EE333 Introduction to Microcontrollers

Lab#3 Digital Clock

Allan A. Douglas, Associate Professor

**Lab Overview**

In this lab, students will learn how to interface the Arduino UNO with an LCD display, and a Real-Time Clock IC. The two additional components will be used to create a digital clock that displays the time and date.

**Required Materials**

|  |  |
| --- | --- |
| Breadboard | Various resistors |
| Arduino UNO (or Spark Fun RedBoard equivalent) | Jumper wires |
| ATmega328P Microcontroller Datasheet | Oscilloscope |
| SparkFun LCD-00255 Liquid Crystal Display (LCD) | Arduino ISE Software |
| 10K Potentiometer | DS1307 Real Time Clock (RTC) IC |
|  | 32.768 kHz Crystal |
|  |  |

**Required Work**

Part 1: Liquid Crystal Display (LCD)

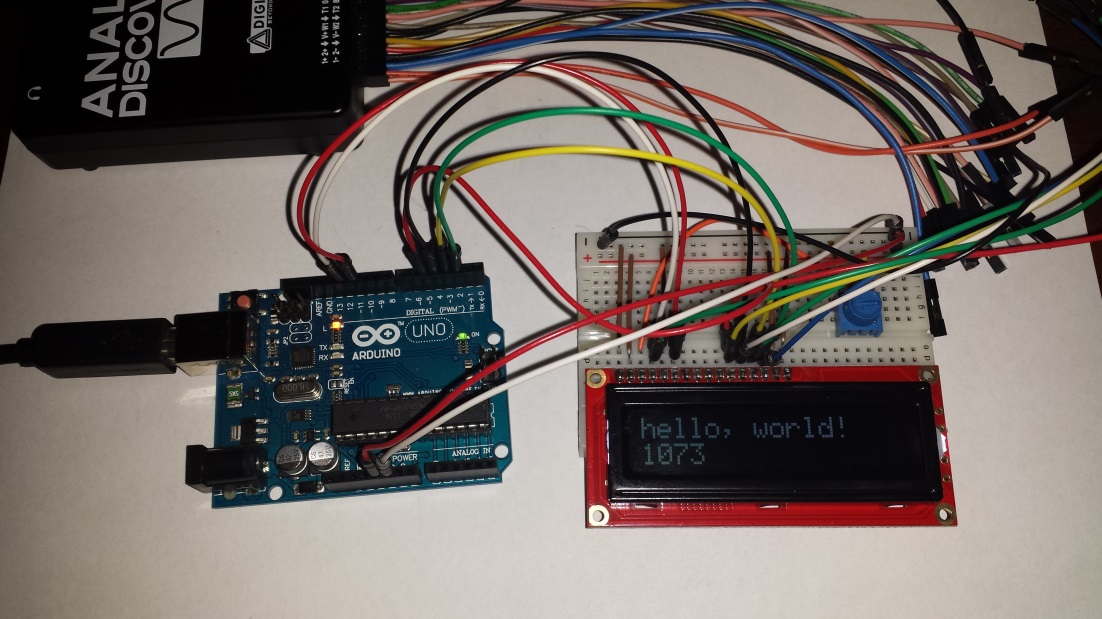
Technical information about the LCD can be found at <https://www.sparkfun.com/products/709>. The best resource is the Extended Data Sheet.

Connect the LCD and potentiometer as shown:

c

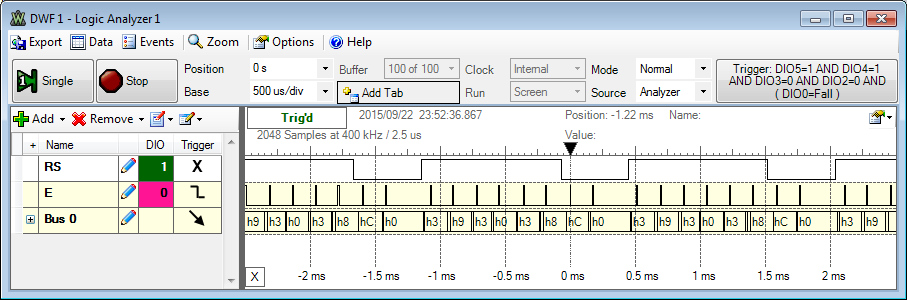


Load the example sketch called HelloWorld (File>Examples>HelloWorld). You should see something like the following figure. Make sure to adjust the potentiometer to set the LCD contrast so that the display is visible.



The number that is incrementing in the second row is the number of seconds that the microcontroller has been running since the reset was asserted.

Next, connect the signals called RS, E, D4-D7 to a logic analyzer. You should get a set of signals that looks like the following figure.



The microcontroller writes character data to LCD internal memory over a 4-bit parallel bus.

Read the datasheet for the LCD and describe how the microcontroller code in the sketch is creating the display. Relate the commands in the sketch to the signals you observe on the logic analyzer.

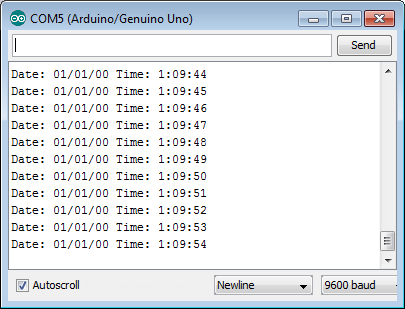
Part 2: Real Time Clock (RTC)

The datasheet for the DS1307 RTC can be found here <http://datasheets.maximintegrated.com/en/ds/DS1307.pdf>.

Place the RTC into your breadboard and connect the 32.768 kHz crystal across the X1 and X2 pins. Connect 5V and ground to the chip. Finally, connect the I2C SCL and SDA pins from the microcontroller to the RTC. Connect the RTC Vbat pin to ground.

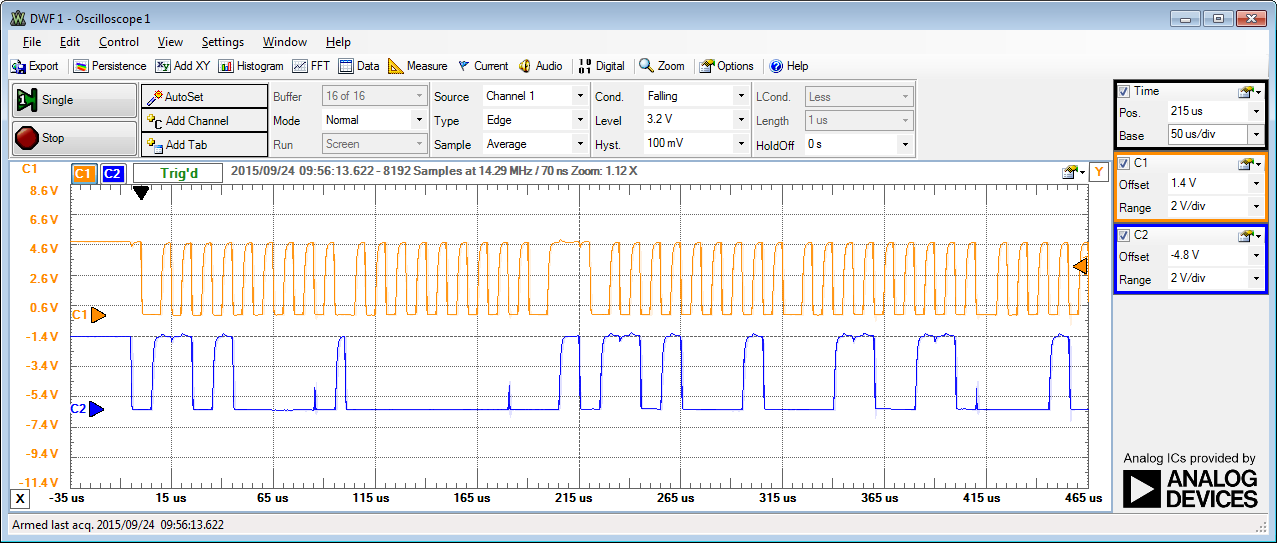
Download the sketch called **Real\_Time\_Clock.ino** from Canvas. Load the sketch into the microcontroller.

You should see an output on the Serial Monitor that looks like the following:



If the sketch does not work, check your connections.

Observe the SCL and SDA pins on the microcontroller. You should see something like the following:



Decode every bit on the I2C bus and explain how this relates to the microcontroller code in the sketch.

Examine the code in the sketch and explain how it works in your own words. Make sure to carefully explain the portions of code that write to the serial port. What do these portions of the code do?

Serial.print((minutes & 0x70)>>4);

Serial.print(minutes & 0x0F);

Why is the display of the month treated differently? Relate your discussion of this code to the datasheet for the RTC on Table 2. Timekeeper Registers. Make sure to fully explain the bitwise operators.

Also, carefully explain the section of code in the setup() function that implements the following instructions

Wire.beginTransmission(0x68);

Wire.write(0); // set the address

byte sec = Wire.read();

Wire.endTransmission(); // stop transmitting

Wire.beginTransmission(0x68);

Wire.write(0); // set the address

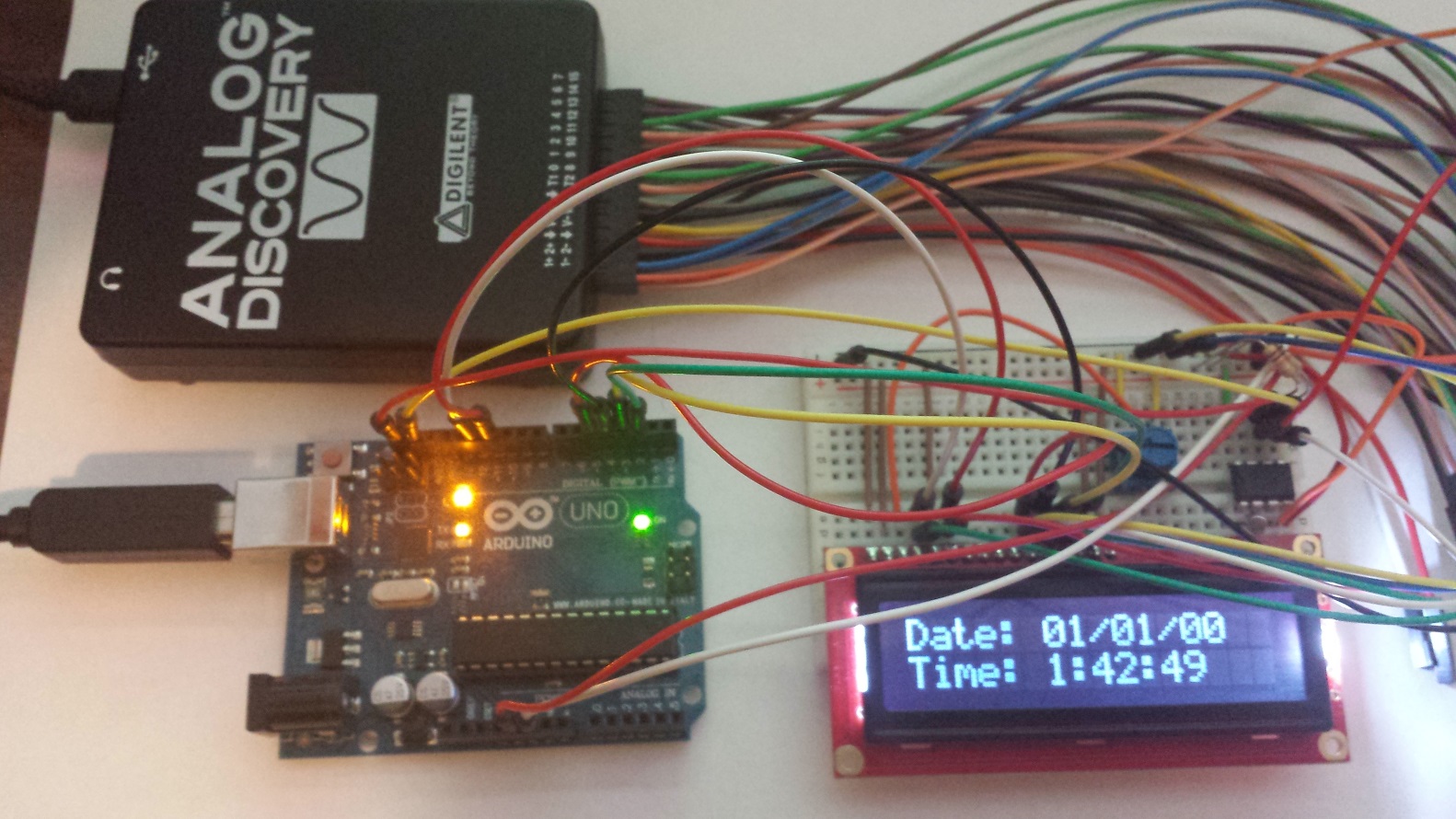
Wire.write(sec & 0x7F); // clear CH bit

Wire.endTransmission(); // stop transmitting

Why is this necessary?

Part 3: Creating a Digital Clock

Use what you have learned about the LCD and the RTC to build a digital clock. You will need to write your own code for this part of the lab. You may use functions from the other two sketches. Your working sketch should look like the following figure.



**Required Work Checklist**

□ Part 1.

Include all measured waveforms.

Answer all questions stated in the section.

Include a complete description of the HelloWorld sketch.

□ Part 2.

Include all measured waveforms.

Answer all questions stated in the section.

Include a complete description of the Real\_Time\_Clock sketch.

□ Part 3.

Include a complete copy of your sketch. Make sure your code is commented.

Include a picture of the working LCD displaying both the formatted date and time.

**Extra Credit Work**

1. Add a function to the sketch you wrote in Part 3 that sets the time and date on the RTC. You will need to write data to the RTC over the I2C bus.
2. Connect a 3V power supply (battery) to the Vbat pin and connect the power supply to ground on your breadboard.
3. Note the current time and disconnect the microcontroller from USB (power).
4. Reconnect after about 10 seconds. Confirm that the RTC time continues tracking the time accurately.

**Extra Credit Checklist**

□ Include a copy your sketch and describe your new function.

□ Include a picture of your RTC Vbat power supply.

□ Explain how the RTC continues to operate without power from the Arduino UNO board.