Test Plan

And

Execution

Mike Minabian
04-29-2018

Table of Contents:

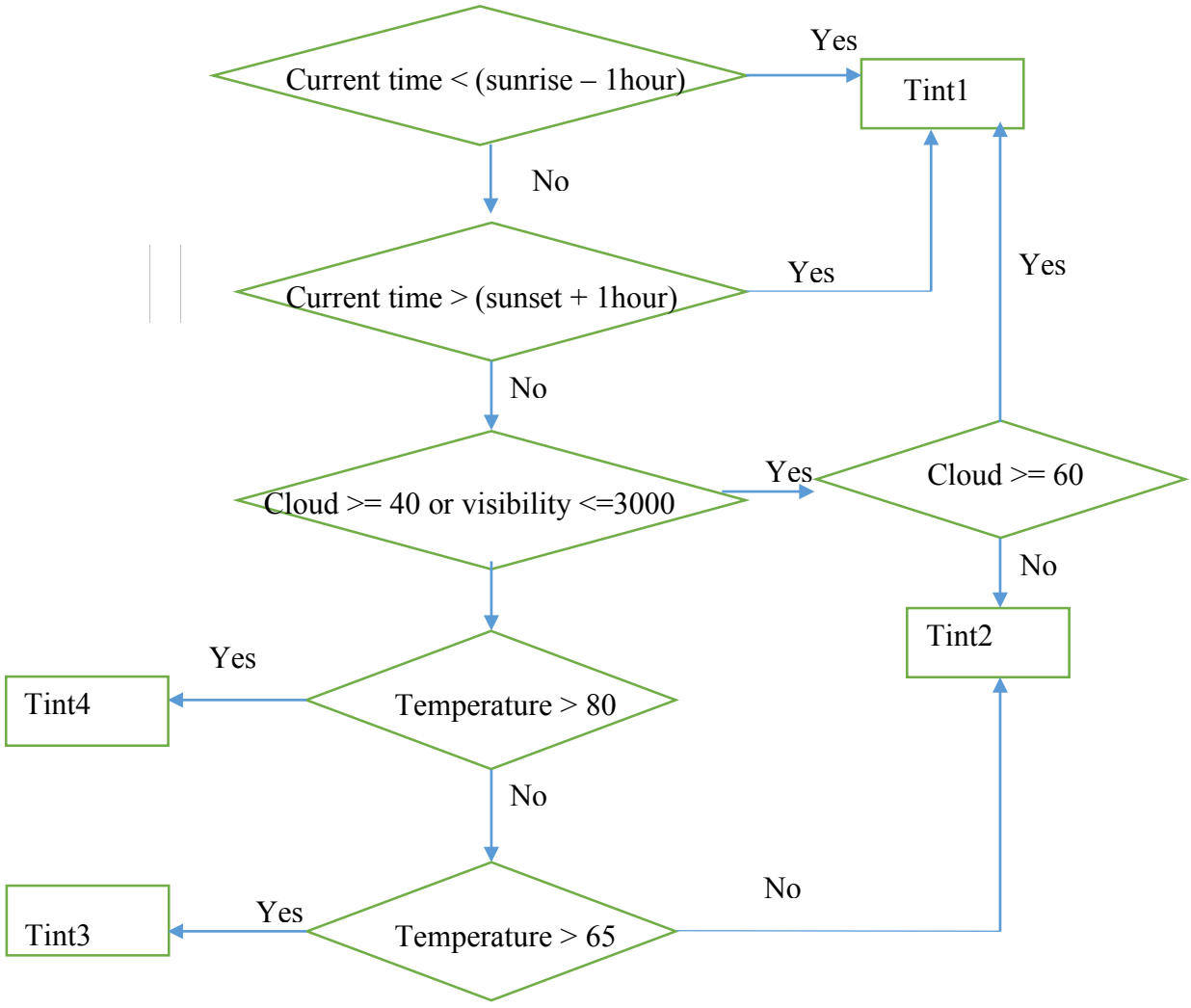
1- Introduction

Test kit is based on Raspberry Pi to emulate View Glass. It takes into account the current weather, visibility and temperature of the environment to determine the tint of the glass.

2- Feature Requirements

Test kit gets current weather condition including visibility, cloudiness, sunset and sunrise time for zip code 94085 through on API call and temperature of the environment using a local sensor reading. Then based on the collected data it determines the glass tint level. There are four glass tint levels which use to change the color of LED light on Raspberry Pi.

The tint level gets selected based on the following condition:



Assumption: Local temperature sensor can read from -30 to 60 Celsius.

3- Test Set-up

- **Equipment**
  Raspberry Pi Test kit, HDMI Cable, Display, Keyboard, Network Cable, USB **C**harger, Temperature **C**hamber and Web Power Switch

- **Set-ups**

  Set-up 1

  1- Connect Raspberry Pi to a display using HDMI cable
  2- Connect Raspberry Pi to network
  3- Power Raspberry Pi using laptop USB port in the lab

  Set-up 2

  Like set-up 1 but should put Raspberry Pi in Temperature Chamber

  Set-up 3

  Like set-up 1 except that Raspberry Pi is powering up by a USB Charger which is plugged in Web Power Switch

4- Test Scenarios

4-1- Test temperature sensor

- Use Set-up 2
- Configure Temperature Chamber to start from -30 to 60 Celsius and Change 10 Celsius every hour.
- 30 minutes after starting Temperature Chamber, run a script on Raspberry Pi to get temperature and compare it with expected result.

4-2- Compare the LED tint to the applied tint

- Use Setup 1
- Run a script on Raspberry Pi to apply tint 1, tint 2, tint 3 and tint 4
- Script should verify the LED tint level after each tint

4-3- Test the behavior of Test kit in all possible conditions

- Use Set-up 1
  Possible conditions:
  - Current time is less than one hour before sunrise
  - Current time is greater than one hour after sunset
  - Current time is between one hour before sunrise and one hour after sunset:
    - Cloud >= 40 or visibility <= 3000:
      - Cloud >= 60
      - else
    - Cloud < 40 and visibility > 3000:
      - temperature >80
      - else temperature >65
      - else

To test all the above conditions, we need to feed the desire weather data for each condition to the weather.py script but weather script is using an API call to get weather condition and local temperature reading. So there is a need to change weather.py to get the desire data to test all the above conditions.

4-4- Stress Hardware against power cycle

- Use Set-up 3
- Configure Power Switch to turn ON plugged in port for 1 minute and then turn OFF the port for 10 seconds and continue running this configuration for 24 hours.
- Make sure that Raspberry Pi powers up and boots up

- Use a script to check if Raspberry Pi can tint the LED to tint 1, tint 2, Tint 3, tint 4

4-5- Stress the LED

- Use set-up 1
- Run a script on Raspberry Pi to change tint level to tint 1,tint 2,tint 3 and tint 4 for 10000 times and check if the LED tint level is same as the applied one, every time after applying the tint level.

4-6 Longevity Testing

- Use set-up 1
- Run weather.py
- Run a script to check continuously if the tint level of LED is the expected tint level based on weather condition from API call and local sensor temperature reading.

5- Execute Test Scenarios

All scripts have been placed under the /home/pi/ directory. After the Raspberry Pi has booted up, enable the eth0 interface, by running the "ifup eth0" command before running the scripts in the /home/pi/ directory as described in the following sections.

5-1 Execute Scenario 4-1

- Use Set-up 2
- Configure Temperature Chamber to start from -30 to 60 Celsius and change 10 Celsius every hour (-30,-20,-10,0,10,20,30,40,50,60).
  30 minutes after starting Temperature Chamber, run test_reading_temperature.py on Raspberry Pi. This script as default every hour checks the sensor temperature with the expected result and prints Pass or Fail.
  Finally, the script prints sensor temperature reading at -30,-20, -10, 0, 10, 20,30,40,50 and 60 Celsius.
  To determine time interval between temperature reading –delay (seconds) argument can be Used. For example, to check temperature every 30 minutes instead of every hour use python test_reading_temperature.py –delay 1800

5-2 Execute Scenario 4-2

- Use Setup 1
- Run test_led_tint.py on Raspberry Pi to apply tint 1, tint 2, tint 3 and tint4

I ran the script which applied tint1= [220,220,200], tint2= [180,180,200], tint3= [100,100,200] and tint4= [10, 10,200] but the LED did not tint to these tint level and the result was [216,220,200],[176,180,200], [96,100,200] and [8,8,200]. So there is a bug on LED,

5-3 Execute Scenario 4-3

To check the behavior of the Test kit in all possible conditions according to the requirement, I rewrote weather script to get weather condition as an argument instead of API call and local sensor temperature reading. Also, weather.py has a bug which is in tintMap dictionary for key 3 the value is tint2 instead of tint3 and this causes to tint to tint2 when it should tint to tint3.

In rewritten script which is weather3.py I fixed this bug. Also, I wrote test_weather3.py script to call weather3.py and feed the desired data for all conditions and verify the LED tint level.
Executed test_weather3.py for this Scenario.

5-4 Execute Scenario 4-4

- Use Set-up 3
- Configure Power Switch to turn ON plugged in port for 1 minute and then turn OFF the port for 10 seconds and continue running this configuration for 24 hours
- Make sure that Raspberry Pi powers up and boots up
- Make sure Ethernet port works
- Run script test_tint_led_by_appliable_tint_to_led.py

5-5 Execute Scenario 4-5

- Use set-up 1
- Run script stress_led.py on Raspberry Pi. This script gets number of cycle as an argument and in each cycle the script tints the LED to the four tint levels and checks if the LED tint is the applied level. If the

user does not use –cycle_count argument, the default number of cycle is 3. Finally, after running all cycles, the script prints number of failures and passes for every tint level.

- To run 10000 cycles use the following command on Raspberry Pi:
  python stress_led.py –cycle_count 10000

5-6- Execute Scenario 4-6

- Use set-up 1
- Run weather.py script in background using python weater.py&
- Run test_weather.py script. This script continuously checks the LED tint level based on weather data from the API call and local temperature sensor. The script writes the results in a file for each day and at the end of the day it prints how many times LED tinted to the expected tint level and how many times it did not. Every day, at the start of the day, the script will create a new file name as follows:
  Result_<new-day-timestamp>.text

6- Bugs found

Hardware bug

- LED for tint1 tints to [216,220,200] instead of [220,220,200]
- LED for tint2 tints to [176,180,200] instead of [180,220,200]
- LED for tint1 tints to [96,100,200] instead of [100,100,200]
- LED for tint1 tints to [8, 8,200] instead of [10, 10,200]

Software bug

- In weather.py, tintMap dictionary has value “tint2” instead of “tint3” for key “3”. tintMap={1:tint1, 2:tint2, 3 :tint2, 4:tint4}