

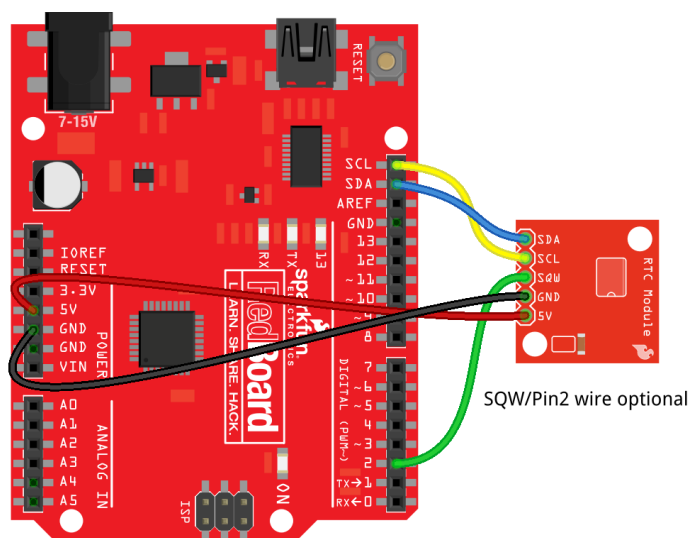
Lab #4a

I2C Real Time Clock (RTC)

Ver 0.54

Background:

This lab we will use an I2C Real Time Clock Module wired as follows (wire from Pin 2 to SQW not required and on many Arduino boards SCL and SDA are NOT labeled, but are the two pins above AREF as shown below)



fritzing

Figure 1- Wiring of RTC

Start an ATMELE Project as normal (with Usart support and printf enabled). Go to the author's website at https://nicerland.com/eduFiles/AVR/codes/source_code.php?chapter=18 and copy the code for program Chapter18/Program18_13.C and copy that into main.c in your project (erase all but the #include <atmel_start.h> and atmel_start_init(); from the original main).

This example code did not use printf and/or the ATMELE Studio USART Functions but instead controlled the USART directly (as we did in lab 2).

Remove the functions usart_init, usart_sendByte and usart_sendPackedBCD.

In main, replace the usart_sendByte and usart_sendPackedBCD function calls with ONE printf statement. Since the RTC returns all its values as "PackedBCD" we can actually use %X to display the correct values (since BCD represents digits 0-9 in 4 bits, displaying these values as hex will display the correct values). For Min and Sec, we do need to force the leading 0 if the value is <10 (eg: we want 9:02:01 for the time not 9:2:1). To do this we can add additional formatting to our printf statement, so

instead of just using %X, we will use %02X which tells printf to print leading zeros and there must be at least 2 digits. This will produce the desired output.

We will use an “Escape Sequence” that tells putty (acting as a VT100 terminal) to overwrite the existing line with the new time and date by adding the escape sequence \E[H at the start of the line. See <http://ascii-table.com/ansi-escape-sequences-vt-100.php> for details.

The output should be a single line of text as follows:

```
2020-10-19 21:07:13
```

Setting the clock: You can call the functions rtc_setTime and rtc_setDate to set the current time and date (entering these as hex values representing the BCD so 10/10/20 9:45:30 would be 0x10 0x10 0x20 0x09 0x45 0x30). When you download the program, let it run (but do NOT START PUTTY...arduino’s reset when you open the com port so the time will be reset a 2nd time). Once it runs a bit, remark out the two lines of code and then compile and download to the board. After this, you can run putty to see the output.