CALIFORNIA STATE POLYTECHNIC UNIVERSITY, POMONA
COLLEGE OF ENGINEERING

ECE 3301L Spring 2019 Microcontroller Lab **Felix Pinai**
Session 2

LAB 13: Final Project

This final project is the accumulation of all the topics that we have covered this semester:

- GPIOs
- A/D Converter
- D/A Converter
- Temperature sensor
- Light sensor
- PWM – Fan & speaker
- Timer and Counters
- System interrupts
- TFT interface
- I2C bus
- SPI bus
- RTC

The final project will integrate a design that will have the following functions:

- A TFT panel provides a main display.
- A digital clock with the time/date programmable through a remote control.
- The digital clock has an alarm function with the time programmable through a remote control. The alarm will sound a sound when the alarm is matched. The sound is reset through a push-button switch. A multi-color LED is activated when the alarm sounds.
- A thermostat function that will turn on a heater when the temperature is below a programmable temperature.
- A light sensor will activate a LED when the light is below a light threshold.
- A light dimmer controlled through the remote control

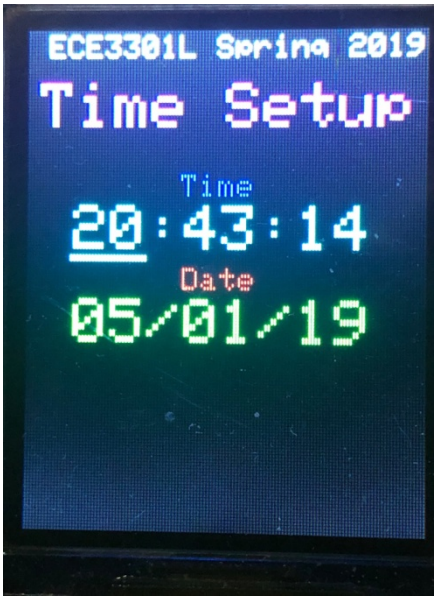
The hardware schematics is provided on a separate pdf file.

Here is the screenshot of the main screen:

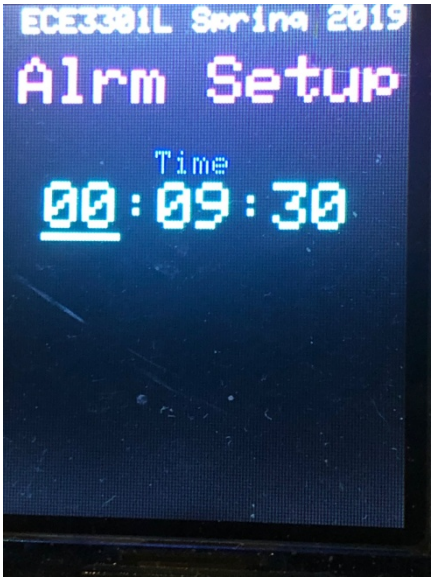
Here are the descriptions of the fields:

- ‘25C/77F’ : Temperature display
- ’21:34:06’ : Time display
- ‘05/01/19’ : Date display
- ‘Alarm’ : ON or OFF depending on the Alarm DIP switch
- ‘Heater’ : ON or OFF depending of the Heater DIP switch
- ‘LED’: value from 0 to 90 to control the intensity of the LED
- ‘Heater Set Temp’: Set temperature for the Heater.

Here is the screenshot of the Time Setup Screen:



Here is the screenshot of the Alarm Setup Screen:



Here is the screenshot of the Heater Setup Screen:

Operation:

A) From the Main screen, the display will show:

- Actual temperature in Degree C and Degree F
- Actual Time in hour:minute:second format
- Actual Date in month:day:year
- Actual setting of the DIP Switches 'Alarm Enable' and 'Heater Enable'
- Actual setting level of the dimmable LED
- Actual setting of the heater threshold level when the heater is on. '—F' should be displayed if the heater switch is off

The following buttons on the remote will control the operations on the display:

- 'CH-' button will lower the level of the dimmable LED. The level should be decreased by 10 every time this button is pressed. The range of the level is from 0 to 90. The level should be wrapped around to 90 when the value is at 0.
- 'CH+' button will increase the level of the dimmable LED. The level should be increased by 10 every time this button is pressed. The range of the level is from 0 to 90. The level should be wrapped around to 0 when the value is at 90.
- 'EQ' button will force to the Time Setup Screen.
- '+' button will force to the Alarm Setup Screen.
- '-' button will force the Heater Setup Screen.

B) In the Time Setup Screen, the following buttons will control the display:

- '>>|' button will select the next field for setup. A cursor will be moved to the next field when this button is pressed to show the field to be modified. The cursor will be wrapped around for all the fields.
- 'CH+' button will increase the value in the field.
- 'CH-' button will decrease the value in the field.
- '>||' button will confirm the change in the setup and return to the Main display
- |<<' button will abort the setup and return to the Main display.

C) In the Alarm Setup Screen, the same buttons on the Time Setup Screen will have the same effects on this screen.

D) In the Heater Setup Screen, beside the button '>>|', the other buttons in the Time Setup Screen will work the same in this screen.

E) When the 'Alarm Enable' DIP switch is on, then the Alarm mode is activated. When the Alarm time setup in the RTC matches with the actual time, a falling edge signal 'Alarm_Alert' will be generated from the RTC chip to the INT1 pin. An interrupt should be generated and a software alarm flag should be set to 1. In the main loop, when the alarm flag is detected to be set, then the buzzer should be activated and let in the ON state. At the same time, the RGB LED at D1 will change with 8 different colors every second. The buzzer and the RGB LED will be turned off with the 'Alarm Reset' push-button switch is pressed. This button is connected to the INT0 pin. An interrupt should be generated when the button is pressed to clear the buzzer.

F) When the 'Heater Enable' DIP switch is on, the temperature should be monitored. When the temperature is lower than the Heater Set temperature, the fan should be turned on and stays on until the temperature is higher than the set value.
#

G) the SPI_out() function in a previous lab to output the LED intensity to the D/A chip. Take the value of this LED intensity and add 100 to it before using the SPI_out function to output it to the D/A converter.
#

H) The voltage output from a light sensor is measured through the A/D port. If that voltage is above 2.5V, then the RGB at D2 must be turned on with the White color. Else, D2 should be off..

Guidance:

1) To help in the design, a basic program is provided with the Main screen handling along with the implementation for the Time Setup routine. The student can base on the sample example to implement the Alarm Setup and the Heater Setup routines.

2) The main loop should have the duty to monitor for the following events:

#

- Second change in order to update the TFT display
- INT0 interrupt for the detection of the Alarm Reset
- INT1 interrupt for the Alarm Alert signal asserted by the RTC when the Alarm Mode is on
- IR remote key detected (nec_ok). When this flag is set, check for what kind of key is pressed and execute the appropriate function
- Monitor the change of the two DIP switches Alarm Mode and Heater Mode. Update the associated variables so that the TFT screen shows the proper setting. Also,

3) To enable the RTC's Alarm, just clear the Interrupt Flag bit in RTC's register 0F and set the Interupt Enble bit in the RTC's register 0E. To access and modify the RTC's register, look at the provided function DS3231_Init(device).

#