

Homework II

Professor Gary J. Minden

Date Assigned: 09/11/18

Benjamin Streit

EECS 388

Date Due: 09/18/18

1. Suppose there is the following declaration in a C program:

```
1 union U { int A; char Word[4]; } Foo;
2 .
3 .
4 .
5 .
6 Foo.A = 378;
```

What is the value in hexadecimal of Foo.Word[2]?

After the above declaration of the Union U and the variable Foo, the value of Foo.Word[2] in hexadecimal is 00.

2. What are the two forms (syntax) for writing comments in C (C99)?

Comments in C99 can be written in two different ways, they are as follows:

```
1 // single line code comment
2
3 /*
4 multi-line
5 code
6 comment
7 */
```

3. Which of the following identifiers are not keywords?

```
1 do      int      goto      comefrom
2 static  volatile  extern    exponent
```

The following of the above identifiers are not keywords:

```
1 comefrom      exponent
```

4. What is the value (in hexadecimal) and C syntax for the character that terminates a C string?

A proper C String is terminated with the \0 character, this character has a value in hexadecimal of 00.

5. Consider the following switch statement with Foo = 4; and Baz = -3;

```
1 switch ( Foo ) {
2     case 1: { Baz++; break }
3     case 2: { Baz--; }
4     case 3: { Baz = (2 * Foo); break; }
5     case 4: { Baz = 1 }
6 }
```

What is the final value of Baz?

When the switch statement is reached, the value of Foo is 4, consequently **case 4** of the switch statement is executed. **case 4** sets the value of Baz equal to 1. Afterwards, the switch statement is exited. Thus, the final value of Baz is 1.

6. How many times does the following for loop execute? What is the final value of Sum?

```

1  Uint32_t i, Sum;
2  for ( i = 5; i < 12; i++ ) {
3      Sum = Sum + i;
4      i--;
5  }

```

The loops begins with variable of type `uint32_t` set to a value of 5, it's looping condition is that of `i` being less than 12, and it is to increment the value of `i` by 1 upon each iteration. Within the loop, `Sum` is set to the value of itself plus the value of `i`. Next, the value of `i` is decremented by 1. As a result of the value of `i` being decremented and incremented by 1 each iteration, this loop will never reach an `i` value greater than 12, and thus will be infinite. Consequently, the value of `Sum` is indeterminate, that is aside from the fact that the loop was adding values to an already indeterminate value due to `Sum` being uninitialized.

7. When a hardware value might change due to external circumstances or a value might change due to an interrupt, what type modifier do you attach to the variable that might change?

Should a hardware value be open to change due to an external circumstance or a value be open to change due to an interrupt, the `volatile` modifier must be attached to the variable that might change. Example below:

```

1  volatile uint32_t SysTickCount = 0;

```

8. What is the index value of the first element in a C array?

Indices in a C array range from 0 to `n-1`, where `n` is the size of the array. Thus the index value of the first element in a C array is 0.

9. What are the operators (syntax) for (a) logical or, (b) logical and, (c) logical exclusive or, and (d) logical compliment?

The logical operator for `logical or` is as follows:

```

1  ||

```

The logical operator for `logical and` is as follows:

```

1  &&

```

The logical operator for `logical exclusive or` is as follows:

```

1  ^

```

The logical operator for `logical compliment` is as follows:

```

1  ~

```

10. A GPIO output generates 3.3 VDC (Volts Direct Current). The load is 1,500 Ohms (1.5 KOhm). What is the design (expected) current?

With the given values, the design current can be computed using **Ohm's Law**, $I * R = V$. Computation below:

$$I * R = V$$

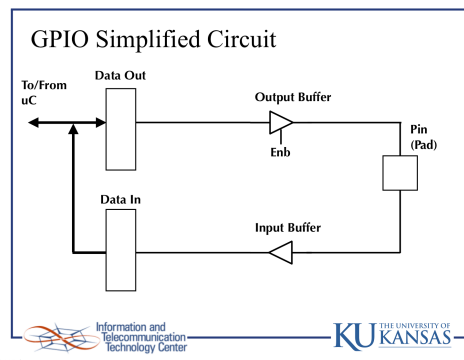
$$I = \frac{V}{R}$$

$$I = \frac{3.3 \text{ VDC}}{1500 \, \Omega}$$

$$I = 2.2 \text{ mA}$$

11. A GPIO pin is set to “Output” mode. Can one read (input) the current value of the GPIO pin?

Yes, should a GPIO pin be set to **Output** mode, one can read the current value of the GPIO pin. Refer to the below image from Professor Gary J Minden’s EECS 388 lecture during explanation. Take note of the location and paths of the **Input Buffer** and the **Output Buffer**. Following the path of the **Output Buffer**, it is observed that after the output’s visit to the **Pin**, it is possible to use the **Input Buffer** from the **Pin**, thus reading the current value sent from the GPIO pin via the **Output Buffer**.



12. Which GPIO port does the “Blinky” task use? Which pin (bit) on the port is used to drive the LED?

According to lecture, the "Blinky" task contains two different write statements relevant to this solution. In plain language, they are as follows:

- 1 Write '1' to PortG<2>... LED is on
- 2 Write '0' to PortG<2>... LED is off

It is clear that the GPIO port used for the task is port G, and the pin on the port used to drive the LED is pin 2.

13. A GPIO output drives a load at 3.3 VDC and 2.75 mA (milli-Ampere) for 15 seconds. How much energy is used?

$$P = IV$$

$$P = (0.00275 \text{ A})(3.3 \text{ VDC})$$

$$P = 9.075 \cdot 10^{-6} \text{ kW}$$

$$E = PT$$

$$E = (9.075 \cdot 10^{-6} \text{ kW})(0.00416667 \text{ h})$$

$$E = 3.781253 \cdot 10^{-8} \text{ kWh}$$

14. A program uses the following subroutine call to read data from a GPIO port (assume the GPIO port has been properly initialized):

```
1 Data = GPIOPinRead( GPIO_PORTH_BASE, GPIO_PIN_6 | GPIO_PIN_5 );
```

If pin 5 has a high voltage (3.3 VDC) and pin 6 has a low voltage (0.0 VDC), what value, expressed in hexadecimal, is returned?

We begin with the declaration that a low voltage is expressed with a 0 and a high voltage is expressed with a 1. The question states that the reader should assume the GPIO port has been properly initialized, meaning all pins are set to 0, before accounting for voltage. Reading the value of the six pins in binary and accounting for voltage returns the following result:

```
1 000010
```

In hexadecimal:

```
1 2
```