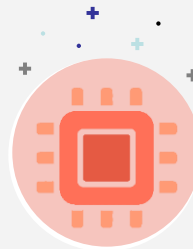




1403/08



Microcontrollers

Lecture 3:

I/O Ports in AVR

By: M.Razeghizadeh

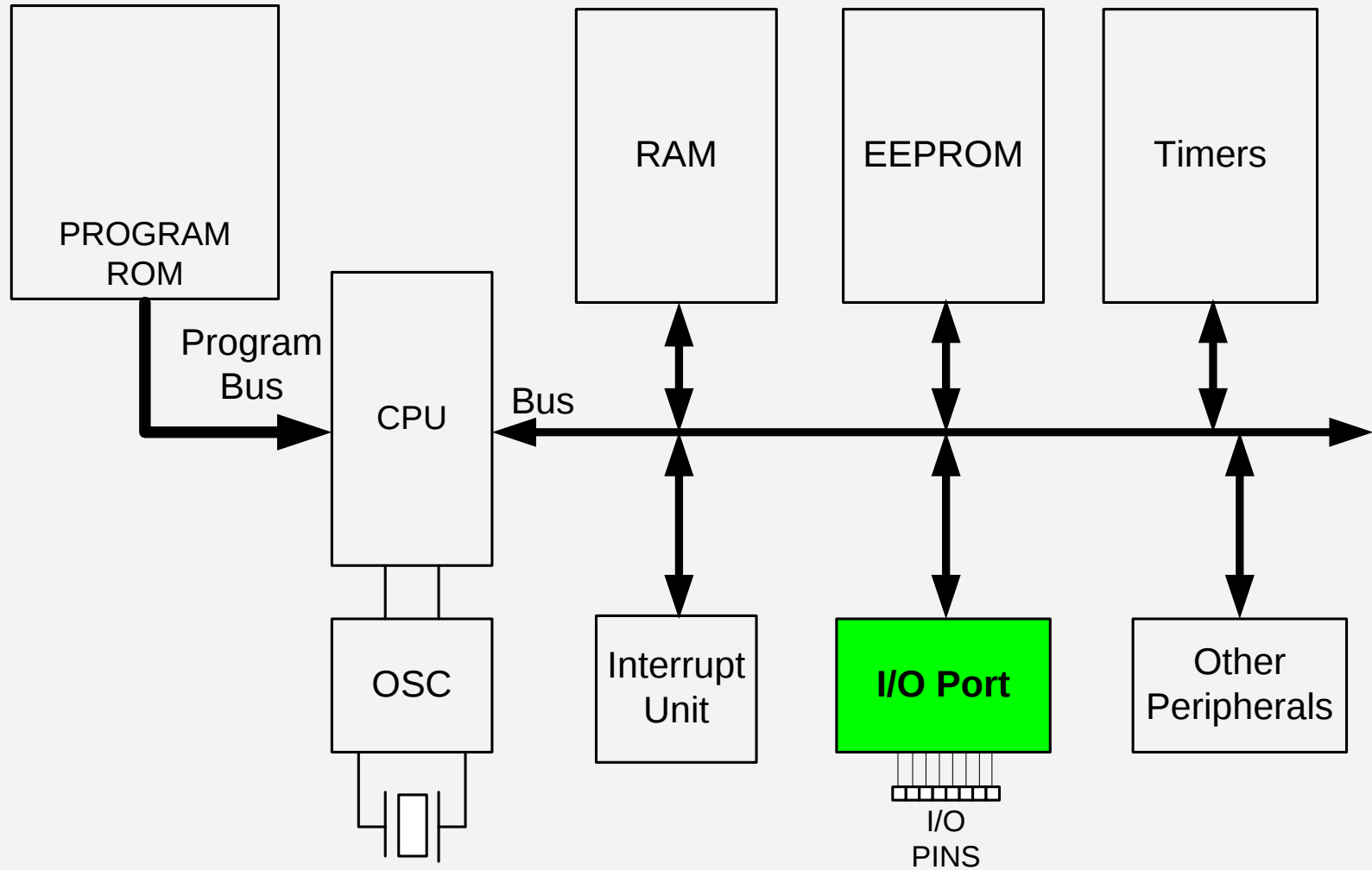
[Start now!](#)



Contents of This Slide

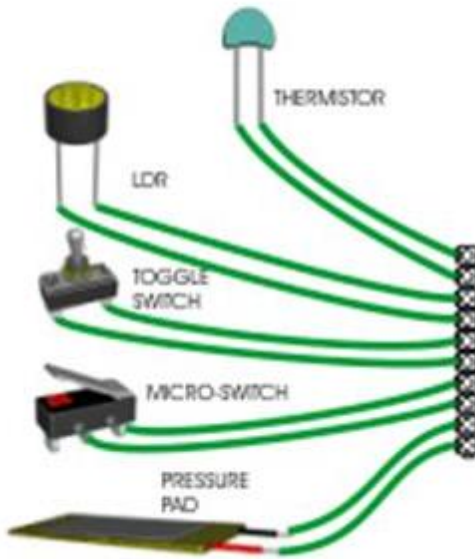
- AVR pin out
- The structure of I/O pins
- I/O programming
- Bit manipulating

I/O unit in AVR



ATmega328 pinout

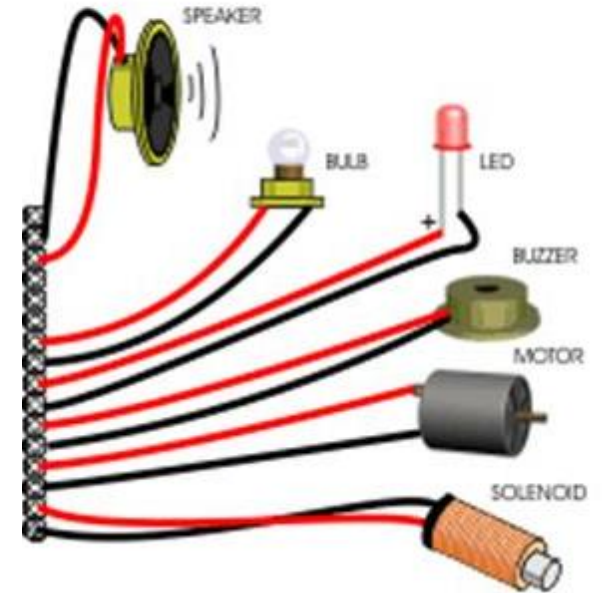
INPUT



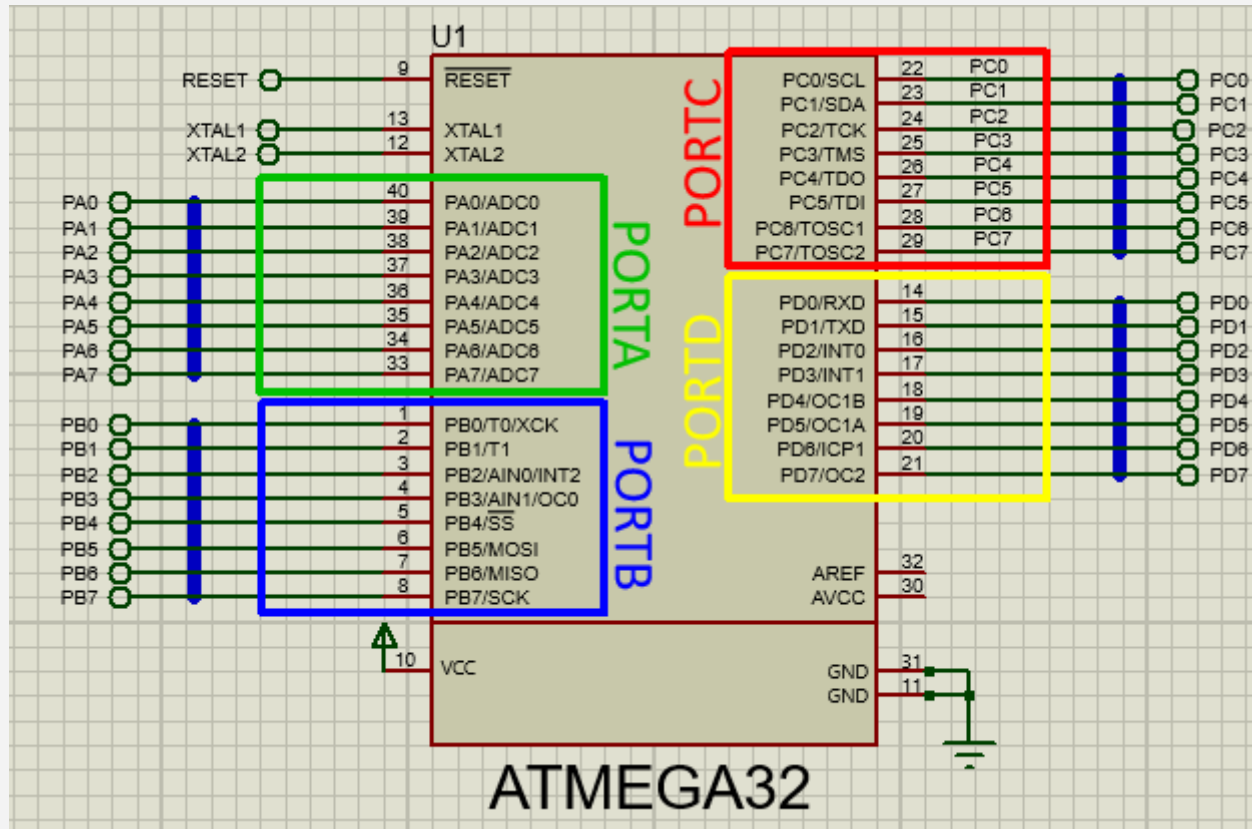
PC6	1	28	PC5
PD0	2	27	PC4
PD1	3	26	PC3
PD2	4	25	PC2
PD3	5	24	PC1
PD4	6	23	PC0
VCC	7	22	GND
GND	8	21	AREF
PB6	9	20	AVCC
PB7	10	19	PB5
PD5	11	18	PB4
PD6	12	17	PB3
PD7	13	16	PB2
PB0	14	15	PB1



OUTPUT



ATmega32 pinout



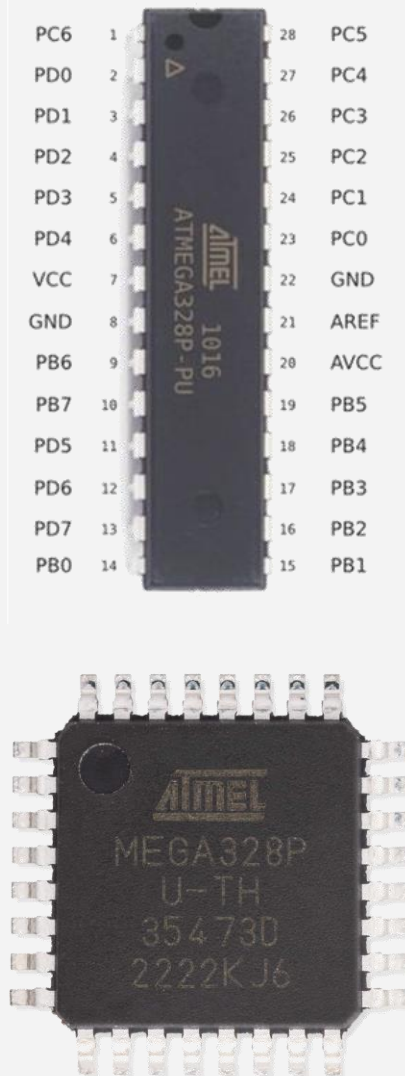


Number of Ports in Some Avr Family Members

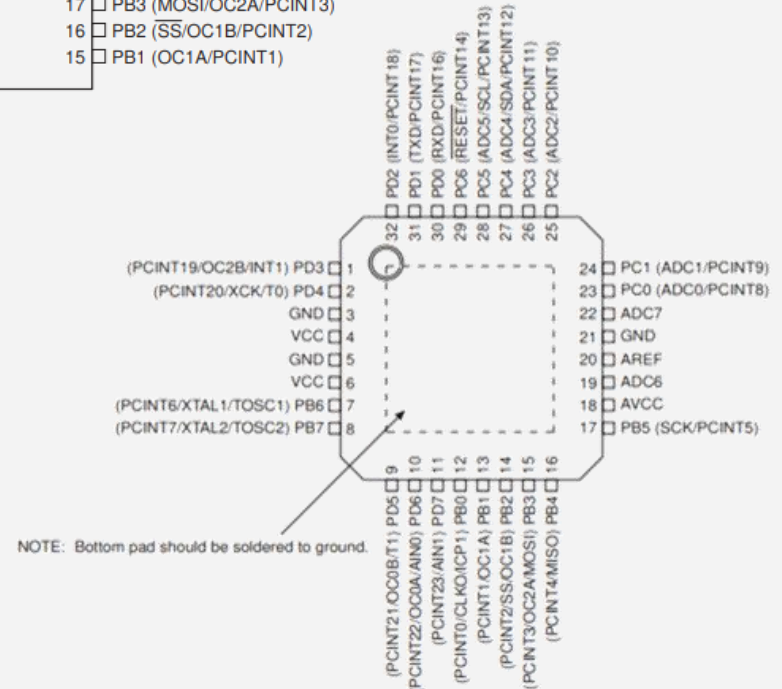
Pins	8-pin	28-pin	40-pin	64-pin	100-pin
Chip	ATtiny25/45/85	ATmega8/328	ATmega32/16	ATmega64/128	ATmega1280
Port A			X	X	X
Port B	6 bits	X	X	X	X
Port C		7 bits	X	X	X
Port D		X	X	X	X
Port E				X	X
Port F				X	X
Port G				5 bits	6 bits
Port H					X
Port J					X
Port K					X
Port L					X

Note: X indicates that the port is available.

I/O unit in AVR

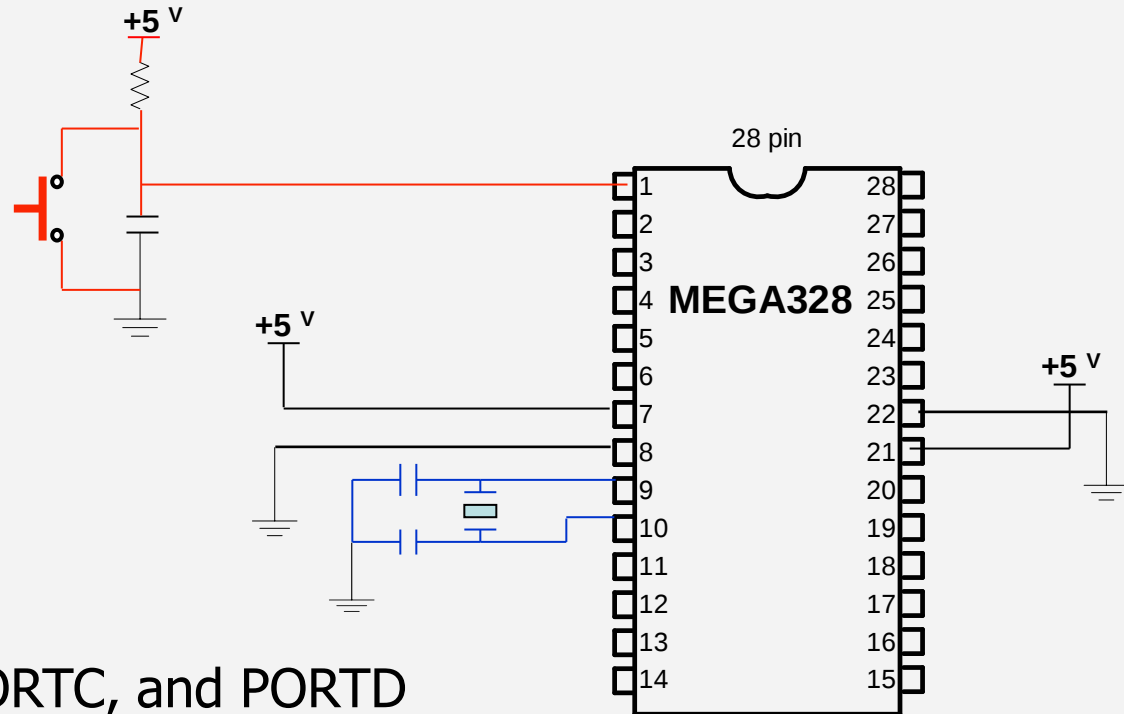


(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)



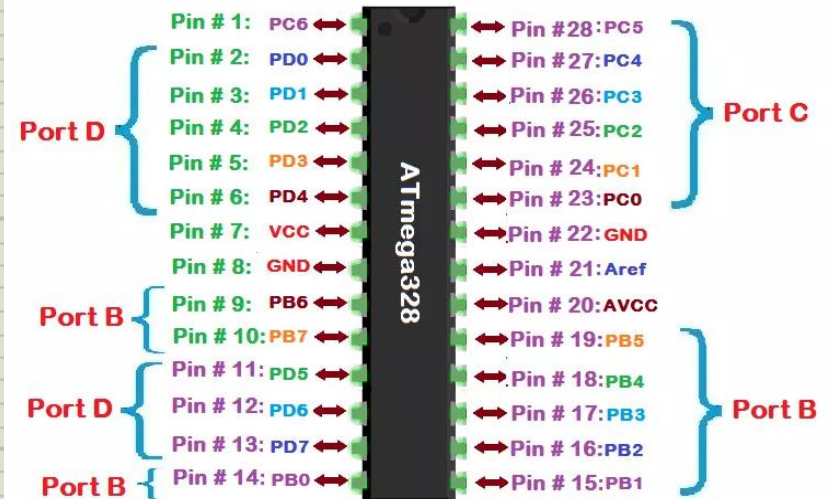
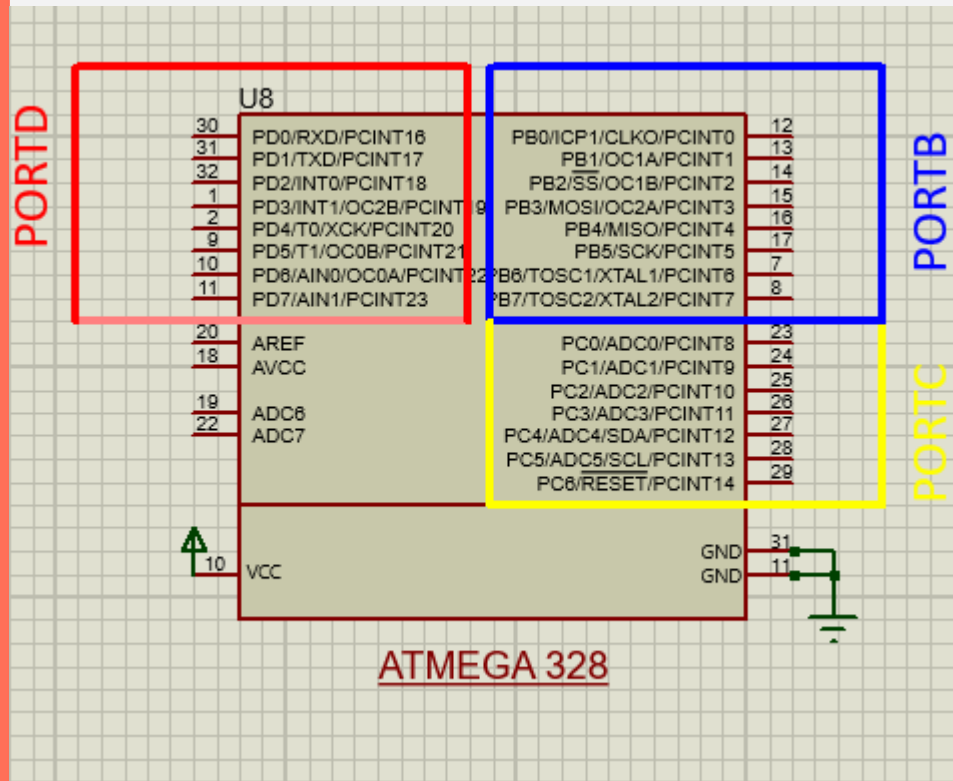
ATmega328 pinout

1. Vital Pins:
 1. Power
 - VCC
 - Ground
 2. Crystal
 - XTAL1
 - XTAL2
 3. Reset
2. I/O pins
 - PORTB, PORTC, and PORTD
3. Internal ADC pins
 - AREF, AVCC, ADCn

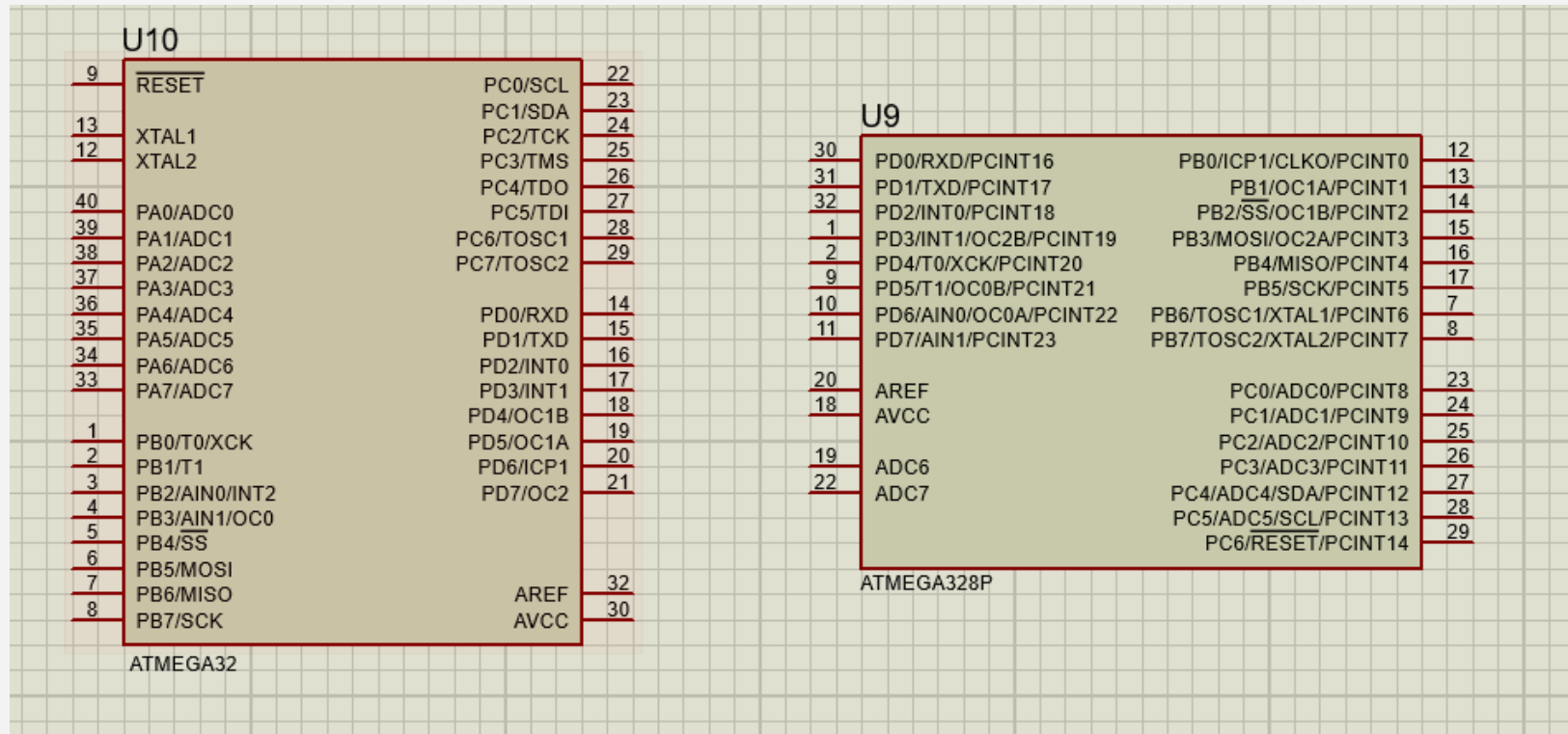


		28 pin		Pinout	
(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)		
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)		
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)		
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)		
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)		
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)		
VCC	7	22	GND		
GND	8	21	AREF		
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC		
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)		
(PCINT21/OC0B) PD5	11	18	PB4 (MISO/PCINT4)		
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)		
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)		
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)		

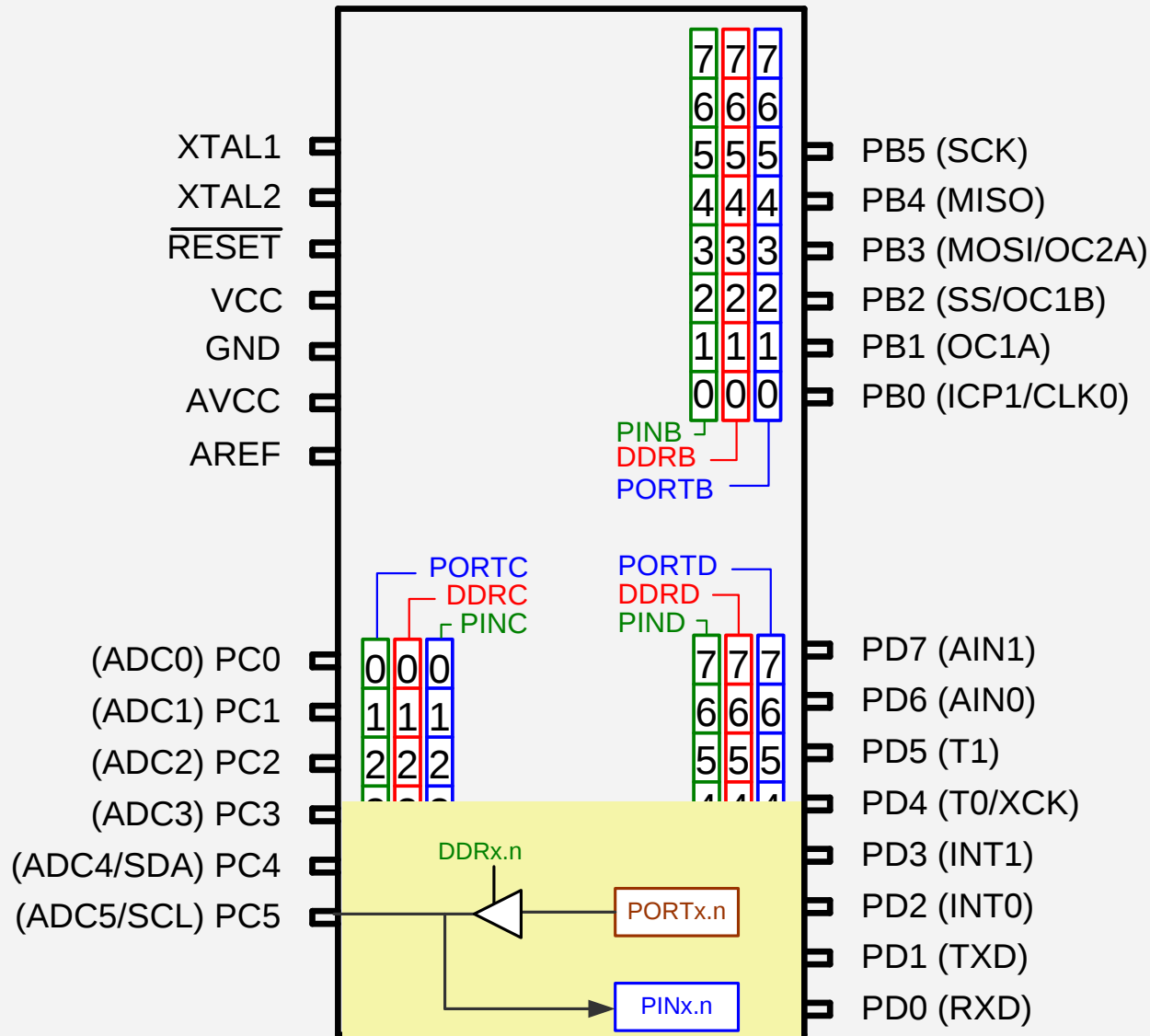
ATmega328 pinout



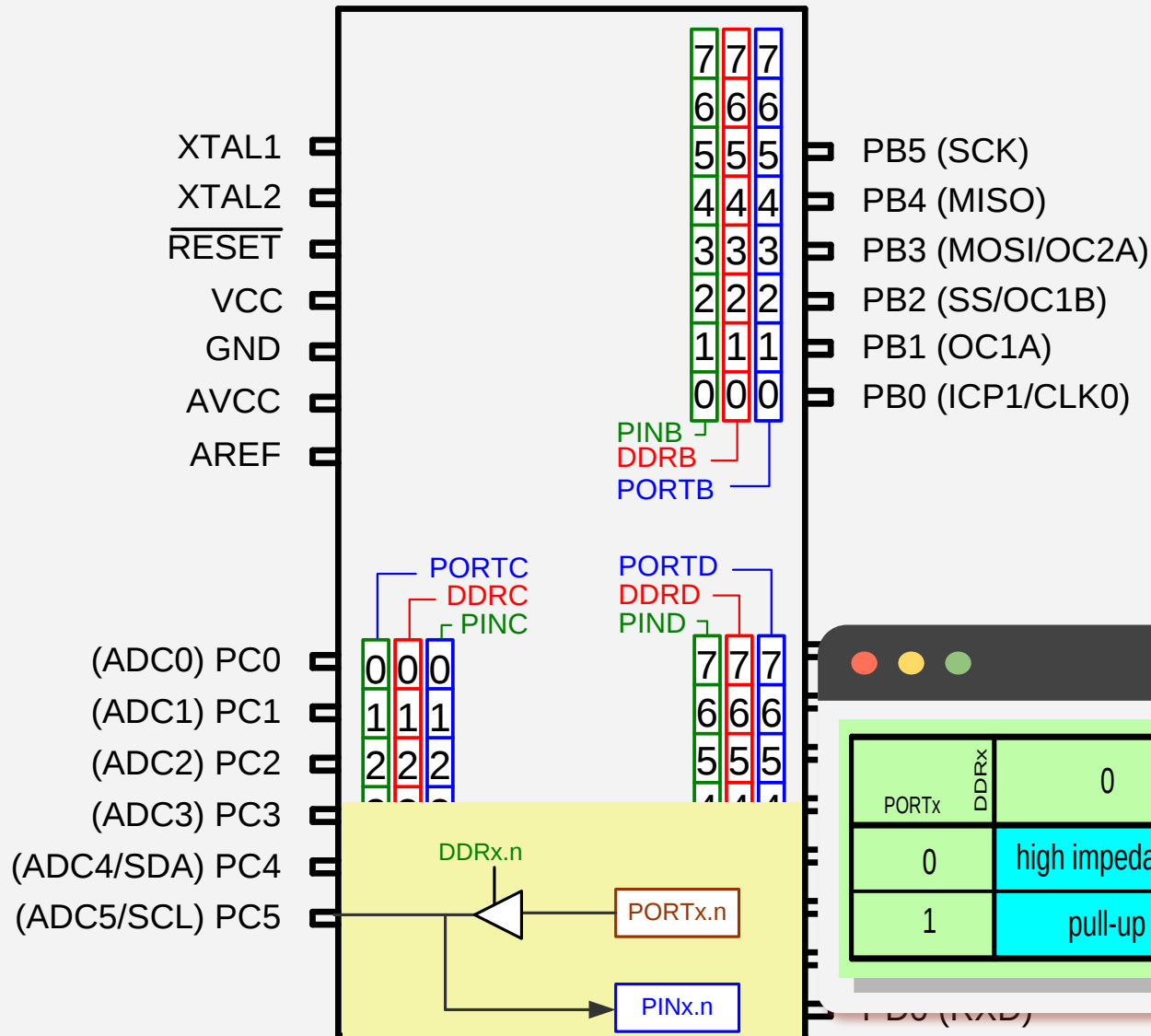
ATmega328 pinout in Proteus



The structure of I/O pins

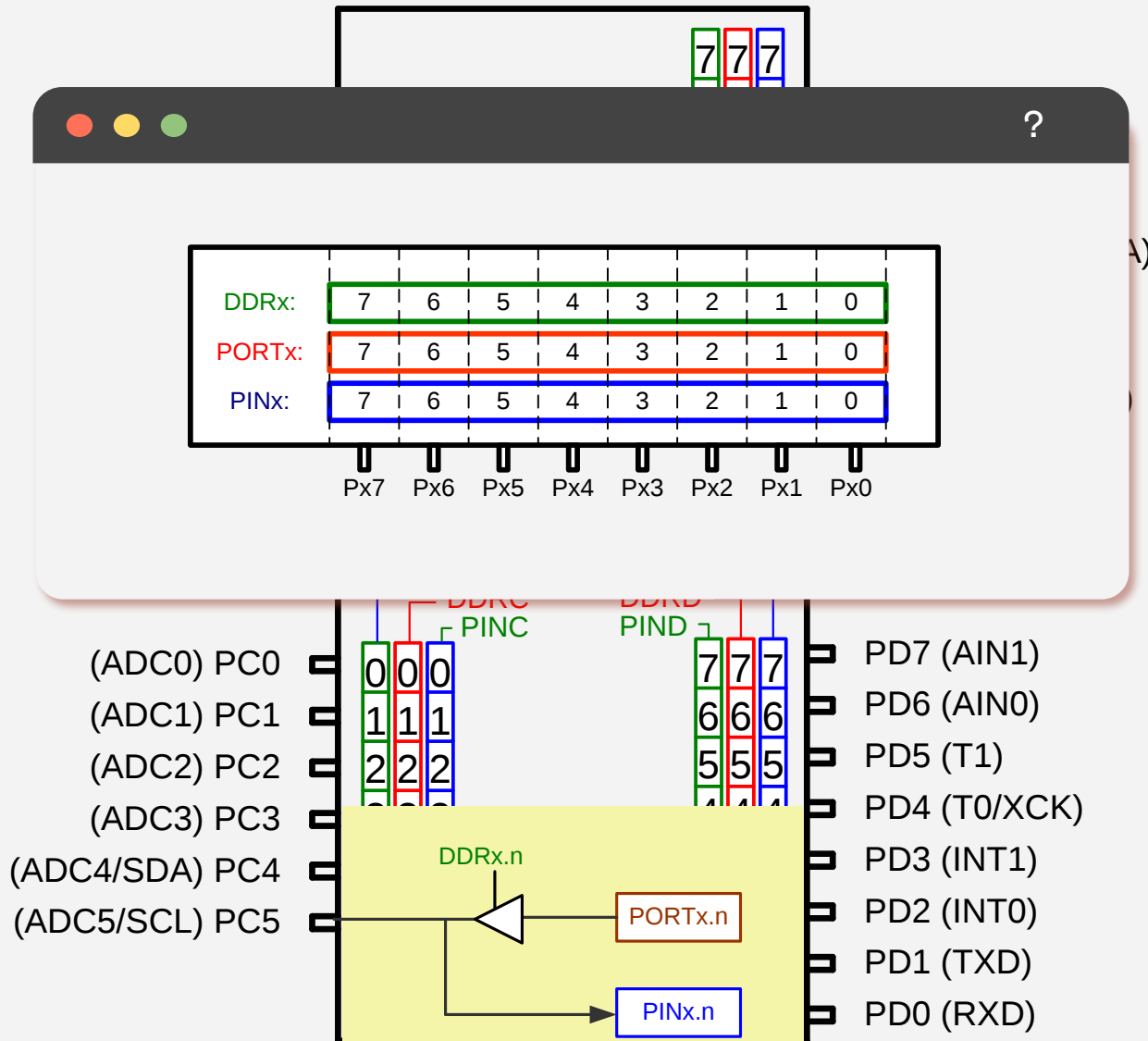


The structure of I/O pins

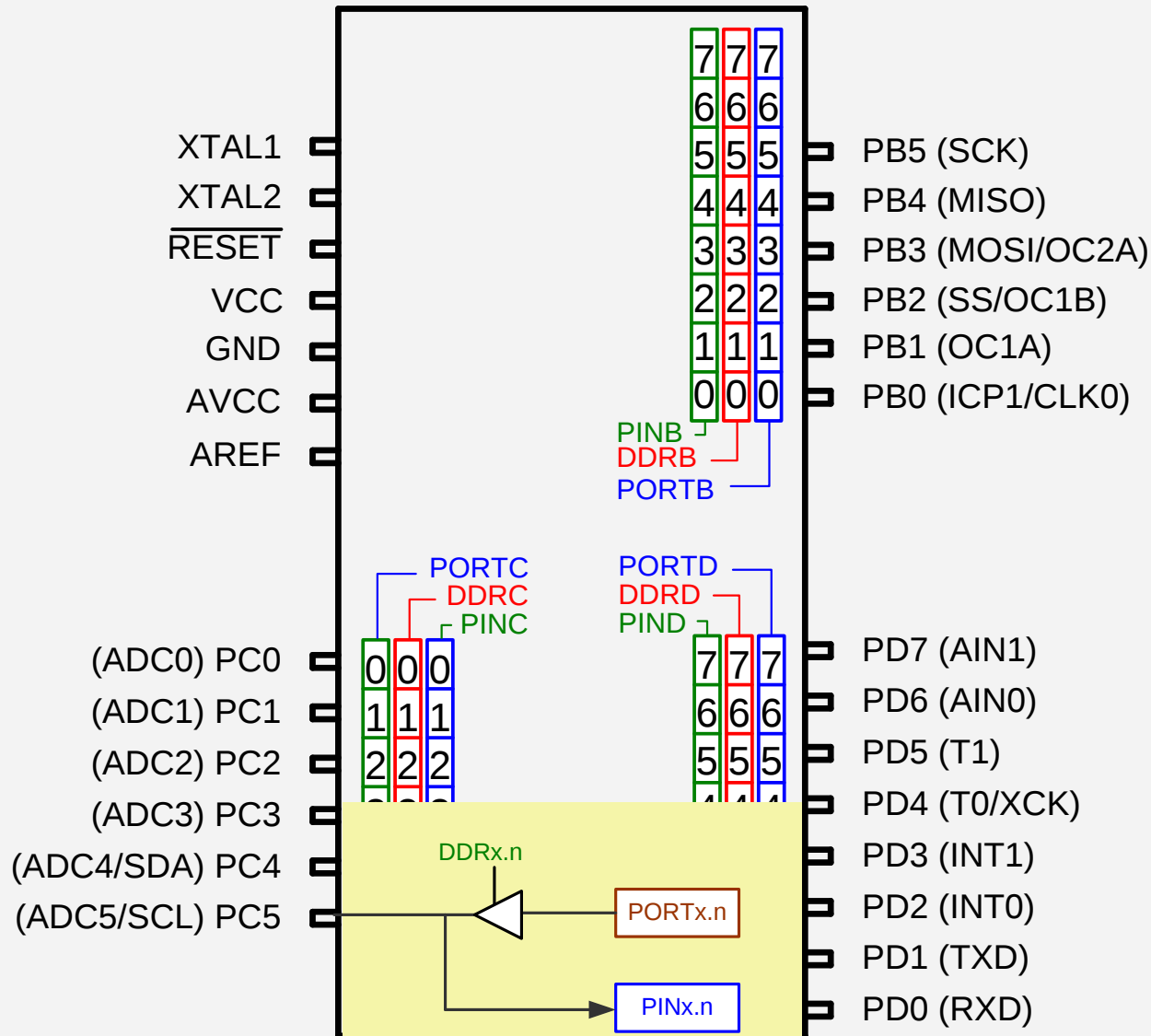


PORTx	DDRx	0	1
0	0	high impedance	Out 0
1	1	pull-up	Out 1

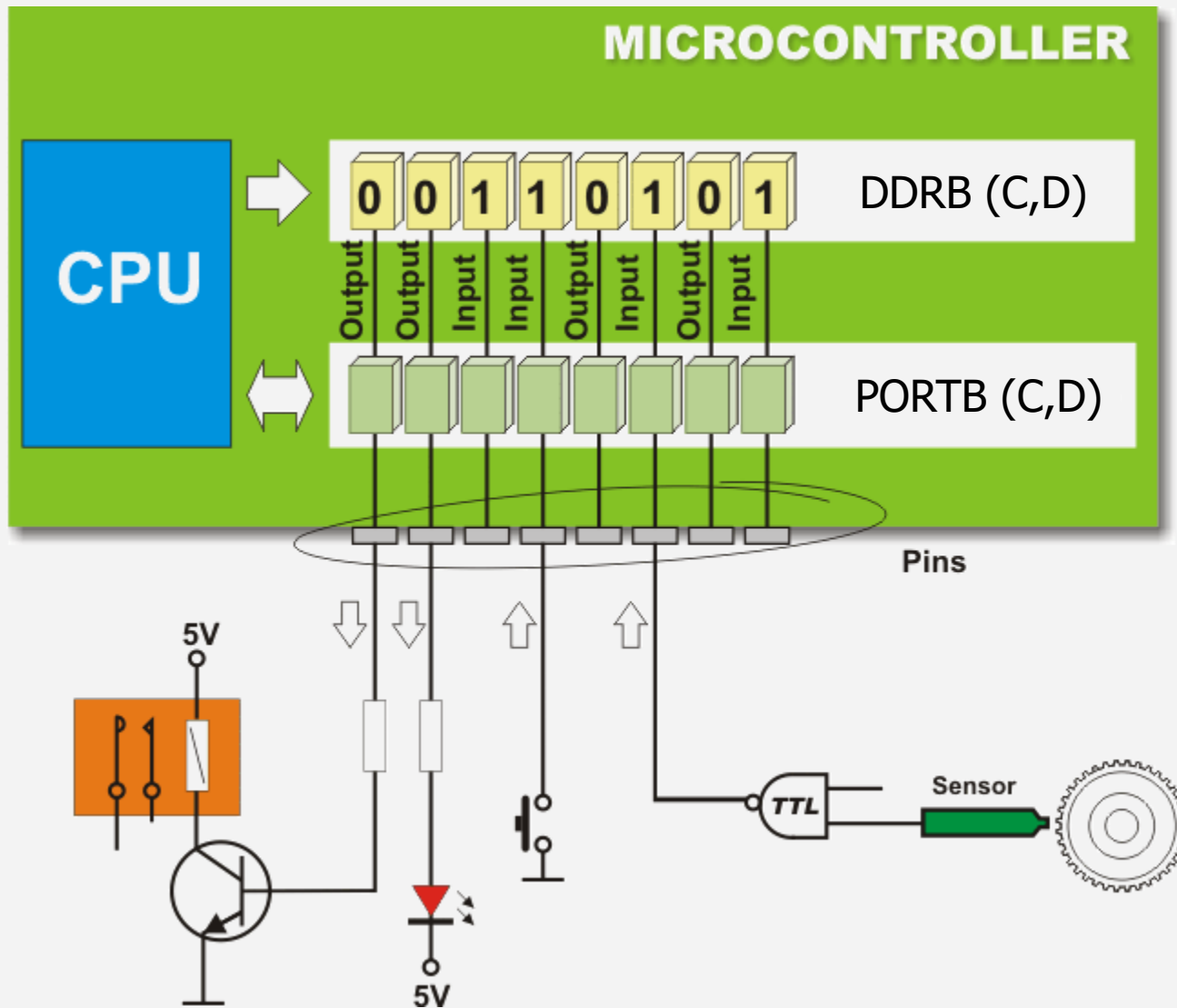
The structure of I/O pins



The structure of I/O pins

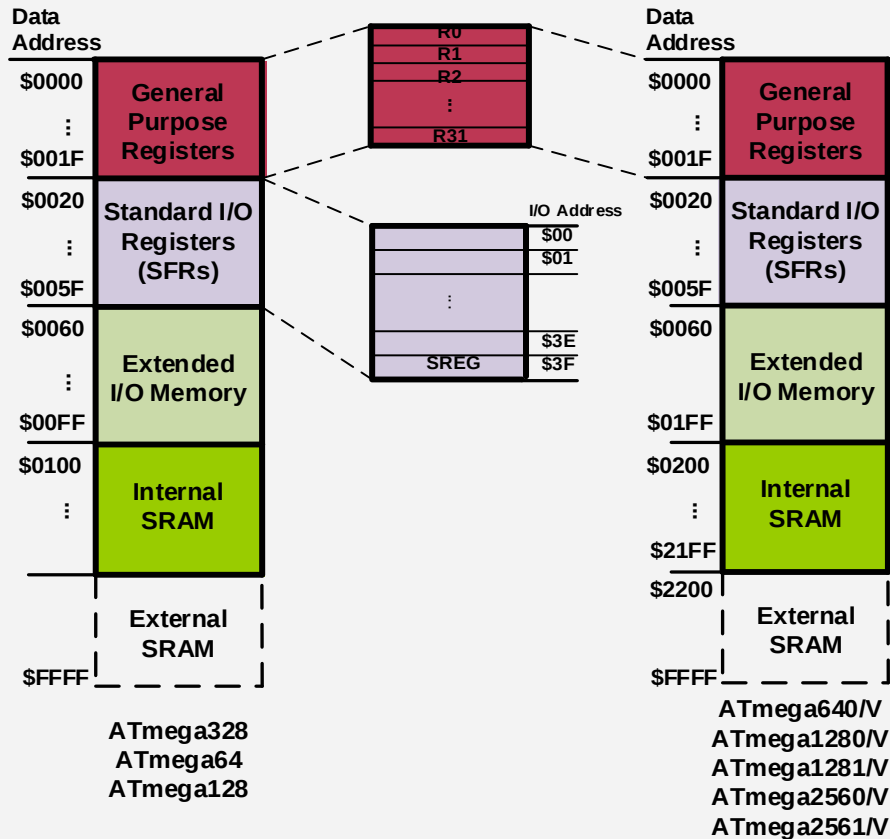


The structure of I/O pins



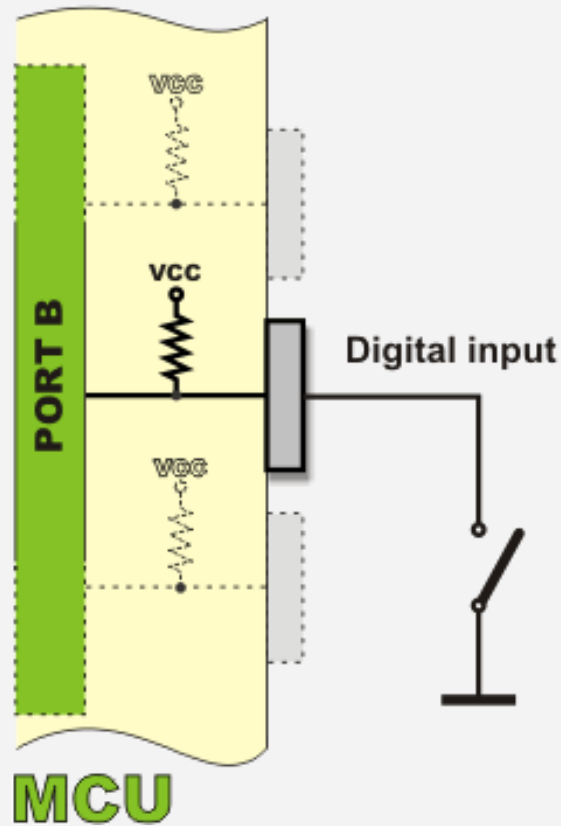


The Address of I/O pins

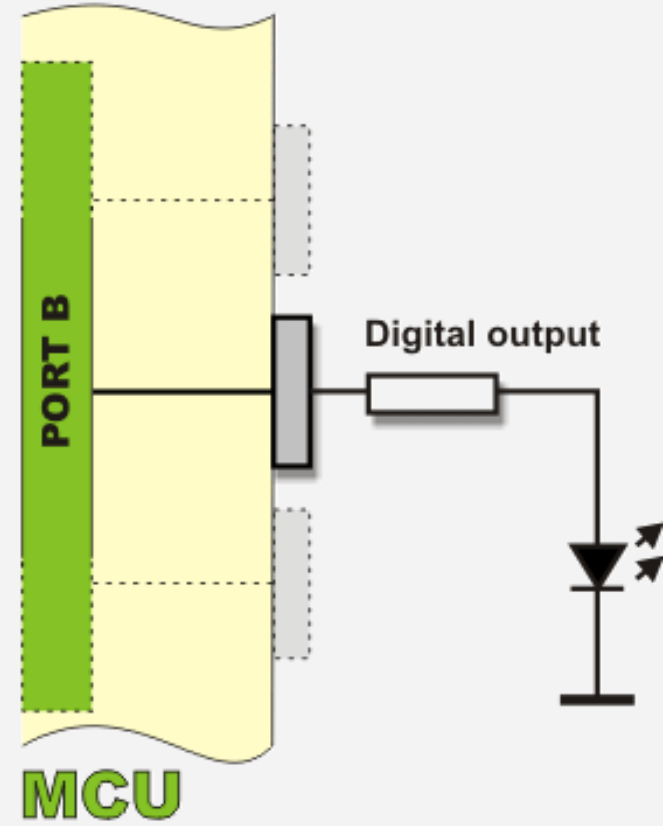


Offset	Name
0x20	PINA
0x21	DDRA
0x22	PORTA
0x23	PINB
0x24	DDRB
0x25	PORTB
0x26	PINC
0x27	DDRC

Pin with pull-up resistor



Pin without pull-up resistor



Example 1

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

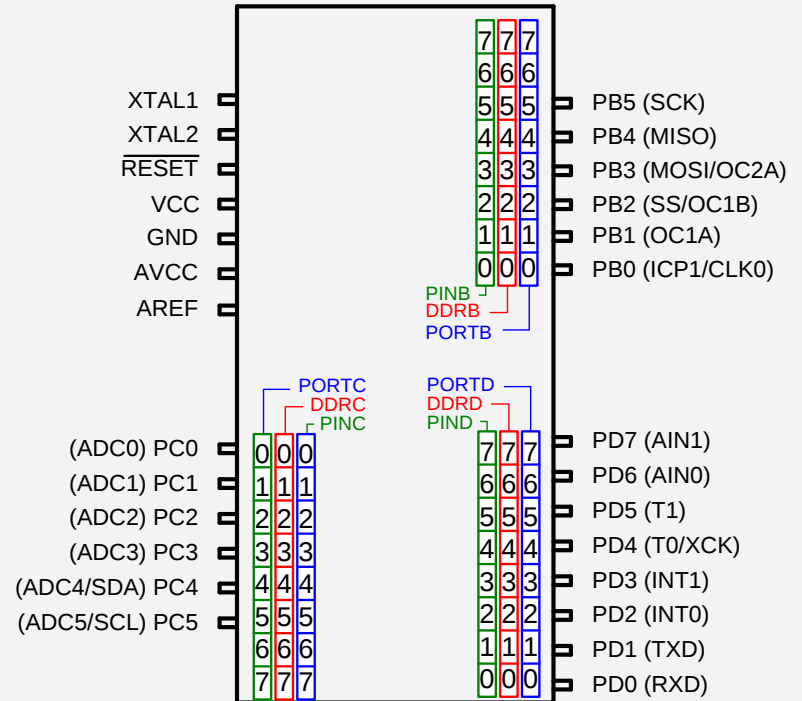
Solution

DDRB: 1 1 1 1 1 1 1 1

PORTB: 1 1 1 1 1 1 1 1

```
LDI R20,0xFF ;R20 = 11111111 (binary)
OUT PORTB,R20 ;PORTB = R20
OUT DDRB,R20 ;DDRB = R20
```

Example 1



?

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

Example 2

- Write a test program for the ATMEGA328 chip to toggle all the 8 bits of PORTB forever with some time delay between "on" and "off" states:

```
LDI    R16,0xFF      ;R16 = 0xFF = 0b11111111
OUT    DDRB,R16      ;make Port B an output port (1111 1111)
L1:    LDI    R16,0x55 ;R16 = 0x55 = 0b01010101
OUT    PORTB,R16     ;put 0x55 on port B pins
CALL   DELAY
LDI    R16,0xAA      ;R16 = 0xAA = 0b10101010
OUT    PORTB,R16     ;put 0xAA on port B pins
CALL   DELAY
RJMP   L1
```

Example 3

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

Solution

DDRD:

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

PORTD:

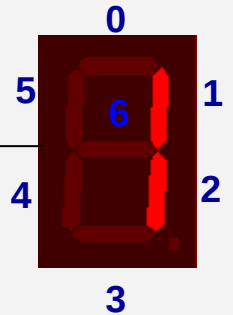
0	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---

```
LDI R20,0x06 ;R20 = 00000110 (binary)
OUT PORTD,R20 ;PORTD = R20
LDI R20,0xFF ;R20 = 11111111 (binary)
OUT DDRD,R20 ;DDRD = R20
L1: RJMP L1
```

ATmega328

PORTD

8/



PORTx

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

Example 4

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

Solution

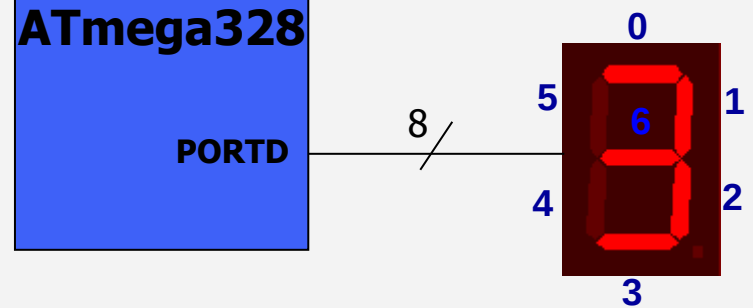
DDRD:

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

PORTD:

0	1	0	0	1	1	1	1
---	---	---	---	---	---	---	---

```
LDI R20,0x4F ;R20 = 01001111 (binary)
OUT PORTD,R20 ;PORTD = R20
LDI R20,0xFF ;R20 = 11111111 (binary)
OUT DDRD,R20 ;DDRD = R20
L1: RJMP L1
```



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

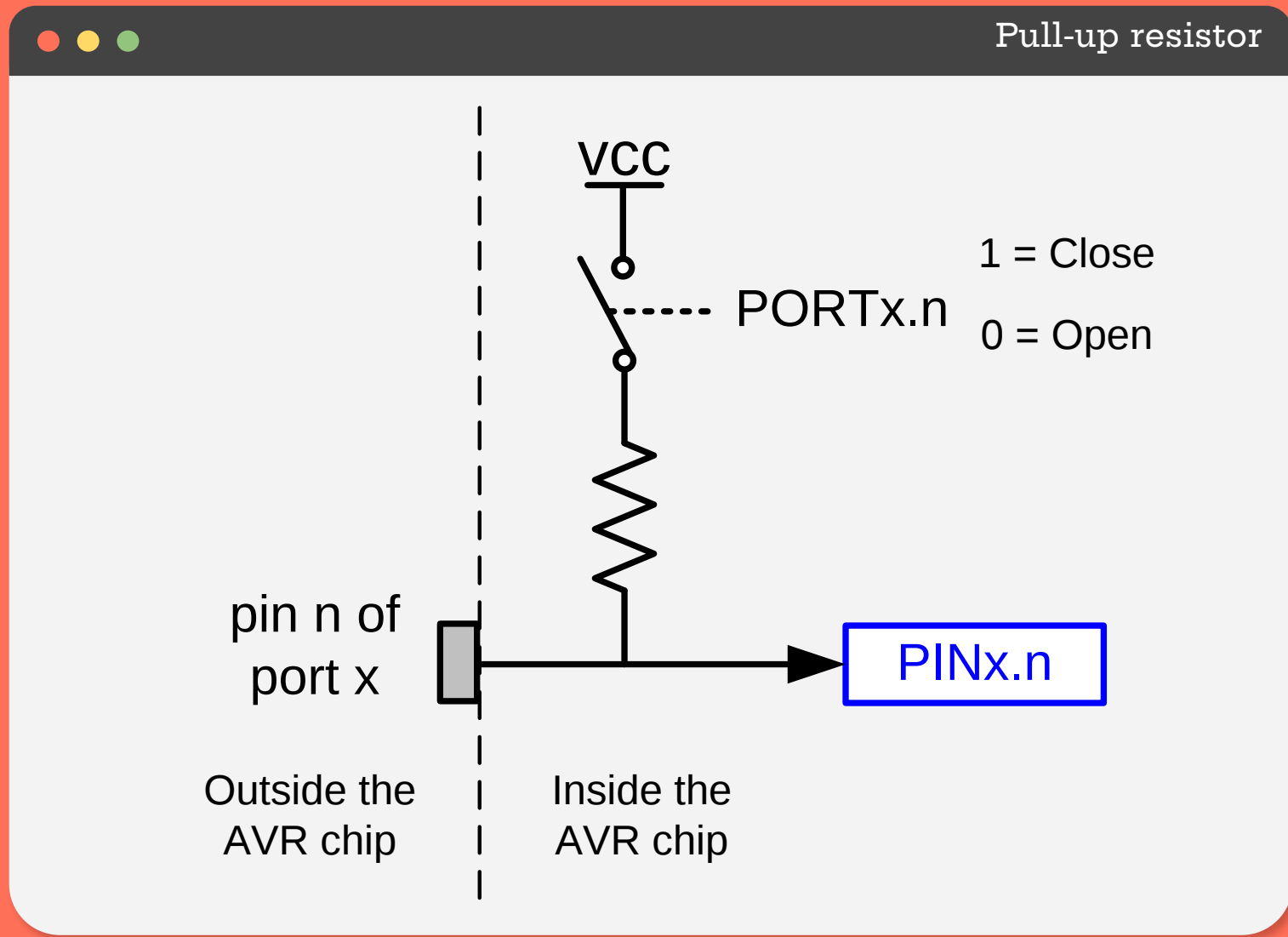
Example 5

- The following code gets the data present at the pins of port C and sends it to port B indefinitely, after adding the value 5 to it:

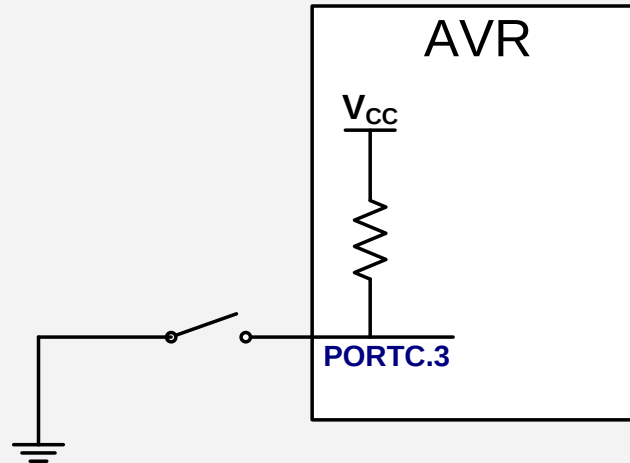
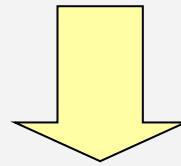
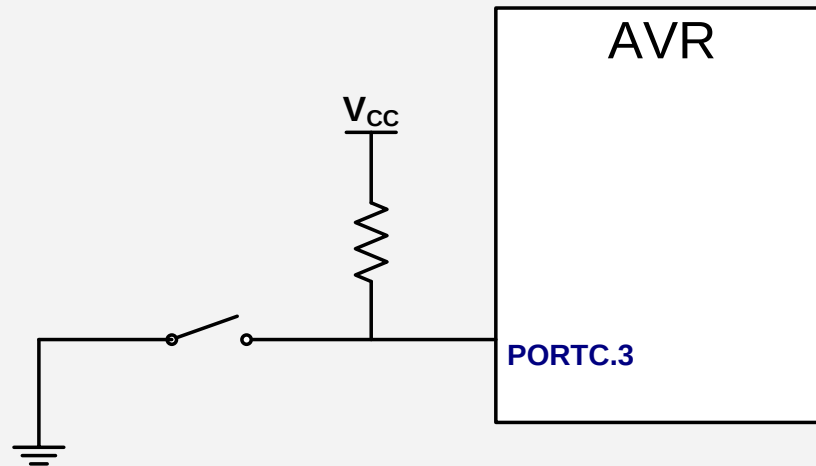
```

LDI    R16,0x00      ;R16 = 00000000 (binary)
OUT    DDRC,R16      ;make Port C an input port
LDI    R16,0xFF      ;R16 = 11111111 (binary)
OUT    DDRB,R16      ;make Port B an output port(1 for Out)
L2:    IN     R16,PINC ;read data from Port C and put in R16
LDI    R17,5
ADD    R16,R17        ;add 5 to it
OUT    PORTB,R16      ;send it to Port B
RJMP   L2             ;jump L2
    
```

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



Example





I/O bit manipulation programming

SBI and CBI instructions

- SBI (Set Bit in IO register)
 - SBI ioReg, bit ;ioReg.bit = 1
 - Examples:
 - SBI PORTD,0 ;PORTD.0 = 1
 - SBI DDRC,5 ;DDRC.5 = 1
- CBI (Clear Bit in IO register)
 - CBI ioReg, bit ;ioReg.bit = 0
 - Examples:
 - CBI PORTD,0 ;PORTD.0 = 0
 - CBI DDRC,5 ;DDRC.5 = 0



Example

- Write a program that toggles PORTB.4 continuously.

Solution

```
SBI   DDRB, 4
L1:   SBI   PORTB, 4
      CBI   PORTB, 4
      RJMP  L1
```

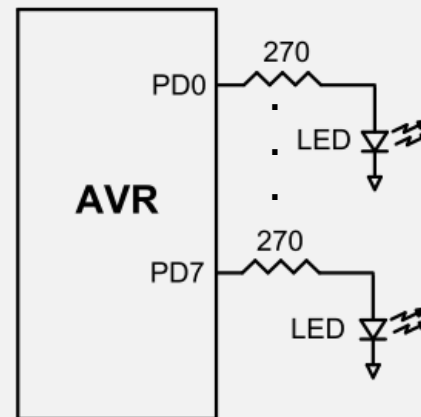
Example

- An LED is connected to each pin of Port D. Write a program to turn on each LED from pin D0 to pin D7. Call a delay module before turning on the next LED.

Example

```

LDI      R20, 0xFF
OUT      DDRD, R20           ;make PORTD an output port
SBI      PORTD,0             ;set bit PD0
CALL     DELAY               ;delay before next one
SBI      PORTD,1             ;turn on PD1
CALL     DELAY               ;delay before next one
SBI      PORTD,2             ;turn on PD2
CALL     DELAY
SBI      PORTD,3
CALL     DELAY
SBI      PORTD,4
CALL     DELAY
SBI      PORTD,5
CALL     DELAY
SBI      PORTD,6
CALL     DELAY
SBI      PORTD,7
CALL     DELAY
    
```



SBIC and SBIS

- SBIC (Skip if Bit in IO register Cleared)
 - SBIC ioReg, bit ; if (ioReg.bit = 0) skip next instruction
 - Example:

```
SBIC  PORTD,0 ;skip next instruction if PORTD.0=0
INC   R20
LDI   R19,0x23
```
- SBIS (Skip if Bit in IO register Set)
 - SBIS ioReg, bit ; if (ioReg.bit = 1) skip next instruction
 - Example:

```
SBIS  PORTD,0 ;skip next instruction if PORTD.0=1
INC   R20
LDI   R19,0x23
```

Example

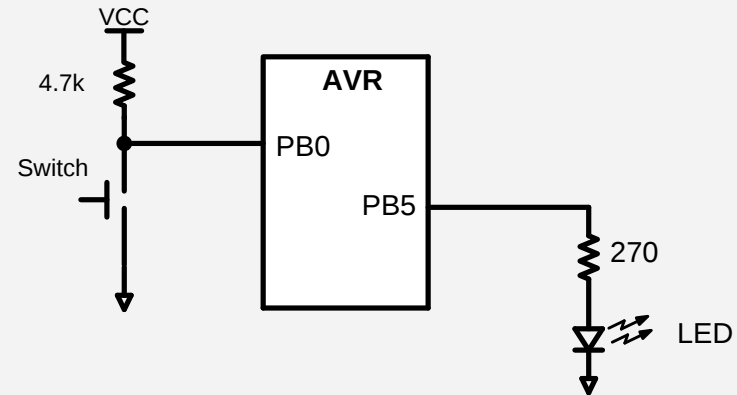
- Write a program to perform the following:
- (a) Keep monitoring the PB2 bit until it becomes HIGH;
- (b) When PB2 becomes HIGH, write value \$45 to Port C, and also send a HIGH-to-LOW pulse to PD3.

Solution

```
                CBI   DDRB, 2           ;make PB2 an input
                LDI   R16, 0xFF
                OUT   DDRC, R16         ;make Port C an output port
                SBI   DDRD, 3           ;make PD3 an output
AGAIN:          SBIS  PINB, 2           ;Skip if Bit PB2 is HIGH
                RJMP  AGAIN             ;keep checking if LOW
                LDI   R16, 0x45
                OUT   PORTC, R16        ;write 0x45 to port C
                SBI   PORTD, 3          ;set bit PD3 (H-to-L)
                CBI   PORTD, 3          ;clear bit PD3
HERE:           RJMP  HERE
```

Example

- A switch is connected to pin PB0 and an LED to pin PB5. Write a program to get the status of SW and send it to the LED.



Solution

```

                CBI   DDRB,0           ;make PB0 an input
                SBI   DDRB,5           ;make PB5 an output
AGAIN:          SBIC  PINB,0           ;skip next if PB0 is clear
                RJMP  OVER             ;(JMP is OK too)
                CBI   PORTB,5
                RJMP  AGAIN            ;we can use JMP too
OVER:           SBI   PORTB,5
                RJMP  AGAIN            ;we can use JMP too
    
```

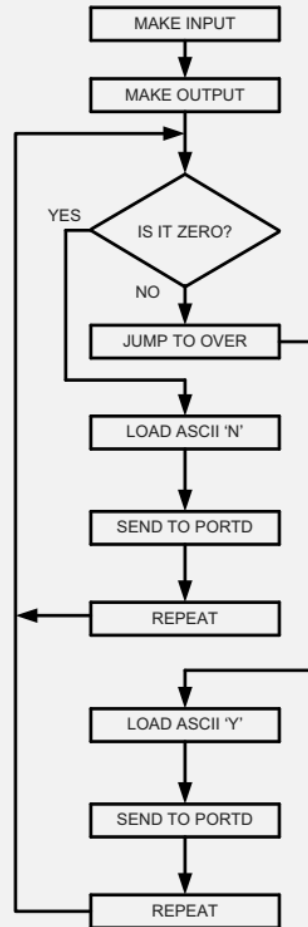


Example

- A switch is connected to pin PB2. Write a program to get the status of SW and perform the following:
 - (a) If $SW = 0$, send the letter 'N' to PORTD.
 - (b) If $SW = 1$, send the letter 'Y' to PORTD.



Example Flowchart



INSTRUCTIONS

CBI DDRB, 2

LDI R16, 0XFF
OUT DDRD, R16

AGAIN: SBIC PINB, 2

RJMP OVER

LDI R16, 'N'

OUT PORTD, R16

RJMP AGAIN

OVER: LDI R16, 'Y'

OUT PORTD, R16

RJMP AGAIN



Example

- A switch is connected to pin PB2. Write a program to get the status of SW and perform the following:
 - (a) If SW = 0, send the letter 'N' to PORTD.
 - (b) If SW = 1, send the letter 'Y' to PORTD.

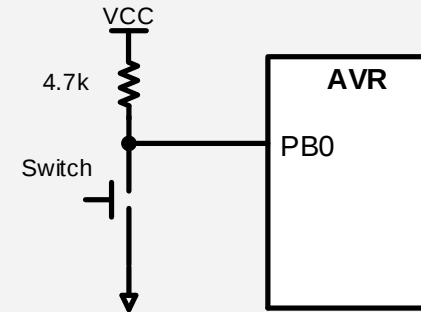
Solution

34

```
        CBI   DDRB,2           ;make PB2 an input
        LDI   R16,0xFF
        OUT   DDRD,R16         ;make PORTD an output port
AGAIN:   SBIS  PINB,2           ;skip next if PB2 is clear
        LDI   R16,'N'          ;R16 = 'N' (ASCII letter N)
        OUT   PORTD,R16        ;PORTD = 'N'
        RJMP  AGAIN
OVER:    LDI   R16,'Y'          ;R16 = 'Y' (ASCII letter Y)
        OUT   PORTD,R16        ;PORTD = 'Y'
        RJMP  AGAIN
```

Example

- A switch is connected to pin PB0. Write a program to get the status of SW and save it in location 0x200.



Solution

```
.EQU MYTEMP = 0x200      ;set aside location 0x200

        CBI   DDRB,0      ;make PB0 an input
AGAIN:  SBIC  PINB,0       ;skip next if PB0 is clear
        RJMP  OVER
        LDI   R16, 0
        STS  MYTEMP,R16   ;save it in MYTEMP
        RJMP  AGAIN       ;we can use JMP too
OVER:   LDI   R16,0x1      ;move 1 to R16
        STS  MYTEMP,R16   ;save it in MYTEMP
        RJMP  AGAIN
```



Thanks!



Do you have any questions?

razeghizade@gmail.com

Razeghizade.pudica.ir

CREADITS: This presentation was created by M.Razeghizadeh
Please keep this slide for attribution