LAB 3

Group 7:

Rui Ge - rg3105
Yuekun Guo - yg2519
Miguel Angel Gutierrez - mag2293

My_time Object:

Our code defines a My_time object that holds the current time, and is updated every second by a virtual timer. My_time contains functions to increment the time, and to generate a formatted string with the time.

OLED Control:

Communication to the OLED module was done via I2C. The micropython libraries available on the esp8266 provide code for interfacing with the ssd1306 OLED. Adjusting the brightness on the screen was a matter of adjusting the contrast on the board. Horizontal scrolling is already implemented in the OLED hardware. To enable horizontal scrolling, commands were sent to the ssd1306 to set scrolling parameters and finally to activate scrolling. Vertical scrolling had to be implemented in software. To do this, we simply sent commands to change the OLED start page, effectively moving the image up or down.

Buttons A,B and C are read from the OLED board and used to increment the hour, minute, and to activate alarm-set mode, respectively.

SPI Communication:

The connection circuit is :

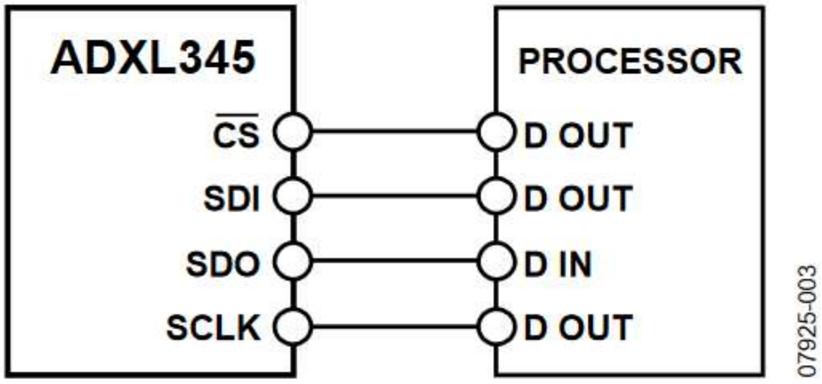

Figure 4. 4-Wire SPI Connection Diagram

And the Pin is :

```
    spi = machine.SPI(baudrate=100000, polarity=0, phase=0,
    sck=machine.Pin(16), mosi=machine.Pin(0), miso=machine.Pin(15))
```

In here, the sck means clock, mosi means machine out and slave (accelerometer) in (SDA/SDI), miso means SD)

cs = machine.Pin(2,machine.Pin.OUT) which means ‘chip select’.

The communication process is first transfer the data and address to ‘bytes’, then choose the chip using ‘cs’ write the address which means the register holds the data and then write data. The read process is similar with write data first and then read 1 or 2 bytes from the address.

Alarm:

A my_time object, alarm_time, is initialized to hold the time set to be the alarm. alarm_time is incremented by pressing button_c. A timer compares alarm_time and my_time every second. The alarm sounds (turns on an LED) for one full minute.

Scrolling:

The accelerometer is sampled at 10Hz. The device’s orientation is calculated and a flag is set to describe the orientation and the appropriate scrolling function to call.

References:

Micropython Documentation:
----

http://docs.micropython.org/en/latest/esp8266/

Ambien Light Sensor Datasheet:
----

http://www.everlight.com/file/productfile/201407061531031645.pdf

SPI Communication:
----

https://learn.adafruit.com/adafruit-feather-huzzah-esp8266/pinouts

https://docs.micropython.org/en/latest/esp8266/library/machine.SPI.html?highlight=spi

https://docs.micropython.org/en/latest/esp8266/library/ustruct.html?highlight=ustruct#ustruct.calcsize

https://morf.lv/mems-part-1-guide-to-using-accelerometer-adxl345

https://github.com/adafruit/Adafruit_Python_ADXL345/tree/master/examples

https://medium.com/@mosdeo

OLED:
----

https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf https://github.com/micropython/micropython/blob/master/drivers/display/ssd1306.py