Project 1 – Sunrise Alarm

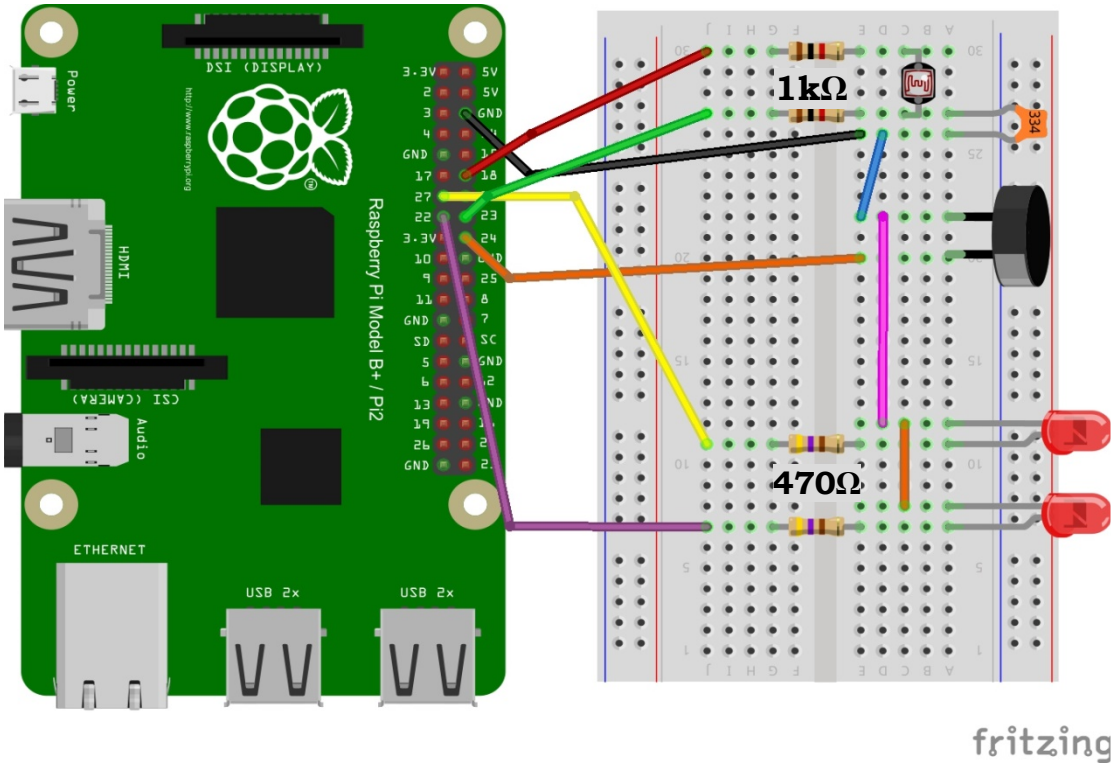
This project will allow you to build an alarm clock that will activate when the sun rises, using the Photoresistor to measure the light levels, and the buzzer and the two LEDs to try and wake you up!

You will need:

- Raspberry Pi (fully set up and connected to the internet)
- Breadboard
- Photoresistor
- 330nF Ceramic (bidirectional) Capacitor
- 2x Red LEDs
- Buzzer
- 2x 1kΩ Resistors
- 2x 470Ω Resistors
- 6x Male to Female Jumper Wires
- 3 x Male to Male Jumper Wires

Instructions:

1. Ensuring that the Raspberry Pi is turned off, make the circuit and connect it to the Raspberry Pi as shown in the diagram.



2. Turn on the Raspberry Pi and install the software using the following command in LXTerminal:
   `$ git clone https://github.com/henrybudden/rpesk-advanced`

3. Once the software has downloaded, change directory with the command:
   `$ cd rpesk-advanced`

4. Finally, run the code using the command:
   `$ sudo python 01_sunrise_alarm.py`

How to Use

By default, the alarm activates when the photoresistor's value is 50, which works well if used in a fairly light room, being activated by a torch shining on it (for testing). If you want to change this so that it is more accurate and will work properly, you can change the value on line 14 of the code. I would suggest a value of about 25-30 for accurate sunrise detection. To do this:

1. $ nano 01_sunrise_alarm.py
2. Use the arrow keys and keyboard to navigate the file and change the value.
3. Press Ctrl+X
4. Type 'y'
5. Press Enter

How it works

A Photoresistor is a variable resistor, meaning that its resistance changes depending on the light levels, unlike the other four resistors used, as their resistance values remain constant. When more light hits the component, its resistance decreases, and vice-versa. Therefore, the code is simply reading the resistance of the Photoresistor via some clever electronics that allow the Pi to convert this resistance into a voltage, and use a capacitor to turn this into pulses that can be measured by the Pi using a simple timing algorithm. This is needed as the Pi is incapable of measuring analogue signals. Finally, if the returned value exceeds the value set on line 14, the alarm is triggered!