

التاريخ: / /  
⇒ AVR AT mega 32 microcontroller

↳ Features

\* Advanced RISC Architecture  
32 \* 8 registers bits General purpose registers

↳ Peripheral Features

- \* Two 8-bit timer
- \* One 16-bit timer
- \* Four PWM
- \* 8-channel 10-bit ADC
- \* Two wire serial interface, USART, SPI
- \* Watchdog timer & for stucked loops
- \* Analog comparator

↳ Pin configurations 32 programmable I/O lines

↳ Block Diagram ✓

↳ AVR CPU core

- \* Harvard architecture memory X SRAM memory
- \* single level pipeline fetch & decode
- \* single level ALU & fast \*
- \* Status register "SREG" 8bit each bit means a flag

↳ system clock

↳ Internal  $\Rightarrow 1 \text{ MHz}$   
 ↳ External  $\Rightarrow \text{up to } 16 \text{ MHz}$   
 ↳ has a few types  
 depends on the app

↳ Accessing I/O registers

↳ Control  
 ↳ Data  
 ↳ Status

\*01 Get the address 0x3B

\*02 pointer to point to the address

↳ volatile uint8\_t \*PORTA = (volatile uint8\_t\*) (0x3B)  
 ↳ type      8-bits      ↳ convert (0x3B) to an address

⇒ OR

\*01 \*(volatile uint8\_t\*) (0x3B) = 0x10 or #define

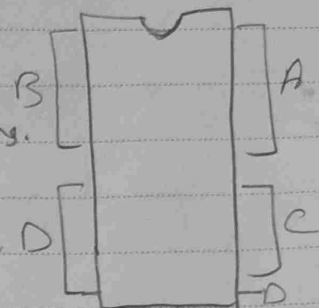
# التاريخ: DIO programming

الموضوع:

AVR ATmega 32 has 32 programmable I/O pins

\* 32 pins  $\rightarrow$  4 ports

$\hookrightarrow$  each port contains 8 pins.



\* each port can be configured as general I/O per pins

\* each pin has a specific function

\* each pin can be configured using 3 I/O registers  
"DDR<sub>xn</sub>", "PORT<sub>xn</sub>", "PIN<sub>xn</sub>"

$\hookrightarrow$  Data direction register "DDR<sub>x</sub>"

\* used to configure each pin

\* 8-bit register, each bit controls its mapped pin

$\hookrightarrow$  Configuring pin 2 in Port A as an input pin

$\Rightarrow$  DDRA &= ~ (1 << 2);  $\leftarrow$  clear bit

$\hookrightarrow$  Configuring pin 6 in Port A as an output pin

$\Rightarrow$  DDRA |= (1 << 6);  $\leftarrow$  set bit

$\hookrightarrow$  Port register "Port<sub>x</sub>"

\* used to pull each pin high "1" or low "0"

$\hookrightarrow$  pull pin 3 in Port A low  $\Rightarrow$  PORTA &= ~ (1 << 3);

$\hookrightarrow$  pull pin 5 in Port A high  $\Rightarrow$  PORTA |= (1 << 5);

Port Input register "PIN"

\* store each pin state high or low-level

Read state of Pin 7 in Port A  $(PIN \& (1 \ll 7)) \ll 7$

$(PIN \& (1 \ll 2)) \ll 2$

Configuring the pin

\* I/O pins

\* Current

\* Pin states

Configuring the pin