**Laboratory 1 - Introduction to Microprocessors: ‘Hello World’**

**Introduction:**

This assignment introduces you to the basic tools and devices that you will use throughout the rest of the semester:

1. The Arduino microcontroller board assembly. The board contains the ATMega168 microcontroller and several other components that make it a self-contained package which can be interfaced directly to other hardware and programmed from a PC.
2. You will also learn how to interface the board to the PC over a USB connection and work on writing a simple program for the ATMega168 microcontroller. The programs are written in a language very much like C using an Integrated Development Environment (IDE).

**Procedure:**

This lab consists of two parts A and B:

1. Several conceptual questions based on your study of the Arduino board and IC datasheet. Each member of your team must answer these questions independently first and get them examined.
2. A programming exercise. Each person must write an algorithm for the solution in the space provided. This can be in ‘flowchart’ format. You must get your algorithm examined independently.
Both team members work together to connect your board to the PC and setup the IDE. Write the program, compile and download to the board. Demonstrate its functionality to the TA. No write-up is expected for this part. Write well-documented programs and organize the algorithm into functions as much as possible. Save your program on the PC with a name that is easy to recall. You will definitely re-use a lot of the code in later assignments!

**Part A:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. What is the clock frequency of running the ATMega168 chip as setup on the Arduino board given to you? (Refer to Atmega168 datasheet and Arduino board schematic provided on your desktop PC)
\_\_\_\_\_\_\_\_\_\_\_
2. What component on the Arduino board would you change if you wanted to ‘overclock’ i.e. run the microcontroller at a higher frequency? Identify the physical component on the board and point it out to a TA.
3. What is the maximum safe clock frequency you can use?
\_\_\_\_\_\_\_\_\_\_\_\_
4. What is the maximum input voltage range for the analog *input* pins on the Arduino board?
\_\_\_\_\_\_\_\_\_\_\_\_
5. What are the digital I/O voltage values used by the Arduino board?
\_\_\_\_\_\_\_\_\_\_\_\_

**Part B:**

**Setup:** The Arduino board is programmed over USB and has two power-supply options:
(a) power is drawn from a 9V power supply or (b) power is drawn from the USB cable. You can select between these two options by setting a jumper on the board. For this exercise set the jumper to draw power from USB, if it is not already set.

Connect the board to the PC over USB.

Start up the Arduino Integrated Development Environment (IDE) on the PC and familiarize yourself with it. When you connect your Arduino board to the PC USB port, you should see the LED’s next to the USB connector on the board flash, indicating that a connection has been established. In the desktop PC , the kernel USB driver will detect the connection and you should see /dev/ttyUSB0 available as a connection under Tools->Serial connections in the Arduino IDE. The board ID setting must be set to ‘Arduino Deicimila with Atmega168’

**Program Exercise:** The board has an on-board LED connected internally to Digital I/O pin 13. Identify it on the PCB. This serves as a useful diagnostic indicator when your program is running.

Write your program algorithm here: (1 mark)

 Write a program that does the following:

1. Set pins 12 and 13 for digital output
2. Drive digitial I/O pin 12 high
3. Drive digital I/O pin 13 high.
4. Wait for 100 millisec.
5. Drive digital I/O pin 12 low.
6. drive digital I/O pin 13 low.
7. Wait for 100 millisec.

Make sure that the instructions to drive pin 12 and pin 13 are next to each other in your program code, with no other instructions in between.

**Question 1:** What do you expect to see when you observe the signal on digital I/O pins 12 and 13 on a digital oscilloscope?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Important:*** Please use the provided breakout header wires to attach the oscilloscope probes. NEVER attach probes directly to the pins on the board – you are likely to short adjacent pins if you do connect DSO probes directly to the Arduino pins

**Question 2:**  What is the observed delay between the signal waveforms on pins 12 and 13?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question 3:**  Based on the chip’s clock frequency you have determined in Part A, how many clock cycles elapse between the two simple instructions to drive the pins into a state HIGH or LOW? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question 4:** Although the Arduino board has 14 pins (numbered 0 to 13) which can be used for *either* digital input or output, why is it impractical to use pin 13 for input?

*Hint: Notice the subsidiary function that pin 13 serves on the Arduino board*

/15

/2

/1

/1

/1

A.1+A.2

A.3

A.4

A.5

/4

B.2

/2

B.3

/3

B.4

Total