Fundamentals of Embedded Systems Design
&
Programming

U.C. Irvine Division of Continuing Education

EECS X497.32

Instructor: Saleem Yamani

Programming Assignment:
LED ON/OFF using switches

Programming Assignment:
LED ON/OFF using using switches

1. Create a code using Robin Robin architecture to blink both LED’s when no switch is pressed just like previous assignment #1.

2. When any switch is pressed, then only blink the LED(s) associated to the switch number. Use the SW1 and SW2 switches.
   See the link provided to get information on the port being used for these switches on the XMega Xplained board. Note the port being used for the Switches will be configured as an input port.
   http://ww1.microchip.com/downloads/en/AppNotes/doc8394.pdf

1. Create a function that controls the blink operation

2. The main task loop should check the switch setting, and make decision whether to blink both LED’s one at a time OR only blink the LEDS associated to the switches that are pressed.

Programming tips

- Make sure the maximum data value used in your program can be stored in the data variable used for storage. The compiler will not flag this as an error and the data stored in the variable will be truncated.

- Example:

  ```
  unsigned char value ;  //Maximum value that can be stored is 256 (0xFF)
  value = 1000 (0x3E8) ;  // Actual stored value will be truncated to 0xE8
  ```

Programming tips

- Delay routine using "for loop" timing

```
delay(unsigned int delay_count)
{
  volatile unsigned int increment;
  for ( increment=0; increment < delay_count; increment++)
   ;
}
```

Processor speed = 16MIPS (16e6 Instructions per sec)

One assembly instruction execution time = 1/16e6 = .0625uSec

Assuming 10 instructions take to execute delay loop, and delay_count value passed to function is 1000

What will be the delay function timing?

.0625uSec/Instruction * 10 Instruction/loop * 1000 loops = 625mSec

Programming tips

- Do not compare an unsigned variable type with a signed variable, otherwise you will get unexpected results

Example:

Bits	Unsigned	Signed
000	0	0
001	1	1
010	2	2
011	3	3
100	4	-4
101	5	-3
110	6	-2
111	7	-1

Programming tips

- Where possible use shift operations instead of multiply and divide operations to speed up computation and save code space.

```
4 = 2 <<1   // same as multiply by 2
8 = 32 >>2   // same as divide by 4
```

Example:

```
unsigned int x,y;
y = x *2 ; // Uses more instructions
y = x << 1;  // Uses less instruction
```

Programming tips

•Bit manipulation operations used extensively in embedded programming.

•OR : |
0x10FF = 0x1000 **|**   0xFF

•AND: &
0x00AA = 0x55AA **&** 0x00FF

•XOR : ^
0 = x ^ x
Complement: ~
0x55AA = ~(0xAA55)