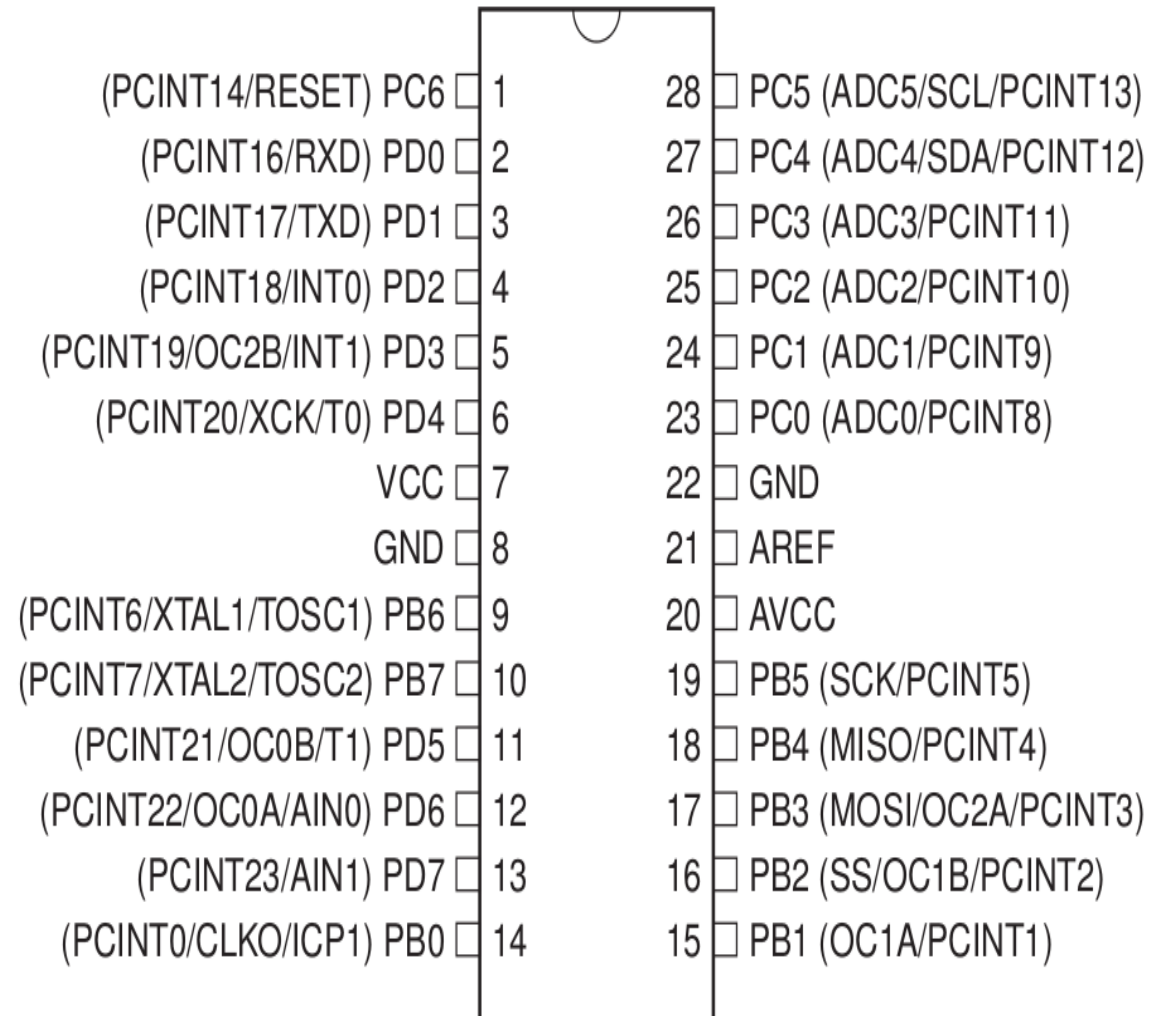


CSE3023.1

Lecture 2 Digital I/O Port Operations in AVR

Pin Configuration

- Pins are organized into 8-bit “Ports”
- PORTB 8 bit “Ports”
- PORTC 7 bits / Ports
- PORTD 8 bit “Ports”



ATmega328P pin mapping

Arduino function

reset
 digital pin 0 **RX**
 digital pin 1 **TX**
 digital pin 2
 digital pin 3 **PWM**
 digital pin 4
 VCC
 GND
 crystal
 crystal
 digital pin 5 **PWM**
 digital pin 6 **PWM**
 digital pin 7
 digital pin 8

PC6 1
 PD0 2
 PD1 3
 PD2 4
 PD3 5
 PD4 6
 VCC 7
 GND 8
 PB6 9
 PB7 10
 PD5 11
 PD6 12
 PD7 13
 PB0 14



28 PC5
 27 PC4
 26 PC3
 25 PC2
 24 PC1
 23 PC0
 22 GND
 21 AREF
 20 AVCC
 19 PB5
 18 PB4
 17 PB3
 16 PB2
 15 PB1

Arduino function

analog input 5
 analog input 4
 analog input 3
 analog input 2
 analog input 1
 analog input 0
 GND
 analog reference
 AVCC
 digital pin 13
 digital pin 12
PWM digital pin 11
PWM digital pin 10
PWM digital pin 9

When using
ISP to program
the chip

Register

- A processor register is a memory space within the CPU itself so that they can be accessed very frequently and fast.
- These registers are linked with the operation of the MCU.

8 bits = 1 byte

Each register consumes only 1 byte

Bit No.	7	6	5	4	3	2	1	0
---------	---	---	---	---	---	---	---	---



Port Operation Registers

- DDRx – Data Direction Register
- PORTx – Pin Output Register
- PINx – Pin Input Register
- where x = GPIO port name (B, C or D)



DDRx - Data Direction Register

- The GPIO pins are the digital I/O pins i.e. they can act as both input and output.
- Now, how do we know that the pin is an output pin or input?
- The DDRx (Data Direction Register) controls one pin with each bit.
- **1** stands for **output** and **0** stands for **input**.

DDRx Register

Bit No.	7	6	5	4	3	2	1	0
Name	DDx7	DDx6	DDx5	DDx4	DDx3	DDx2	DDx1	DDx0
Initial Value	0	0	0	0	0	0	0	0



PORTx - Pin Output Register

- Also one pin per bit
- If configured as an output:
 - 0 -> the pin is held at 0 V
 - 1 -> the pin is held at +5 V

PORTx Register

Bit No.	7	6	5	4	3	2	1	0
Name	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0
Initial Value	0	0	0	0	0	0	0	0



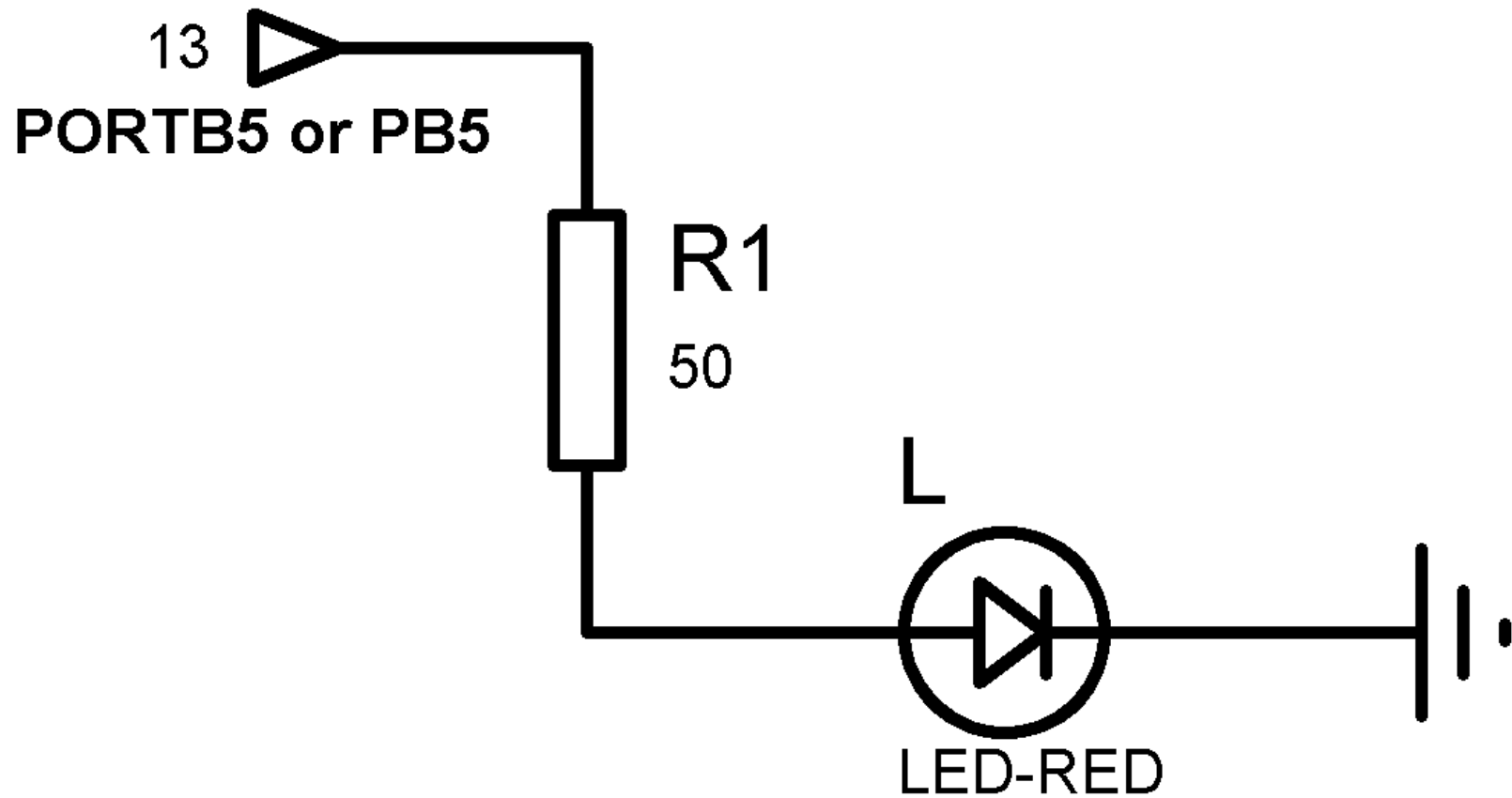
PINx - Pin Input Register

- One pin per bit
- Reading from the register:
 - 0 -> the voltage of the pin is near 0 V
 - 1 -> the voltage of the pin is near +5 V
- If nothing is connected to the pin, then the pin will appear to be in a random state

Hello World

- It's customary to write hello world whenever we are learning something new.
- In case of embedded system.. BLINK a Light Emitting Diode (L.E.D.)...

The Circuit



Blink v. 1.00

```
#include <avr/io.h> // for register name
#include <util/delay.h> //for _delay_ms built-in
function

int main(void)
{
    DDRB = 0b00100000; // Set PB5 as ouput
    PORTB = 0x00; //clear PortB
    while(1) //Run for-ever
    {
        PORTB = 0b00100000; // Turn on
        _delay_ms(1000); //wait for 1 second
        PORTB =0b00000000; // Turn off
        _delay_ms(1000); //wait for 1 second
    }
}
```

Blink v. 1.10

```
#include <avr/io.h>
#include <util/delay.h>
int main(void)
{
    DDRB |= 1<<DDB5; //PB5 as output
    PORTB = 0x00;
    while(1)
    {
        PORTB|=1<<PORTB5; //turn on
        _delay_ms(5000);
        PORTB&=~(1<<PORTB5); //turn off
        _delay_ms(5000);
    }
}
//Good tutorial on bit shift operations
//
http://www.robotplatform.com/howto/blinker/blinker\_8.html
```

Blink v. 1.20

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

```
#include <util/delay.h>
```

```
int main(void)
```

```
{
```

```
    DDRB |= _BV(DDB5); //PB5 as output
```

```
    PORTB = 0x00;
```

```
    while(1)
```

```
    {
```

```
        PORTB |= _BV(PORTB5); //turn on
```

```
        _delay_ms(5000);
```

```
        PORTB &= ~_BV(PORTB5); //turn off
```

```
        _delay_ms(5000);
```

```
    }
```

```
}
```

Blink in Arduino Processing

```
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  pinMode(led, OUTPUT); // initialize the digital pin as an output.  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```