Kobe Maristela
CPE 301
9/11/19

University of Nevada, Reno

CPE 301 – Embedded Systems Design - Fall 2019

HOMEWORK No. 2 - DUE BEFORE **11:59pm**, September 11

Statement of Purpose

The purpose of this assignment is to help learn how to write declaration statements for the Atmega2560 as well as understand if a statement is valid or not. In addition, this assignment's purpose is to teach us basic operations in working with addresses in which we will be manipulating data from binary and hex.

REMINDER: If you hand in the HW with your statement of purpose on time you will receive full credit for the entire assignment (no matter if right or wrong). The reason for homework is to provide guidance for you while you are studying and learning the course material. Thus, what is really important is that you learn the course material, not that you hand-in perfect homework.

You will submit a pdf file but, there will be no requirement as to format. You may do the HW in pencil, by hand, and scan in the results. You may submit it in any form that is legible in the pdf file. For each HW assignment you must write one or two sentences explaining your understanding of what material (or process) you are trying to learn by answering the question(s). This is the "statement of purpose".

1. Write declaration statements (for Atmega2560 volatile data) for the following.

   a. The variable pointed to by y_addr is an integer. **Int* y_addr;**
   b. The variable pointed to by ch_addr is an unsigned character. **unsigned char* ch_addr;**
   c. The variable pointed to by z is an integer. **Int* z;**
   d. date_pt is a pointer to an integer. **Int* date_pt;**
   e. pt_chr is a pointer to an unsigned character. **Unsigned char* pt_chr;**

2. For the following declarations,

```
int *x_pt, *y_addr; long
*dt_addr, *pt_addr;
double *pt_z;
int a; long
b; double c;
```

Determine which of the following statements is valid.

```
a. y_addr = &a; Valid   b. y_addr = &b; NV     c. y_addr = &c; NV
d. y_addr = a; NV       e. y_addr = b; NV      f. y_addr = c; NV
g. dt_addr = &a; NV     h. dt_addr = &b;Valid  i. dt_addr = &c; NV
j. dt_addr = a;         k. dt_addr = b;        l. dt_addr = c;
m. pt_z = &a;           n. pt_addr = &b;       o. pt_addr = &c;
p. pt_addr = a;         q. pt_addr = b;        r. pt_addr = c;
s. y_addr = x_pt;       t. y_addr = dt_addr;   u. y_addr = pt_addr;
```

3. If a variable is declared as a pointer, what must be stored in the variable?

   a. **Memory Address**

Kobe Maristela
CPE 301
9/11/19

4. if var2 is a variable, what does &var2 mean?

   a. **Reference of var2**

5. Using the sizeof ( ) operator, determine the number of bytes used by your PC to store the address of an integer, character, and double precision number. (Hint: sizeof (*int) can be used to determine the number of memory bytes used for a pointer to an integer.) Would you expect the size of each address to be the same? Why or why not?

   a. **Different data types need more memory**

6. Determine the results of the following operations:

   a. `11001010` b. `11001010` c. `11001010 & 10100101` `| 10100101` `^ 10100101`

7. Write the hexadecimal representations of all binary numbers in Question 6.

8. Determine the binary and hexadecimal results of the following operations, assuming unsigned numbers:

   a. the hexadecimal number 0x0157, shifted left by one bit position
   b. the hexadecimal number 0x0701, shifted left by two bit positions
   c. the hexadecimal number 0x0673 shifted right by two bit positions
   d. the hexadecimal number 0x0057 shifted right by three bit positions

9. a. Assume that the arbitrary bit pattern xxxxxxxx, where each x can represent either 1 or 0, is stored in the integer variable named flag. Determine the hexadecimal value of a mask that can be ANDed with the bit pattern to reproduce the third and fourth bits of flag and set all other bits to zero. The rightmost bit in flag is considered bit 0.

   b. Determine the hexadecimal value of a mask that can be inclusively ORed with the bit pattern in f1ag to reproduce the fifth and seventh bits of flag and set all other bits to one. Again, consider the rightmost bit of flag to be bit 0.

   c. Determine the hexadecimal value of a mask that can be used to complement the values of the first and third bits of f1ag and leave all other bits unchanged. Determine the bit operation that should be used with the mask value to produce the desired result.

10. The BIOS (Basic Input Output Services) controls low level I/O on a computer. When a (PC) computer first starts up the system BIOS creates a data area starting at memory address 0x400 for its own use.

    Address 0x0417 is the Keyboard shift flags register, the bits of this byte have the following meanings:

Bit	Value	Meaning	
7	0/1	Insert	off/on
6	0/1	CapsLock	off/on
5	0/1	NumLock	off/on
4	0/1	ScrollLock	off/on
3	0/1	Alt key up/down	
2	0/1	Control key	up/down
1	0/1	Left shift key	up/down

Kobe Maristela
CPE 301
9/11/19

0 0/1 Right shift key up/down

This byte can be written as well as read. Thus we may change the status of the CapsLock, NumLock and ScrollLock LEDs on the keyboard by setting or clearing the relevant bit.

Write a C function using pointers and bit operators to turn Caps lock on without changing the other bits. NOTE: Do NOT try to execute this on your PC unless you boot it to DOS in real mode.

(Portions of these Questions are modified from: "C++ for Engineers and Scientists" by Gary Bronson, and http://www.scs.ryerson.ca/~mth110/Handouts/bitwise.pdf)