Anthony Chavez

EEE 174-CpE 185 Summer 2020

Monday, Wednesday

Lab 5

Introduction to Data Logging

Dennis Dahlquist

Required Hardware:

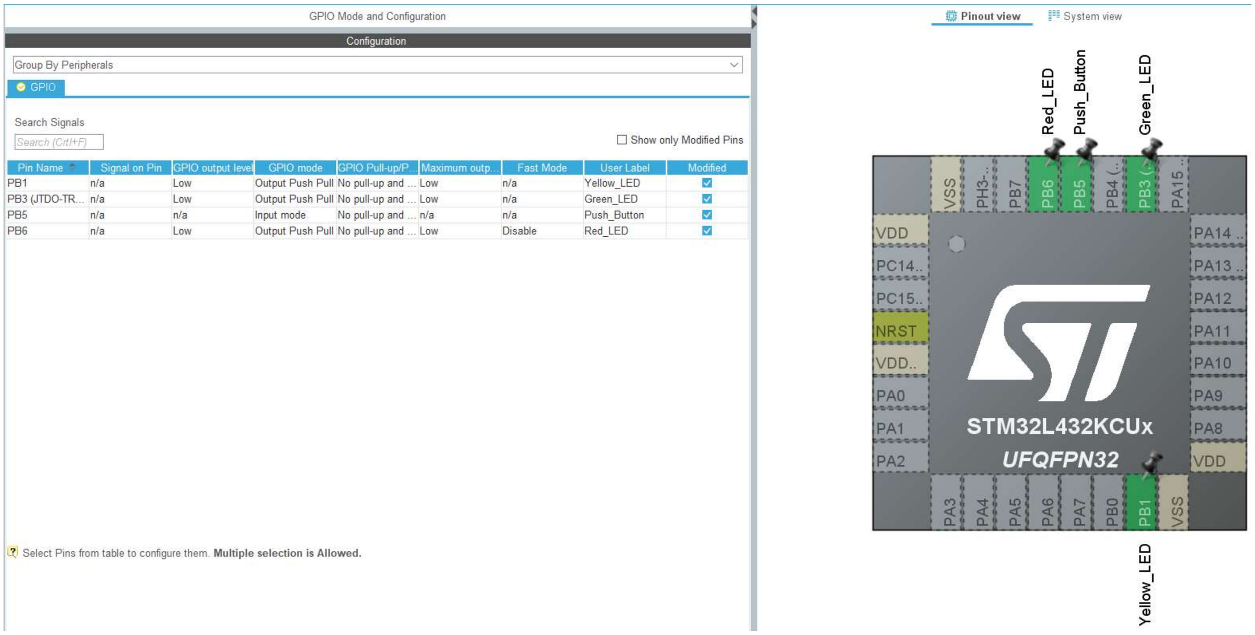
- Raspberry Pi 4
- Micro SD Card with OS loaded
- Simple Keyboard that is linux compatible
- USB mouse
- HDMI Micro to HDMI
- Monitor
- USB C Power Cable
- Three LEDs (Red, Yellow, Green)
- Two Push Buttons
- Five Resistors (1K ohms)
- Nucleo Microcontroller (will be using L432KC model)
- Wires (male-to-male and female-to-male)

Required Software:

- STM32CubeMX software
- Atollic TrueSTUDIO for STM32 IDE
- Python3

Part 1: Setting Up Nucleo MicroController and BreadBoard

First, open the STM32CubeMX software, click on ‘File’, and select ‘New Project’. Second, find the microcontroller you have, in this case we will be using the L432KC model, and click Start Project. Third, set pins PB1, PB3, and PB6 to GPIO_Output and pin PB5 to GPIO_Input. Fourth, open the GPIO panel, change the User Labels, and verify your configuration matches the image shown below.



GPIO Mode and Configuration

Configuration

Group By Peripherals

GPIO

Search Signals

Search (Crtl+F)

□ Show only Modified Pins

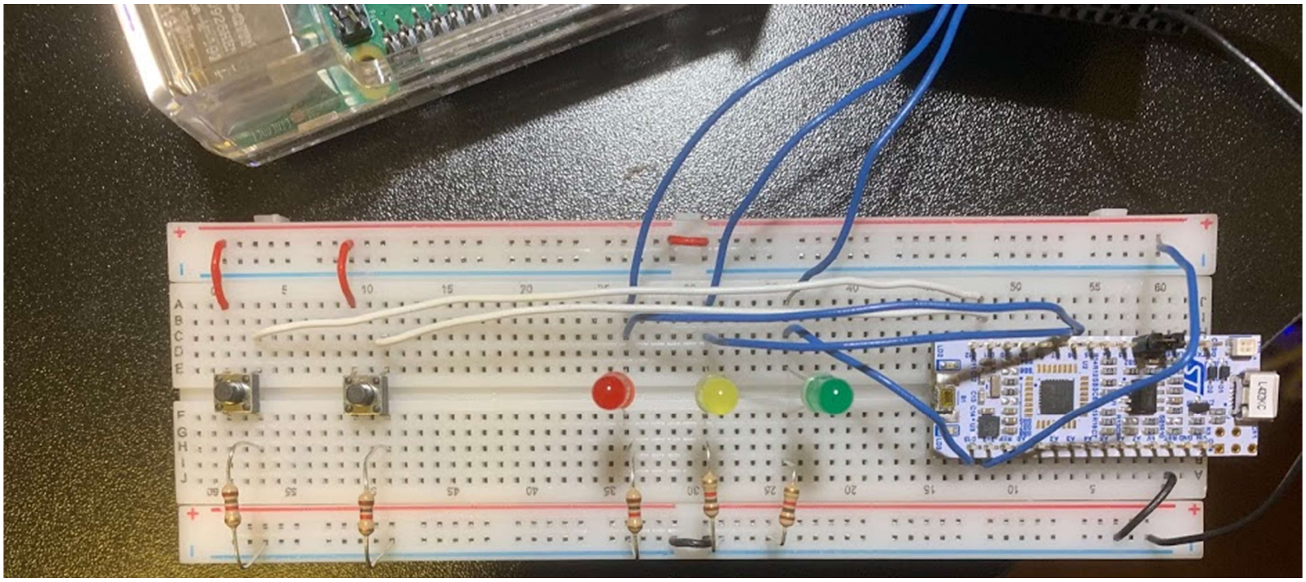
Pin Name	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/P...	Maximum outp...	Fast Mode	User Label	Modified
PB1	n/a	Low	Output Push Pull	No pull-up and ...	Low	n/a	Yellow_LED	☑
PB3 (JTDO-TR...	n/a	Low	Output Push Pull	No pull-up and ...	Low	n/a	Green_LED	☑
PB5	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	Push_Button	☑
PB6	n/a	Low	Output Push Pull	No pull-up and ...	Low	Disable	Red_LED	☑

Select Pins from table to configure them. **Multiple selection is Allowed.**

Next, click on ‘Generate Code’ and ‘Open the Project’. Then, navigate to the main.c file under the Core folder in the Src folder. Now add the following code into the infinite while loop in the main() function, build the project, and upload the code to the board.

```
if(HAL_GPIO_ReadPin(GPIOB, Push_Button_Pin)) {
     HAL_GPIO_WritePin(GPIOB, Red_LED_Pin, GPIO_PIN_SET);         // Red LED "ON", signify door closed
     HAL_Delay(3000); // Delay 3 seconds
     HAL_GPIO_WritePin(GPIOB, Red_LED_Pin, GPIO_PIN_RESET);       // Red LED "OFF", signify door closed
     HAL_Delay(3000); // Delay 3 second
     HAL_GPIO_WritePin(GPIOB, Yellow_LED_Pin, GPIO_PIN_SET);      // Yellow LED "ON", signify door
opening/closing
     HAL_Delay(3000); // Delay 3 seconds
     HAL_GPIO_WritePin(GPIOB, Yellow_LED_Pin, GPIO_PIN_RESET);    // Yellow LED "OFF", signify door not
opening/closing
     HAL_Delay(3000); // Delay 3 seconds
     HAL_GPIO_WritePin(GPIOB, Green_LED_Pin, GPIO_PIN_SET);       // Green LED "ON", signify door opened
     HAL_Delay(3000); // Delay 3 seconds
     HAL_GPIO_WritePin(GPIOB, Green_LED_Pin, GPIO_PIN_RESET);     // Green LED "OFF", signify door no opened
     HAL_Delay(3000); // Delay 3 seconds
     HAL_GPIO_WritePin(GPIOB, Yellow_LED_Pin, GPIO_PIN_SET);      // Yellow LED "ON", signify door
opening/closing
     HAL_Delay(3000); // Delay 3 seconds
     HAL_GPIO_WritePin(GPIOB, Yellow_LED_Pin, GPIO_PIN_RESET);    // Yellow LED "OFF", signify door done
opening/closing
     HAL_Delay(3000); // Delay 3 seconds
     HAL_GPIO_WritePin(GPIOB, Red_LED_Pin, GPIO_PIN_SET);         // Red LED "ON", signify door closed
}
else{}
```

As for the breadboard, connect the wires to the nucleo as shown in the image below. The three blue wires connected to the LEDs and the black wire in the bottom, right corner are connected to the Raspberry Pi' s GPIO. The Red LED is connected to BCM 17 (Board Pin 11), the Yellow LED is connect to BCM 27 (Board Pin 13), the Green LED is connected to BCM 22 (Board Pin 15), and the Black wire is connect to Ground (Board Pin 9).

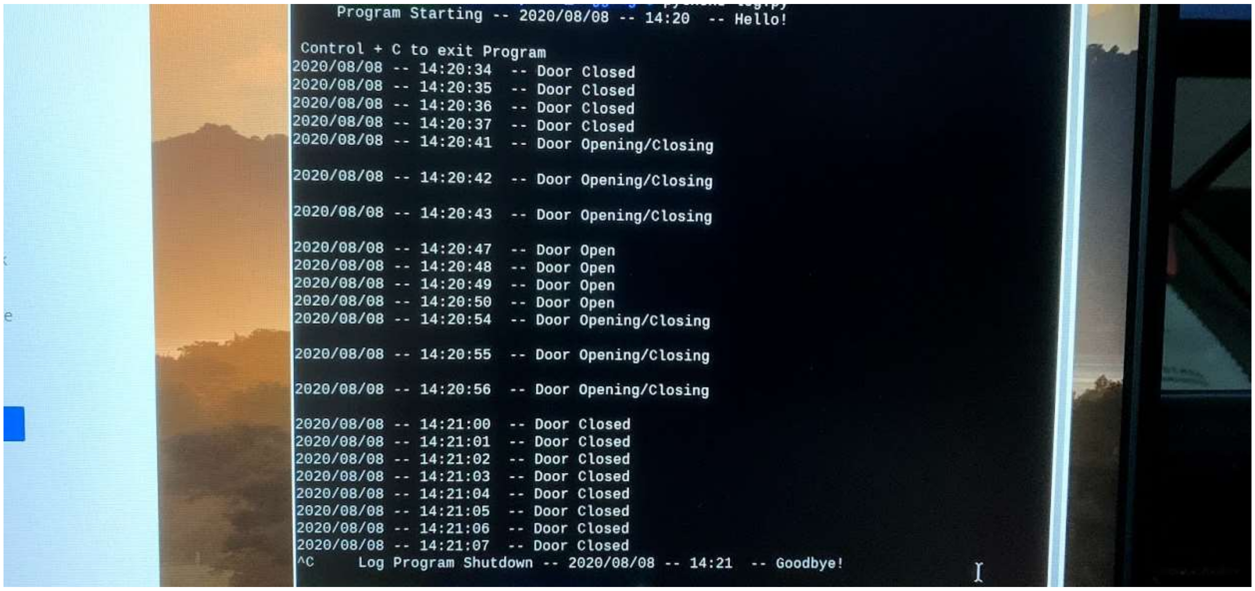


Part 2: Python Script Logger

First create a new folder and name it ‘Lab5’. Then, create a new file in a text editor of your choice, copy the code below, and save the file as ‘log.py’.

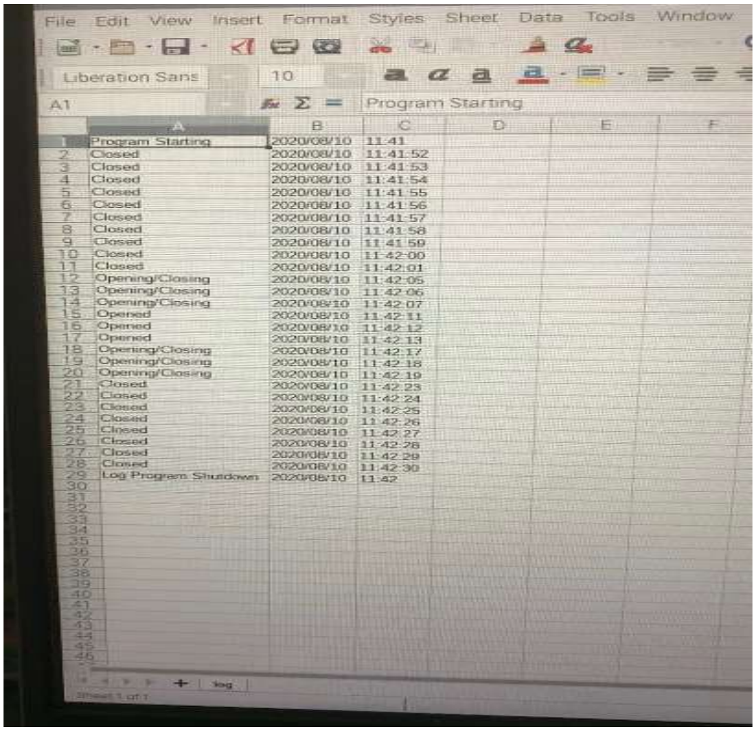
```
import os
import RPi.GPIO as GPIO
import time
from datetime import datetime


logfile = open("/home/pi/Desktop/Lab5/log.csv","a")
logfile.write(datetime.now().strftime("Program Starting,%Y/%m/%d,%H:%M \n"))
logfile.close()
print(datetime.now().strftime("Program Starting -- %Y/%m/%d -- %H:%M  -- Hello! \n"))

print (" Control + C to exit Program")

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

GPIO.setup(17, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)   # Green LED
GPIO.setup(27, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)   # Yellow LED
GPIO.setup(22, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)   # Red LED
time.sleep(1)

try:
        while 1 >= 0:
         time.sleep(1)

         if GPIO.input(17) == GPIO.LOW and GPIO.input(27) == GPIO.HIGH and GPIO.input(22) == GPIO.LOW:  # Door Closing/Opening
           logfile = open("/home/pi/Desktop/Lab5/log.csv","a")
           logfile.write(datetime.now().strftime("Opening/Closing,%Y/%m/%d,%H:%M:%S \n"))
           logfile.close()
           print(datetime.now().strftime("%Y/%m/%d -- %H:%M:%S  -- Door Opening/Closing \n"))  # Used for debugging

         if GPIO.input(17) == GPIO.LOW and GPIO.input(27) == GPIO.LOW and GPIO.input(22) == GPIO.HIGH:  # Door is Closed
           logfile = open("/home/pi/Desktop/Lab5/log.csv","a")
           logfile.write(datetime.now().strftime("Closed,%Y/%m/%d,%H:%M:%S \n"))
           logfile.close()
           print(datetime.now().strftime("%Y/%m/%d -- %H:%M:%S  -- Door Closed \n"))  # Used for debugging

         if GPIO.input(17) == GPIO.HIGH and GPIO.input(27) == GPIO.LOW and GPIO.input(22) == GPIO.LOW:  #Door is Open
           logfile = open("/home/pi/Desktop/Lab5/log.csv","a")
           logfile.write(datetime.now().strftime("Opened,%Y/%m/%d,%H:%M:%S \n"))
           logfile.close()
           print(datetime.now().strftime("%Y/%m/%d -- %H:%M:%S  -- Door Open \n"))  # Used for debugging

except KeyboardInterrupt:
        logfile = open("/home/pi/Desktop/Lab5/log.csv","a")
        logfile.write(datetime.now().strftime("Log Program Shutdown,%Y/%m/%d,%H:%M \n"))
        logfile.close()
        print(datetime.now().strftime("Log Program Shutdown -- %Y/%m/%d -- %H:%M  -- Goodbye! \n"))
        GPIO.cleanup()
```

To test if the ‘log.py’ script is working properly, make sure the nucleo has the program created in Part 1 downloaded. Once the python program is running in the terminal, press one of the push buttons to start the button sequence. The messages on the terminal should match the LED color it corresponds to on the breadboard. You should have a similar output shown in the image below.



```
  Program Starting -- 2020/08/08 -- 14:20  -- Hello!

 Control + C to exit Program
2020/08/08 -- 14:20:34  -- Door Closed
2020/08/08 -- 14:20:35  -- Door Closed
2020/08/08 -- 14:20:36  -- Door Closed
2020/08/08 -- 14:20:37  -- Door Closed
2020/08/08 -- 14:20:41  -- Door Opening/Closing

2020/08/08 -- 14:20:42  -- Door Opening/Closing

2020/08/08 -- 14:20:43  -- Door Opening/Closing

2020/08/08 -- 14:20:47  -- Door Open
2020/08/08 -- 14:20:48  -- Door Open
2020/08/08 -- 14:20:49  -- Door Open
2020/08/08 -- 14:20:50  -- Door Open
2020/08/08 -- 14:20:54  -- Door Opening/Closing

2020/08/08 -- 14:20:55  -- Door Opening/Closing

2020/08/08 -- 14:20:56  -- Door Opening/Closing

2020/08/08 -- 14:21:00  -- Door Closed
2020/08/08 -- 14:21:01  -- Door Closed
2020/08/08 -- 14:21:02  -- Door Closed
2020/08/08 -- 14:21:03  -- Door Closed
2020/08/08 -- 14:21:04  -- Door Closed
2020/08/08 -- 14:21:05  -- Door Closed
2020/08/08 -- 14:21:06  -- Door Closed
2020/08/08 -- 14:21:07  -- Door Closed
^C     Log Program Shutdown -- 2020/08/08 -- 14:21  -- Goodbye!
```

Now, navigate to the ‘Lab5’ folder where the log.csv file was written to and you should have a file like the one shown below.



	A	B	C
1	Program Starting	2020/08/10	11:41
2	Closed	2020/08/10	11:41:52
3	Closed	2020/08/10	11:41:53
4	Closed	2020/08/10	11:41:54
5	Closed	2020/08/10	11:41:55
6	Closed	2020/08/10	11:41:56
7	Closed	2020/08/10	11:41:57
8	Closed	2020/08/10	11:41:58
9	Closed	2020/08/10	11:41:59
10	Closed	2020/08/10	11:42:00
11	Closed	2020/08/10	11:42:01
12	Opening/Closing	2020/08/10	11:42:05
13	Opening/Closing	2020/08/10	11:42:06
14	Opening/Closing	2020/08/10	11:42:07
15	Opened	2020/08/10	11:42:11
16	Opened	2020/08/10	11:42:12
17	Opened	2020/08/10	11:42:13
18	Opening/Closing	2020/08/10	11:42:17
19	Opening/Closing	2020/08/10	11:42:18
20	Opening/Closing	2020/08/10	11:42:19
21	Closed	2020/08/10	11:42:23
22	Closed	2020/08/10	11:42:24
23	Closed	2020/08/10	11:42:25
24	Closed	2020/08/10	11:42:26
25	Closed	2020/08/10	11:42:27
26	Closed	2020/08/10	11:42:28
27	Closed	2020/08/10	11:42:29
28	Closed	2020/08/10	11:42:30
29	Log Program Shutdown	2020/08/10	11:42

References:

https://github.com/shrocky2/GarageWeb

https://pinout.xyz/#

https://os.mbed.com/platforms/ST-Nucleo-L432KC/