

GE 423 Mechatronics Homework Assignment #1
Spring 2009, Due In Lecture Wednesday February 4th
Most answers should be typed. C code included. Graphs, etc. can be hand drawn if you wish.

1. Read Chapters 1-5 in “Teach Yourself C”. You will probably also have to look at Chapter 11 if you are not already familiar bitwise operators. Read about “RS-232” at Wikipedia. Also at the bottom of the “RS-232” Wikipedia page go to the external link: *Serial Port Basics* (<http://www.acumeninstruments.com/Support/documentation/SerialPortBasics/index.shtml>). Read the EZ430 Users Guide http://coecsl.ece.uiuc.edu/ge423/datasheets/MSP430Ref_Guides/ez430userguide.pdf. Keep in mind that we will be swapping out the F2013 processor board for the F2012 Board.
2. Solder your microcontroller board, solder your breakout board and run a demo program. You will need to do most of this work in lab and during office hours.
 - a. Modify the plastic housing of your EZ430 board and solder a connector on the board. You will be given details on how to do this during Lab #1.
 - b. Solder the EZ430 Breakout board as follows:
 - i. Serial Communication Module:
 1. Solder on the MAX3232IDWR chip. (Given to you during Lab #1)
 2. Solder two 6 inch wires to pins 2 (Transmit Line = TX) and pin 5 (Ground Line = GND) of your DB9 serial connector. Next, connect the wire you just soldered to pin 2 (TX) to the TX1 pin of connector SV1 (very bottom left side of board). Finally, solder pin 5’s wire (GND) to the GND pin of SV1.
 3. Solder the five 0.1uF capacitors needed for the MAX3232IDWR.
 4. Solder a wire from P1.7 to the TX1(TTL) input of the MAX3232IDWR.
 - ii. Input Module:
 1. Solder the 8 switch block to your board. One side of each switch must be soldered to GND. Solder two wires to connect 2 of the switches to the I/O lines P2.6 and P2.7.
 - iii. Output Module:
 1. Solder 4 LEDs to your board. The LED’s short wire is its cathode and this should be connected to GND. See <http://en.wikipedia.org/wiki/LED>. Limit the current to the LED with a 220 ohm resistor. Connect the input of these in series resistor-LED circuits to the I/O Lines P1.0, P1.1, P1.2, P1.3.
 - c. Run the EZ430 project creator found at V:\msp430\ProjectCreator\EZ430ProjectCreator.exe or http://coecsl.ece.uiuc.edu/ge423/datasheets/MSP430Ref_Guides/exe/EZ430ProjectCreator.exe. (Also, if for some reason you cannot run the executable file on your home PC you can very easily create your new project manually. Instructions are given at http://coecsl.ece.uiuc.edu/ge423/datasheets/MSP430Ref_Guides/exe/manualProject/howto.txt. Build and run the project in IAR Workbench. LED 1 should be blinking and “Hello” should be printing to the serial port. Launch HyperTerminal (under the Start Menu: Program->Accessories->Communications->HyperTerminal) and set it up for COMx (check which COM your cable is connected to) at 9600 bit/second. You should then see “Hello” printing to the PC’s screen. If you are running Vista on your home computer you will need to download and install HyperTerminal. http://coecsl.ece.uiuc.edu/ge423/datasheets/MSP430Ref_Guides/exe/hpte63.exe .
3. Find the result of the following C statements. First do the calculations by hand and show your work. Then try these calculations on your microcontroller to verify the answers. An easy way to display the result is to send the value over the serial port using the “UART_printf” function. Remember that each character sent over the serial port takes 10/9600 seconds. Why is that?
 - a. `x = 143 & 81;`
 - b. `x = 0x143 & 0x81;`
 - c. `x = 0x2362 | 0x8cf3;`
 - d. `x = 0xac65 >> 5;`
 - e. `x = 3856 << 3;`
 - f. `x = 6292 & (7 << 3);`
4. What are the five basic types in C? See the “Teach Yourself C” book.
5. Read section 7.3 of the TMS320C6713 (DSP used for our Robots) C reference data sheet (http://coecsl.ece.uiuc.edu/ge423/datasheets/C6713Ref_Guides/OptimizingCCompiler.pdf). How many bytes does this processor use for each of the following variable types: int, long int, unsigned int, short int, char, double, float?

6. Same as in question 5 but now for the microcontroller, read section “Data Representation” of the MSP430 C reference (http://coecsl.ece.uiuc.edu/ge423/datasheets/MSP430Ref_Guides/EW430_CompilerReference.pdf). How many bytes does this processor use for each of the following variable types: int, long int, unsigned int, short int, char, double, float? *We will NOT be using floats and doubles on the F2012 microcontroller. There is not enough program flash memory (only 2Kbytes) to store the software routines needed to implement floating point math. Also the calculations are very slow.*
7. In Lab #1 a function `get_switchstate()` was given to you that returned the state of four switches on the TMS320C6713 DSK board. Write a function for the EZ430 board that serves the same purpose. The function should be passed `void` parameters and return a `char`. (i.e. `get_switchstate(void)`) Return a `char` instead of an `int` because we always have to keep in mind that there is only 128 bytes of RAM on the MSP430F2012 so we need to conserve memory where ever possible. This function will return the state of the two switches wired to P2.6 and P2.7. The possible return values then will be 0, 1, 2, or 3. You will have to read the “Digital I/O” section of the MSP430 users guide http://coecsl.ece.uiuc.edu/ge423/datasheets/MSP430Ref_Guides/MSP430x2xx_usersguide.pdf to find how to set P2.6 and P2.7 as general purpose inputs with internal pull-up resistors enabled. (Also setup P1.1, P1.2, and P1.3 as outputs so you can turn on and off the other 3 LEDs. Perform all these initializations in `main()`) Then use this function in a program that changes the default project creator program slightly. Instead of printing just “Hello” to the serial port, print four different messages to the serial port depending on the state of the two switches. Also blink a different LED depending on the switches state. The example C files found at http://coecsl.ece.uiuc.edu/ge423/datasheets/MSP430Ref_Guides/Cexamples/c/ are also helpful in figuring out how to do this assignment. Turn in your C code for this problem and demo your working code to your TA to get credit for this problem.
8. Make the 4 LEDs soldered to your microcontroller board chase each other. Are you too young to remember the “old” Knight Rider car <http://www.knightrideronline.com> or the “old” Cylons from the “old” BattleStar Galactica <http://it.youtube.com/watch?v=61ybqkaS478?>
9. If an oscilloscope was connected to the transmit line on a standard UART (i.e. the 9 pin connector on the back of most PCs, or the transmit line of your microcontroller), what would be displayed if the baud rate was set to 38400 bits per second and only two ascii characters, “R” and “3”, were sent in sequence?
10. Given the following DSP/BIOS application specifications, draw a *time loading diagram* for the application’s processes. One CLK process runs every 2ms for 0.3 ms. One PRD process runs every 4 ms for 0.9 ms. One SWI process with priority 2 is posted by the CLK process every 7 ms and runs for 1.5 ms. A second SWI process with priority 1 is posted by the PRD every 14 ms and runs for 3 ms. The PRD objects have been assigned the SWI priority of 3. Notes: open the DSP/BIOS configuration tool and select SWI to see how DSP/BIOS prioritizes individual SWIs and PRDs. Also note that none of these processes start at time 0 seconds. They wait until their first sample period to attempt to launch their process.

F2012 Play-Time: (These Items are not graded nor required)

1. Start reading about the ADC10 (analog to digital converter) unit of the F2012.
 - a. Read the internal temperature sensor and print the result to HyperTerminal.
 - b. Read an external voltage source in the range 0 to 3.3V and print to HyperTerminal. Be careful here, the F2012’s inputs cannot handle voltages greater than 3.3V or less than 0V. Make sure to scope the input voltage before you connect it to the F2012’s ADC.
2. Flash LEDs in different patterns depending on how long a switch is held in the low state.