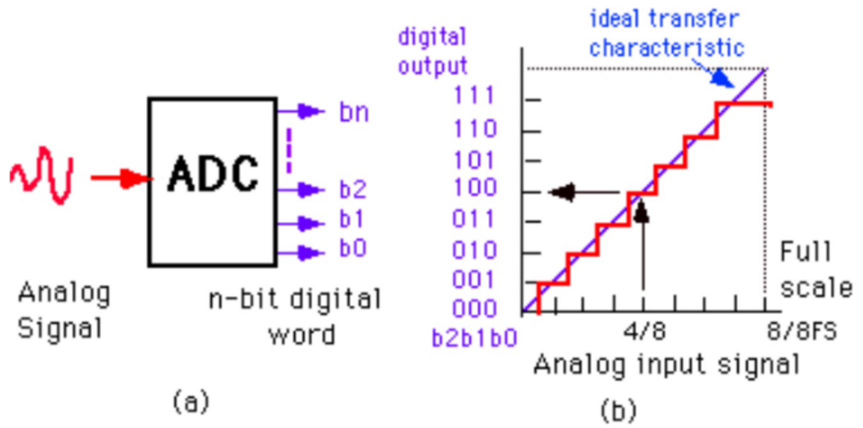
ANALOG-TO-DIGITAL CONVERSION WITH RASPBERRY PI AND ARDUINO

1. INTRODUCTION

In spite of everything the Raspberry Pi can do, there are still some fundamental functions that the Raspberry Pi can't do. One of which is reading in an analog signal. What is an analog signal? An analog signal simply refers to a signal that is somewhere between 0V and the maximum voltage the pins on the Raspberry Pi can take, 5V in this case.

Reading analog signals may be necessary in order to use a given sensor, or for a given application of the Raspberry Pi. For this reason, we use an Arduino to measure our analog inputs and convert them into digital values that can be read by the Raspbery Pi via serial communication.

The Arduino has an onboard analog to digital converter with a 10-bit resolution. What this means is that the Arduino can take a voltage somewhere between 0 – 5V and output a value between 0 and 1023 representing the voltage it received. To calculate the analog voltage sent to the Arduino, we multiply the value received from the Arduino by 5/1024. While the value we calculate will not always be exactly what we had as our input, at most it will be off by 5/2048, or 0.00244V, which should be accurate enough for our uses.



2. Installation and Setup

a. Download and install pySerial version 3.4 directly on your Raspberry Pi:

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

b. Download Arduino IDE by typing the following into the terminal:

```
sudo apt-get install arduino
```

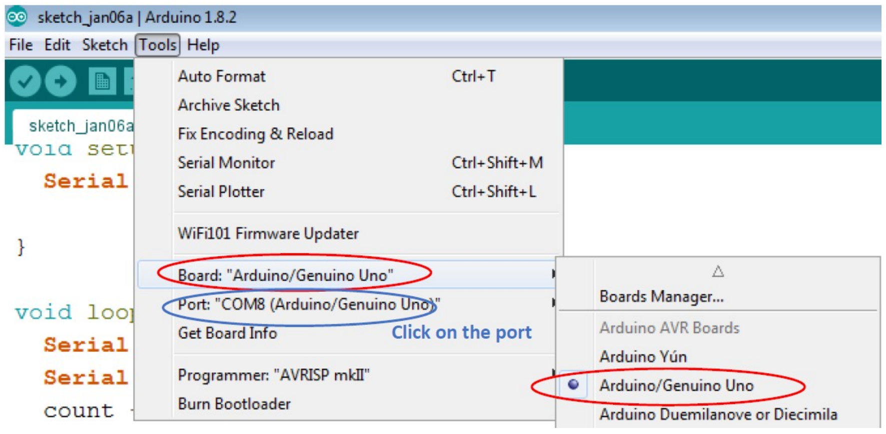
To open the IDE, find it in the menus and right click to add it to the desktop icons or type `arduino` into the terminal.

Note:

- Be sure to connect the RPi to the Arduino using the USB cable provided
- The two devices are able to exchange serial data over this connection

3. Testing The Arduino

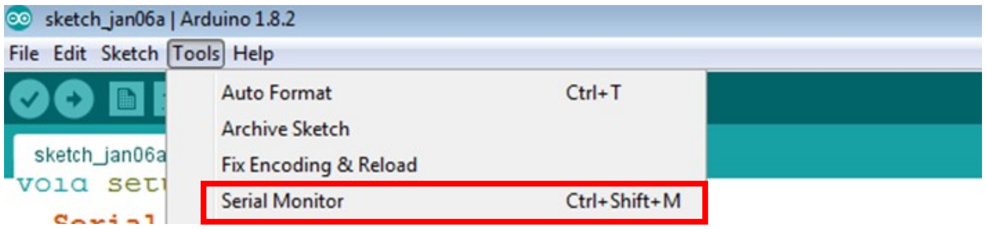
a. Make sure you check which type of Arduino and serial port is used:


sketch_jan06a | Arduino 1.8.2
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM8 (Arduino/Genuino Uno)"
Get Board Info
Click on the port
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Arduino Yún
Arduino/Genuino Uno
Arduino Duemilanove or Diecimila

b. Create a new Arduino sketch and write an Arduino program that will regularly send serial data to the RPi. Remember to compile and upload the program. Use following code:

```
int count=0;
void setup(){
Serial.begin(9600); //bits per second
}
void loop(){
Serial.print("The current count is: ");
Serial.println(count);
count += 1;
delay(1000);
}
```

c. Upload the program and open the serial monitor. A new window should open with the printed messages.


sketch_jan06a | Arduino 1.8.2
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
sketch_jan06a

4. Receiving Data from The Arduino Using Python

Create a new Python file and write a program to read from the Arduino port:

```
import serial
arduinoData = serial.Serial('/dev/ttyACM0',9600)
i=0
while i<20:
      if arduinoData.inWaiting()>0:
            serialData= arduinoData.readline()
            print(str(serialData)[2:-5])
            i = i+1
```

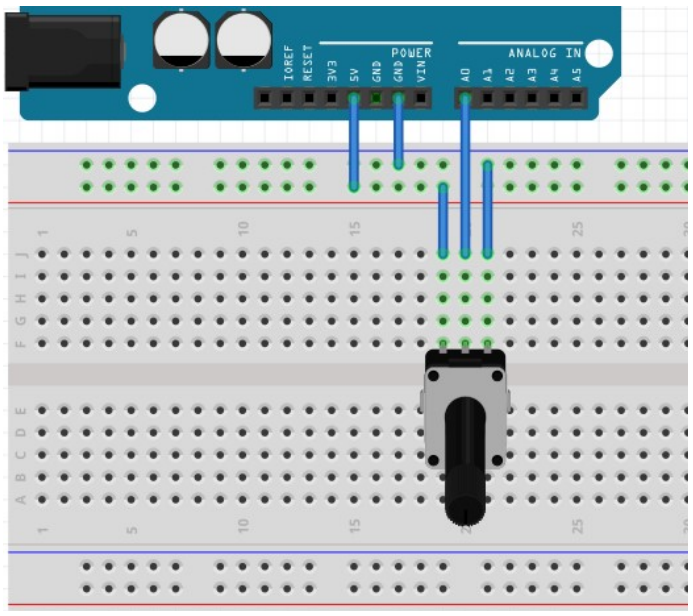
When you run the program, a Python shell should open up with the output from the Arduino.

Hints:

- Be sure to close the Serial Monitor window in order to allow Python to use the serial communication line.
- Make sure that the baud rate and COM port are identical to the one in the Arduino IDE (9600).
- If you are unsure which COM port is being used, go to the Arduino IDE and click **Tools > Port** to check. (Use single quotation marks for the COM)

5. Analog Reading Through Arduino

Connect a potentiometer to the Arduino as shown below:



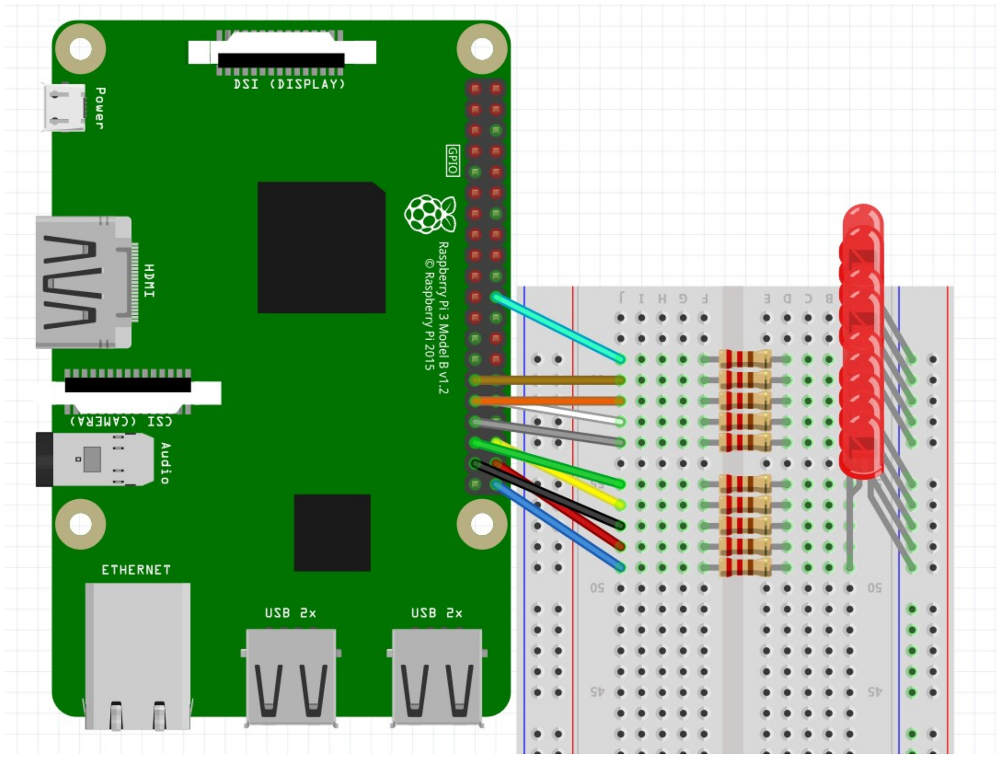
In a new **Sketch file** called "sensorRead.ino", write an Arduino program that reads the potentiometer values and sends them to the Raspberry Pi:

```
int sensor = A0; //declare pin A0 where potentiometer is connected
void setup(){
pinMode(sensor, INPUT); //set potentiometer as input
Serial.begin(9600); //bits per second
}
void loop(){
Serial.println(analogRead(sensor));
delay(200);
}
```

6. Mini-Project: Visual Indicator Using LED's

Using the circuit above and the starter code and LED circuit below write a program that reads the ADC values from the serial port and represents them on 10 LEDs. E.g: 10 LEDs are on when the read value is 1023, 9 LEDs are on when the read value is at least 920, etc. Make sure to wire the appropriate circuit as well.

LED Circuit



Starter Code

```python
import serial
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
#List of channels that are used as output
chan_list = [40,22,29,31,32,33,35,36,37,38]
#Set all channels in chan_list as output
GPIO.setup(chan_list,GPIO.OUT)
arduinoData = serial.Serial('/dev/ttyACM0',9600)

while True:
    if arduinoData.inWaiting() > 0:
        serialData = arduinoData.readline()
        readIn = int(str(serialData)[2:-5])
```