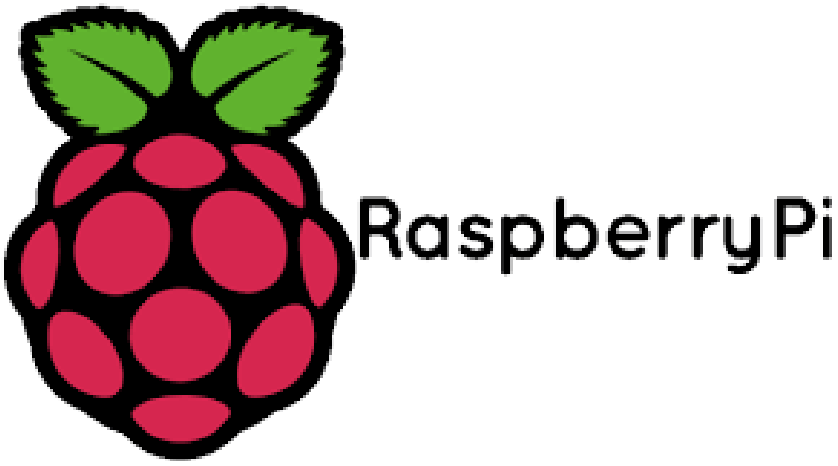
Birzeit University

Electrical and Computer Engineering Department

Interfacing Techniques Lab

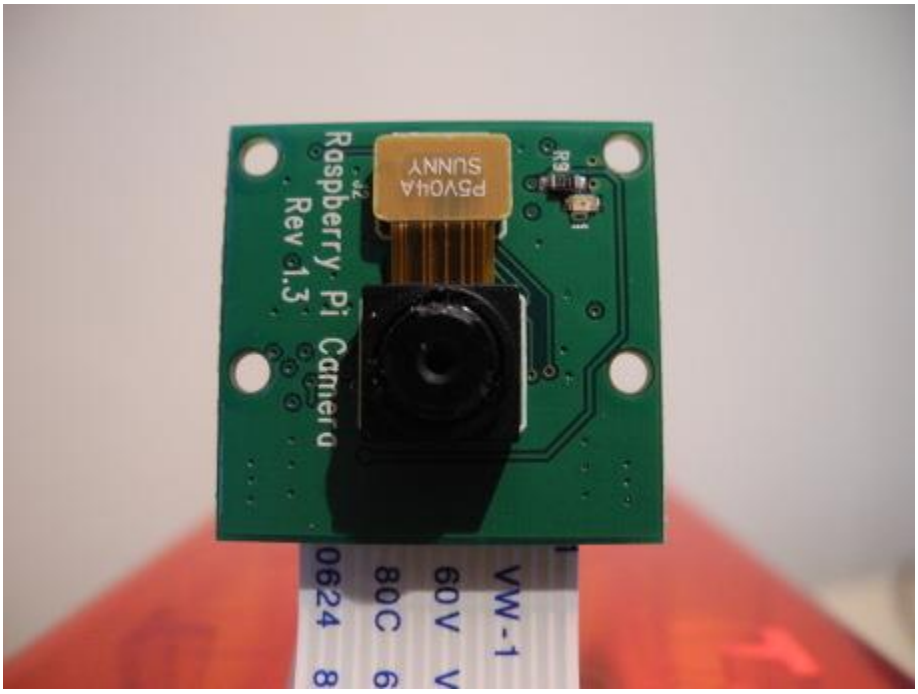
Experiment No. 8



Raspberry pi camera

Since 2012, the Raspberry Pi Foundation had been reporting that an official camera module was in development. In May 2013, an announcement was made by RS Components and Premier Farnell/Element 14, distribution partners of Raspberry Pi, that the camera module was available.

The camera consists of a small (25mm by 20mm by 9mm) circuit board, which connects to the Raspberry Pi's Camera Serial Interface (CSI) bus connector via a flexible ribbon cable. The camera's image sensor has a native resolution of five megapixels and has a fixed focus lens. The software for the camera supports full resolution still images up to 2592x1944 and video resolutions of 1080p30, 720p60 and 640x480p60/90. The camera module is shown below:



The "Pi NoIR" version of the camera was released on the 28th of October 2013. It has the same sensor with the IR filter removed, and a black PCB. With no IR filter, it can see near-IR wavelengths (700 - 1000 nm) like a security camera, with the tradeoff of poor color rendition. It is otherwise the same and uses the same software as the normal Pi camera.

Software

The following software configuration is used for Raspberry pi camera.

Since its inception, the camera has been supported in the latest version of Raspbian, the preferred operating system for Raspberry Pi. The first step is to get the latest Raspberry Pi firmware, which supports the camera. You can do that from a console by running:

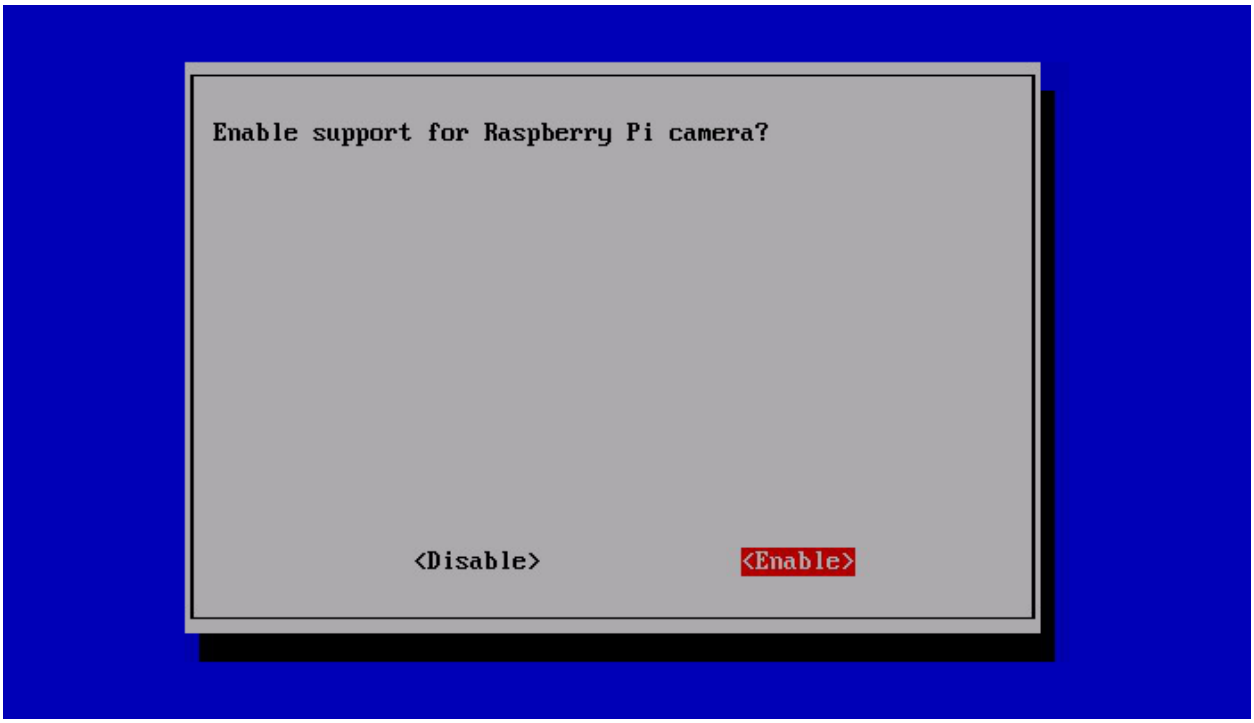
```
sudo apt-get update

sudo apt-get upgrade
```

You then need to enable the camera from the Raspberry Pi configuration program by running:

```
sudo raspi-config
```

Choose "camera" from the program and then select "Enable support for Raspberry Pi camera". You should then reboot when prompted by the raspi-config program. The camera will be enabled on subsequent boots of the Raspberry Pi.

Select “Enable” and press “Enter”.



Several applications should now be available for the camera: the rapistill program captures images, raspivid captures videos, and raspiyuv takes uncompressed YUV format images. These are command line program. They accept a number of options, which are documented if you run the commands without options.

Capturing images:

To capture images using the Raspberry pi camera you need to use “rapistill” command.

Open lxterminal and type rapistill which wil show a lot of options on how to configure the picture that you want to capture(size, quality, background lights…).

To capture a simple image just type the following command:

```
rapistill –o “name of image”.jpg
```

Recording videos:

In the same way you can record a video with the command “raspivid”, which will show the option of the videos.

For example on how to take a 10 second of HD video, use the following command:

```
raspivid –o “video name”.h264 –t 10000
```

This will take video for 10 second (10000 ms) in the formate of h264.

You will find the images and video saved in the file browser.

Web camera

In this experiment we will use a web camera instead of the raspberry pi camera since it is cheaper and it’s good for our goal here.

In order to take a picture using Raspberry pi and web camera we have to connect all the relevant components as the previous experiment including the web camera this time.

Capturing images and videos:

We need to install libraries for the web camera in order to use its commands and tools, which can be done easily by using the following commands:

Still images library:

```
sudo apt-get install fswebcam
```

video library:

```
sudo apt-get install ffmpeg
```

video player:

```
sudo apt-get install mplayer
```

OpenCV library

OpenCV is an open source computer vision library originally developed by Intel. It is free for commercial and research use under a BSD license. The library is cross-platform, and runs on Mac OS X, Windows and Linux. It focuses mainly towards real-time image processing, as such, if it finds Intel's Integrated

Performance Primitives on the system, it will use these commercial optimized routines to accelerate itself.

This implementation is not a complete port of OpenCV. Currently, this library supports:

- real-time capture
- video file import
- basic image treatment (brightness, contrast, threshold, …)
- object detection (face, body, …)
- bubles detection

OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are now full interfaces in Python, Java and MATLAB/OCTAVE (as of version 2.5). The API for these interfaces can be found in the online documentation. Wrappers in other languages such as C#, Perl, Ch, and Ruby have been developed to encourage adoption by a wider audience.

Installing OpenCV on a Raspberry Pi:

```
sudo apt-get install libopencv-dev python-opencv
```

Raspberry pi with Arduino

Many projects need a lot of input and outputs, digital and analogue in order to use analogue sensor or keypad…, to do that we may use Arduino with raspberry pi.

Install Arduino IDE on raspberry pi:

```
sudo apt-get install  Arduino
```

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

Procedure:

Connect the camera to the raspberry pi and connect to the computer using ethernet cable the power it.

Open LXterminal and type the following order:

1- Take pictures using the following command:

```
fswebcam -d /dev/video0 -r 640x480 pic.jpeg
```

note:

the captured images and videos will be saved in the following direction:

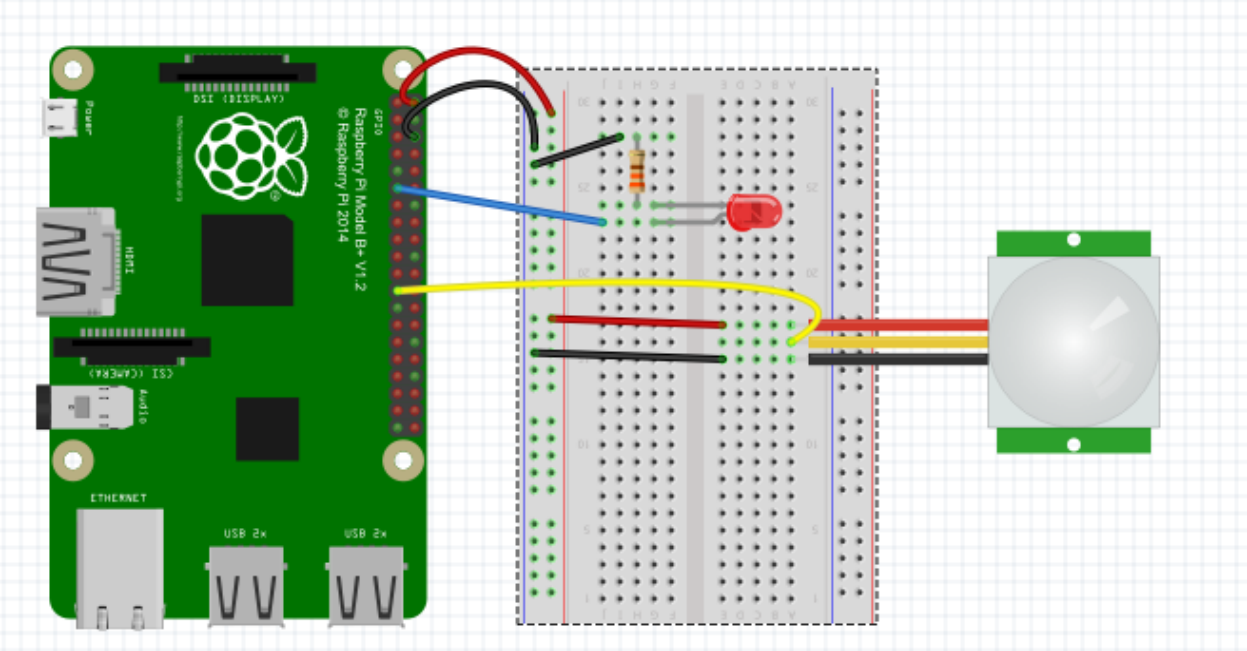
/home/pi/

Smart camera

Understanding the previous experiment and today's experiment, will allow you to make a simple, smart camera using PIR sensor and the web camera.

Your task is to write a code that will make the web camera take picture only if a motion is sensed from the PIR sensor.

Connect the following circuit:



Writing the code:

1- Create new folder

```
mkdir “folder name”
```

2- Enter the folder

```
cd “folder name”
```

3- Open the editor

```
sudo nano “program name”.py
```

Complete the following code:

Note:

Import os: is used for calling the command of capturing pictures for the webcam and to right a command in the program os.system(commamd):

```
Os.system(“”)
```

```
import time
import RPi.GPIO as GPIO
import os
GPIO.setmode(GPIO. )


while True:
    if (  ):
    os.system(" ")
    else:

    time.sleep(1)
```

Save and run the program as you learned in the previuos experiment.

TimeLapse Camera:

Time-lapse photography is a technique whereby the frequency at which film frames are captured (the frame rate) is much lower than that used to view the sequence. When played at normal speed, time appears to be moving faster and thus lapsing. For example, an image of a scene may be captured once every second, then played back at 30 frames per second; the result is an apparent 30 times speed increase. Time-lapse photography can be considered the opposite of high speed photography or slow motion.

Simply you can use it to show the life time of a living being or the process of building something in just few minutes.

Timelapse in raspberry pi can be used by fswebcam tool which will be shown in the following steps.

Writing the code:

1- Create new folder

```
mkdir “folder name”
```

2- Enter the folder

```
cd “folder name”
```

3- Open the editor

```
sudo nano “program name”.py
```

4- Write the following code:

```
import os
import time
import sys

FRAMES = 30 #is the number of the wanted pictures in 5 minutes which is 5*6=30


frameCount = 0
while frameCount < FRAMES:
    imageNumber = str(frameCount).zfill(4)
    #str: is for converting numbers to a string and zfill is to make it 4 digits.

    os.system("fswebcam -d /dev/video0 -r 640x480 /home/pi/New/image%s.jpg"%(imageNumber))
    #os.system to execute the command between branches.

    frameCount += 1
    time.sleep(10) #Takes roughly 10 seconds to take a picture
sys.exit()

```

5- save and run the code.

Note:

1- The raspberry pi will start to take pictures every 10 second for 5 minutes, and the picture will be saved in the folder that you have created. Now we need to make a vedio using the capture images using a tool that is called “MEncoder” which is installed on your raspberry pi.
2- MEncoder is an open source program that can build video in different formats and sounds.

Making timelapse:

1- Create a txt file with all the captured picture using the following command:

```
$ls *.jpg > list.txt
```

2- Type the following command to make the video from the captured images:

```
mencoder -nosound -ovc lavc -lavcopts vcodec=mpeg4:aspect=16/9:vbitrate=8000000 -vf
scale=640:480 -o timelapse.avi -mf type=jpeg:fps=24 mf://@list.txt
```

3- type the following command to play the video with omxplayer:

```
Omxplayer videoName.avi
```

Autostart python programs after login

To add a script that you want to run after the boot you need to change the file of /etc/profile using the following code:

```
Sudo nano /etc/profile
```

then add timelapse to the file:

```
Sudo python  /home/pi/ timelapse /pythonprogrogram.py
```

Color Detection:

openCV will be used to detect colors using the webcam, the color that will be detected is blue starting from(50, 50, 110) to(130, 255, 255).

Writing the code:

1- Create new folder

```
mkdir “folder name”
```

2- Enter the folder

```
cd “folder name”
```

3- Open the editor

```
sudo nano “program name”.py
```

4- Write the following code:

```python
import cv2
import numpy as np

cap = cv2.VideoCapture(0)

while(1):

    # Take each frame
    _, frame = cap.read()

    # Convert BGR to HSV
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    # define range of blue color in HSV
    lower_blue = np.array([110, 50, 50], dtype=np.uint8)
    upper_blue = np.array([130,255,255], dtype=np.uint8)

    # Threshold the HSV image to get only blue colors
    mask = cv2.inRange(hsv, lower_blue, upper_blue)

    # Bitwise-AND mask and original image
    res = cv2.bitwise_and(frame,frame, mask= mask)

    cv2.imshow('frame',frame)
    cv2.imshow('mask',mask)
    cv2.imshow('res',res)

    k = cv2.waitKey(5) & 0xFF
    if k == 27:
        break

cv2.destroyAllWindows()
```

5- Save and run the program:
   3 windows will open now(original picture, the mask and the results)
6- Make a test on blue objects and notice the results.

Face detection:

Face detection is one of the exciting application that can be done by raspberry pi and openCV, to run the face detection program enter the following folder:

```
cd /home/pi/face
```

you will find 2 files “face.xml” and “facedetect.py”

the run the program:

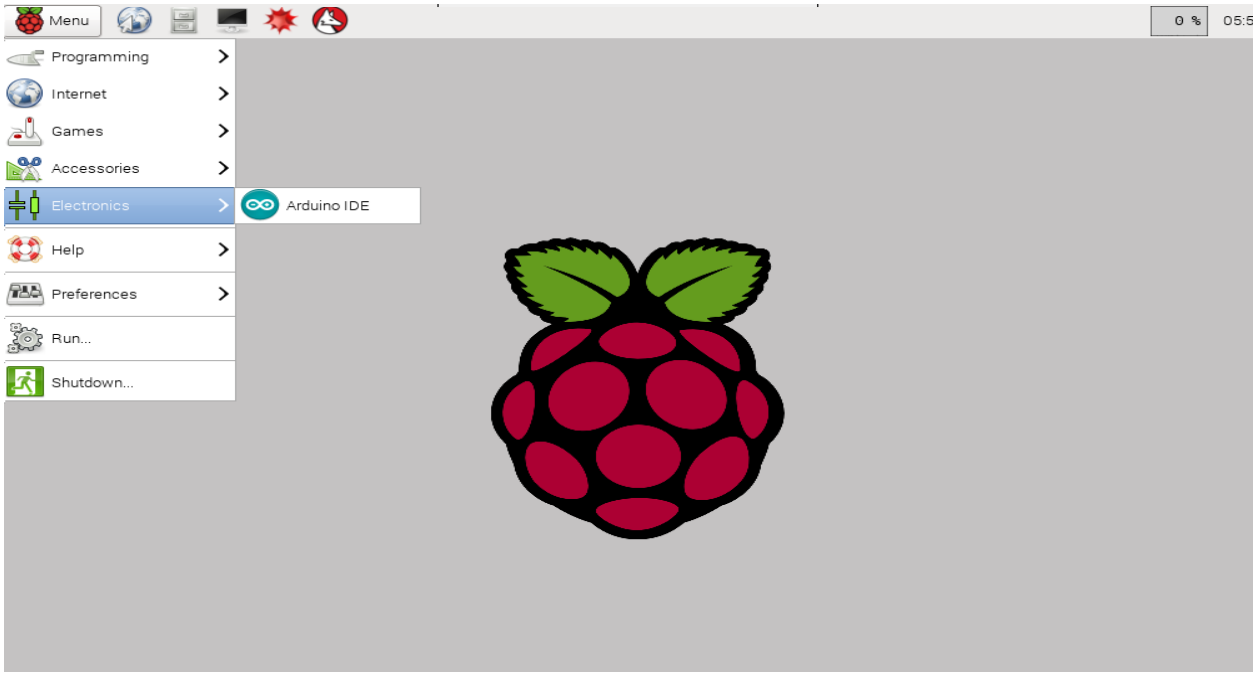
```
python facedetect.py –cascade=face.xml   0
```

Do not forget the zero(0) at the end of the command.

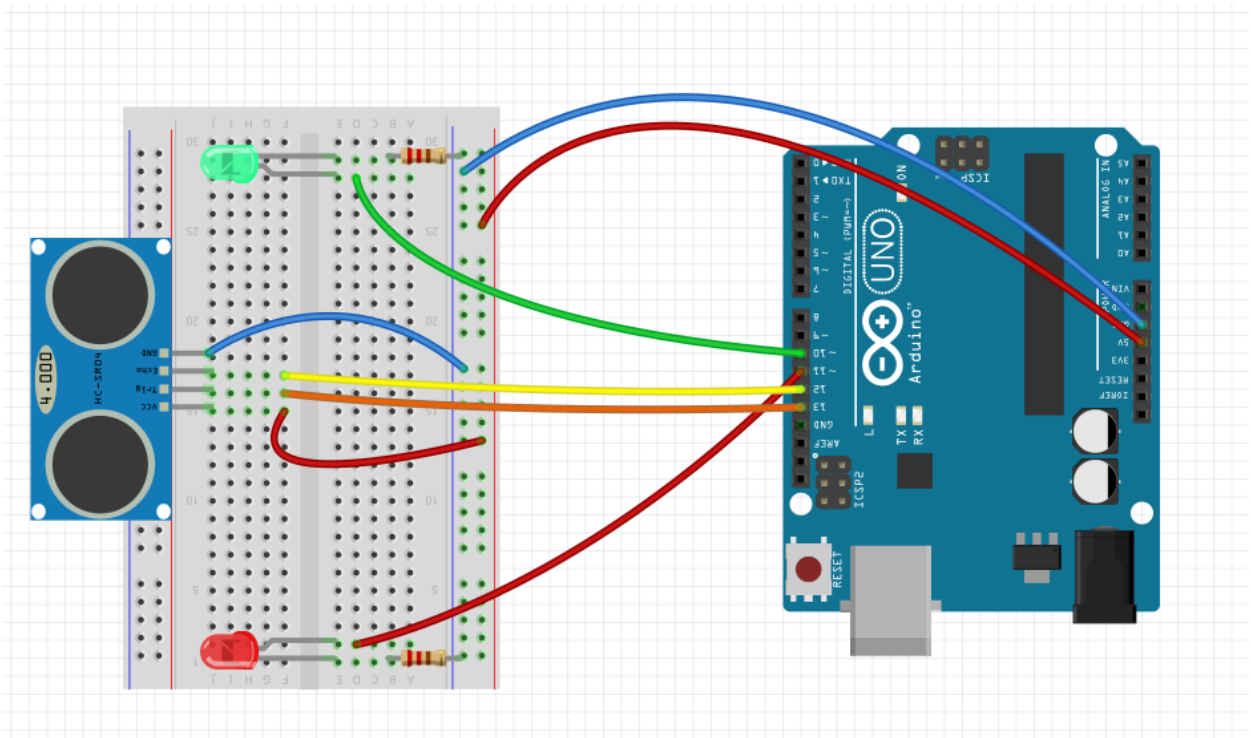
The Interface will be shown to enable you to detect faces.

Arduino with UltraSonic:

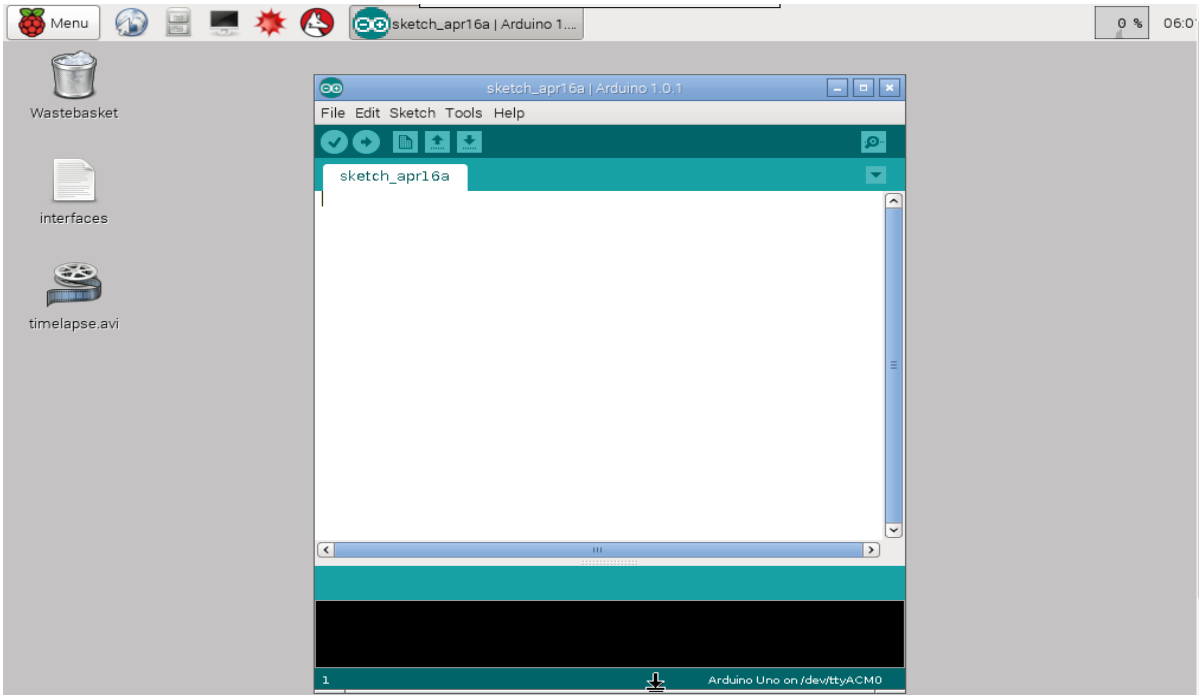
Go to electronics from tools menu to open Arduino IDE:



Connect the following circuit:



Write the following code in the Arduino IDE:




```
/*
 HC-SR04 Ping distance sensor]
 VCC to arduino 5v GND to arduino GND
 Echo to Arduino pin 13 Trig to Arduino pin 12
 Red POS to Arduino pin 11
 Green POS to Arduino pin 10
 220 ohm resistor to both LED NEG and GRD power rail
 */

#define trigPin 13
#define echoPin 12
#define led 11
#define led2 10

void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(led, OUTPUT);
```

```
  pinMode(led2, OUTPUT);
}

void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;

  if (distance >= 200 || distance <= 0){
        Serial.println("Out of range");
        digitalWrite(led,HIGH);
        digitalWrite(led2,LOW);
  }
  else {
    Serial.print(distance);
    Serial.println(" cm");
        digitalWrite(led,LOW);
    digitalWrite(led2,HIGH);
  }
  delay(500);
}
```

Then load the code to Arduino and run it, open serial to see the results.