DIO programming

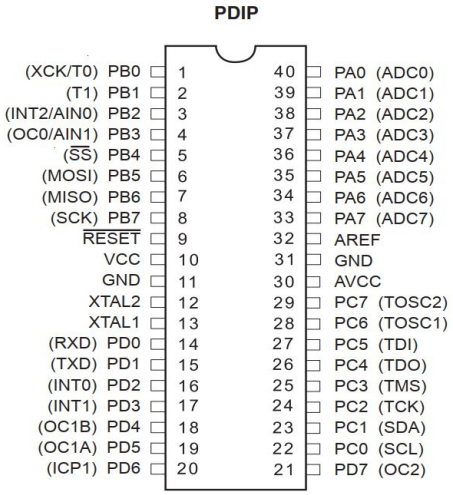
By: Yehia M. Abu Eita

Outlines

- Introduction
- The Data Direction Register (DDRx)
- The Port Register (PORTx)
- The Port INput Register (PINx)
- Configuring the Pin

Introduction

- AVR ATmega32 has **32 programmable I/O** pins.
- These pins are grouped into **4 ports**.
  - **Port A, Port B, Port C, and Port D**
- Each **port contains 8 pins**.
- Ecah pin can be configured as general-purpose inputs/outputs.
- Each pin has special function.
- Each pin can be configured using **three I/O registers** for each port:
  - **DDRxn, PORTxn, PINxn**



The Data Direction Register (DDRx)

- It can be called also a **control/configuration** register.
- It is used to configure each pin, if it is an **input or an output pin**.
- It is an **8-bit register**, **each bit controls its mapped pin**.
- **Input or Output depends on the connected device if it is input or output device.**

DDRA – Port A Data Direction Register

Bit	7	6	5	4	3	2	1	0	
	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Example**:
  - Configuring **Pin2** in **Port A** as an **input pin**, `DDRA &=~(1<<2); // clear bit`
  - Configuring **Pin6** in **Port A** as an **output pin**, `DDRA |= (1<<6); // set bit`

The Port Register (PORTx)

- It can be called also a **data** register.
- It is used to **pull each pin high or low**, writing **1 or 0**.
- It is an **8-bit register**, **each bit controls its mapped pin**.
- **Set to 1 means 5 volts and 0 means ground.**

PORTA – Port A Data Register

Bit	7	6	5	4	3	2	1	0	
	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Example**:
  - Pull **Pin3** in **Port A <u>low</u>**, `PORTA &=~(1<<3); // clear bit`
  - Pull **Pin5** in **Port A <u>high</u>**, `PORTA |= (1<<5); // set bit`

The Port INput Register (PINx)

- It can be called also a **status** register.
- It is used to **store each pin state**, **high-level or low level**.
- It is an **8-bit register**, **each bit stores its mapped pin status**.
- **If the bit stores 1, this means high-level, and if 0, this means low-level.**

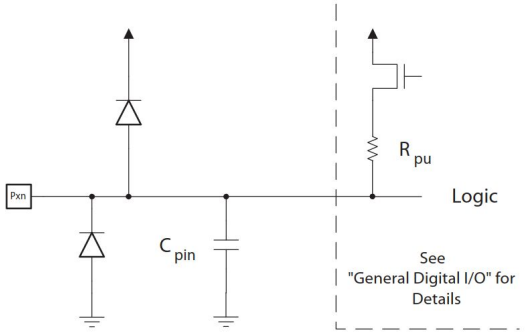
PINA – Port A Input Pins Address

Bit	7	6	5	4	3	2	1	0	
	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

- **Example**:
  - Read state of **Pin7** in **Port A**, `(PINA&(1<<7))>>7; // if pin7 connected to the vcc the result is 1`
  - Read state of **Pin2** in **Port A**, `(PINA&(1<<2))>>2; // if pin2 connected to the ground the result is 0`

Configuring the Pin

- **I/O pins**:
  - Each pin has selectable pull-up resistors.
  - Each pin has protection diodes to both VCC and Ground.
- **Current**:
  - Each pin can sink or source a maximum current of 40mA.
  - All the ports combined, should not exceed 200mA.
- **Pin states**:
  - Each pin has four different states depending on the state of the bits corresponding to this pin in DDRx and PORTx of that specific port.



DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

Summary

- Now you are familiar with the DIO in ATmega32
- Remember that DDRx register is used to configure pin direction
- Remember that the PORTx register is used to control pin output
- Remember that the PINx register is used to get pin state
- Remember floating pins will not be considered a 0 or 1 unless it is pulled up or down externally
- Remember to use set-bit, clear-bit, and read-bit macros