# Lab Guide 1: Getting Raspberry PI to communicate with Arduino（标题1，Alt+1）编写提示：本上机指导书实用于版本V200R002，最后更新时间2005-05-08

## Description

RPI is a development board with relatively more power than Arduino, in addition to that RPI has built-in network connectivity. Arduino, on the other hand, have 6 analog ports and its operation are more real-time\* compare to RPI. Thus, there are times that we need to make use of both together as one system. In this lab guide will make use of RPI as the host and Arduino as the peripheral unit.

## Lab Objective(s)

To use RPI to program Arduino Development Board

To enable communication between RPI and Arduino

To enable data collected from Arduino to be transmitted to the Internet via RPI internet.

## Grouping Method

This lab guide is to be completed individually.

## Lab Environment

For this lab, we will need the following apparatus: -

A monitor that has HDMI input and power supply

HDMI cable

USB mouse and keyboard

RPI Kits (RPI, Power supply, SD card, SD card Reader)

A Windows 10 computer with Internet access.

Wi-Fi Access Point with direct Internet connection.

Arduino Development Kit

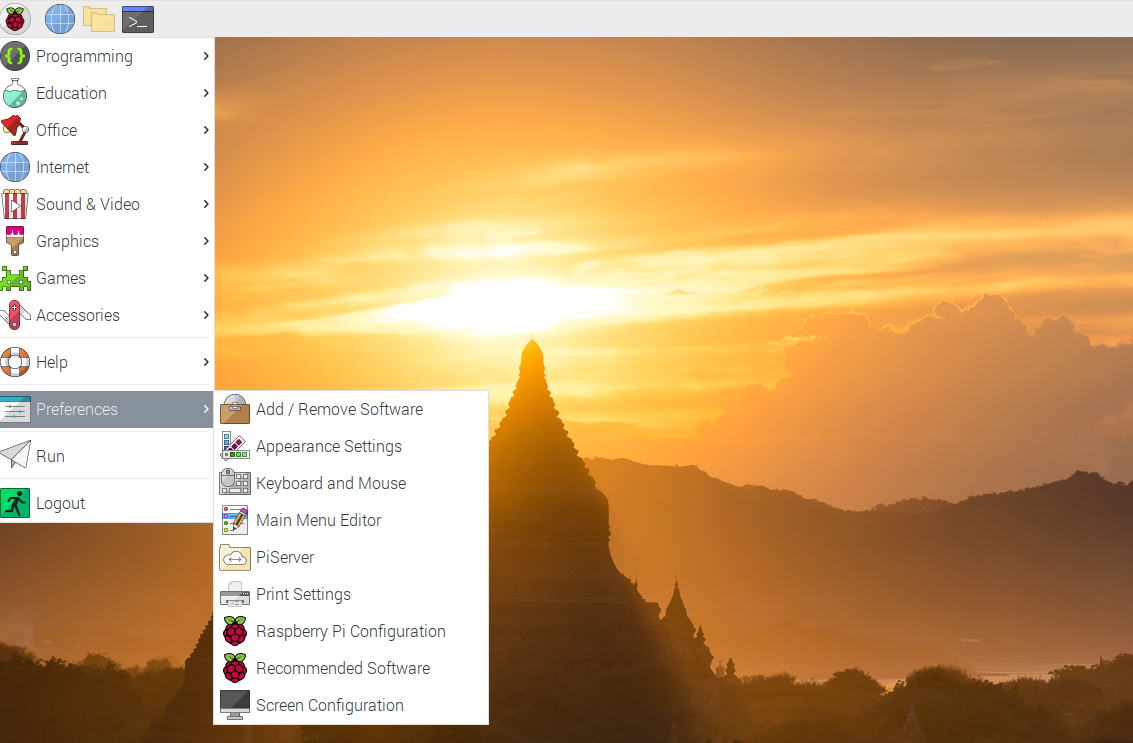
## Task

### Programming Arduino from RPI

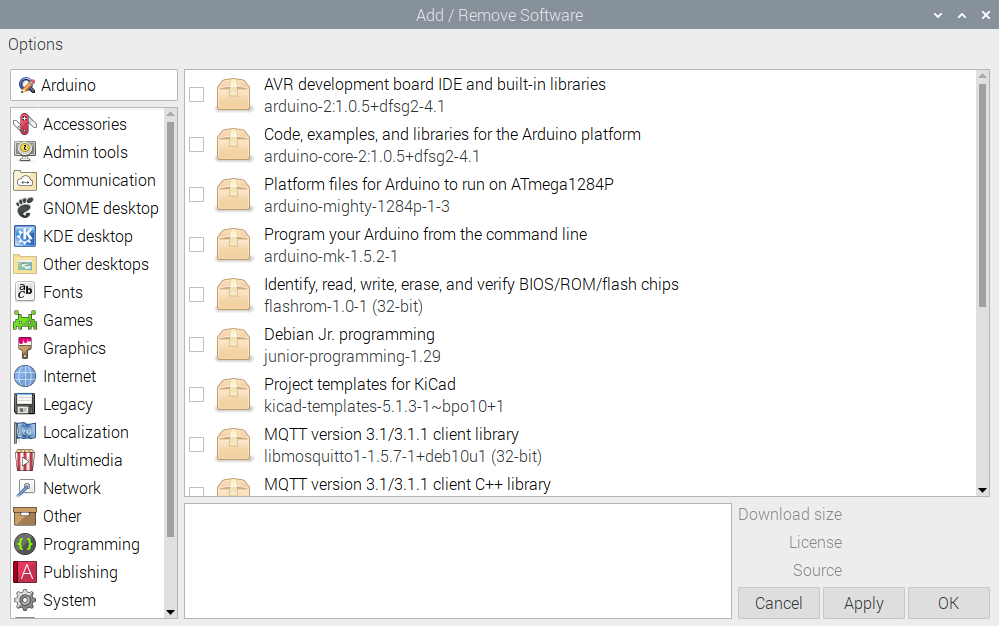
#### Setup your RPI, attached the RPI with HDMI to monitor, connect RPI with mouse and keyboard.

#### Power up your RPI and login. Similar to your program, we need to install Arduino IDE in RPI.

#### In order to install Arduino IDE in RPI, click on Raspberry Pi > Add / Remove software option. See the screenshot below:-

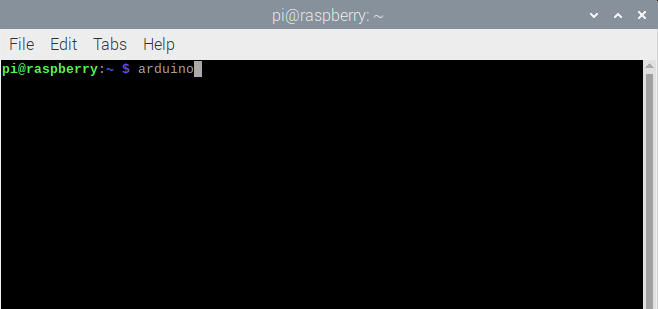


#### The following screen will pop up, search for Arduino and hit enter.

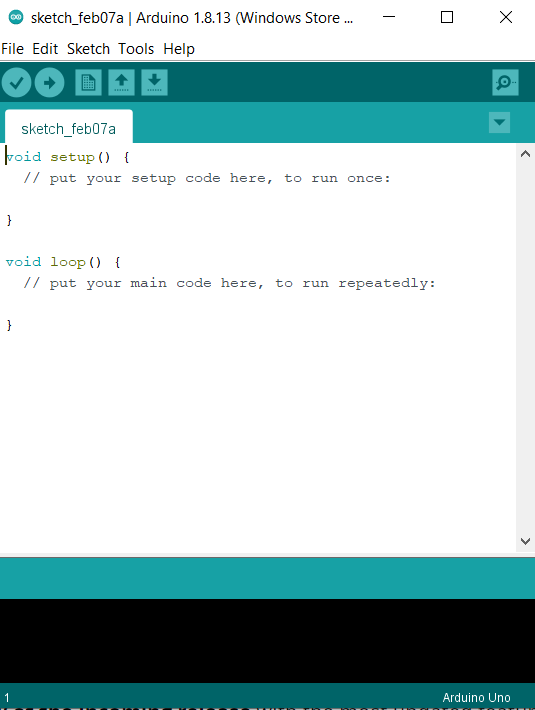


#### Double click on the “AVR development board IDE and built-in libraries” to install Arduino IDE into RPI.

#### Once the installation is completed on the RPI, open a terminal and start the Arduino IDE with the Arduino command. E.g.



#### Once you hit enter, you will see the familiar Arduino IDE pop up. Do not close the terminal when the Arduino IDE is running.



#### Connect your Arduino Development board to your RPI using a USB cable. Your Arduino will be powered by your RPI therefore you don’t have to power it separately.

#### Once it is connected, under the Tools > Port check if RPI is able to communicate with your Arduino board. Look for /dev/ttyUSB0.

#### Now you can try to load your application into it.

### Sending data from Arduino to Serial Port

#### In order to send data via the Serial Port, use the following code in Arduino.

void setup()

{

Serial.begin(9600); // send and receive at 9600 baud

}

int number = 0

void loop()

{

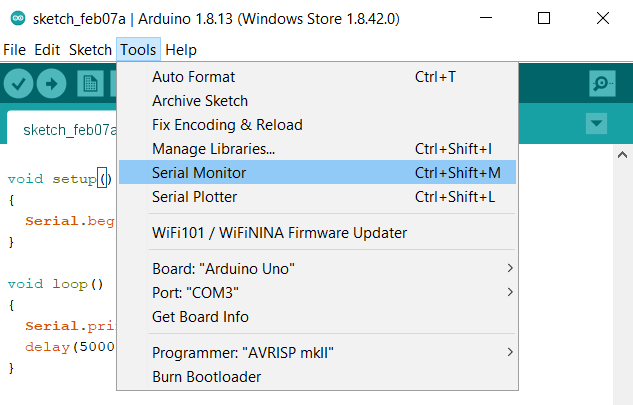
number++;

Serial.print(number);

delay(5000); // delay half second between numbers

}

#### Similar to a computer, we can start to monitor the serial port.



#### From the newly open “Serial Monitor” confirm that we are receiving data from the Arduino.

Arduino

**Activity**: Use the lab guide 7 Section 1.5.3, use Arduino to get the temperature data and transmit it to RPI.

### Sending data from RPI to Arduino using Serial Port

#### Change the Arduino code with the following code so that the Arduino can accept data from the host. In this case the RPI.

void setup()

{

Serial.begin(9600); // send and receive at 9600 baud

}

int number = 0;

void loop()

{

number++;

if (Serial.available() > 0){

String s1 = Serial.readStringUntil('\n');

Serial.print("S:");

Serial.println(s1);

}

delay(500);

}

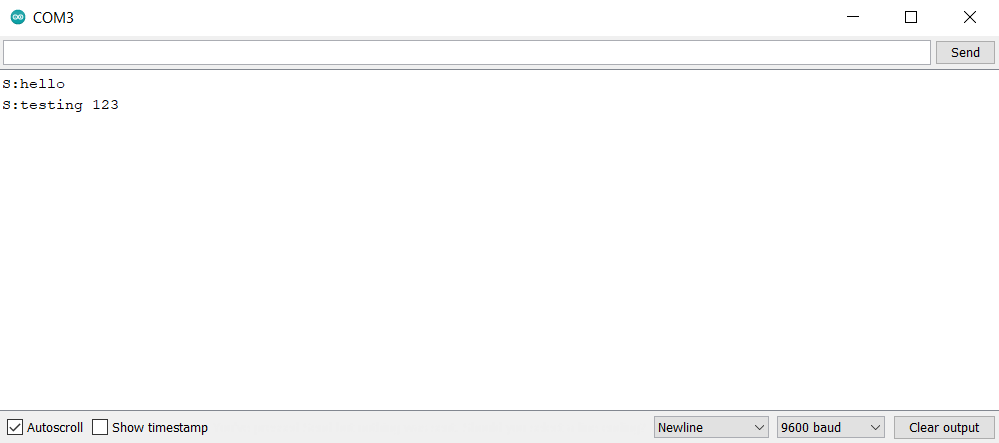
#### Connect your Arduino Development board to your RPI using a USB cable. Your Arduino will be powered by your RPI therefore you don’t have to power it separately.

#### Once it is connected, under the Tools > Port check if RPI is able to communicate with your Arduino board. Look for /dev/ttyUSB0.

#### Now you can try to load your application into it.

#### After loading the application into your Arduino, start the “Serial Monitor”.

#### In order to test sending data from RPI to Arduino, put some data in the textbox above and press send.



#### When you see “S:”<your string>, it means that Arduino gets your message and the communication is successful.

**Activity**: Use the lab guide 7.0 section 1.5.1, modify the code where RPI can send instruction in order to turn on and off the LED.

### Sending and receiving data between RPI and Arduino using Python.

#### In your Raspberry, open your Python code and run the following code.

#### First, we need to install pySerial, a Python library for serial communication.

#### Please refer to Lab Guide 5 on how to install a library.

#### After the library is installed, use the following code to send and receive data.

#To receive

import serial

ser = serial.Serial('/dev/ttyUSB0')

ser.flushInput()

while True:

while ser.in\_waiting:

x = ser.readline().strip()

line = x.decode('ascii')

print (line)

#To send

import serial

ser = serial.Serial('/dev/ttyUSB0')

message = “hello world”

ser.write(message.encode())

#### Please make sure you close all the other “Serial Mode” application before running the code.

#### Please make sure that your Arduino is working.

## Additional Tasks

Develop a code that uses the Arduino to capture temperature data and display the data using the Sense HAT 8x8 display matrix.

**Suggested Answer**

Arduino Code (C++):

int sensorPin = 0;

void setup()

{

Serial.begin(9600);

}

void loop()

{

//getting the voltage reading from the temperature sensor

int reading = analogRead(sensorPin);

// converting that reading to voltage, for 3.3v Arduino use 3.3

float voltage = reading \* 5.0;

voltage /= 1024.0;

float temperatureC = (voltage - 0.5) \* 100 ;

Serial.print(temperatureC); Serial.println(" degrees C");

delay(1000);

}

RPI Code (Python):

from sense\_hat import SenseHat

import serial

ser = serial.Serial('/dev/ttyUSB0')

ser.flushInput()

sense = SenseHat()

while True:

while ser.in\_waiting:

x = ser.readline().strip()

line = x.decode('ascii')

sense.show\_messages(line)