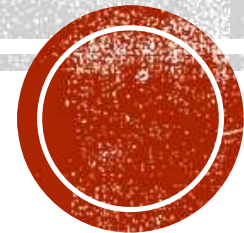




EMBEDDED SYSTEMS

CMPE-453

Department of Computer Engineering



Programming AVR Microcontroller-II

C DATA TYPES FOR AVR C

Table 7-1: Some Data Types Widely Used by C Compilers

Data Type	Size in Bits	Data Range/Usage
unsigned char	8-bit	0 to 255
char	8-bit	-128 to +127
unsigned int	16-bit	0 to 65,535
int	16-bit	-32,768 to +32,767
unsigned long	32-bit	0 to 4,294,967,295
long	32-bit	-2,147,483,648 to +2,147,483,648
float	32-bit	$\pm 1.175e-38$ to $\pm 3.402e38$
double	32-bit	$\pm 1.175e-38$ to $\pm 3.402e38$

The AVR Microcontroller and Embedded Systems Using Assembly And C, 2nd ed. By Muhammad Ali Mazidi, Sepehr Naimi, Sarmad Naimi.

- The mostly used one is unsigned char: 0-255
- 00-FF (Hex)
- Unsigned char is also for String of ASCII characters



EXERCISE-1

Write an AVR C program to send values 00-FF to port B.

Solution:

→ appropriate IO definitions

```
#include <avr/io.h>                //standard AVR header

int main(void)
{
    unsigned char z;
    DDRB = 0xFF;                    //PORTB is output
    for(z = 0; z <= 255; z++)
        PORTB = z;

    return 0;
}

//Notice that the program never exits the for loop because if you
//increment an unsigned char variable when it is 0xFF, it will
//become zero.
```



EXERCISE-2

- Unsigned char is also for String of ASCII characters

Write an AVR C program to send hex values for ASCII characters: 0, 1, 2, 3, 4, 5, A, B, C, D

Solution:

```
#include <avr/io.h>                //standard AVR header

int main(void)                     //the code starts from here
{
    unsigned char myList[] = "012345ABCD";
    unsigned char z;
    DDRB = 0xFF;                   //PORTB is output
    for(z=0; z<10; z++)            //repeat 10 times and increment z
        PORTB = myList[z];         //send the character to PORTB

    return 0;
}
```



EXERCISE-3

Write an AVR C program to toggle all bits of Port B 200 times

Solution:

```
//toggle PB 200 times
#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;
}
```

//standard AVR header

//the code starts from here

//PORTB is output

//PORTB is 10101010

//run the next line 200 times

//toggle PORTB

//stay here forever

Bitwise not



EXERCISE-4

Write an AVR C program to send values -4 to 4 to port B

Solution:

```
#include <avr/io.h>                                //standard AVR header

int main(void)
{
    char mynum[] = { -4,-3,-2,-1,0,+1,+2,+3,+4} ;
    unsigned char z;

    DDRB = 0xFF;                                     //PORTB is output

    for(z=0; z<=8; z++)
        PORTB = mynum[ z] ;

    while(1);                                       //stay here forever
    return 0;
}
```

Run the above program on your simulator to see how PORTB displays values of FCH, FDH, FEH , FFH, 00H, 01H, 02H, 03H, and 04H (the hex values for -4, -3, -2, -1, 0, 1, etc.). See Chapter 5 for discussion of signed numbers.



EXERCISE-5

Write an AVR C program to toggle all bits of Port B 50000 times

Solution:

```
#include <avr/io.h>           //standard AVR header
int main(void)
{
    unsigned int z;
    DDRB = 0xFF;              //PORTB is output

    for(z=0; z<50000; z++)
    {
        PORTB = 0x55;
        PORTB = 0xAA;
    }

    while(1);                 //stay here forever
    return 0;
}
```



EXERCISE-6

Write an AVR C program to toggle all bits of Port B 100000 times

Solution:

```
//toggle PB 100,00 times
#include <avr/io.h>
int main(void)
{
    unsigned long z;

    DDRB = 0xFF;

    for(z=0; z<100000; z++){
        PORTB = 0x55;
        PORTB = 0xAA;
    }

    while(1);
    return 0;
}
```

//standard AVR header

//long is used because it should
//store more than 65535.

//PORTB is output

//stay here forever

important



DELAY

1. Using a simple for loop

A function of crystal frequency

XTAL1

XTAL2 input pins

2. Using predefined C functions

3. Using AVR timers

- `_delay_ms()` or `_delay_us()` in WinAVR
- `delay_ms()` or `delay_us()` in CodeVision



EXERCISE-7

Write an AVR C program to toggle all bits of Port B continuously with a 100ms delay (with for loop)

Solution:

```
#include <avr/io.h>
void delay100ms(void)
{
    unsigned int i;
    for(i=0; i<42150; i++);
}

int main(void)
{
    DDRB = 0xFF;
    while (1)
    {
        PORTB = 0xAA;
        delay100ms();
        PORTB = 0x55;
        delay100ms();
    }
    return 0;
}
```

//standard AVR header

important

//try different numbers on your
//compiler and examine the result.

//PORTB is output



EXERCISE-8

Write an AVR C program to toggle all bits of Port B continuously with a 10ms delay (with predefined C functions)

Solution:

```
#include <util/delay.h>           //delay loop functions
#include <avr/io.h>               //standard AVR header

int main(void)
{
    void delay_ms(int d)          //delay in d microseconds
    {
        _delay_ms(d);
    }
    DDRB = 0xFF;                  //PORTA is output
    while (1){
        PORTB = 0xFF;
        delay_ms(10);
        PORTB = 0x55;
        delay_ms(10);
    }
    return 0;
}
```

