

Assignment #1

Clock_LCD_Interrupts_AI

Dr. Raed Alqadi

In this assignment you will design and write the code for a Simple Digital Clock, read AI, use the Interrupts, Serial, and LCD. We will control the device locally by using the Push-buttons and remotely by using the UART (Serial Port).

Digital Clock:

We will design a simple Digital Clock that displays the time every 1 second. Use **3 Variables: Seconds** (0 –59), **Minutes** (0 –59), and **Hours** (0—23). Use Timer 0 to create an Interrupt every 1 second. Increment the Seconds every interrupt (rollover on 60 and increment the minutes), increment the Minutes variable every minute (rollover on 60 and increment the hours). Rollover on the hours on 24.

- There will be 2 modes for the clock operation (Clock Mode) are: **Setup** mode and **Normal**. The modes will be set by Interrupt INT0 at RB0. INT0 will circulate each time it is pressed through the Modes (You can use 2 values e.g., 0 – 1).
- Once in the **Setup** mode then INT1 (RB1) will circulate through setting second, minutes, hours. You can use 3 for the Setup mode (values e.g., 0 – 2) for setting the Second, Minutes and Hours respectively. If the mode is not Setup Mode for the clock, then INT1 shall be ignored or do not enable it in that case.
- To increment/decrement the Seconds, Minutes, or Hours. Increment/decrement the Value by using **RB2** and **RB3** respectively in the main loop of the main program, do not use the Interrupts in this case. If the user keeps pressing the button then increment/decrement every 200 milli seconds which can be easily achieved by using a 300ms delay in the main loop which will also debounce the buttons.

Important Note: Do not display or read Analog Inputs in the Interrupt routines, do this in the main Loop.

Analog Inputs:

The Simulator has 3 Analog Inputs AI0, AI1, AI2. In this application we will only read AI2.

Analog Input 2 (AI2): This is already connected to a simulator that reads a voltage already connected to a sensor. Read the Voltage (0 –5) then scale it to temperature by just multiplying it by 100 since an increase of 10ms represent an increase by 1C. The minimum Value that it reads is around 27 degrees without doing extra changes. We will leave it as is to simplify things and we will control Temperatures above the minimum Value. The Value of this is the actual temperature of the room

And we shall Refer to it as **T (Temperature)**.

Heater and Cooler

Heater: This is already connected to RC4. Turn it ON/Off by setting /Clearing this bit.

Cooler (Fan): This is already connected to RC2. Turn it ON/Off by setting /Clearing this bit.

Operation Modes: This is a Value that we should set Digitally through Buttons **RB4** and **RB5**. Read these buttons in the main loop. To turn the heater ON click (press RB4) clicking it again will Turn it OFF. Similarly for the cooler but use RB5. Debounce the buttons by using the delay as described for RB2 and RB3. If the user keeps pressing button it should not change unless the button is released.

LCD Display:

On the LCD Display, the followings in a Nice Format.

- Line 1: Time Temperature.
Display time on first half of line and Temperature on the second half
- Line 2: C: ON/OFF H: ON/OFF
Display Cooler Status either ON or OFF on First half and Heater Status on Second Half.
- Line3: Normal/Setup --/second/Minutes/Hours
Display Clock mode, if setup then display Minutes or Seconds or Hours. If Normal display nothing
- Line 4: Name 1 Name 2
Display your first name and your partner's first name

You can change the display, but you must display all values included in the lines above.
Once You change anything through the Serial Port Protocol described in

Serial Port

Implement the following Protocol to read/write values of the Application. You should read the character by using the RX Interrupt. DO not send back the response from the Interrupts. Do it in the Main Loop. Read the first character if it is not one of the starting characters in this table just ignore it.

	Command	Format	Value Read
1	Read Time	R	Reads the Time as string in the format hh-mm-ss
2	Read Temperature	T	Reads the Temperature e.g. 34.6
3	Read Hours	whh:mm:ss	Write the complete time

			<p>Examples</p> <p>w06:23:09 make the time 6:23:09</p> <p>w22:12:45 make the time 22:12:45</p> <p>w:00:02:01 make time 00:02:01</p> <p>if there is any error in the 9-character command just ignore the settings. Do not write until you read 9 correct letters.</p> <p>Check each letter once you notice and error then ignore</p>
4	Anything else		ignore

Submit:

The complete code and hex file

A small text file with your