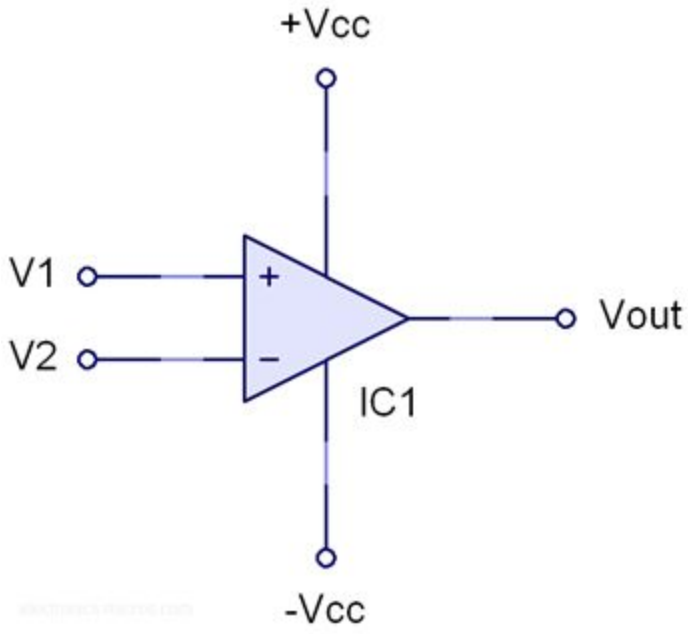
CORRIDOR FOLLOWING USING ULTRASONIC DISTANCE SENSOR AND OMNI DRIVE

DAY 1

OMNI WHEELS

Omni wheels or poly wheels, similar to Mecanum wheels, are wheels with small discs around the circumference which are perpendicular to the turning direction. The effect is that the wheel can be driven with full force, but will also slide laterally with great ease.

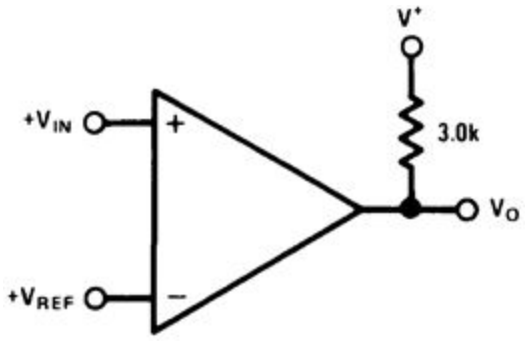
COMPARATOR



Comparator is used to convert analogue voltage signal to digital.

when V1 > V2 then Vout = floating
when V1 < V2 then Vout = 0

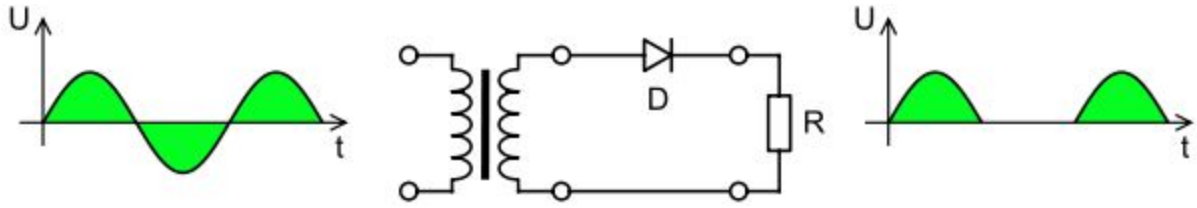
pull up or pull down resistors can be used to convert floating voltage to a fixed value.



RECTIFIER

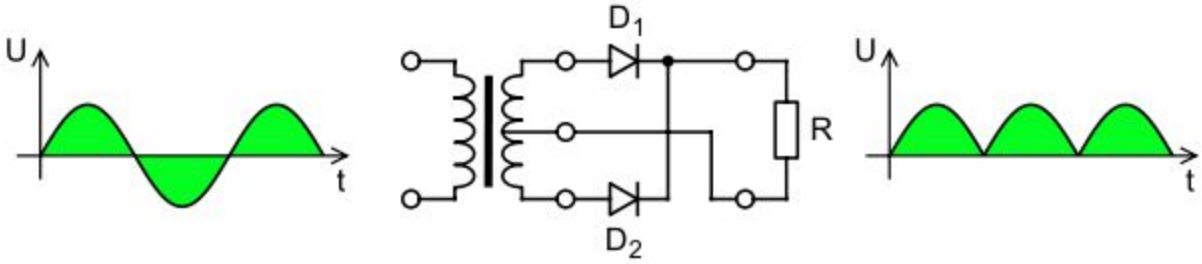
HALF WAVE RECTIFIER

If a diode is connected across ac voltage supply it allows current flow only in one direction thus allowing current to flow only for the positive half of the voltage cycle. Thereby we get voltage supply which is in a single direction but varies and has gaps of zero volt.



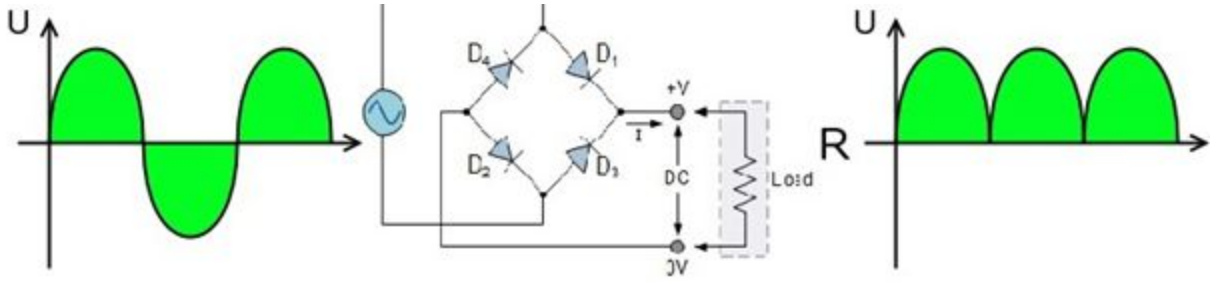
FULL WAVE RECTIFIER

Full wave rectifier convert ac voltage supply into voltage supply which is in a constant direction has no gaps of zero voltage but varies sinusoidally .It can be achieved using a centre tap transformer and two diodes or using four diodes in bridge configuration.



full wave rectifier using centre tap transformer and 2 diodes

For such rectifier voltage in secondary coil should twice the intended voltage so each half gets the intended voltage.



full wave rectifier using 4 diodes in bridge configuration

The second option is better as it costs less and less power loss.

FILTER CIRCUIT

A capacitors can be used to decrease the variation in voltage as it stores charge when voltage increase and releases charge when voltage decreases.

The small dip in voltage is known as ripple voltage.

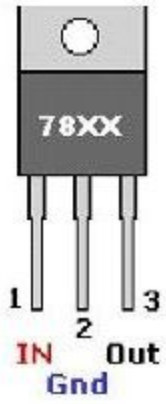
VOLTAGE REGULATOR IC

IC78xx

voltage regulator are used to give constant dc voltage supply . the xx signifies the output voltage.

It can be IC7805 , IC7806 , IC7809 , IC7812.

it has three terminals one for ground one for input and one for output.



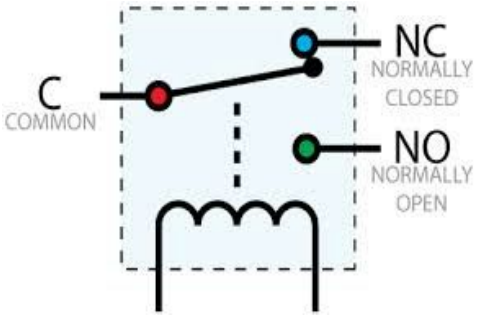
The voltage drop is provided by heat loss through ground therefore it is recommended to connect heat sink
the input voltage should be greater than the rated output.

LDR

A Light Dependent Resistor (LDR) or a photo resistor is a device whose resistivity is a function of the incident electromagnetic radiation. Hence, they are light sensitive devices. They are also called as photo conductors, photoconductive cells or simply photocells. They are made up of semiconductor materials having high resistance.

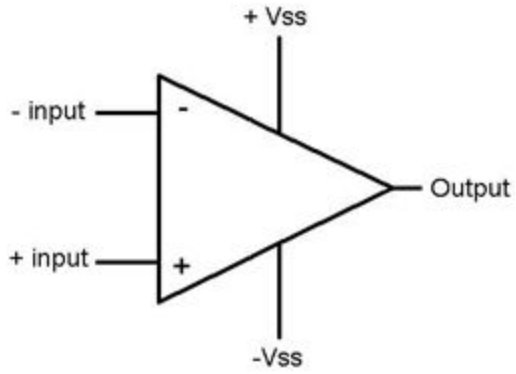
LDR's are light dependent devices whose resistance decreases when light falls on them and increases in the dark. When a light dependent resistor is kept in dark, its resistance is very high. And if the device is allowed to absorb light its resistance will decrease drastically.
LDR's have low cost and simple structure. They are often used as light sensors. They are used when there is a need to detect absences or presences of light.

RELAYS



Relays are switches that open and close circuits electromechanically or electronically. They control one electrical circuit by opening and closing contacts in another circuit. When a relay contact is normally open (NO), there is an open contact when the relay is not energized.

OP-amp(OPERATIONAL AMPLIFIER)

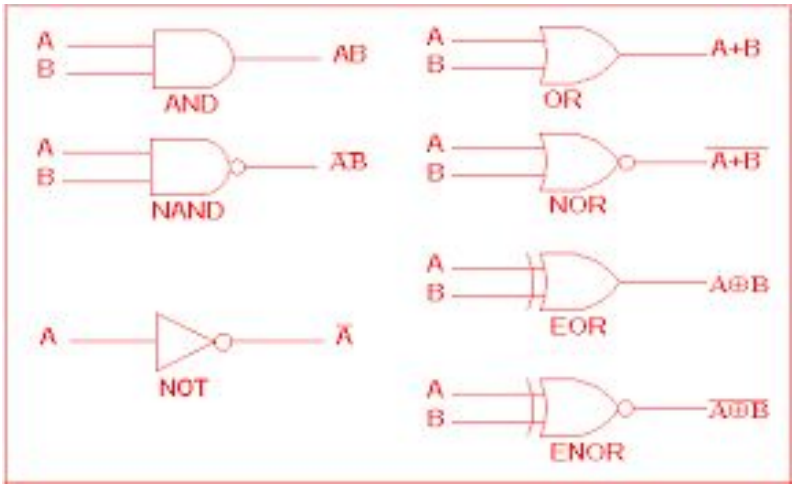


The amplifier's differential inputs consist of a non-inverting input (+) with voltage V_+ and an inverting input (–) with voltage V_-; ideally the op-amp amplifies only the difference in voltage between the two, which is called the *differential input voltage*.
Vout =Av(V+ -V-)
the amplification is limited by Vss.
|V| <= Vss

Key points-

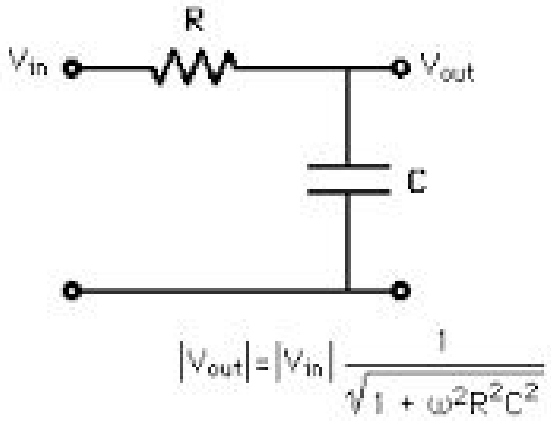
1.input resistance is high
2.output resistance is low
3.amplification factor is high.

LOGIC GATES



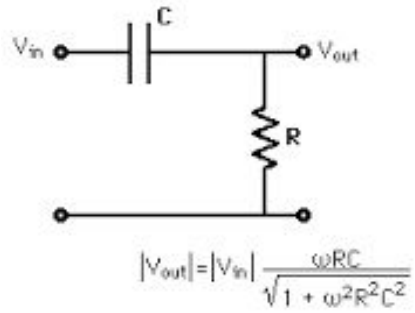
NAND and NOR gates are universal gates.

LPF(LOW PASS FILTER)



A low-pass filter is a filter that passes signals with a frequency lower than a certain cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency. The amount of attenuation for each frequency depends on the filter design.

HPF(HIGH PASS FILTER)

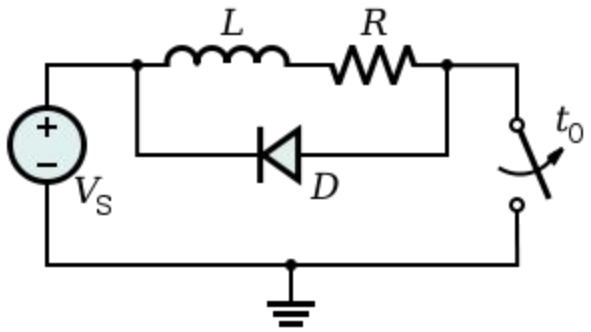


A High Pass Filter or HPF, is the exact opposite to that of the previously seen LowPass filter circuit, as now the two components have been interchanged with the output signal (Vout) being taken from across the resistor as shown.

POLAR AND CERAMIC CAPACITORS

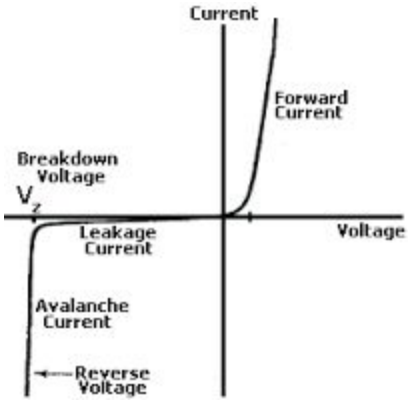
Polar capacitors have some internal inductance.These are basically used for energy filter and helps in ripple conversion.
ceramic capacitors are used to eliminate unwanted high frequency noise from the input waveform.

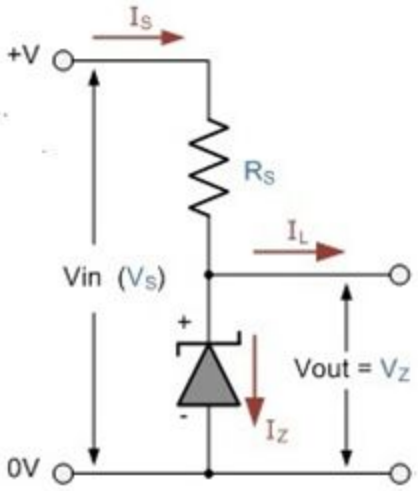
FLYBACK DIODE



A flyback diode is a diode used to eliminate flyback, which is the sudden voltage spike seen across an inductive load when its supply voltage is suddenly reduced or removed.

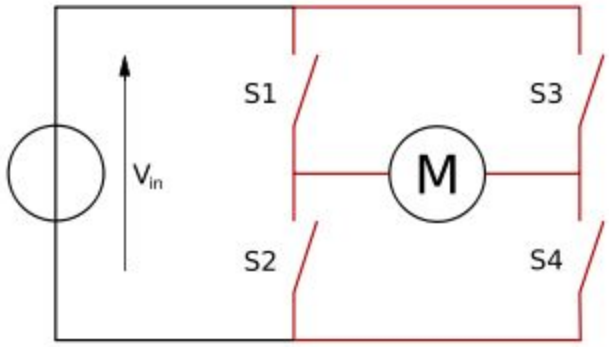
ZENER DIODE





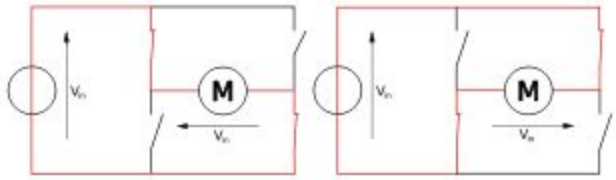
Zener diode is used to create constant voltage source.It is basically a standard pn junction diode with a low predefined reverse breakdown voltage . when operated under reverse bias with voltage greater than breakdown voltage the current increases greatly . potential difference across it remains almost constant for very large changes in current.

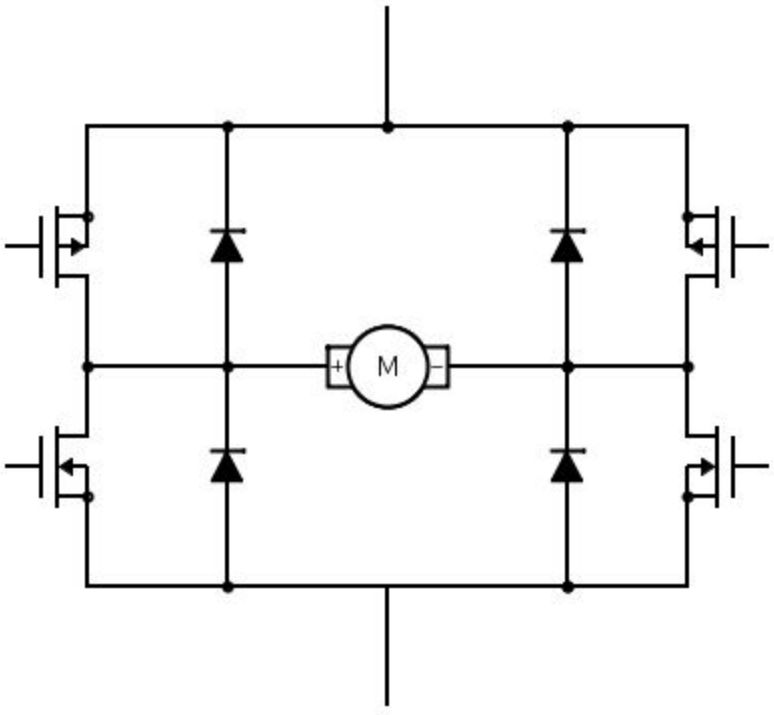
H bridge



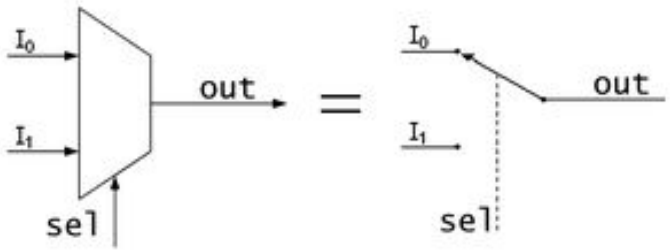
H bridge circuit can be used to make a motor run in clockwise or anticlockwise direction using four switches

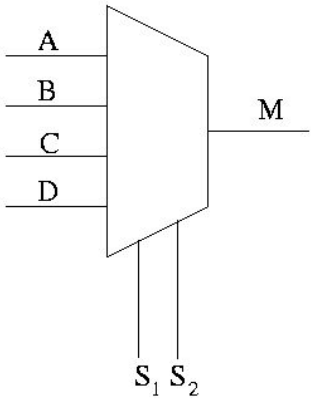
S1 and S2 closed - clockwise
S3 and S4 closed - anti-clockwise
S1 and S3 closed - breaking





MULTIPLEXER

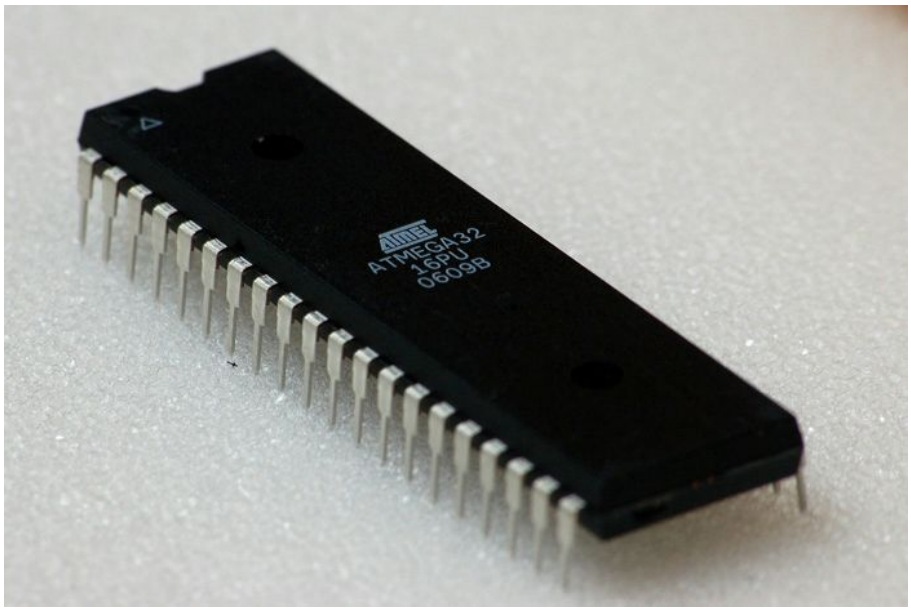




In electronics, a multiplexer (or mux) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. A multiplexer of 2^n inputs has n select lines, which are used to select which input line to send to the output.

DAY 2

MICROCONTROLLER AND MICROPROCESSOR



Microprocessor is an IC which has only the CPU inside them i.e. only the processing powers such as Intel's Pentium 1,2,3,4, core 2 duo, i3, i5 etc. These microprocessors don't have RAM, ROM, and other peripheral on the chip. A system designer has to add them externally to make them functional. Application of microprocessor includes Desktop PC's, Laptops, notepads etc

But this is not the case with Microcontrollers. Microcontroller has a CPU, in addition with a fixed amount of RAM, ROM and other peripherals all embedded on a single chip. At times it is also termed as a mini computer or a computer on a single chip. Today different manufacturers produce microcontrollers with a wide range of features available in different versions. Some manufacturers are ATMEL, Microchip, TI, Freescale, Philips, Motorola etc.

MEMORY

1. RAM

RAM is an acronym for random access memory, a type of computer memory that can be accessed randomly; that is, any byte of memory can be accessed without touching the preceding bytes. RAM is the most common type of memory found in computers and other devices, such as printers.

2. ROM

Read-only memory (ROM) is a class of storage medium used in computers and other electronic devices. Once data has been written onto a ROM chip, it cannot be removed and can only be read.
Unlike main memory (RAM), ROM retains its contents even when the computer is

turned off. ROM is referred to as being nonvolatile, whereas RAM is volatile.

3. EPROM

EPROM (erasable programmable read-only memory) is programmable read-only memory (programmable ROM) that can be erased and re-used. Erasure is caused by shining an intense ultraviolet light through a window that is designed into the memory chip.

4. EEPROM

EEPROM stands for Electrically Erasable Programmable Read-Only Memory and is a type of non-volatile memory used in computers and other electronic devices to store small amounts of data that must be saved when power is removed.Unlike bytes in most other kinds of non-volatile memory, individual bytes in a traditional EEPROM can be independently read, erased, and rewritten.

What Is Motor Driver IC?

A motor driver IC is an integrated circuit chip which is usually used to control motors in autonomous robots. Motor driver ICs act as an interface between microprocessors in robots and the motors in the robot. The most commonly used motor driver IC's are from the L293 series such as L293D, L293NE, etc. These ICs are designed to control 2 DC motors simultaneously. L293D consist of two H-bridge. H-bridge is the simplest circuit for controlling a low current rated motor. L293D has 16 pins, they are comprised as follows:

Ground Pins - 4

Input Pins - 4

Output Pins - 4

Enable pins - 2

Voltage Pins - 2

.

Why We Need Motor Driver IC?

Motor Driver ICs are primarily used in autonomous robotics only. Also most microprocessors operate at low voltages and require a small amount of current to operate while the motors require a relatively higher voltages and current. Thus current cannot be supplied to the motors from the microprocessor. This is the primary need for the motor driver IC.

How Motor Driver Operates?

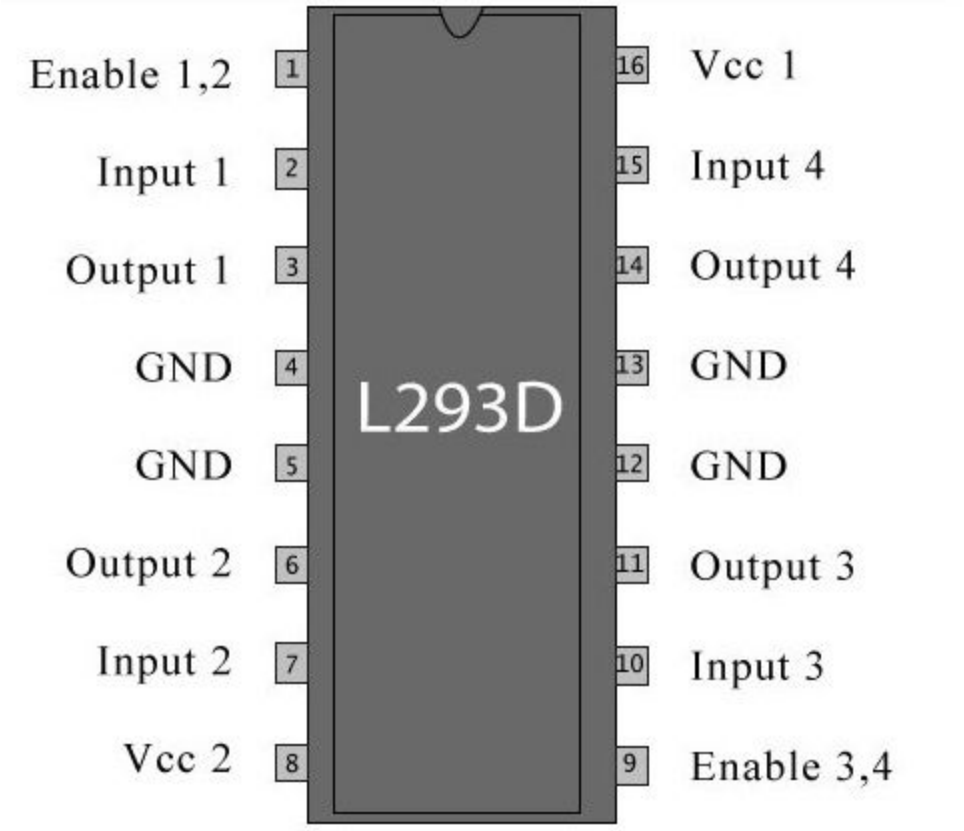
The L293D IC receives signals from the microprocessor and transmits the relative signal to the motors. It has two voltage pins, one of which is used to draw current for the working of the L293D and the other is used to apply voltage to the motors. The L293D switches it output signal according to the input received from the microprocessor.

For Example: If the microprocessor sends a 1(digital high) to the Input Pin of L293D, then the L293D transmits a 1(digital high) to the motor from its Output Pin. An important thing to note is that the L293D simply transmits the signal it receives. It does not change the signal in any case.

L293D AND ITS WORKING

The L293D is a 16 pin IC, with eight pins, on each side, dedicated to the controlling of a motor. There are 2 INPUT pins, 2 OUTPUT pins and 1 ENABLE pin for each motor. L293D consist of two H-bridge. H-bridge is the simplest circuit for controlling a low current rated motor.

L293D Pin Diagram



Pin No.	Pin Characteristics
1	Enable 1-2, when this is HIGH the left part of the IC will work and when it is low the left part won't work. So, this is the Master Control pin for the left part of IC
2	INPUT 1, when this pin is HIGH the current will flow though output 1
3	OUTPUT 1, this pin should be connected to one of the terminal of motor
4,5	GND, ground pins
6	OUTPUT 2, this pin should be connected to one of the terminal of motor
7	INPUT 2, when this pin is HIGH the current will flow through output 2
8	VC, this is the voltage which will be supplied to the motor. So, if you are driving 12 V DC motors then make sure that this pin is supplied with 12 V
16	VSS, this is the power source to the IC. So, this pin should be supplied with 5 V
15	INPUT 4, when this pin is HIGH the current will flow through output 4
14	OUTPUT 4, this pin should be connected to one of the terminal of motor
13,12	GND, ground pins
11	OUTPUT 3, this pin should be connected to one of the terminal of motor
10	INPUT 3, when this pin is HIGH the current will flow through output 3
9	Enable 3-4, when this is HIGH the right part of the IC will work and when it is low the right part won't work. So, this is the Master Control pin for the right part of IC

WORKING MECHANISM

Now depending upon the values of the Input and Enable the motors will rotate in either clockwise or anticlockwise direction with full speed (when Enable is HIGH) or with less speed (when Enable is provided with PWM).

Let us assume for Left Motor when Enable is HIGH and Input 1 and Input 2 are HIGH and LOW respectively then the motor will move in clockwise direction.

So the behaviour of the motor depending on the input conditions will be as follows :

INPUT1	INPUT 2	ENABLE 1,2	Result
0	0	1	Stop
0	1	1	Anti-clockwise rotation
1	0	1	Clockwise rotation
1	1	1	Stop
0	1	50% duty cycle	Anti-clockwise rotation with half speed
1	0	50% duty cycle	Clockwise rotation with half speed

Why 4 grounds in the IC?

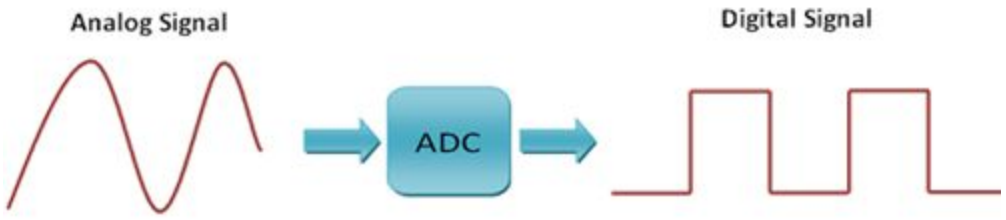
The motor driver IC deals with heavy currents. Due to so much current flow the IC gets heated. So, we need a heat sink to reduce the heating. Therefore, there are 4 ground pins. When we solder the pins on PCB, we get a huge metallic area between the grounds where the heat can be released.

POTENTIOMETER

A potentiometer , informally a pot, is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider. If only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat.

ANALOG TO DIGITAL CONVERTER (ADC)

Microcontroller understands only digital language. However, the inputs available from the environment to the microcontroller are mostly analog in nature, i.e., they vary continuously with time. In order to understand the inputs by the digital processor, a device called Analog to Digital Converter (ADC) is used. As the name suggests this peripheral gathers the analog information supplied from the environment and converts it to the controller understandable digital format, microcontroller then processes the information and provides the desired result at the output end.



The AVR ATmega16 has an inbuilt 8 channel, 10 bit analog to digital converter. Here, we will first convert a 5V signal and then a 0V signal with a reference voltage of 5V of ADC. After each conversion, the analog to digital converter of ATmega16 will give a 10-bit value for each signal (5V and 0V).These outputs of the analog to digital converter are displayed in a 1×8 LED array. When a 5V signal is converted, the output of analog to digital converter is 0x3ff (1023) and when 0V signal is converted, the output is 0x00 (0).

$$ADC = (V*1023)/5$$

PROGRAM FOR BLINKING OF LED

```
#define F_CPU 16000000UL

#include<avr/io.h>

#include<util/delay.h>

int main()

{

  DDRB=0b00000001;

  DDRB|=(1<<PB0);

  while(1)

    {

     PORTB|=(1<<PB0);

     _delay_ms(500);

    PORTB&=(1<<PB0);

    _delay_ms(500);

    }

return 0;

}
```

DAY 3

TIMER

As the name implies, timers can tell the time and count. Counting and timing allows for some really cool things, like controlling the brightness of LEDs, controlling the angle of servo shafts, receiving sensor data that transmit in PWM, making a timer, or just simply adding a time variable to your microcontroller project.

There are three timers present in ATmega16 , two 8-bit timers and one 16-bit timer. 8 bit timers are known as TCNT0 and TCNT2. 16-bit timer is named TCNT1. TCNT1 in advance have two registers namely TCCR1A and TCCR1B.

PRESCALE FACTOR

The timer and counter functions in the microcontroller simply count in sync with the microcontroller clock. However, the counter can only count up to either 256 (8-bit counter), or 65535 (16-bit counter). That's far from the 1,000,000 ticks per second that the standard AVR microcontroller provides.

The microcontroller provides a very useful feature called prescaling. Prescaling is simply a way for the counter to skip a certain number of microcontroller clock ticks. The AVR microcontrollers allow prescaling (skipping) numbers of: 8, 64, 256 and 1024. That is, if 64 is set as the prescaler, then the counter will only count every time the clock ticks 64 times. That means, in one second (where the microcontroller ticks one million times) the counter would only count up to 15,625. you could see that if the counter counts up to that number. Usually the prescale factor used in ADC conversion is 2^7 .

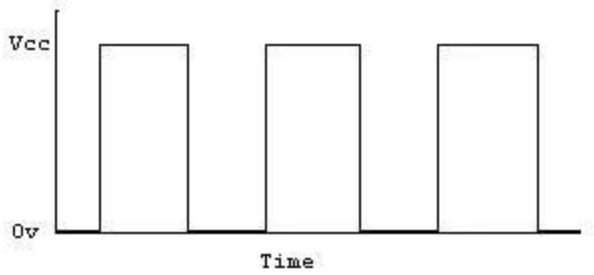
MODES OF OPERATION

1. NORMAL MODE

This is the simplest mode of operation. For 8-bit timer TCNT register starts from BOTTOM (0) and counts till TOP (255). At any point of time you can initialise the value of TCNT according to your wish and note the value of TCNT at other point of time. By this you can measure the time taken by the atmega16 to do the operation.

PULSE WIDTH MODULATION

As the name suggest we can get a hint that there is a Pulse which is generated here. To understand this concept I will consider a Pulse a square wave of voltage vs time.



Now what does Width Modulation mean. It means adjusting the time for which the source voltage will be HIGH and the time for which the source Voltage will be LOW.

We can control the Width Modulation by using the two properties of this Pulse

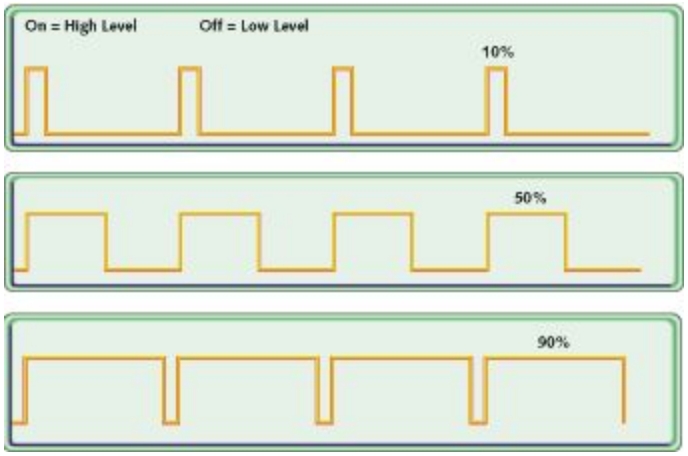
- **Frequency**
- **Duty Cycle**

FREQUENCY

The frequency is the time period of the Square Wave. Time period correspond to the time for one voltage cycle(HIGH to HIGH). If we half the frequency, the time period of one oscillation becomes doubled. So, you can see that is the graph when we adjust the frequency the width of the oscillation can be adjusted.

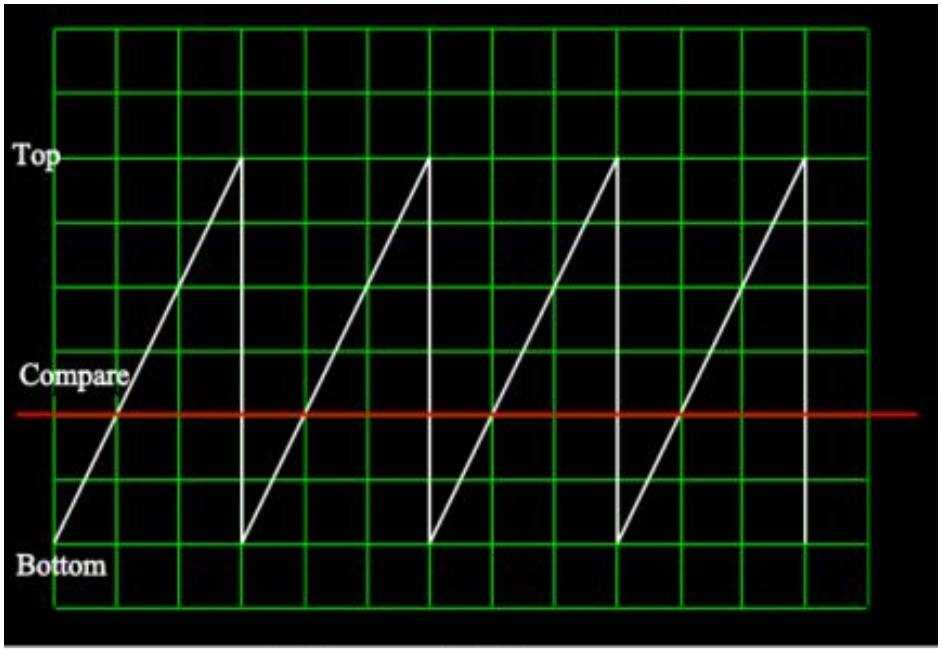
DUTY CYCLE

Duty Cycle is the percentage of time in a time period for which the Voltage is HIGH. The pulse given in the figure suggests that for half of the time the Voltage is HIGH and for the Rest half the Voltage is LOW. So, the duty cycle of the Pulse in the given diagram is 50%.

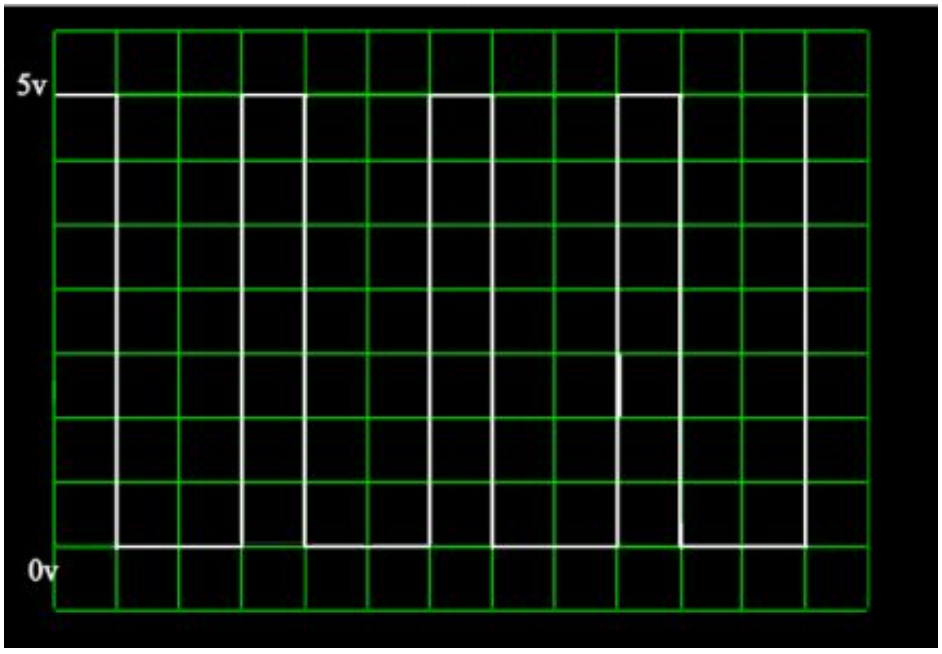


2. CLEAR TIMER ON COMPARE MATCH MODE (CTC)

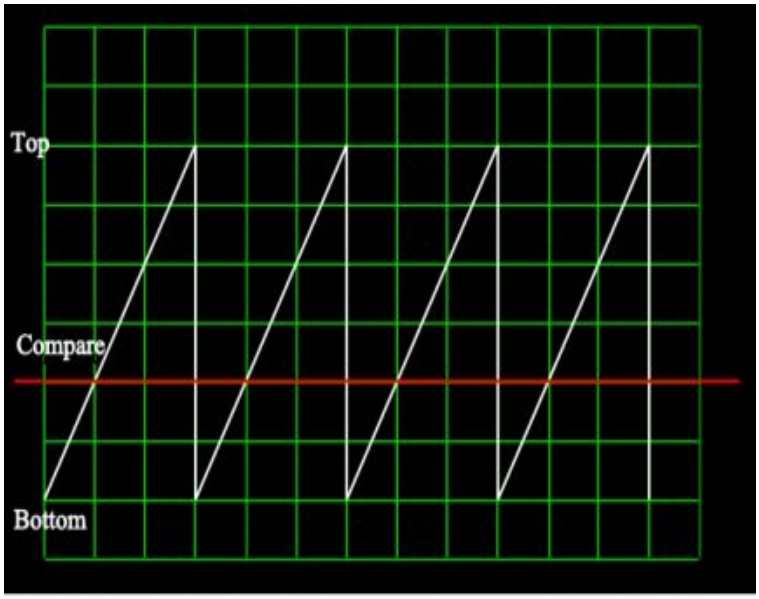
In this mode you can set the value of the TOP. So if you set the value of TOP as 100 instead of 255(default TOP value), the TCNT counts from BOTTOM (0) to 100 and then again drops to BOTTOM (0). So, if the counts of TCNT decrease then the time period of the oscillation decreases. So, in this mode we can adjust the frequency of the. For Adjusting the TOP value in this case we use the OCRn (Output Compare Register). In this mode the OCn pin(for 8-bit timers, OC0 and OC2) is the output pin for PWM.



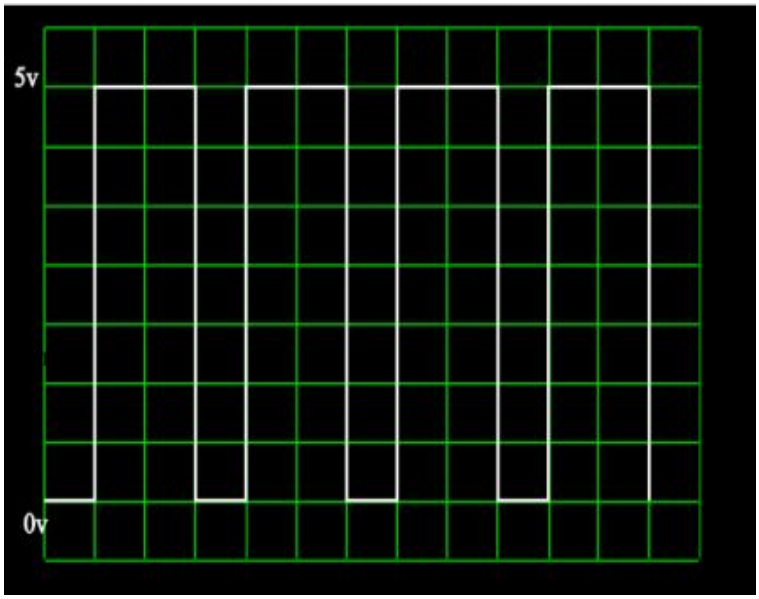
Produces an output of



3. SET ON COMPARE MATCH



Produces an output of



4. FAST PWM MODE

In this mode the OCn pin(for 8-bit timers, OC0 and OC2) is the output pin for PWM of duty-cycle dependent on the value of OCRn register. The TCNT value goes from BOTTOM(0) to TOP(255). When the value of TCNT matches with that of the OCR register, the OC pin can be reset/set according to the mode selected (non-inverting/ inverting). The diagram for How Fast PWM works is given below

5. PHASE CORRECT PWM

In this mode the TCNT increases from BOTTOM to MAX and then falls from MAX to BOTTOM. So in this case the duty cycle will remain same as that in the case of the Fast PWM but frequency is halved.

.

DAY 4

INTERRUPTS

Interrupts are basically events that require immediate attention by the microcontroller. When an interrupt event occurs the microcontroller pause its current task and attend to

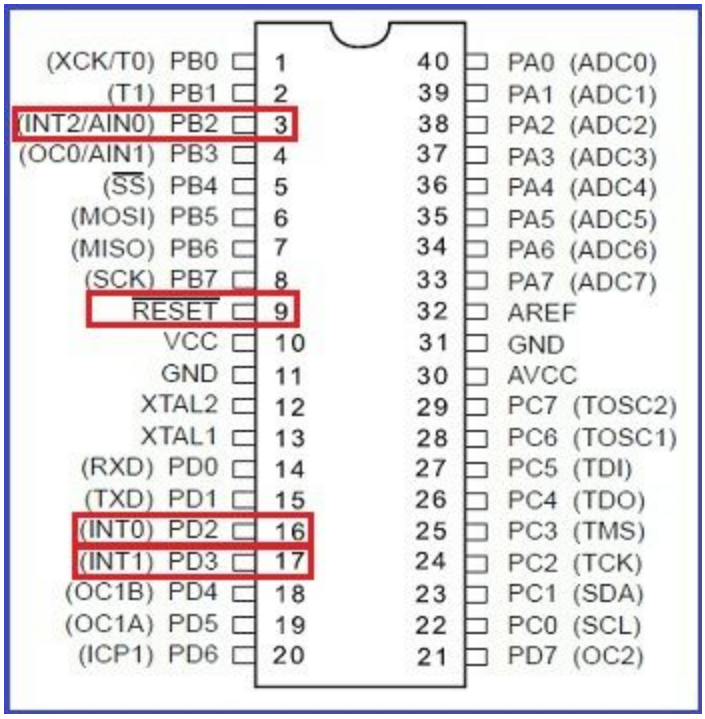
the interrupt by executing an Interrupt Service Routine (ISR) at the end of the ISR the microcontroller returns to the task it had pause and continue its normal operations.

In order for the microcontroller to respond to an interrupt event the interrupt feature of the microcontroller must be enabled along with the specific interrupt. This is done by setting the Global Interrupt Enable bit and the Interrupt Enable bit of the specific interrupt.

INTERRUPT SOURCES PROVIDED WITH AVR MICROCONTROLLER

The AVR 8-bits microcontroller provide both internal and external interrupt sources. The internal interrupts are associated with the microcontroller's peripherals. That is the Timer/Counter, Analog Comparator, etc. The external interrupts are triggered via external pins. The figure below shows the pins, on which the external interrupts can be triggered, for an AVR 8-bit microcontroller. On this microcontroller there are four (4) external interrupts:

1. **The RESET interrupt** - Triggered from pin 9.
2. **External Interrupt 0 (INT0)** - Triggered from pin 16.
3. **External Interrupt 1 (INT1)** - Triggered from pin 17.
4. **External Interrupt 2 (INT2)** - Triggered from pin 3.



While writing the code command sei() must be used as it activates the global interrupt.

UART AND USART

UART stands for Universal Asynchronous Receiver/Transmitter. From the name itself, it is clear that it is asynchronous i.e. the data bits are not synchronized with the clock pulses.

USART stands for Universal Synchronous Asynchronous Receiver/Transmitter. This is of the synchronous type, i.e. the data bits are synchronized with the clock pulses.

MODES OF OPERATION

The USART of the AVR can be operated in three modes, namely-

1. Asynchronous Normal Mode
2. Asynchronous Double Speed Mode
3. Synchronous Mode

1. ASYNCHRONOUS NORMAL MODE

In this mode of communication, the data is transmitted/received asynchronously, i.e. we do not need (and use) the clock pulses, as well as the XCK pin. The data is transferred at the BAUD rate we set in the UBBR register. This is similar to the UART operation.

2. ASYNCHRONOUS DOUBLE SPEED MODE

This is higher speed mode for asynchronous communication. In this mode also we set the baud rates and other initializations similar to Normal Mode. The difference is that data is transferred at double the baud we set in the UBBR Register.

3. SYNCHRONOUS MODE

This is the USART operation of AVR. When Synchronous Mode is used (UMSEL = 1 in UCSRC register), the XCK pin will be used as either clock input (Slave) or clock output (Master).

BAUD RATE GENERATION

The baud rate of UART/USART is set using the 16-bit wide UBRR register. The register is as follows:

Bit	15	14	13	12	11	10	9	8	
	URSEL	–	–	–	UBRR[11:8]				UBRRH
	UBRR[7:0]								UBRRL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
	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	
	0	0	0	0	0	0	0	0	

UBRR Register

Since AVR is an 8-bit microcontroller, every register should have a size of 8 bits. Hence, in this case, the 16-bit UBRR register is comprised of two 8-bit registers – UBRRH (high) and UBRRL (low). This is similar to the 16-bit ADC register (ADCH and ADCL). Since there can be only specific baud rate values, there can be specific values for UBRR, which when converted to binary will not exceed 12 bits. Hence there are only 12 bits reserved for UBRR[11:0].

The USART Baud Rate Register (UBRR) and the down-counter connected to it functions as a programmable prescaler or baud rate generator. The down-counter, running at system clock (FOSC), is loaded with the UBRR value each time the counter has counted down to zero or when the UBRRL Register is written. A clock is generated each time the counter reaches zero.

Below are the equations for calculating baud rate and UBRR value:

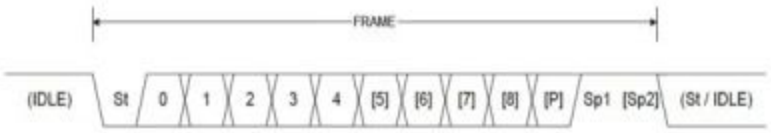
Operating Mode	Equation for Calculating Baud Rate[1]	Equation for Calculating UBRR Value
Asynchronous Normal mode (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{OSC}}{2(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

Baud Rate Calculation (Click to Enlarge)

1. **BAUD** = Baud Rate in Bits/Second (bps) (Always remember, Bps = Bytes/Second, whereas bps = Bits/Second)
2. **FOSC** = System Clock Frequency (1MHz) (or as per use in case of external oscillator)
3. **UBRR** = Contents of UBRRL and UBRRH registers

ORDER OF BITS

1. Start bit (Always low)
2. Data bits (LSB to MSB) (5-9 bits)
3. Parity bit (optional) (Can be odd or even)
4. Stop bit (1 or 2) (Always high)



St	Start bit, always low.
(n)	Data bits (0 to 8).
P	Parity bit. Can be odd or even.
Sp	Stop bit, always high.
IDLE	No transfers on the communication line (RxD or TxD). An IDLE line must be high.

PARITY BITS

Parity bits always seem to be a confusing part. Parity bits are the simplest methods of error detection. Parity is simply the number of '1' appearing in the binary form of a number. For example, '55' in decimal is 0b00110111, so the parity is 5, which is odd.

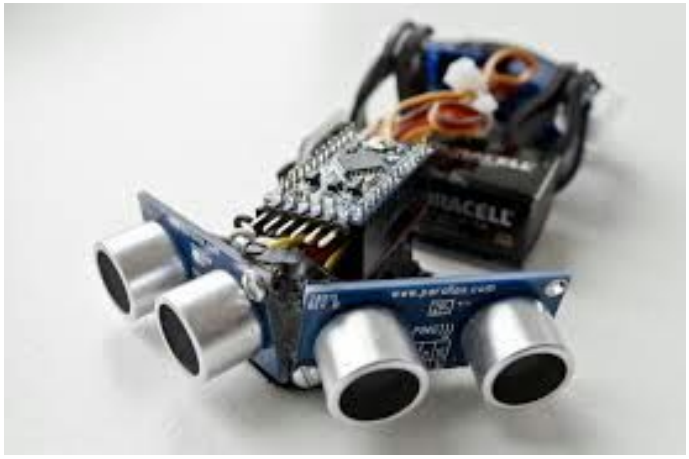
EVEN AND ODD PARITY

In case of even parity, the parity bit is set to 1, if the number of ones in a given set of bits (not including the parity bit) is odd, making the number of ones in the entire set of bits (including the parity bit) even. If the number of ones in a given set of bits is already even, it is set to a 0. When using odd parity, the parity bit is set to 1 if the number of ones in a given set of bits (not including the parity bit) is even, making the number of ones in the entire set of bits (including the parity bit) odd. When the number of set bits is odd, then the odd parity bit is set to 0.

EXAMPLE: 0b10100111. This has five 1s in it. So in case of even parity, we add another 1 to it to make the count rise to six (which is even). In case of odd parity, we simply add a 0 which will stall the count to five (which is odd). This extra bit added is called the parity bit.

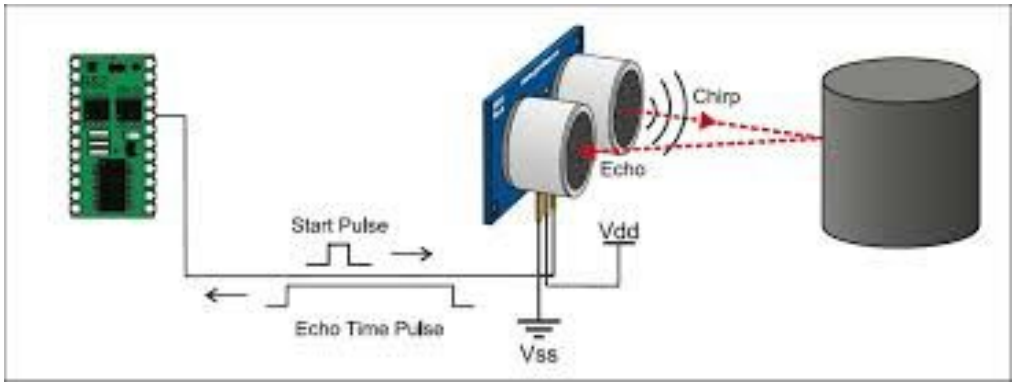
DAY 5

SONAR



It is an acronym that stands for Sound Navigation And Ranging .This instrument make use of sound waves for detecting and locating the objects underwater likes rocks, submarines ,icebergs and to find the depth of water bodies .

PRINCIPLE



This instrument is based on a very simple principle i.e a pulse of ultrasonic waves is sent into the water there it strikes the target and get bounced back towards the source .

The exact location of underneath object or the distance of it from the source can be calculated by measuring the time it takes for the sound to return back to the source after striking the target . We know the speed of sound in water and easily calculate the distance by simple speed formula .But keep in mind that compute the distance by multiplying the speed by one-half of the time traveled (i.e just take one one-way trip). This is what we call active sonar ranging (or more common echolocation).

ARDUINO PROGRAMS

FOR LED BLINKING

```
void setup()
{
  pinMode(13,OUTPUT);
}
void loop()
{
 digitalWrite(13,HIGH);
delay(200);
digitalWrite(13,LOW);
delay(200);
}
```

FOR PWM

```
void setup()
{
 pinMode(13,OUTPUT);
}

int v=0;
int j=1;

void loop()
{
analogWrite(13,v);
v=v+j;

if(v==0||v>=255)
j=j*(-1);
}
```