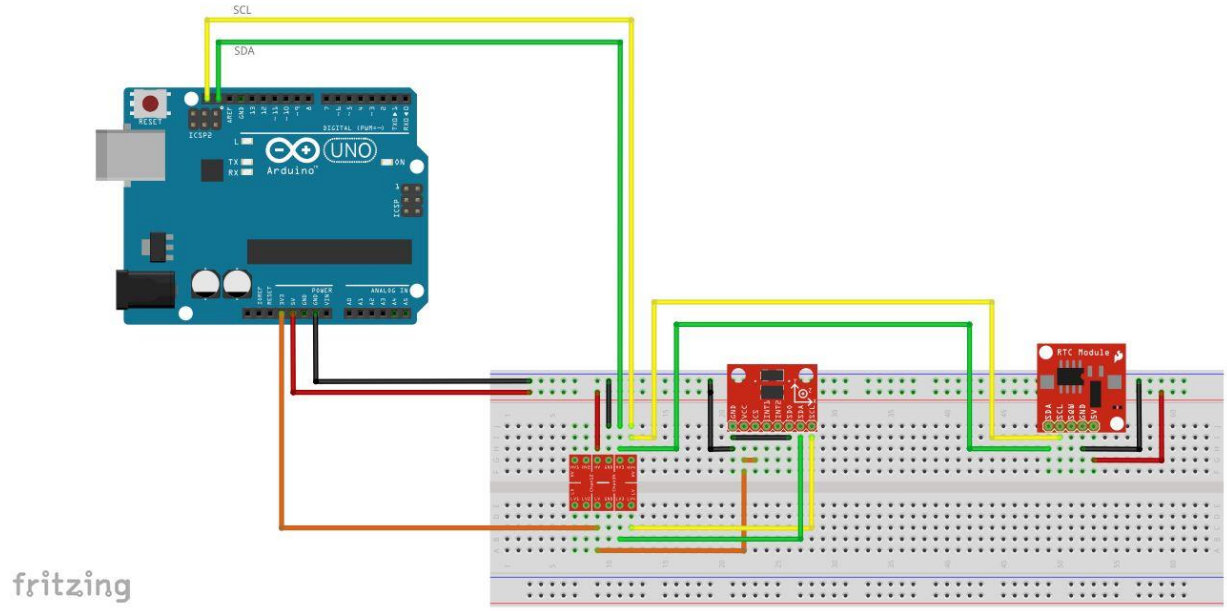
Lab #3b
RTC and ADXL345

Ver 0.8

Background:

This week we will combine what we did in Lab #3a with the ADXL345 Accelerometer in I2C mode. To hook up both devices to the I2C bus, we will need a Level Converter (since the RTC is 5v and the ADXL345 is 3.3). Hook them up as shown:



Now copy the project for Lab 3a and create a new copy for Lab 3b and open the project.

We will make one alteration to the existing functions. Remove the i2c_int() function call in the rtc_init() function and move it to main (Before the call to rtc_int()). This is better coding since we will be using i2c for two devices, it will be clear that the i2c is initialized before trying to initialize the RTC and ADXL345.

Now we will add functions to talk to the ADXL345. The code already has functions for stop, write, read, start and init, we will be creating the functions necessary to initialize the ADXL345 and the read the X,Y,Z values.

From the datasheet:

I²C

With $\overline{CS}$ tied high to $V_{DD\ I/O}$, the ADXL345 is in I²C mode, requiring a simple 2-wire connection as shown in Figure 8. The ADXL345 conforms to the *UM10204 I²C-Bus Specification and User Manual*, Rev. 03—19 June 2007, available from NXP Semiconductor. It supports standard (100 kHz) and fast (400 kHz) data transfer modes if the timing parameters given in Table 11 and Figure 10 are met. Single- or multiple-byte reads/writes are supported, as shown in Figure 9. With the SDO/ALT ADDRESS pin high, the 7-bit I²C address for the device is 0x1D, followed by the R/$\overline{W}$ bit. This translates to 0x3A for a write and 0x3B for a read. An alternate I²C address of 0x53 (followed by the R/$\overline{W}$ bit) can be chosen by grounding the SDO/ALT ADDRESS pin (Pin 12). This translates to 0xA6 for a write and 0xA7 for a read.

If other devices are connected to the same I²C bus, the nominal operating voltage level of these other devices cannot exceed $V_{DD\ I/O}$ by more than 0.3 V. External pull-up resistors, R_P, are necessary for proper I²C operation. Refer to the *UM10204 I²C-Bus Specification and User Manual*, Rev. 03—19 June 2007, when selecting pull-up resistor values to ensure proper operation.

Table 10. I²C Digital Input/Output Voltage

Parameter	Limit[1]	Unit
Digital Input Voltage		
Low Level Input Voltage (V_{IL})	0.25 × $V_{DD\ I/O}$	V max
High Level Input Voltage (V_{IH})	0.75 × $V_{DD\ I/O}$	V min
Digital Output Voltage		
Low Level Output Voltage (V_{OL})[2]	0.2 × $V_{DD\ I/O}$	V max

[1] Limits based on characterization results; not production tested.
[2] The limit given is only for $V_{DD\ I/O}$ < 2 V. When $V_{DD\ I/O}$ > 2 V, the limit is 0.4 V max.

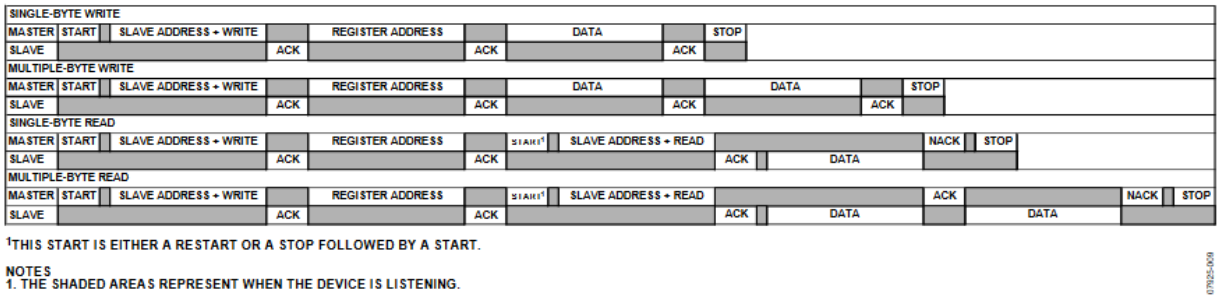


Figure 9. I²C Device Addressing

It has the following registers that are of interest for this experiment:

Register (Hex)	Name	Description
0x00	DEVID	Device ID (should return 0xE5)
0x2D	POWER_CTL	Power-saving features control.
0x31	DATA_FORMAT	Data format control
0x32	DATAX0	X-Axis Least Significant Byte
0x33	DATAX1	X-Axis Most Significant Byte
0x34	DATAY0	Y-Axis Least Significant Byte
0x35	DATAY1	Y-Axis Most Significant Byte
0x36	DATAZ0	Z-Axis Least Significant Byte
0x37	DATAZ1	Z-Axis Most Significant Byte

Refer to the ADXL345 data sheet[1] for further details, make sure to read the S Communications section. Also take note of the $R/\overline{W}$ bit (first bit sent in address byte) and the MB (Multiple-byte bit – 2nd bit sent in address byte).

[1] ADXL345 Data Sheet

Other notes:

1) You must set the Measure Bit to 1 in the POWER_CLT Register of the device.
2) Make the Data_Format register a 0x0b (SPI, Full Res and ±16g range).
3) You can test if I2C is communicating correctly by reading the DEVID register
4) The X, Y and Z data comes in on 10 bits, so you will have to convert the two 8 bit values back into a 10 bit value for proper output. Also note that the 10 bit values will be SIGNED.

To read the x,y,z values back, we will use the Multiple-Byte Read as shown above.

Also note, since we are connecting SDO to GND, we are using the alternate 7-bit address 0x53.

It is suggested that you pass the full 16 bit values of X,Y,Z using the same method used for the RTC hour, minute and seconds.

Your output should be in the format of

mm/dd/yy hh:mm:ss x y z

and should be written to the terminal every 200ms.