**EE 474 Winter 2017 Lab 1:**

**Tools; Test Blinker; Soldering Pins;**

**Serial Debug**

****

**University of Washington**

**by Ryan Linden, Khalid Alzuhair, and Brandon Ngo**

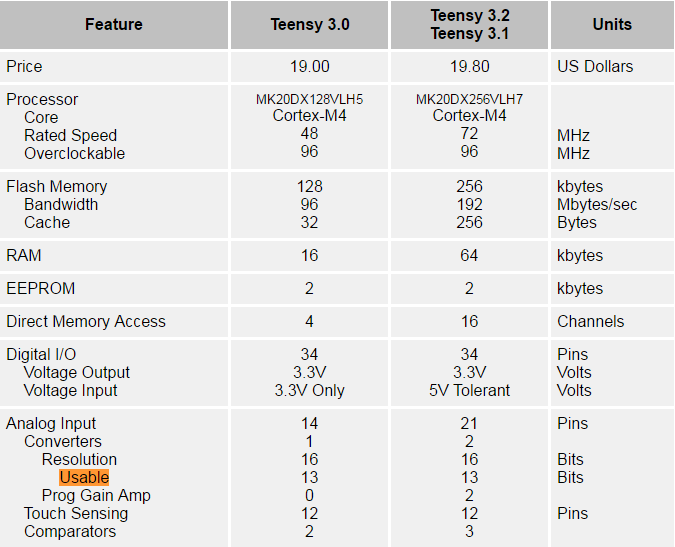
**January 17, 2016**

**Question #1: What happened to the blink rate? Why?**

**The blink rate increased after decreasing the amount in the delay calls after each digitalWrite call to change ledPin to HIGH(on) or LOW(off). The value in delay is the number of milliseconds before continuing the program. Thus, with less delay between each digitalWrite, the blink rate increased.**

**Question #2: how many bits wide is the default ADC data output in the MK20DX256?**

**According to the datasheet, the default ADC data output in the MK20DX256 is 13 bits wide.**



**Figure 1. Snippet of the analog to digital converter technical specifications in Arduino. Notice for the Teensy 3.2/3.1 that there are 13 usable bits for the Analog Input.**

**Question #3: How many bits do you need for the delay, and how do you calculate the delay?**

**The value for delay is determined from reading the specified analog pin using the analogRead() function from the Arduino library. Since the analogRead() function maps input voltages between 0 and 5 volts (3.3 volts for our application in lab) into integer values between 0 and 1023, the amount of bits we need for the delay are 10.**

**To calculate the delay, we calculate 3.3 volts/1024 units which is .0032 volts per unit when changing the potentiometer value. This is due to the analogRead() function from the Arduino library.**

**Question #4: How do you change the sample width?**

**In order to change the sample width, you would use analogReadResolution(). The value inside analogReadResolution() will change the size (in bits) of the value returned by analogRead(). The default value returned by analogRead() is 10 bits (returns values between 0-1023), but can be changed up to a max of 13 bits since the default ADC data output in the MK20DX256 is 13 bits.**

**In order to change the sample width, you would increase the input value in Serial.begin() since that is what sets data rate in bits per second (baud) for serial data transmission.**