Experiment 11 09

Aim:

To perform AVR I/O programming in C.

Theory:

What is AVR?

AVR MCU's are extremely popular MCU's used in various applications specially in embedded devices. AVR is a 8-bit RISC architecture microcontroller in market since 1996 having on-chip programmable flash memory, SRAM, IO data space & EEPROM. It is the first MCU in market having on-chip flash storage.

Features.

1] 2KB internal static RAM
2] 31 x 8 general purpose registers
3] 32 KB of in system flash programmable memory.
4] 1024 bytes EEPROM
5] Programmable Serial USTART
6] 8 channel, 10 bit ADC
7] One 16-bit timer with separate prescalar, compare & capture mode.
8] 40 pin DIL, 44-pad QFN/MLF & 44-lead QTFP

9] Two 8 bit counters with separate prescalars & comparable modes.
10] 32 programmable I/O lines.
11] In system On-chip boot program
12] master/slave serial interface
13] Programmable timer with seperate on-chip controller.

what are the major reasons for writing program in C?
1) It is easier and less time consuming
2) Easier to modify & update
3) Use available code in function libraries.
4) Portable to other microcontroller with little or no modification.

widely used data types in AVR

Data Type	Size in Bits	Usage
unsigned char	8-bit	0-255
char	8-bit	-128 to +127
unsigned int	16-bit	0 to 65535
int	16-bit	-32768 to 32767
unsigned long	32-bit	0 to 4946 7295
long	32-bit	-2147 48 to +21477
float	32-bit	±1.17e-38 to 3.40

Program 1
Write an AVR to send values 00-FF to port 3.

```
#Include <avr/io.h>
int main (void) {
    unsigned char z;
    DDRB = 0xFF;
    for (z=0; z <=255; z+7) {
        PORTB = z;
        return 0;
    }
```

Program 2
Write an AVR to send HEX values for ASCII characters of 0,1,2,3,4,5,A,B,C,D to PORTB.

```
#include <avr/io.h>
int main (void){ unsigned char z;
unsigned char might[] = "012345ABCD";
DDRB = 0xFF;
  for (z=0; z < 10; z++) {
      PORTB = might [z]; while(1);
      return 0;
}
```

Program 3

Write an AVR to toggle all bits of PORTB 200 times.

```
#include <stdio.h>
#include <avr/io.h>
int main(void) {
  DDRB = 0xFF;
  PORTB = 0xAA;
  unsigned char z;
for (z=0; z<200; z++) {
    PORTB = ~PORTB
  while(1);
  return 0;
}
```