

CSSE2010/CSSE7201

Learning Lab 12

AVR C Programming

School of Information Technology and Electrical Engineering
The University of Queensland

Today

- Microchip Studio C Programming
- AVR C programming exercises

Discussion Question

- What is printed by this line of code:
`printf("%d", (23>>2)<<3);`

Bit Shifts and AVR Code

- Common to use bit-shifts to indicate which bits are being operated on
- Example: Set bit 5 of port B to be an output and other bits to be inputs
 - Assembly language
 - `ldi r16, (1<<5)`
`out DDRB, r16`
 - C equivalent
 - `DDRB = (1<<5);`
- `1<<5` makes it clear bit 5 is being set

Bit Shifts and AVR Code (cont.)

- Example: Set bits 3 and 4 of port B to be outputs whilst keeping the direction of the other bits the same
- Assembly language:
 - `in r16, DDRB`
`ori r16, (1<<4) | (1<<3)`
`out DDRB, r16`
- C equivalent:


Bit masking operations – more than just bitwise ANDs

- Bitwise AND
 - Clear particular bits (e.g. zero out high 4 bits)
 - `var = PINB & 0x0F;`
 - Test particular bits (e.g. is bit 1 set?)
 - `if(PINB & 0x02) ...`
- Bitwise OR
 - Set particular bits (e.g. set sign bit to 1)
 - `PORTC = PINB | 0x80;`
- Bitwise XOR
 - Invert particular bits (e.g. invert low 4 bits)
 - `PORTC = PINB ^ 0x0F;`

Task 1 – Seven Segment Display

- a) Create an AVR C project and type in and build the seven-segment display (SSD) example from lecture 14. The code can be found on the last slide of the lecture and also on Blackboard.
- b) Connect up AVR pins to the IO board pins:
 - Connect IO board SSD inputs (A to G and DP) to port A (You can use 2 x 4-way cables and headers for this. SSD pin A should be connected to the least significant bit of port A.)
 - Connect IO board pin CC (SSD digit select input) to button B0
 - Connect IO board switches 0 to 3 to the lower four pins of port C
- c) Download your program to the board and test it – check that the number you enter on switches 0 to 3 shows up as a digit on the seven-segment display. Press and release button B0 and check that the number shows up on the other SSD digit

Task 2 – Adding Hex Patterns

- a) Modify the task 1 code so that your program can also output values 10 to 15 as hex digits as shown in the following patterns:
- The image shows the hexadecimal digits A, B, C, D, E, and F rendered in a 7-segment display style. Each digit is composed of black segments on a white background, with some segments having a 3D effect.
- b) Build, download and test your program.

Task 3 – Hex Patterns and Binary

- a) Modify the code in Task 2 so that your program also outputs the digit (in binary) on AVR port B. Connect the lower 4 bits of port B to 4 LEDs.
- b) Build, download and test your program.

Task 4 – Addition Results on SSD

- a) Write a program that repeatedly reads the lower 4 bits of port A and the lower 4 bits of port B and adds these two values together and displays (on the SSD, using a port of your choice) the hexadecimal value of the lower 4 bits of the result.
- b) Wire up the IO board and AVR 324 Dev board as required.
- c) Build, download and test your program.