Electronic Basics

-Great Scott

1.The Multimeter:

A multimeter is capable of measuring all the important parameters.There are two probes(Black and red).The black probe is always connected in the common socket.The red socket will only be changed when measuring current.
When we measure the resistance of a resistor in a circuit,it may not give exact result because current finds simplest path when it travels.
When measuring voltage,in the case of DC we have to connect the red probe to the positive side and right probe to the negative side.
When measuring current,change the red probe to 10amp socket always to save the multimeter.Connect the probe to the opening point of the circuit so that the current can flown through the multimeter.
If the fuse is gone,unscrew the multimeter and change the fuse at the bottom right corner.

3.Programming an Attiny+Homemade Arduino shield

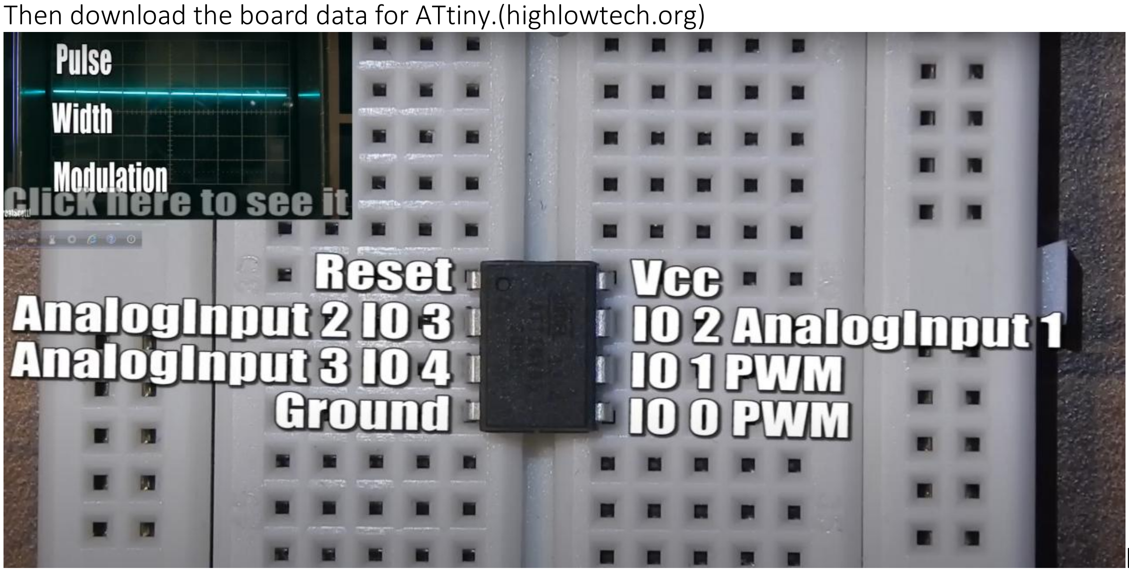
The microcontroller is ATtiny 85.

5 IOs.

8kbytes flash memory.

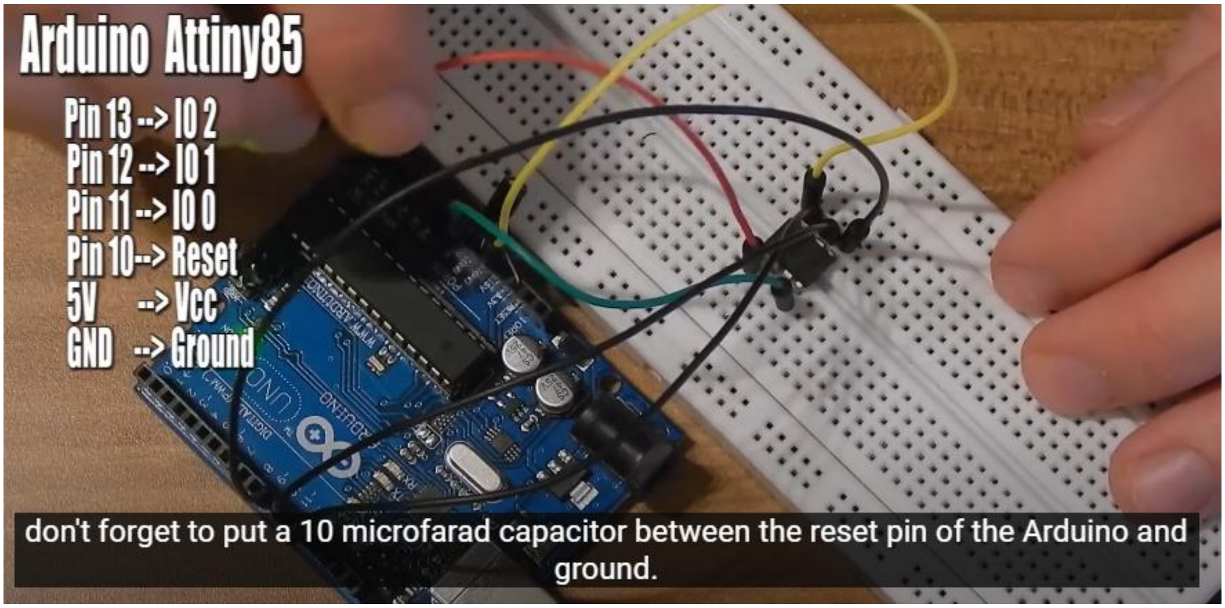
Arduino 1.0.5.

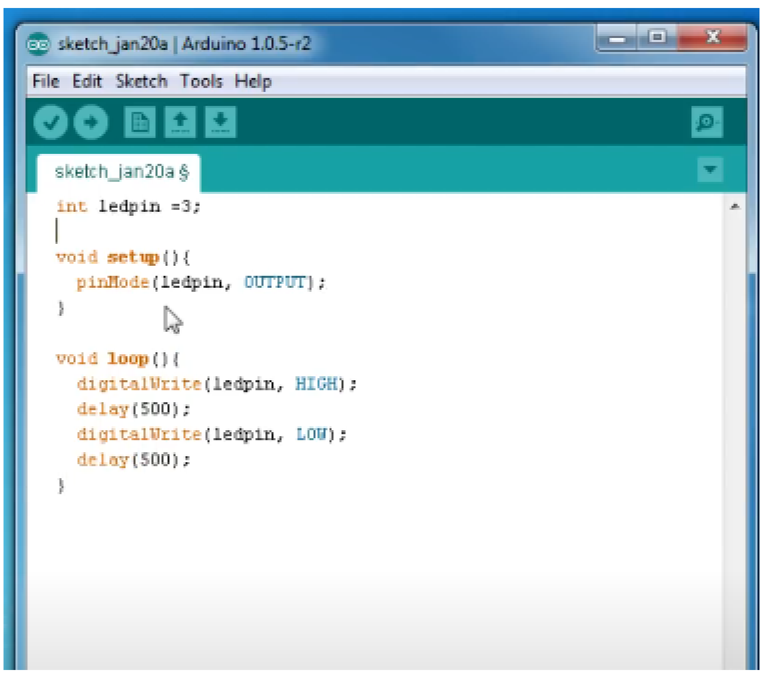
Then download the board data for ATtiny.(highlowtech.org)

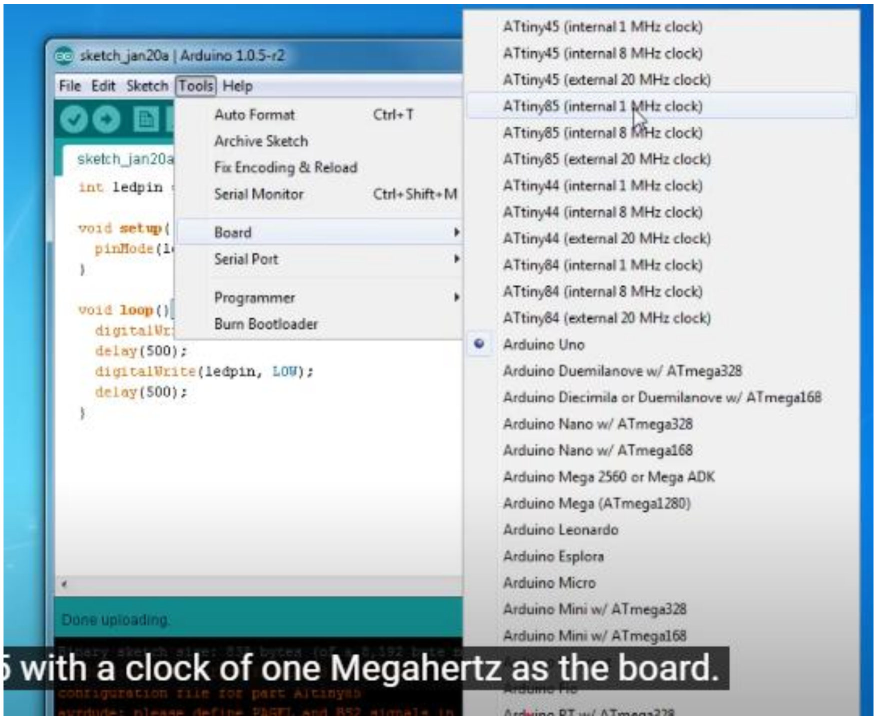


he

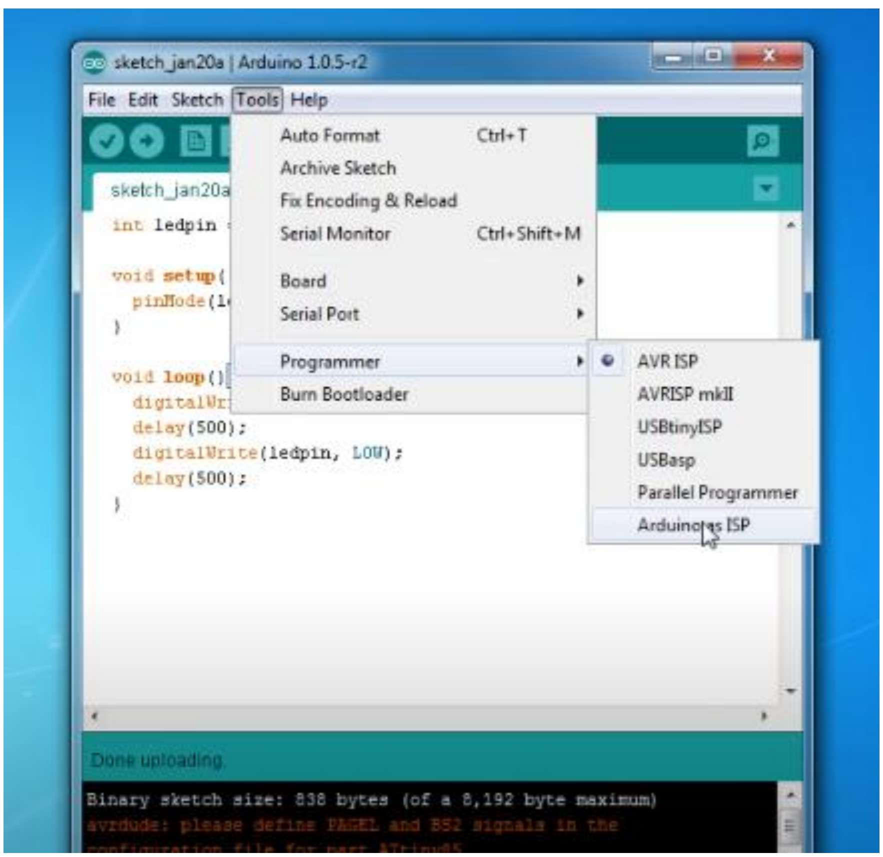
Don’t forget toput a 10 microfarade capacitor between the reset pin of Arduino and ground.







sketch_jan20a | Arduino 1.0.5-r2
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader
ATtiny45 (internal 1 MHz clock)
ATtiny45 (internal 8 MHz clock)
ATtiny45 (external 20 MHz clock)
ATtiny85 (internal 1 MHz clock)
ATtiny85 (internal 8 MHz clock)
ATtiny85 (external 20 MHz clock)
ATtiny44 (internal 1 MHz clock)
ATtiny44 (internal 8 MHz clock)
ATtiny44 (external 20 MHz clock)
ATtiny84 (internal 1 MHz clock)
ATtiny84 (internal 8 MHz clock)
ATtiny84 (external 20 MHz clock)
Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Esplora
Arduino Micro
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
delay(500);
digitalWrite(ledpin, LOW);
delay(500);
Done uploading.
5 with a clock of one Megahertz as the board.



sketch_jan20a | Arduino 1.0.5-r2
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader
AVR ISP
AVRISP mkII
USBtinyISP
USBasp
Parallel Programmer
Arduino as ISP
int ledpin
void setup(
void loop()
delay(500);
digitalWrite(ledpin, LOW);
delay(500);
Done uploading.
Binary sketch size: 838 bytes (of a 8,192 byte maximum)
avrdude: please define PAGEL and BS2 signals in the

4.Arduino+Bluetooth+Android

TX-Transfer pin

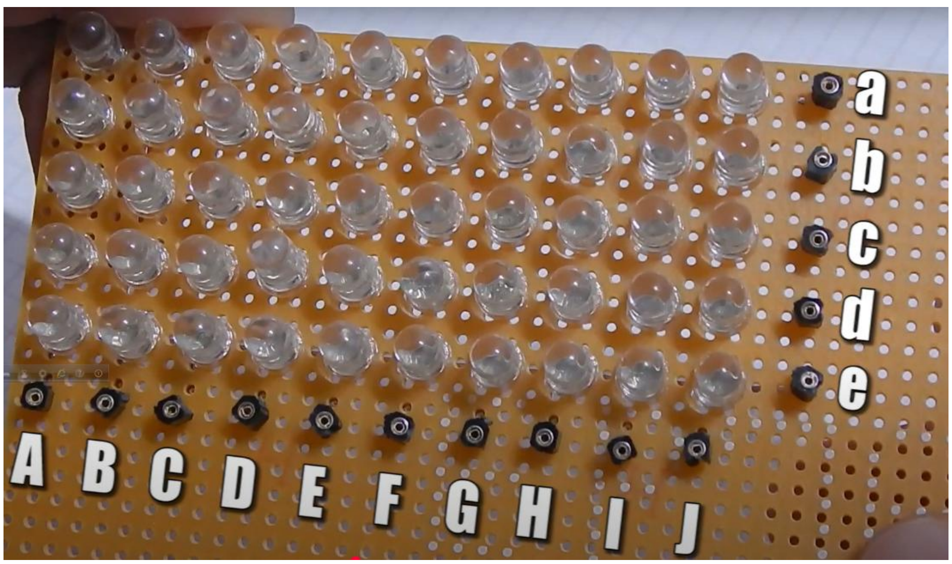
RX-Receive pin

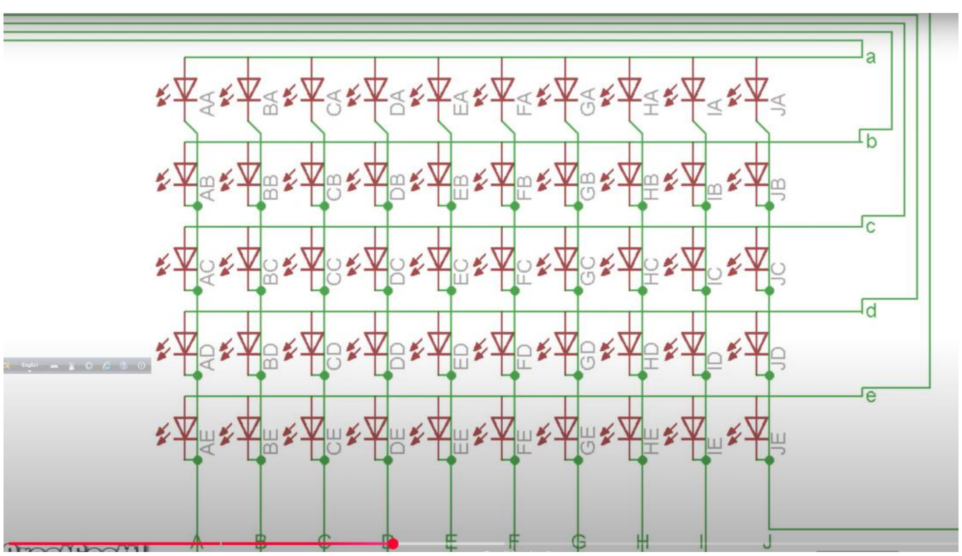
-Coding with S2 terminal app

The differences in voltage levels between Bluetooth modules and Arduino boards are explained.

5.How to multiplex

How to control many LEDs with only a few iOS.
Uses 1 arduino nano,1 TLC5940 LED driver,5 p-channel MOSFETS F9540N,1 2k ohm resistor,5 1k ohm resistor.
Connect all the LEDs in PCB and cconnect all the cathods of a column by soldering.
Connect all the anodes in a row.





Connect positive voltage to ‘a’ and ground to A and LED Aa lights up.

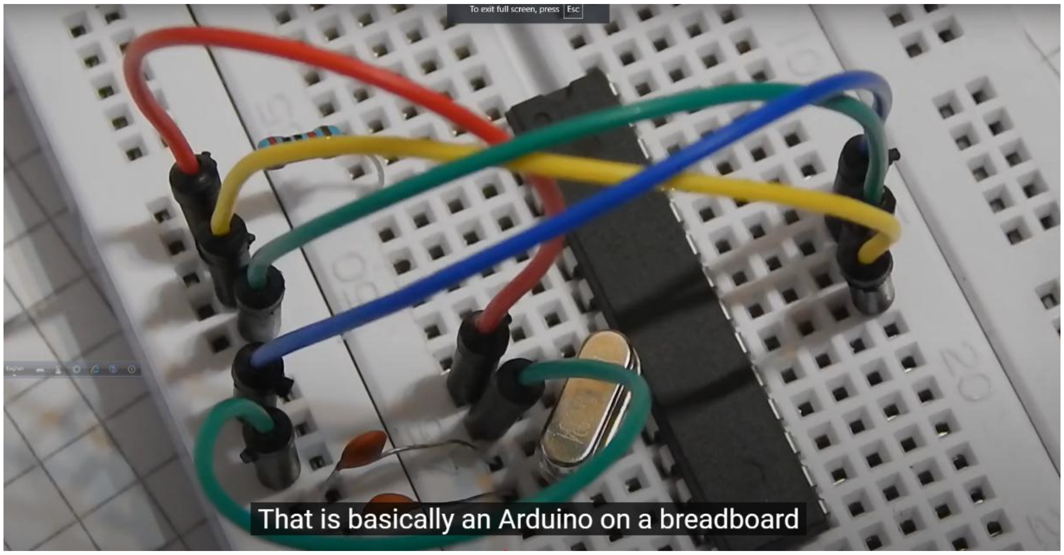
We light up each row individually and one offer the other but we go through the rown so fast that our eyes don’t notice and just see a static complete picture.

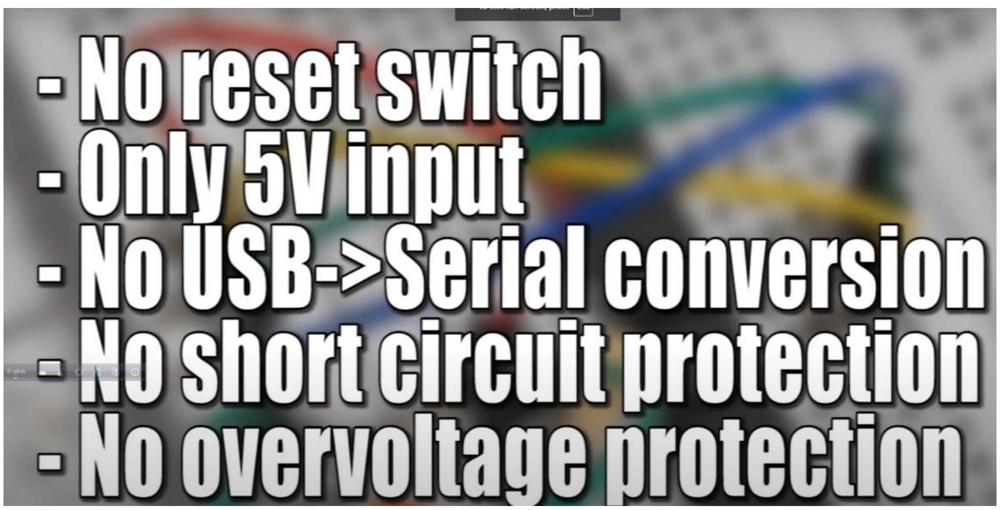
One row can draw up to 200mA with 10 LEDs so wee cannot connect those to one i/o.We have to use p-channel MOSFET as switch and connect gate to my i/o source to 5V and drain to the anode rows and I use 10 outputs of my LED driver.
Gate of MOSFET row ‘a’ to nano pin 8.
MOSFET row ‘a’ to nano pin 8.
MOSFET row ‘b’ to nano pin 7.
TLC pin 28 to column 1.
TLC pin 1 to column 2.
TLC pin 2 to column 3 and so on.
We are going to use TLC library from the Arduino playground.

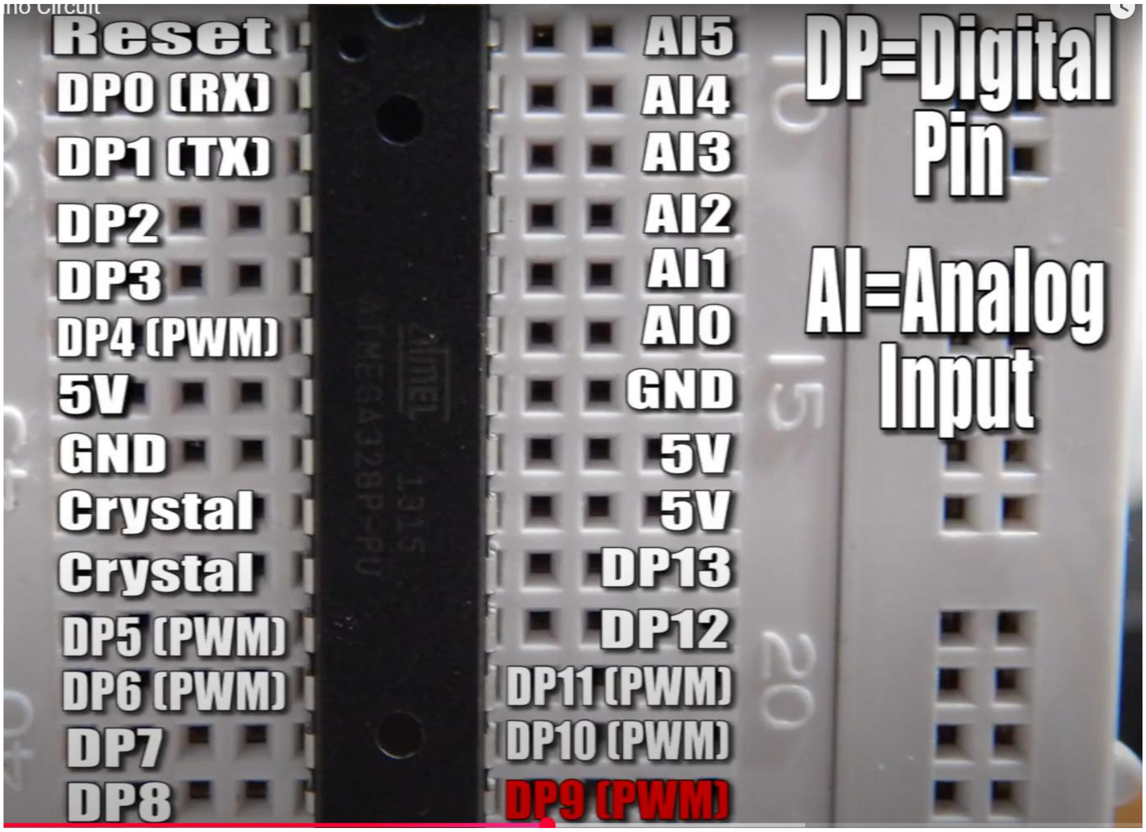
6.Standalone Arduino circuit:

Here we see how to break ATmega328p microcontroller free from its Arduino prison and embed it into a circuit.
First always test your circuit on a breadboard.
Components-1 16MHz clock crystal,2 22pF capacitors to generate the external clock signal and 1 10kohm resistor which connects between the reset pin of the ATmega and 5 volts.This way on microcontroller does not reset itself.
The crystal connects to pin 9 and 10 and 1 capacitor between each pin and ground.If you only need a clock of 8 MHz you could also scrape those parts and use internal oscillator but you would need to upload another bootloader.Check out the Arduino site for more information about that.

Pin 7,20 and 21 connects to 5 volts and pin 8 and 22 to ground.


To exit full screen, press Esc
That is basically an Arduino on a breadboard


- No reset switch
- Only 5V input
- No USB->Serial conversion
- No short circuit protection
- No overvoltage protection



Now I want to change my code.There is 3 ways:

1.You just get your ATmega out of the breadboard and plug it in the Arduino and reprogram it.
2.Connect Tx of the Arduino board to pin 3,Rx to pin 2 and reset to pin 1.

##Atmel studio 6.

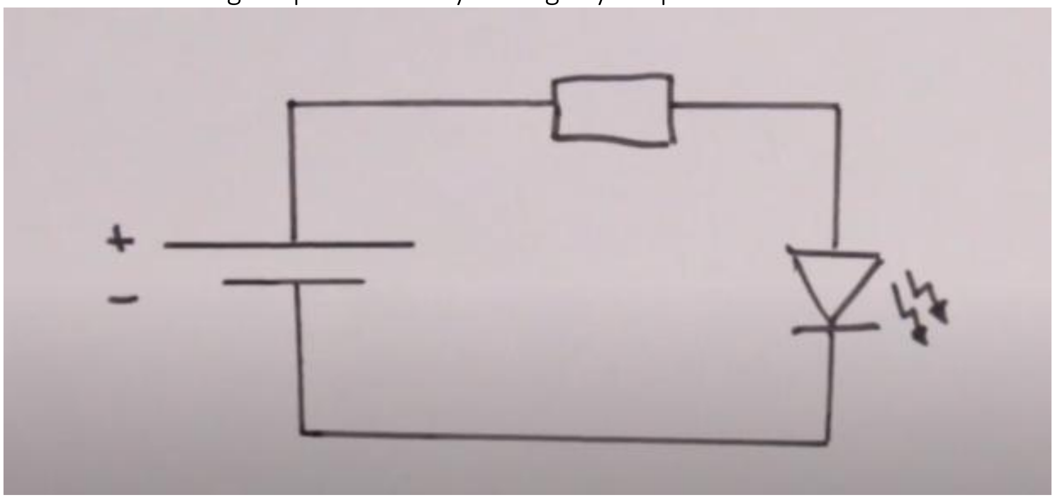
7.7 Segment Display

Always check the displays datasheet to get the pinouts.In this case,we have to download the datasheet.The display is just a combination of LEDs and the pins control different LEDs.
How can we display numbers using the microcontrollers?
The easy solution is the BCD to seven-segment display driver.
We are using SN7LS247 which has a active low level which means it is capable of controlling all individual cathodes of our display.If you would use a common cathode display you would need an active high level.
We have to feed the IC four inputs which are ABCD to let it know which number should light up.We have look at the function list in this regard.
A-low
B,C-high
D-low
It will represent number 6.
This displays have 16 LEDs inside and two common anode to activate either the left or the right digit.Ofcourse we cannot control all 42 LEDs of those two displays with one Arduino.

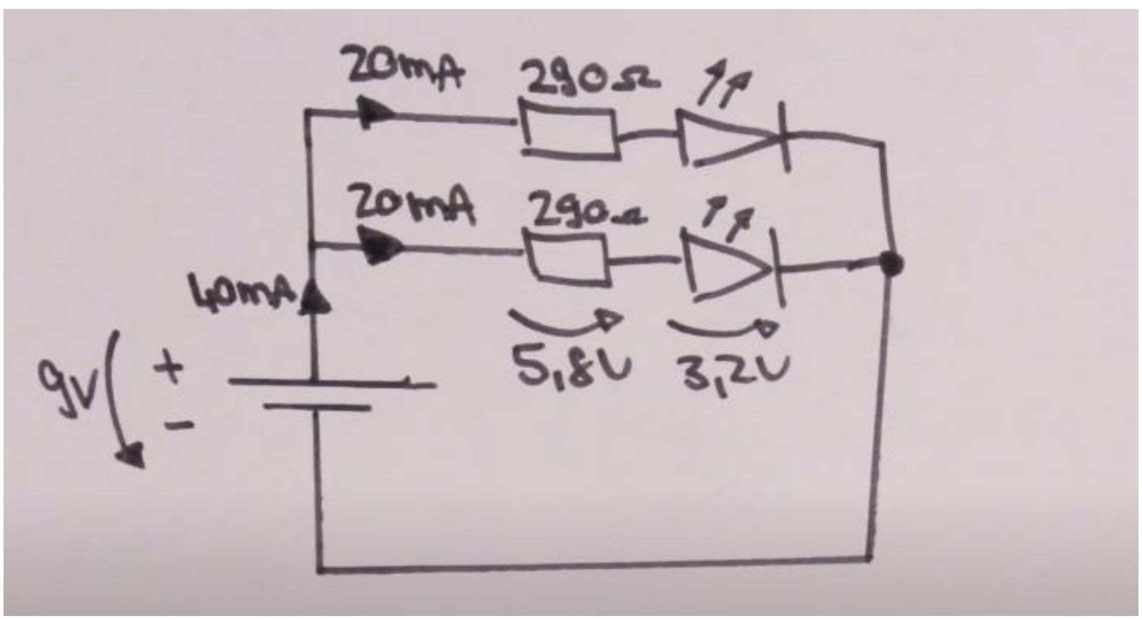
So we have to solve the problem by multiplexing.
##Tonic staff tutorial

8.Everything about LEDs and current limiting resistors:

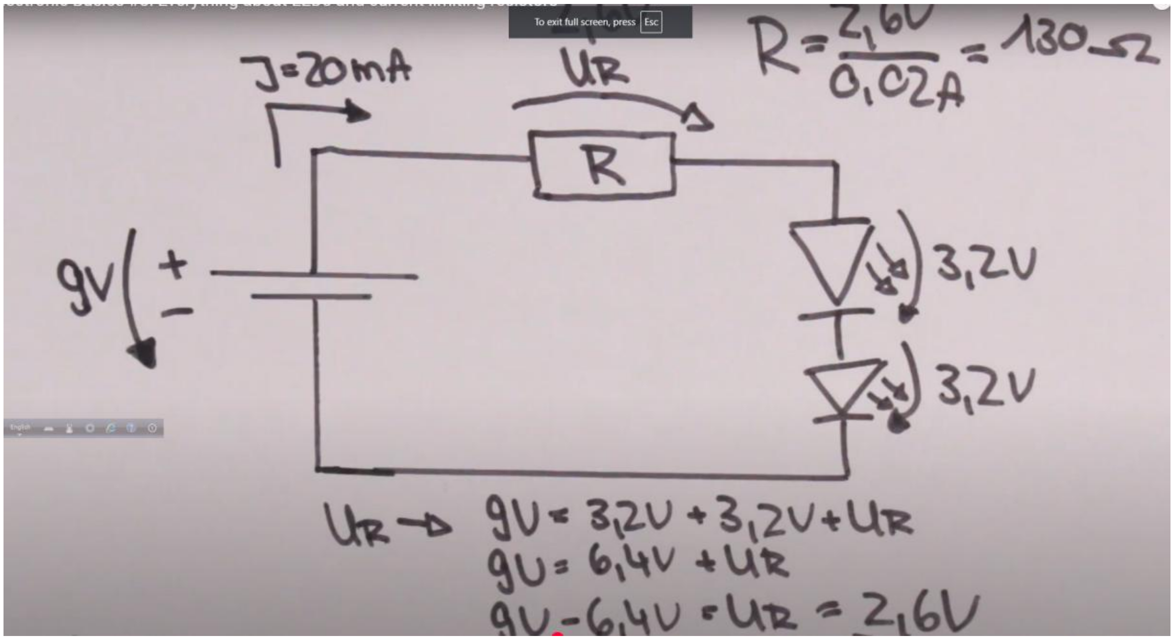
LEDs have no data sheets but they have two important parameters-The forward voltage and the current need to light up ideal.Now you've got your power.



If you not use a resistor,the LED would die fast.We have to drop extra voltage of the voltage source across a resistor.

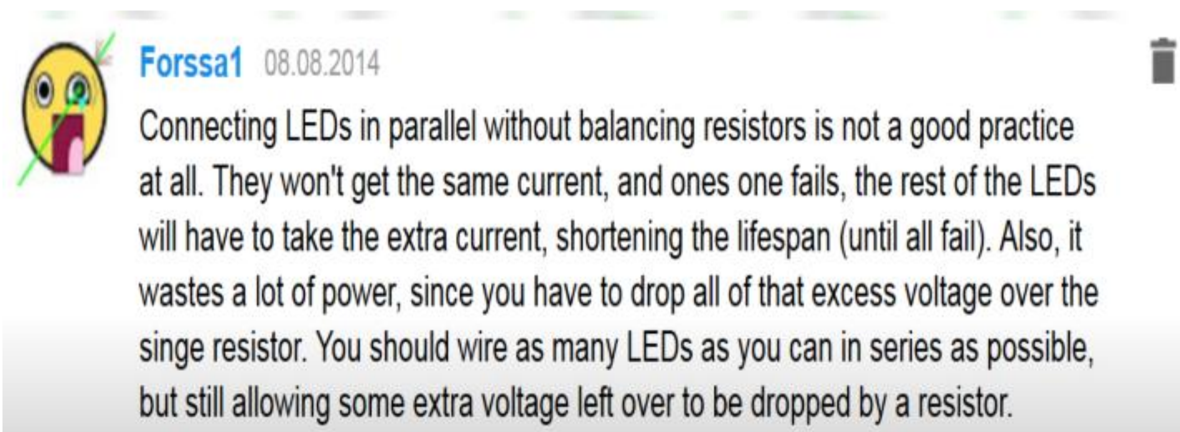


If you want to light up two LEDs,you can make the connections like this.But this is waste off power.

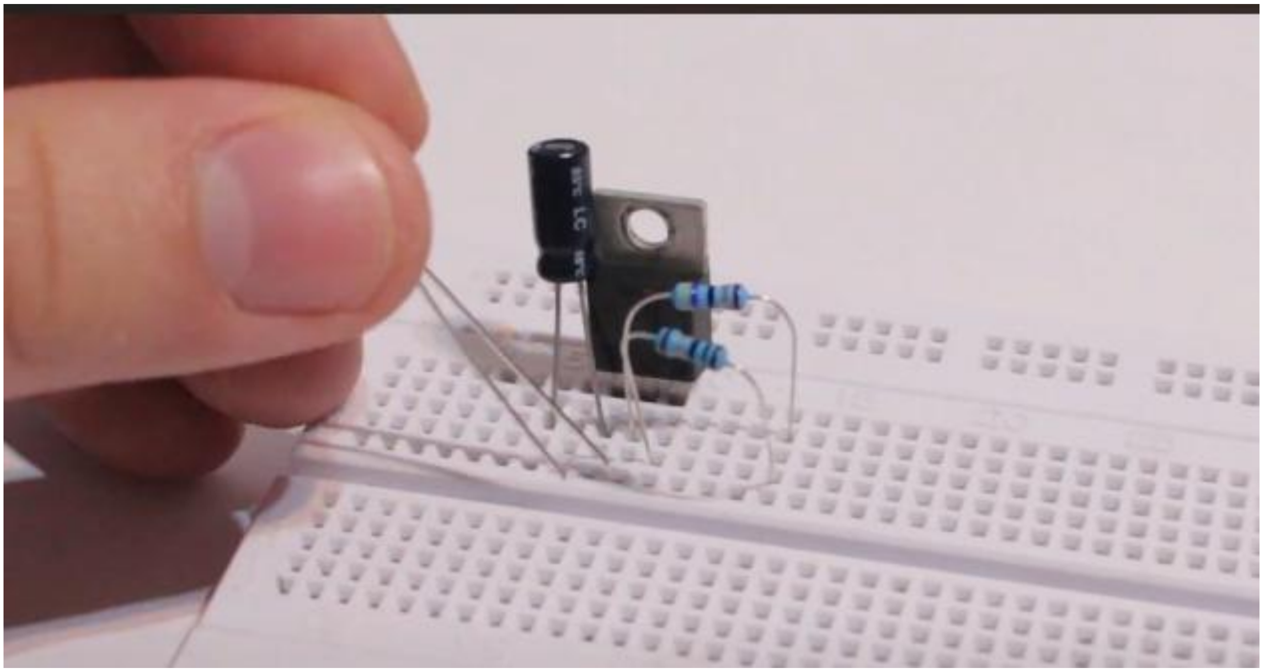


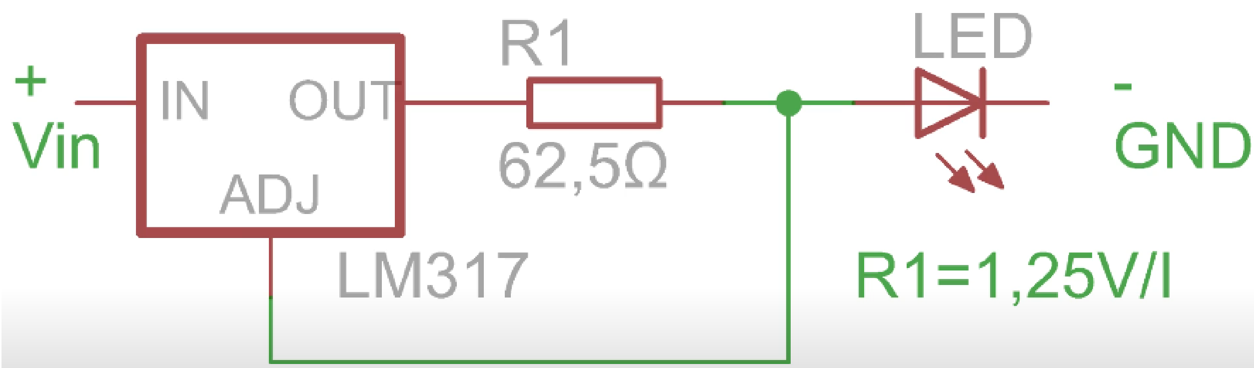
Just connect two LEDs in series and follow the same calculation process.

If you change in your power source it can destroy all your LEDs.Incase try using a small resistor to linearize the current consumption.This way voltage change do not affect the LEDs that much.



Connecting many LEDs in parallel,it is not possible to use one bigger resistor because all of those want a different voltage level.
The best way to drive LEDs not in a constant voltage mode but in a constant current mode.You can create a simple constant current source using a **LM317.**

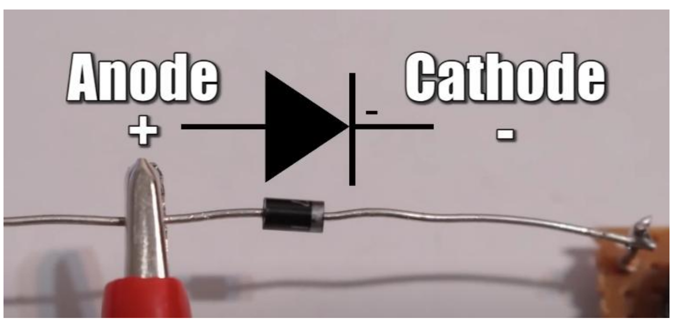




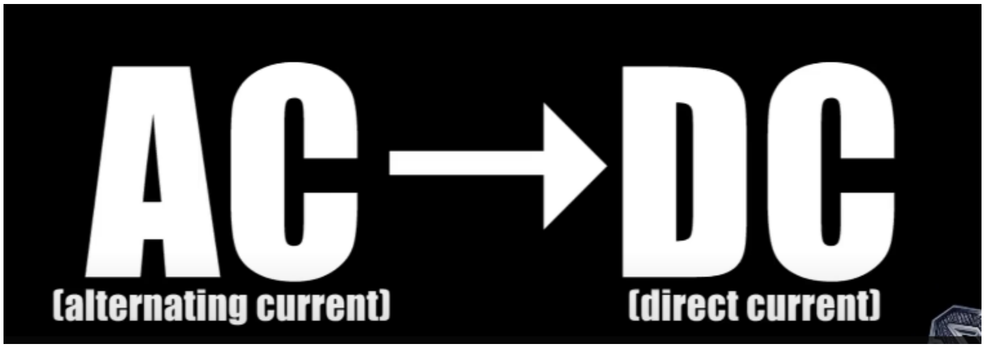
The TLC5940 is also a popular example of a constant current driver.

9.Diodes and Bridge Rectifiers:

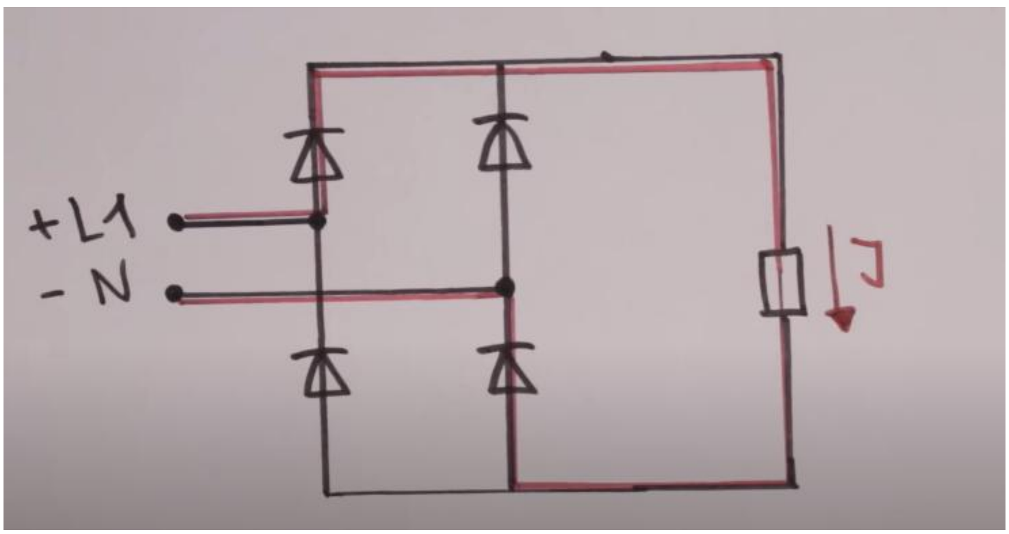
If we connect the power supply in reverse polarity which will damage the circuit,a diode would save the circuit in this case which prevents the current to flow in reverse direction.

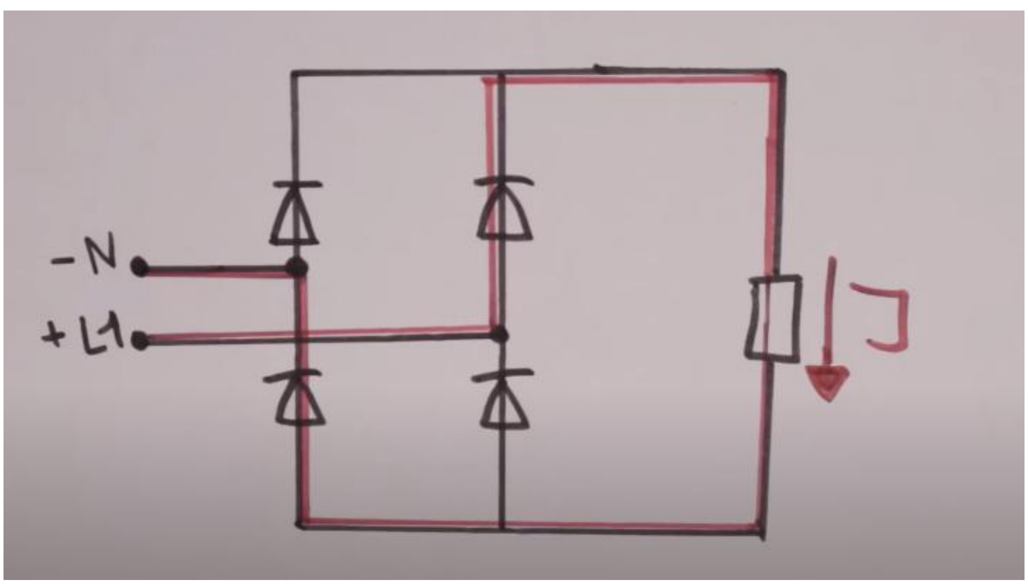


No diode is perfect.There is always a small voltage drop across the diode.



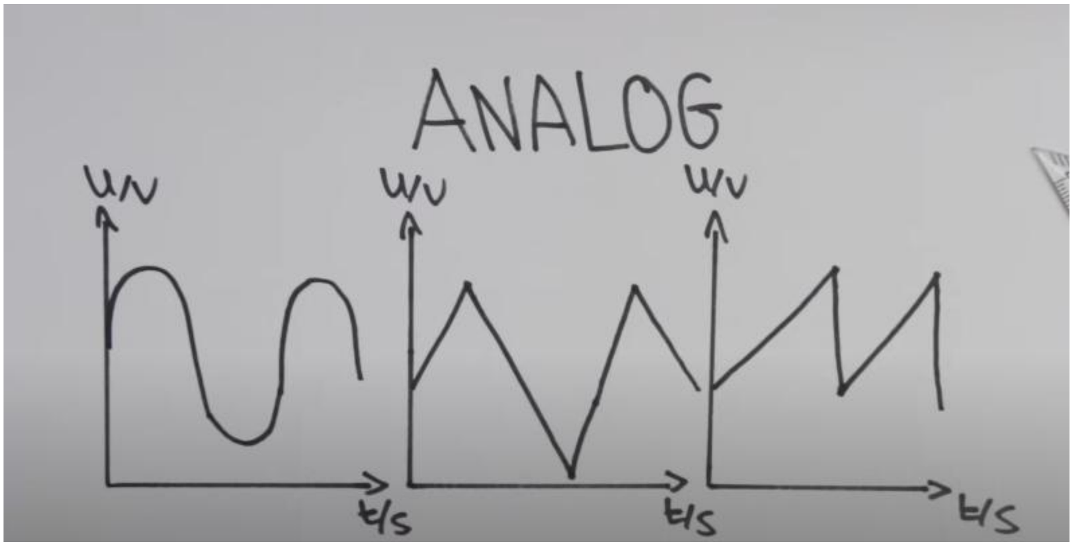
We can use Capacitor in series with a diode to convert the pulsating DC into constant DC.



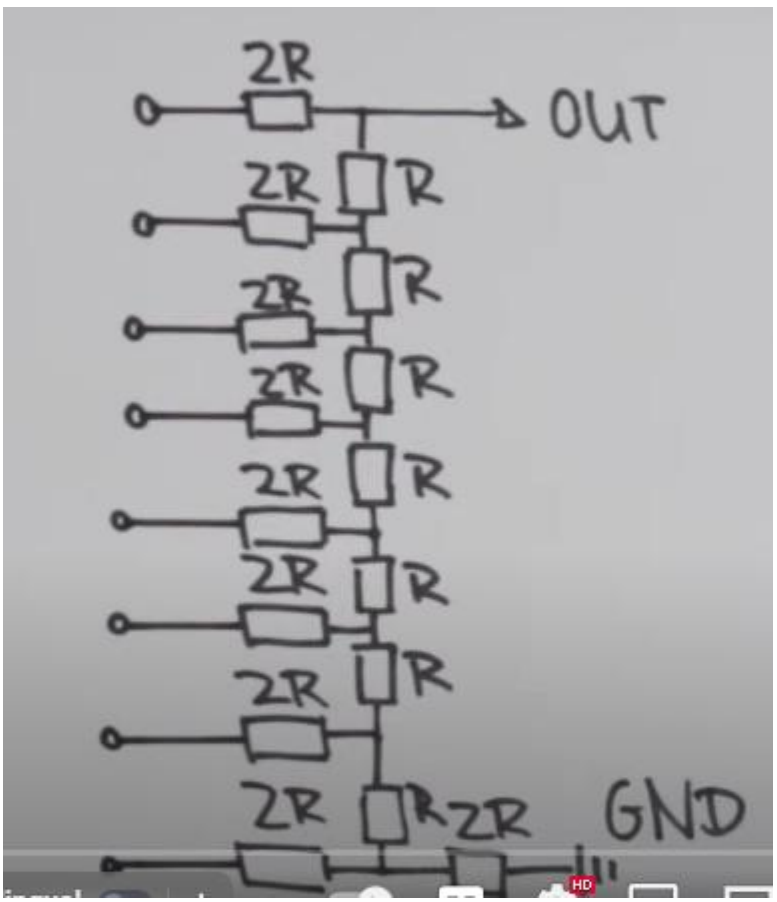


10.Digital to Analog Converter:

An analog signal can be sign wave or a triangle wave or ramp function

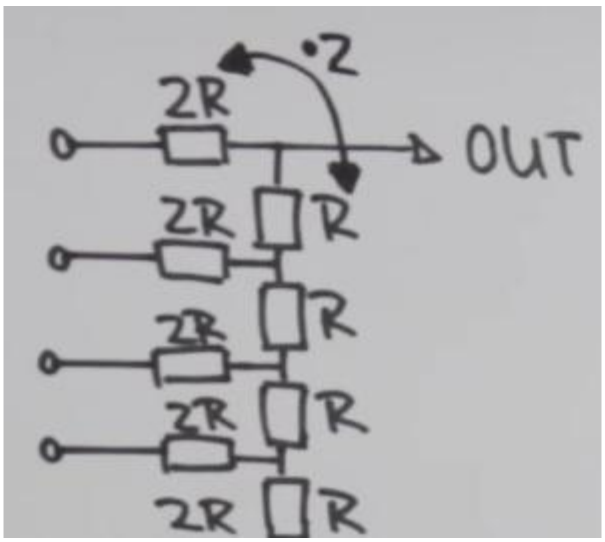


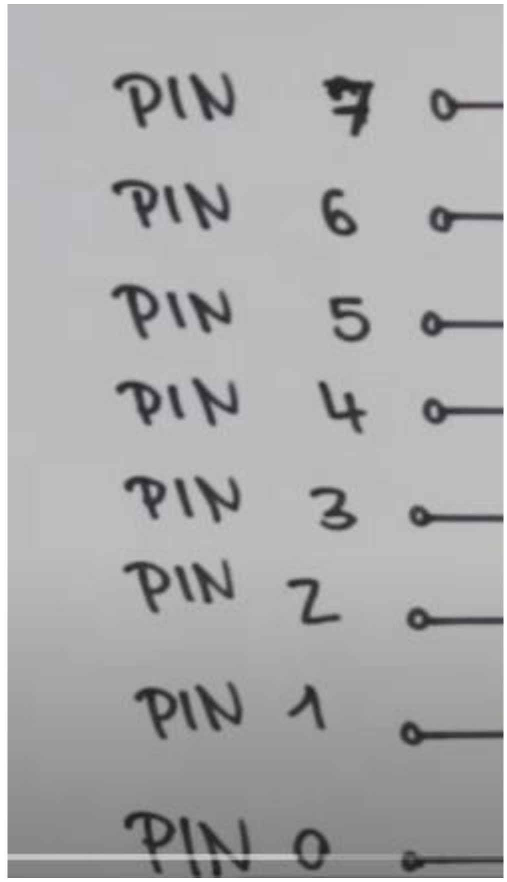
There are many techniques on how to convert digital to analog.
Now just focus on the **R-2R resistor ladder method.**



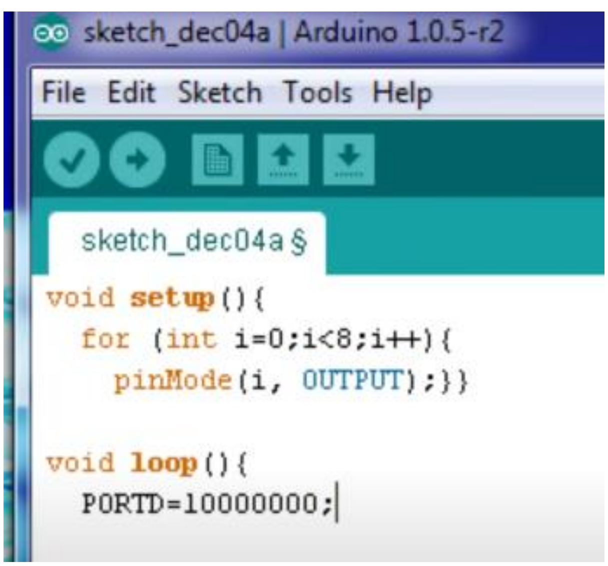
What does 8-bit mean?

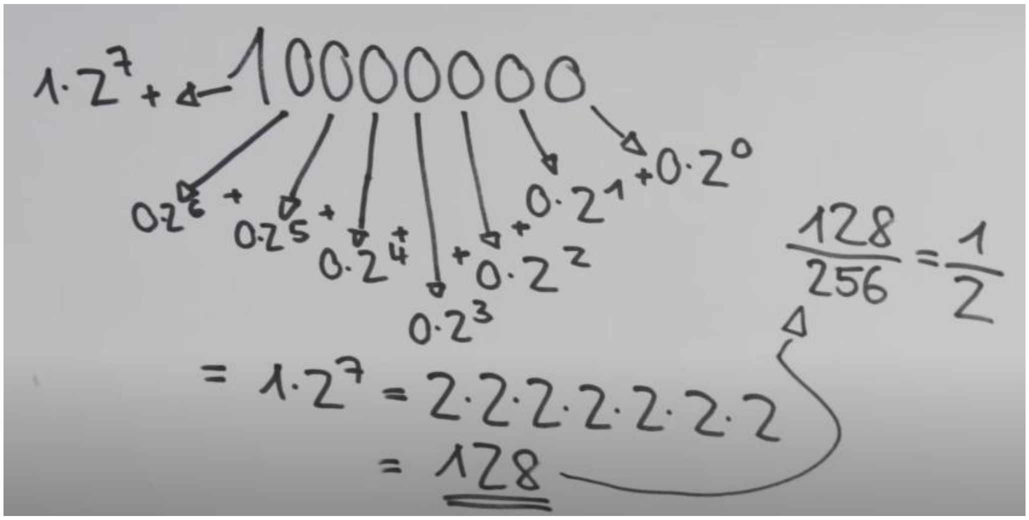
-That means we have 256 voltage values between 4.8 volts and 0 volts.We could also call this Resolution of DAC.





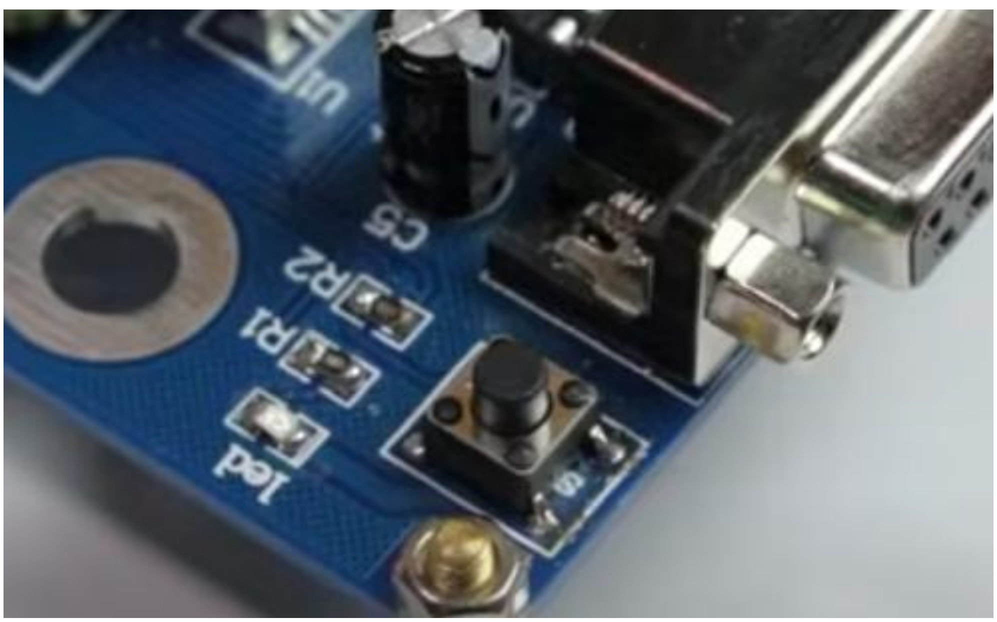
The left side inputs are the digital pins of Arduino nano.





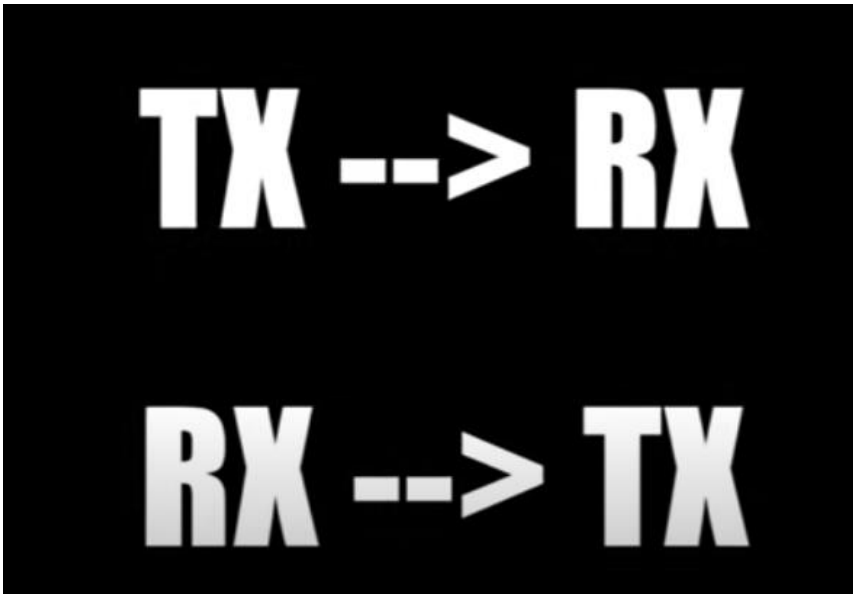
11.Sending SMS with Arduino || TC 35 GSM Module

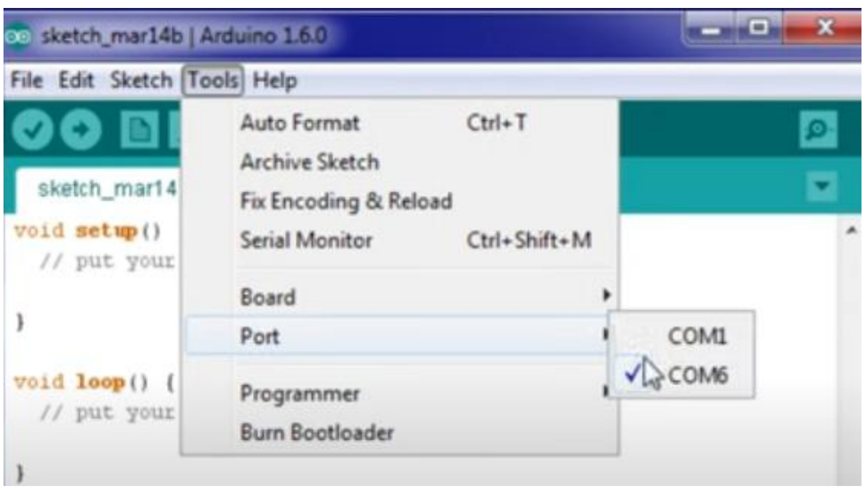
You’ve to remove MAX 232 before powering it.
The board draws 6mA in standby mode but it’s not ready for action yet.We have a button which starts the login process to the mobile network.When the status LED links shortly every 2 to 3 seconds then you know that the module connected successfully.

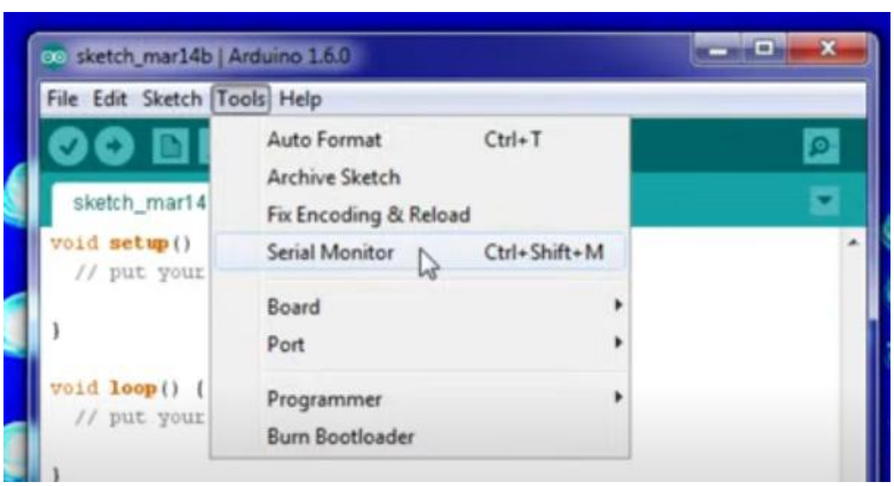


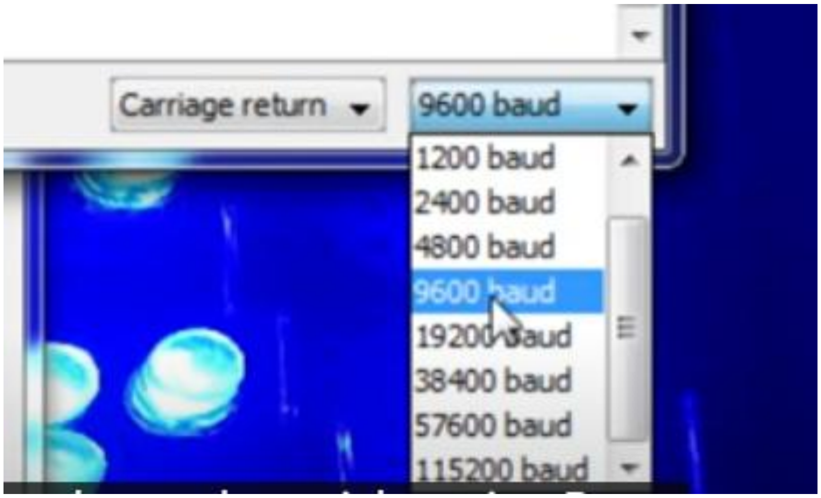
But this button itself is not practical if you want to automate something using a microcontroller. Left side of the switch is ground and the right side of the switch is connected to the module as an input.Whenever the right side connects to ground,the login process begins.I solved this soldering a jumper wire to the right side of the switch which connects to the pin 10 of the arduino UNO. Whenever the pin goes low,the login process starts the board then draws around 13mA while being in this state.

I connect Tx to txd0 and Rx to rxd0.This is backward.

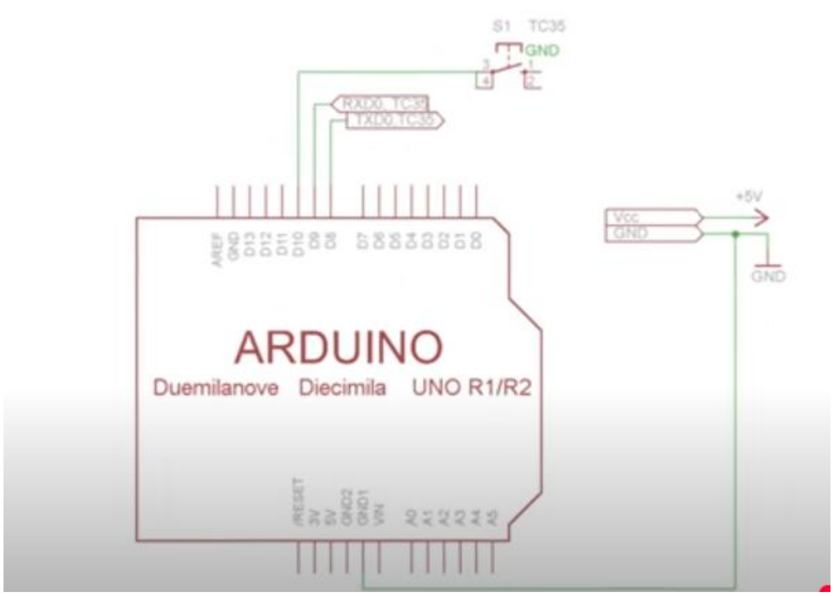






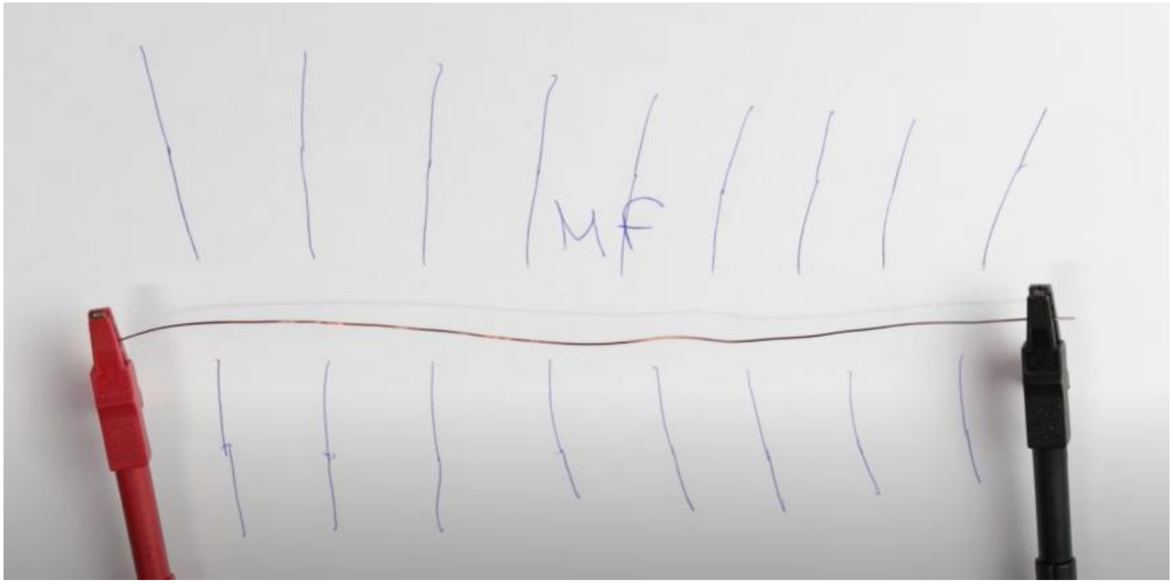


The module uses simple AT commands to communicate.All of those are listed in the AT command reference.

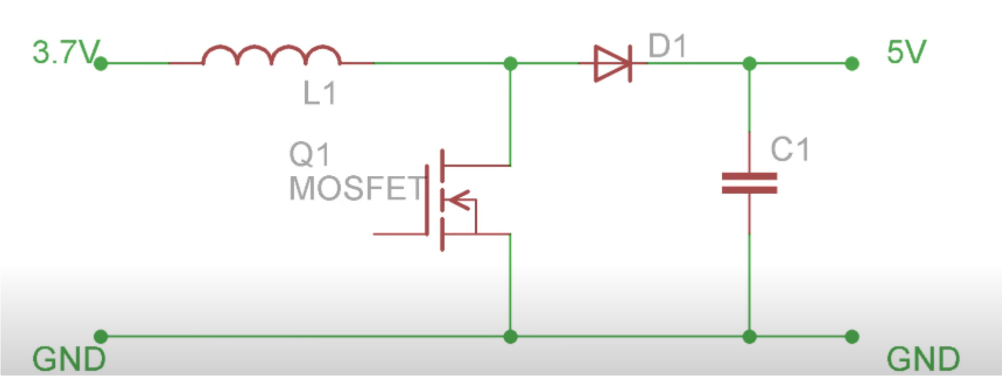


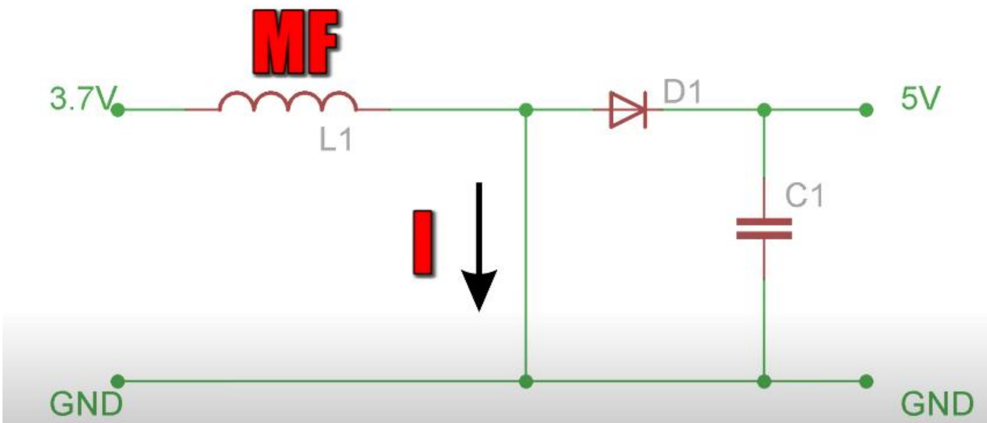
12.Coils/Inductors(Part 1)

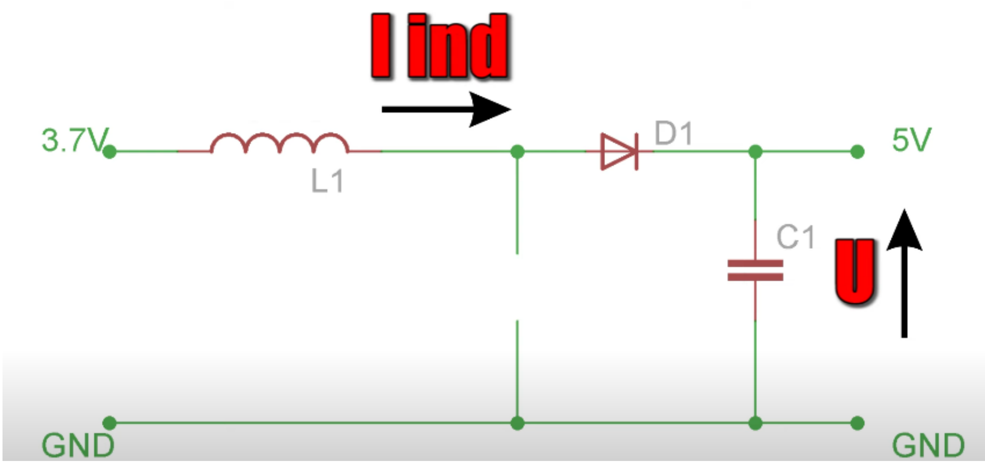
Motor,Transformer,Relay all consist of coils.

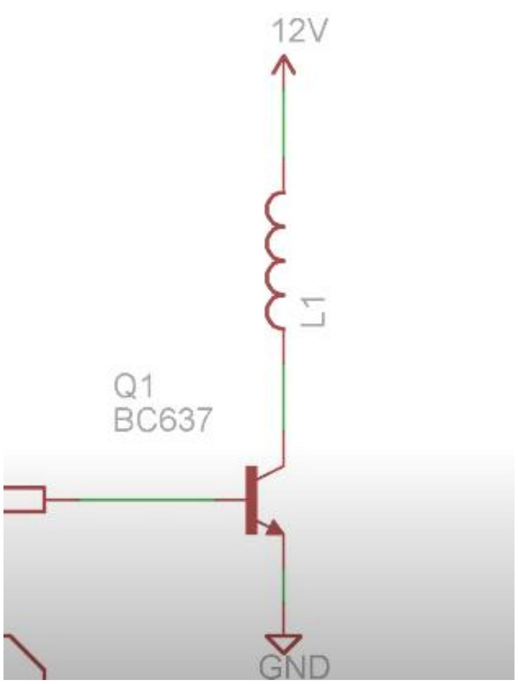


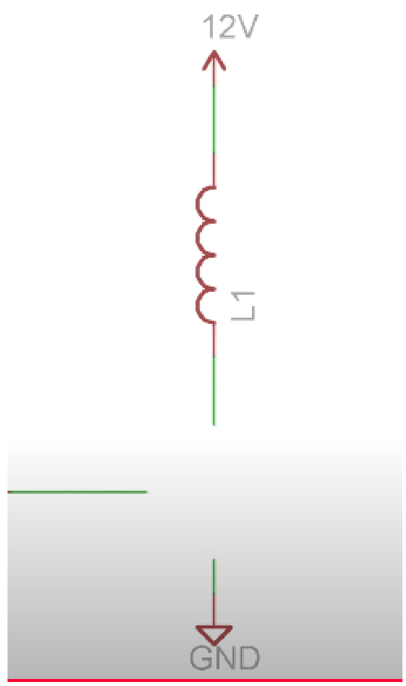
More current,bigger magnetic fields.
The magnetic field of plain wire is very weak.That's why we can wind up our wire to create a bigger length which increases our magnetic force.
We can use a ferromagnetic material as a core to enhance the fields.
Relays use such a trick to trigger a switch that can handle high AC currents.
Current through the coil does not follow the input voltage wave immediately and takes a bit of time to reach its final state.
With more inductance this can take longer and less inductance of our coil it can take less time.
The reason is called Lenz law.
We want current to flow.So our coil will hinder it as good as it can or we want currents to stop flowing,so out coil will use the energy of its magnetic field to pump current into the circuit again and it hinders again as good as it can.Seems like coils are plain annoying,but we can use them to our advantage.We use the energy stored in the inductor in Boost converters to get 5 volts out of 3.7 volt batteries.



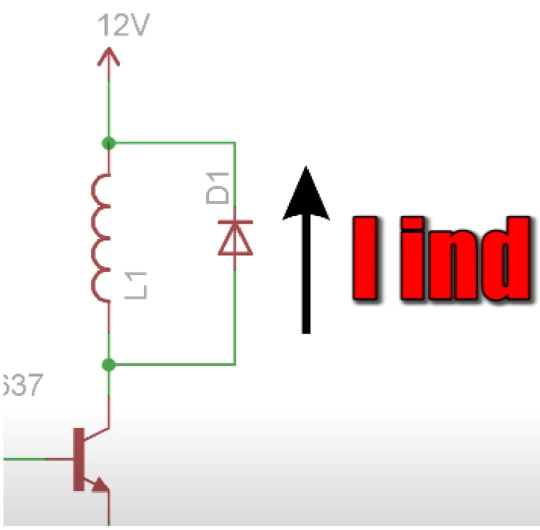

MF
3.7V
L1
D1
5V
C1
I
GND
GND


I ind
3.7V
L1
D1
5V
C1
U
GND
GND


12V
L1
Q1
BC637
GND

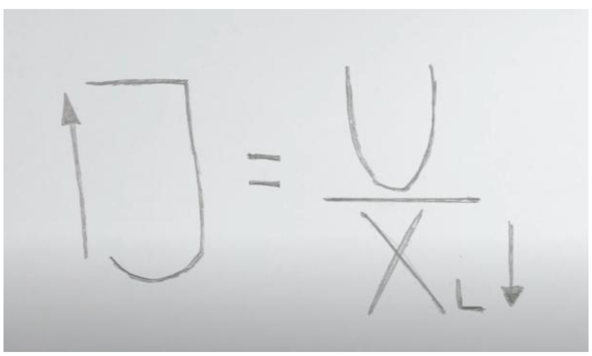

12V
L1
GND

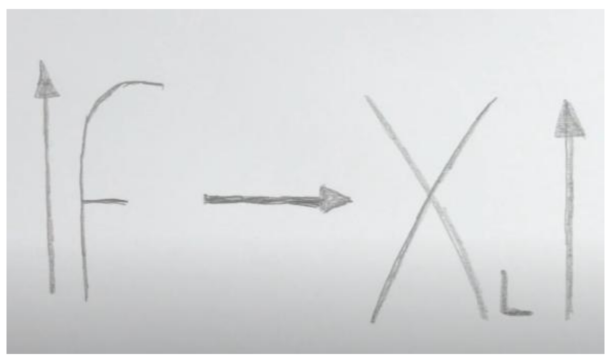
We have a problem when our transistor switch closes.The coil tries to push the energy inside the open circuit which produces a huge electron access and thus peak voltage spikes which can destroy our transistor.So we use a **flyback diode** to protect our switch by offering the current a way to flow.



13.Coils/Inductors(Part 2) || Reactance

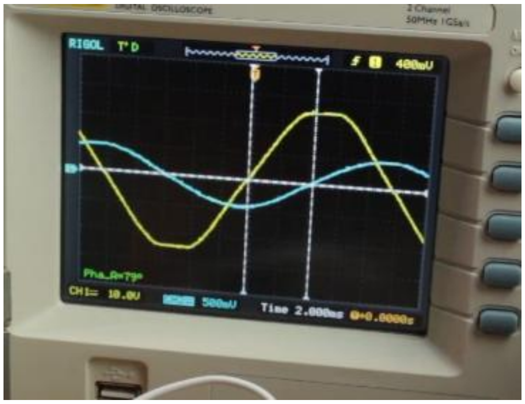
A high magnitude of current flows through an LED and the LED dies a blink of eye.When we connect a coil of 1H and 34.5 ohm resistance in series,the LED works properly.But if we use resistor of 34 ohm,the LED still gets destroyed.
In inductors,power can't be converted into heat like resistors.But in a magnetic field which builds up and collapses.

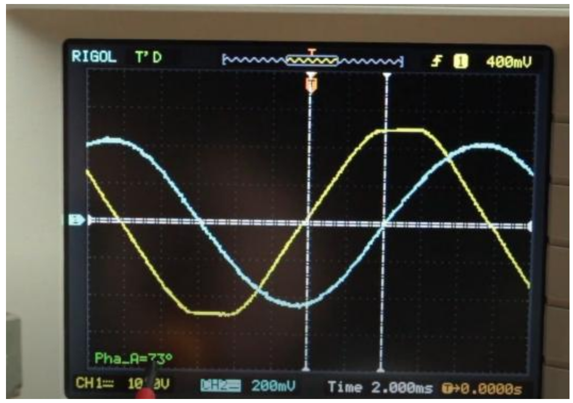




As this is a variable resistor based on the frequency we can build **Noise Filters** or **Music Frequency** Filters quite easily.

I hooked my 1H coil with a 10 ohm resistor to my 50V rms transformer.If we take a look at the voltage and current form e can clearly see that they are not lines up perfectly like it would be with a simple resistor circuit.This is called a **phase shift.**



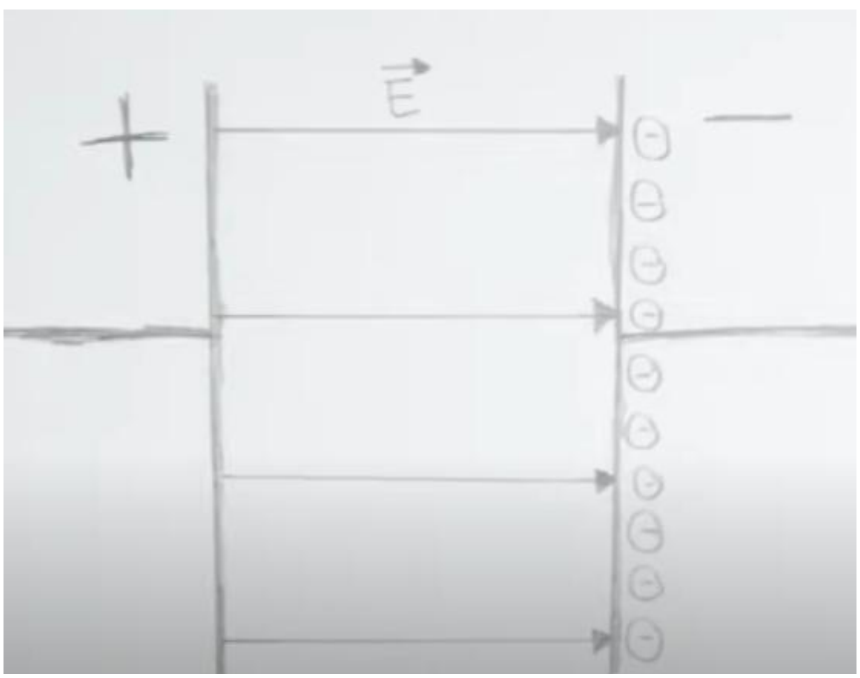


The phase shift can reach up to 90 degrees with an ideal inductive load.The phase changes with the change in inductance.

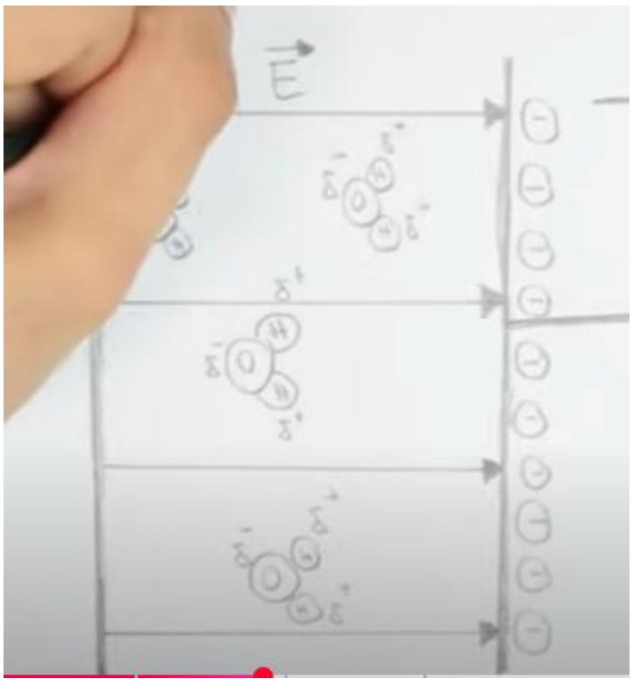
14.Capacitors:

If you ever had the problem that your monitor or TV just stops working at some points then there's a big chance that you can repair it by replacing the dodgy capacitors on the circuit board and when you think about it almost every circuit of consumer electronics has one capacitor of one form or other.

Take an alluminium sheet and draw a line at the middle of the material.Another line at the 1/3rd of the length at the other side.Then cut the pieces.Place the pieces parallel to each other in a very small distance.

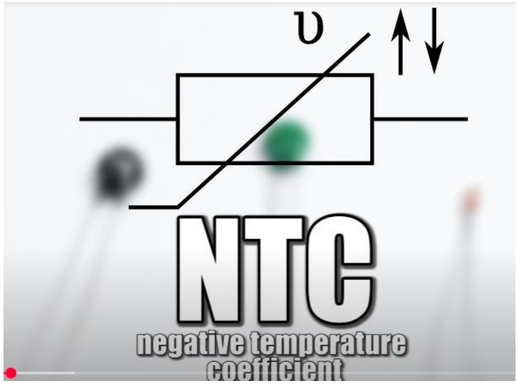


If we connect the capacitor,we can see that just for a fraction of a second a very small amount of current is flowing.This currents charged up the plates by creating an electron excess on the negative side and thus an electrostatic force in created between the plates.
Normally they store energy and can supply the stored energy from the electrostatic field just like a battery can supply power.
We can increase the area of the alluminium sheets to store more electrons.
We can keep dielectric materials between the sheets to increase the capacitance.Dipoles line with the electrostatic fields and increase the force on the electrons which again creates more space for others.

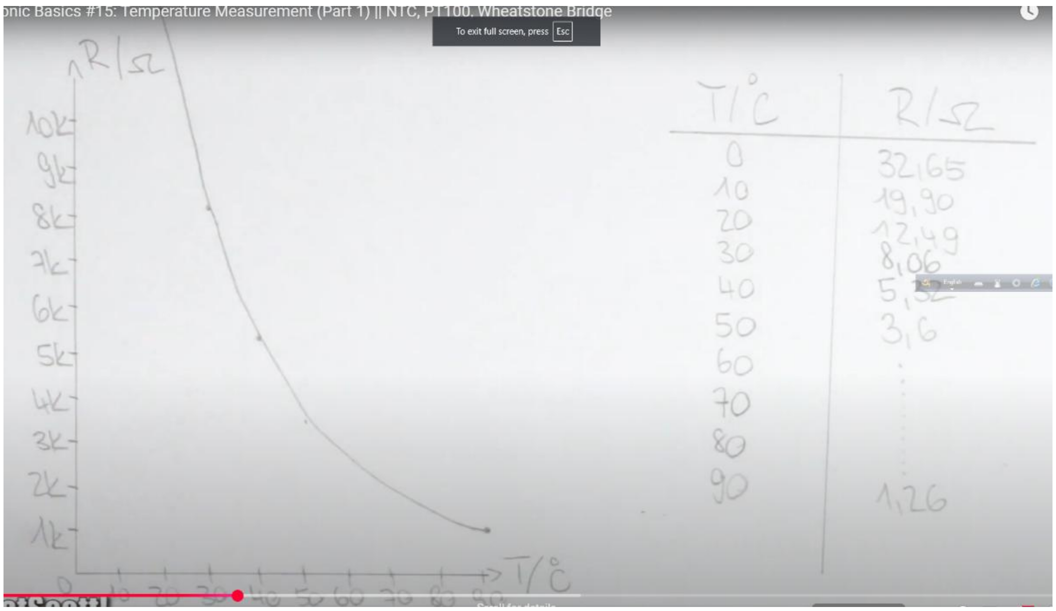


The voltage of a capacitor cannot change instantly because it needs to build up its electrostatic fields or turn it into another kind of energy.
But the current will change immediately and will slowly decrease while the capacitor reches its maximum voltage.
We can use a combination with a resistor to charge them up in a specific time.This way they can be used to dirrefent signals.
The reactive power strains our power grid.In order to get rid of this we can add a capacitor in Parallel.

15.Temperature Measurement(Part 1) || NTC,PT100,Wheatstone Bridge



A big range of resistance is useful to achieve high resolution but the rise is not linear.

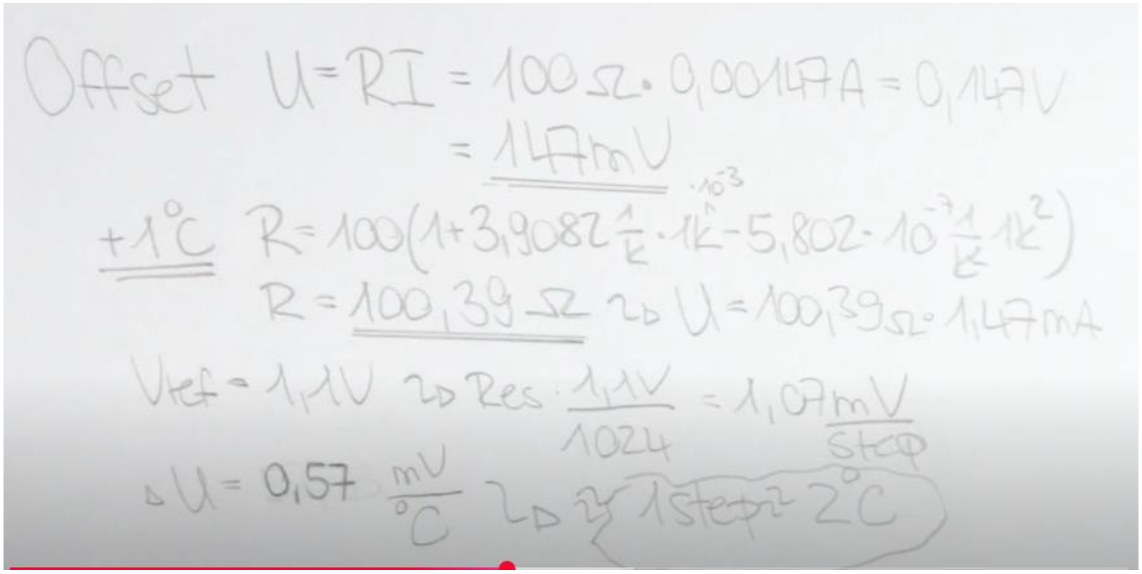


Due to that and the fact they are not awfully precise.We could also use the industrial classic PT100.

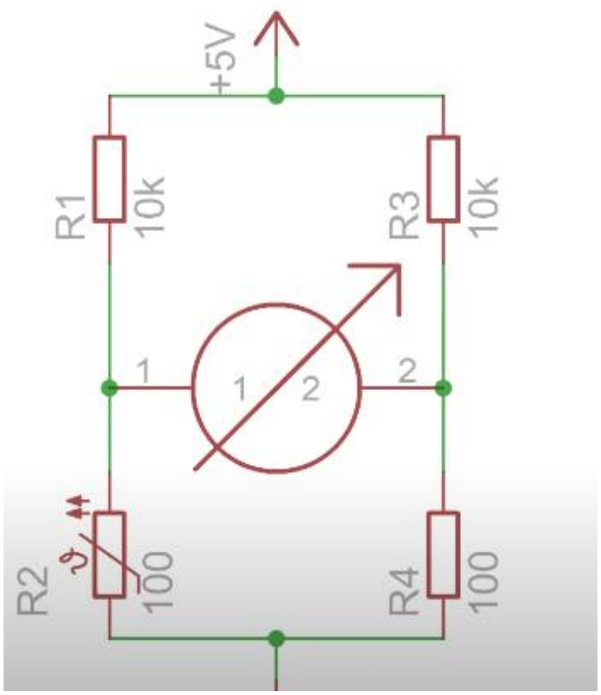
Those have a nominal resistance of 100 ohms at 0 degree celcius are so called RTD(Resistance temperature Detector).Which means if the resistance increases with higher temperatures the characteristic graph can be described with a mostly linear function and offers a decent accuracy.

In order to measure the resistance properly we need to supply a low constant current of around 1 mA with the help of LM317.

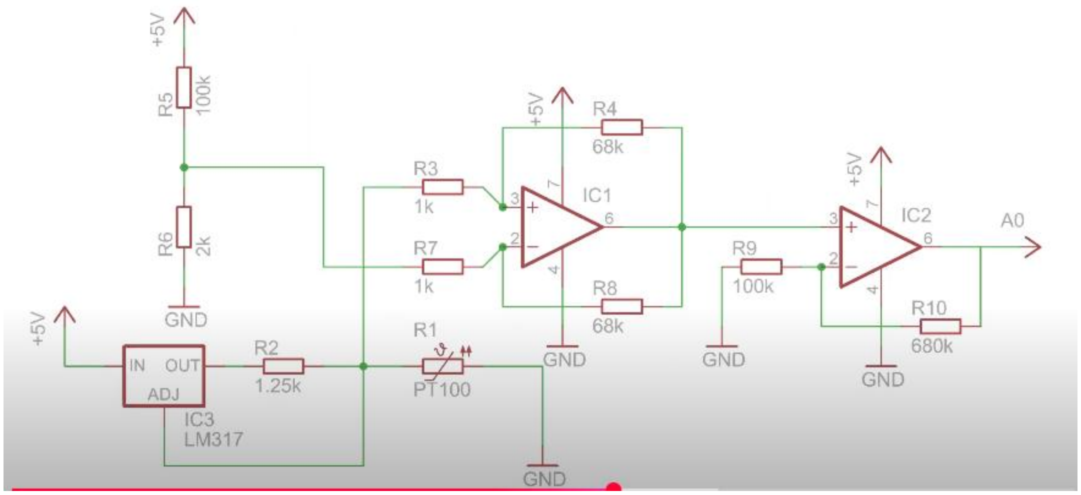
Then we measure the volt across it and calculate the resistance with the help of ohms law.



We can use Wheatstone bridge to achieve the same effect by measuring the voltage.
Voltage proportional to Resistance.
Then calculate Temperature.



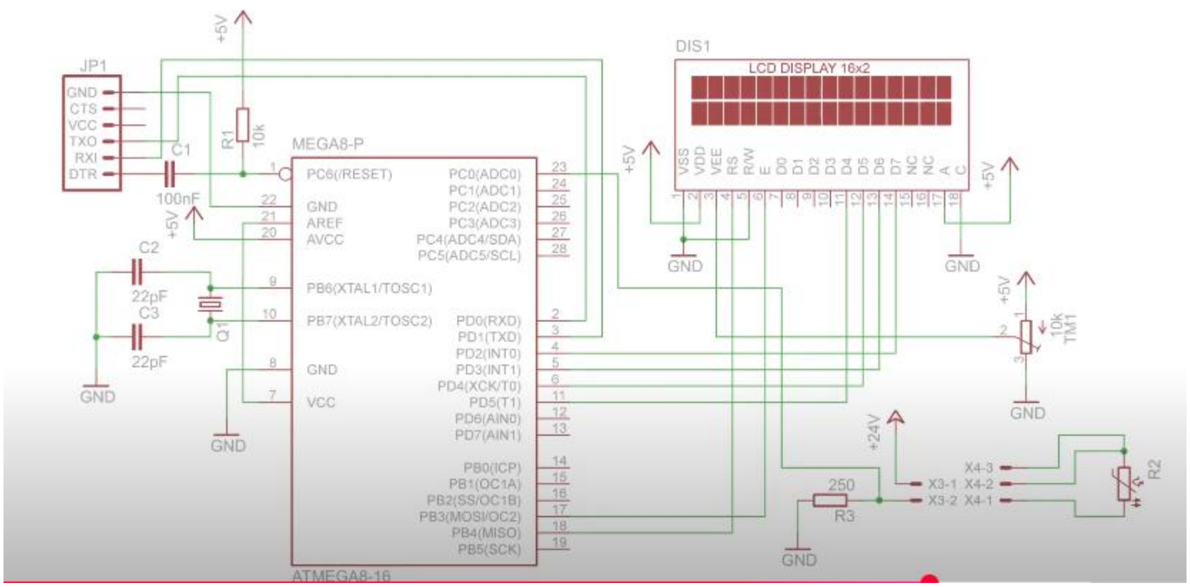
Both methods require an amplification afterwards with the help of non inverting Op-Amp circuit.



At the end we can connect the output to an analog input of the microcontroller.

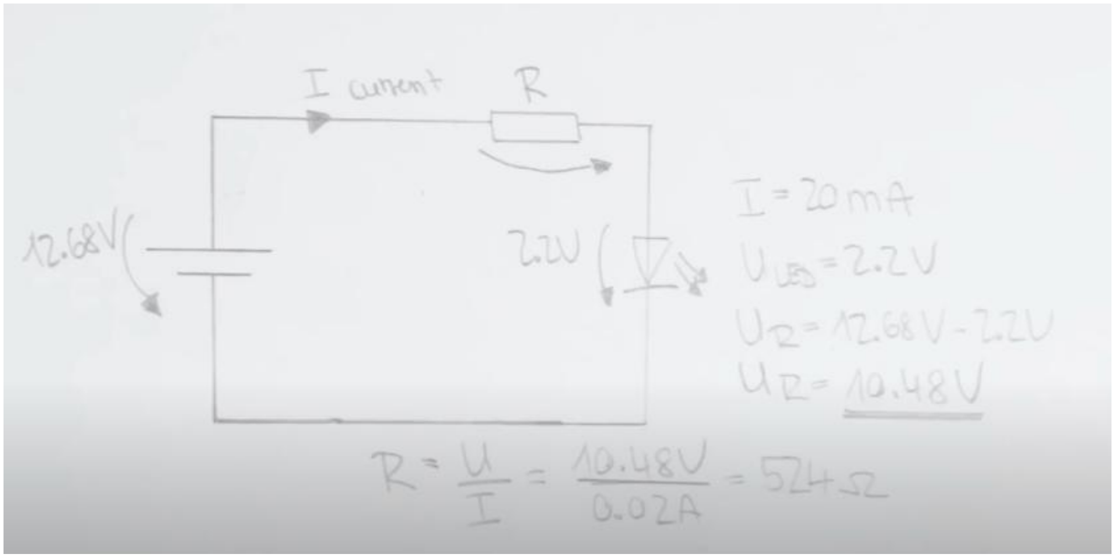
The only problem is that all of these require a lot of precison with the resistor values which is manageable through **10 turn potentiometers**.But the easier solution would be a **pre-made transmeter**.

We can connect the voltage drop of a resistor by connecting it to an analog input and use code to handle the rest of the signal processing.

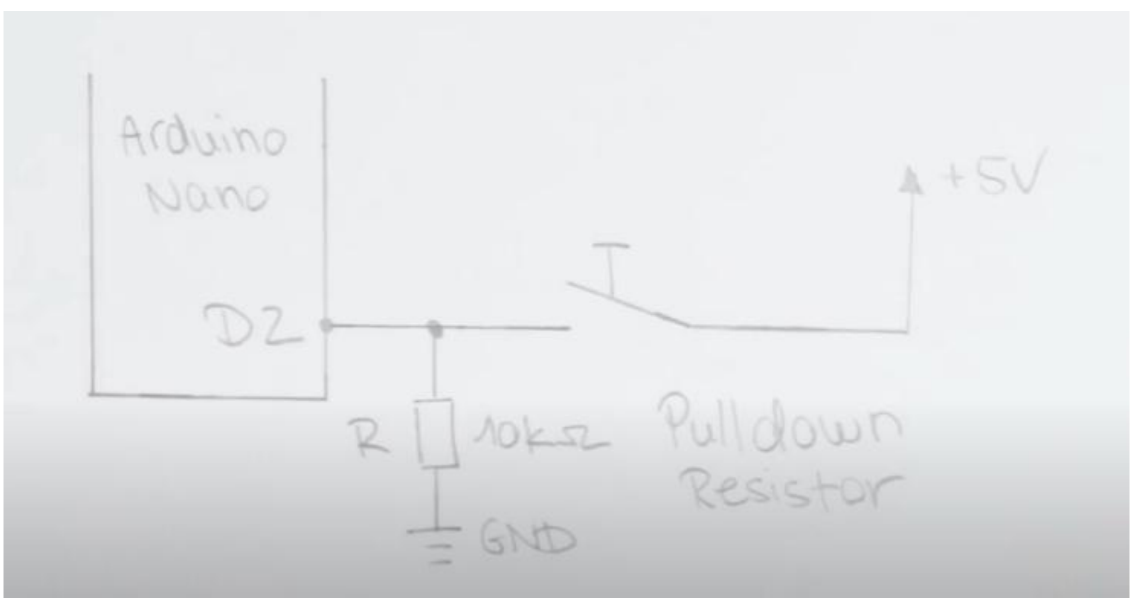


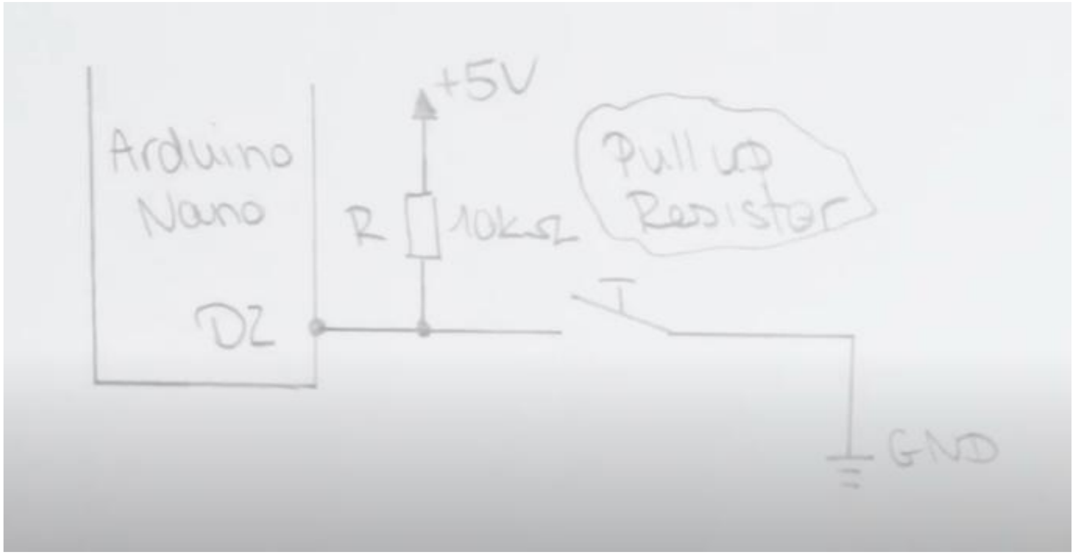
16.Resistors:

Resistor resists or limits the excess current flow and converts the additional unused power into heat.

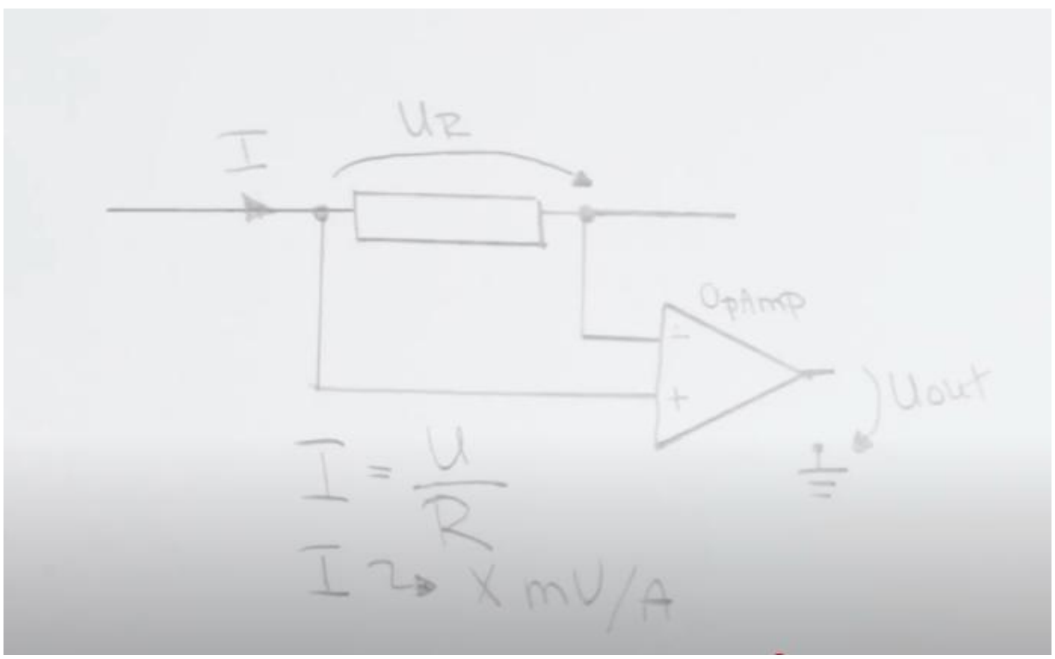


If you want to power something that needs more current but you have to keep in mind another important value for resistor-the power rating.
This can be very useful to convert 5 volt data signals from the arduino to 3.3 volt signals for the **ESP8266 Wi-Fi Module.**But this can’’tn be used as a stable 3.3 volt power source.
If you want to adjust the output voltage on the fly **Potentiometers** are the best.



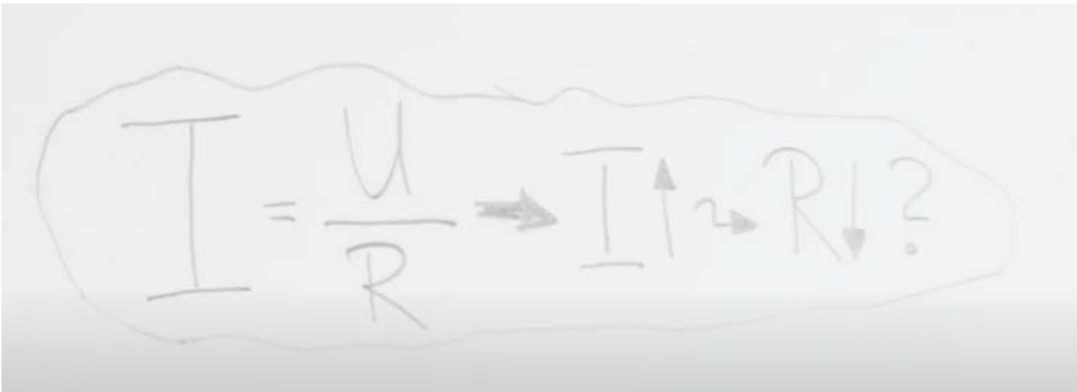


Later we can use very small value resistors,which are called ‘Shunts’ to measure current. Even the current display or multimeter use those.By amplifying the voltage drop across these,with a differential operational amplifier circuit we can calculate the current flow by knowing the value of the resistance.

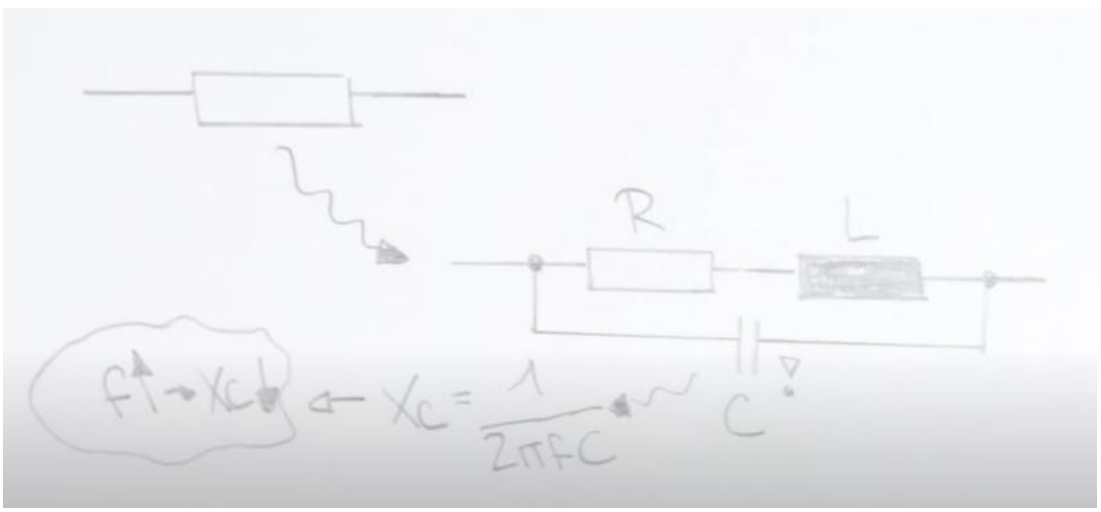


This can be useful to build a constant current source for our load.
Apart from these functions,the resistor can be used as a fuse to protect your circuit by deliberately going beyong the power rating.
They can be used to detect light like a **photoresistor.**
Or to measure **temperature like a PT100.**
Or to measure weight as a **Strain gauge.**

Resistors do not produce a phase shift in the case of AC.
As I slowly increase the frequency of the AC wave little by little,more current flows which means that the resistance should have been reduced.

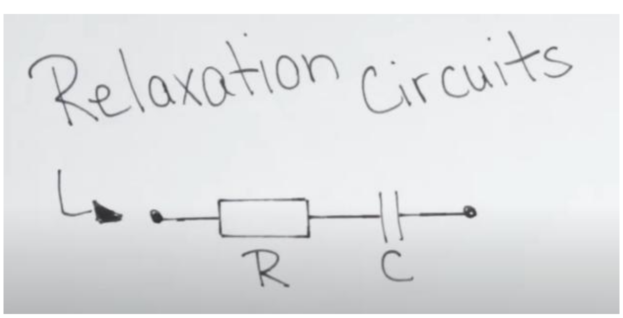


There actually exist the **parasitic** inductances and capacitances due to the structure of the resistor.In this case the decreasing reactance decreases the overall impedence and thus more current flows.

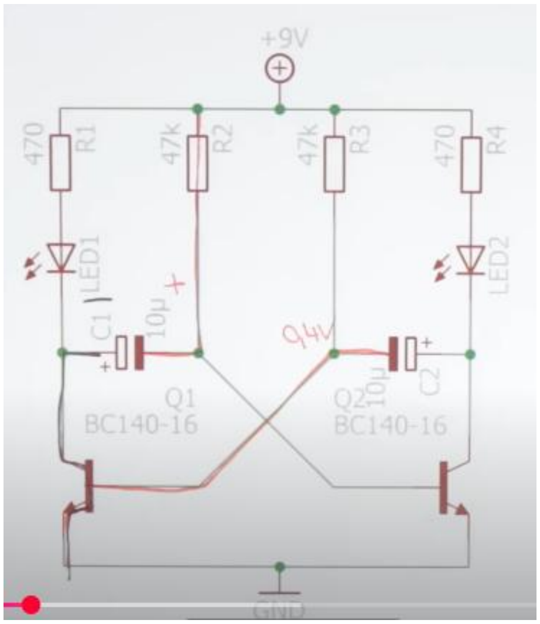


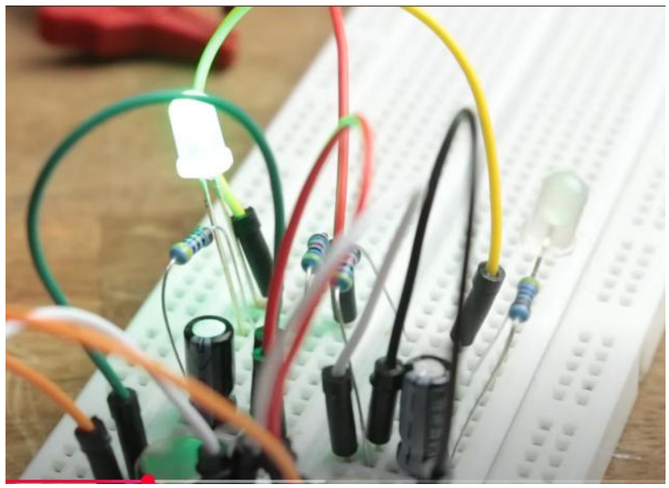
17.Oscillators || RC,LC,Crystal

First of all relaxation circuits especially with RC components.This stable multivibrator is a classic and rather simplex example.



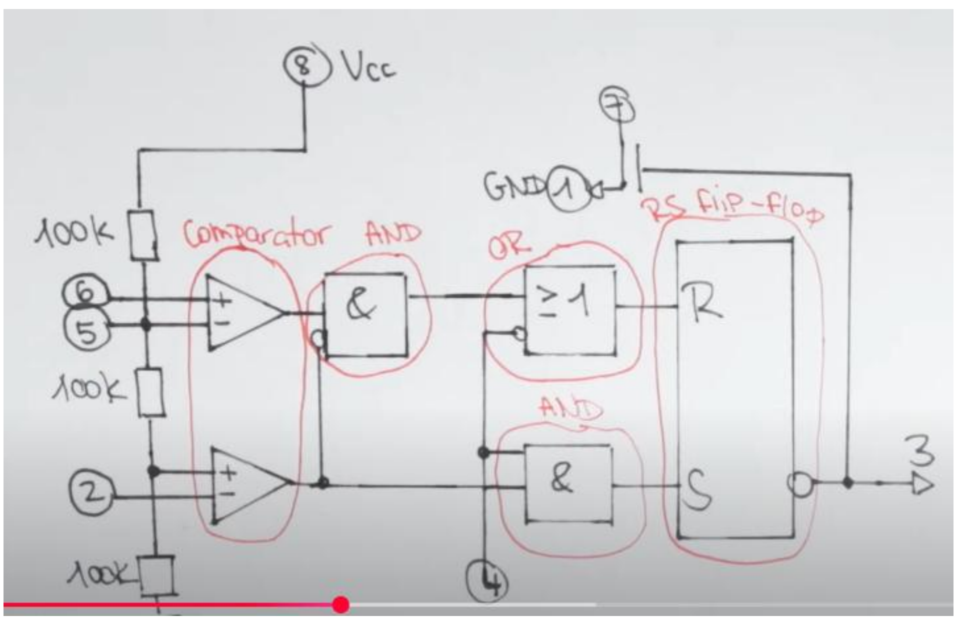
The main principle is about using two capacitors C1 and C2 which get charged alternating through a resistor to a certain threshold voltage of the transistor.This way the ttransistor becomes conductive and discharges the capacitors C1 and while the other one C2 gets charged.C2 then reaches the threshold voltage the cycle repeats with the other transistor and we successfully have created a rectangle waveform.Which is visualized by the LEDs.



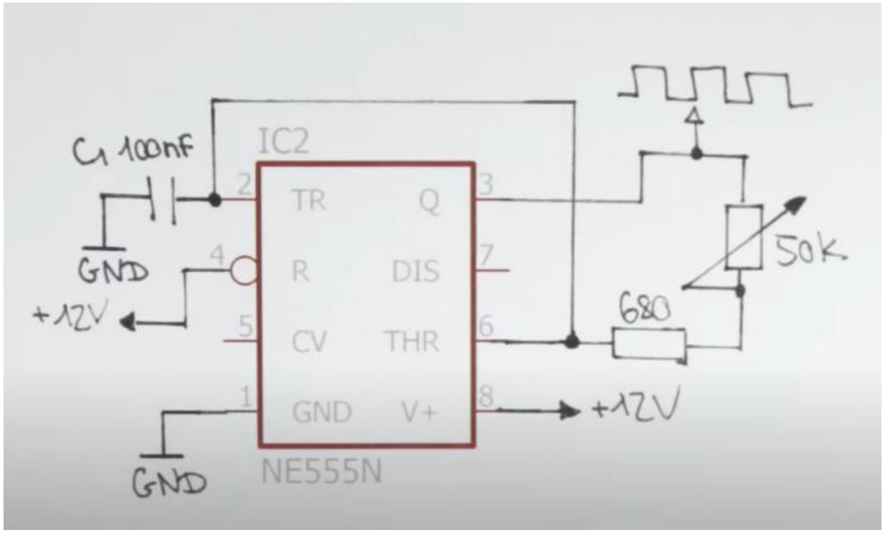


If you decrease the resistance or the capacitance,the charging and discharging process would get shorter and the frequency of the rectangle wave increases.

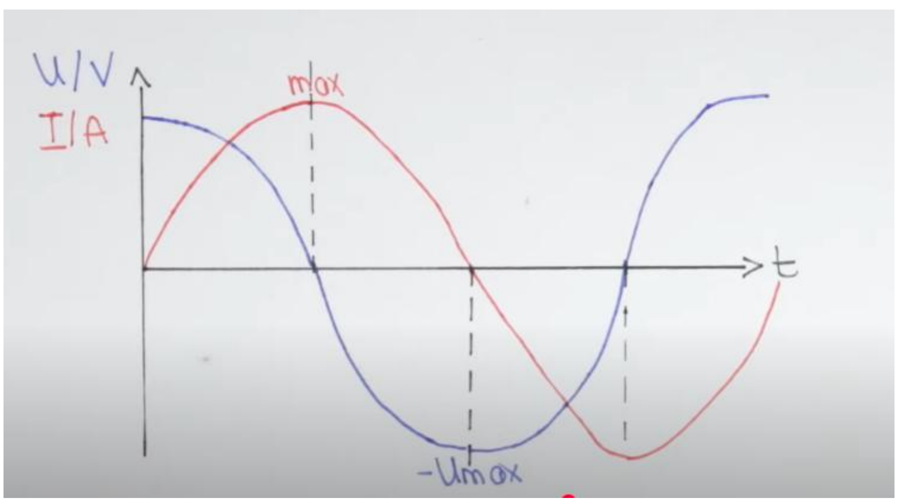
555 Timer consists of two comparators,two logic ANDS and one logic OR gates and the RS Flipflop.



The timer IC is super easy to use.



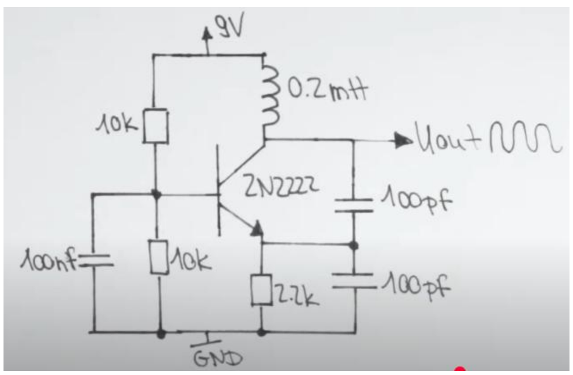
By this we control the charging and discharging of capacitor.
The capacitor can charge up to a maximum voltage and has its maximum electrostatic energy stored and after disconnecting the power supply the capacitor discharges through the **inductor** and since the current though the inductor cannot change instantly it slowly builds up until the coil has reached maximum of stored magnetic energy.This energy then converts back into the electrostatic energy of the capacitor by charging it up in reverse and the cycle repeats again.

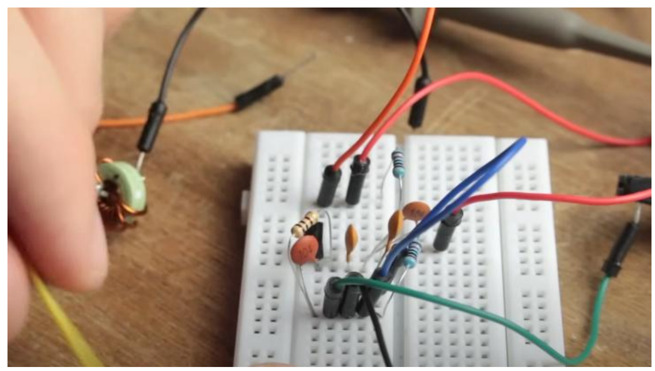


But in reality this oscillation does not last long because **parasitic resistance** is everywhere.

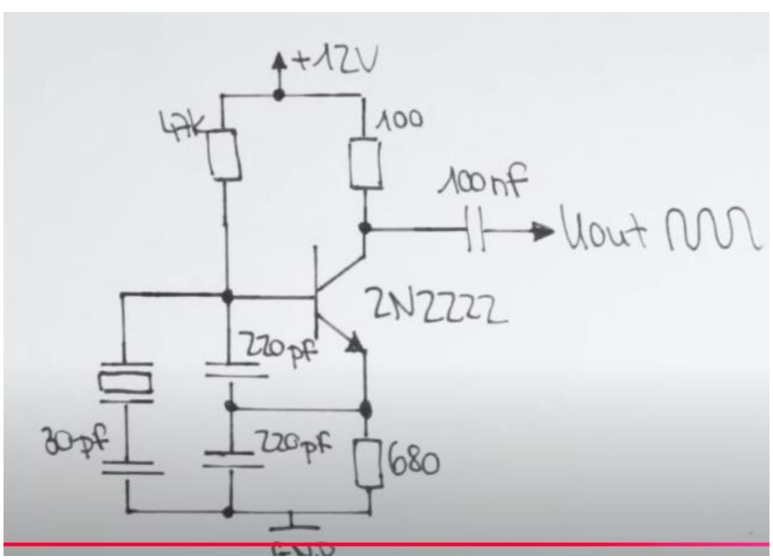
The voltage or current in an LC circuit can exceed the intial applied values of the power supply but only near the **resonance frequency.**We still need to find a way to feed the circuit energy at the right time so that the oscillation does not stop.

This can be achieved by connecting the output of the tank circuit to the input of an amplifier.

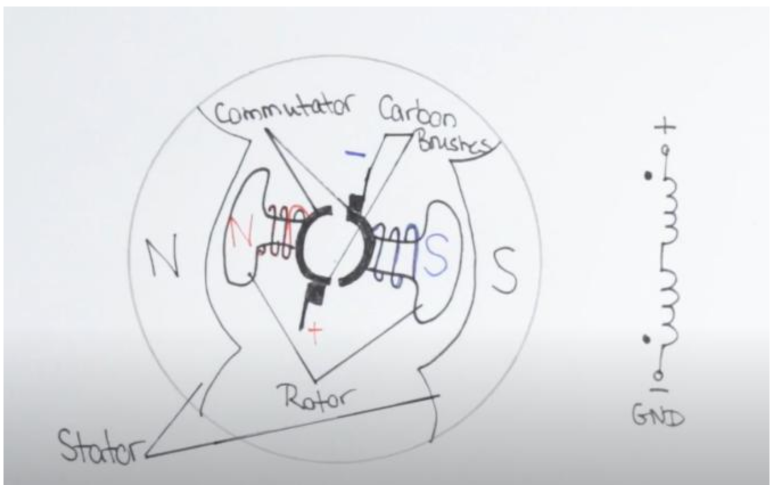


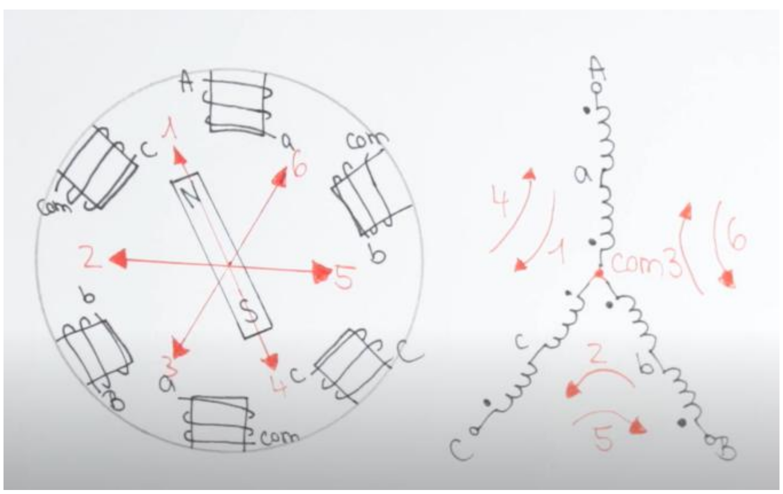


IFf we need more stable frequencies we can also use a crystal oscillator.It acts just like an LC resonator.

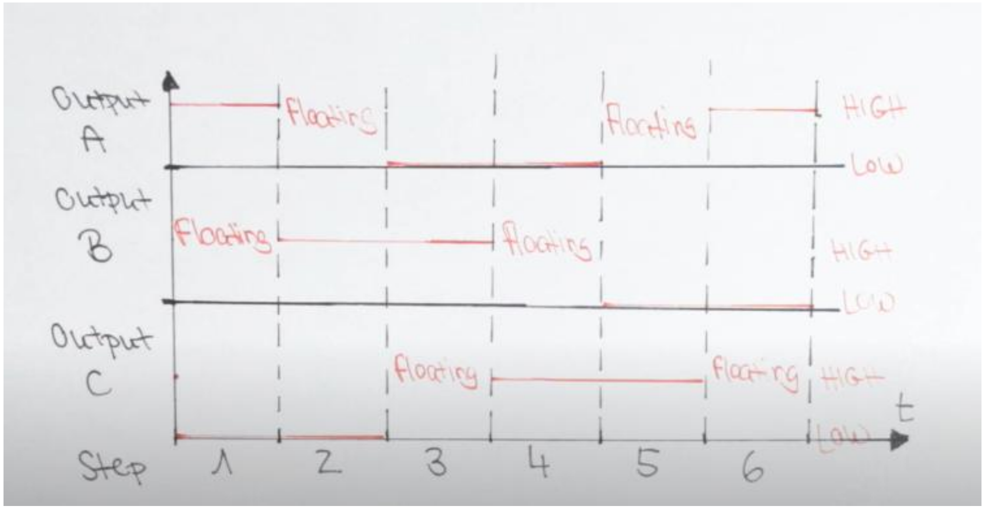


19.DC & brushless DC Motor + ESC(Electric Speed Controller):

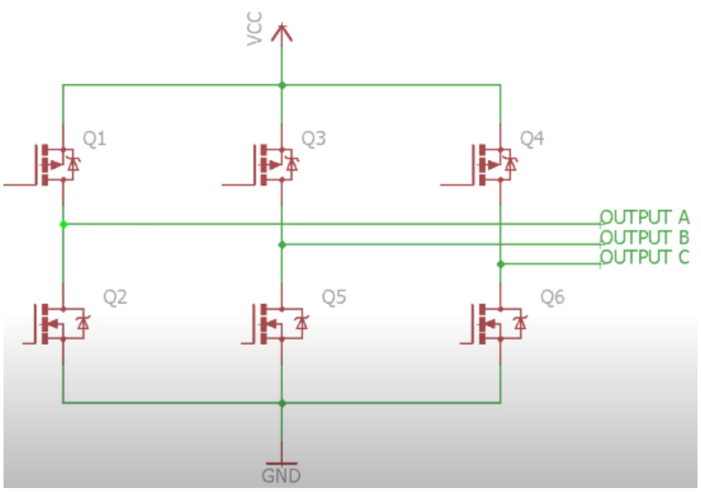


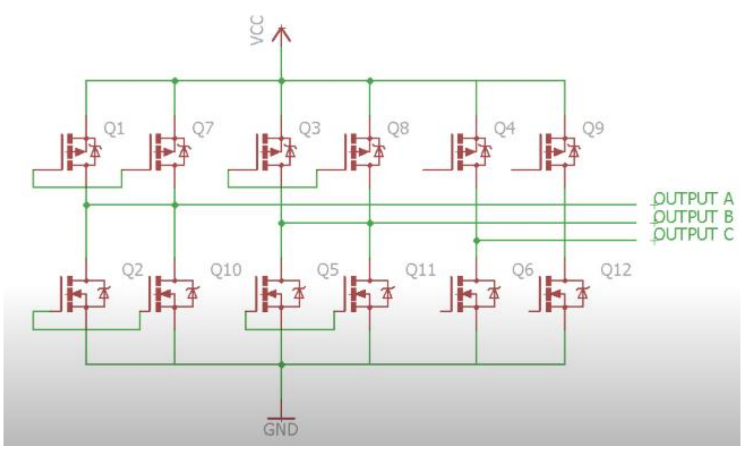


The brushless motor uses ESC.



The three step of output are higher,Low and Floating.

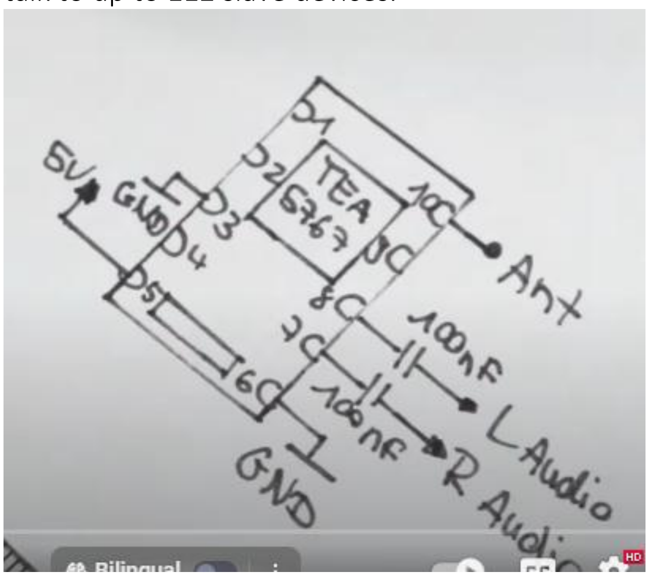


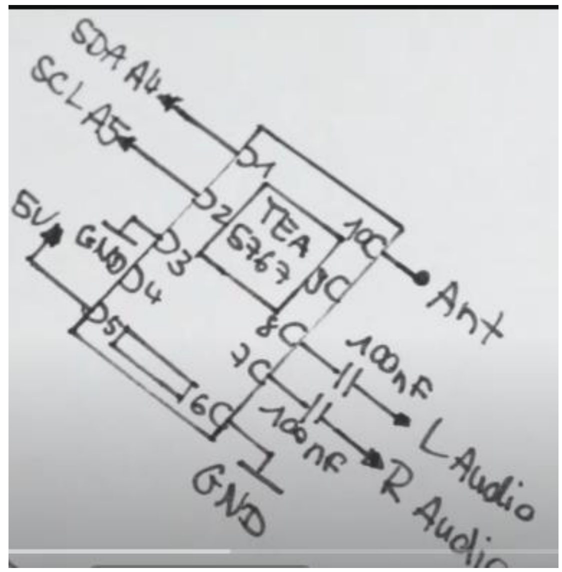


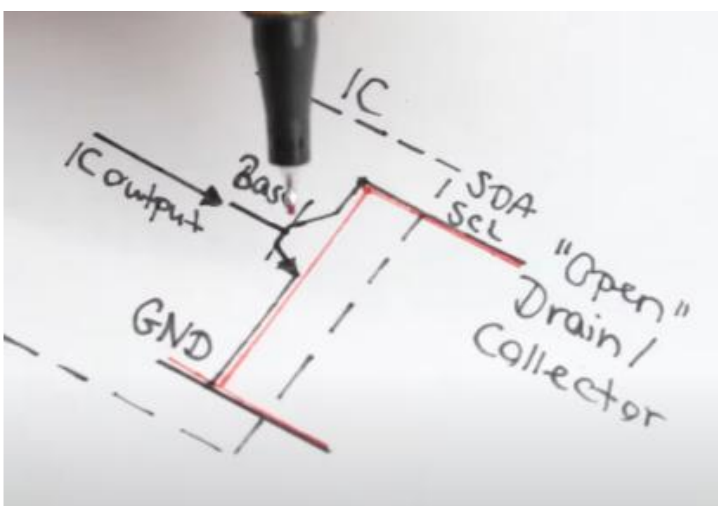
The higher KV raing you can get more RPM.

19.I2C and how to use it:

If you ever implemented a row of complex IC with many functions like telling you the exact date and time expanding your iio outputs with 16 12 bit pwn pins or receiving your favourite fm radio project then you might be familiar with **i²C** also known as **2-wire bus interface**.It is a popular communication protocol that allows one or more Master devices in this case not arduino nano to talk to up to 112 slave devices.







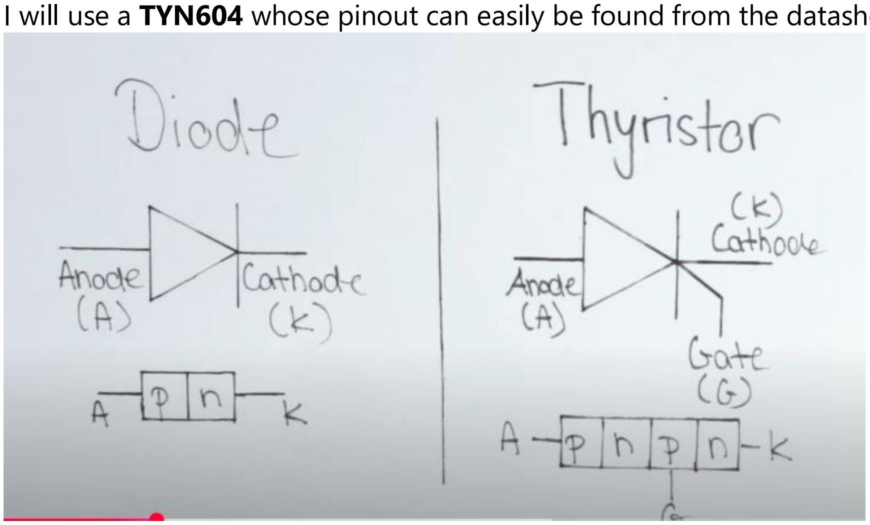
The only resource to know about the i²C devices is the Datasheet.

```
Serial.begin(9600);
Wire.begin();
Wire.beginTransmission(0x60);
Wire.write(0x2D);
Wire.write(0xB1);
Wire.write(0x10);
Wire.write(0x10);
Wire.write(0x40);
Wire.endTransmission();
delay(100);

Wire.beginTransmission(0x62);
Wire.write(0x05);
Wire.write(0xFF);
Wire.endTransmission();
delay(100);

unsigned char buffer[5];
Wire.requestFrom(0x60,5);
if (Wire.available()){
 for (int i=0; i<5; i++){
   buffer[i]=Wire.read();
 }
fa=(((buffer[0]&0x3F)<<8)+buffer[1])*32768/4-225000;
Serial.print("FM ");
Serial.println((fa/1000000));
```


20.Thyristor,Triac || Phase Angle Control:

Thyristors, controllable diodes, allow power regulation in AC circuits. The video demonstrates their operation, including a phase angle control circuit using a microcontroller to manage power delivery to a light bulb. It highlights the importance of latching current and discusses the implications of using thyristors in practical applications.

Thyrister has fours layers and adds an additional gate terminal.
I will use a **TYN604** whose pinout can easily be found from the datasheet.



A Thyristor (also called a Silicon Controlled Rectifier, SCR) is a four-layer semiconductor device that acts as a switch, controlling high voltages and currents. It has three terminals: Anode (A), Cathode (K), and Gate (G).

Working Principle of a Thyristor:

1.Forward Blocking Mode (OFF State)

When the Anode is positive with respect to the Cathode, but no voltage is applied to the Gate, the thyristor remains OFF.

A small leakage current flows due to the junction capacitance, but no significant conduction occurs.

2.Triggering (Turning ON)
If a small positive voltage is applied to the Gate terminal, it injects carriers into the semiconductor layers, reducing the resistance and allowing large current flow from Anode to Cathode.

Once turned ON, the thyristor remains in conduction even if the Gate signal is removed.

3.Forward Conduction Mode (ON State)

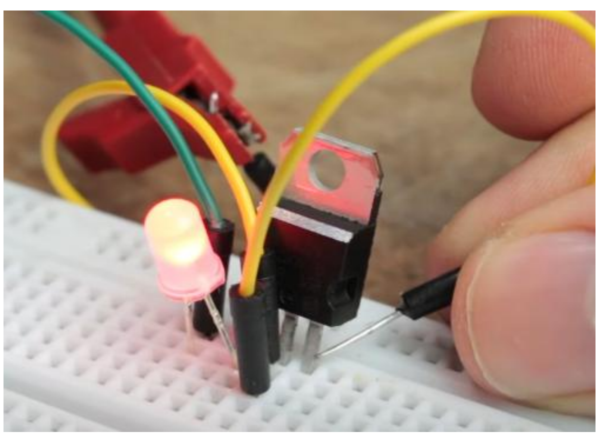
After being triggered, the thyristor behaves like a closed switch, allowing continuous current flow.

The only way to turn it OFF is to reduce the anode current below the holding current (by switching off the supply or using a commutation circuit).

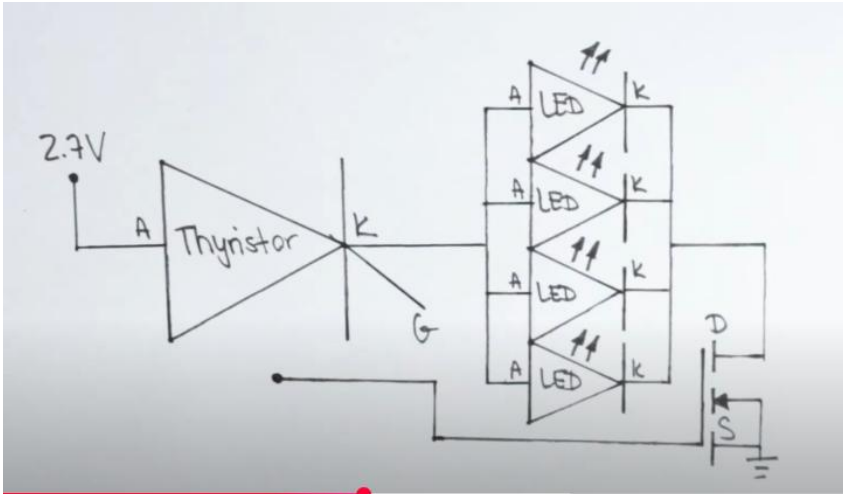
4.Reverse Blocking Mode

When the Cathode is positive relative to the Anode, the thyristor behaves like a reverse-biased diode, blocking current flow.

A small reverse leakage current flows, but breakdown can occur at very high reverse voltages.



In this connection the LED does light but not for long.
The thyristor remains in conduction even if the Gate signal is removed.The reason for this unwanted behavior we need to reach a latching current.



The thyristor remains conducted until we fall below the holding current.
The process of using a MOSFET to interrupt current flow showcases the differences between MOSFETs and thyristors in electronic circuits. This is essential for understanding their applications.
MOSFET vs. Thyristor in Current Interruption:

MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor):

Acts as a voltage-controlled switch.

It has three terminals: Gate (G), Drain (D), and Source (S).

The current flow is controlled by applying a voltage to the Gate.

Turning ON: When a sufficient positive gate-to-source voltage (V_GS) is applied, the MOSFET enters conduction mode.

Turning OFF: When the Gate voltage is removed, the MOSFET stops conducting immediately.

Advantage: It can switch ON and OFF quickly and does not require continuous current to maintain its state.

Thyristor (Silicon Controlled Rectifier, SCR):

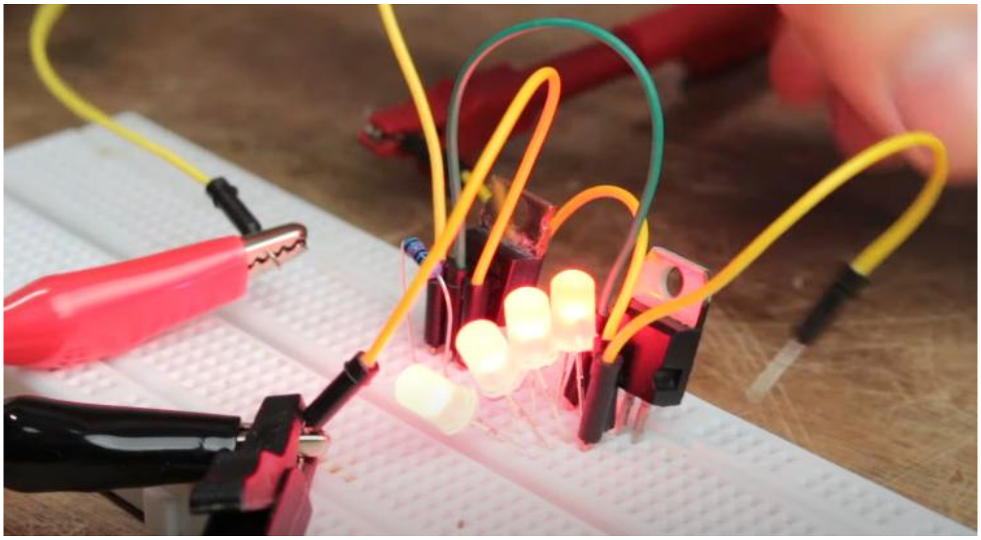
Acts as a latching switch.

It has three terminals: Gate (G), Anode (A), and Cathode (K).

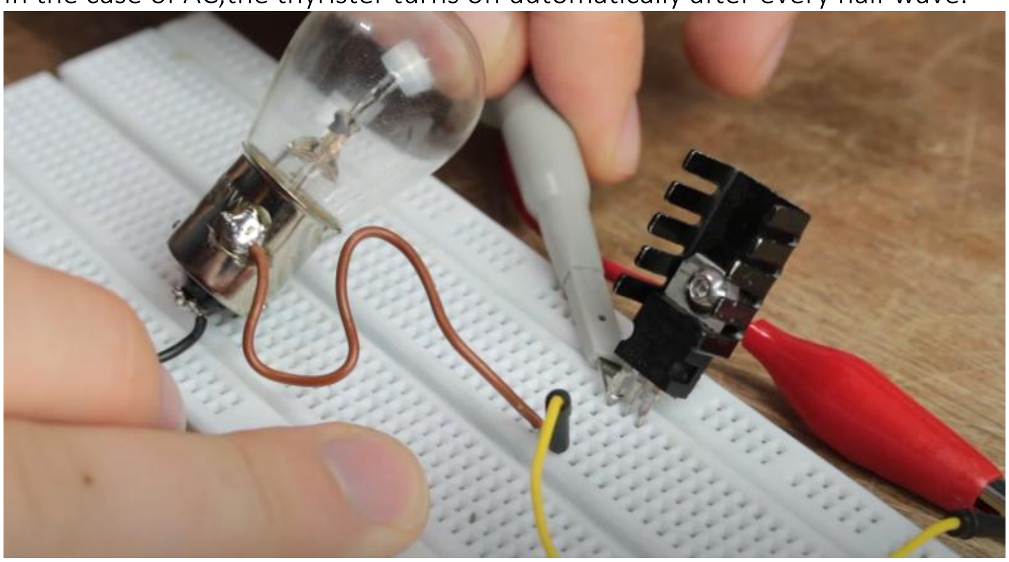
Turning ON: A small pulse to the Gate allows large current to flow from Anode to Cathode.

Turning OFF: Unlike MOSFETs, the thyristor stays ON even after the Gate pulse is removed. To turn it OFF, the current must be reduced below the holding current (e.g., using an external commutation circuit).

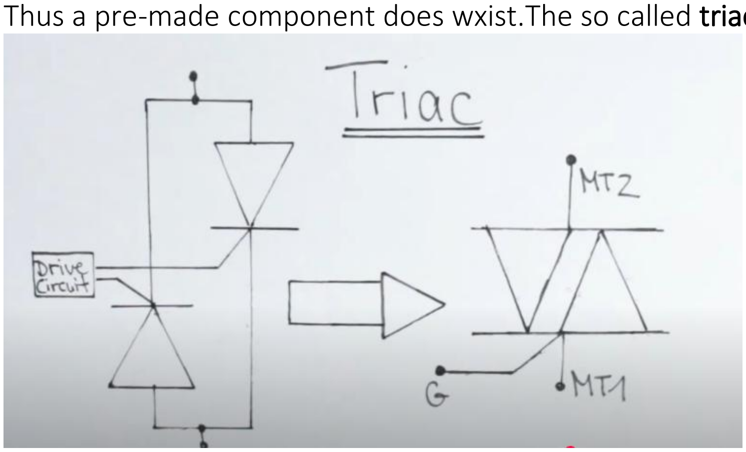
Advantage: Suitable for high-power applications, but requires external methods to turn it off.



If we replace the LED with a heavy load,the temperature of the thyrister increases gradually.
In the case of AC,the thyrister turns off automatically after every half wave.



In order to solve this,we need two thyristors in a inverse parallel configuration.
Thus a pre-made component does wxist.The so called **triac**.



A TRIAC is a bidirectional semiconductor switch that can control AC power. It is essentially an extension of a thyristor (SCR) but can conduct in both directions, making it highly useful for AC applications like dimmers, motor control, and heaters.

Structure of TRIAC

A TRIAC consists of five semiconductor layers (PNPNP) and has three terminals:

1.Main Terminal 1 (MT1)

2.Main Terminal 2 (MT2)

3.Gate (G)

Unlike an SCR, which only conducts in one direction, a TRIAC can conduct in both positive and negative cycles of AC voltage.

Operating Modes of TRIAC

A TRIAC can be turned ON by applying a small trigger current to the Gate. It operates in four quadrants based on the polarity of MT1, MT2, and Gate:

1. Quadrant I (MT2 +, Gate +) – Forward Triggering

MT2 is positive with respect to MT1.

A positive Gate pulse triggers conduction.

Current flows from MT2 to MT1.

2. Quadrant II (MT2 +, Gate -) – Forward Triggering (Negative Gate)

MT2 is positive with respect to MT1.

A negative Gate pulse can also trigger conduction.

Current still flows from MT2 to MT1.

3. Quadrant III (MT2 -, Gate -) – Reverse Triggering

MT2 is negative with respect to MT1.

A negative Gate pulse triggers conduction.

Current flows from MT1 to MT2.

4. Quadrant IV (MT2 -, Gate +) – Reverse Triggering (Positive Gate)

MT2 is negative with respect to MT1.

A positive Gate pulse can also trigger conduction.

Current flows from MT1 to MT2.

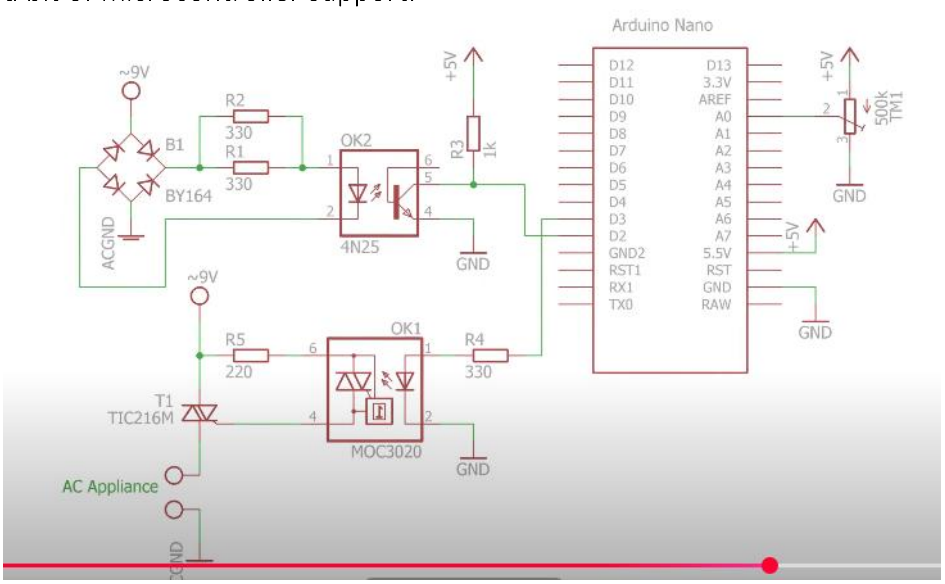
Once turned ON, the TRIAC remains conducting even if the Gate signal is removed until the current falls below the holding current (similar to an SCR).

How TRIAC Controls AC Power

Since a TRIAC can be triggered in both halves of an AC cycle, it is used in AC power control applications.

By controlling the firing angle (delaying the trigger pulse), the amount of power delivered to a load can be varied smoothly, which is useful for light dimmers, fan speed controllers, and heater regulators.

The only problem and control each half wave by letting only a part of thourgh the triac,we need a bit of microcontroller support.



21.OpAmp(Operational Amplifier)

Basic Features of an Op-Amp

- It has **five terminals**, but mainly, we focus on **three:**
    1. **Inverting Input (-)**
    2. **Non-Inverting Input (+)**
    3. **Output**

- Two additional terminals:
  4. **Positive Power Supply (+V)**
  5. **Negative Power Supply (-V)**
- **Very High Gain**: A tiny difference in input voltage produces a large output voltage.
- **High Input Impedance**: It does not draw much current from the input source.
- **Low Output Impedance**: Provides strong output to the next circuit stage.

Op-Amp Working Principle

- The Op-Amp **compares the voltages** at the two inputs and **amplifies the difference**.
- **If the non-inverting input (+) is higher** than the inverting input (-), the output is **positive**.
- **If the inverting input (-) is higher**, the output is **negative**.

This property is used in various circuit applications.

Common Op-Amp Configurations

1 Buffer (Voltage Follower)

- **Purpose**: Transfers voltage without amplifying or loading the source.
- **Gain**: 1 (Output = Input).
- **Use case**: Used in signal isolation.

2 Inverting Amplifier

- **Input applied to (-) terminal**, the output is **inverted**.
- **Gain = -Rf / Rin** (negative means phase reversal).
- **Use case**: Audio processing, signal inversion.

3 Non-Inverting Amplifier

- **Input applied to (+) terminal**, the output is **in phase**.
- **Gain = 1 + (Rf / Rin)**.
- **Use case**: Signal amplification without phase change.

4 Summing Amplifier

- **Adds multiple input voltages**.
- **Use case**: Audio mixing, signal processing.

5 Differential Amplifier

- **Outputs the difference between two input voltages**.

- **Use case**: Used in instrumentation and sensor applications.

6 Integrator

- **Performs mathematical integration** (output is the integral of input).
- **Use case**: Used in waveform generation.

7 Differentiator

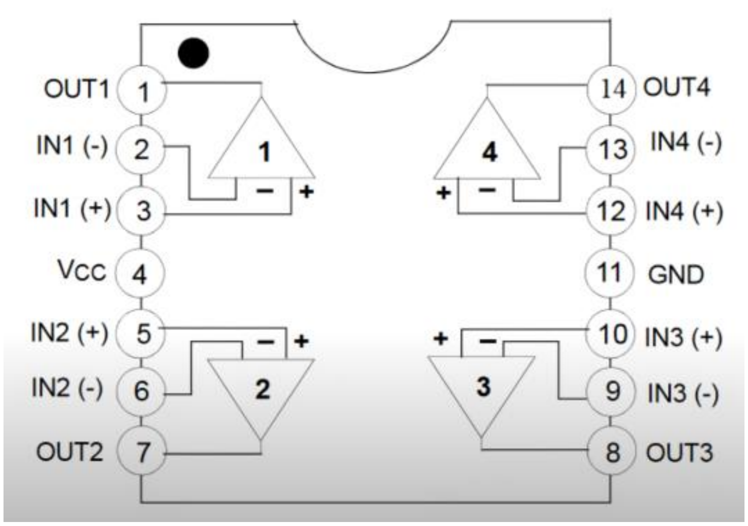
- **Performs differentiation** (output is the derivative of input).
- **Use case**: Edge detection in signals.

Applications of Op-Amps

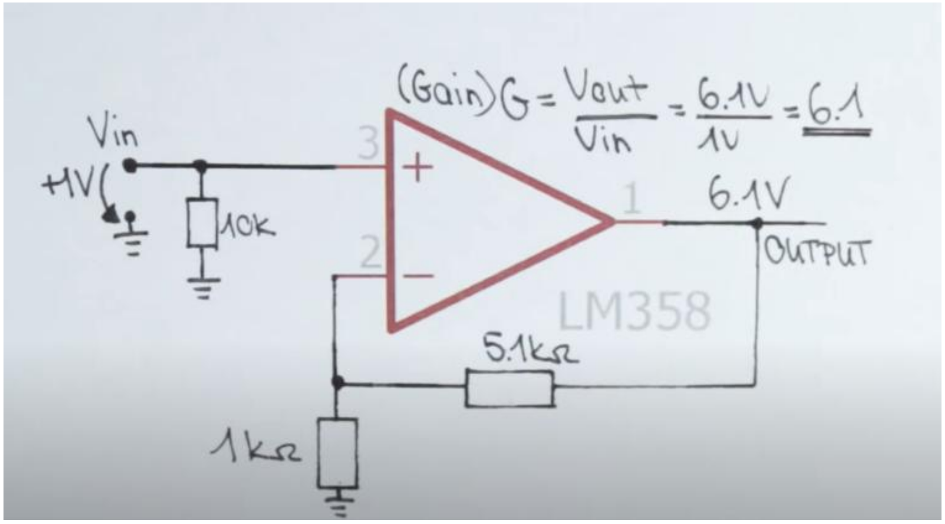
✅ **Signal amplification** (microphones, speakers)
✅ **Filters** (low-pass, high-pass, band-pass)
✅ **Oscillators** (waveform generation)
✅ **Voltage regulators**
✅ **Analog computation** (addition, subtraction, etc.)
✅ **Comparator circuits**

Why Are Op-Amps So Important?

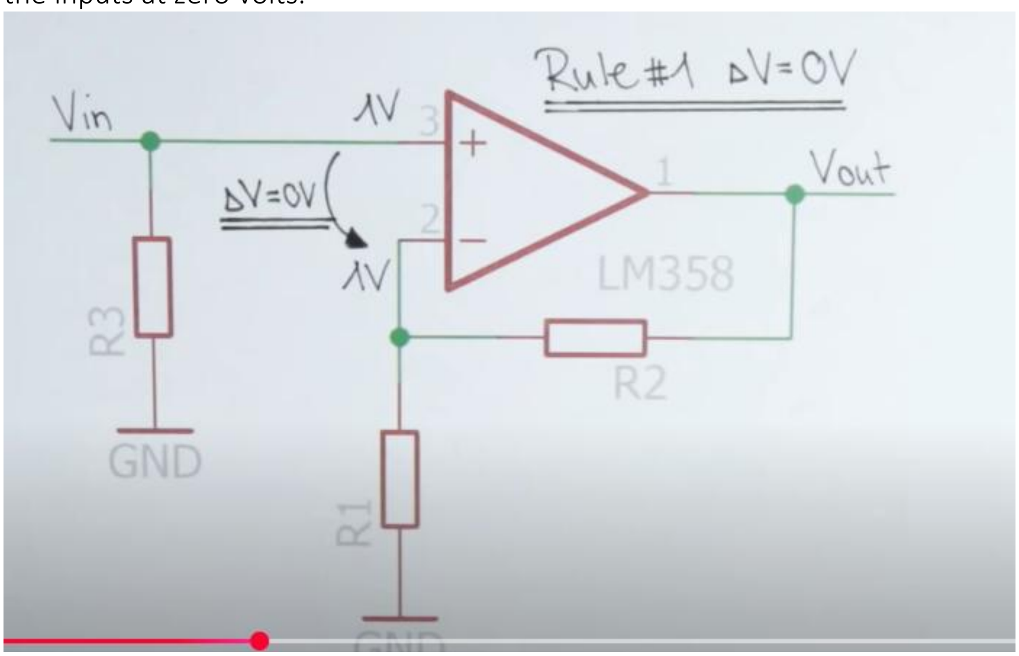
They are **versatile, low-cost, and highly efficient**, making them the **heart of analog electronics** in almost every circuit.



Here we used the popular LM358.

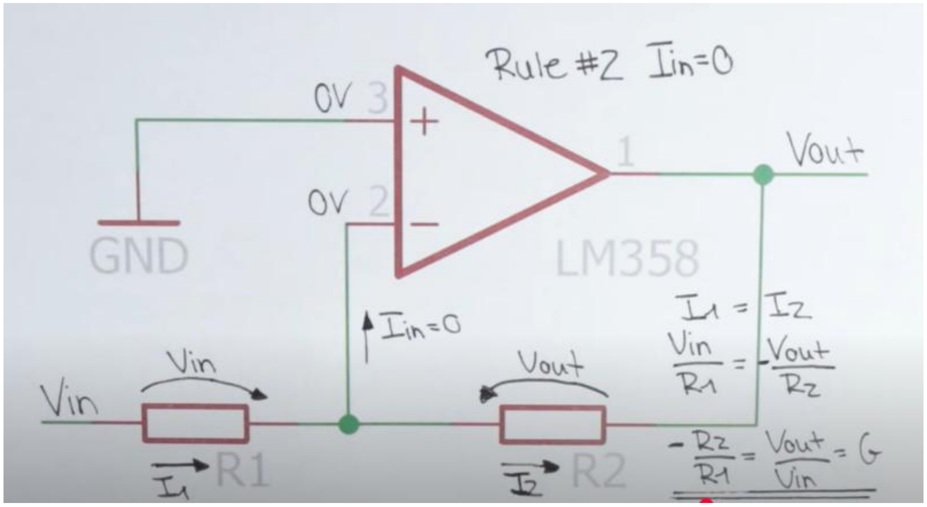


The output of an op-amp will always attempt everything to keep the voltage difference between the inputs at zero volts.

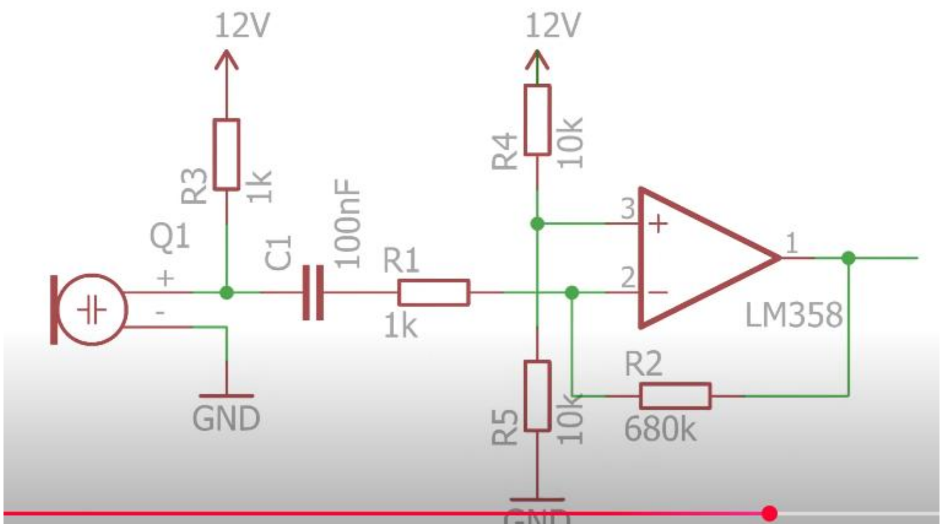


That means they needs to be the same voltage drop across R1 as the input voltage Vin.

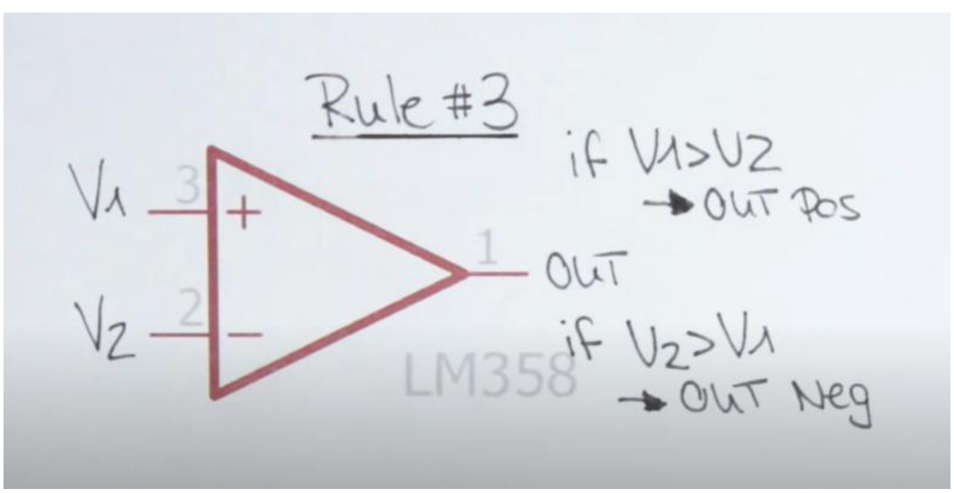
There exists Rail-to-rail Op-Amp that can achieve an output voltage which equals their supply voltage.



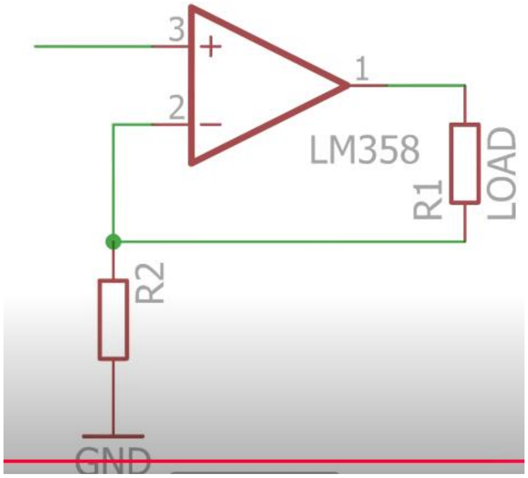
Rule #2 Iin=0
0V 3
0V 2
Vout
GND
LM358
Iin=0
I1 = I2
Vin/R1 = -Vout/R2
-R2/R1 = Vout/Vin = G
Vin
Vin
Vout
R1
R2
I1
I2

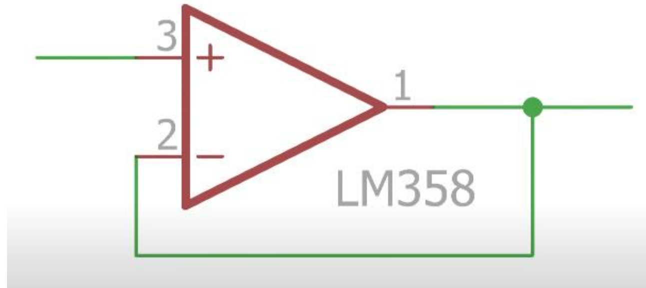


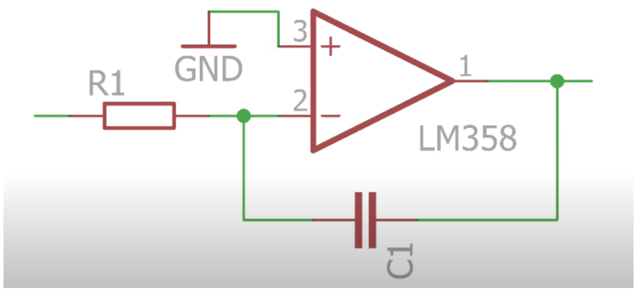
12V
12V
R3
1k
R4
10k
Q1
C1
100nF
R1
1k
LM358
GND
R5
10k
R2
680k

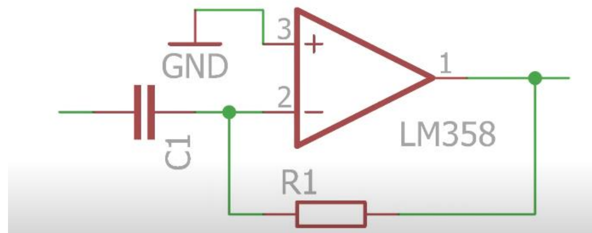


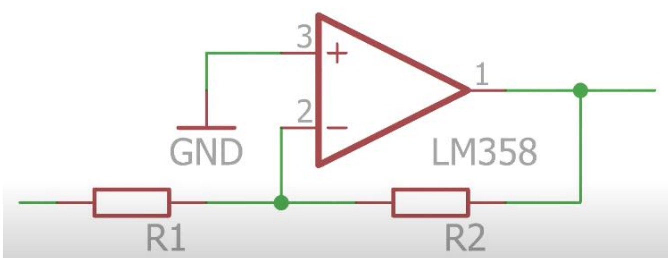
Rule #3
V1
V2
OUT
LM358
if V1>V2
→ OUT Pos
if V2>V1
→ OUT Neg

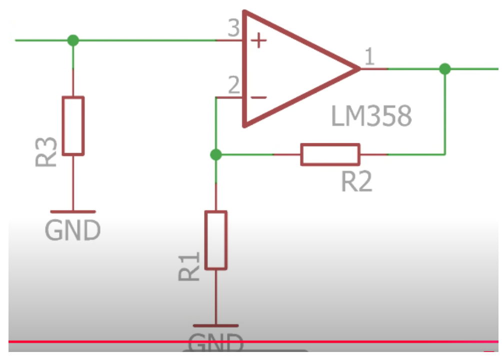

3
2
1
LM358
R1
LOAD
R2
GND

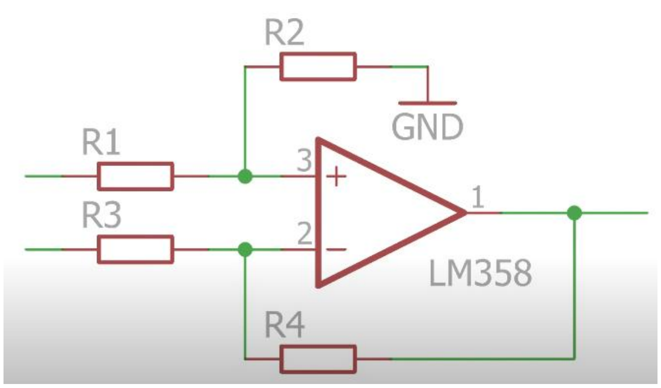

3
2
1
LM358

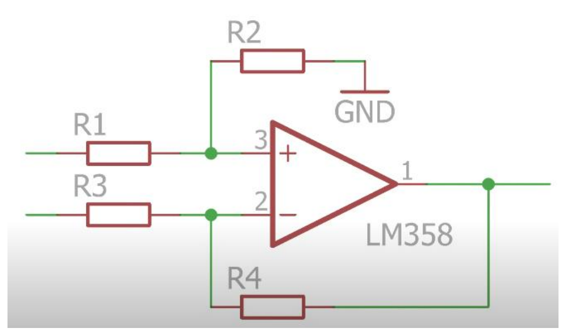

GND
R1
3
2
1
LM358
C1

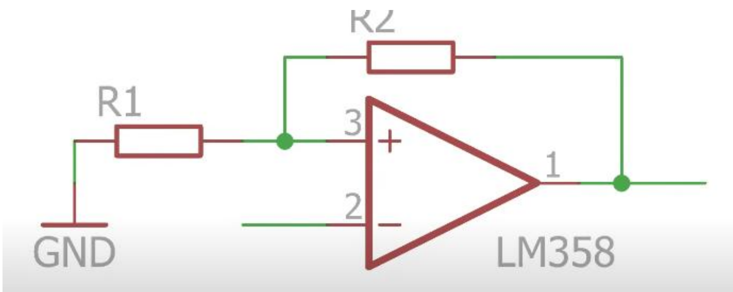

GND
3
2
1
C1
LM358
R1


3
2
1
GND
LM358
R1
R2


3
2
1
LM358
R3
R2
GND
R1
GND

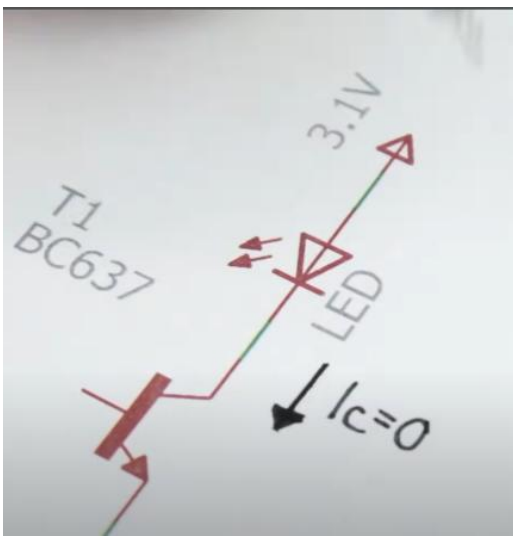

R2
GND
R1
3
R3
2
1
LM358
R4


R2
GND
R1
3
R3
2
1
LM358
R4

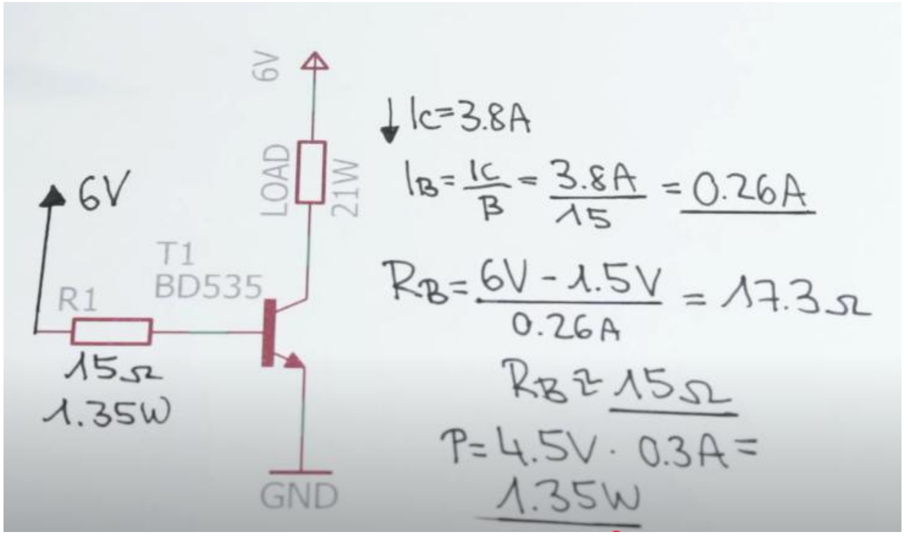


22.Transistors:

Here we used BC637 npn BJT.You should check out the datasheet for the pinouts.
You should be always carefull that your supply voltage doesn't exceed the Vce.
I powered up the circuit and noticed that no collector current was flowing so far because the BJTs collector current is the product of its base current and emitter current.



So lets fix that by connecting the base straight to the supply voltage.Which lets indeed a rather big base urrent flow but also destroys the transistor along the way.The reason for that is the base emitter path basically consists of a diode with a forward voltage of around 1 volt.Thus every voltage above this 1 volt value will increase the current drastically.So we need a current limiting transistor to keep that under control.
But the output is not perfect that the collector current is smaller than we estimated because of the voltage drop of the collector emitter path.
Lets use the PNP BJT which connects to the circuit as before but tha polarities reversed.In nutshell that means the ground potential needs to get applied to the base resistor in order to switch on the load instead of a positive voltage.
If we want to handlea heavy AC load we have to use BD535.



BJTs operate less efficiently due to higher power loss, typically around 6 watts, when dealing with substantial collector current. This can lead to overheating issues.

1 Power Loss Due to Collector-Emitter Voltage Drop (V_CE)

- In a **saturated BJT**, there is always a small **voltage drop** between the **collector (C) and emitter (E)**, typically **0.2V to 2V** depending on the current.
- **Power loss (P) = V_CE × I_C**
- If the collector current (**I_C**) is high, even a small **V_CE** leads to **significant power dissipation** (e.g., **6W or more** in high-power applications).

---

2 High Base Drive Current (I_B)

- BJTs are **current-controlled devices**, meaning they require a **base current (I_B)** to operate.
- Power loss also occurs in the base circuit:
  - **Base drive power loss = V_BE × I_B**
  - Even though **V_BE ≈ 0.7V**, the base current can be significant in high-power applications, increasing overall power loss.

---

3 Switching Losses (Slow Turn-On and Turn-Off)

- Unlike **MOSFETs**, which are **voltage-controlled**, BJTs take time to turn **ON and OFF** due to **charge storage** in the base.
- During this transition period, both **V_CE and I_C** are nonzero simultaneously, leading to **energy dissipation** as heat.
- Faster switching requires **more base drive current**, increasing total power loss.

---

4 Thermal Runaway (Self-Heating Effect)

- As **temperature increases**, the **BJT's collector current (I_C)** increases due to the **positive temperature coefficient** of the transistor.
- More current leads to **higher power dissipation**, which further increases **temperature**, potentially causing **thermal runaway** (destruction of the transistor).
- **MOSFETs do not suffer from this issue as severely**, making them more efficient in power applications.

Darlington transistors, like the TIP 142, consist of two BJTs, which greatly increases current gain and allows for lower base current usage for switching. While Darlington transistors improve current handling, they also introduce higher voltage drops, leading to potential additional power loss compared to regular BJTs.

Structure of a Darlington Transistor

A Darlington transistor consists of **two BJTs** connected in **cascade**:

- The **first (small) transistor (Q1)** amplifies the input current.
- The **second (larger) transistor (Q2)** further amplifies the current from Q1.

Connections:

1. **Emitter of Q1 → Base of Q2**
2. **Collector of both Q1 and Q2 are connected together**
3. **Emitter of Q2 is the final output**

This configuration results in a **much higher gain** than a single transistor.

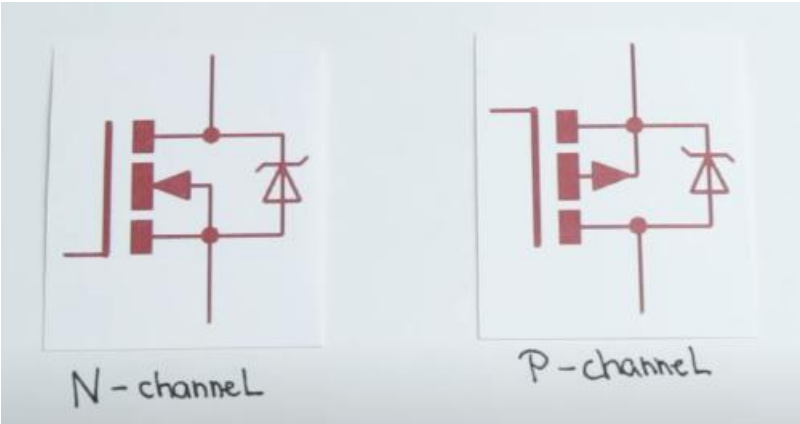
⚙ Working Principle

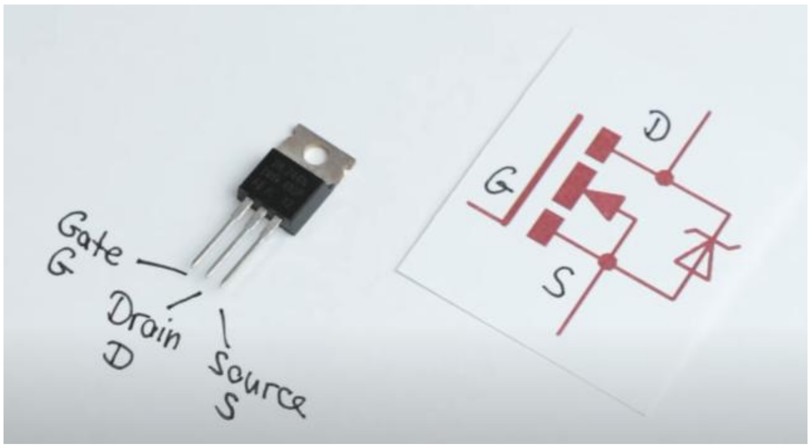
1. **Small Base Current (I_B) → Large Output Current (I_C)**
   - A **tiny current at the base** of Q1 gets **amplified** by its current gain **(β1)**.
   - This amplified current is then used as the **base current for Q2**, which further amplifies it.
   - The final current gain is **$\beta_{total} = \beta1 \times \beta2$**, meaning a **very high overall current gain**.
2. **Voltage Requirement**
   - Since the Darlington pair has two transistors in series, the total **Base-Emitter voltage (V_{BE})** is the sum of both transistors: VBE≈1.2V−1.4VV_{BE} \approx 1.2V - 1.4VVBE ≈1.2V−1.4V instead of the usual **0.7V** for a single BJT.
3. **Higher Saturation Voltage ($V_{CE(sat)}$)**
   - A Darlington transistor has a higher **$V_{CE(sat)}$ (~1V to 2V)** compared to a single transistor (**~0.2V to 0.3V**), leading to **more power dissipation and heat**.

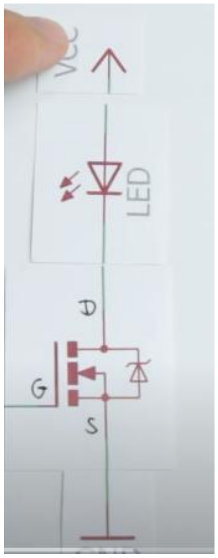
Advantages of Darlington Transistors

✅ **Extremely high current gain (1000+), useful for weak signals**
✅ **Requires very little base current to drive large loads**
✅ **Simple and compact solution for power amplification**

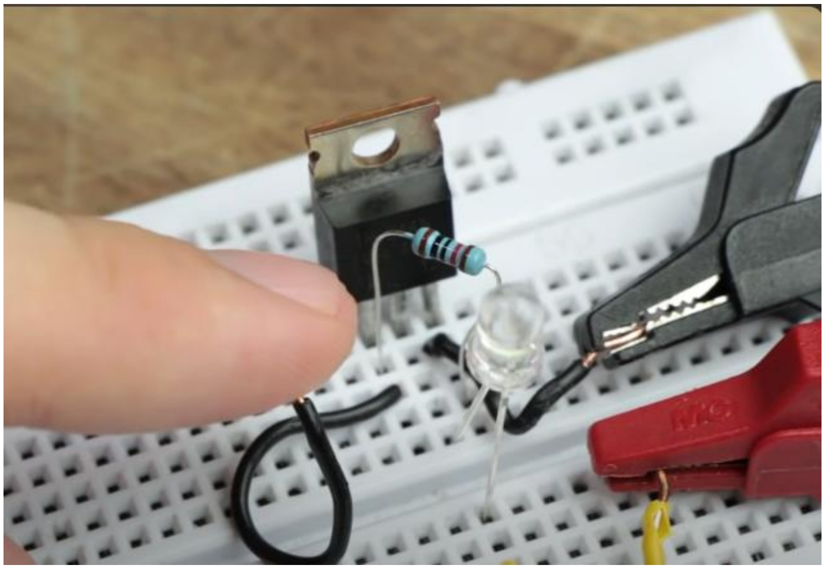
23.Transistor(MOSFET) as a switch



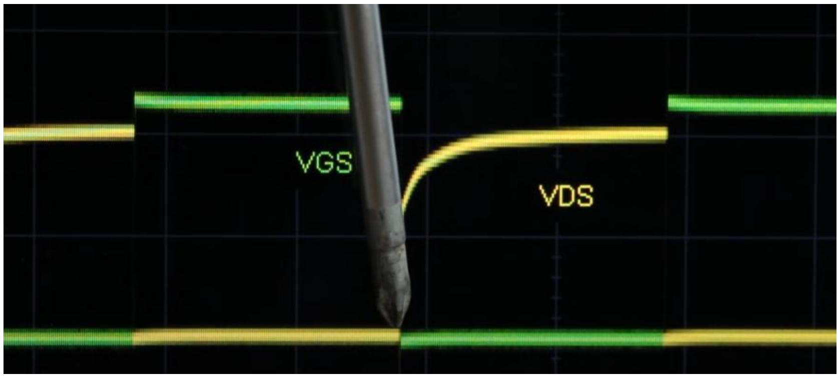




Connecting the LED one problem that was immediately noticeable was that even electrostatic voltages of my body can turn on the loads.Even the bigger loads.To prevent that we connected a 10k ohm pulldown resistor between gate and source.



The region we are working is the linear region in which the resistance of the drain to source path is almost constant.



While the arduino voltage goes high,the drain to source voltage goes low.

We just created an LED dimmer,but lets say you are below it that is tied around.

What is Bootstrap Operation?

Bootstrap operation is a technique used in **power electronics and analog circuits** to generate a **higher voltage** than the supply voltage using a capacitor. It is commonly used in **high-side MOSFET drivers** to switch a **high-voltage MOSFET** in circuits like **DC-DC converters, motor drivers, and Class-D amplifiers**.

1️⃣ Why is Bootstrap Needed?

In circuits like **half-bridge or full-bridge configurations**, the **high-side MOSFET** requires a **gate voltage higher than the source voltage** to turn ON. However, if the MOSFET's **source is**

floating (not connected to ground), we need a way to provide this higher voltage. **Bootstrap operation solves this problem** by using a capacitor to temporarily store energy and provide the required voltage.

---

2 Basic Components of a Bootstrap Circuit

A typical **bootstrap circuit** consists of:

1. **Bootstrap Capacitor (C_B)** → Stores and supplies the gate voltage.
2. **Bootstrap Diode (D_B)** → Prevents backflow of current and charges the capacitor.
3. **High-Side MOSFET (Q_H)** → The transistor that needs a boosted gate voltage.
4. **Low-Side MOSFET (Q_L) or Resistor** → Provides a path to charge the capacitor.
5. **Gate Driver Circuit** → Controls the operation of MOSFETs.

---

3 How Bootstrap Operation Works

🔹 Step 1: Charging the Bootstrap Capacitor

- When the **low-side MOSFET (Q_L) is ON**, the **source of the high-side MOSFET (Q_H) is at ground (0V)**.
- The **bootstrap capacitor (C_B) charges** from the supply voltage (**V_CC**) through the **bootstrap diode (D_B)**.
- The capacitor voltage becomes **V_B ≈ V_CC**.

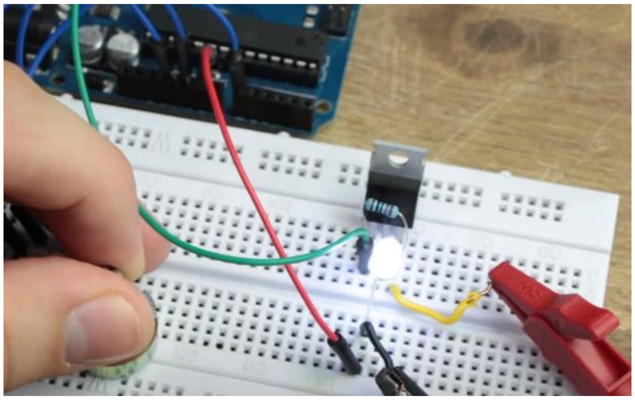
🔹 Step 2: Switching the High-Side MOSFET ON

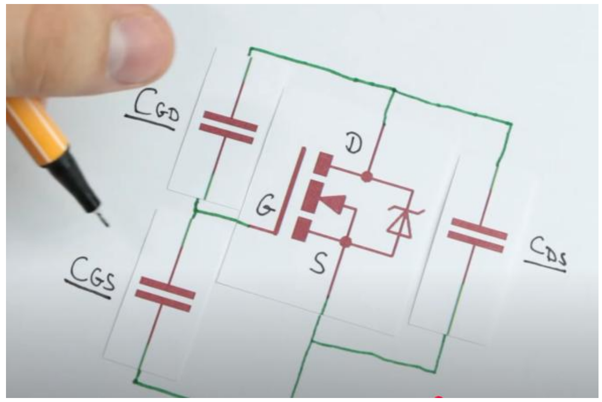
- When we turn ON **Q_H**, its **source voltage rises** to the output voltage (**V_OUT**).
- Since the capacitor is connected between the **gate and source of Q_H**, it provides the needed **higher gate voltage (V_G = V_OUT + V_B)**.
- This ensures that the high-side MOSFET turns ON correctly.

🔹 Step 3: Recharging the Bootstrap Capacitor

- When the **low-side MOSFET turns ON again**, the source of Q_H returns to **0V**, allowing the capacitor to **recharge** through the bootstrap diode.

🔹 This cycle **repeats continuously**, ensuring that the high-side MOSFET gets the necessary gate drive voltage.





24.Stepper Motors and how to use them:

1 What is a Stepper Motor?

A **stepper motor** converts electrical pulses into **precise mechanical movement**. Instead of **spinning continuously** like a regular motor, it moves in **fixed angular steps** when electrical pulses are applied to its coils.

---

2 How a Stepper Motor Works

🔹 Basic Structure

A stepper motor consists of:

1. **Rotor** → The rotating part, made of permanent magnets or soft iron.
2. **Stator** → The stationary part with multiple electromagnetic coils.
3. **Driver Circuit** → Controls the sequence of pulses sent to the coils.

🔹 Working Principle

- The **stator coils** are energized in a specific sequence.
- This creates a **magnetic field** that pulls the **rotor** into alignment.
- By changing the sequence of coil activation, the motor moves **step-by-step**.
- The **number of steps per revolution** determines the motor's **precision**.

Example: A **200-step** stepper motor has a **step angle of 1.8°** (360°/200 = 1.8° per step).

---

3 Types of Stepper Motors

There are **three main types** of stepper motors:

🔹 1. Permanent Magnet Stepper Motor

- **Rotor has permanent magnets**.
- Works on the principle of **attraction and repulsion** between magnetic poles.
- **Advantages:** High torque, cost-effective.
- **Used in:** Printers, small automation devices.

🔹 2. Variable Reluctance Stepper Motor

- **Rotor is made of soft iron** (no magnets).
- Works on the principle of **minimum reluctance** (rotor aligns with energized stator).
- **Advantages:** Fast response, simple construction.
- **Used in:** High-speed applications.

🔹 3. Hybrid Stepper Motor (Most Common)

- **Combination of permanent magnet and variable reluctance motors**.
- **High torque, better precision**.
- **Used in:** CNC machines, robotics, 3D printers.

---

4 Stepper Motor Driving Modes

Stepper motors can be controlled in different modes for **precision and efficiency**:

🔹 1. Full-Step Mode

- Moves **one step per pulse**.
- Example: A **200-step motor moves 1.8° per step**.
- **Advantage:** Maximum torque.

🔹 2. Half-Step Mode

- Alternates between **full steps and intermediate steps**.
- Example: A **200-step motor behaves like a 400-step motor** (0.9° per step).
- **Advantage: Smoother motion, better resolution**.

🔹 3. Microstepping Mode

- Uses **precise current control** to create **smaller steps** (e.g., 1/16th or 1/32nd of a full step).
- **Advantage:** Extremely smooth motion, reduced vibrations.
- **Used in: High-precision applications (CNC, 3D printing, robotics)**.

---

5 Stepper Motor Driver Circuits

To operate a stepper motor, a **driver circuit** is needed to send pulses to the motor. Common types include:

🔹 1. L293D & ULN2003 (Basic Drivers)

- Used for **small stepper motors** (5V–12V).
- Common in **Arduino projects**.

🔹 2. A4988 & DRV8825 (Microstepping Drivers)

- Used in **CNC machines, 3D printers**.
- Supports **microstepping** for smoother motion.

🔹 3. TB6600 (High-Power Driver)

- Controls **high-current stepper motors** (up to 4A).
- Used in **industrial automation**.

---

6 Advantages of Stepper Motors

✅ **Precise Positioning** → Moves in **exact steps**, no feedback required.
✅ **High Torque at Low Speeds** → Good for holding loads.
✅ **Reliable & Long Life** → No brushes, minimal wear and tear.
✅ **Easy Control** → Works with **digital pulses** (microcontrollers, PLCs).

---

7 Disadvantages of Stepper Motors

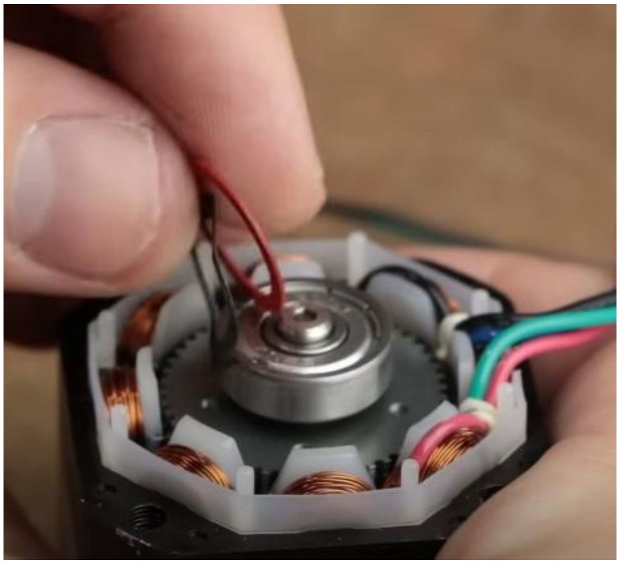
❌ **High Power Consumption** → Draws current even when holding position.
❌ **Limited Speed** → Loses torque at high RPM.
❌ **No Feedback (Open Loop)** → Can lose steps without an encoder.

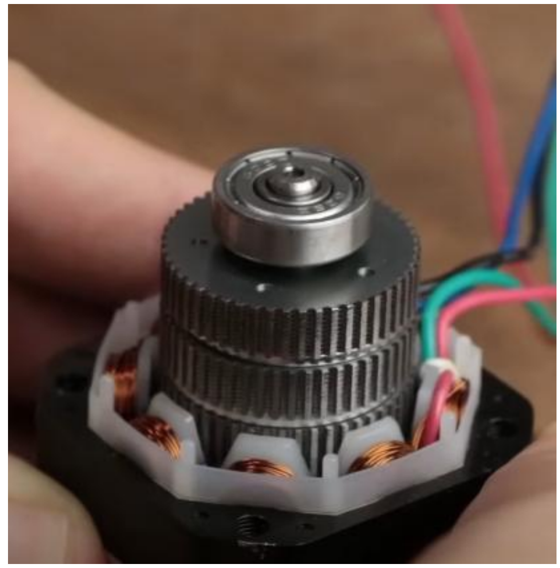
---

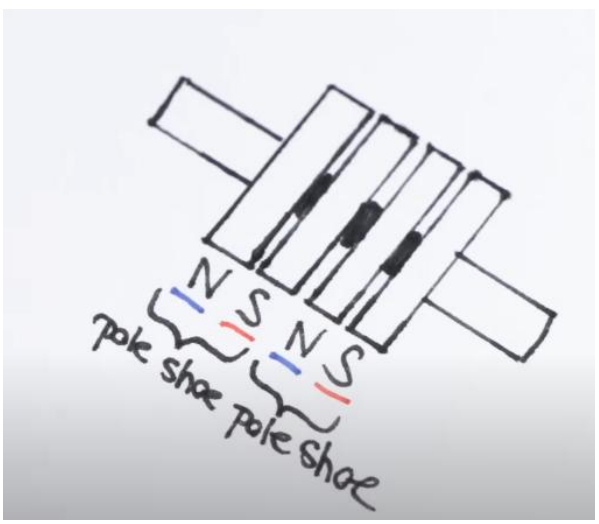
8 Applications of Stepper Motors

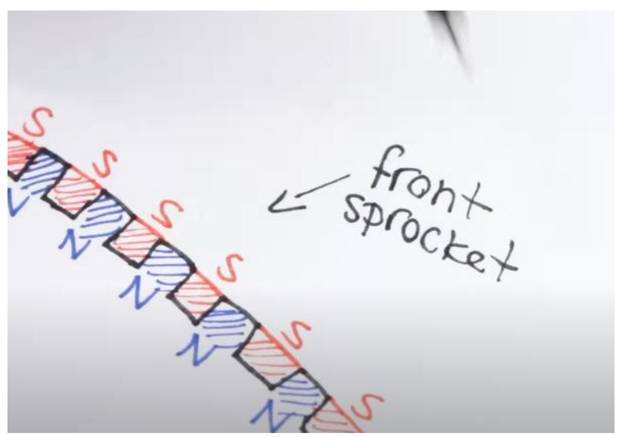
🚀 **3D Printers** → Precise layer movement.
🛠 **CNC Machines** → Accurate cutting and engraving.
□ **Robotics** → Arm movement, automation.

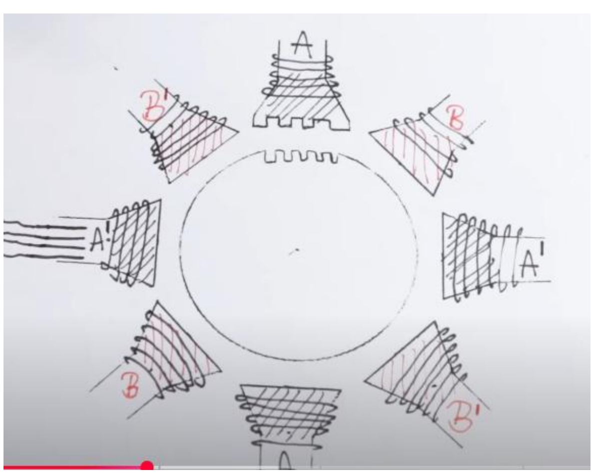
🖨️ **Printers & Scanners** → Paper feeding, head positioning.
⚙️ **Industrial Machines** → Automated assembly lines.



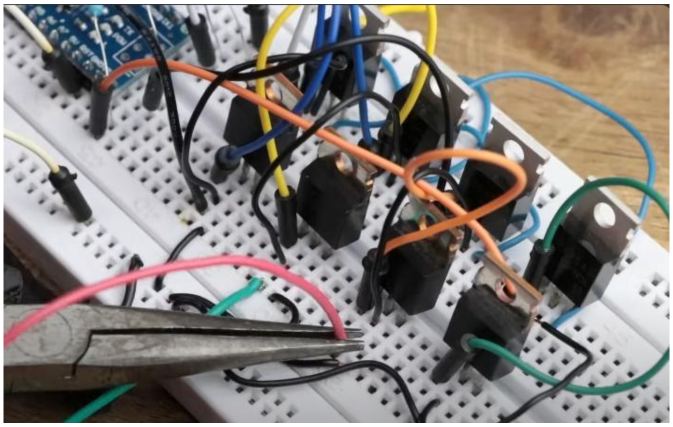








There are 50 theeths per sprocket.



🔹 What is Microstepping?

Microstepping is a technique used to increase the **resolution and smoothness** of a stepper motor's movement by dividing each full step into **smaller steps**. Instead of moving in **fixed full steps** (e.g., 1.8° per step for a 200-step/rev motor), microstepping allows **fractional steps**, such as **half-step, quarter-step, or even 1/16th step**, by precisely controlling the current in the motor coils.

1️⃣ Why is Microstepping Important?

✅ **Smoother Motion** → Reduces vibration and mechanical noise.
✅ **Higher Resolution** → Improves positioning accuracy.
✅ **Reduced Resonance Issues** → Prevents motor skipping or oscillations.
✅ **Better Torque Control** → Allows finer adjustments in motion.

2️⃣ How Microstepping Works

In a standard **full-step mode**, one coil is fully energized at a time.
In **microstepping**, the current in the coils is adjusted gradually to create **intermediate positions** between steps.

🔹 **Example:**
A **200-step motor (1.8° per step)** can be microstepped to:

- **Half-step mode (1/2 step):** 400 steps/rev (0.9° per step)
- **Quarter-step mode (1/4 step):** 800 steps/rev (0.45° per step)
- **1/16th step mode:** 3200 steps/rev (0.1125° per step)

This is achieved by **controlling the coil current** with a **sine-wave approximation** instead of simple on/off pulses.

---

3 Current Waveform in Microstepping

Instead of switching coils **fully ON/OFF**, microstepping **modulates the current** through **PWM (Pulse Width Modulation)**, creating a **smooth transition** between steps.

- **Full Step Mode:** 100% current in one coil at a time.
- **Half Step Mode:** 100% and 50% current alternately.
- **Microstepping:** Uses sine-wave-like control (e.g., 71%, 38%, etc.).

🔹 Example of Coil Current at 1/4 Microstep Mode

Step Position	Coil A Current	Coil B Current
Step 0 (0°)	100%	0%
Step 1 (1/4 step)	71%	71%
Step 2 (1/2 step)	0%	100%
Step 3 (3/4 step)	-71%	71%
Step 4 (Full step, 90°)	-100%	0%

🔹 This current waveform **smooths the transition** and **reduces vibration**.

---

4 Microstepping Driver Circuits

To implement microstepping, **special driver ICs** are needed. These drivers generate the required **PWM signals** to control the coil currents.

🔹 Common Microstepping Drivers

1. **A4988 (1/16 Step)** → Used in 3D printers, CNC machines.
2. **DRV8825 (1/32 Step)** → Higher current, better precision.
3. **TB6600 (1/64 Step)** → Used for high-power applications.

---

5 Advantages & Disadvantages of Microstepping

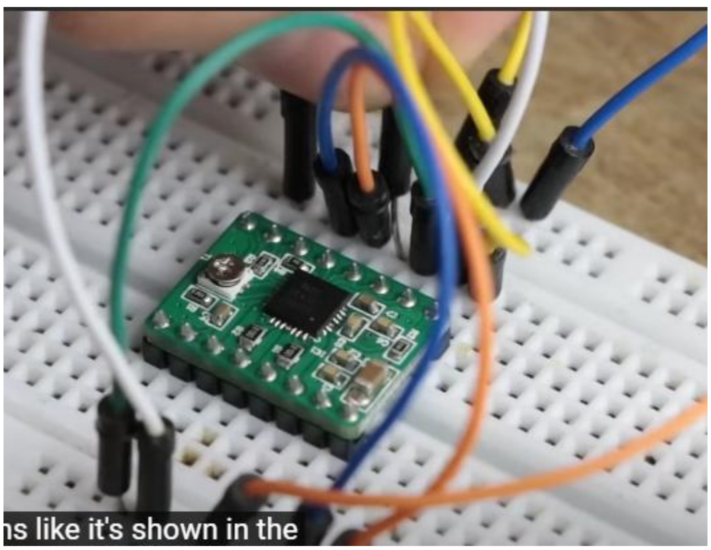
✅ Advantages:

✔ **Smoother movement** with reduced noise.
✔ **Higher positioning resolution**.
✔ **Reduced resonance & vibrations**.

❌ Disadvantages:

❌ **Lower torque at higher microsteps** (Current is divided across steps).
❌ **Requires complex drivers**.
❌ **Diminishing returns beyond 1/16 step** (Limited accuracy improvement).

But in order to implement micro stepping,we don't want to use a constant voltage applied to our coils.Instead we need a constant current which we can vary in its strength to create the different steps.An easy solution to this problem is **A4988 microsteping IC.**



The **A4988** is a **popular stepper motor driver IC** that supports **microstepping**, allowing precise control of stepper motors. It is widely used in **3D printers, CNC machines, and robotics**.

---

1 Key Features of A4988

✅ **Supports Full, 1/2, 1/4, 1/8, and 1/16 Microstepping**
✅ **Operates from 8V to 35V supply voltage**
✅ **Can handle up to 2A per coil (with proper cooling)**
✅ **Built-in overcurrent, thermal, and short-circuit protection**
✅ **Simple Step and Direction control**

---

2 A4988 Pinout and Connections

Pin	Function
VDD	Logic power (3.3V–5V)
GND	Ground
VMOT	Motor power (8V–35V)
2B, 2A, 1A, 1B	Connect to stepper motor coils
STEP	Pulse input to move one step
DIR	Controls motor direction
MS1, MS2, MS3	Set microstepping mode
ENABLE	Active LOW to enable driver
SLEEP	Active LOW to put the driver in sleep mode
RESET	Resets the driver when LOW

---

3 Microstepping Modes

The **MS1, MS2, and MS3** pins control the **microstepping resolution**:

MS1	MS2	MS3	Microstep Mode	Steps per Revolution (for 200-step motor)
LOW	LOW	LOW	Full Step	200 steps (1.8° per step)
HIGH	LOW	LOW	1/2 Step	400 steps (0.9° per step)
LOW	HIGH	LOW	1/4 Step	800 steps (0.45° per step)

MS1	MS2	MS3	Microstep Mode	Steps per Revolution (for 200-step motor)
HIGH	HIGH	LOW	1/8 Step	1600 steps (0.225° per step)
HIGH	HIGH	HIGH	1/16 Step	3200 steps (0.1125° per step)

◆ **If MS1, MS2, and MS3 are left unconnected, the default mode is Full Step.**

---

4 Wiring the A4988 with Arduino

🔹 Circuit Connections

Power Connections:

- **VMOT** → **12V or 24V** (Stepper motor power)
- **VDD** → **5V** (Logic power)
- **GND** → **Common ground**

Stepper Motor Connections:

- **1A & 1B** → First coil of stepper motor
- **2A & 2B** → Second coil of stepper motor

Control Pins (Connected to Arduino):

A4988 Pin	Arduino Pin
STEP	Digital Pin 9
DIR	Digital Pin 8
ENABLE	GND (Always enabled)
MS1, MS2, MS3	HIGH / LOW as per microstepping mode

6 Adjusting Current Limit on A4988

To prevent overheating, set the **current limit** using the **VREF** voltage on the **A4988**.

Formula:

Formula:

$$I_{max} = \frac{VREF}{8 \times R_s}$$

Where **R_s** is the sense resistor (typically **0.1Ω** or **0.05Ω**).

- **For 1A current limit:** Set **VREF = 0.8V**
- **For 1.5A current limit:** Set **VREF = 1.2V**

7️⃣ Common Issues and Solutions

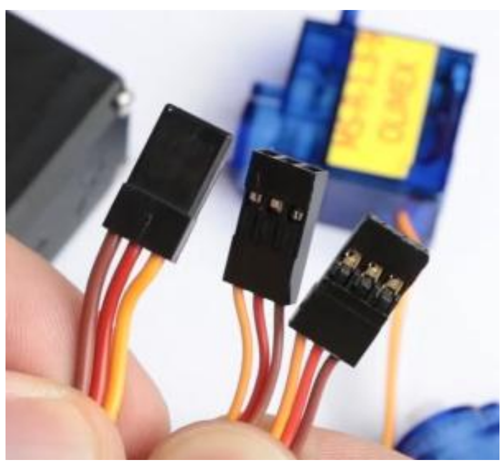
Issue	Possible Cause	Solution
Stepper motor not moving	Wrong wiring	Check coil connections
Motor vibrating but not rotating	Low current limit	Adjust VREF
Overheating driver	High current setting	Use a heatsink
Skipping steps	Low power supply	Use a stable **12V or 24V** supply

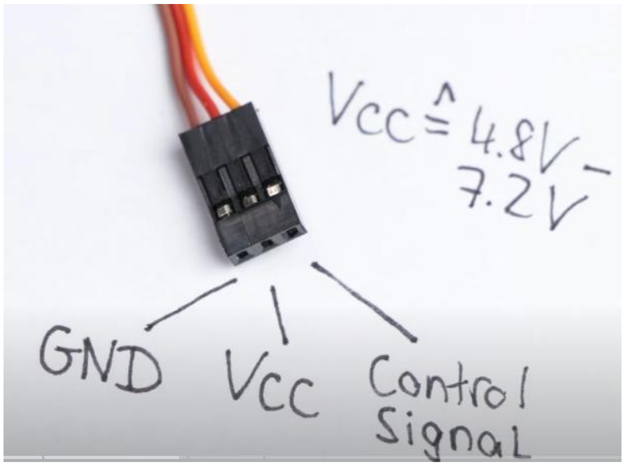
---

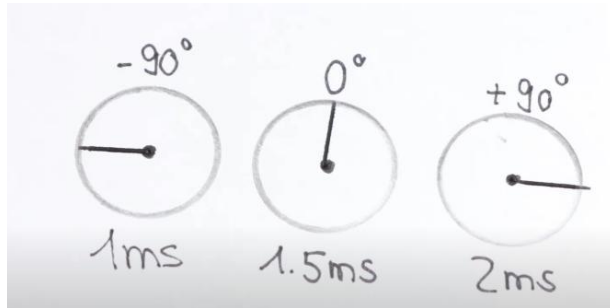
8️⃣ Applications of A4988

🚀 **3D Printers** (e.g., Ender 3, Prusa i3)
🛠 **CNC Machines**
□ **Robotics & Automation**
🖨 **Plotters & Engraving Machines**

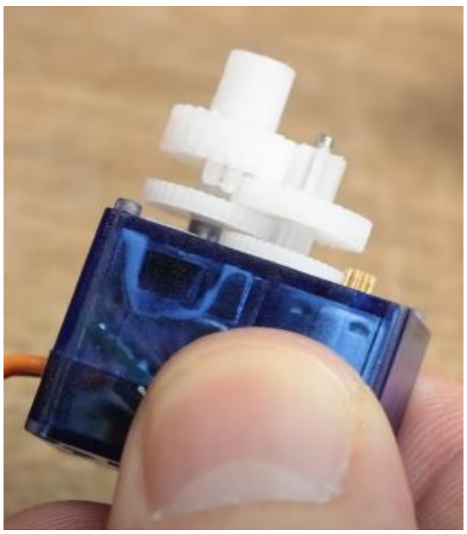
25.Servos and how to use them





So we can rotate the shaft a total of 180^0.



The keys on the top decrease the original RPM of the motor.
Underneath one gear height,the shaft of a potentiometer that is located inside the servo.This potentiometer creates a voltage divider that outputs a variable voltage according to the position of the motor shaft and acts as a feedback for the utilized control IC KC5188.

A **servo motor** is a **precise, position-controlled motor** commonly used in **robotics, automation, RC vehicles, and industrial machinery**. Unlike regular DC motors, servo motors allow **precise control of position, speed, and torque** using **feedback mechanisms**.

1️⃣ Key Features of Servo Motors

✅ **Precise Position Control** → Moves to an exact angle.
✅ **High Torque at Low Speeds** → Good for load handling.

✅ **Closed-Loop Control** → Uses **feedback** for accuracy.
✅ **Available in Different Sizes** → Mini servo, standard servo, industrial servo.

2️⃣ Types of Servo Motors

🔹 1. Positional Rotation Servo

✔ Moves **within a fixed range** (e.g., **0° to 180°**).
✔ Used in **robot arms, camera gimbals, and RC cars**.

🔹 2. Continuous Rotation Servo

✔ Rotates **freely in both directions**, like a DC motor.
✔ Used in **wheeled robots and conveyor belts**.

🔹 3. Linear Servo

✔ Converts rotary motion into **linear movement**.
✔ Used in **actuators, robotic arms, and CNC machines**.

3️⃣ How Does a Servo Motor Work?

A **servo motor system** consists of:
1️⃣ **Motor** (DC or AC motor inside).
2️⃣ **Control Circuit** (Processes input signals).
3️⃣ **Feedback System** (Potentiometer or encoder).

📌 **Working Principle:**

- The **controller sends a PWM signal** to the servo.
- The **servo moves to the desired position** based on this signal.
- A **feedback sensor (potentiometer or encoder)** ensures precise positioning.

4️⃣ PWM Control of a Servo Motor

🔹 A **servo motor is controlled using a PWM (Pulse Width Modulation) signal.**

- **Standard PWM Frequency**: **50 Hz (20ms period)**
- **Pulse Width Controls the Position:**

- **1ms pulse** → 0° position
- **1.5ms pulse** → 90° (center)
- **2ms pulse** → 180° position

📌 **Example:** If a servo receives a **1.5ms pulse every 20ms**, it stays at **90°**.

---

5 Wiring a Servo Motor to Arduino

Servo Pin	Arduino Pin	Function
VCC (Red)	**5V**	Power
GND (Black/Brown)	**GND**	Ground
Signal (Yellow/Orange)	**PWM Pin (e.g., D9)**	Control Signal

7 Advantages & Disadvantages of Servo Motors

✅ Advantages

✔ **High precision & accuracy**
✔ **Low power consumption**
✔ **Fast response time**
✔ **Works well under load**

❌ Disadvantages

❌ **Limited rotation (for standard servos)**
❌ **Expensive compared to DC motors**
❌ **Requires PWM signals for control**

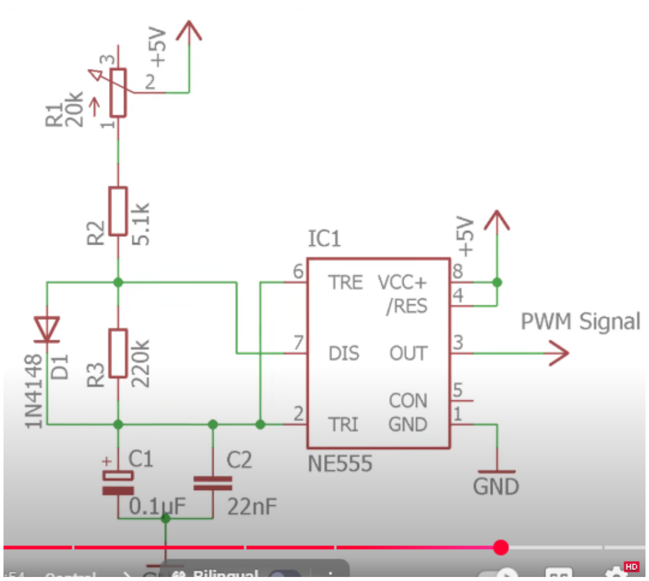
---

8 Applications of Servo Motors

🚗 **RC Cars & Drones** → Steering and movement control.
□ **Robotics** → Robotic arms and humanoid robots.
🎥 **Cameras & Gimbals** → Precise positioning.
🏭 **CNC Machines** → High-precision cutting.
🎮 **Automated Systems** → Factory automation, medical equipment.

🔹 Conclusion

Servo motors are **powerful, precise, and versatile** motors used in **robotics, automation, and industrial applications**. They work using **PWM control and feedback systems**, allowing **accurate motion control**.



If you want to use the servo motor as a normal motor that rotates 360^0,simply remove the mechanical which is usually attached to one gear,remove the feedback potentiometer and replace it with two 10k ohm resistors to create a voltage divider.

26.555 Timer IC

The **555 Timer IC** is one of the most widely used integrated circuits in electronics. It is a versatile chip that can function as a **timer, pulse generator, or oscillator** in various applications.

---

1. Overview of 555 Timer IC

- **Introduced**: 1972 by Signetics (now owned by ON Semiconductor)
- **Type**: Monolithic timing circuit
- **Package**: Available in DIP, SOIC, and other packages
- **Voltage Range**: 4.5V to 15V (typically operates at 5V or 12V)
- **Current Output**: Can source or sink up to **200mA**

---

2. Pin Configuration of 555 Timer IC

A standard **8-pin** Dual-Inline Package (DIP) layout:

Pin	Name	Function
1	**GND**	Ground (0V)
2	**Trigger**	Activates timing cycle (low signal starts output)
3	**Output**	Provides the output waveform
4	**Reset**	Resets the timing cycle (active low)
5	**Control Voltage**	Alters threshold voltage (usually connected to a capacitor)
6	**Threshold**	Determines when to reset the timing cycle
7	**Discharge**	Connected to timing capacitor to discharge it
8	**VCC**	Power supply voltage (4.5V to 15V)

---

3. Operating Modes of 555 Timer IC

The 555 Timer operates in three modes:

A. Monostable Mode (One-shot Mode)

- Produces a single output pulse when triggered.
- Used in **timing applications** such as delay circuits, pulse width modulation (PWM), and debouncing switches.

📌 Formula for Pulse Width:

T=1.1×R×CT = 1.1 \times R \times CT=1.1×R×C

where **T** is the pulse duration, **R** is the resistor in ohms, and **C** is the capacitor in farads.

B. Astable Mode (Oscillator Mode)

- Generates a continuous square wave.
- Used in **flashing LEDs, clock pulses, tone generation**, etc.

📌 **Frequency and Duty Cycle Formulas:**

f=1.44(R1+2R2)Cf = \frac{1.44}{(R_1 + 2R_2)C}f=(R1+2R2)C1.44 D=R2(R1+2R2)D = \frac{R_2}{(R_1 + 2R_2)}D=(R1+2R2)R2

where **R1, R2** are resistors, and **C** is a capacitor.

C. Bistable Mode (Flip-Flop Mode)

- Works as a **flip-flop**, switching between **two stable states**.
- Used in **push-button circuits, memory storage, and toggle switches**.

---

4. Applications of 555 Timer IC

The 555 Timer is used in a wide range of applications, including:

✅ **Timers** (delays, time-based control circuits)
✅ **Oscillators** (square wave generators, tone generators)
✅ **PWM circuits** (motor speed control, LED dimming)
✅ **Frequency dividers**
✅ **Sequential circuits**
✅ **Alarm systems**
✅ **Pulse generation circuits**

---

5. Advantages and Disadvantages

✔ Advantages

- Simple and easy to use.
- Operates in both analog and digital circuits.
- **Low cost** and **widely available**.
- Works with a **wide voltage range (4.5V to 15V)**.
- Can provide relatively high current output (**up to 200mA**).

✖ Disadvantages

- **Limited frequency range** (not suitable for high-frequency applications).
- **Power consumption** can be high in some cases.
- **Accuracy is affected** by temperature variations and component tolerances.

6. Variants of the 555 Timer IC

There are different versions of the 555 Timer:

Version	Description
NE555	Standard bipolar 555 Timer
LM555	Low-power version
CMOS 555 (ICM7555, TLC555)	Low power, low current consumption
Dual 555 Timer (556 IC)	Contains two 555 timers in one package
Quad 555 Timer (558 IC)	Contains four 555 timers in one package

7. Example Circuit: Blinking LED using 555 Timer (Astable Mode)

Components Required:

- 555 Timer IC
- Resistors: **1kΩ, 10kΩ**
- Capacitor: **10µF**
- LED
- Power supply (5V to 12V)

Circuit Connections:

- **Pin 1** → Ground (GND)
- **Pin 2** → Connect to Pin 6 (Threshold)
- **Pin 3** → LED (through a **330Ω** resistor)
- **Pin 4** → Connect to **VCC** (to disable reset)
- **Pin 5** → No connection (or add a 10nF capacitor for stability)
- **Pin 6** → Connect to Pin 2
- **Pin 7** → Connect to **VCC via 10kΩ resistor**, and also to Pin 6 via **1kΩ resistor**
- **Pin 8** → VCC (5V-12V)
- **Between Pin 6 and GND**: **10µF capacitor**

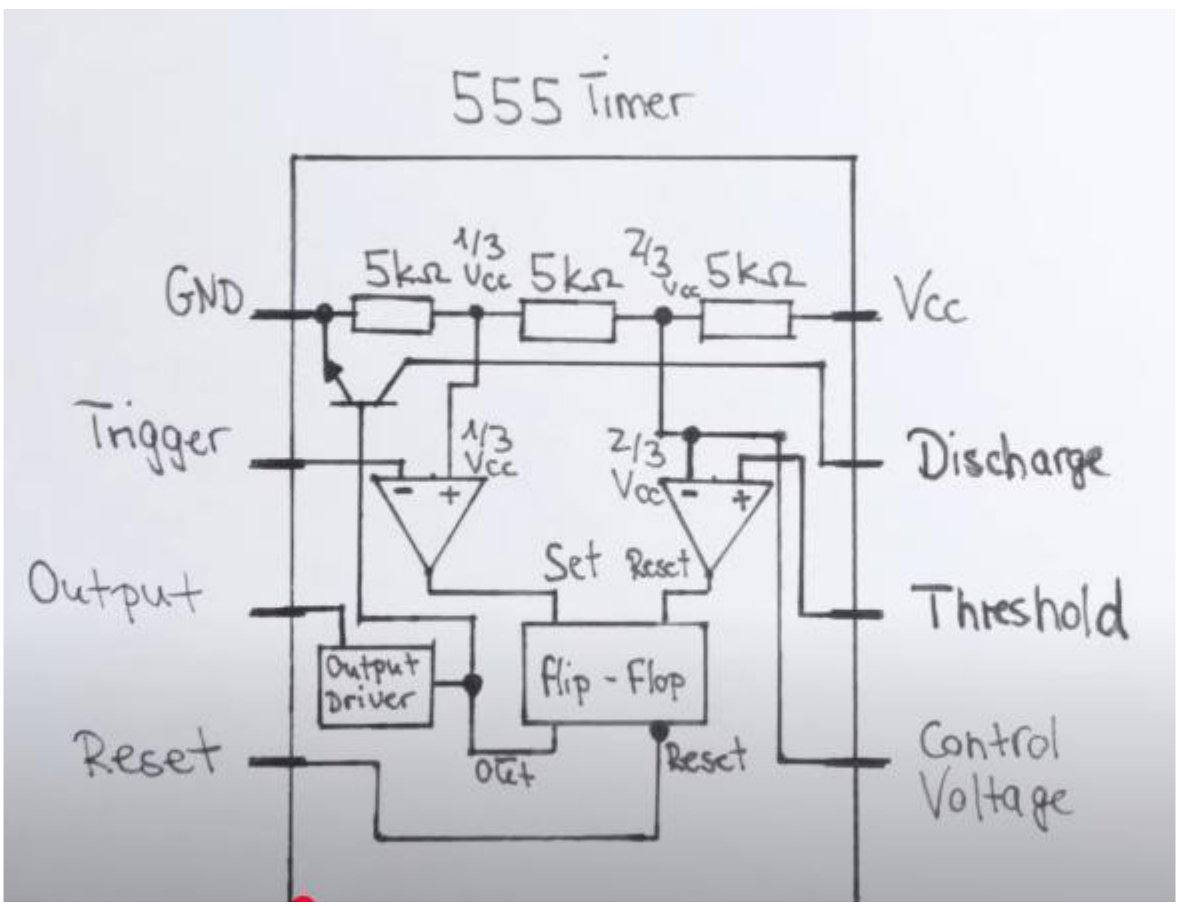
📌 **Working**: The LED will blink at a frequency determined by resistors and the capacitor.

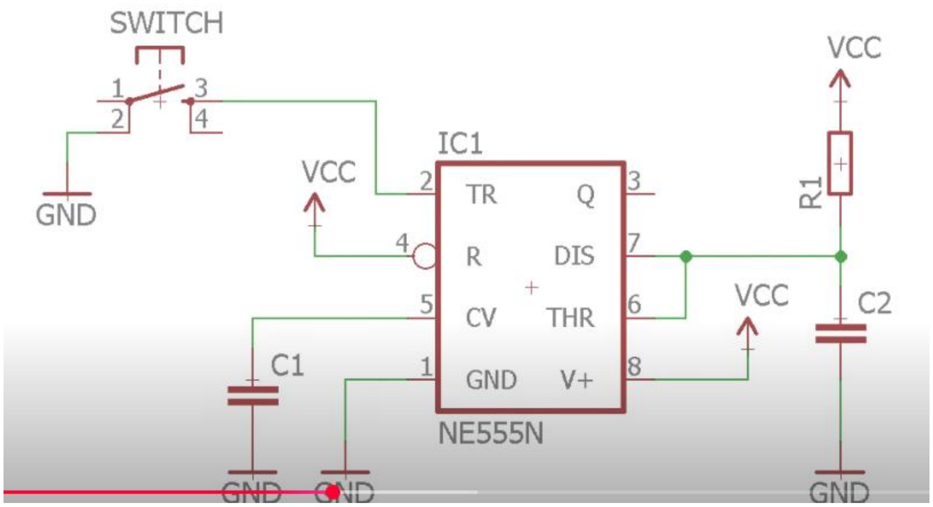
8. Summary

- **The 555 Timer IC is a versatile chip** used for generating pulses, delays, and oscillations.

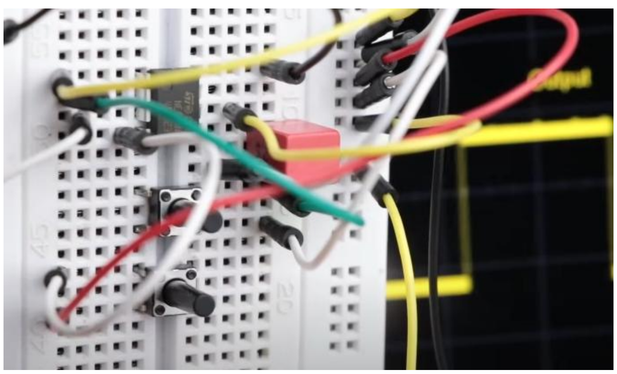
- **It operates in three modes:** Monostable (one-shot), Astable (continuous pulses), and Bistable (flip-flop).
- **Common applications** include clocks, alarms, motor speed control, LED flashing, and frequency generation.
- **It is available in different variants**, including **CMOS versions for low-power applications**.

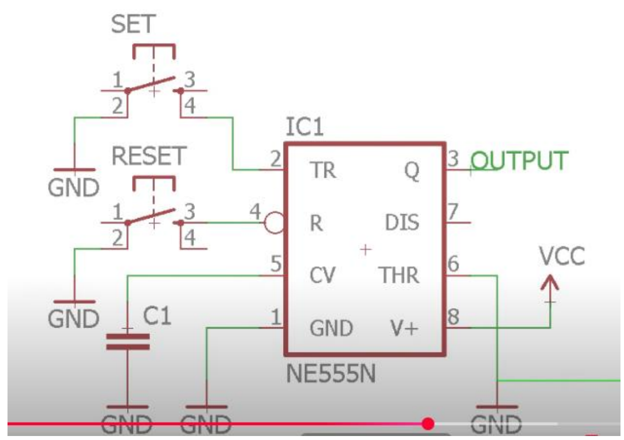
Inside the IC,
Pin 1 and 8 are connected through three 5k ohm resistors in series.
Pin 2 is the trigger pin which directly connects to the negative input of a comparator.The positive input of this comparator connects to the voltage divider which has a potential of one thirt of the supply voltage at this point and the out of the comparator is connected to the set pin of the integrated flip-flop.
Pin 3 is output which is connected to the output driver subsequently to the output of the flipflop.
Pin 4 is the reset pin which directly connects to the reset pin of the flip-flop and can by connecting it to the ground,reset the flip-flop instantly.That is why they are usually tied top the supply voltage.
Pin 5 is the control voltage which not only connecs to the negative input of the second comparator but also the voltage dividers to its potential of the supply voltage.
Pin 6 is the threshold pin which connects to the positive input of the second comparator,whose output connects to the reset pin of the flip-flop.
Last but not least,we got the discharge pin which directly connects to the collector of a bipolar junction transistor.Its emitter connects to the ground and base connects to the output of the flip-flop.



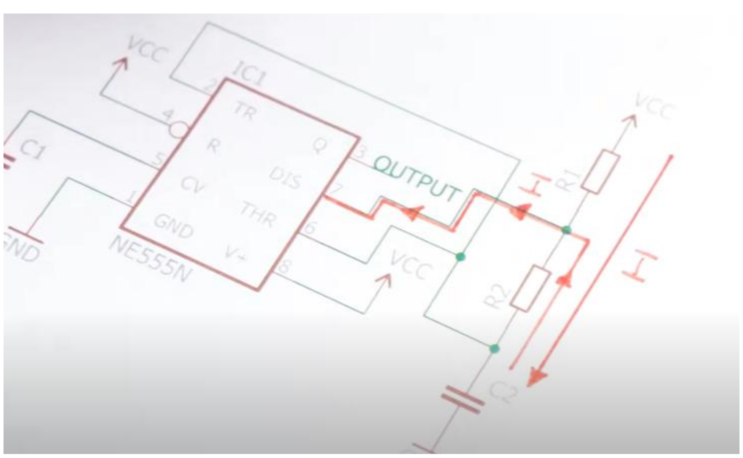


This is a **monostable multivibrator configuration**.This configuration is called monostable because only the low output is stable.

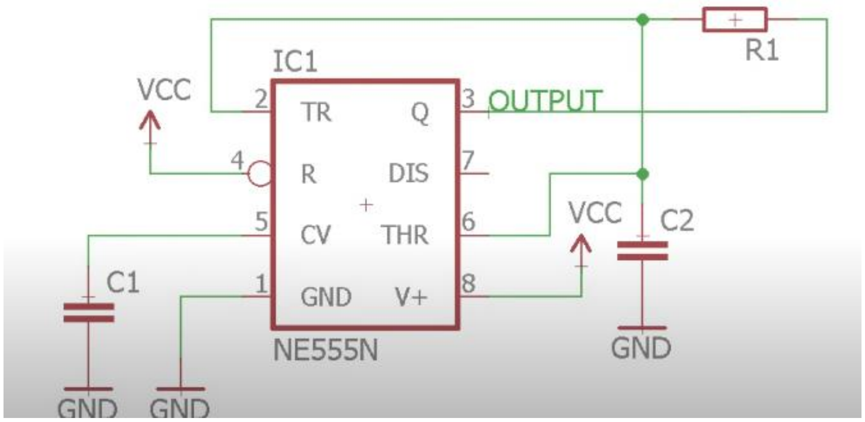




This is a multistable multivibrator configuratuion.



Stable multivibrator.



27.ADC(Analog to Digital Converter)

An **Analog-to-Digital Converter (ADC)** is an electronic circuit that converts **analog signals (continuous values) into digital signals (discrete values)**. ADCs are essential for interfacing real-world signals (temperature, sound, voltage, etc.) with digital devices like microcontrollers, computers, and digital signal processors (DSPs).

---

1. Why Do We Need ADC?

Most real-world signals are **analog**, meaning they vary continuously over time. However, digital systems (like microcontrollers and computers) work with **discrete binary values (0s and 1s)**.

🔹 Example: A microphone captures sound as an analog waveform, but a computer needs a digital representation of that sound for processing or storage.

📌 **Solution** → ADC converts the **continuous analog signal** into a **discrete digital signal** that can be processed by digital circuits.

---

2. ADC Block Diagram & Working Principle

Basic Steps in ADC Conversion:

1. **Sampling** – Takes discrete samples of the analog signal at regular time intervals.
2. **Quantization** – Assigns discrete digital values to each sampled signal level.
3. **Encoding** – Converts the quantized values into binary format for digital processing.

Block Diagram:

```
mathematica
CopyEdit
```

```
Analog Input → Sample & Hold → Quantization → Encoding → Digital Output
```

---

3. ADC Characteristics

A. Resolution (n-bit ADC)

- **Defines the number of discrete levels an ADC can output.**
- Given by: 2n2^n2n where **n** is the number of bits.
- Example:
    - **8-bit ADC** → 256 levels (0 to 255).
    - **10-bit ADC** → 1024 levels (0 to 1023).
    - **12-bit ADC** → 4096 levels.

📌 **Higher resolution = more precise conversion.**

B. Sampling Rate (Sampling Frequency)

- The number of samples taken per second (**Hz or samples per second**).
- Governed by **Nyquist Theorem**, which states: fs≥2fmf_s \geq 2f_mfs≥2fm where **fs** is the sampling frequency and **fm** is the highest frequency in the signal.
- **Example:** If the highest frequency of a signal is **5 kHz**, the sampling rate should be **≥ 10 kHz**.

C. Quantization Error

- **The difference between the actual analog signal and its digital representation.**
- Higher bit resolution reduces quantization error.

📌 **More bits = Less quantization error = More accuracy.**

D. Signal-to-Noise Ratio (SNR)

- Measures how well the ADC converts the signal without noise.
- Given by: SNR=6.02n+1.76 dBSNR = 6.02n + 1.76 \text{ dB}SNR=6.02n+1.76 dB where **n** is the resolution in bits.
- Higher resolution = better SNR.

---

4. Types of ADC

There are different types of ADCs based on speed, accuracy, and power consumption.

A. Flash ADC (Parallel ADC)

✓ **Fastest** ADC (used in high-speed applications).
✓ Uses **2^n-1 comparators** to compare the input voltage with reference voltages.
✓ Used in **oscilloscopes, radar, and high-speed communication**.
❌ **Expensive** and **requires more power** due to many comparators.

B. Successive Approximation Register (SAR) ADC

✓ **Most common ADC** in microcontrollers.
✓ Uses a **binary search** algorithm to approximate the input signal.
✓ **Medium speed & high accuracy** (used in data acquisition, industrial control).
❌ Slower than Flash ADC.

C. Delta-Sigma (ΣΔ) ADC

✓ **Very high accuracy** (used in audio processing and instrumentation).
✓ Uses oversampling and noise shaping techniques.
❌ **Slow conversion speed** (not suitable for high-speed applications).

D. Dual Slope ADC

✓ **High accuracy, low noise** (used in digital multimeters).
✓ Integrates the input signal over time for precision.
❌ **Slow conversion speed**.

E. Pipeline ADC

✓ **Faster than SAR, but slower than Flash ADC**.
✓ Used in **video, communications, and medical imaging**.
❌ **More complex** than SAR ADC.

---

5. ADC Applications

ADC is widely used in various fields:

✅ **Microcontrollers & Embedded Systems** (Arduino, STM32, Raspberry Pi)
✅ **Digital Signal Processing (DSP)** (Audio & image processing)
✅ **Medical Equipment** (ECG, EEG, MRI)
✅ **Communication Systems** (Software-defined radio, RF signal processing)
✅ **Industrial Automation** (Temperature sensors, pressure monitoring)
✅ **Instrumentation** (Digital oscilloscopes, voltmeters)

7. ADC vs DAC (Digital-to-Analog Converter)

Feature	ADC (Analog to Digital)	DAC (Digital to Analog)
Input	Analog Signal	Digital Signal (Binary)
Output	Digital Data	Analog Signal
Purpose	Converts real-world signals for digital processing	Converts digital signals back to analog
Example	Microphone to Computer	Audio Player (MP3 to Speaker)

📌 **ADC → Converts Sound to Digital for Processing**
📌 **DAC → Converts Digital Audio Back to Sound (Speaker Output)**

---

8. Summary

- **ADC converts analog signals to digital format** for use in computers and microcontrollers.
- **Key parameters**: Resolution (bits), Sampling Rate, Quantization Error, SNR.
- **Types of ADCs**: Flash (fastest), SAR (most common), Sigma-Delta (high precision), Dual Slope (accurate but slow).
- **Used in** microcontrollers, audio processing, medical devices, and industrial automation.

1. What is Sampling Rate?

The **sampling rate (sampling frequency)** is the number of times an analog signal is measured (sampled) per second during **Analog-to-Digital Conversion (ADC)**. It is measured in **Hertz (Hz) or Samples per Second (SPS)**.

📌 **Example:**

- **44.1 kHz sampling rate** → 44,100 samples taken per second (used in audio processing).
- **1 MHz sampling rate** → 1 million samples per second (used in high-speed oscilloscopes).

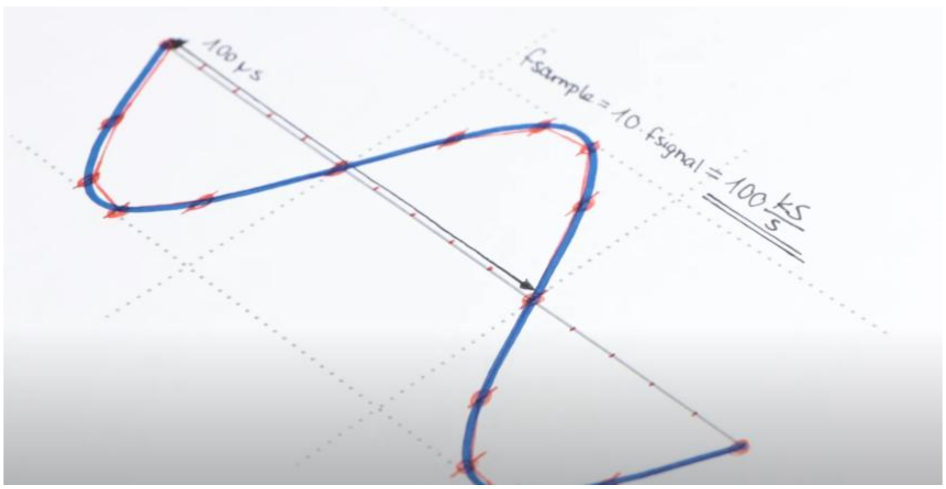
2. Why is Sampling Rate Important?

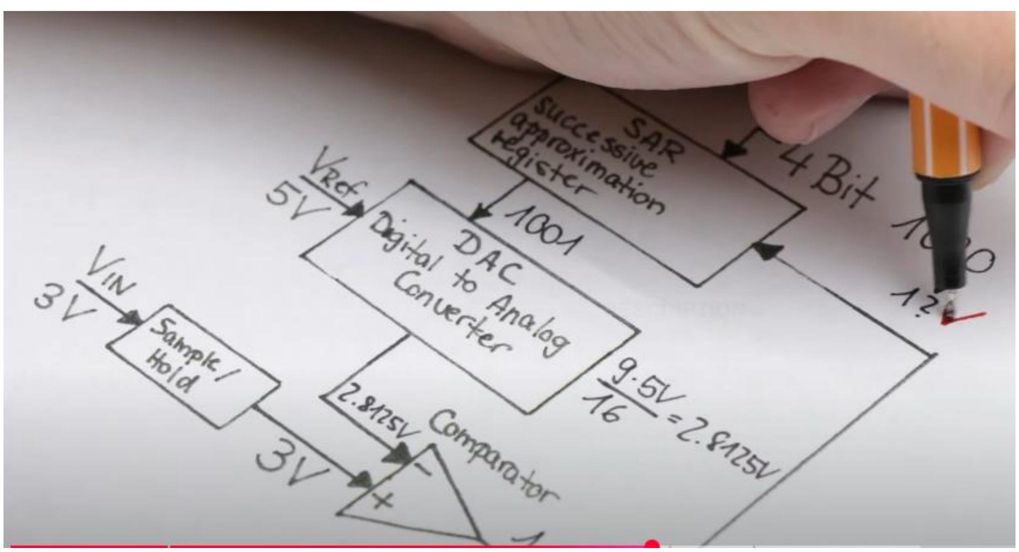
The sampling rate determines how accurately an analog signal can be represented in digital form.

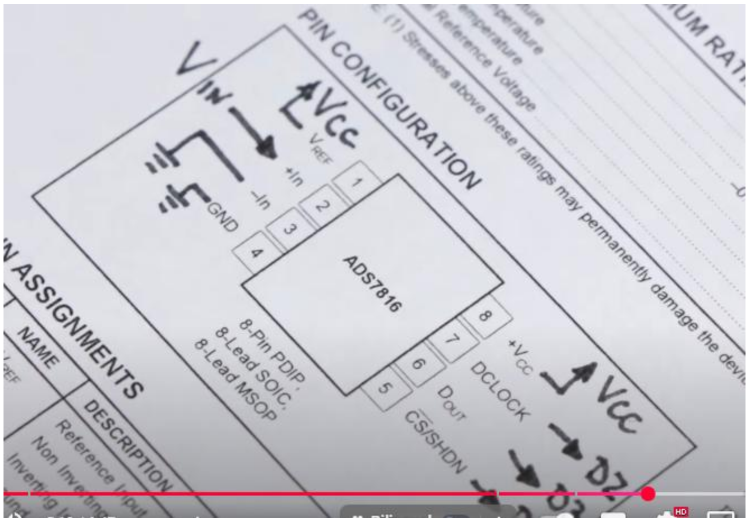
🔹 **Higher sampling rate** → More accurate signal representation but larger data size.
🔹 **Lower sampling rate** → Less accurate but saves memory and processing power.

📌 **Applications:**

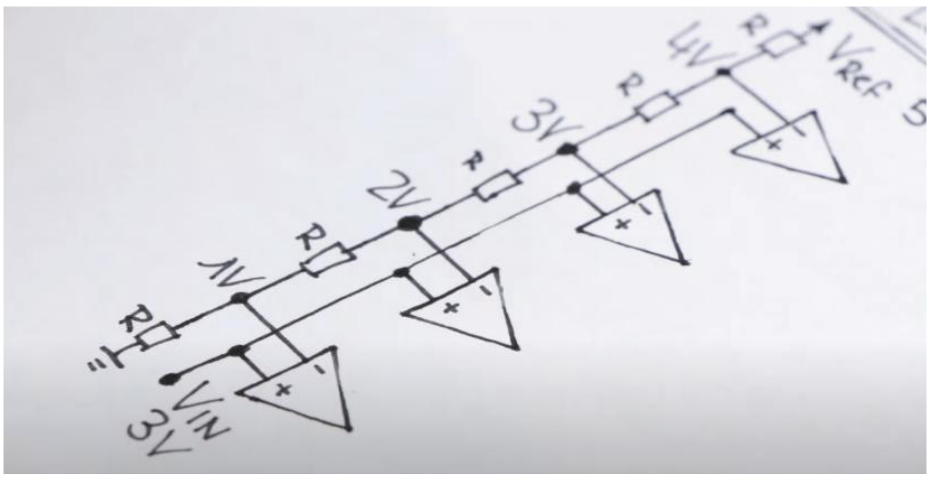
- **Audio recording (44.1 kHz, 48 kHz, 96 kHz)**
- **Medical imaging (ECG, EEG sampling at 1 kHz - 10 kHz)**
- **Radar & communication systems (MHz - GHz sampling rates)**



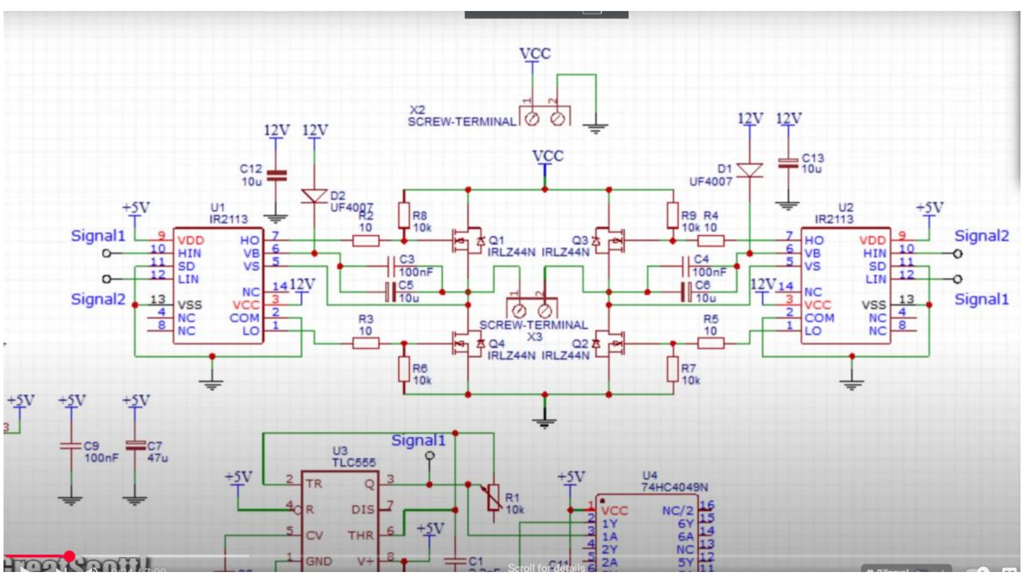




Pinouts of **ADS7816.**



28.IGBT and when to use them



The **Insulated Gate Bipolar Transistor (IGBT)** is a **power semiconductor device** that combines the advantages of **Bipolar Junction Transistors (BJTs)** and **Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs)**. It is widely used in **high-power applications** such as motor drives, inverters, and power electronics.

---

1. What is an IGBT?

An **IGBT (Insulated Gate Bipolar Transistor)** is a **three-terminal power semiconductor device** that is used as a switch in power electronics.

✅ **Combines the best of two technologies:**

- **MOSFET**: High-speed switching, high input impedance.
- **BJT**: Low conduction losses, high current-handling capability.

📌 **Advantages:** High efficiency, fast switching, and ability to handle high voltages and currents.

---

2. IGBT Symbol & Structure

IGBT Symbol

The IGBT has **three terminals**:

1. **Gate (G)** – Controls the switching operation.
2. **Collector (C)** – Input terminal for current.
3. **Emitter (E)** – Output terminal for current.

IGBT Internal Structure

IGBT consists of **four layers (PNPN)**, forming a structure similar to a **thyristor**, but with a **MOSFET-controlled gate**.

🔹 **Layers:**

- **Emitter (N+ region)**
- **Gate (MOSFET structure)**
- **Collector (P+ region)**

---

3. Working Principle of IGBT

The IGBT operates in **two modes**:

- **OFF State (Cutoff Mode)**: When the **Gate-Emitter voltage (V_GE) = 0V**, the IGBT is **OFF**, and no current flows from Collector to Emitter.
- **ON State (Saturation Mode)**: When **V_GE > threshold voltage (V_TH)**, the MOSFET section turns ON, allowing current to flow from Collector to Emitter.

📌 **IGBT is a voltage-controlled device** (requires voltage, not current, at the gate).

---

4. IGBT Characteristics

A. Forward Characteristics (I-V Curve)

- **Low conduction loss** due to bipolar operation.
- **High input impedance** like MOSFET.

B. Switching Characteristics

- **Turn-On Time (Ton)** → Faster than BJT but slower than MOSFET.
- **Turn-Off Time (Toff)** → Slightly slower due to tail current effect.

C. Voltage & Current Ratings

- **Voltage ratings:** 600V, 1200V, 1700V, 3300V, 6500V.
- **Current ratings:** From a few **amperes (A) to kiloamperes (kA)**.

📌 **Example:** A 1200V, 100A IGBT is commonly used in industrial motor drives.

---

5. Types of IGBT

IGBTs are classified based on their **gate structure and performance**.

A. Based on Punch-Through (PT) & Non-Punch-Through (NPT)

1. **Punch-Through IGBT (PT-IGBT)**
    - Has an additional buffer layer.
    - **Lower conduction loss, faster switching.**
    - Used in **high-speed applications** (inverters, power supplies).
2. **Non-Punch-Through IGBT (NPT-IGBT)**
    - No buffer layer, making it more robust.
    - **Better voltage withstand capability.**
    - Used in **high-voltage applications** (HVDC transmission).

B. Based on Switching Speed

1. **Low-Frequency IGBT** – Used in industrial motor drives (50-60 Hz).
2. **High-Frequency IGBT** – Used in power inverters and welding machines (>20 kHz).

---

6. Applications of IGBT

IGBTs are used in various **high-power** and **high-efficiency** applications.

✅ **Power Electronics:**

- **Inverters** (DC-AC conversion for solar panels, UPS).
- **Motor Drives** (electric vehicles, industrial motors).
- **HVDC Transmission** (high-voltage direct current systems).

✅ **Renewable Energy Systems:**

- **Solar Inverters** (converting DC from solar panels to AC).
- **Wind Turbine Power Control.**

✅ **Industrial Applications:**

- **Electric Trains, Locomotives** (railway traction systems).
- **Induction Heating** (for melting, welding, and hardening).

✅ **Consumer Electronics:**

- **Air Conditioners, Refrigerators** (motor control circuits).
- **UPS (Uninterruptible Power Supply)**.

📌 **Why is IGBT used in Electric Vehicles (EVs)?**

- Handles **high power & voltage** efficiently.
- Used in **EV motor controllers** and **battery chargers**.

---

7. IGBT vs MOSFET vs BJT

Feature	IGBT	MOSFET	BJT
Control	Voltage-controlled	Voltage-controlled	Current-controlled
Switching Speed	Medium	Fast	Slow

Feature	IGBT	MOSFET	BJT
Conduction Loss	Low	Higher than IGBT	Lowest
Voltage Rating	High (up to 6500V)	Medium (up to 900V)	High
Current Handling	High	Low to Medium	High
Efficiency	High	High	Medium
Applications	Power electronics, motor drives	High-frequency switching, low-power circuits	Amplifiers, low-speed applications

📌 **MOSFETs** are better for **high-speed, low-power applications** (e.g., SMPS, RF circuits).
📌 **IGBTs** are better for **high-power, moderate-speed applications** (e.g., inverters, EVs, industrial motor drives).

---

8. Advantages & Disadvantages of IGBT

✅ Advantages

- **High efficiency** (low conduction losses).
- **Handles high voltage & current** (ideal for power electronics).
- **Simple gate drive** (voltage-controlled like MOSFET).
- **Compact size & reduced component count**.

❌ Disadvantages

- **Slower switching** compared to MOSFET.
- **Higher switching losses** than MOSFET (due to tail current).
- **Not suitable for very high-frequency applications** (>100 kHz).

---

9. Example Circuit – IGBT in Motor Control

Components Required:

- **IGBT (e.g., IRG4PC50U)**
- **Microcontroller (Arduino, STM32)**
- **12V/24V DC Motor**
- **Diode (for flyback protection)**
- **Resistors & capacitors**

📌 Circuit Connection:

1. **Gate (G) → PWM signal from microcontroller** (via a gate resistor).
2. **Collector (C) → Motor (load).**
3. **Emitter (E) → Ground.**
4. **Diode across the motor** (for protection against back EMF).

2. When to Use a MOSFET

✅ Use MOSFET if:

1. **High-Frequency Switching (>100 kHz) is needed**
    - **Example:** Switching Power Supplies (SMPS), Class D Audio Amplifiers.
2. **Operating Voltage is Below 400V**
    - **Example:** Battery-powered devices, DC-DC converters.
3. **Low Conduction Losses at Low Voltage are Required**
    - **Example:** Low-voltage motor controllers, LED drivers.
4. **Fast Switching is Required**
    - **Example:** RF applications, power supplies, induction heating.

🚀 Typical Applications of MOSFETs:

- **Switched-Mode Power Supplies (SMPS)**
- **DC-DC Converters**
- **High-frequency inverters**
- **Low-voltage motor controllers (<100V)**
- **Class D audio amplifiers**
- **RF amplifiers & communication systems**

---

3. When to Use an IGBT

✅ Use IGBT if:

1. **High Voltage (>400V) & High Current is needed**
    - **Example:** Electric Vehicle (EV) Motor Controllers.
2. **Medium Switching Frequency (up to ~100 kHz) is acceptable**
    - **Example:** Industrial motor drives, induction heating.
3. **Low Conduction Losses at High Power are Important**
    - **Example:** High-power inverters for renewable energy.
4. **Soft Switching Applications**
    - **Example:** Resonant converters, HVAC systems.

🚀 Typical Applications of IGBTs:

- **Industrial motor drives (400V – 1200V)**

- **Electric vehicle inverters (EV motor controllers)**
- **HVDC power transmission**
- **Solar & wind power inverters**
- **Induction heating**
- **Air conditioner & refrigeration compressors**

29.Solar Panel and Charge Controller

A **solar panel (photovoltaic panel)** is a device that converts **sunlight into electricity** using the **photovoltaic effect**. It is widely used in **renewable energy systems** such as homes, industries, and solar farms.

---

1. Construction of a Solar Panel

A **solar panel** is made up of multiple **solar cells** connected together. Each solar cell is a **semiconductor device** that converts sunlight into electricity.

Layers of a Solar Panel:

🔹 **1. Glass Cover:**

- Protects solar cells from weather (rain, dust, hail).
- Made of **tempered glass** for durability.

🔹 **2. Anti-Reflective Coating (ARC):**

- Reduces reflection to increase light absorption.

🔹 **3. Solar Cells (Photovoltaic Cells):**

- The **heart of the panel**, made from **silicon (Si)**.
- Converts sunlight into electricity.

🔹 **4. Encapsulation Layer:**

- Protects the solar cells from moisture and dust.
- Usually made of **EVA (Ethylene Vinyl Acetate)**.

🔹 **5. Back Sheet:**

- Provides insulation and protection from heat & weather.

❖ 6. Frame:

- Made of **aluminum** for mechanical strength and easy mounting.

❖ 7. Junction Box:

- Contains **diodes** to prevent reverse current flow.
- Connects to **external circuits** (battery, inverter, load).

---

2. Working Principle of Solar Panels

Step-by-Step Working of a Solar Panel:

1 Sunlight Strikes the Solar Panel

- The **sun emits photons (light energy)**.
- These **photons hit the solar cells** in the panel.

2 Photon Energy Excites Electrons

- Each **solar cell** is made of **silicon (Si), a semiconductor**.
- When photons strike the **silicon atoms**, electrons gain energy and become **free electrons**.

3 Electric Field Drives Electrons

- The solar cell has a **PN Junction** (formed by doping silicon with phosphorus & boron).
- This junction creates an **electric field** that forces electrons to move in one direction.

4 Flow of Electrons Produces DC Electricity

- The **movement of electrons** creates **direct current (DC)** electricity.
- Metal **conductors collect the electrons** and send them to an external circuit.

5 Power Conversion for Use

- **Inverters** convert **DC to AC** for household & industrial use.
- The electricity can be used **immediately or stored in batteries**.

---

3. Types of Solar Panels

❖ 1. Monocrystalline Solar Panels (Most efficient)

- Made from **single-crystal silicon**.
- **Efficiency: 18% – 22%**.

- Expensive but long-lasting.

🔹 **2. Polycrystalline Solar Panels** (Affordable)

- Made from **multiple silicon crystals**.
- **Efficiency: 15% – 18%**.
- Cheaper but slightly less efficient.

🔹 **3. Thin-Film Solar Panels** (Flexible & lightweight)

- Made of **amorphous silicon or other materials**.
- **Efficiency: 10% – 12%**.
- Used in portable solar devices & building-integrated PV.

---

4. Applications of Solar Panels

✅ **Residential Use** (rooftop solar power for homes).
✅ **Industrial Power Generation** (large-scale solar farms).
✅ **Solar Street Lights & Traffic Signals**.
✅ **Solar-Powered Water Pumps & Irrigation Systems**.
✅ **Portable Solar Chargers (for phones, laptops, etc.)**.

---

5. Advantages & Disadvantages

✅ **Advantages:**

- Renewable & eco-friendly.
- Low maintenance cost.
- Reduces electricity bills.
- Works in remote locations.

❌ **Disadvantages:**

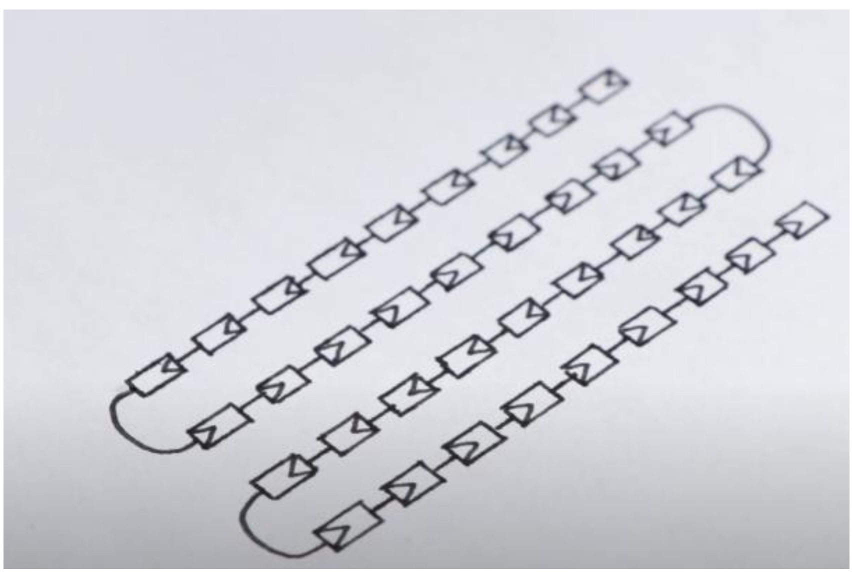
- High initial cost.
- Efficiency depends on sunlight availability.
- Requires space for installation.

---

6. Summary

- **Solar panels convert sunlight into electricity using the photovoltaic effect.**
- **Made of silicon solar cells, encapsulated in glass & polymer layers.**
- **Generates DC power, which is converted to AC for use.**
- **Types: Monocrystalline, Polycrystalline, Thin-Film.**
- **Used in homes, industries, and solar farms.**

The solar panel connects the cells in series to get bigger output.



Disadvantage of Connecting Solar Cells in Series

When solar cells are connected **in series**, the total voltage increases, but the **current is limited by the weakest cell** in the series. This means:

⬤ Shading or Partial Blockage Reduces Power Output Significantly

- If one solar cell is **shaded, damaged, or malfunctioning**, it reduces the **current** for the entire string of cells.
- Since **current remains the same in a series circuit**, even if one cell produces less current, the whole series output is limited to that weak cell's current.

⬤ Hot Spot Effect & Potential Damage

- When a shaded or faulty cell **does not generate power**, it acts like a **resistor**.
- The unshaded cells **force current through the weak cell**, causing **overheating** (hot spot effect).
- This can **damage the panel permanently**.

Impact of Clouds on Series-Connected Solar Cells

☁ When clouds cover the solar panel, they reduce the sunlight intensity, leading to:

- **Reduced power generation** since solar cells get less energy.
- If some cells receive more sunlight than others, **power loss increases** due to the weak cells limiting the entire series.

🔹 Partial Cloud Cover (Non-uniform shading) → More Impact

- If clouds **partially cover the panel**, only some cells are shaded while others receive full sunlight.
- This creates an **imbalance**, and the entire series output is affected.

🔹 Full Cloud Cover (Uniform shading) → Less Impact

- If the entire panel is under a cloud, all cells produce the same lower current, so the system still works but at reduced efficiency.

Solutions to Minimize These Problems

✅ Bypass Diodes:

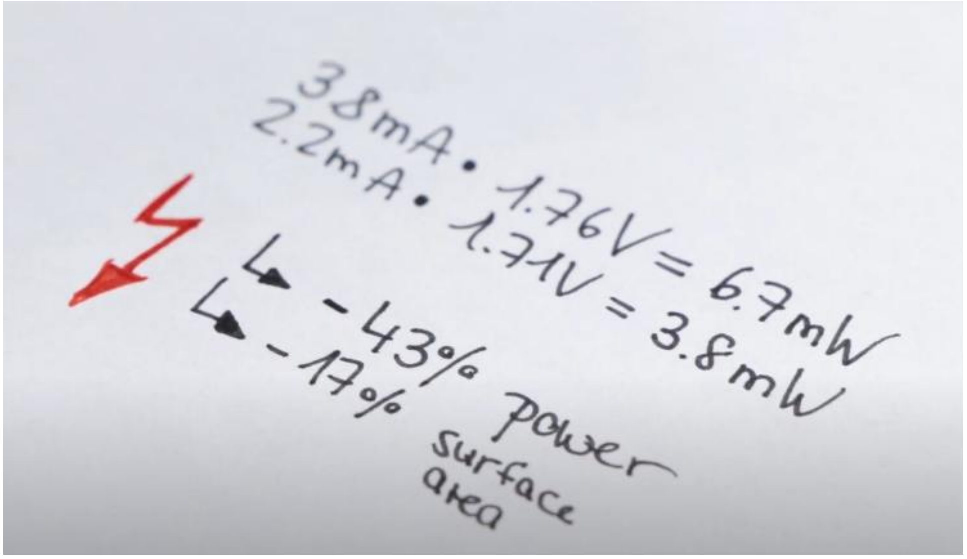
- Installed across individual cells or groups of cells to **allow current to bypass** shaded or weak cells.
- Prevents **power loss and hot spots**.

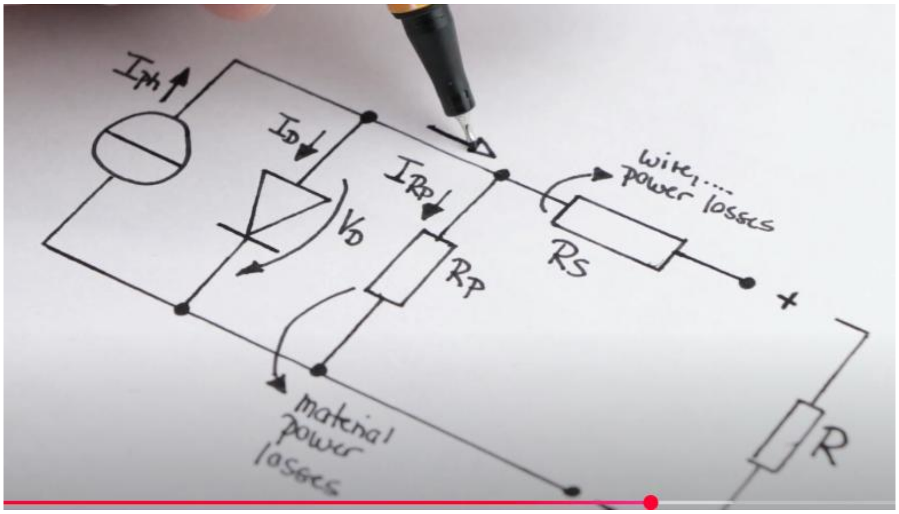
✅ Parallel Connection of Solar Panels:

- Instead of connecting all panels in series, a **parallel connection** ensures that shading affects only part of the system, not the whole array.

✅ **MPPT (Maximum Power Point Tracking) Controllers:**

- Adjust the voltage and current dynamically to **optimize power output**, even under shading conditions.





30.Microcontroller(Arduino) Timer

Timers in **microcontrollers (including Arduino)** are essential for handling **precise timing operations**, such as **delays, PWM (Pulse Width Modulation), interrupts, and frequency generation**.

---

1. What is a Timer in Arduino?

A **timer** is a hardware unit in a microcontroller that **counts clock pulses** to measure time intervals. It operates independently of the CPU, allowing precise time-based operations.

🔹 **Timers are used for:**
✅ Generating **precise delays** without using `delay()` (which blocks code execution).
✅ Creating **PWM signals** for motor control, LED dimming, etc.
✅ Handling **interrupts** for real-time applications.
✅ Measuring **external events** like frequency counting.
✅ Generating **square waves** for clocks or signals.

---

2. Types of Timers in Arduino

Arduino boards (like the **Uno, Mega, and Nano**) have **three** built-in timers:

Timer	Bit Size	Used For	Prescaler
Timer0	8-bit	Delay functions (`millis()`, `delay()`)	1, 8, 64, 256, 1024
Timer1	16-bit	Servo control, frequency generation, interrupts	1, 8, 64, 256, 1024
Timer2	8-bit	PWM, tone generation (`tone()` function)	1, 8, 32, 64, 128, 256, 1024

📌 **Bit Size:**

- **8-bit Timer**: Counts from `0 to 255` (faster but lower resolution).
- **16-bit Timer**: Counts from `0 to 65535` (higher precision).

📌 **Prescaler:**

- Controls the timer speed by dividing the clock frequency (e.g., with a **prescaler of 64**, the timer runs at `16 MHz / 64 = 250 kHz`).

---

3. Timer Modes in Arduino

Timers work in different modes to perform various functions:

◈ 1. Normal Mode (Overflow Mode)

- The timer **counts up** and resets when it reaches its maximum value (`255` for 8-bit or `65535` for 16-bit).
- Used for **delays and periodic interrupts**.

◈ Example: Generating an **interrupt every 1ms** using **Timer1**.

```
ISR(TIMER1_OVF_vect) {
  // Code to execute every overflow (1ms delay)
}

void setup() {
  noInterrupts();  // Disable interrupts
  TCCR1A = 0;      // Normal mode
  TCCR1B = (1 << CS12);  // Prescaler 256
  TIMSK1 = (1 << TOIE1); // Enable Timer1 overflow interrupt
  interrupts();  // Enable interrupts
}
```

◈ 2. CTC Mode (Clear Timer on Compare Match)

- The timer **resets** when it reaches a specific compare value, instead of overflowing at `255` or `65535`.
- Used for **precise timing intervals** and generating fixed-frequency signals.

◈ Example: Generate a **1 kHz signal** using **Timer2**.

```
void setup() {
  pinMode(3, OUTPUT);  // Set pin as output
  TCCR2A = (1 << WGM21);  // CTC Mode
  TCCR2B = (1 << CS22);   // Prescaler 64
  OCR2A = 249;  // Compare Match (1 kHz output)
  TIMSK2 = (1 << OCIE2A); // Enable Timer Interrupt
}

ISR(TIMER2_COMPA_vect) {
  digitalWrite(3, !digitalRead(3));  // Toggle pin
}
```

◈ 3. PWM Mode (Fast PWM & Phase Correct PWM)

- Generates **PWM (Pulse Width Modulation) signals** for motor control, LED brightness, etc.
- **Fast PWM**: Higher speed, less accuracy.
- **Phase Correct PWM**: More stable, better for motors.

◈ Example: **PWM on Pin 9 using Timer1 (16-bit)**

```
void setup() {
  pinMode(9, OUTPUT);
  TCCR1A = (1 << COM1A1) | (1 << WGM11);  // Fast PWM
  TCCR1B = (1 << WGM12) | (1 << WGM13) | (1 << CS11);  // Prescaler 8
  ICR1 = 39999;  // Set PWM frequency (50 Hz for servos)
  OCR1A = 3000;  // Set duty cycle
}
```

---

4. Timer Functions in Arduino

🔹 `millis()` and `micros()`

- `millis()`: Returns **time in milliseconds** since the program started. (Uses **Timer0**)
- `micros()`: Returns **time in microseconds** (Uses **Timer0**)

```
unsigned long time = millis();  // Get current time
if (millis() - time > 1000) {  // 1-second delay without blocking
  digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
}
```

🔹 `delay()` vs Timers

- `delay(ms)`: Blocks execution; better to use timers for precise delays.

🔹 `tone(pin, frequency)`

- Uses **Timer2** to generate a square wave at a given frequency.

```
cpp
CopyEdit
tone(8, 1000);  // Generate 1 kHz tone on pin 8
```

---

5. Applications of Arduino Timers

✅ **Generating Precise Delays Without Blocking Code**
✅ **PWM Signal Generation (for LEDs, motors, servo control)**
✅ **Creating Frequency Signals for Communication**
✅ **Interrupt-based Real-Time Tasks**
✅ **Measuring Time Intervals Between Events**

---

6. Summary

- Arduino has **3 timers (Timer0, Timer1, Timer2)** used for delays, PWM, and interrupts.
- Timers can operate in **Normal Mode, CTC Mode, and PWM Mode**.

- **PWM is used for motor control & LED dimming**.
- **Interrupts allow precise, real-time execution** without blocking code.