

NAME(S) _____

CS 341 – Lab 4
Computer Architecture and Organization
Memory Mapped I/O

Equipment: Arduino UNO microcomputer, PC with Arduino IDE installed, and a USB cable.

The ATMEGA processor on the Arduino UNO board implements a concept called memory mapped I/O. Instead of using a separate I/O address space and special assembly language instructions such as `in` and `out` to access I/O device registers like the Intel architecture does, the I/O device registers are mapped into memory space. In the ATMEGA Harvard architecture, I/O devices are mapped into the data memory space – not the program memory space. The 64 normal I/O device registers are located in data memory space from location 0x20 to 0x5f. There is also room reserved in the data memory space for 160 extended I/O registers from location 0x60 to 0xff. RAM starts at 0x100.

For each I/O pin on the board (you can find pin out of the ATMEGA368 in <http://arduino.cc/en/Reference/Atmega168Hardware>), there is a mode bit that controls whether it is configured as an input or output. If it is an input, another bit represents its state when it is HIGH as a one and when it is a LOW as a zero. If it is an output, another bit controls its state with a one for HIGH and a zero for LOW. If we know the address and bit that is mapped to the mode and data state for a pin, we can read or write the state of the pin without using the normal library functions. We set a pointer to the address.

If we want to read an input pin's value, we dereference the pointer to read the byte, and use a bit-wise operation to mask the appropriate bit.

If we want to set or reset an output pin's value, we must change the state of only that single bit. We dereference the pointer to read the byte, use bit-wise operations to change the state of the appropriate bit, and dereference the pointer to write the byte back.

The memory addresses used for digital pins 2 through 13 are located between 0x20 and 0x2f. Write a sketch with a setup function to display that set of memory locations in HEX. Then, use the library functions to set pin 13 as an output and set it HIGH. Display the memory locations again. Then, use the library functions to set pin 13 LOW and display the memory locations again.

Study those three versions of the state of data memory for the 16 locations designated above. See if you can determine which bits were changed by the library functions. In particular, find the address and data bit for the output state of pin 13. Write down the address and bit mask here for your report:

Address: 0x _____
Bit Mask: 0x _____

Now, add to your sketch some code in the loop function to setup a pointer to the location in memory you determined above. Dereference the pointer to read the byte, flip the state of the appropriate bit with a bit-wise exclusive-or using the mask for the bit that you determined above, dereference the pointer again to write the byte, and delay for about 1 second (1000 milliseconds). If you have done it correctly, the LED on the board should be flashing on and off at the rate you set with the delay call.

To get credit for this assignment, show the TA after you have your sketch flashing the LED on your breadboard via memory mapped I/O writes to the appropriate data memory location and bit. If you have time, figure out the address and bit mask for other digital output pins numbered 2 through 12. Demonstrate with an LED and resistor on your breadboard (like in Lab 1) that you can use memory mapped I/O to flash an LED attached to one of those pins.

As a team, write your lab report to explain what you did, how you did it, and what you learned about interfacing hardware to a microprocessor and its software (the “sketch”). Attach the source code for your sketch. Turn in your report at your next lab session.

____ / 10