

Department of Computer Engineering Technology

Green House

**Course**: CET 4962 Applied Software Technology

**Section**: HD70

**Semester**: Spring 2019

**Instructor**: Prof. Danielle Safonte

**Final Project Lab Report**

**Student**: Cristhian Suriel

**Date**: 05/24/2019

# Table of Contents

***Section*  *Pages***

**Introduction ------------------------------------------------- 01**

**Procedure ------------------------------------------------- 01-06**

**Design ------------------------------------------------- 07**

**Program ------------------------------------------------- 08-11**

**Conclusion ------------------------------------------------- 11**

# Introduction

If there is one thing everyone knows is that plants need sunlight to grow strong, but they’re forced to grow at a steady pace given that sunlight is the only proper source of light that it’s good for them. As of recent years, a new source of light has been discovered that has the same effects as sunlight when it comes to plants and thus it’s been used in what you might call *Green Houses* across the world. But for those greenhouses that use this new source of light for plants find themselves only using this source of light. In this project a simple way of not only saving electricity, but also a way of allowing plants to receive the same kind of sunlight they’ve always been given when it’s available.

# Procedure

Note: All code referred to in this procedure is provided in the *Program* section of the lab report to avoid repetition. Please refer to the designated section for any required code.

**Preparation**

* The following are the components needed for this lab: A raspberry pi 3, an 32gb SD card, an HDMI, an TV with HDMI input, a keyboard, a mouse, headphones, 5V power supply for the raspberry pi 3 B, WIFI.
* To set up the SD card first connected to the computer and format it, then download NOOBS which is in the main raspberry website on the downloads section.
* After downloading NOOBS extract the zip file inside the SD card.
* Insert the SD card into the SD card slot of the raspberry pi 3.
* Proceed to connect all external cables with the power supply being last.
* The other end of the HDMI goes connected to the TV, and if no WIFI is available there’s always use for the Ethernet port and a router.
* Once the raspberry pi 3 B is booted up follow the instructions. (make sure to only install Raspbian in the beginning)
* Once all the instructions are followed the user is ready to use the OS.
* Once the user is connected to the WIFI go to the command prompt.
* Here type the following commands:

sudo apt-get update

sudo apt-get upgrade

* These two lines of codes will update and upgrade the raspberry pi to date.
* Building material such as cardboard. Make sure at least the material for the ceiling is light enough for a servo motor to control
* Gorilla glue, staples, and electrical tape
* Gather a 1uF capacitor, a 330 Ohms Resistor, a Light detector (LDR), 4 LEDs, and a servo motor

**Building the Green House**

* In this project the material used is cardboard so just get a empty cardboard box to rip apart



* From the box cut away three pieces, on for the base of the house, one for the walls of the house and one for the ceiling of the house
* Put the walls and the base of the house together (I recommend making the base on the longer side so that it sticks out from on side of the green house. This would be where the raspberry would be placed)
* Use a combination of gorilla glue with staples to hold them together.
* Punch holes on the side of the was that has the base sticking out, these punctures are meant for the 4 LED
* Stick each LED through the holes on the wall and bend the other end a little to keep them in place
* Next cut a square on the wall big enough to hold a bread board in pace, this part is unnecessary but very helpful for the next part of the project
* Next take the servo motor and attach it to the opposite wall of the breadboard
* It is important to attach the servo motor high enough so that the cardboard ceiling can be attached to it and that the angle is at either 0 degrees or 180 degrees
* To attach he ceiling to the servo motor you could either use screws or like in this case saw it together with a sawing kit
* Make sure the wires for the servo motor stick out of the house before the set up
* Solder together all the positive ends and negative ends of the LEDs sticking out of the green house (make sure there’s a way to connect to a breadboard)
* At this point your structure should look like this:

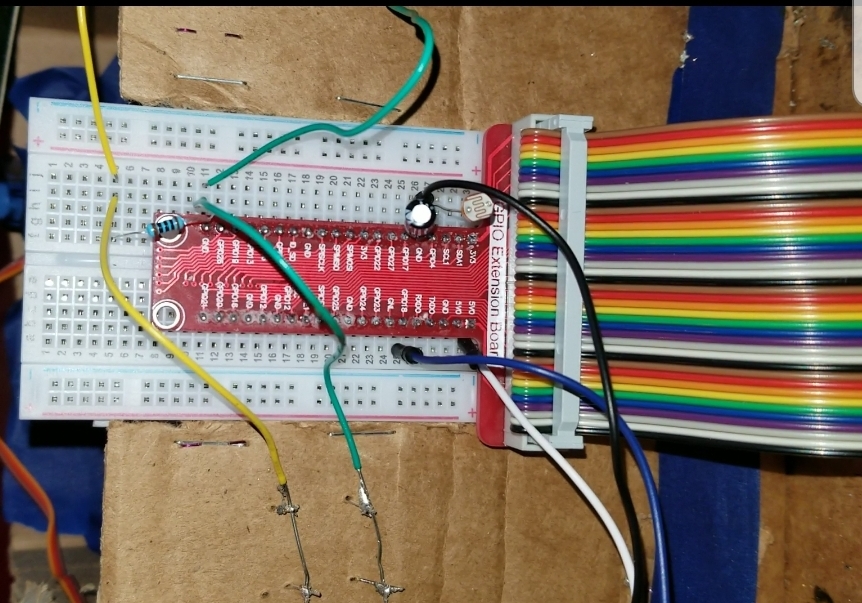


**Wiring of the green house**

* If available use a Rainbow wire Pin Out extension for the raspberry pi as it makes this part a lot easier
* Connect one of the of LDR to 5v and the other end to GPIO 4
* Connect the positive end of the 1uF capacitor to GPIO 4 and the negative end to GND
* It should look like so as of now:



* Next connect a 330 Ohms resistor to GPIO 26 and the other end to the positive side of the line of LEDs
* The negative side of the line of LEDs should be on GND
* Finally, the Servo motor’s orange wire should be connected to GPIO 18, while the red one to 5v and brown one to ground
* At this point the circuit should look more or less like this:

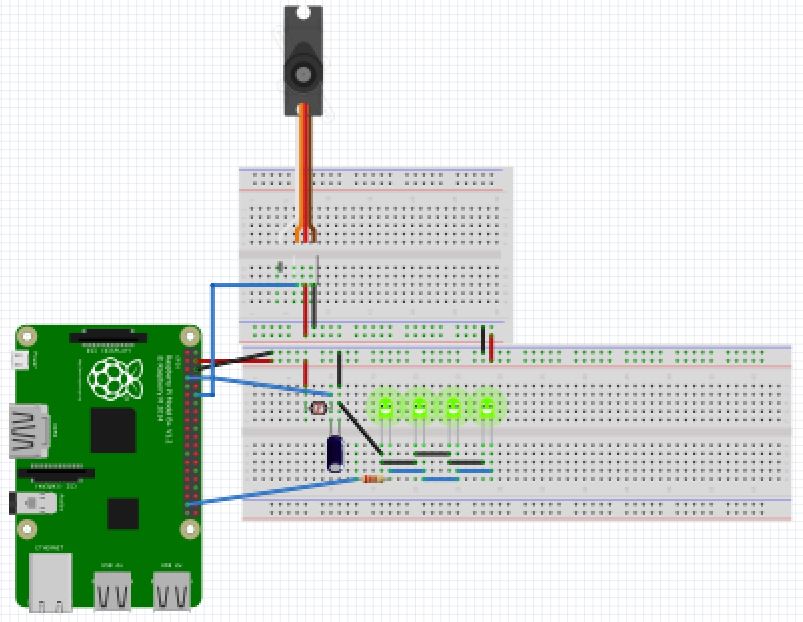


**Code of the green house**

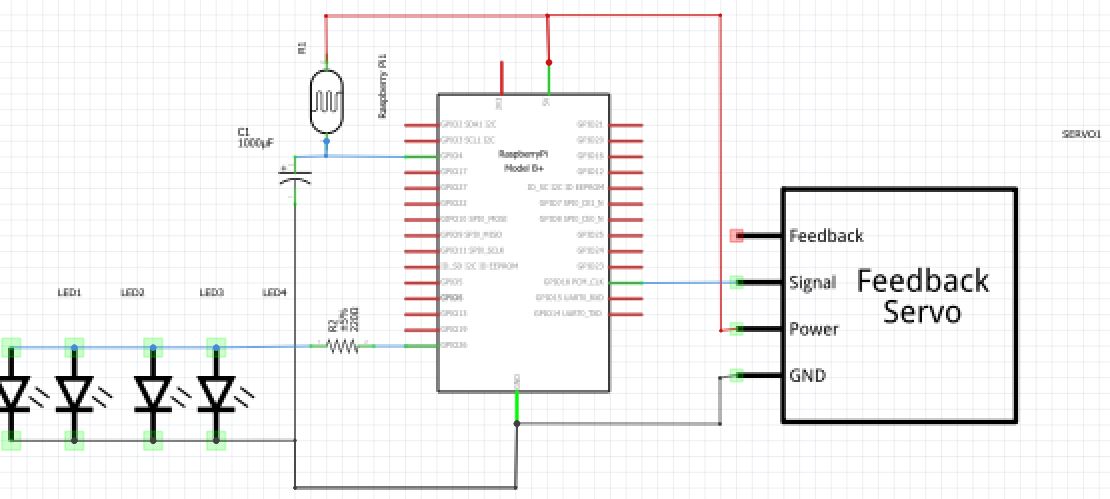
* First import the libraries needed for the code
* Those being the gpiozero library for the light sensor, the time library for the sleep command, and the RPi.GPIO for the raspberry commands
* Define the light sensor to GPIO 4 and set GPIO 18, and 26 as outputs
* Additionally, make sure to set the frequency of GPIO 18 to 50Hz
* Build a function that will allow for the duty cycle to change and make the parameter the angle
* Keep in mind that to calculate the duty cycle in this case would be the angle/18+2
* Create a while loop that runs forever and prints the LDR values as the program goes on
* Now add program to it so that the angle of the servo motor changes and the voltage across pin 26 as well so that when the LDR detects high light the ceiling opens, and the LEDs turn off, but when there’s not enough light the ceiling closes and the LEDS turn on
* Test the code to make sure it works
* To set the code to a more accurate result run it for about a week during sun set to see what the value is when the sun is setting down
* That value would be the cut off point for this project

# Design

**Wiring**



**Schematic**



# Program

**Code for the green house with sundown value**

from gpiozero import LightSensor

from time import sleep

import RPi.GPIO as GPIO

ldr = LightSensor(4)

GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)

GPIO.setup(18, GPIO.OUT)

GPIO.setup(26, GPIO.OUT)

pwm = GPIO.PWM(18, 50)

pwm.start(0)

#function to assign angle

def SetAngle(angle):

duty = angle/18+2 #this is to calculate the duty cycle

GPIO.output(18, True) #turns the pin for output

pwm.ChangeDutyCycle(duty) #changes the duty cycle to match what's calculated

sleep(1) #waits 1 second

GPIO.output(18, False) #turns the pin for output

# changes the duty back to 0 so we aren't continuously sending inputs to the servo

pwm.ChangeDutyCycle(0)

while True:

print(ldr.value)

if (ldr.value > 0.167):

GPIO.output(26,GPIO.LOW)

sleep(1)

SetAngle(0)

else:

SetAngle(180)

GPIO.output(26,GPIO.HIGH)

sleep(1)

**Logic Flowchart**

Yes

No

Open Ceiling

Turn inner LED off

Is there enough light?

Light sensor sensing Light

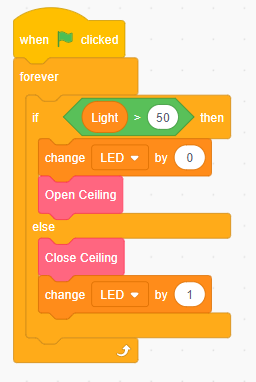
Turn inner LED on

Close Ceiling

Start

End

**Scratch Block Diagram**



# Conclusions

Although the LEDs are a poor substitute for the Light that’s supposed to be good for the plants as sunlight, this project gets the message across. I encountered a few errors here and there for starters I wanted to include more code involving matplotlib and even a wirelessly android controller but due to the servo motor reacting to the alternating signals every time the raspberry pie makes a change the project wouldn’t work. Another error I encountered with this lab is that I kept getting a warning that wouldn’t let me run the lab in this case all I had to do was add “GPIO.setwarnings(False)” to the code. These are some of the errors I encountered that taught me a lot about signals and the difficulties that would come with pulling off this project in a much bigger scale.