School of Engineering

Grand Valley State University

EGR 326 – Lab #5

**Interfacing with a Real Time Clock (RTC) IC Using An I2C Bus**

**Objectives**

* To develop a C program function that manages the I2C bus as a master and talks to the RTC as a slave device
* To develop a C program function for the MSP432 microcontroller that interfaces to a real time clock (RTC) IC to set the time and date and then later recall current time and date

For the embedded system design project, you will be writing program functions in C that use the MSP432 on-chip eUSCI peripheral in I2C mode to interface with a RTC IC to set and keep time for your security system. Even though the MSP432 has a capable RTC as an on-chip peripheral, the Launchpad board is not conveniently set up for battery operation. This lab provides an opportunity to develop some experience working with the RTC as an I2C slave and write some code to manage the I2C bus protocol.

**Procedure:**

Much of what is required for this lab has been developed during the week for the pre-lab assignment and in a previous lab. Use lab time to test your code using the CCS IDE on your Launchpad board and the hardware debugging feature. The RTC breakout board should be plugged into a breadboard so that the SDA and SCL lines can be connected to the appropriate I/O pins for the eUSCI peripheral on the MSP432 microcontroller.

---------------------------------------------------------------------------------------------------------------------

**Part I – Setting the current time on the RTC IC**

In this part, you will connect the MSP432 I2C bus interface pins to the RTC SDA and SCL pins and send it commands over this I2C bus to set the current time and date.

1. Make sure that you’ve disabled the battery charging circuit on the RTC breakout board and have properly installed the battery per instructions in the pre-lab.
2. Wire up your RTC breakout board so that the SDA and SCL signals are routed to the appropriate pins on the MSP432. Use the 3.3V power and ground pins on the Launchpad board to power the RTC.
3. Write a program that prompts the user over the MSP432 serial port connected to a terminal emulator or console (for the design project, you will display time and date information on your LCD) to enter the date and time via the keypad interface you established in a previous lab. Then send commands to the RTC to set the time and date to these values. Use the time and date displayed in the lower corner of your Windows PC screen as a reference value.
4. Prompt the user to press the “\*” key on the keypad to read the time and date information and then send a command from your program to the RTC. Print the time in hours, minutes, and seconds as well as the day of the month, day of the week, and year to the CCS console or terminal emulator. These values should be within a few seconds of the PC time/date display that was consulted when you set the RTC. Every time the user presses the ”\*” key, the time and date information must be displayed.
5. If you’re having problems communicating with the RTC, it might be time to move on to Part II and come back later, otherwise, demonstrate the working program to the instructor.

**Part II – Displaying the SDA and SCL lines on your oscilloscope**

In this part you will gain experience working with a digital logic analyzer that will be an invaluable aid to trouble-shooting communications on an I2C bus. The lab scope has a nice built in feature that allows you to display all the features of your I2C messages. Lab support also has portable Saleae logic analyzers that can do the same thing with the help of software from [www.saleae.com](http://www.saleae.com).

1. Graphical user interface

   Description automatically generatedConnect two scope probes to your Tektronix MSO 2022B oscilloscope. Note, you could also use the logic analyzer probes (available from lab support)
2. Select the Purple “B1” button

Graphical user interface

Description automatically generated

1. Next several menu button options appear on the bottom of

Screen and a new purple probe line should appear in the display.

1. Select the first button to setup the probe for I2C
2. A screenshot of a computer

   Description automatically generated with medium confidenceNext select the second button to set your probes as SDA and SCLK.

Make sure they are in the correct order, or it will not register.

1. See your instructor if you have questions on the other buttons.

Now, if you do a single capture, you can expand an I2C group,

And the scope will identify the start and stop bits, and all the data sent.

Timeline

Description automatically generated

SAMPLE

1. Keep a snapshot of some interesting screens to include in your lab report.
2. When you have this set up, demonstrate this for your instructor.

**Part III – Reading and displaying the temperature on the RTC board**

The RTC IC provides an added bonus of housing a temperature sensor that may be accurate enough to monitor ambient temperatures to detect a fire hazard (references on the web seem to indicate so, my experience with it seems to bear this out). If so, a temperature sensor for your design project will be as simple as reading a register on the DS3231!

1. Study the DS3231 data sheet and figure out how to read the registers (2 bytes) that represent temperature in degrees C (one byte is an integer value, one byte is a fractional value).
2. Write a C routine that will read these registers and format the temperature for display (you don’t have to convert to Fahrenheit for this lab unless you wish). Start by displaying the integer value, then format the display for both integer and fractional parts.
3. Modify your main program so that calls to read the RTC are now in a subroutine. When the user enters a “\*” followed by a “1” on the keypad, the current day, month, and year are displayed on the serial monitor. When the user enters a “\*” followed by a “2” on the keypad, the hour, minute, and second are displayed. And if the user enters a “\*” followed by a “3” on the keypad, the temperature is displayed in Celsius or Fahrenheit.
4. Demonstrate this for your instructor.

**Questions:**

How can the RTC signal the MCU that the time in seconds has changed (without the MCU initiating an I2C message)?

How can the RTC signal the MCU that the time in minutes has changed?

A careful reading of the DS3231 data sheet will suggest an answer to these questions. You may want to consider using this feature in your project design.

---------------------------------------------------------------------------------------------------------------------

**At the End of the Laboratory**

* Clean up your workstation.
* Make sure you clearly understand the laboratory deliverables and due date as posted on Blackboard.
* Logoff and reboot your laboratory computer (if used).

**Laboratory Deliverables**

Submit a report to Blackboard before the due date/time according to the format posted on Blackboard in the Assignments folder. You must demonstrate your working program to your instructor no later than the beginning of the next lab period.

* Make sure you remember to answer the questions asked. Write out the question and answer completely in a few sentences.
* Include well commented source code listings of the complete program of Part IV with a detailed description header section for each routine. These listings should be appended to your report.