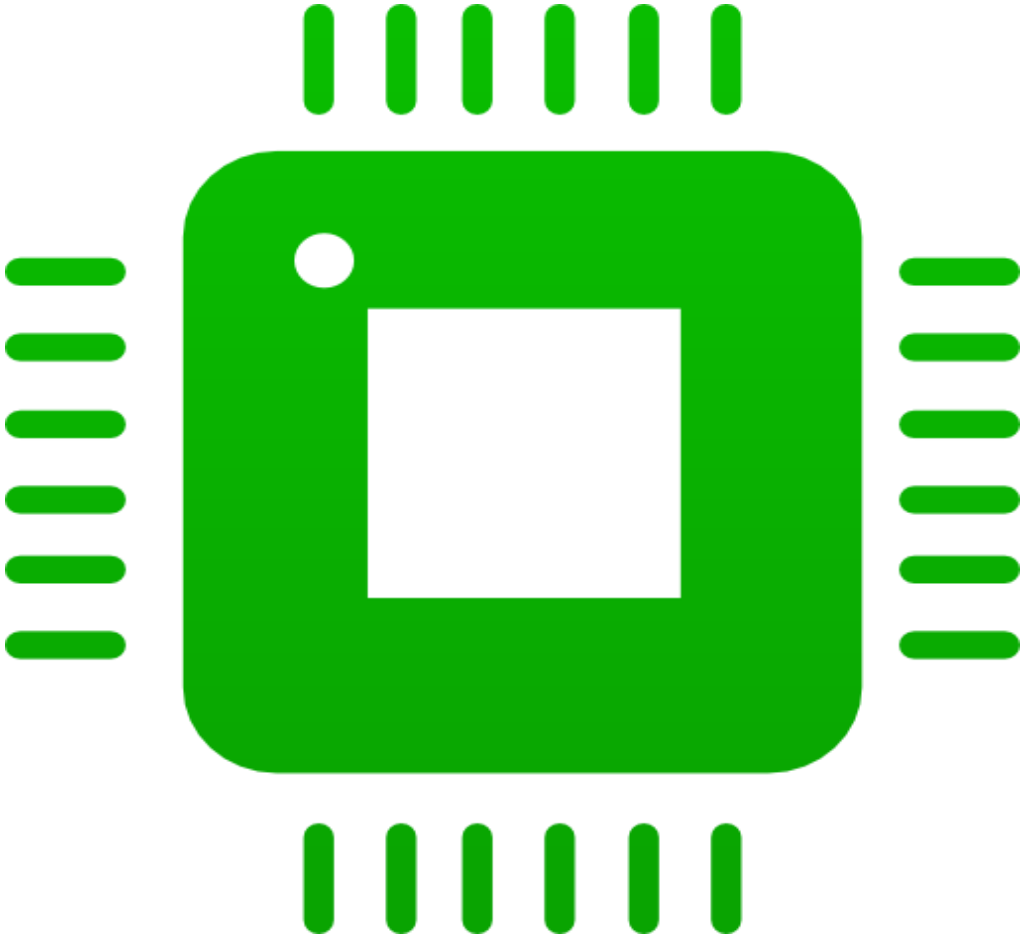
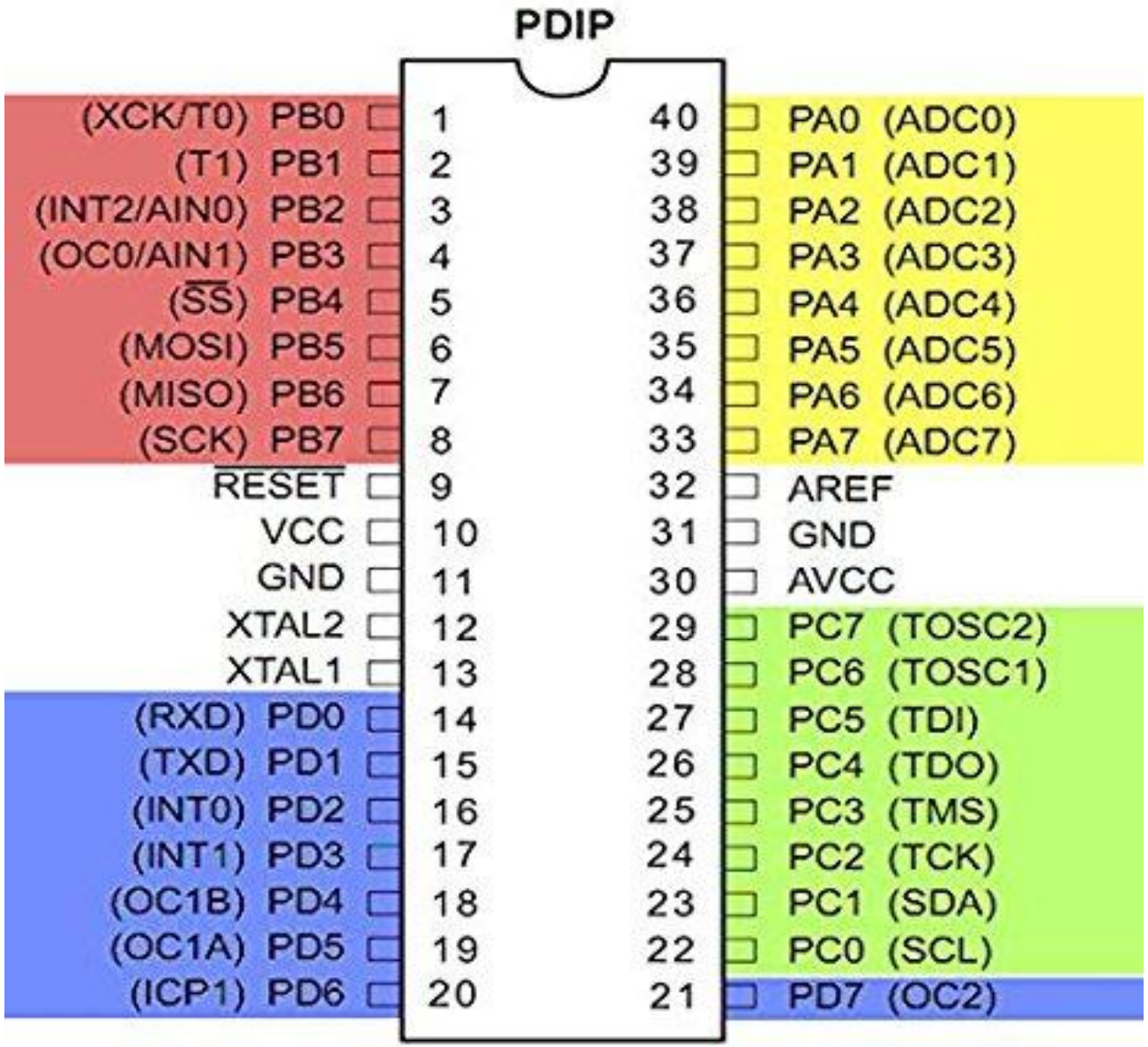
Fab Lab Ismailia represent :
Embedded Systems Workshop
by : Mohammed hemed



3- GPIO PORTs Interfacing

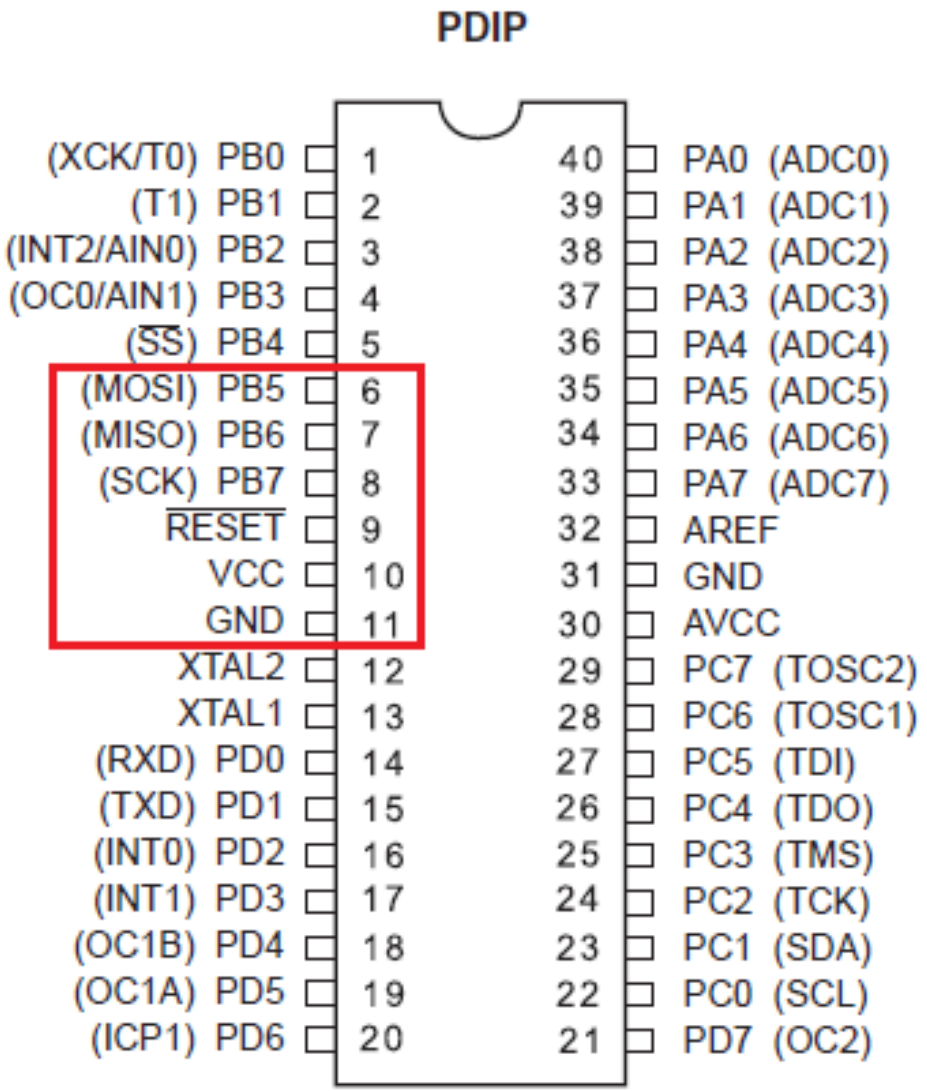
- General purpose Input Output
- Digital I/O basics
- Leds & buuton interfacing
- Seven segment
- DC motor
- Servo Motor

uC PORTs


PDIP
(XCK/T0) PB0 1
(T1) PB1 2
(INT2/AIN0) PB2 3
(OC0/AIN1) PB3 4
(SS) PB4 5
(MOSI) PB5 6
(MISO) PB6 7
(SCK) PB7 8
RESET 9
VCC 10
GND 11
XTAL2 12
XTAL1 13
(RXD) PD0 14
(TXD) PD1 15
(INT0) PD2 16
(INT1) PD3 17
(OC1B) PD4 18
(OC1A) PD5 19
(ICP1) PD6 20
40 PA0 (ADC0)
39 PA1 (ADC1)
38 PA2 (ADC2)
37 PA3 (ADC3)
36 PA4 (ADC4)
35 PA5 (ADC5)
34 PA6 (ADC6)
33 PA7 (ADC7)
32 AREF
31 GND
30 AVCC
29 PC7 (TOSC2)
28 PC6 (TOSC1)
27 PC5 (TDI)
26 PC4 (TDO)
25 PC3 (TMS)
24 PC2 (TCK)
23 PC1 (SDA)
22 PC0 (SCL)
21 PD7 (OC2)

usbasp programmer

MOSI	1	2	VTG
NC	3	4	GND
/RES	5	6	GND
SCK	7	8	GND
MISO	9	10	GND



Digital I/O Basics:

- AVR uCs have three ports to control any port :

- DDRx ---> Data Direction Register

- PORTx ---> Port Output Register

- PINx ---> Port Input Register

Where x = (A – B – C – D) ports .

1- DDRx : 8bit register used to control any port to be input or output , or even a specific bit in any port to be input = 0 , or outbut = 1 , to send signals to leds , motors , LCD , Relays , speaker , ...

Port A Data Direction Register – DDRA

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	

- PORTx Register : Read/Write reister is used to control output signal to be zero or one , HIGH or LOW .

Port A Data Register – PORTA

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	

- By default all pins of uC work as input port (not only AVR but also most of microcontrollers)
- PINx Register : Read Only register via it uC alaways read all pins and store them in PINx Register , this register is updated = the speed of uC , so if uC work on 1MHz this mean that PINx registers are updated 1 million update / second .
- This register is benefit in know the feedback of uC.

Port B Input Pins Address – PINB

Bit	7	6	5	4	3	2	1	0	
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
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	

Pull Up & Pull Down Resistor :

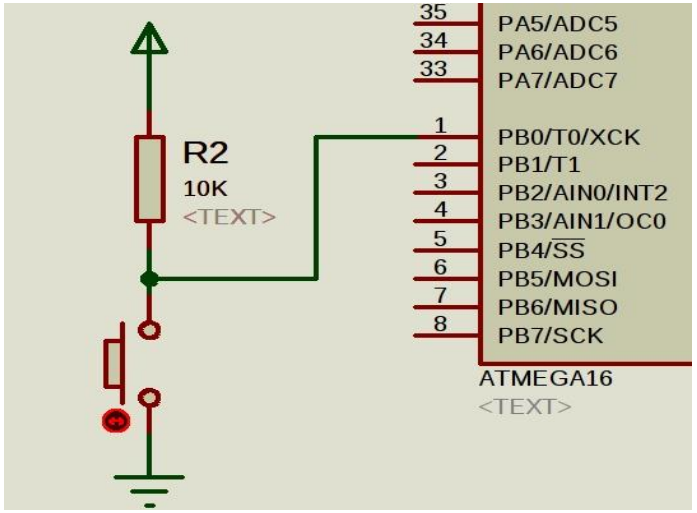
- In the world of digital electronics a common word you may hear many times

Pull up & down resistor used in digital gates (And – OR - NOT) .

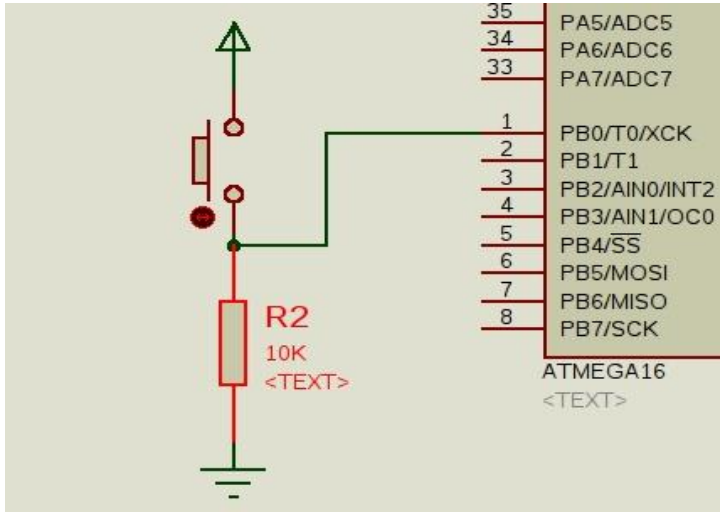
- Why we use them ?

- To overcome Floating area (noise margin) : diff between logic zero and logic one as if you put a button without using them uC will take random or crazy decisions .

Pull-Down

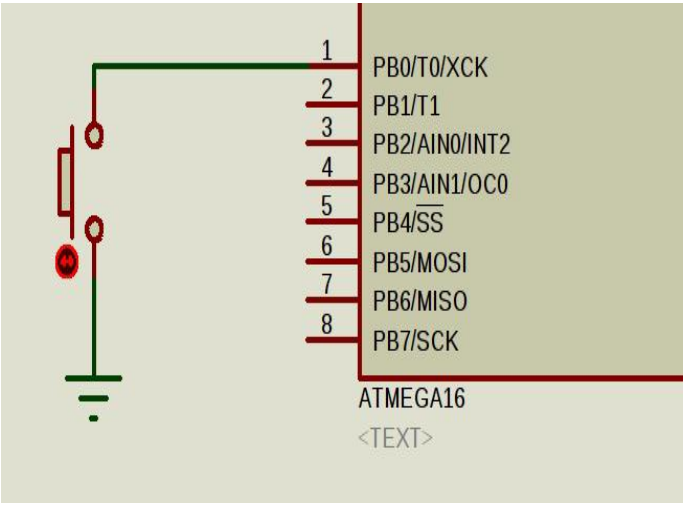


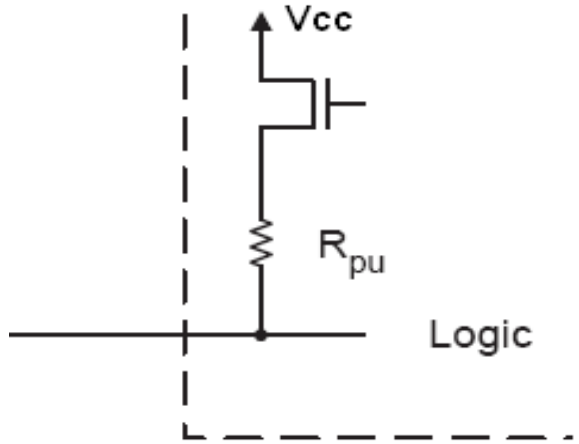
Pull-Up



Internal Pull-up resistor

- Most of AVR uCs have a nice feature called internal pull-up , enable you to use switches without need to external pull-ups .
- By default this resistor is disabled , the user enable it by make the pin work as input then write in PORTx register to enable it .
- When you read PINx register the reading is inverted .





Bouncing and de-bouncing

- Bouncing Effect : Is a phenomenon happened to mechanical switches especially push buttons , they consist of two metal sheets when you press , contact , but because of the way of manufacture of them the metal sheets make tens of contacts per second before being stable .
- This phenomenon is annoying to Embedded developers , as the switches send undesirable signals the affect the whole system .

So we have two solutions called de-bouncing :

- Software de-bouncing : via delay to make sure “twice” that the switch has pressed ,this way is not efficient in world of Embedded as the delay make the CPU idle especially in RTOS – Critical systems .

- Hardware de-bouncing : via filter circuit to filter the electric oscillation

by connecting a capacitor in parallel with the switch = 0.1uF

Light Emitting Diode (LED) :

- Light Emitting Diode (LED) : type of diode that convert electrical energy into light .
- The longer side is the positive .
- The wavelength of light, and therefore the color, depends on the type of semiconductor material used to make the diode. That's because the energy band structure of semiconductors differs between materials, so photons are emitted with differing frequencies.
- The unit for measuring luminous intensity is called the candela, although when you're talking about the intensity of a single LED you're usually in the millicandela range
- The luminous intensity of LEDs can range from the tens to the tens-of-thousands of millicandela.

Led types

- RGB (Red-Green-Blue) LEDs are actually three LEDs in one! But that doesn't mean it can only make three colors. Because red, green, and blue are the additive primary colors, you can control the intensity of each to create every color of the rainbow.

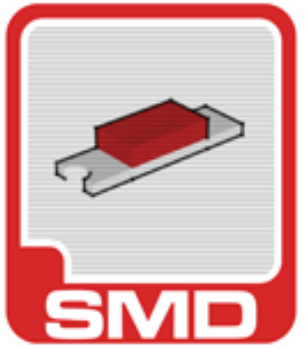
Some leds are smarter than others for ex :

- Flashing leds : Inside these LEDs, there's actually an integrated circuit that allows the LED to blink without any outside controller.

- SMD LEDs : aren't so much a specific kind of LED but a package type.

As electronics get smaller and smaller, manufacturers have figured out how to cram more components in a smaller space.

SMD (Surface Mount Device) parts are tiny versions of their standard counterparts.



- High-Power LEDs : Generally, an LED is considered High-Power if it can dissipate 1 Watt or more of power , Arrays of them can even be built for spotlights and automobile headlights.

- Special LEDs : There are even LEDs that emit light outside of the normal visible spectrum.

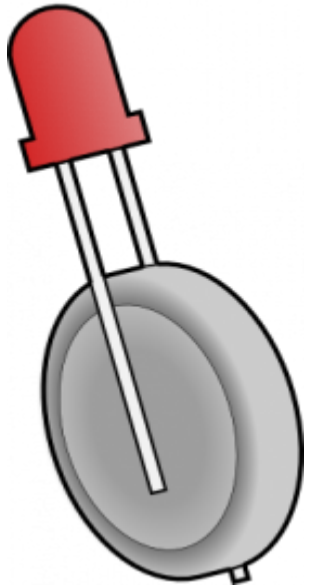
You probably use Infrared LEDs every day, for instance.

They're used in things like TV remotes to send small pieces of information in the form of invisible light! On the opposite end of the spectrum you can also get Ultraviolet LEDs.

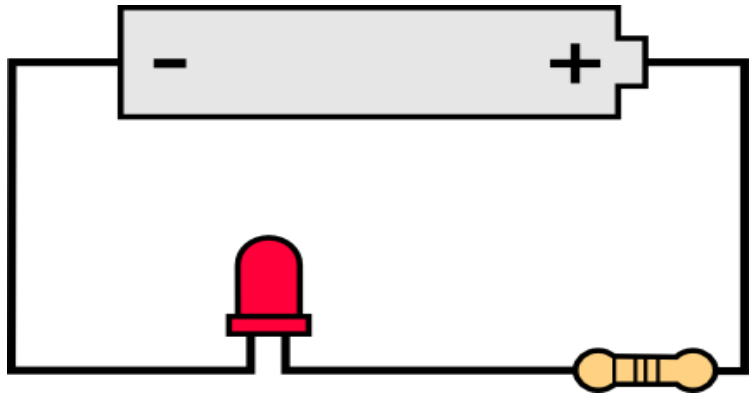
Ultraviolet LEDs will make certain materials fluoresce, just like a blacklight!

They're also used for disinfecting surfaces, because many bacteria are sensitive to UV radiation.

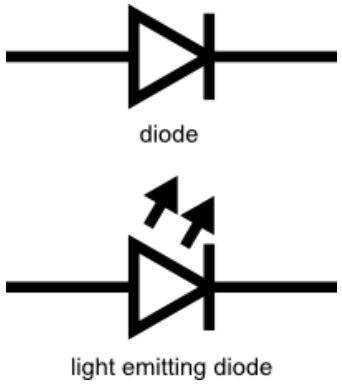
LED



Led test with Cell



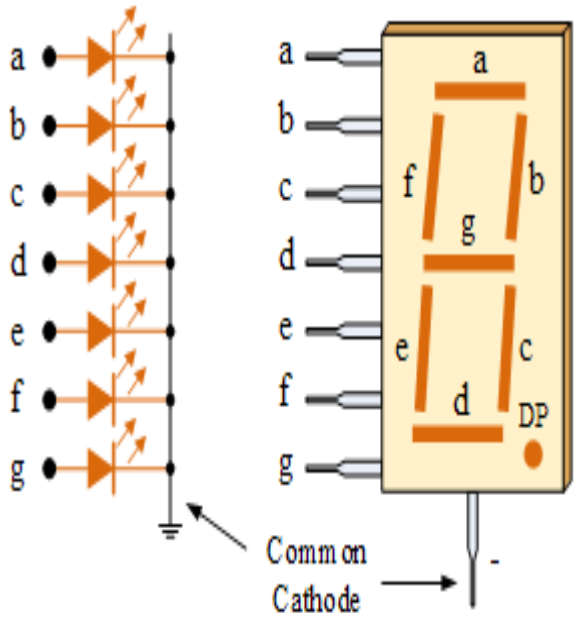
Led with resistor to limit current

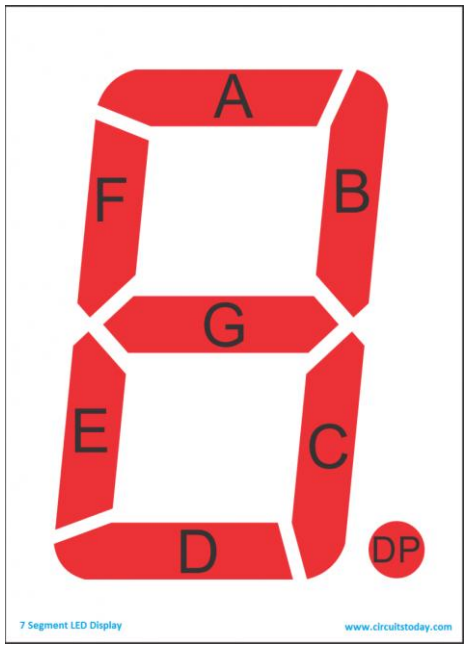


Led Symbol

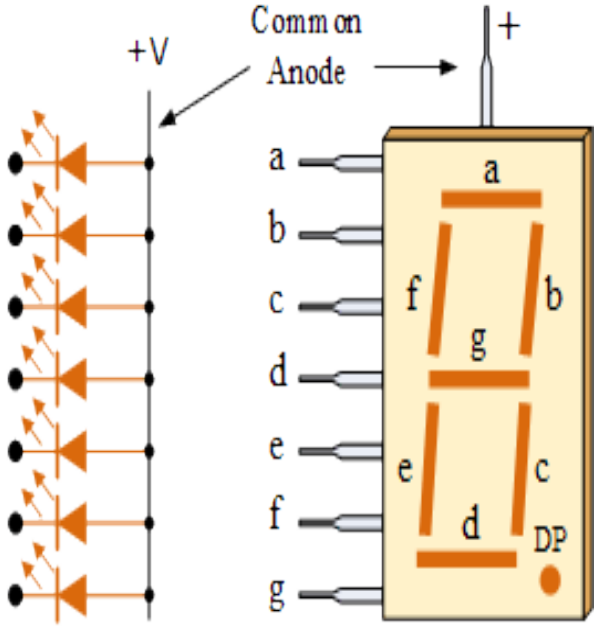
Seven Segment

- Consist of seven leds arranged in a rectangular fashion to display numbers and some English letters , You may see them in traffic lights - old cheap digital watch . Its colors (red - green - blue) .



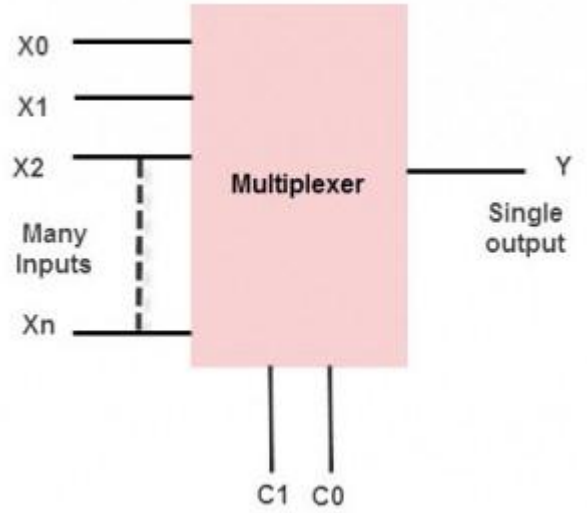


DIGIT	LEDs TO GLOW						
	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1



Seven segment multiplexing

- what multiplexing ?
- A Multiplexer is a device that allows one of several analog or digital input signals which are to be selected and transmits the input that is selected into a single medium. Multiplexer is also known as Data Selector



- The Multiplexer acts as a multiple-input and single-output switch.
- Why multiplexing ?
- Often we need to use two, three or more SSDs and that too using only a single MCU, but one problem that we face is the lack of I/O pins in the MCU, as one SSD would take 8 pins, and so three SSDs would take 24 pins , so we solve this problem by use the same 7 output to all seven segment via Multiplexing .
- so you will use 10 pins only to display 3 SSD instead of 24 pins , then we saved 14 pins to control other devices – read sensors ,.....

Control large loads

- As you saw in the electrical specs section in uC datasheet , the uC alone isn't able to control large loads so we use transistors (BJT up to 5A and Mosfet up to 40A) .
- DC current per I/O = 40 mA
- Most of uC couldn't provide more than 50mA per pin so we use Driver Circuits

Electrical Characteristics

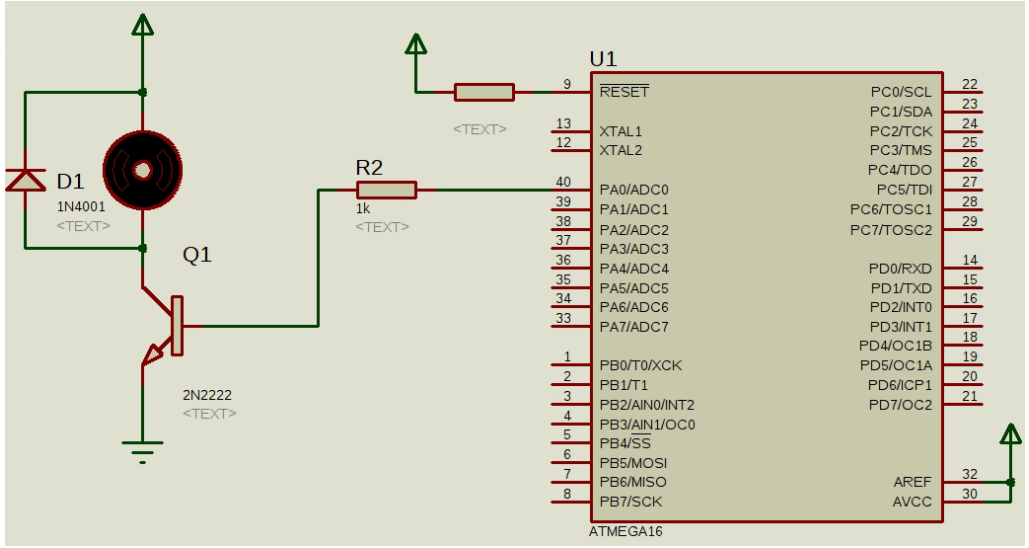
Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to V_{CC}+0.5V
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins	200.0 mA PDIP and 400.0 mA TQFP/MLF

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Motor

- Motors is used to convert
- Motors is one of the most common Electromechanical elements , we usually use in Robots . DC motor is one of them , it consumes a lot of current , so we couldn't connect it directly to uC , and must use Driver circuit .
- In proteus there is component called (Animated DC) you could use it In simulation .
- Note that there is a diode connect in parallel with the motor to prevent the EMF effect from dam



H-Bridge

- So what if you want to run the motor in two different directions ?
- We can do that by using H-Bridge : it's consist of 4 transistors looks like 'H'
- Every two swithes work with each others so if we close S1 – S4 the motor will run clockwise , and if we close S2 – S3 the motor will run anticlockwise

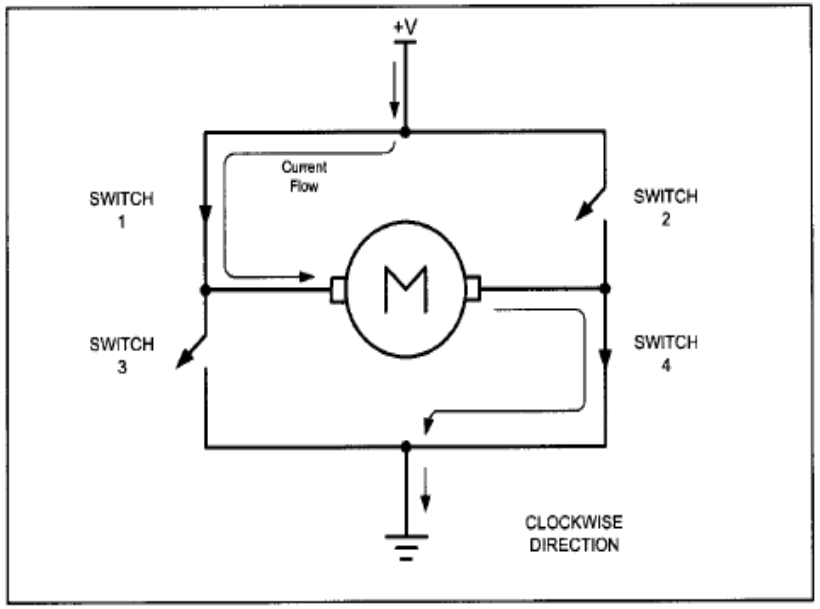


Figure 16-3. H-Bridge Motor Clockwise Configuration

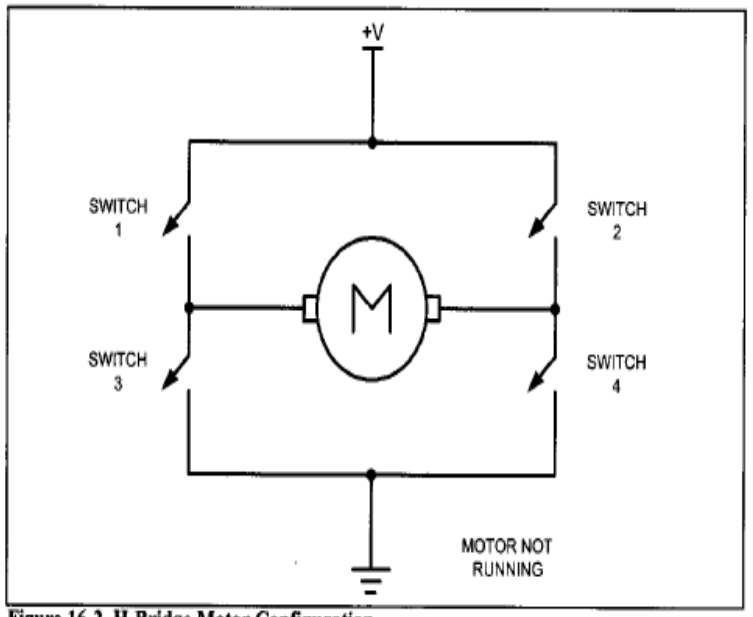


Figure 16-2. H-Bridge Motor Configuration

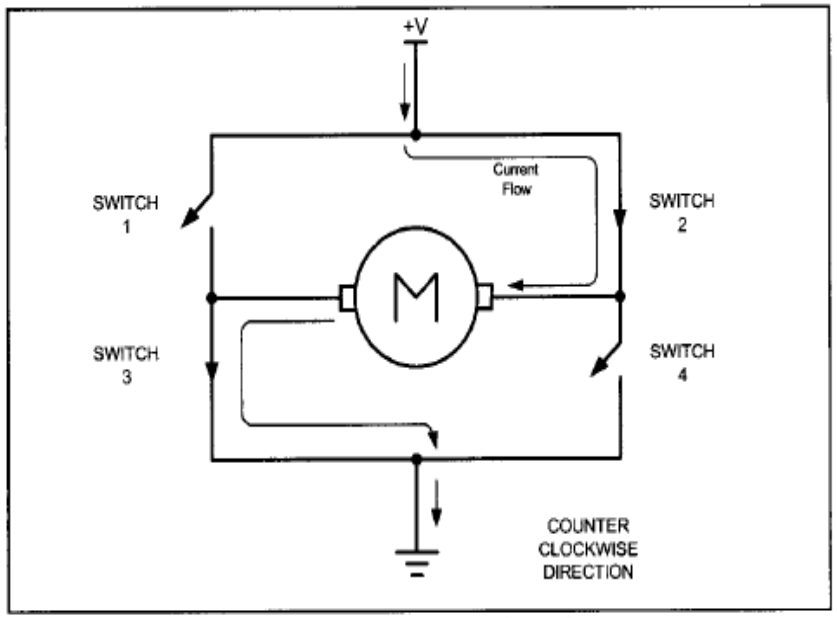
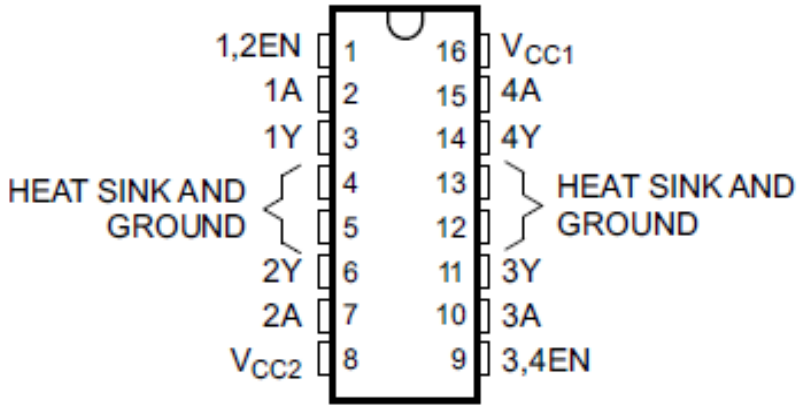


Figure 16-4. H-Bridge Motor Counterclockwise Configuration

L293D Chip

- Each chip contains two full H-bridges (four half H-bridges). That means you can drive four solenoids, two DC motors bi-directionally, or one stepper motor.
- There's a PWM input per driver so you can control motor speed. Runs at 5V logic. Good for motor voltages from 4.5V up to 36V! This wont work well for 3V motors. The motor voltage is separate from the logic voltage.

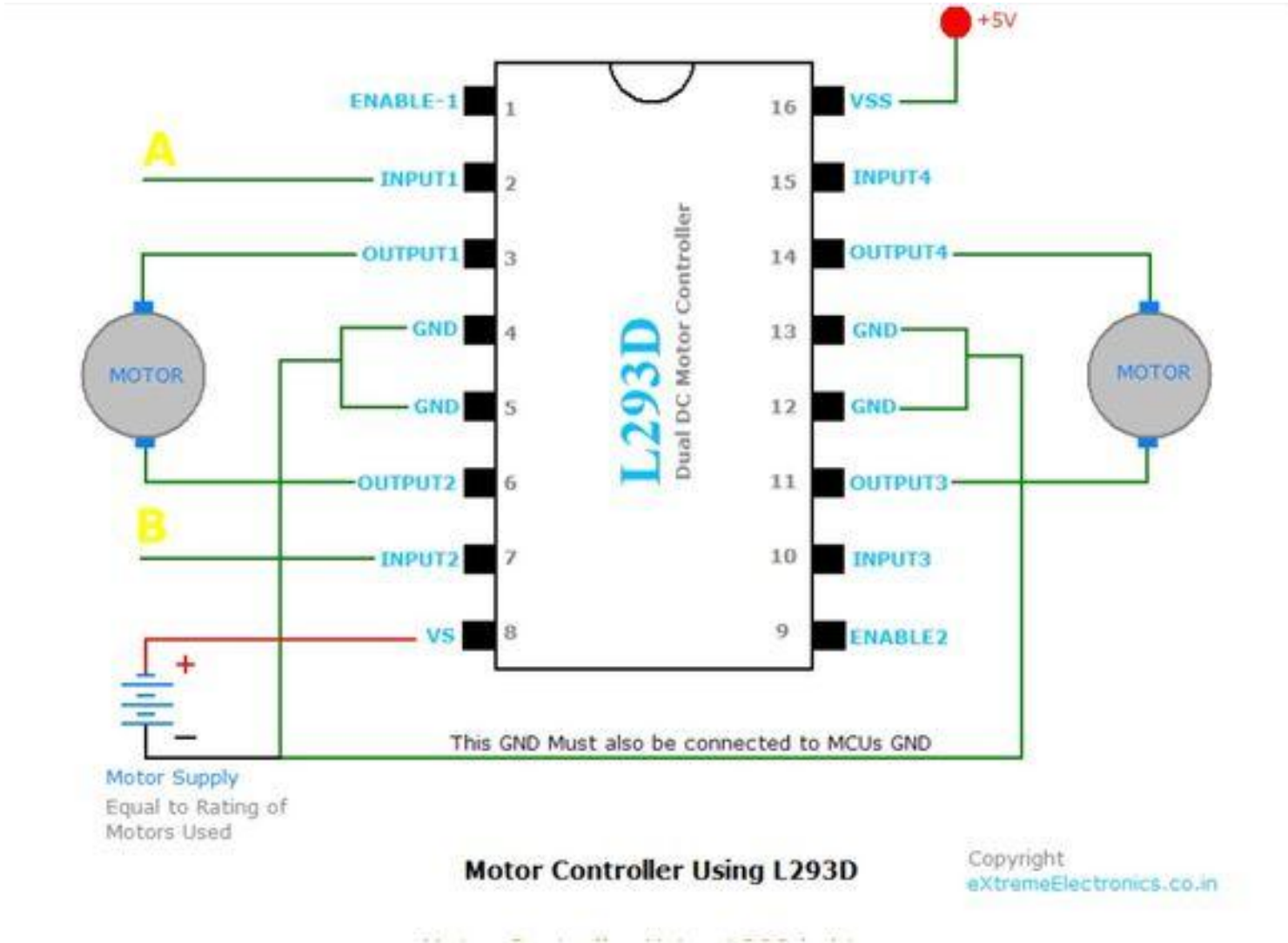
NE Package
16-Pin PDIP
Top View



Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
1,2EN	1	I	Enable driver channels 1 and 2 (active high input)
<1:4>A	2, 7, 10, 15	I	Driver inputs, noninverting
<1:4>Y	3, 6, 11, 14	O	Driver outputs
3,4EN	9	I	Enable driver channels 3 and 4 (active high input)
GROUND	4, 5, 12, 13	—	Device ground and heat sink pin. Connect to printed-circuit-board ground plane with multiple solid vias
V_{CC1}	16	—	5-V supply for internal logic translation
V_{CC2}	8	—	Power VCC for drivers 4.5 V to 36 V

L293D with two motors



Pulse Width Modulation

- Changing or modulating the width of pulse applied to a load like a DC motor , then we can increase or decrease the amount of power provided to the motor then control the speed of motor , Although the voltage has a fixed amplitude , it has a variable duty cycle = Ton pulse + Toff pulse , that mean the wider the pulse, the higher the speed .

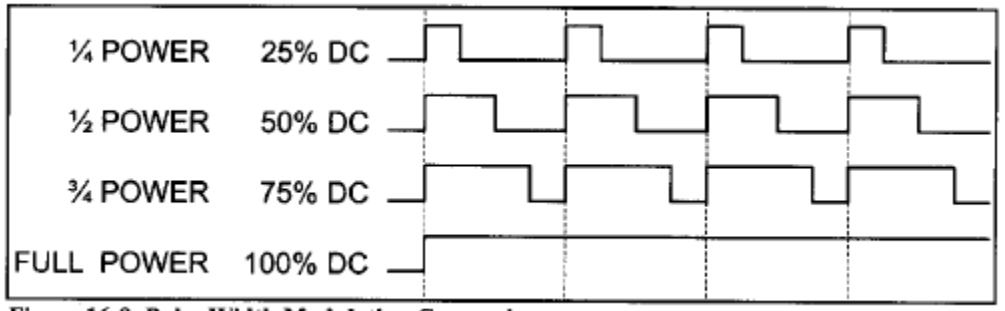


Figure 16-9. Pulse Width Modulation Comparisons

- Most of uCs have an embedded PWM module , just we load the proper registers with the values of high and low portions of the desired pulse ,

  this allows our uC to do other things .
- If this feature isn't embedded in uC we have to create the various duty cycle pulses using software via (delay) , which prevents the uC from doing other things as we said during the delay our uC is idel .

Code example to control DC motor by PWM

Example 16-5 (C version of Example 16-3)

Refer to the figure in this example. Write a C program to monitor the status of SW and perform the following:

(a) If SW = 0, the DC motor moves with 50% duty cycle pulse.
(b) If SW = 1, the DC motor moves with 25% duty cycle pulse.

Solution:

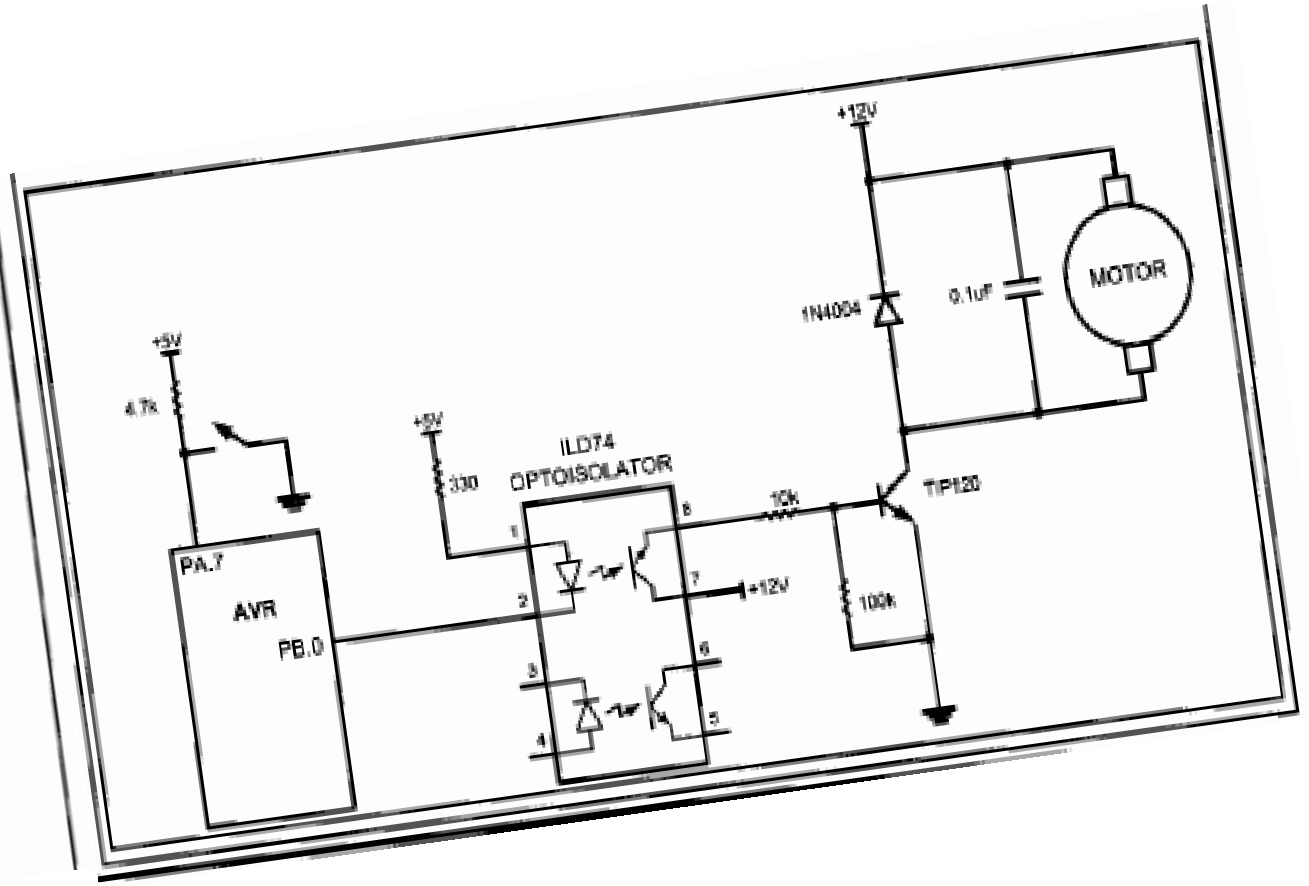
```
#define F_CPU    8000000UL          //XTAL = 8 MHz
#define SW       (PORTA&(1<<7))

#include "avr/io.h"
#include "util/delay.h"

void main()
{
    DDRA=0x7F;              //make PA7 input pin
    DDRB=0x01;              //make PB0 output pin
    while(1)
    {
        if(SW == 1)
        {
            PORTB = PORTB | (1<<0);
            _delay_ms(75);
            PORTB = PORTB & (~(1<<0));
            _delay_ms(25);

        }
        else
        {
            PORTB = PORTB | (1<<0);
            _delay_ms(50);
            PORTB = PORTB & (~(1<<0));
            _delay_ms(50);

        }
    }
}
```

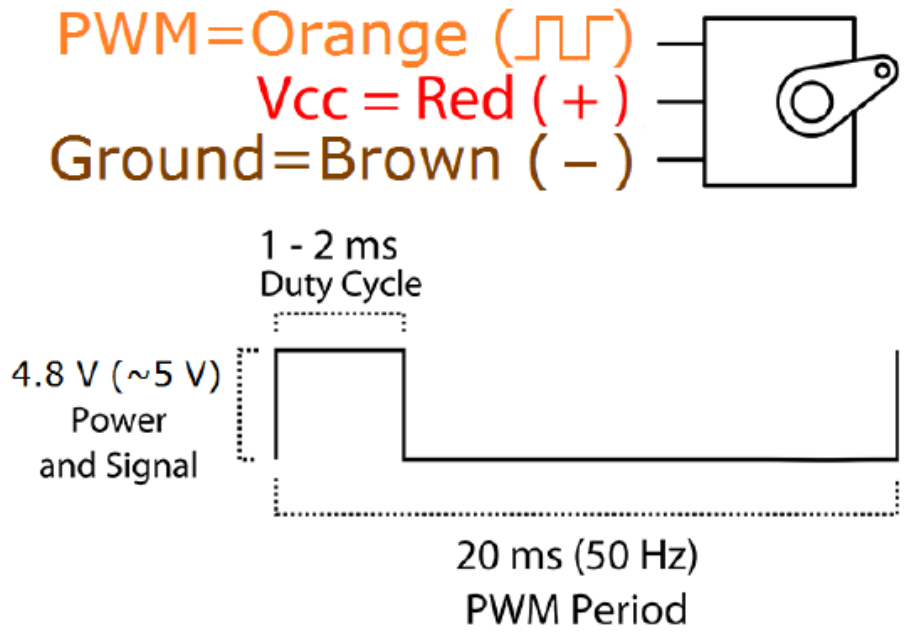


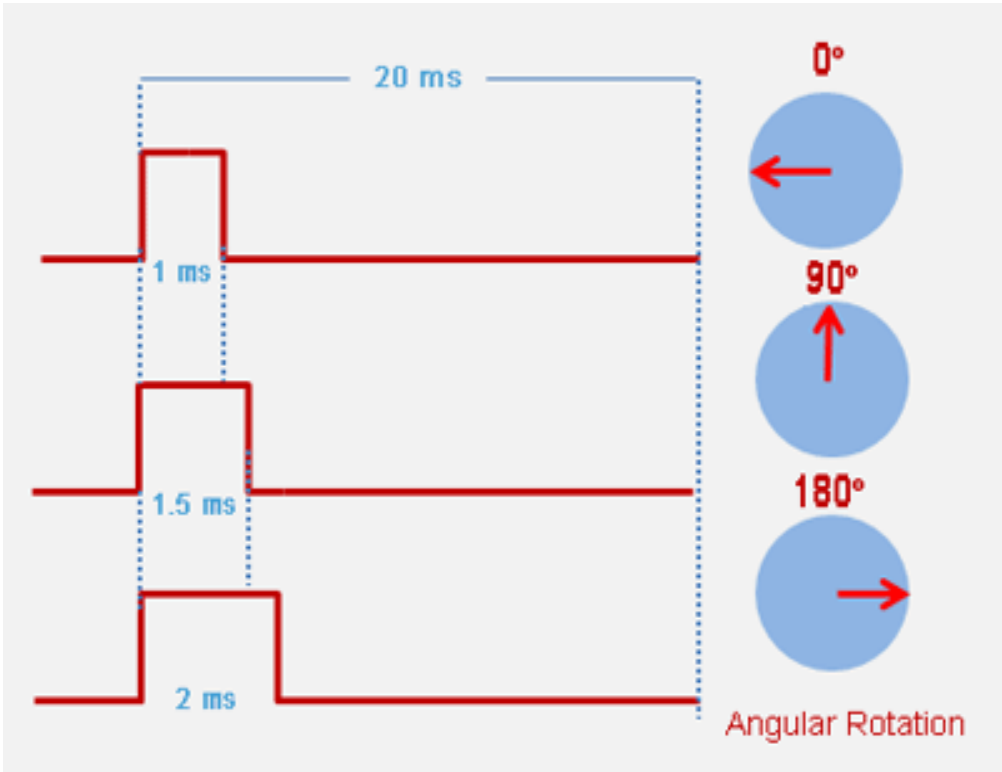
Servo Motor

- Servo motors are common in robots , they are used in robotic arms and legs, RC toys like RC helicopter, airplanes and cars.
- "servomechanism" (servo for short) is a device that uses feedback to achieve the desired result.
- Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction)

Sg90 micro servo

Servo Control via PWM





Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.

References :

- Books :
- Simply AVR - > Abdallah Ali
- The AVR microcontroller & Embedded Systems using Assembly & c -- > Mazidi
- ATMEGA 32A Datasheet
- PIC microcontroller -- > Milan Verle
- Websites :
- https://www.sparkfun.com
- http://maxembedded.com
- https://www.tutorialspoint.com/cprogramming
- https://stackoverflow.com
- https://www.quora.com
- https://www.lucidchart.com

Any questions ?

- Instructor : Mohammed Hemed

- Embedded Systems developer at fab lab Ismailia

Repository link of Embedded workshop Material:

https://github.com/FabLab-Ismailia/Embedded-Systems-Workshop

Contact me :

- Gmail :

mohammedhemed23@gmail.com

- Linkedin :

https://www.linkedin.com/in/mohammedhemed

See you „„