Progress Report

To do list MCU

* Figure out how to write print statements in mplabx
* Turn on LED (proves I can configure output and set pins as either high or low)
* Blink LED (proves I can select the internal oscillator frequency as well as using delays)
* Read switches (proves I can read digital inputs, use conditional branching, use internal pull ups, and switch debouncing)
* Timer using polling
* Timer using interrupts
* ADC Input
* Read temperature using the digital thermometer and output to screen or maybe LED
* Read battery percentage using ADC
* Talk to Martin about communication method and which we will use to send and recive

To do list VAC

* Document accurate temperature ranges
* Measure power consumption in low med high modes

March 22nd 2022

Today I sat at my desk from 8-12 following tutorials on how to configure the configuration bits by hand. How to declare an GPIO pin as either input or output and how to set that input or output high or low. I followed the first googlium tutorial as well as compared it to the code configurator to figure out what exactly the code configurator was doing. I noticed that a few configuration bits were different. My code would not run if the JTAGEN was set to on. I am not sure what this does. I have to ask Skylear about it. I also discovered that there are two ways to initialize GPIO.

Setting as output

TRISA = 0x0197; // configure RA9 (only) as an output

TRISA = 1536; // sets both RA9 and RA10 as outputs using decimal

TRISA = 1536;

TRISA = ~(1<<9); // different way to configure RA9 as output

Setting as high

LATA = 0x0200; // set RA9 high

LATAbits.LATA9 = 1; // set RA9 high

That’s pretty much it but I am happy I learned how to access the pins and their different registers for configuration