CSSE1000/CSSE7035 - Prac 7

AVR C Programming

Goal

- Practice C programming skills
- Gain some experience with AVR C programming.

See the following topics:

Back to Lab Experiment Index

Preparation

C Tutorials

Revise or complete C tutorials 1 and 2.

Task 1: Accumulator

Write a Windows C program (i.e. using lcc) to model the accumulator design from Prac 6. The program starts with an accumulated total of 0. The program continually prompts the user to enter numbers in the console window. Each time a new number is entered by the user it is added to the total and the new total is printed to the screen. The program exits if and when the user enters a 0.

Print out your code and staple/tape/glue it into your workbook, bring your working program to the session. Comment your code to describe how different C programming constructs (such as loops and ifs) implement different parts of the specification.

An example executable for this program (Windows) can be found here.

Task 2: Combination Lock

Write a Windows C program to model the combination lock design from Prac 6. The program prompts the user to enter two single digits (separately). Once the two digits have been entered they are printed on the screen. If the two digits entered are the correct combination "83" then a message a printed to the screen informing the user that the lock is now open. If the code entered is incorrect, a message is printed informing the user that the lock is closed. If the user enters the code "00" then the program exits.

Print out your code and staple/tape/glue it into your workbook, bring your working program to the session. Comment your code to describe how different C programming constructs (such as loops and ifs) implement different parts of the specification.

An example executable for this program (Windows) can be found here.

Optional preparation: You can work through the tasks described below if desired.

Procedure

AVR C Tutorial

Work through the WinAVR C Tutorial. This will show you how to create AVR C projects, edit, compile and simulate C programs.

Bit Inversion Program

Write a C program for the AVR8515 Project Board to

- Input 8 bits of data from the Port C pins.
- Invert the data.
- Output the result to Port B.

The skeleton code for this program is provided in prac7-1.c. Replace the lines "/* <-YOUR CODE HERE-> */" with your code.

The following IO registers will be used:

DDRB	Port B Data Direction Register
PORTB	Port B Data Register
DDRC	Port C Data Direction Register
PINC	Port C Input Pins Address

Compile and download your program to the board.

Accumulator

Modify your prepared accumulator C program to function on the AVR board and implement the accumulator specification from prac 6. Instead of getting user input from the terminal, numbers are entered using a set of 8 toggle switches (S) and a push button (P). The 8 bit output (L) is wired up to two hex display digits. When the button (P) is pressed, the number entered on the 8 toggle switches (S) is added to the accumulated total (L) and displayed on the hex display.

Combination Lock

Adapt your prepared combination lock C program to work on the AVR board and implement the combination lock specification from prac 6. Instead of getting user input from the terminal, numbers are entered using a set of 4 toggle switches and a push button and the output goes to two LEDs . The specification from prac 6 is below for reference.

The combination lock system has

- a 2 digit 7-segment hex display for showing the input combination
- 4 toggle switches and a push button to enter the code
- one LED indicating that the lock is open,
- another LED indicating that it is closed
- a reset button (in this case the AVR board reset button).

It operates as follows: After a reset, the display is set to "00" and the lock is closed. When the push button is pressed, the value on the 4 toggle switches is saved and the left hand hex digit display is set to that value. When the same button is pressed again, the second digit entered (using the same 4 toggle switches) is saved and is displayed on the left hand hex digit display. The digit that was displayed there moves to the right hand hex digit display. Only the last two digits entered are remembered. If the two digit code displayed is correct, the lock is opened with the "open" LED turned on. (Otherwise, the "close" LED remains lit.)

The correct code for the combination lock for this prac is 83.

- Use Port A pins 3-0 for the left hand display digit.
- Use Port B pins 3-0 for the right hand display digit.
- Use Port C pins 3-0 for the toggle switch inputs.
- Use Port C pin 4 for the push button
- Use Port D pin 0 for the *open* LED
- Use Port D pin 1 for the *closed* LED

Show your working combination lock to the tutor.

Tutor Task

The tutor will ask you to modify your combination lock design to meet a slightly different specification.

Back to Contents

Challenge Task (Complete after being marked)

Modify your combination lock program such that instead of being wired up to toggle switches, it could be wired up to a numeric keypad. The keypad consists of 16 push buttons corresponding to the hex digits 0-F inclusive. Each time one of the buttons is pressed, that particular digit is entered into the combination lock. What are the advantages and disadvantages of using the keypad instead of toggle switches? Discuss your ideas with the tutor at the end of the session.

Assessment

This practical is marked out of 4 and worth 2% of your mark for CSSE1000/CSSE7035:

- Preparation - Programming task completed, documented and tested - ***1 Mark***
- Documentation - All testing results documented in workbook with an appropriate conclusion - ***1 Mark***
- Demonstration - AVR combination lock demonstrated. Part marks given if an earlier exercise is demonstrated - ***1 Mark***
- Tutor Task - Tutor task completed and documented in workbook - ***1 Mark***

Equipment

- Computer
- 12V Power Supply
- AVR8515 Project Board with STK200 Download Cable
- Logic Workstation
- Hookup Wire

Back to Contents

References

- Atmel AVR Resources
- AVR Tutorial
- PonyProg Tutorial
- WinAVR C Tutorial

Back to Contents
