The goal of this assignment is to continue using general purpose IO (GPIO) and use i2c. First do the exercises on the *Calendar and Exercises*, and then do this project. **Keep all your work in a github repository.** The grayed parts are optional.

## TMP101

In your kit are two TI, TMP101 i2c temperature sensors. Go to <http://ti.com> and look up its datasheet. On its pinout you’ll see the clock (SCL) and data (SDA) for the i2c interface. There is also an address line (ADD0). It can appear at one of 3 i2c addresses on the i2c bus and the ADD0 line decides which address. What are those addresses? (Hint: check the datasheet.) There is also an ALERT pin which can be programmed to transition when the temperature is above THIGH or below TLOW.

1. Wire up your two TMP101 on the i2c bus so each has a different address. Also wire the ALERT pin to a GPIO port.
2. Use the shell commands to read the temperature of each. Write a shell file to read the temperature and convert it to Fahrenheit. Hint: **temp=`i2cget -y 1 0x48`** assigns the output of **i2cget** to the variable **temp**. Hint 2: **temp2=$(($temp \*2))** multiplies **temp** by two.
3. Use the **i2cset** command to set the temperature limits THIGH and TLOW. Test that they are working.
4. Write a program that sets the temperature limits on each TMP101 and waits for an interrupt on the ALERT pin, then prints the temperature in F. To keep things simple you may use a shell file to set things up.

## TMP006

You also have a TMP006 i2c sensor in your kit. Wire it up and see what you can do with it.

## Etch-a-sketch

Modify your etch-a-sketch program to use the bicolor LED matrix in your kit. The matrix will work off 3.3V.

1. Wire the matrix up to the same bus as your TMP101’s.
2. Use the programs in **exercises/displays/matrix8x8** to set the matrix before modifying your Etch-a-sketch program.
3. Once working, interface the LED matrix to your Etch-a-sketch.

## Extras

Here are some other things you can do to make the project more interesting. Feel free to think of your own ideas.

* Use a couple of potentiometers and the analog in lines to be the knobs controlling your Etch-a-sketch.
* Make your Etch-a-sketch get brighter as the temperature rises
* Make your Etch-a-sketch erase when it gets too hot.

## What to turn in

1. Make a subdirectory in your github repository called **hw03**.
2. Put all your files in the directory, include a **ReadMe.md** that explains what you homework does and how to use it.
3. Document your code.