

ECE 3724 Test #4 – Summer 2005 - Reese

You may use a calculator and the provided reference materials. If a binary result is required, give the value in HEX. For any required I2C functionality, use subroutine calls to *i2c_start()*, *i2c_rstart()*, *i2c_stop*, *i2c_put(char byte)*, *char i2c_get(char ackbit)*. If you use *i2c_put*, you must pass in as an argument the byte that is to be written to the I2C bus. If you use *i2c_get*, you must pass in an as argument the bit value to be sent back as the acknowledge bit value.

Part I: (76 pts) You must answer all of these questions.

- a. (10 pts) Write a function called *char adc_chk (char chan)* that performs a PIC18 conversion on ADC input *chan* and returns a '1' if $2.0\text{ V} \leq \text{adc_val} \leq 3.0\text{ V}$, where *adc_val* is the upper 8-bits of the 10-bit PIC18 ADC value. The function returns a '0' if *adc_val* is out of this range. Assume $V_{\text{ref+}}$ is 5 V, $V_{\text{ref-}}$ is 0V and that the ADC is already configured, with left justification. The *chan* value can be 0 to 3, with '0' selecting AN0, '1' selecting AN1, '2' selecting AN2, and '3' selecting AN3. You must delay for 20 μs after selecting the channel using *DelayUs*.

- b. (10 pts) Write a loop that causes the MAX 517 to output a voltage of VREF (or as close as you can get), and decrease all of the way to 0 V in steps of 1 LSB (least significant bit). You must use I2C functions when causing the MAX517 to perform a conversion.
- c. (10 pts) Write a C code fragment that reads three bytes starting at location 0xBF08 in the 24LC515 serial EEPROM into *char* variables a, b, and c. You must use the I2C function calls listed at the start of the test. Assume the A1, A0 lines of the EEPROM are tied high. When you begin reading bytes, you must read them in the same I2C transaction.

- d. (10 pts) Use the PWM module of the PIC18 to generate a square wave with a period of $500\ \mu\text{s}$ and a high pulse width of $100\ \mu\text{s}$ (the low pulse width is $400\ \mu\text{s}$). Show the calculations that you use to calculate the needed register values. Then write code for *main()* that configures the PWM for this operation. Your *while(1){}* loop should be empty. Use an FOSC value of 10 MHz.
- e. (6 pts) How many Timer1 tics are in 5 ms assuming an FOSC = 16 MHz and a prescale of 8?
- f. (6 pts) When we measured pulse width from a pushbutton switch using capture mode, pin CCP1, and Timer1, we used two interrupt flags in the ISR. What were these interrupt flags and what was each one used for?

- g. (8 pts) Assume the code used in lab to measure the pulse width of a pushbutton switch. On the falling edge (pushbutton pressed), the capture register captures the hex value 0xFFF0 from timer1. On the rising edge (pushbutton released), the capture register captures the value 0x1000 from Timer1, with THREE timer1 overflows between the falling and rising edge captures. Give the answer in decimal.
- h. (10 pts) Explain EITHER the operation of a 3-bit successive approximation ADC or a 3-bit flash ADC. For both ADCs, use $V_{in} = 3.2 \text{ V}$ and $V_{ref} = 4 \text{ V}$. If you explain the successive approximation ADC, you have to give the internal VDAC voltage used at each comparison step, and list all steps. If you explain a flash ADC, you have to give the number of comparators and resistors, the output value (1 or 0) of all comparators. For either ADC, you have to give the final 3-bit output code.
- i. (6 pts) In lab, a sine wave was generated from a 64-entry lookup table using a periodic interrupt. If we wanted the sinewave to have a frequency of 50 Hz, then what does the interrupt interval have to be? Give the answer in microseconds.

4. Write a C code fragment that will initiate a STOP condition on the I2C bus and wait for it to be finished using the PIC18. (do not use the `i2c_stop` function, I want to know what is inside the `i2c_stop` function!)

5. How many bit times are there in the I2C transaction to the MAX517 DAC for a conversion? Count the start and stop conditions each as one bit time.

6. Assume the PIC18 A/D is configured with a $V_{ref+} = 4V$, and a $V_{ref-} = 0V$, and that two successive reads of the ADC are done, with the first value returning 0xA0, and the next value returning 0x80 (only the most significant 8 bits are read). What is the change in voltage?

7. In using PWM to control either voltage or current, what is varied? What is kept constant?

