Ronning, Peter T

20221208

CIS11 – Assembly Programming

**Street Light / Walk Signal (Final Project)**

1. Situation: The programming of indicator lights for the purposes of traffic control is an essential part of understanding the application of assembly programming in the professional world. In this exercise, the logic for the crosswalk has been laid out resembling a traffic-control scenario as a pedestrian would approach.
2. Mission: To execute a safe, timed period in which the indicator lights show the user they are clear to cross after stopping traffic using the walk button. Additionally, to flash warning when that period is coming to an end. Finally, the simulation requires the ability to end on command as well as run the signal routine.
3. Execution: This program should be run with a Raspberry Pi and a Bridge (or “Cobbler”) to a breadboard. Not all pins may be the same for hardware components but can be easily adjusted in the header of the code.
   1. Steps to Run:
      1. Set up hardware per section 4.a. and attached images/diagrams
      2. Ensure GPIO software is installed
      3. Run makefile with ‘-lwiringPi’ included in command line
      4. Run executable
      5. Press Walk button to watch indicator light routine
         1. This may be done as many times as the user desires without exiting the program
      6. Press Off to exit program and end simulation
4. Administrative Notes:
   1. Hardware
      1. Raspberry Pi 4B, Revision 05 (8GB)
      2. 40-Pin Cable
      3. Breadboard
      4. Cobbler
      5. 6 LEDs
      6. 2 Buttons
      7. 8 Resistors (220R)
      8. Additional wires as necessary
   2. Diagrams and Preparatory Documentation
      1. Diagrams and photos are provided at the conclusion of this write up.
      2. Video demonstration can be found online at:
   3. Programming Notes
      1. This program is comprised of ARM assembly language and a C++ header file for calling GPIO functions (‘wiringPi.h’).
      2. Major values can be easily adjusted at the top of the program to include delay times for various stages of lights, number of times warning blinks for crosswalk, or pin number assignments
      3. GPIO version number: 2.52
      4. This program will remain running until the off switch is pressed – if there are indicator lights still turned on after running due to error, restart the program and push solely the off switch.
5. Included Materials:
   1. Psuedocode:
      1. *Initialize .equ values for reference programming*
      2. *Align*
      3. *Initialize stack*
      4. *Set hardware mapping*
      5. *Set input/output modes for pins*
      6. *Set default values for pins*
         1. *Green Traffic On*
         2. *Red Walk On*
         3. *All others Off*
      7. *Begin loop for basic state*
         1. *Check for Off switch*
         2. *Check for walk switch*
      8. *Walk Switch Branch*
         1. *Delay from press*
         2. *Set Green Traffic to off*
         3. *Set Yellow Traffic to on*
         4. *Wait for Yellow Timer*
         5. *Set Yellow Traffic to off*
         6. *Set Red Traffic to on*
         7. *Wait for Safety Timer*
         8. *Set Red Walk to off*
         9. *Set Green Walk to on*
         10. *Wait for Walk Timer*
         11. *Flash Yellow and Green Walk seven times (1.5s interval)*
         12. *Set Yellow and Green Walk to off*
         13. *Set Red Walk to on*
         14. *Set Green Traffic to on*
         15. *Return to State Loop*
      9. *Off Switch Branch*
         1. *Set Red Walk to off*
         2. *Set Green Traffic to off*
         3. *Exit Program*
   2. Diagrams and Photos
      1. Encl.1: src.zip
         1. Base files to make and execute
      2. Encl.2:
         1. Photo of hardware configuration
      3. Encl.3:
         1. Preparatory notes and hardware schematics