Mechatronics Engineering

Course Introduction and
ATmega328P Overview

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Materials Science
German University in Cairo

Organization

Lecture
- Two hours lecture
- For tutorials and **labs** please refer to the schedule.

Assessment
- Mid term exam 20%, Final Exam 40%
- Quizzes 10%
- Lab reports: 15%
- Project (groups up to 3): 15%

Textbooks
- References: check the CMS
- slides and tutorials are on CMS

Course Overview (Topics)

- Course introduction.
- Architecture overview of ATmega328P.
- ATmega328P C programming
- Microcontroller peripherals (I/O ports, Timers, CCP, ADC, Comm. Protocols). ISR.
- A/A and A/D interfacing.
- Multitasking and Scheduling
- Discrete-time signals, Sampling Theorem, Aliasing.
- Z-transform, Sampled-data systems
- Discrete Equivalent controllers

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Integrated Product Design Project "Microcontroller Based Mechatronic Design"

- Each group must design, build, test, and demonstrate a device controlled by a MC.
- The device should have functioning elements in all of the following five categories:
  1. **Output Display (**LED , 7-segment digit display , **LCD**, …)
  2. **Data Input (**switch, button, potentiometer, joystick, keypad, keyboard)
  3. **Sensors (**Ranging sensors, temperature sensor, pressure sensor, accelerometer, encoder, strain gauge, Hall effect sensor, piezoelectric sensors …etc.)
  4. **Actuators (**solenoid, dc motor, stepper motor, RC servo motor, …etc.)
  5. **Logic, Processing, and Control; AND Miscellaneous (functional elements not covered in the categories above)**
     - Minimum requirement: closed-loop feedback control may be with:
     - programmed logic, menu-driven software, advanced and/or multiple interfaced PIC microcontrollers, components not included in other categories

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Project Deliverables

- The choice of your project concept could have a large impact on the grade you receive; so please evaluate your alternative concepts carefully
- Project Deliverables

  Each group must present the following over the course of the project:

  1. A proposal
  2. A device containing functional elements in each of the categories listed above
  3. A final design report including schematics and code.
  4. A group presentation

Proposal

The proposal must contain:

- A title page with title, group number, group member names, and date.
- A concise overview of what your proposed device is and how it will work. Include well-labeled figure(s) to illustrate your device concept (what it will look like, how it works, what it does). Be sure to label key components in your figures (with concise text and arrows).
- A functional diagram showing all major components and their connections.
- A list of proposed components in each of the functional element categories.
- You should consider the proposal as a preliminary draft for the final design report. If you do a good job with the proposal and create high quality illustrations and diagrams, you will be able to reuse the material in your final report.

Rating	Description of performance
0	nothing implemented.
4	something implemented, but non functional.
8	something implemented, but not functioning as designed in a repeatable and reliable way.
10	something functioning as designed (i.e., performs some intended, useful function) and repeatable (i.e., it works every time), but did not require much research or effort on your part (e.g., you purchased something requiring very little interfacing and work).
12	something functioning as designed and repeatable, and required significant research and effort on your part (e.g., you built something discussed in the class, but not presented in detail in class or Lab, that required significant research and effort).
15	something functioning as designed (i.e., performs some intended, useful function) and repeatable, and required substantial independent research and effort on your part (e.g., you built something requiring knowledge and skills not presented in the class or in Lab).

NOTE: These ratings are somewhat qualitative, so official scores will not be released until the end of the semester, after the instructor and TAs meet to discuss all of the results.

Final Design Report

- **The final report and presentation is due at your last meeting of the Lab session.**
- The report should include:
- **Title Page** with title, group number, group member names, and date.
- **Table of Contents**: List each section along with their page numbers.
- **Design Summary**: concise overview of what the device does and how it works. Include, number, and refer to a well-labeled figure or photograph illustrating the overall device.
- **System Details**: EOM as well as concise descriptions and illustrations of the system's basic design (not detailed drawings or parts) and function. Include illustrative figures and photographs with key features and components clearly labeled, circuit schematics, functional diagrams, and concise software flowcharts. Be sure to number and refer to all Figures and describe them briefly. Also refer to and briefly describe anything in the Appendix.
- **Design Evaluation**: Briefly describe the success of the device in meeting the functional element categories as opposed to your dynamic model simulations.
- **Partial Parts List**: For each unique and/or interesting component in your design, list the following information: part name or brief description, model number, and price. Include only actuators, sensors, sound modules, special purpose amplifiers, specialty drivers, external A/Ds or D/As, and other components not used in Lab (i.e., don't list common components like resistors, capacitors, small LEDs, basic LCD displays, basic keypads, etc.)
- **Appendix**: detailed wiring diagrams (if details are not included in earlier figures), well-commented software listings, and anything else supporting the System Details section.

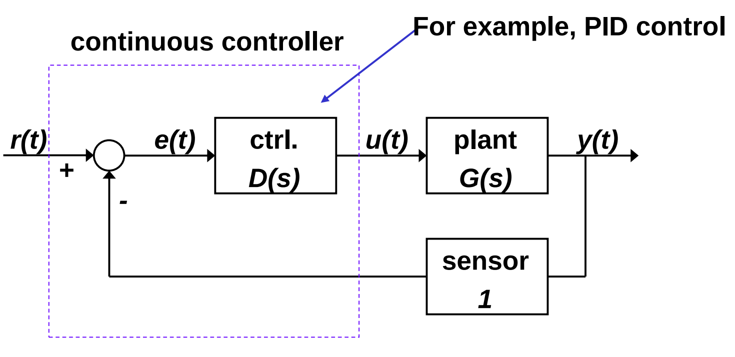
- **After looking at the figures and schematics and after reading the BRIEF descriptions in "Design Summary" and "System Details," the reader should be able to fully understand what your device is, what it looks like, and how it functions (without seeing the actual device).**
- Please use software tools (Word, PowerPoint, Proteus, CAD) for all work.

Project Deadlines

- Immediately begin to develop project ideas
- Project proposal and team names **due** before the midterm exams.
- Prototype, Project presentations and Final report.
- The final report is due last Thursday immediately before the revision week.
- Project presentation is in your last lab session.

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

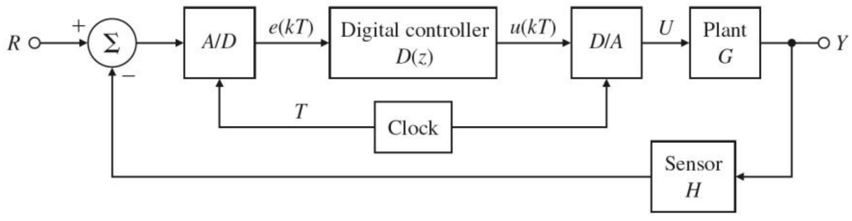
Analog Control System



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Digital Control System

- Digital Control System
  - T is the sample time (s)
  - Sampled signal : $e(kT) = e(k)$



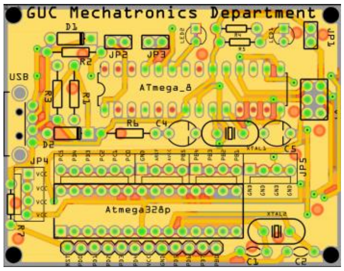
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

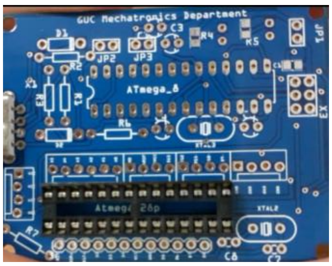





Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Home built development board





Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

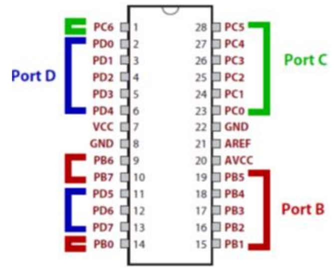
ATmega Peripherals

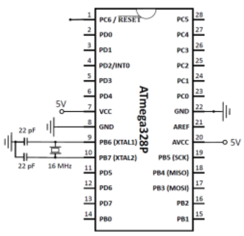
▪ 23 General Purpose IO Bits	▪ 6 or 8 ADC channels (depends on package)
▪ Two 8 bit & one 16 bit timer/counters	▪ Serial USART
▪ Real time counter with separate oscillator	▪ SPI & I2C (TWI) Serial Interfaces
▪ 6 PWM Channels	▪ Analog comparator
	▪ Programmable watchdog timer

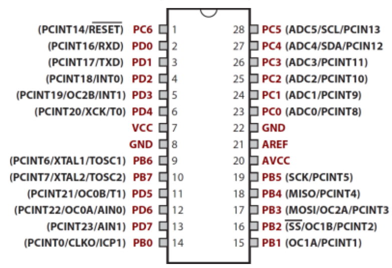
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

ATmega328P Pinout
ATmega328P GPIO Ports

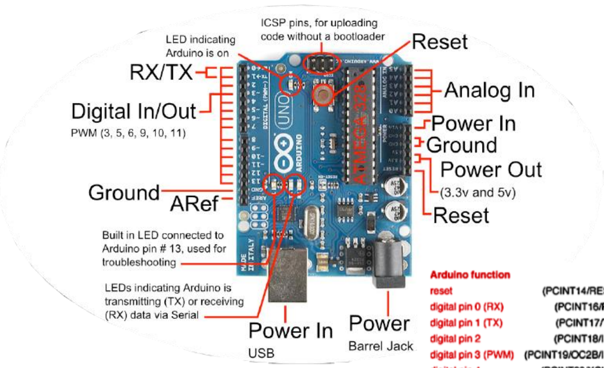
- Three 8 Bit IO Ports
  - Port B, Port C & Port D
  - Pins identified as PBx, PCx or PDx (x=0..7)
- Each pin can be configured as:
  - Input with internal pull-up
  - Input with no pull-up
  - Output low
  - Output high

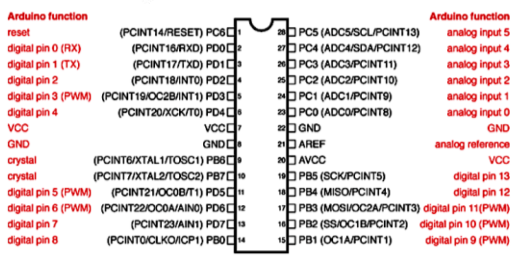




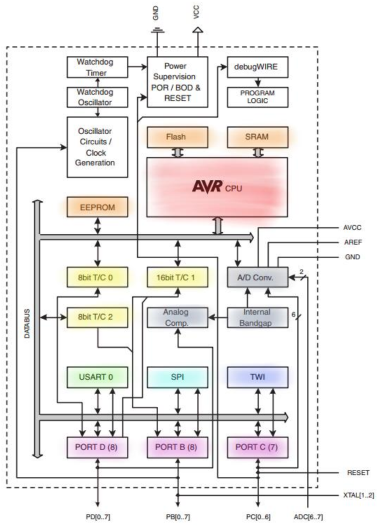


Arduino Uno Overview

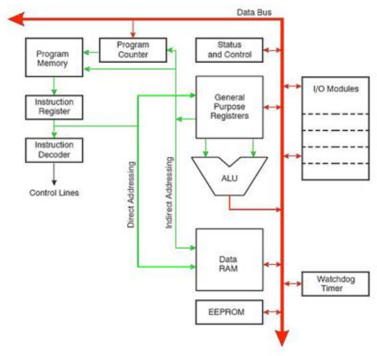




Atmega 328P Internals

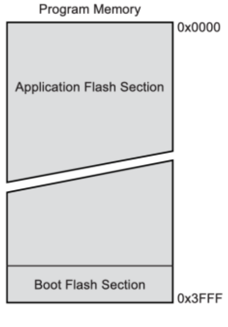


Processor Layout



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

ATmega328P Memory



Data Memory	
32 Registers	0x0000 - 0x001F
64 I/O Registers	0x0020 - 0x005F
160 Ext I/O Registers	0x0060 - 0x00FF
Internal SRAM (2048 x 8)	0x0100
	0x08FF

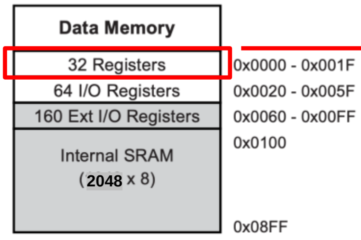
- Program Memory: 32K Bytes of programmable flash memory 1K Bytes of EEPROM

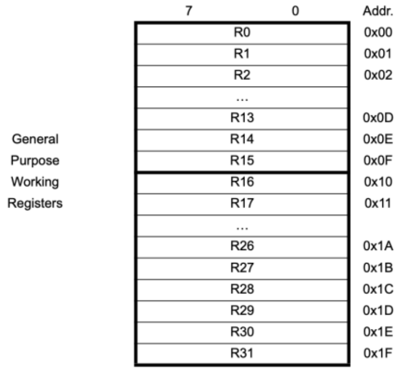
Data Memory

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

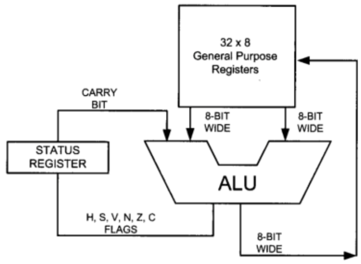
ATmega328P Memory

Data Memory:





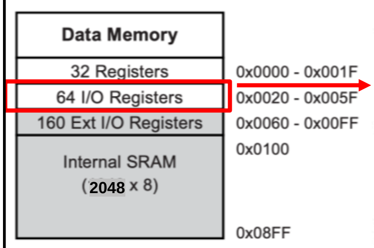
- **GPWR**: **Registers used by the CPU to store data temporarily**



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

ATmega328P Memory

Data Memory:

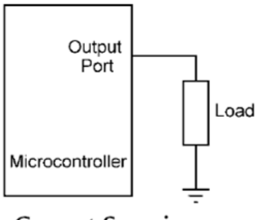


Address		Name
Mem.	I/O	
$20	$00	TWBR
$21	$01	TWSR
$22	$02	TWAR
$23	$03	TWDR
$24	$04	ADCL
$25	$05	ADCH
$26	$06	ADCSRA
$27	$07	ADMUX
$28	$08	ACSR
$29	$09	UBRRL
$2A	$0A	UCSRB
$2B	$0B	UCSRA
$2C	$0C	UDR
$2D	$0D	SPCR
$2E	$0E	SPSR
$2F	$0F	SPDR
$30	$10	PIND
$31	$11	DDRD
$32	$12	PORTD
$33	$13	PINC
$34	$14	DDRC
$35	$15	PORTC
$36	$16	PINB
$37	$17	DDRB
$38	$18	PORTB
$39	$19	PINA
$3A	$1A	DDRA
$3B	$1B	PORTA
$3C	$1C	EECR
$3D	$1D	EEDR
$3E	$1E	EEARL
$3F	$1F	EEARH
$40	$20	UBRRC
		UBRRH
$41	$21	WDTCR
$42	$22	ASSR
$43	$23	OCR2
$44	$24	TCNT2
$45	$25	TCCR2
$46	$26	ICR1L
$47	$27	ICR1H
$48	$28	OCR1BL
$49	$29	OCR1BH
$4A	$2A	OCR1AL
$4B	$2B	OCR1AH
$4C	$2C	TCNT1L
$4D	$2D	TCNT1H
$4E	$2E	TCCR1B
$4F	$2F	TCCR1A
$50	$30	SFIOR
$51	$31	OCDR
		OSCCAL
$52	$32	TCNT0
$53	$33	TCCR0
$54	$34	MCUCSR
$55	$35	MCUCR
$56	$36	TWCR
$57	$37	SPMCR
$58	$38	TIFR
$59	$39	TIMSK
$5A	$3A	GIFR
$5B	$3B	GICR
$5C	$3C	OCR0
$5D	$3D	SPL
$5E	$3E	SPH
$5F	$3F	SREG

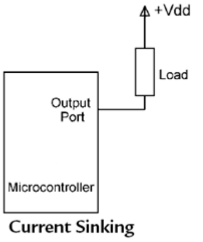
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

I/O Interface

- When the pin is sourcing current, one pin of the load is connected to the MC port and the other pin to ground. The load is then energized when the port output is at logic 1.
- When the pin is sinking current, one pin of the load is connected to the supply voltage and the other pin to the output of the port. The load is then energized when the port output is at logic 0.



Current Sourcing



Current Sinking

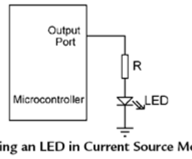
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

LED Interface

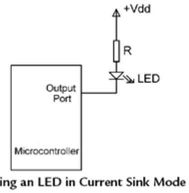
- Standard size LEDs consume about 10 mA of current for normal brightness.
- The voltage drop across an LED is about 2 V, but the voltage at the output of a MC port is about 5 V when the port is at logic 1 level.
- As a result, it is not possible to connect an LED directly to a MC output port. What is required is a resistor to limit the current in the circuit.
- We need to drop 3 V across a resistor

$$R = \frac{5-2\,\mathrm{V}}{10\,\mathrm{mA}} = 0.3\,\mathrm{K}$$

- The nearest physical resistor we can use is 330Ω



Connecting an LED in Current Source Mode

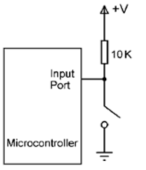


Connecting an LED in Current Sink Mode

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Button Input

- Switches are extensively used in embedded systems. Our main initial interest is not to switch directly a voltage or current, but to convert the switch position to a logic level that can be read by a microcontroller port bit.
- One of the most common types of inputs is a button (a push-button switch) input where the user can change the state of an input pin by pressing a button.
- Basically, button input can use two different methods: **active low** and **active high**.
- In **active low** implementation, the microcontroller input pin is connected to the supply voltage using a resistor (this is called a **pull-up resistor**).
- Normally the MC input is pulled to logic 1 by the resistor.
- When the button is pressed, the input is forced to ground potential, which is logic 0.
- The change of state in the input pin can be determined by a program.
- Some ports have internal pull-up resistors and these resistors can be enabled by manipulating bits in SFRs.
- When one of these port pins is used for button input, there is no need to use an external pull-up resistor and the button can simply be connected between the port pin and ground.

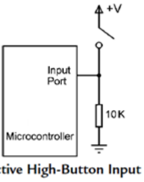


Active Low-Button Input

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Button Input

- A button can also be connected in **active high** mode.
- In this configuration, a resistor (called **pull-down resistor**) is connected between the port pin and ground.
- Normally, the port pin is at logic 0.
- When the button is pressed the port pin goes to the supply voltage, which is logic 1.
- To avoid that the MC read wrong switch state during bouncing of the metal parts, is to delay reading the input after the switch state changes.
- For example, when the switch is pressed, wait about 10 ms before the state of the switch is read.



Active High-Button Input

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Input-Output Configuration Registers in ATmega328P

ATmega328P has 3 configurable bi-directional input-output ports. They have internal pull-up resistors which can be selected for each bit.

- PORTB (PB7:0)
- PORTC (PC6:0)
- PORTD (PD7:0)

Each Input-Output port has 3 registers associated with it. They are designated as:

- PORTx (PORTx Data Register)
- DDRx (PORTx Data Direction Register)
- PINx (PORTx Input Pins Register)

Each of these registers is 8 bits wide. So, each bit affects only one pin that it is associated with.

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Input-Output Configuration Registers in ATmega328P

- **For example, for PORTD:**

PORTD – The Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
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	

DDRD – The Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
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	

PIND – The Port D Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x09 (0x29)	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
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	

When specifiying a pin as input, we check pins in PINx register.

DDRDx	PORTDx	Configuration
0	0	Input
0	1	Input with pull-up resistor enabled
1	0	Output (LOW)
1	1	Output (HIGH)

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

The MCU Control Register

MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	BODS[1]	BODSE[1]	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PUD: Pull-Up Disable

When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups (DD*xn* = 0, PORT*xn* = 1).

GPIO as output

- The pin is in a state of low impedance.
- The pin can supply current to other circuits (40mA max).
- Shorts or attempting to draw more current can fry the microcontroller.
- Use external resistors to limit current from the pins.

GPIO as Input

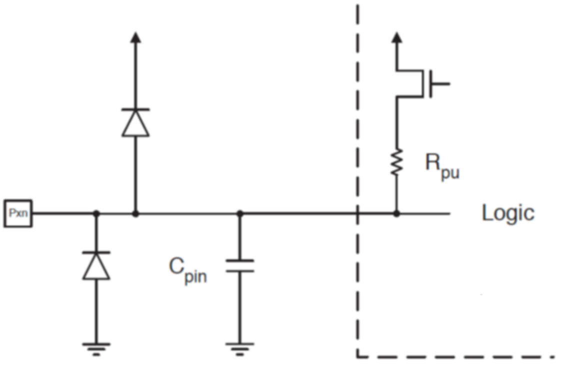
- All inputs can be used as digital inputs.
- Pins setup to be inputs are in the High impedance (High-Z) state.
- In this state, they draw minuscule current, and don't load the circuit to which they are attached.
- If the pin is not connected to a circuit while in this High-Z state, it is "floating" and read unpredictable values.

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

GPIO Overview

- The direction of one port pin can be changed independently of the other pins.

Port: ***x***
Pin: ***n***

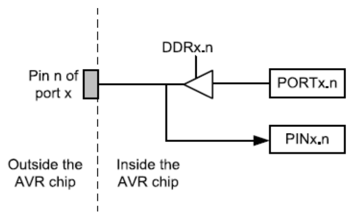


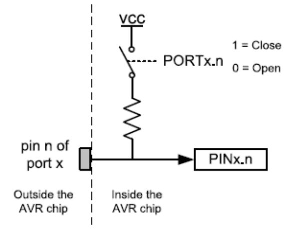
Circuit Diagram for Each Pin

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Port Pin Configurations

DDxn	PORTxn	PUD (in MCUCR)	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)





A schematic of a single port pin containing the DDR bit, the Data bit and the input pin bit with the addition of a pull-up resistor.

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Port Registers

- **PORT*x*** **x = B,C,D**
  - *controls the value of the pins*
- **DDR*x***
  - *controls the direction of the pins (I/O)*
- **PIN*x***
  - *reads the value of the pins*

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Ways of Setting Pins

- Directly (All bits in a port)
  - DDRC = 0b01101100;
- Bit Shifting
  - PORTB = (1<<2)|(1<<3)|(1<<5)|(1<<6);
- OR-ing with a Mask (specific bits, leaving the rest alone)
  - DDRD |= 0b00010100; //sets bits 2 and 4.

Ways of Clearing Pins

- Directly (All bits in a port)
  - DDRC = 0b01101100;
- AND-ing with a Mask

(specific bits, leaving the rest alone)

  - DDRD &= 0b11101011; //clears bits 2 and 4.
  - DDRD &= ~((1<<2)|(1<<4));

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

How to Check a Pin State

- AND with a mask
  - If (PINB & 0b00010000)

```
{
        // what to do if pin 4 of port B is 1.
}
```

How to Toggle a Pin State

- Even though PIN*x* is for reading a port, if you write a 1 to any pin, it will toggle that pin.!
- The pins in PINx are supposedly read-only. !
- Writing a 1 to the pin doesn't set the pin, but toggles it.

Writing a 0 to a pin doesn't clear it, it leaves it unchanged.

```
PINB = (1<<5);  // toggles only pin 5
```

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Example

- A door sensor is connected to bit 1 of Port B, and an LED is connected to bit 6 of Port C. Write an AVR C program to monitor the door sensor and, when it opens, turn on the LED

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

int main(void)
{

        DDRB = DDRB & 0b11111101;
        DDRC = DDRC | 0b1000000;

        while(1)
        {
                if (PINB & (0b00000010))
                        PORTC = PORTC |0b10000000;
                else
                        PORTC = PORTC & 0b0111111;
        }

        return 0;
}
```

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science