True:
```

```
read_ser=ser.readline()print(read_ser)
```

```
if(read_ser=="Hello From Arduino!"):
```

```
blink(11)
```

Now open Arduino IDE and upload the following code to your Arduino.String data="Hello From Arduino!";

```
void setup()
```

```
{
```

```
// put your setup code here, to run once:Serial.begin(9600);
```

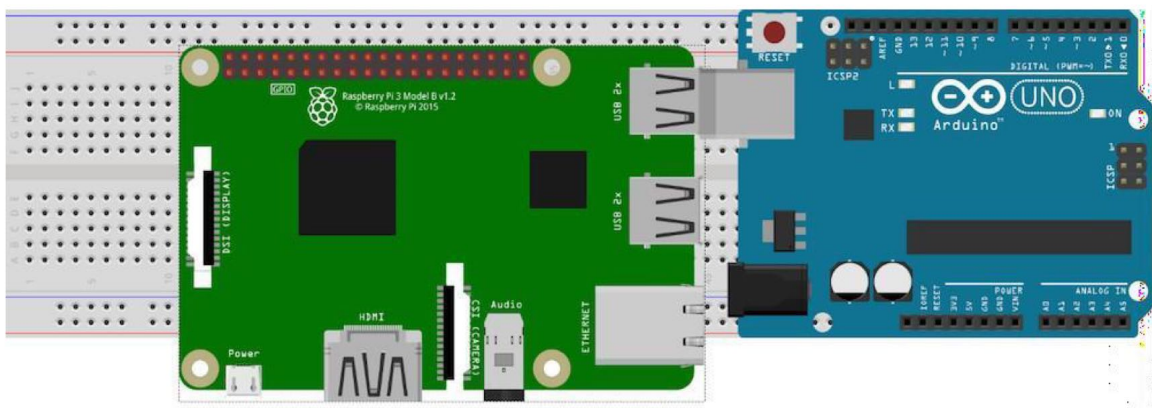
```
}
```

```
void loop()
```

```
{
```

```
// put your main code here, to run repeatedly:Serial.println(data);//data that is being Sent delay(200);
```

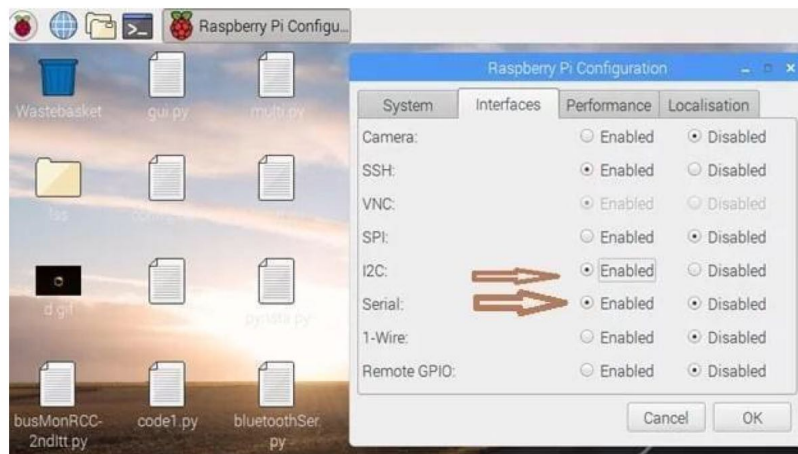
```
}
```



Make sure the code is uploaded to Arduino.

In your Raspberry Pi interface, be sure to enable Serial and I2C in PiConfig.





Next, you'll need to restart your Raspberry Pi. Open the Terminal and execute these commands:

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

```
sudo pip install pyserial
```

Connect your Arduino to your Raspberry Pi. Execute.

```
ls /dev/tty*
```

Then find a line with `/dev/ttyACM0` or something like `/dev/ttyACM1` etc. (check for an ACM with any number 0,1,2 etc.)

```

pi@raspberrypi: ~
File Edit Tabs Help
/dev/tty1 /dev/tty20 /dev/tty31 /dev/tty42 /dev/tty53 /dev/tty7
/dev/tty10 /dev/tty21 /dev/tty32 /dev/tty43 /dev/tty54 /dev/tty8
/dev/tty11 /dev/tty22 /dev/tty33 /dev/tty44 /dev/tty55 /dev/tty9
/dev/tty12 /dev/tty23 /dev/tty34 /dev/tty45 /dev/tty56 /dev/ttyACM0
/dev/tty13 /dev/tty24 /dev/tty35 /dev/tty46 /dev/tty57 /dev/ttyAMA0
/dev/tty14 /dev/tty25 /dev/tty36 /dev/tty47 /dev/tty58 /dev/ttyprintk
/dev/tty15 /dev/tty26 /dev/tty37 /dev/tty48 /dev/tty59
/dev/tty16 /dev/tty27 /dev/tty38 /dev/tty49 /dev/tty6
/dev/tty17 /dev/tty28 /dev/tty39 /dev/tty5 /dev/tty60
/dev/tty18 /dev/tty29 /dev/tty4 /dev/tty50 /dev/tty61
pi@raspberrypi:~$ ls /dev/tty*
/dev/tty /dev/tty19 /dev/tty3 /dev/tty40 /dev/tty51 /dev/tty62
/dev/tty0 /dev/tty2 /dev/tty30 /dev/tty41 /dev/tty52 /dev/tty63
/dev/tty1 /dev/tty20 /dev/tty31 /dev/tty42 /dev/tty53 /dev/tty7
/dev/tty10 /dev/tty21 /dev/tty32 /dev/tty43 /dev/tty54 /dev/tty8
/dev/tty11 /dev/tty22 /dev/tty33 /dev/tty44 /dev/tty55 /dev/tty9
/dev/tty12 /dev/tty23 /dev/tty34 /dev/tty45 /dev/tty56 /dev/ttyACM1
/dev/tty13 /dev/tty24 /dev/tty35 /dev/tty46 /dev/tty57 /dev/ttyAMA0
/dev/tty14 /dev/tty25 /dev/tty36 /dev/tty47 /dev/tty58 /dev/ttyprintk
/dev/tty15 /dev/tty26 /dev/tty37 /dev/tty48 /dev/tty59
/dev/tty16 /dev/tty27 /dev/tty38 /dev/tty49 /dev/tty6
/dev/tty17 /dev/tty28 /dev/tty39 /dev/tty5 /dev/tty60
/dev/tty18 /dev/tty29 /dev/tty4 /dev/tty50 /dev/tty61
pi@raspberrypi:~$

```

Open Python again and change `ser=serial.Serial("dev/ttyACM1",9600)` to the ACM number you found. So, if in your case you got `ACM0`, the line should look like this: `ser=serial.Serial("dev/ttyACM0",9600)`. Now run the program you just created in Python3. You will see "Hello From Arduino!" in the Python terminal, and your LED should be blinking as well!