Lab 6:- **DATA LOGGER**

## OBJECTIVE:[¶](http://localhost:8888/notebooks/Desktop/Important%20Documents%20/IoT%20final/Combined/Lab%206/Lab%206%20(FData%20Logger).ipynb#OBJECTIVE:)

This simple data logger takes regular light measurements with an analogue LDR (Photoresistor) and stores them in a text file on your Raspberry Pi. This data logger will measure and record the light level every second, enabling you to monitor how the brightness changes over a length of time. Later we send the data to ThingSpeak cloud platform and visualize the data.

## **COMPONENTS:**

1) RPi 3

2) Breadboard

3) Photoresistor x 1

4) 1µF Capacitor

5) Connecting Wires

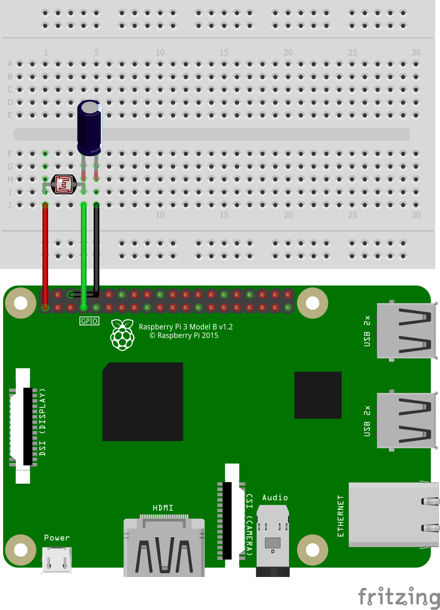
## **Light Dependent Resistor:**

The light dependent resistor or also known as the LDR sensor is the most important piece of equipment in our circuit . Without it we wouldn’t able to detect whether it is dark or light. In the light this sensor will have a resistance of only a few hundred ohms whilst in the dark it can have a resistance of several megohms.

resistance of several megohms. The capacitor in our circuit is there so we’re able to measure the resistance of the LDR sensor. A capacitor essentially acts like a battery charging up whilst receiving power and then discharging when no longer receiving power. Using this in series with the LDR we can work out how much resistance the LDR is giving out thus whether it is light or dark.

If we want to use analogue sensors with the Raspberry Pi, we would need to be able to measure the resistance of the sensor. Unlike the Arduino, the Raspberry Pi's GPIO pins are unable to measure resistance and can only sense if the voltage supplied to them is above a certain voltage (approximately 2 volts). To overcome this issue, you could use an Analogue to Digital Converter (ADC), or you could use a relatively cheap capacitor instead.

## **CIRCUIT DIAGRAM:**



### **CIRCUIT DESCRIPTION:**[**¶**](http://localhost:8888/notebooks/Desktop/Important%20Documents%20/IoT%20final/Combined/Lab%206/Lab%206%20(FData%20Logger).ipynb#CIRCUIT-DESCRIPTION:)

1. BOARD PIN CONFIGURATION IS USED.
2. One end of LDR to 3.3v pin.
3. Other end of LDR to pin #7.
4. One end of Capacitor is connected to LDR.
5. Other end of capacitor to Ground ping

### **CREATING THE PROGRAM:**

1. Go to TERMINAL
2. Navigate to workspace folder (created in lab 1) by typing the command: $ cd workspace
3. Open this folder in your file system
4. Right click the mouse and choose new file and name it as lab3.py
5. Right click on the file and open with IDLE3. Make it default program.
6. Type the below code and save the file.
7. Go to terminal and run the program with the below command:

$ python lab5.py

Code: -

#Data logger stored on RPi

import RPi.GPIO as GPIO

import time

import datetime

ldr\_pin = 7

log\_interval = 1 #in seconds

file\_name = "data\_logger.txt"

open(file\_name, 'w').close()

GPIO.setmode(GPIO.BOARD)

def loop():

while True:

for x in range (1, 6):

if x == 5:

ldr = recharge\_time(ldr\_pin) #call recharge time method

print(ldr)

data\_to\_write = (str(datetime.datetime.now()) + " , " + str(ldr)) #format data to write

write\_to\_file(data\_to\_write, file\_name) #call write to file method

time.sleep(log\_interval)

def recharge\_time(ldr\_pin):

count = 0 #initialize counter to zero

GPIO.setup(ldr\_pin, GPIO.OUT) #initially set it to Output

GPIO.output(ldr\_pin, GPIO.LOW)

time.sleep(0.1)

GPIO.setup(ldr\_pin, GPIO.IN) #change the pin back to input

while (GPIO.input(ldr\_pin) == GPIO.LOW): #count until the pin goes high

count += 1

return count

def write\_to\_file(text, file):

f = open(file,'a') #open file with append option

f.write(text+'\n')

f.close()

if \_\_name\_\_ == '\_\_main\_\_':

try:

loop()

except KeyboardInterrupt:

GPIO.cleanup()

### EXPLANATION:

With reference to the above code:

Run the above code and see your terminal. You can see the count printed on the console (Figure 47). It is updated every second. Count value accumulates for the time ldr pin is LOW. Higher the light intensity, lower the count value and vice versa. Open "data\_logger.txt" which is your workspace folder. You can see data being logged with its respective timestamp (Figure 48).

