Garden Lighting System Report.

Intermediate Electricity and Electronics T2.

March 2021

Tutor: Brian Leslie.

2345063M.

2463524H.

2387527G.

2509621O.

Table of Contents.

Analysation of the Project Brief.

Inputs:

Process:

Outputs:

Basic:

Develop a system of **4 lights** for a garden which **turn on automatically when it is dark**, the user should be able to **adjust the threshold** this to occur. In addition to automatically **turning on when it is dark**, the lights must be able to be **turned on or off individually by manual control.**

Advanced:

Further a user activated Monday to Sunday **scheduler** should be able to **control all four lights** in hourly segments set by the user within a **custom schedule** (**time modulant**) set up by the user. The hierarchy of the lighting control is thus; manual control > timer control > dark sensing.

Interpretation of Task.

Our interpretation of the task showed that we had to create a garden light system that will automatically turn on when it gets dark at night, dimming gradually the darker the night gets, and turning off when it gets bright. On top of the main function of the garden light dimming, we had to integrate manual switches to each that can turn on or off each light individually according to the users need. Finally, our last task was to apply a schedular to the system that allows the user to set a time for each light depending on a certain day and time, which can be altered at any given point during the week.

Process.

As a group, before looking at any electronics or how the system may run, we wanted to carefully analysis the brief we had been given. We wanted to make the brief as simple as possible, so we created a specification with clear goals in it that we wanted to achieve.
Our next step was to try and understand how our system would function, we played with many ideas and made a block diagram to represent the principal functions needed in the system. After understanding these functions, we started to put together a flow chart to show an ideal workflow we would like the system to have. This flowchart allowed us to further conceptualize how our circuit could potentially look. With this conception in our head, we moved to the software Tinker Cad to start playing with some initial ideas. Tinker Cad allowed us to be able to quickly edit and share our circuits with each other. Tinker Cad also allowed us to establish and progress with our coding skills. With a combined effort we managed to create a simulation which was successful on ¾ goals we had in our specification; with this circuit we could recognise what components we would need to physically create this circuit. Our physical model had many successes and fails as we realised in the real world it can be quite unpredictable at times. We went back to our past work and updated our flowchart and block diagram to cohere with our new functional circuit.

Project Specifications.

1. 4 LEDs will automatically turn on when it turns dark.
2. The LEDs will be able to be individually turned on and off by the user.
3. The sensitivity of the LDR will be able to be altered by the user to suit their desired threshold.
4. The user will be able to adjust a 7-day schedular to control when the LEDs go on and at what hour.

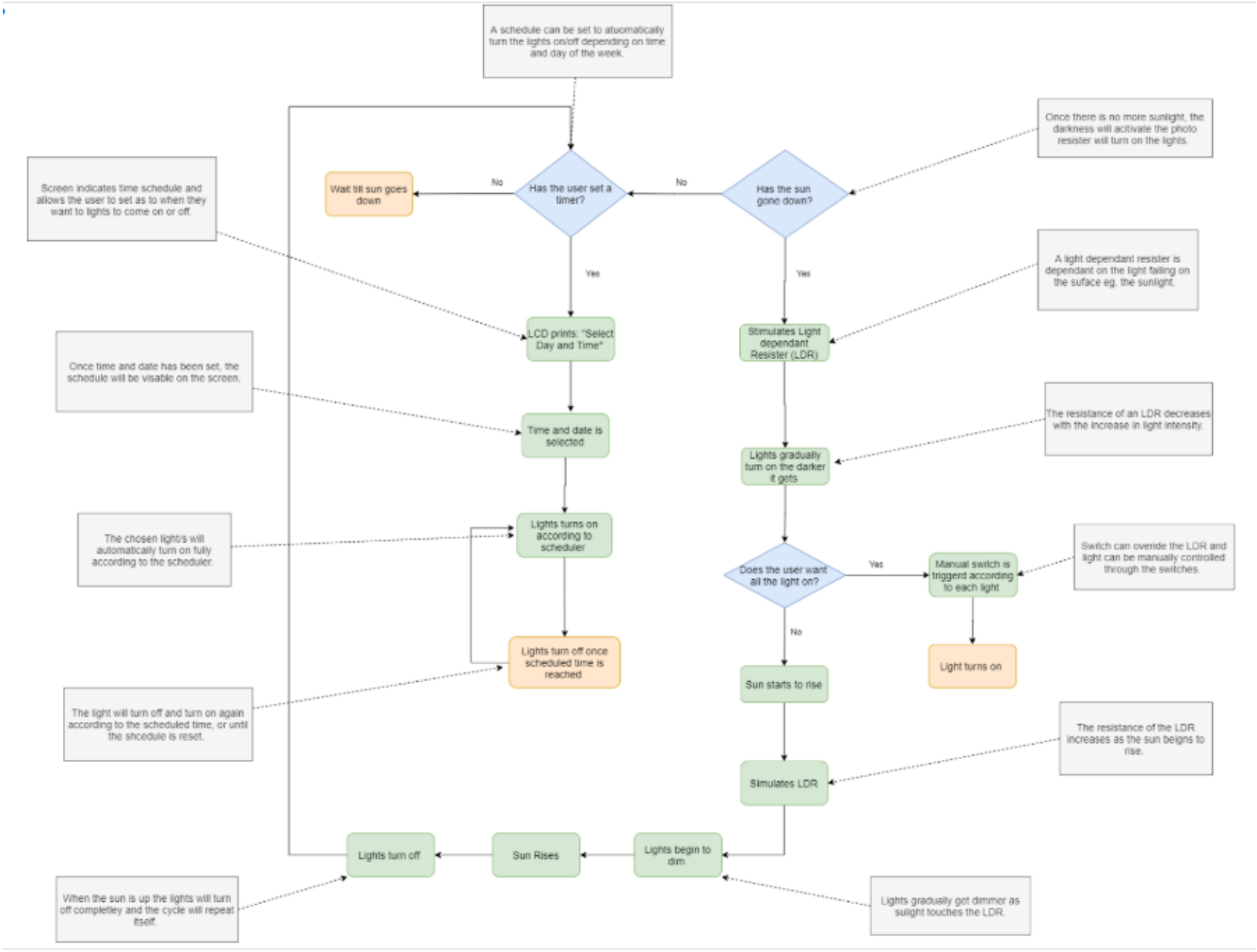
Flow Chart.

There are two main starting points for the garden light system. The first step is to allow the user to set a schedule according to time and day of the week they want the lights (LEDs) to automatically turn on. To do this the user will click the LCD screen where it will say select time and day, this is where the user can set the lights accordingly to come on when selected.

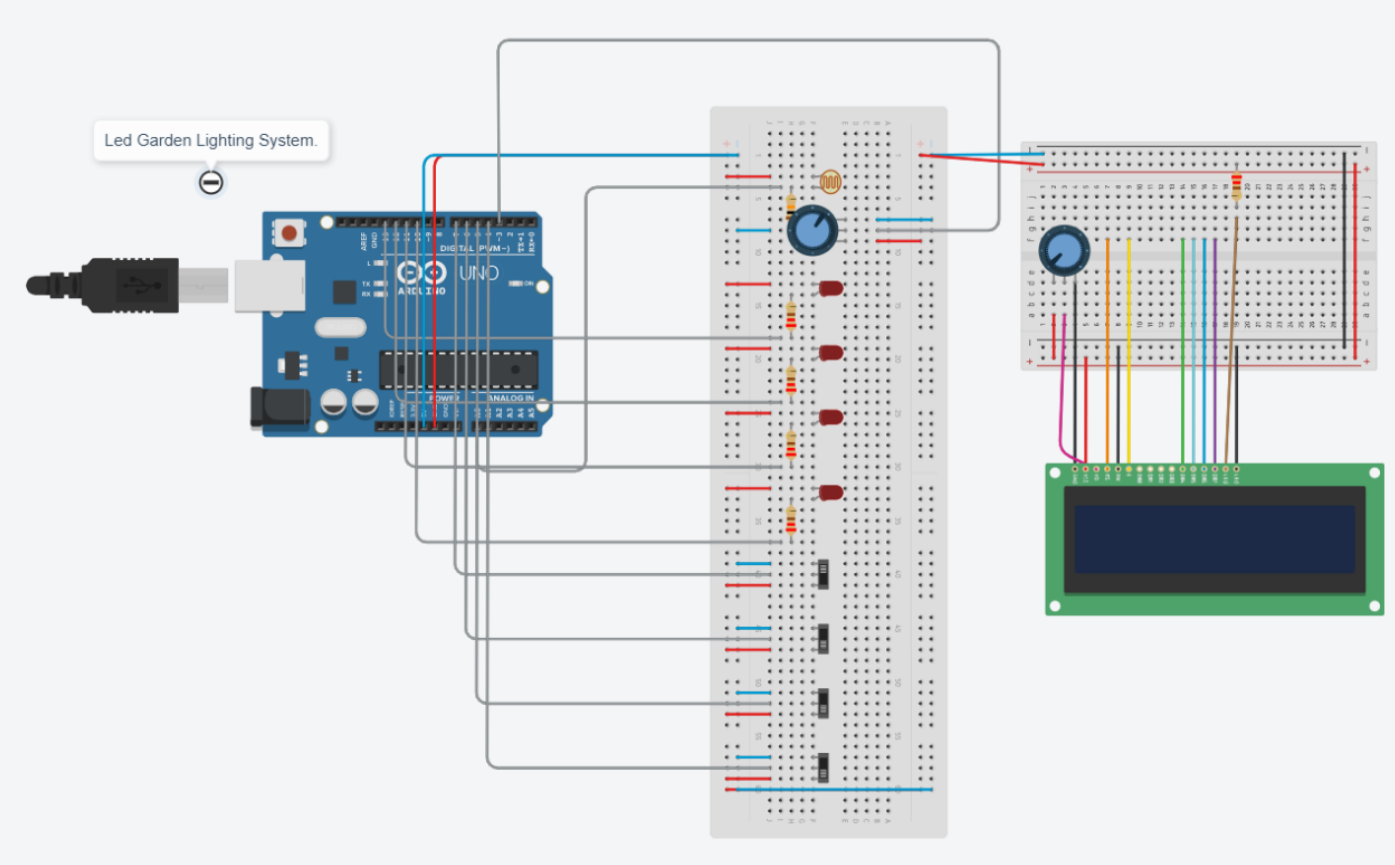
The other option available is to wait till dark, this is when the lights will automatically turn on. This system starts by stimulating the light dependant resistor (LDR) as it will detect the sun setting and the lights will gradually get brighter the darker the night gets. If the user does not want all four lights switched on, then there are four manual switches according to each light which can be flipped at any point, overriding the LDR in place. (This switch function can be used at any point allowing the user to switch the light on/off.)

The final process of the flow chart shows the LDR becoming resistant to the light. As the sun starts to rise the process will begin to reverse. The brighter it gets, the dimmer the lights will get until the light is completely off. Thus, repeating to process over again once the sun goes down.

Flowchart Diagram.



Schematic Diagram.



Bill of Materials.

- Arduino Uno R3
- Protoboard
- 4 Slide Switches
- Potentiometer
- Light Dependent Resistor
- 4 LEDs
- 4 220Ω Resistors (for LEDs)
- 1 10kΩ Resistor (for LDR)
- LCD Screen
- Input Touchpad/keypad

Inputs	Outputs
Arduino Uno R3.	4 LEDs.
Potentiometer.	LCD Screen.
4 220Ω Resistors.	
1 10kΩ Resistor.	
4 Slide Switches.	
Light Dependent Resistor (LDR).	
Input Touchpad / Keypad.	
	-

Block Diagram.

C++ Programming with Annotations.

```
void setup() //Variables get initialized here.

{

  pinMode(13, OUTPUT); // Initialize LED 1.
  pinMode(12, OUTPUT); // Initialize LED 2.
  pinMode(11, OUTPUT); // Initialize LED 3.
  pinMode(10, OUTPUT); // Initialize LED 4.
  pinMode(3, INPUT); // Initialize the Potentiometer.
  pinMode(A0,INPUT); // Initialize the LDR.
  pinMode(7, INPUT); // Initialize Slide Switch 1.
  pinMode(6, INPUT); // Initialize Slide Switch 2.
  pinMode(5, INPUT); // Initialize Slide Switch 3.
  pinMode(4, INPUT); // Initialize Slide Switch 4.

}

void loop() //Loops consecutively, allowing the program to change and respond.

{
   int pot = analogRead(3); // Potentiometer value
{
  pot = map(pot, 0, 1023, 0, 255);     //Map value 0-1023 to 0-255 (PWM)
  digitalWrite(A0, pot);               //Send PWM value to LDR
  delay(100);                          //Small delay
}


 if (digitalRead(7)) //If Slide Switch 1 is switched.
 {
  digitalWrite(13, HIGH); //LED 1 will turn HIGH.
 }
 if (digitalRead(!7)) //If Slide Switch 1 is not switched.
  {
  digitalWrite(13, LOW);  //LED 1 will turn LOW.
  }
  if (digitalRead(6)) //If Slide Switch 2 is switched.
  {
   digitalWrite(12, HIGH); //LED 2 will turn HIGH.
  }
  if (digitalRead(!6)) //If Slide Switch 2 is not switched.
  {
  digitalWrite(12, LOW);  //LED 2 will turn LOW.
  }
  if (digitalRead(5)) //If Slide Switch 3 is switched.
  {
  digitalWrite(11, HIGH); //LED 3 will turn HIGH.
   }
   if (digitalRead(!5)) //If Slide Switch 3 is not switched.
     {
   digitalWrite(11, LOW); //LED 3 will turn LOW.
   }
  if (digitalRead(4)) //If Slide Switch 4 is switched.
  {
   digitalWrite(10, HIGH); //LED 4 will turn HIGH.
   }
   if (digitalRead(!4)) //If Slide Switch 4 is not switched.
  {
   digitalWrite(10, LOW); //LED 4 will turn LOW.
   }


    int val=analogRead(A0); // LDR resistance value
    if (val>75) //If the LDR detects more than 75 resistance (the LUX intensity decreases (no sun light)).
  {
    digitalWrite(13,HIGH);  //LED 1 will turn HIGH.
    digitalWrite(12,HIGH);  //LED 2 will turn HIGH.
    digitalWrite(11,HIGH);  //LED 3 will turn HIGH.
    digitalWrite(10,HIGH);  //LED 4 will turn HIGH.

    Serial.println("Night mode, LED ON"); //This will displayed on the LCD screen.
  }
  if (val<50) //If the LDR detects less than 50 resistance (the LUX intensity increases(sunlight)).
  {
   digitalWrite(13,LOW);  //LED 1 will turn LOW.
   digitalWrite(12,LOW);  //LED 2 will turn LOW.
   digitalWrite(11,LOW);  //LED 3 will turn LOW.
   digitalWrite(10,LOW);  //LED 4 will turn LOW.

   Serial.println("Day mode, LED OFF"); //This will displayed on the LCD screen.
  }

}
```

Serial Monitor

Bibliography.

C,codebender, 2020. *How to Use Potentiometer - Arduino Tutorial. [online] Instructables.com. Available at: <https://www.instructables.com/How-to-use-Potentiometer-Arduino-Tutorial/> [Accessed 14 March 2021].*