

ECE 3724/CS 3124 Quiz #8 Solution

- a. (3 pts) Write a sequence of functions calls using `i2c_start()`, `i2c_get(char ackbit)`, `i2c_put(char byte)`, `i2c_stop()`, `i2c_rstart()` calls that will cause the MAX517 DAC to output a voltage of 2.9 V assuming VREF of 5 V.

```
First, compute 8-bit value to produce 2.9V.  
Vout/Vref * 2N = DAC_code; so 2.9V/5V * 256 = 148.48 = 148 = 0x94  
Assume A1, A0 on DAC tied low, so I2C address 0x58
```

```
i2c_start();  
i2c_put(0x58); //i2c address  
i2c_put(0x00); // cmd code to DAC  
i2c_put(0x94); // value to convert  
i2c_stop();
```

- b. (4 pts) Write a C function called `char wait_adc()` that first selects the AN1 input channel, delays for 20 microseconds using `DelayUs(20)`, then begins a loop where it starts a PIC18 ADC conversion, waits for the conversion to end, only exits the loop if the input voltage is greater than 3.5 V assuming VREF of 5V. Assume that the A/D is configured for left justification and that we are only interested in the upper 8-bits of the result.

```
First, compute 8-bit value that represents 3.5V.  
Vin/Vref * 2N = ADC_code; so 3.5V/5V * 256 = 179 = 148 = 0xB3
```

```
char wait_adc() {  
    CHS2 = 0; CHS1 = 0; CHS0 = 1; // select channel AN1  
    DelayUs(20); // delay for 20 microseconds  
    do {  
        GODONE = 1; //start a conversion  
        while(GODONE); //wait for end of conversion  
    }  
    while (ADRESH < 0xB3); // wait until ADC value exceeds threshold  
}
```

- c. (3 pts) Write C code that configures the PIC18 to select the AN1 ADC channel, and uses the internal ADC RC oscillator clock. The AN3 input must be configured as VREF+, with AN2 as VREF-. The input channels AN0 must also be an analog inputs, I don't care what the other input channels are. Configure for LEFT justification, and ensure that the ADC is turned on. For clarity, use individual bit assignments such as ADFM and ADON, and do not assume any default bit values.

```
CHS2 = 0; CHS1 = 0; CHS0 = 1; // select channel AN1  
ADCS2 = 0; ADCS1 = 1; ADCS0 = 1; //internal ADC RC oscillator  
ADFM = 0; // left justified  
PCFG3 = 1; PCFG2 = 0; PCFG1 = 0; PCFG0 = 0; // AN3-AN0 config  
ADON = 1;
```