1300 Henley Court
Pullman, WA 99163
509.334.6306
www.store.digilentinc.com

Basys MX3 Course Questions Answer Sheet

Lab 1a Q & A

1. Listing 7.1 and Listing 7.2 present two equivalent methods of setting PIC32 pins as inputs.
    1. Do a quick internet search and find out what “magic number” refers to in software programming.

    (See https://en.wikipedia.org/wiki/Magic_number_(programming).)

    2. Which listing, 7.1 or 7.2, uses magic numbers?

    (Listing 7.1 uses the constants 1 and zero to represent TRUE and FALSE. Listing 7.2 assigns values to TRISB and ANSEL using only magic numbers.)

    3. What are the advantages and disadvantages of using magic numbers?

    (See https://en.wikipedia.org/wiki/Magic_number_(programming).)

2. Most of the PIC32 processor outputs have a resistor connected in series with the end circuit or device. Using section 31 of the PIC32MX370 datasheet (see References 2), answer the following questions relative to output drive capability. Assume that the processor is powered with VDD equal to 3.3V.

    1. What is the maximum source current for processor pins and what are the stipulated conditions?

    (12 or 15 ma.)

    2. What is the maximum sink current for the processor pins and what are the stipulated conditions?

    (12 or 15 ma.)

    3. What is the maximum output (source) current to each LED0 through LED7.

    (0.79 ma – ohms law.)

    4. What is the maximum input (sink) current from each of the seven-segment LEDs?

    (2.6 ma – ohms law.)

    5. What is the maximum current that can be supplied by the PIC32 to all IO pins?

    (200 ma.)

3. Using the schematic diagram shown in Fig. A.3 for LED0 through LED7 on the Basys MX3 and Figure 32-1 presented in Reference 2, when Vdd is 3.3V:
    1. Estimate the output voltage when RA0 pin is set high.

       (3.0V.)

    2. Estimate the current output when RA0 pin is set high, assuming that the forward diode voltage drop is 0.7 V.

       (6.9V.)

Lab 1b Q & A

1. What is displayed on the seven-segment LED for the operation X / Y when Y is set to zero?

   (9999.)

2. What does the seven-segment display if all anode outputs (AN0 through AN3) are pulled low at the same time?

   (All digits will display the same value that was programmed to the segment and DP cathode pins.)

3. Using schematic diagram of the Basys MX3 and Figure 31-1 presented in Reference 2 in section 10, when Vdd is 3.3V:
    1. Estimate the output voltage when RA0 pin is set high.

       (3.0V)

    2. Estimate the current output when RA0 pin is set high assuming that the forward diode voltage drop is 0.7V.

       (6.9ma)

4. If RB8 is programmed to be a digital output, what affect will pushing BTNR have on the output signal?

   (The value see on the RB8 pin (as measured at J1 pin 1) will be whatever the processor output for that pin is set to (either 0V or 3.3V) regardless of the state of BTNR. The 10K resistor (R49) between the RB8 pin and BTNR is of sufficient magnitude to eliminated any effects.)

5. The signal BTNC is connected to PIC32 Port F pin 0 and programmed to serve as a digital input or output.
    1. Consider the case when pin RF0 is programmed to be an input and the signal TRIG_1 is a digital output signal from an external device that is either 0 or 3.3V. What effect will pressing the BTNC button have on the processor RF0 pin when TRIG_1 is low and when it is high?

       (The RF0 pin will be at the level of the Trig_1 pin regardless of the position of BTNC.)

    2. Consider the case when pin RF0 is programmed to be an output and the signal TRIG_1 is a digital input signal to an external device. What effect will pressing the BTNC button have on the TRIG_1 signal when the RF0 output is low and when it is high?

BTNC	RF0	TRIG_1
Not Pressed	LOW	0.0V
Pressed	LOW	3.3V/2
Not Pressed	HIGH	3.3V
Pressed	HIGH	3.3V

6. Measure the loop execution time by toggling Pmod connector JA pin 1 at the start of the loop execution using the instruction LATCINV = 0x04. Connect a logic analyzer or oscilloscope probe to JA pin 1 and the ground or common connector to JA pin 5. The loop time is one half the period of the square wave generated at JA pin 1.
    1. What is this time?

       (416.43 us)

7. Is this time expected?

   (Since each of the displays is programmed to have a persistence of 1ms, lighting all displays requires 4ms. The additional 16.43 us can be attributed to processing time.)

Lab 2a Q & A

1. How does your control flow diagram convey the fact that interrupts can occur at any time during execution of the background task?

   (The interrupt service routines are shown as independent (standalone) programs that start and stop at will suspending execution of the infinite background infinite loop. There are no infinite loops in ISRs.)

2. What is the latency from the button press until the LED changes state?

   (2.9832 us.)

Lab 2b Q & A

1. How are you able to accurately determine the number of steps per revolution? How many steps occur in the phase period shown in Fig. 8.1?

   (Set the code to rotate a fixed number of steps. Adjust as necessary until the motor turns one exact revolution or an integer multiple of revolutions.)

   (10 half steps.)

2. From the expression that converts RPM to ms per step, what is the effect of using integer division on the actual stepper motor rotational speed?

   (All speeds will be rounded down after the division. The best accuracy is when the is no remainder when dividing)

3. What is the measured stepper motor maximum rotational speed?

(Under the right condition, 31.87 RPM)

4. Using the pin designated for LCD DB0 on the Basys MX3 processor board (PIC32 RE0), measure the minimum and maximum ISR execution time. Using the MPLAB stopwatch feature and the assembler code for the Timer 1 ISR prologue and epilogue (saving and restoring) times, what is the worst case ISR execution time?

   (Prolog = 42 Inst, Epilog = 43 Inst. ISR Code = 549.84 ns. Total (1.06 + 0.549)us = 1.6115 us)

5. Using the pin designated for LCD DB0 on the Basys MX3 processor board, measure the minimum and maximum background loop. What is the worst-case response time for a user input measured in microseconds?

   (24.922us to 95.631us.)

Lab 3a Q & A

1. How would the LCD display be affected if the application described above is combined with an application that uses interrupts?

   (Interrupts can extend the character writing period but as long as neither the control lines nor data lines change during the interruption, the LCD communications is not affected.)

2. How can the blocking semaphore represented by the Busy Flag be eliminated?

   (Use a time delay for extending the wait period between writing characters and ASCII controls LF and CR. The LCD needs what it needs and published specifications are worst case.)

3. What is the maximum rate at which characters can be written to the LCD?

   (According to Reference 4, the minimum cycle time is 500ns. RS and RW setup times add another 100ns. Since the Busy Flag must be read at least once to see if the LCD is ready, two complete cycles are required. Hence just writing characters, theoretical worst case, can be 830,000 characters per second. Using the logic analyzer and running the solution code, a character is written in 6.3682 us or 157,000 characters per second)

Lab 3b Q & A

1. What are the advantages of using the PMP interface over the bit-banging interface?

   (Once the concept of the peripheral master port bus (PMP) is understood, the application code is more compact that the bit-banging. It is interesting to note that the character writing speed for the bit-banging control is slightly faster than the PMP control. This can be justified by the conservative settings for the bit timing.)

2. What are the advantages of using the bit-banging interface over the PMP interface?

   (This code is not processor dependent.)

3. How would the LCD display be affected if the application described above is combined with an application that uses interrupts?

   (Interrupts can extend the character writing period but as long as neither the control lines nor data lines change during the interruption, the LCD communications is not affected.)

4. What is the maximum rate at which characters can be written to the LCD?

   (According to Reference 4, the minimum cycle time is 500ns. RS and RW setup times add another 100ns. Since the Busy Flag must be read at least once to see if the LCD is ready, two complete cycles are required. Hence just writing characters, theoretical worst case, can be 830,000 characters per second. Using the logic analyzer and running the solution code, a character is written in 6.6355 us or 150,700 characters per second)

Lab 4a Q & A

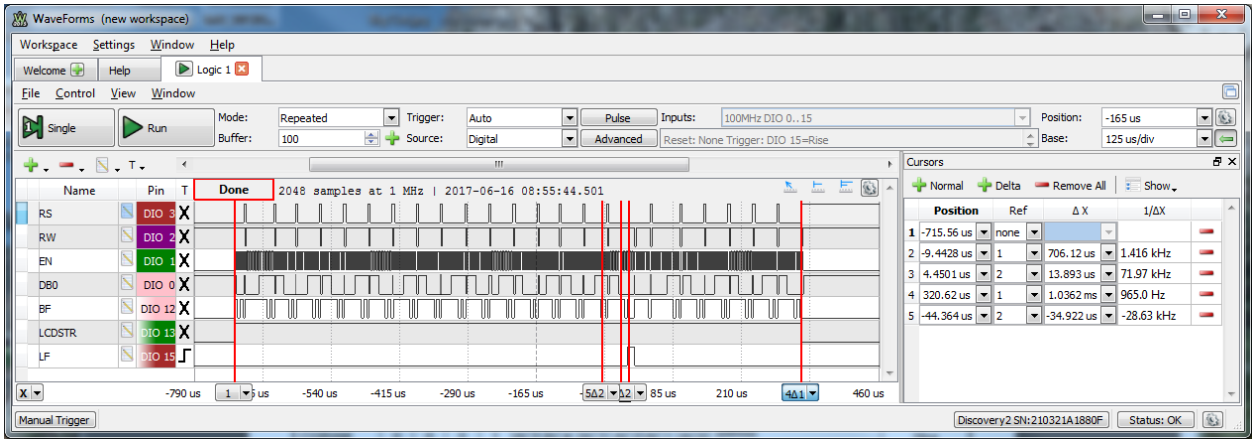
The following oscilloscope captures were made with an Analog Discovery 2. The outputs normally used for controlling the MOTOR driver IC are used to generate timing indicators. AIN1 (DIO 12) is labeled BF that measures the time when the LCD busy flag is detected high. AIN2 (DIO 13) is labeled LCDSTR and is used to measure the time to write the entire LCD string. BIN2 (DIO 15) is labeled LF and indicates the time needed to reposition the LCD cursor.

1. What is the effective data rate of the UART? (Remember to include the period of the stop signal that cannot be measured in testing step 4a.)

   (The theoretical rate is 13.964 bits per second or 1745 characters per second.

2. Based on the data collected in step 4 parts 4.4 and 4.5 of the testing procedure, what is the effective character display rate in characters per second?

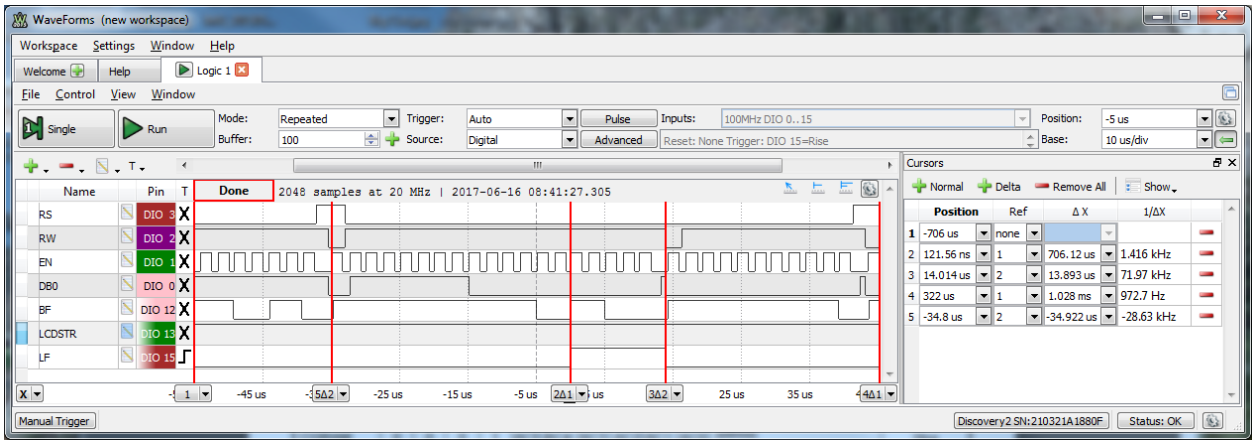
   (



   Based on the Figure above, there are 23 characters written to the LCD in addition to a cursor repositioning command. It requires 706µs to display as a single line of test (16 characters). This results in a writing speed of 22,662 characters per second. Spilling over to the second line reduces the LCD writing speed to 22,196 characters per second. Note that the DB0 trace provides no significant information.)

3. Based on the data collected in step 4 part 4.6, how much does moving the cursor from line 1 to line 2 slow down the LCD character display rate? Justify your answer.

(



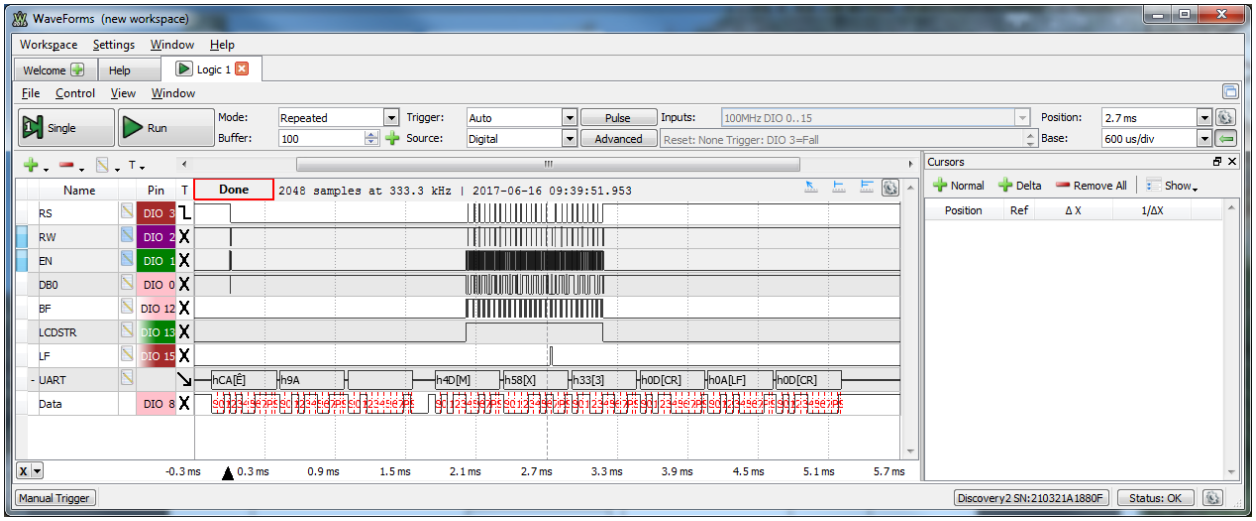
The theoretical worst time to set the cursor address is 39µs.The instrumentation shown above assigns the BIN2 output to "LF" that brackets the code that sets a new cursor position to Line 2. The actual additional time required is on the order of 14µs. The time to repositions the cursor is 2.5 times faster than simply writing a character to the LCD that requires 35µs.

A single line display (16 characters) requires 709.82 µs. Although the additional code to allow instrumentation of program segments adds to the process time, this is small compared to the overall speed of the LCD. The entire 17 character string along with the cursor repositioning requires 1.028ms.)

Additional observations:

The screen capture below shows the UART Tx output along with the string output to the LCD that is generated with the following code listing. The printf statements are executed before the putsLCD statement. However, the code that sends the LF command to the LCD occurs before the UART has sent all of the characters in the second printf instruction. In fact, 8 additional characters continue out of the UART. This is explained by the fact that the PIC32 UART has 8 byte deep FIFO buffers for both transmit and receive.

```
sprintf(textstr,"%d %d %d %d %d %d %d %d  0X%X %uD",
     (sw1>>7 & 0x01), (sw1>>6 & 0x01), (sw1>>5 & 0x01), (sw1>>4 & 0x01),
     (sw1>>3 & 0x01), (sw1>>2 & 0x01), (sw1>>1 & 0x01), (sw1>>0 & 0x01),
      sw1, sw1);
printf("%s\r\n", textstr);
printf("Digilent Basys MX3\n\r");
clrLCD();               // Display sign-on message
LATESET = BIT_8;
putsLCD( textstr );
LATECLR = BIT_8;
```



Lab 4b Q & A

1. Why is it appropriate that the UART getstrU4 function be a background process?

   (At the data rate of 19.2KB, it still requires 572µs to send or receive a complete character. Measurements on the figure above show that the actual character transmission time is 581µs (which agrees well with the theoretical time) or 156 characters per second. Considering that the data is being entered by a human. The average person types between 38 and 40 words per minute -- between 190 and 200 characters per minute or 3.3 characters per second. Hence, in computer time, a human, the computer is waiting a minimum of 97% of the CPU time leaving significant time to complete critical tasks and still check for a new character received.

   If the input is generated by an automated message generator, then we must consider the rate that the UART is checked for new data. Using the data from Lab 4a, we saw that writing to the LCD required 1.036ms. This means that 1.8 characters can be received during the time the LCD is being updated. However, as previously mentioned, the UART Rx has an eight character FIFO buffer extending the time before data is missed to 4.6ms. Considering the dynamics for human generated messages, there is no critical need for a highly responsive preemptive system.)

2. What are the advantages of using serial communications to link to processors?

   (Conductors cost money whether it is across a 2" circuit board or across the Atlantic Ocean. Many serial systems minimize signal corruption due to noise by using electronics or data stream analysis such as parity bits. When interfacing two processors or a processor to a sensor, the biggest savings is the number of processor pins that are used. For systems that consist of interfacing to many different devices, IO pins conservation becomes important.)

3. What are the disadvantages of using serial communications to link to processors?

   (The price of conserving IO pins and conductors is paid for by the reduction of the data communications speed.)

Lab 4c Q & A

1. Why does SPI normally have higher data transfer rates?

   (No hand shaking is required.)

2. Is it necessary for the SPI SCK signal to have a 50% duty cycle?

   (No. The signal sampling instance is indicated by a clock signal transition, not a clock signal level.)

3. Can text data be sent using SPI communications?

   (Bits are bits. It is up to the application to interpret the meaning of the bits.)

4. Can SPI master be configured for only receiving data without transmitting data?

   (Sure. The only requirement for the master to send or receive a SPI message is the generation of the clock signal. If you not sending and data to the slave device, you don't need a wire connected to the processor pin. The PPS processors parts will allow you to not specify a MOSI pin.)

5. How many slave devices can be connected to the SPI bus?

   (This is most dependent on the number of slave select pins are available. The second limiting factor is the fan out capability of the SPI SCL. This constraint is a function of the length of the SPI wire, the slave device static input capacitance, and the speed that the clock is to operate at.)

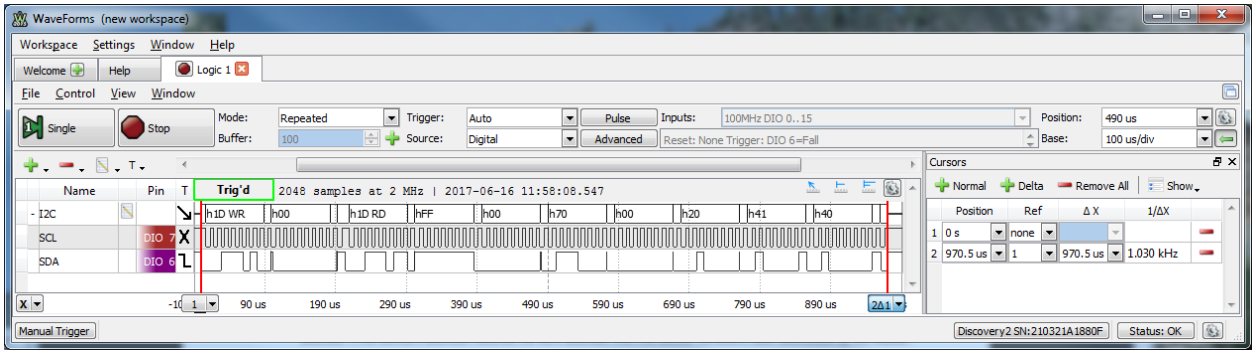
6. What will happen to the SPI communications is an interrupt occurs in the middle of a transmission?

   (Nothing will happen from the master's perspective; the data transfer rate will simply be reduced. See Question 2. However, some slave devices have time-out requirements.)

Lab 4d Q & A

1. Using the waveform captured in step 5 of the testing section, identify all portions of the I2C SDA and SCL signals starting with the START sequence and ending with the STOP sequence.

   (Since the I2C SDA and SCL pins are not brought out to the Analog Discovery Interface, J4, the pins on J7 must be connected to the Analog Discovery separately.



)

2. Compute the effective data rate in bits per second.

   (The data read from the slave device consists of 7 data bytes in addition to the 2 bytes written to the slave for device select and register select along with the one byte slave read making a total of 10 bytes – each requiring 9 bits to include the ACK bit. Hence the result is 102K bits per second.)

3. What effect, if any, would a change in the pull up resistors value for the SDA and SCL have on the I²C bus?

   (The existing 2.2K ohm resistors are selected by the board designer based on the desired operating speed and the anticipated extension of the I2C bus. Lower value pull-up resistors permit longer I2C bus wires and the expense of bus power. See Question 6 regarding termination resistance.)

4. Should there be current (e.g., maximum amperage) specifications associated with these protocols?

   (The maximum specification only applies to the low-level output. When the output is high, the devices simply open circuit the pin supply.)

5. Identify as many things about the circuitry that will limit the maximum rate at which data can be reliably transmitted, and explain why (e.g., what effect might the length of the wires have?).

   (

   1. Number of slave devices.

   2. Input capacitance of slaver devices.

   3. Capacitance of bus wires to ground.

   4. The pull-up resistance.

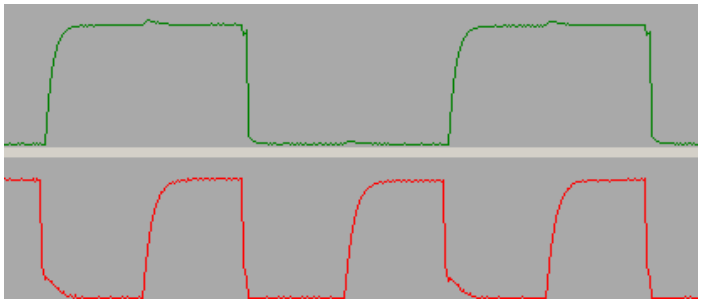
   5. The pull-up voltage.

)

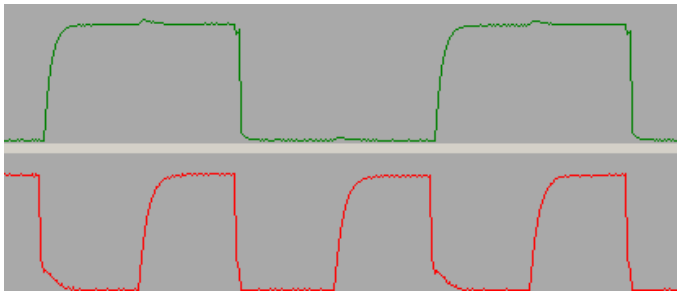
6. What effect might capacitance between data transmission and ground have? Is it possible that the power supply could have an effect, and if so, what?

   (Any change in the ground potential will be coupled into the clock and data lines.

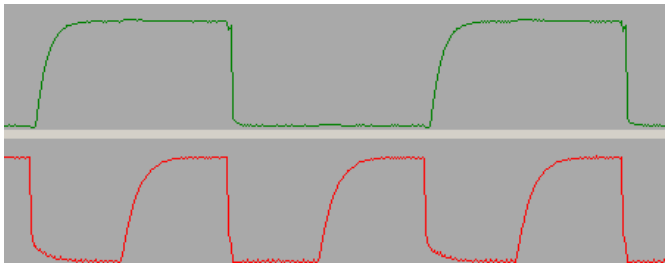
   The main effect is the limitation to the data rate.



SDA (above) and SCL (below) with Rp = 10 kΩ and Cp = 300 pF. The SCL clock runs with 100 kHz (nominal).



The same transfer as above, but this time with a reduced termination resistance (Rp = 2 kΩ, Cp = 300 pF)



The same transfer as above, but this time with a reduced wire capacitance (Rp = 10 kΩ, Cp = 150 pF)

Note that the I2C standard limits Cp to the maximum value of 400 pF. However, with an appropriate termination resistance, it is often possible (although not recommended) to operate I2C buses with higher capacitance.)

7. Explain why you may use I²C, a serial protocol over SPI, another serial protocol.

   (Each SPI slave requires a separate slave select line the total number of IO pins required is equal to 3 plus the number of slave devices. The I2C bus only requires 2 pins provided the number of slave devices does not load the SCL and SDA lines beyond maximum.)

8. What are the strong points of I²C and RS232 protocols compared with each other?

   (RS232 (UART) is capable of simultaneous bi directional communications. Both technologies require 3 wires. UART communications can operate up to 115200 BAUD where as I2C can operate up to 400K bits per second. The maximum efficiency of the UART is 80% whereas the efficiency of I2C approaches 90% as the number of message bytes grows. The UART requires that the bits in the data byte be uniformly spaced whereas the I2C clock signal has only a minimum pulse width constraint.)

9. Describe the Basys MX3 board orientation noted in the three testing assignments.

   (The value range of the X and Y axis is from 0 to 4096. The Z axis varies as the speed of raising or lowering the board changes.)

Lab 5a Q & A

1. What IrDA protocol does your IrDA control device use?

   (Depends on remote control device used.)

2. What is the carrier frequency?

(Depends on remote control device used.)

3. How many different control bytes does your IrDA device generate?

   (Depends on remote control device used.)

4. What type of error checking does your device use?

   (Depends on remote control device used.)

5. What code is generated if a key on the remote control device is held depressed for 10 seconds?

   (Depends on remote control device used.)

Lab 6a Q & A

1. After plotting the data in Phase 2 testing part c, describe the resulting curve.

   (My curve is identical to Fig. 8.3 showing that it requires more and more energy to keep increasing the motor speed.)

2. Explain whether Input Capture, Timer 2, or Timer 3 should be set to the higher interrupt level.

   (Timer 3 should be assigned the higher interrupt level because the accuracy of the period measurement. The PWM output is controlled in hardware and is independent of a Timer 2 interrupt.)

3. If Input Capture, Timer 2, and Timer 3 are set at the same priority level, explain which interrupt should be set to the higher sub interrupt level.

   (In theory, Timer 3 for the same reason that was stated in Question 2. In reality, the chance of both interrupts waiting to be served at the same time is very small and the interrupts will be serviced in a first come – first serve basis.)

4. Explain the difference in interrupt operation between question 2 and question 3.

   (The sub priority level will only be enforced if both interrupts occur in the same core clock period.)

5. Consider the case when the motor draws more current than can be supplied by a single H-Bridge driver and the outputs from A1 must be in parallel with B1 and A2 in parallel with B2. What effect on motor operations does changing the primary output compare registers (OCxR) instead of the secondary output compare registers (OCxRS)?

   (None since both PWM outputs use a common source clock.)

Lab 6b Q & A

No questions asked.

Lab 7a Q & A

1. Since the computed poles for the digital oscillator lie exactly on the unit circle, what affect do you expect truncation and round off errors to have on the oscillator output?

   (Truncation always results in rounding down. Hence poles will move toward the center of the unit circle.)

2. How does the statistical integer rounding code used in the *gen_tones* function shown in Listing A.4 help prevent signal fading? (Hint: See what happens when you comment out that portion of that code and run the program.)

   (Rounding will help keep the poles closer to the unit circle, ie. Magnitude 1.0.)

3. Why is it preferable to synthesize a sine function rather than a cosine function?

   (A sine function always starts at zero level output whereas a cosine function always starts at maximum amplitude resulting is a step in voltage output. This step generates harmonics.)

4. What precautions are necessary if a signal of more than one frequency is generated by adding outputs?

   (The amplitude of the resulting combination of sine functions can cause the PWM period to go beyond the period value or to go negative.)

5. What conclusion do you draw from the data recorded in Table 8.1?

   (

On Switch	Osc. Frequency	IIR Filter Coefficients			Peak Amplitude	Measured Frequency	Frequency Error
		a_0	a_1	a_2			
SW0	500Hz	0x322	0X7F62	0X4000	**± 1.19V**	516Hz	0.032
SW1	1500Hz	0x462	0x7ECA	0X4000	**± 1.19V**	1495Hz	-0.003
SW2	2500Hz	0x5A0	0x7E01	0X4000	**± 1.19V**	2494Hz	-0.002
SW3	3500Hz	0x6DB	0x7D06	0X4000	**± 1.19V**	3473Hz	-0.008
SW4	4500Hz	0x814	0x7BDA	0x4000	**± 1.19V**	**4424Hz**	**-0.017**
SW5	5500Hz	0x94A	0x7A7D	0X4000	**± 1.19V**	5474Hz	-0.005
SW6	6500Hz	0x0AD	0x7FF9	0X4000	**± 1.33V**	6526Hz	0.004
SW7	7500Hz	0x1E2	0x7FC7	0X4000	**± 1.19V**	7458Hz	-0.006

Conclusions:

1. Coefficient a0 does not change over the frequency range

2. The amplitude does not change over the frequency range

3. The maximum frequency error 0.032

4. Frequency error decreases as the frequency increases

)

Lab 7b Q & A

1. Theoretically, how much time is required to fill one data buffer?

   (32 / 16,000 = 2 ms.)

2. Theoretically, what is the maximum rate that the LCD and 7-Segment display can be updated?

   (From Question 3 of Lab 4a, the LCD can update is about 1.5ms + 4.0ms = 5.5ms.)

3. In reality, what is the maximum rate that the LCD and 7-Segment display can be updated?

   (From measurements – 3.48ms)

4. Why would one want to update the displays at a rate slower than the theoretical maximum speed?

   (The display would be blurry.)

5. How many times will a buffer be filled during the time needed to display the results for computing the FFT of one block of data? What are the potential problems here?

   (Almost 2. If the FFT is computed using a partially filled buffer, the results will be inaccurate.)

6. How can your answer to question 5 be resolved?

   (Using double buffering always ensures that the computations are completed using unchanging buffer data.)