I/O Ports in AVR

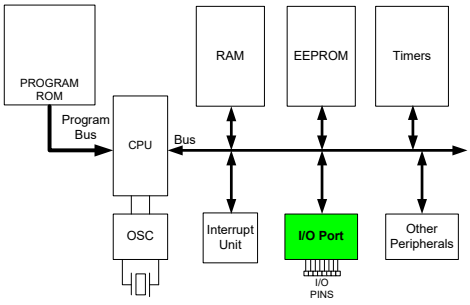
Sepehr Naimi

www.NicerLand.com
www.MicroDigitalEd.com

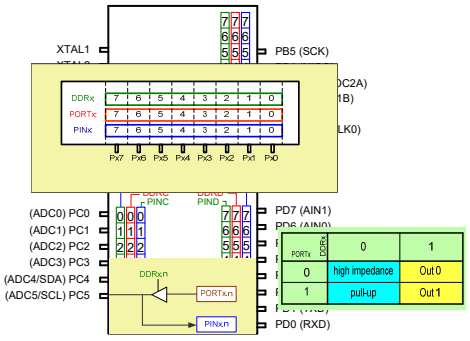
Topics

- AVR pin out
- The structure of I/O pins
- I/O programming

I/O unit in AVR



The structure of I/O pins

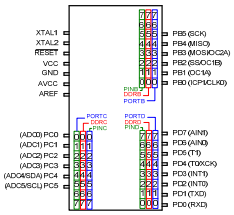


PORTx \ DDRx	0	1
0	high impedance	Out 0
1	pull-up	Out 1

Example 1

- Write a program that makes all the pins of PORTB one.

DDRB: 1 1 1 1 1 1 1 1
PORTB: 1 1 1 1 1 1 1 1

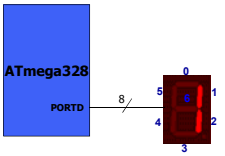


PORTx \ DDRx	0	1
0	high impedance	Out 0
1	pull-up	Out 1

Example 3

- A 7-segment is connected to PORTA. Display 1 on the 7-segment.

DDRD: 1 1 1 1 1 1 1 1
PORTD: 0 0 0 0 0 1 1 0

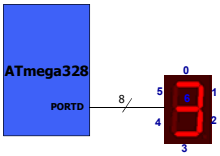


PORTx \ DDRx	0	1
0	high impedance	Out 0
1	pull-up	Out 1

Example 4

- A 7-segment is connected to PORTA. Display 3 on the 7-segment.

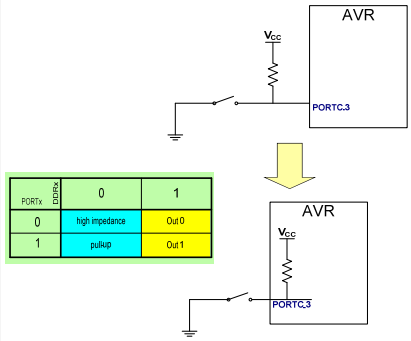
DDRD: 1 1 1 1 1 1 1 1
PORTD: 0 1 0 0 1 1 1 1



PORTx \ DDRx	0	1
0	high impedance	Out 0
1	pull-up	Out 1

7

Example



PORTx \ DDRx	0	1
0	high impedance	Out 0
1	pull-up	Out 1

8

Accessing I/O registers

- Example 1: Write an AVR C program to send value 0xAA to PORTA.

```
#include <avr/io.h>

int main ()
{
  DDRA = 0xFF;
  PORTA = 0xAA;

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

C Programming Review

Data Types

- Use unsigned whenever you can
- unsigned char instead of unsigned int if you can

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	±1.175e-38 to ±3.402e38
double	32-bit	±1.175e-38 to ±3.402e38

Data types (cont.)

- char c;

- long lng;

- int a1;

- unsigned int a;

What is the difference between ***int*** and ***unsigned int***?

Arrays

- Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data.

- Examples:

```
double balance[10];
double balance[5] = {1000.0, 2.0, 3.4, 7.0, 50.0};
```

13

FOR Loop

- A **for** loop is a loop with a built in counter
- General Form:

```
for ( init; condition; increment ) {
   statement(s);
}
```

14

FOR Loop

- Example:

```
for( a = 10; a < 20; a = a + 1 )
{
     printf("value of a: %d\n", a);
}
```

Note: a = a + 1 can be replaced by a++

15