piRover Builds with K2

piRover – Traffic Light Project Rev 1.3

Overview:

In this activity you will collaborate with a partner to use what you have learned from colorLED.py, blink.py, and user_blink.py to create a traffic light solution. You will research requirements by viewing a traffic light simulation and then code a solution that prompts the user to demonstrate either the North-South or the East-West traffic lights.

The requirements for the solution are listed below.

- Teams will research the traffic light simulation video.
- The user will see a welcome message indicating that this is the Traffic Light Simulation project.
- The user will be prompted for which traffic light to simulate, the North-South (NS) or East-West (EW) lights.
- Variables will be used to control light timing, and the simulation will run twice as fast as real time. See the note below the online simulation.
- Teams will research the Python "for loop" and range() function to replace the while True infinite loop statement. The simulation must end after 4 cycles.

Prerequisites:

Prior to beginning the instruction provided in this lesson you must have completed the following:

1. piRover User Blink

Performance Outcomes:

1. Research requirements and Python code extensions.
2. Create GPIO initialization and control code by referencing prior examples.
3. Prompt for user input.
4. Use variables to control loop timing.
5. Implement a for loop in Python.

Resources:

1. Traffic Light Simulation
2. colorLED.py, blink.py, and user_blink.py

piRover Builds with K2

Materials:

1. piRover

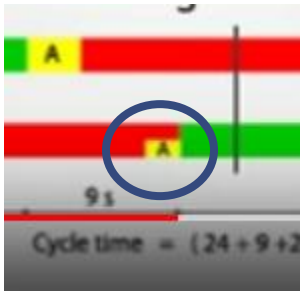
Part 1 – Set Up

a. Prepare your workspace for this activity. Connect to your piRover using VNC and open VS Code
b. Create a directory for this week's work if this was not in the prior session
c. Using the VS Code terminal window, change to this week's directory.
d. Download the starter files for the activity using the wget instructions below.

wget https://k2controls.github.io/piRover01/lessons/25/traffic_light.py

Part 2 – Traffic Light Timing Research

2. Breakout rooms will be used to assign partners.
3. With your partner(s), research the Traffic Light Simulation
4. Determine the time delays for each light (green, amber, and red on both NS and EW roads). Note the simulation runs twice as fast as real time.
5. Ignore the momentary amber conditions shown in the image below.



Part 3 – Initialization

6. Review the starter code provided. Note comments are provided to assist with the components of this solution.
7. Review prior solutions and then enter the variable initialization code including
   a. Importing libraries
   b. Creating pin constants
   c. Creating NS timing constants
   d. Creating EW timing constants
   e. The delay variables are provided for you.
8. Review prior solutions and then enter the GPIO initialization code including

a. General GPIO settings
b. LED pin configurations
c. LED initial state to off.

Part 4 – User input

9. Enter the code required to prompt the user for which direction – NS or EW, which should be simulated.
10. Enter the code required to check the user's direction input. Set the delay variables based on the user's input.

Part 5 – Cycle LEDs

11. Use a while True loop to simulate the cycling of the traffic lights. Review colorLED.py to determine how to produce amber light.
12. Research Python for loop and the range() function. Replace the while True loop so that the simulation loops 4 times and then ends.

Assessment:

Submit your final **traffic_light.py** file to Moodle along with other files in this week's zip file.