

“Microprocessor Design II and Embedded Systems”

“EECE.5520”

“Multithreaded Programming”

“Yan Luo”

“Group number - 12”

*Section 1 : General Lab Info /0.5 points*

“Aravind Dhulipalla”

“Hand in Date – 12/21/2017”

“Lab Due Date – 12/21/2017”

1. Group Member 1 – Aravind Dhulipalla

…………

………….

………….

……………..

…………..

…………..

…………..

1. Group Member 2 – Zubair Nadaph

…………

………….

………….

…………..

………………….

…………..

*Section 2: Contributions /1 points*

*Section 3: Purpose /0.5 points*

…………..

1. Group Member 3 – Dushyanth Kadari

…………

………….

………….

……………….

…………..

…………..

…………..

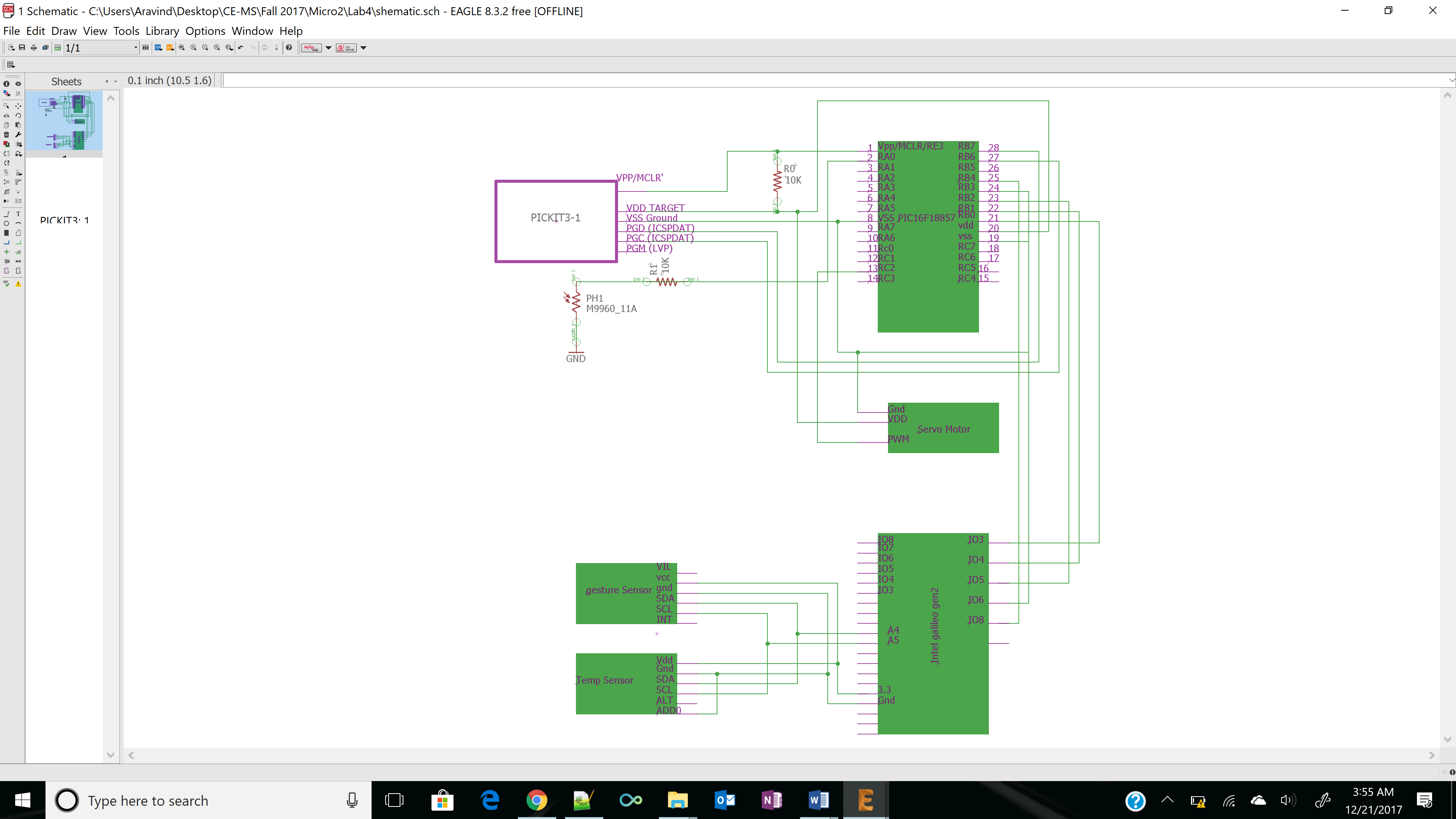
The main purpose of this lab is to understand the multithreading programing using Pthreads. Synchronization of those threads using Mutex. Understanding usage of curl library, HTTP protocol using a client and server application. Understanding of image processing using OpenCV library.

The main objective of this lab is to read the sensor data from a I2C devices Gesture sensor(APDS-9960) and Temperature sensor (TMP102). To read the sensor data (Photo resistor ADC value) from microcontroller PIC16F18857 through strobe communication. Trigger the camera to capture a picture when the required threshold value of the sensor data is reached. Processes the captured image for facial recognition using OpenCV library. And then transfer those images and sensor data to server through HTTP protocol using curl library. Make all these actions concurrent using threads using POSIX thread library.

*Section 5: Materials, Devices and Instruments /0.5 points*

*Section 4: Introduction /0.5 points*

* Bread board
* Wires to connect
* Temperature sensor TMP102
* Gesture sensor APDS-9960
* two 5k Ohm resistors to connect SCL, SDA to VCC
* Serial to USB connector
* Multi-meter
* Voltage supply (3.3V) from Galileo
* Intel Galileo Gen 2 Board
* Yocto Linux
* Putty Software
* PIC16F18857 microcontroller

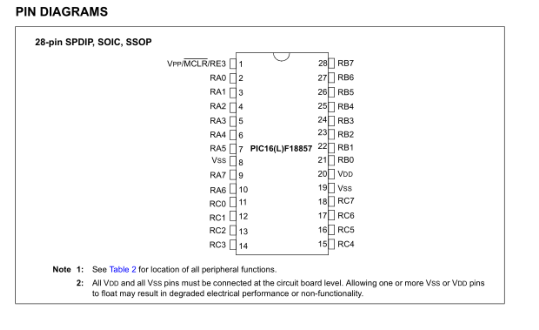
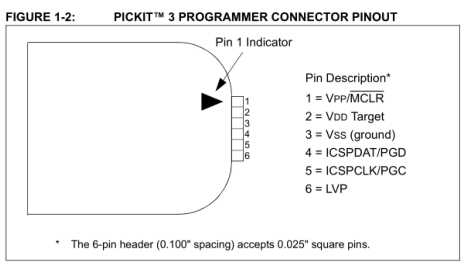


*Section 6: Schematics /0.5 points*

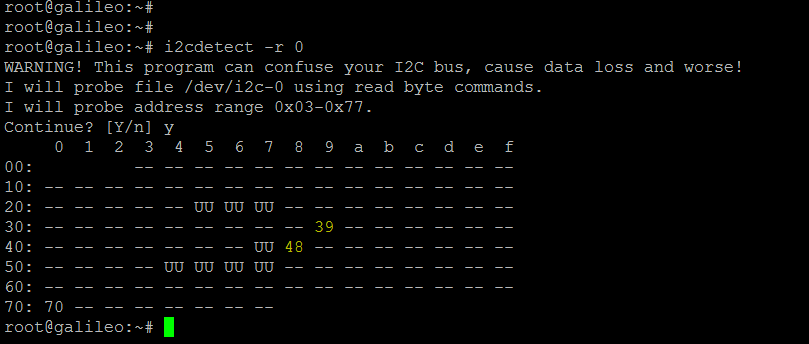
**Hardware design:**

*Section 7: Lab Methods and Procedure /2 points*

**- PIC Microcontroller:** Initially Pickit3 is connected to the microcontroller. If you observe the pin diagram of both Pickit 3 on top and PIC. Both MCLR, Vdd, Vss, ICSPDAT/PGD, ICSPCLK/PGC are connected to each other. ICSPDAT is pin 27 and ICSPCLK is pin 28 for the PIC. The MCLR is connected to Vdd through 10K ohm resistor. The sensor is connected through ADC Channel 2(Pin 4). And LED is connected to the pin PB0 (Pin21). A 220-ohm resistor is connected in series to the LED, for protection. Pin RB2 is connected to strobe(GPIO8) of Galileo. RC0, RC1, RC2 & RC3 pins are connected to the GPIO3,4,5,6 pins of Intel Galileo.

****

* **I2C devices and camera:** Galileo is connected to a laptop using serial to USB connector. It is powered from the adaptor cable. I2C bus is designed on the bread board by connecting SCL, SDA pins from the Galileo board and the sensors as shown in the schematic. Those lines are made active high by connected to VCC through 5k Ohm resistors. On Galileo SCL is A5 and SDA is A4. The VCC (3.3) and ground to two sensors is supplied from the Galileo. In this I2C protocol communication Galileo is the master and the two sensors are slaves. The slave address of Gesture sensor APDS-9960 is 0x39 and Temperature sensor TMP102 is 0x48 (by connecting ADD0 to ground selects default address). After the connection, by typing “i2cdetect -r 0” shows all the I2C devices connected to the Galileo as shown in the below picture. Camera is connected to the Galileo board through the USB cable.



**Wi-Fi connectivity:** It is configured using connmanctl software, after plugging-in the Wi-Fi card to intel Galileo. Use commands from Yacto linux *connmanctl scan* *wifi* to scan the Wi-Fi networks, *connmanctl servies* to view the Wi-Fi networks and *connmanctl connect* $Wi-Fi-id to connect to the selected Wi-Fi network.

**Software design:**

*Sample of topics you should describe here :*

* *Which device modules (i.e.: ADC,TIMER3, CCP2, I2C, etc) do you need to configure*
* *How did you configure your device (device modules ADC, PWM, etc )*
* *Explain your application logic of your main program*

*- Use references or code pieces expelling its purpose and functionality*

………… ………… ………… ………… ………… …………

………… ………… ………… ………… ………… …………

………… ………… ………… ………… ………… ………… ………… ………… ………… ………… ………… …………

………… ………… ………… ………… ………… …………

………… ………… ………… ………… ………… ………… ………… ………… ………… ………… ………… …………

*Section 8: Trouble Shooting /1 points*

*Section 9: Results /0.5 points*

“Charts..”

………… ………… …………

………… ………… …………

………… ………… …………

“Measurement Tables”

………… ………… …………

………… ………… …………

………… ………… …………

“Terminal Screenshots”

………… ………… …………

………… ………… …………

………… ………… …………

“Short explanation of each element you post here”

………… ………… …………

………… ………… …………

………… ………… …………

“How did you test your output signal - post a oscilloscope picture of your signal and explain”

………… ………… …………

………… ………… …………

………… ………… …………

*Section 10: Appendix*