Avoid Obstacle Robot

American International University-Bangladesh

Where Leaders are Created

Electrical Circuit 1 Lab (DC)

Submitted To-

DR. MD. MOHIUDDIN UZZAL

Fall 2019 – 2020

Contribution:

Equipment Collection:

1. Tamzid Ahmed

2. Riazul Haque Rana

3. Katibur Rahman Sunny

4. Equipment Collection:

Circuit Design:

1. Tamzid Ahmed

2. Riazul Haque Rana

Report Writing:

1. Hriday khan

2. Katibur Rahman Sunny

3. Mehedi Hasan

4, Shohanul Islam Shohan

Presentation:

1. Mehedi Hasan

2. Shafayet Hossain Shohan

3. Minhaz Rashik

4. Tamzid Ahmed

Introduction:

Obstacle avoidance is one of the most important aspects of mobile robotics. Without it robot movement would be very restrictive and fragile. This tutorial explains obstacle avoidance using ultrasonic sensors. This project also presents a dynamic steering algorithm which ensures that the robot doesn't have to stop in front of an obstacle which allows robot to navigate smoothly in an unknown environment, avoiding collisions.

Equipment:

1. Arduino UNO R3

2. Motor Driver Shield

3. Wheels(4X)

4. TT Gear Motor(4X)

5. Servo Motor

6. Ultrasonic Sensor

7. 18650 Li-on Battery(2x)

8. 18650 Battery Holder

9. Male and Female Jumper Wire

10. DC Power Switch 6 pin

What is Arduino?

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a micro controller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board. you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

The Uno is one of the more popular boards in the Arduino family and a great choice for beginners.

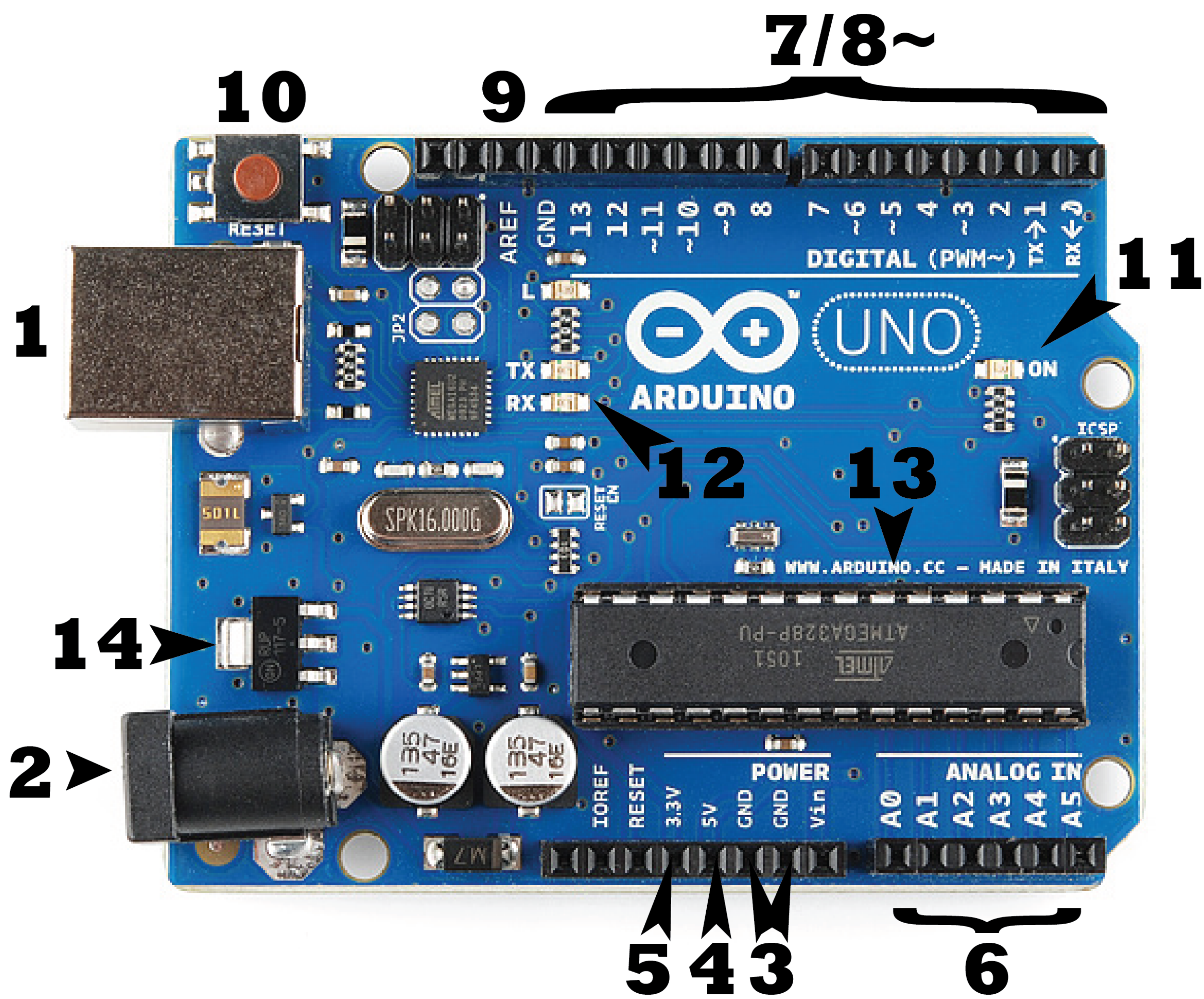
What Does it Do?

The Arduino hardware and software was designed for artists, designers, hobbyists, hackers, newbies, and anyone interested in creating interactive objects or environments. Arduino can interact with buttons, LED, motors, speakers, GPS units, cameras, the internet, and even your smart-phone or your TV! This flexibility combined with the fact that the Arduino software is free, the hardware boards are pretty cheap, and both the software and hardware are easy to learn has led to a large community of users who have contributed code and released instructions for a **huge** variety of Arduino-based projects.

For everything from robots and a heating pad hand warming blanket to honest fortune-telling machines, and even a Dungeons and Dragons dice-throwing gauntlet, the Arduino can be used as the brains behind almost any electronics project.

What's on the board?

There are many varieties of Arduino boards that can be used for different purposes. Some boards look a bit different from the one below, but most Arduino's have the majority of these components in common:



Power (USB / Barrel Jack)

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply that is terminated in a barrel jack. In the picture above the USB connection is labeled **(1)** and the barrel jack is labeled **(2)**.

The USB connection is also how you will load code onto your Arduino board. More on how to program with Arduino can be found in our Installing and Programming Arduino tutorial.

NOTE: Do NOT use a power supply greater than 20 Volts as you will overpower (and thereby destroy) your Arduino. The recommended voltage for most Arduino models is between 6 and 12 Volts.

Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjuction with a breadboard and some wire. They usually have black plastic 'headers' that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

- **GND (3)**: Short for 'Ground'. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- **5V (4) & 3.3V (5)**: As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
- **Analog (6)**: The area of pins under the 'Analog In' label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.
- **Digital (7)**: Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
- **PWM (8)**: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).
- **AREF (9)**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Reset Button

Just like the original Nintendo, the Arduino has a reset button **(10)**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

Power LED Indicator

Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON' **(11)**. This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit!

TX RX LEDs

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the

Arduino UNO where TX and RX appear -- once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs **(12)**. These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program onto the board).

Main IC

The black thing with all the metal legs is an IC, or Integrated Circuit **(13)**. Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC's from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC's, reading the datasheets is often a good idea.

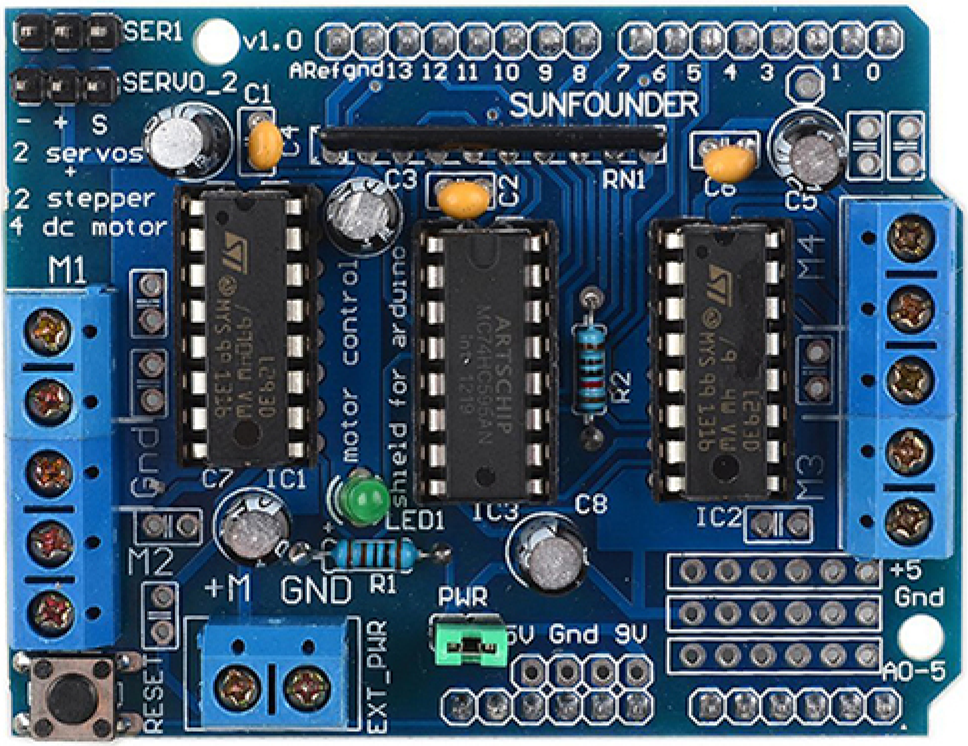
Voltage Regulator

The voltage regulator **(14)** is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says -- it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don't hook up your Arduino to anything greater than 20 volts.

Arduino Uno (R3)

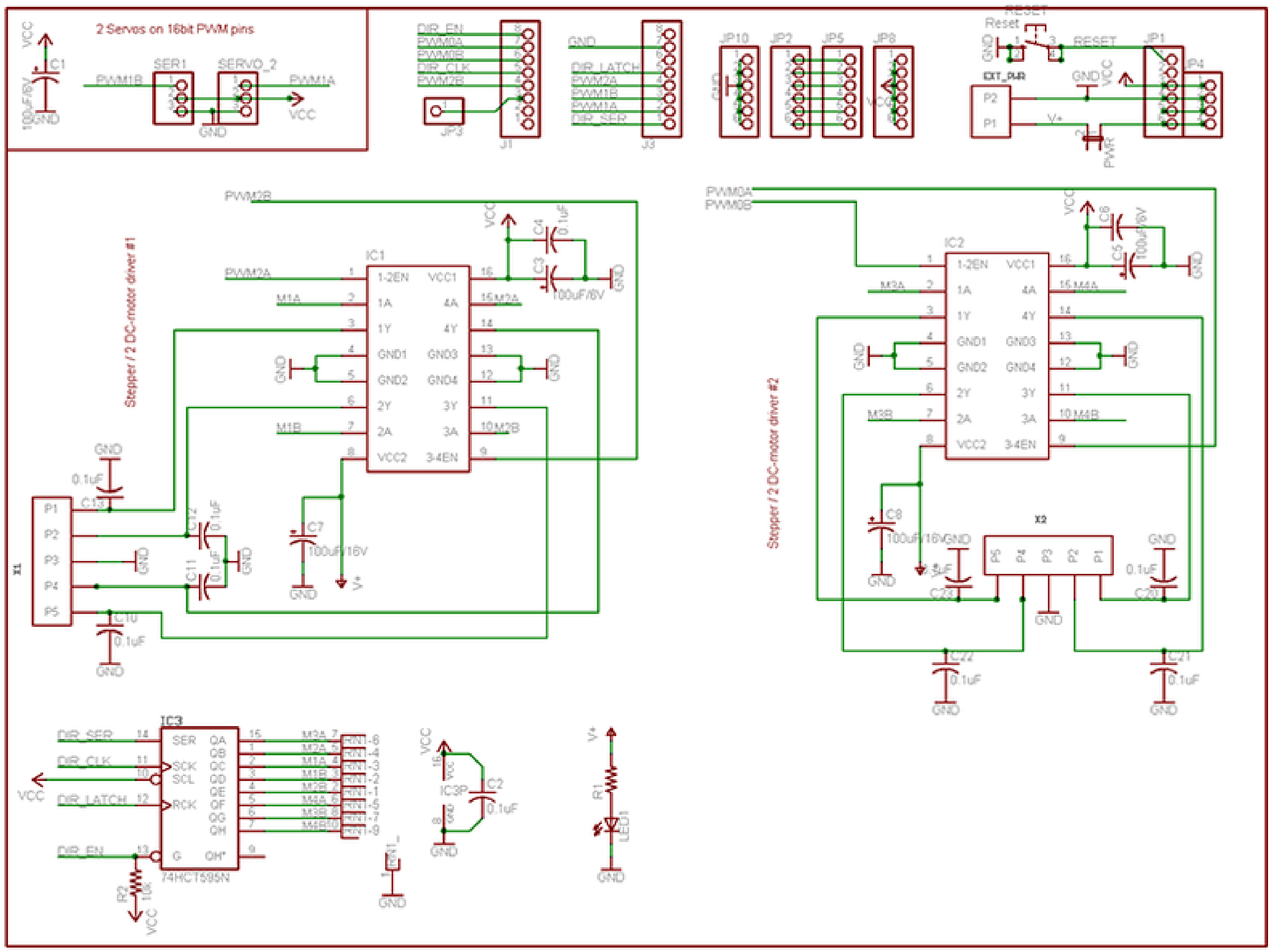
The Uno is a great choice for your first Arduino. It's got everything you need to get started, and nothing you don't. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a USB connection, a power jack, a reset button and more. It contains everything needed to support the micro controller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

What is Motor Driver Shield?



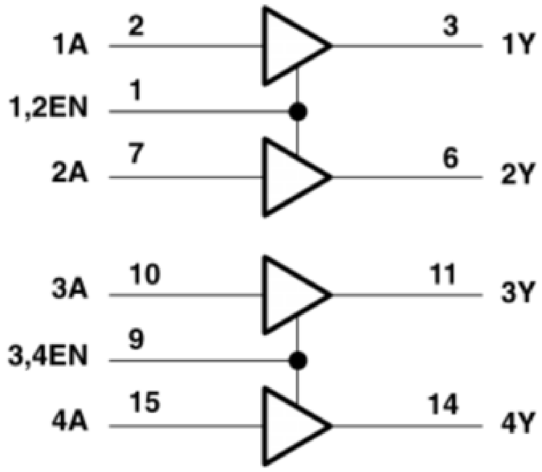
SunFounder L293D is a monolithic integrated, high voltage, high current, 4-channel driver. Basically this means using this chip you can use DC motors and power supplies of up to 16 Volts, that's some pretty big motors and the chip can supply a maximum current of 600mA per channel, the L293D chip is also what's known as a type of H-Bridge. The H-Bridge is typically an electrical circuit that enables a voltage to be applied across a load in either direction to an output.

The schematic diagram is as follows :



L293D:

The L293D is quadruple high-current half-H drivers. It is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications. All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.



Logic diagram

FUNCTION TABLE
(each driver)

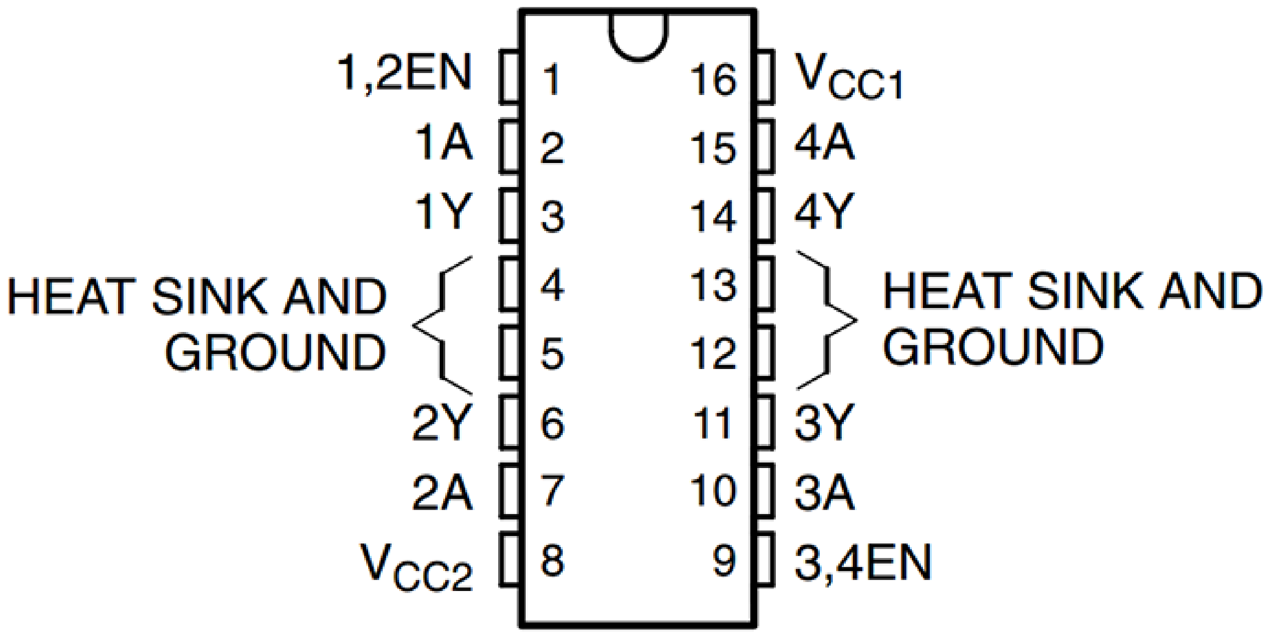
INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high level, L = low level, X = irrelevant, Z = high impedance (off)

† In the thermal shutdown mode, the output is in the high-impedance state, regardless of the input levels.

Function Table

Pin Function:

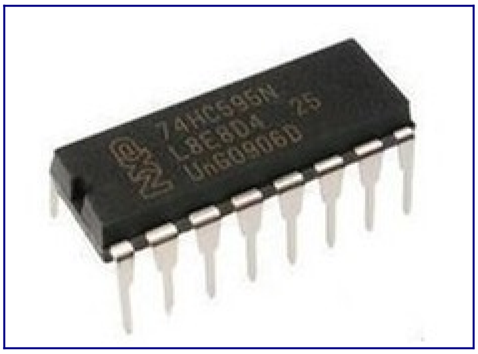


Pin	Name	Function
1	Enable1,2	Enable pin to control 1,2 driver
2	Input 1A	Input to control 1Y
3	Output 1Y	Output, connect to motor
4	GND	Ground and heat sink
5	GND	Ground and heat sink
6	Output 2Y	Output, connect to motor
7	Input 2A	Input to control 2Y
8	Vcc2	Output supply voltage
9	Enable3,4	Enable pin to control 3,4 driver
10	Input 3A	Input to control 3Y
11	Output 3Y	Output, connect to motor
12	GND	Ground and heat sink
13	GND	Ground and heat sink
14	Output 4Y	Output, connect to motor
15	Input 4A	Input to control 4Y
16	Vcc1	Supply voltage(7 max)

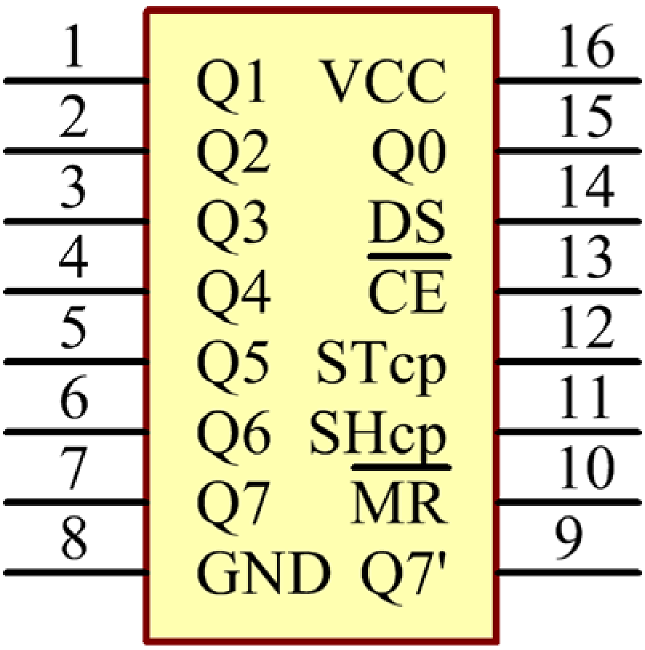
Features

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- Thermal Shutdown
- High-Noise-Immunity Inputs
- Output Current 600 mA Per Channel
- Peak Output Current 1.2 A Per Channel

74HC595



The 74HC595 consists of an 8−bit shift register and a storage register with three−state parallel outputs. It converts serial input into parallel output so that you can save IO ports of an MCU. The 74HC595 is widely used to indicate multipath LEDs and drive multi-bit segment displays. "Three-state" refers to the fact that you can set the output pins as either high, low or "high impedance." With data latching, the instant output will not be affected during the shifting; with data output, you can cascade 74HC595s more easily.



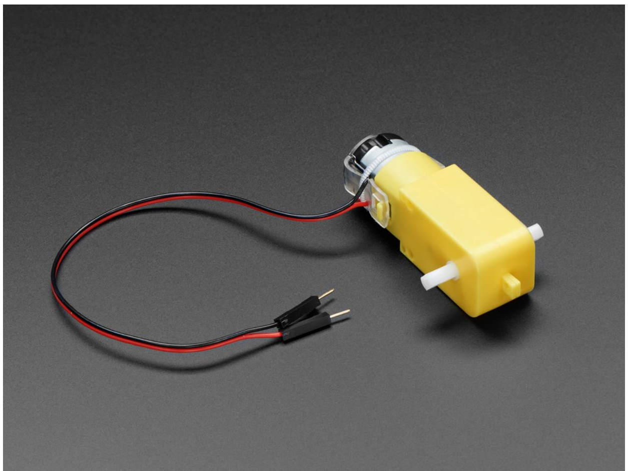
No	Name	Function
1	Q0-Q7	8-bit parallel data output pins, able to control 8 LEDs or 8 pins of 7-segment display directly.
2	Q7’	Series output pin, connected to DS of another 74HC595 to connect multiple 74HC595s in series

3	MR	Reset pin, active at low level; here it is directly connected to 5V.
4	SHcp	Time sequence input of shift register. On the rising edge, the data in shift register moves successively one bit `i.e. data in Q1 moves to Q2, and so forth. While on the falling edge, the data in shift register remain unchanged.`
5	STcp	Time sequence input of storage register. On the rising edge, data in the shift register moves into memory register.
6	OE	Output enable pin, active at low level
7	DS	Serial data input pin
8	VCC	Positive supply voltage
9	GND	Ground

Features

- 8-Bit Serial-In, Parallel-Out Shift
- Wide Operating Voltage Range of 2 V to 6 V
- High-Current 3-State Outputs Can Drive Up To 15 LSTTL Loads
- Low Power Consumption, 80-µA Max ICC
- Typical tpd = 13 ns
- Low Input Current of 1
- Shift Register Has Direct Clear

What is Gear Motor?



These durable (but affordable!) plastic gearbox motors (also known as 'TT' motors) are an easy, low-cost way to get your projects moving. This is a **TT DC Gearbox Motor** with a gear ratio of **1:48**, and it comes with 2 x 200mm wires with breadboard-friendly 0.1" male connectors. Perfect for plugging into a breadboard or terminal blocks.

You can power these motors with 3VDC up to 6VDC, they'll of course go a little faster at the higher voltages. We grabbed one motor and found these stats when running it from a bench-top supply

- At **3VDC** we measured 150mA @ 120 RPM no-load, and 1.1 Amps when stalled
- At **4.5VDC** we measured 155mA @ 185 RPM no-load, and 1.2 Amps when stalled
- At **6VDC** we measured 160mA @ 250 RPM no-load, and 1.5 Amps when stalled

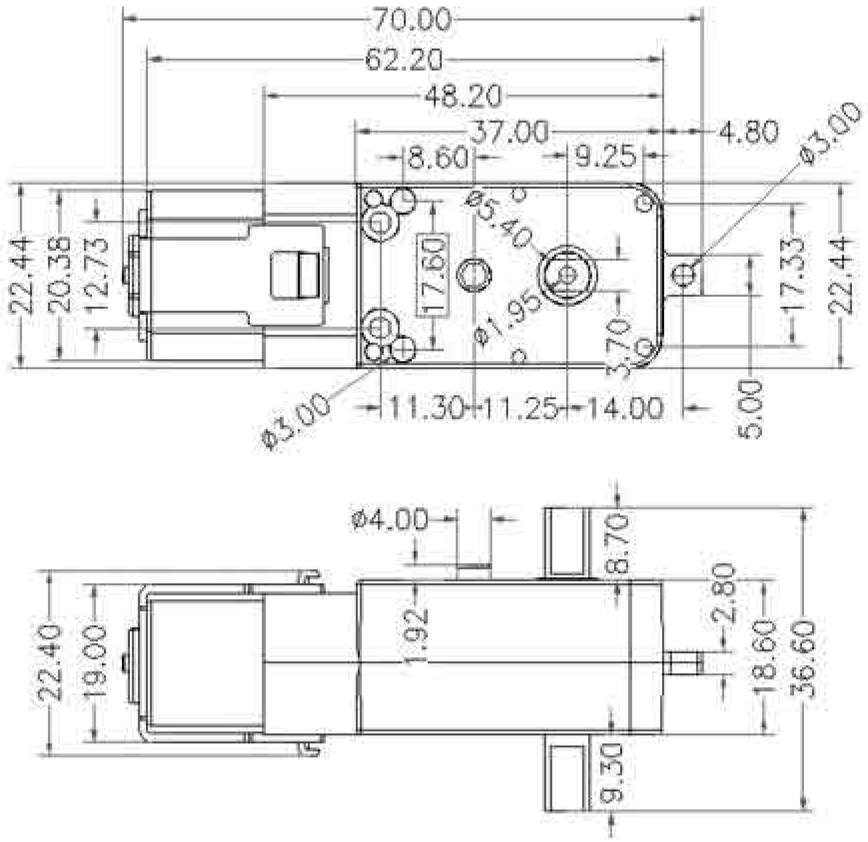
Note that these are very basic motors, and have no built-in encoders, speed control or positional feedback. Voltage goes in, rotation goes out! There will be variation from motor to motor, so a separate feedback system is required if you need precision movement.

Comes 1 x per order, with just the motor + wires. You *cannot* drive these directly from a micro controller, **a high-current motor driver is required**! We recommend DRV8833 motor driver for these motors, as it works well down to 3V and can be set up with current limiting since the stall current on these can get high. The TB6612 can also be used, it's on our shields and wings, but you'll need to supply at least 4.5V – which is what you'll likely want to run these motors at anyhow!

Technical Details

- Rated Voltage: 3~6V
- Continuous No-Load Current: 150mA +/- 10%
- Min. Operating Speed (3V): 90+/- 10% RPM
- Min. Operating Speed (6V): 200+/- 10% RPM
- Torque: 0.15Nm ~0.60Nm

- Stall Torque (6V): 0.8kg.cm
- Gear Ratio: 1:48
- Body Dimensions: 70 x 22 x 18mm
- Wires Length: 200mm & 28 AWG
- Weight: 30.6g



Product Weight: 30.6g / 1.1oz

What is TT Wheels?

Usually when one needs an orange wheel it's a garnish for a cocktail, like a tasty Sidecar. And speaking of cars, this wheel is for driving, not drinking!

Need a great drive solution for your little robotic friends? This **Orange & Clear TT Motor Wheel** is equipped with a nice, thick silicone tread and a press-fit design to make connecting super quick and easy. Make cute robots today with a couple of these orange wheels!

Comes one in an order, this wheel is **only for use with 'TT' gearbox DC motors.**

Technical Details

- Wheel body: 63 x 29mm / 2.4" x 1.1"

Product Weight: 38.0g / 1.3oz

What is Servo Motor?

A servo motor is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use servo motor. It is just made up of simple motor which run through servo mechanism. If motor is used is DC powered then it is called DC servo motor, and if it is AC powered motor then it is called AC servo motor. We can get a very high torque servo motor in a small and light weight packages. Doe to these features they are being used in many applications like toy car, RC helicopters and planes, Robotics, Machine etc.

Servo motors are rated in kg/cm (kilogram per centimeter) most hobby servo motors are rated at 3kg/cm or 6kg/cm or 12kg/cm. This kg/cm tells you how much weight your servo motor can lift at a particular distance. For example: A 6kg/cm Servo motor should be able to lift 6kg if the load is suspended 1cm away from the motors shaft, the greater the distance the lesser the weight carrying capacity.

The position of a servo motor is decided by electrical pulse and its circuitry is placed beside the motor.

Servo Mechanism

It consists of three parts:

1. Controlled device
2. Output sensor

3. Feedback system

It is a closed loop system where it uses positive feedback system to control motion and final position of the shaft. Here the device is controlled by a feedback signal generated by comparing output signal and reference input signal.

Here reference input signal is compared to reference output signal and the third signal is produces by feedback system. And this third signal acts as input signal to control device. This signal is present as long as feedback signal is generated or there is difference between reference input signal and reference output signal. So the main task of servomechanism is to maintain output of a system at desired value at presence of noises.

Working principle of Servo Motors

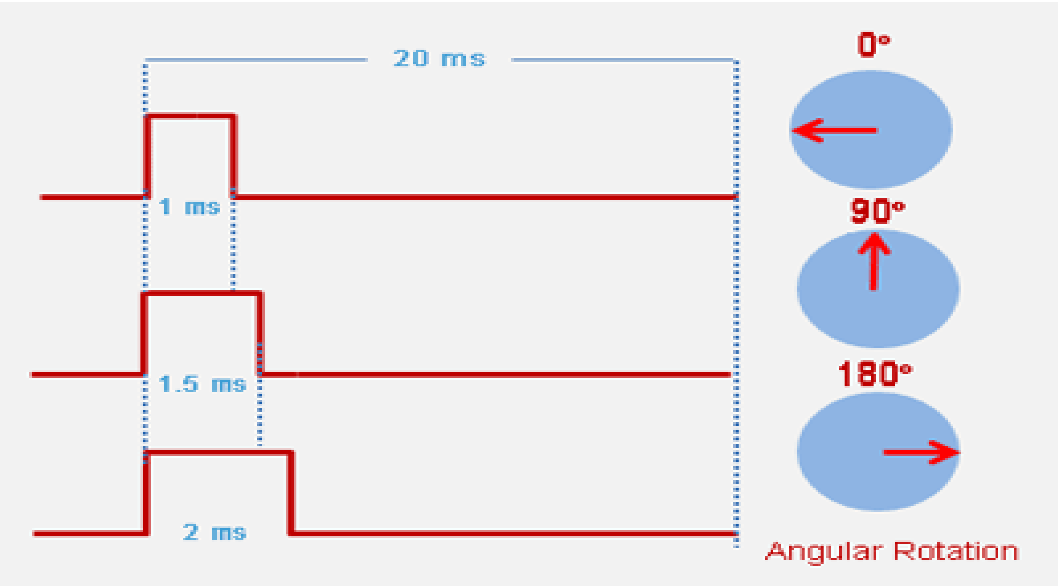
A servo consists of a Motor (DC or AC), a potentiometer, gear assembly and a controlling circuit. First of all we use gear assembly to reduce RPM and to increase torque of motor. Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detector amplifier. Now difference between these two signals, one comes from potentiometer and another comes from other source, will be processed in feedback mechanism and output will be provided in term of error signal. This error signal acts as the input for motor and motor starts rotating. Now motor shaft is connected with potentiometer and as motor rotates so the potentiometer and it will generate a signal. So as the potentiometer's angular position changes, its output feedback signal changes. After sometime the position of potentiometer reaches at a position that the output of potentiometer is same as external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer, and in this situation motor stops rotating.

Controlling Servo Motor:

All motors have three wires coming out of them. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU.

Servo motor is controlled by PWM (Pulse with Modulation) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 90 degree from either direction form its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position, such as if pulse is shorter than 1.5ms shaft moves to 0° and if it is longer than 1.5ms than it will turn the servo to 180°.

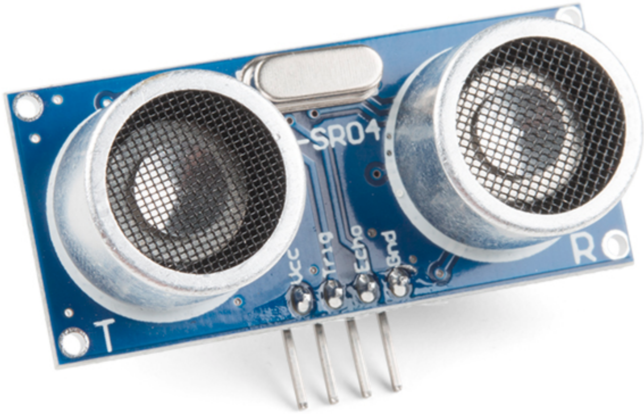
Servo motor works on **PWM (Pulse width modulation)** principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically servo motor is made up of **DC motor which is controlled by a variable resistor (potentiometer) and some gears**. High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. Potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on required angle.



Servo motor can be rotated from 0 to 180 degree, but it can go up to 210 degree, depending on the manufacturing. This degree of rotation can be controlled by applying the Electrical Pulse of proper width, to its Control pin. Servo checks the pulse in every 20 milliseconds. Pulse of 1 ms (1 millisecond) width can rotate servo to 0 degree, 1.5ms can rotate to 90 degree (neutral position) and 2 ms pulse can rotate it to 180 degree.

All servo motors work directly with your +5V supply rails but we have to be careful on the amount of current the motor would consume, if you are planning to use more than two servo motors a proper servo shield should be designed.

What is an ultrasonic sensor?

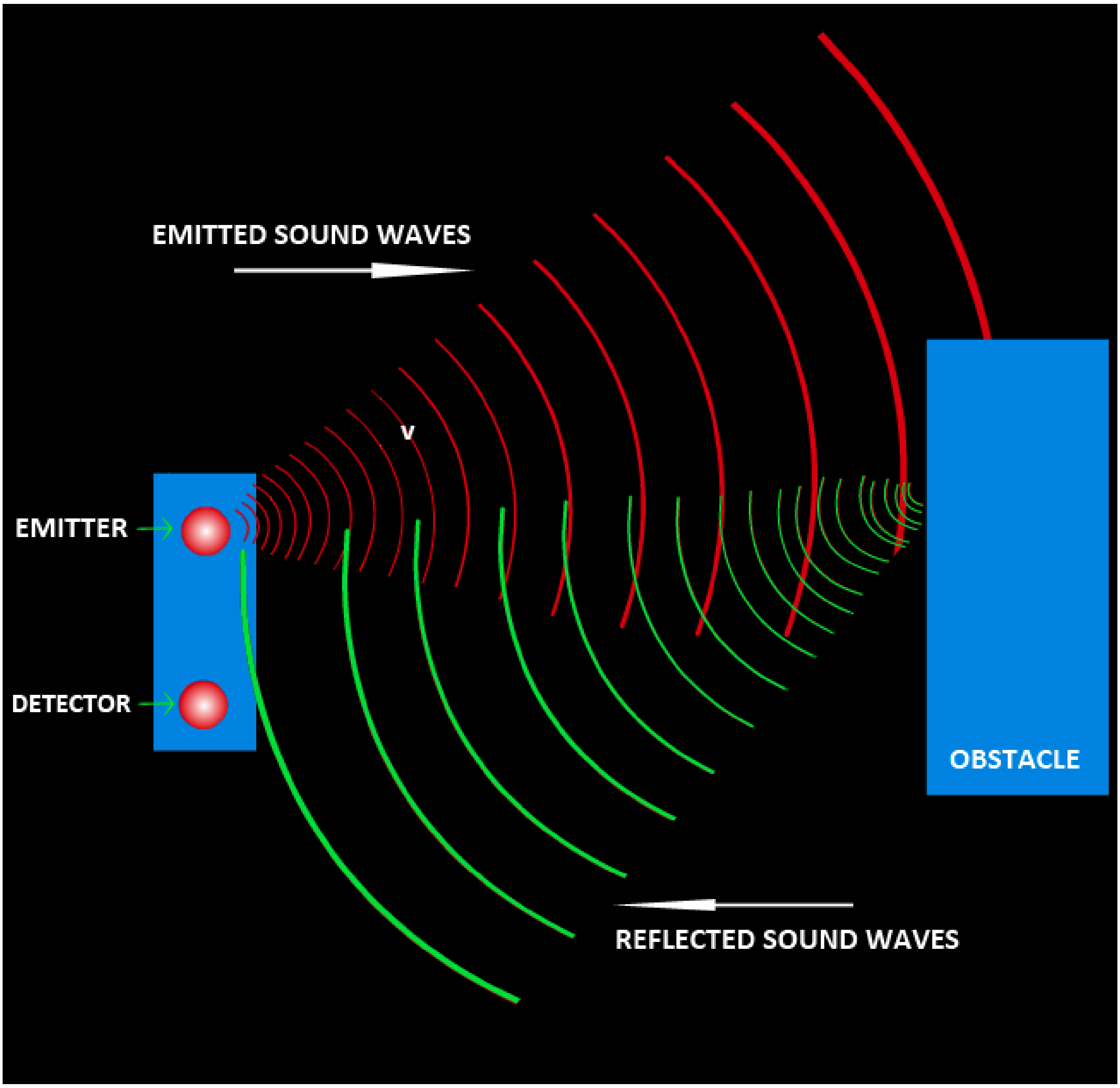


An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has traveled to and from the target).

In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver. The formula for this calculation is D = ½ T x C (where D is the distance, T is the time, and C is the speed of sound ~ 343 meters/second). For example, if a scientist set up an ultrasonic sensor aimed at a box and it took 0.025 seconds for the sound to bounce back, the distance between the ultrasonic sensor and the box would be:

$$\mathbf{D = 0.5 \times 0.025 \times 343}$$

or about 4.2875 meters.



Ultrasonic sensors are used primarily as proximity sensors. They can be found in automobile self-parking technology and anti-collision safety systems. Ultrasonic sensors are also used in robotic obstacle detection systems, as well as manufacturing technology. In comparison to infrared (IR) sensors in proximity sensing applications, ultrasonic sensors are not as susceptible to interference of smoke, gas, and other airborne particles (though the physical components are still affected by variables such as heat).

Ultrasonic sensors are also used as level sensors to detect, monitor, and regulate liquid levels in closed containers (such as vats in chemical factories). Most notably, ultrasonic technology has enabled the medical industry to produce images of internal organs, identify tumors, and ensure the health of babies in the womb.

What is Lithium-ion Battery?

A **lithium-ion battery** or **Li-ion battery** (abbreviated as **LIB**) is a type of rechargeable battery. Lithium-ion batteries are commonly used for portable electronics and electric vehicles and are growing in popularity for military and aerospace applications.The technology was largely developed by John Goodenough, Stanley Whittingham, Rachid Yazami and Akira Yoshino during the 1970s–1980s, and then commercialized by a Sony and Asahi Kasei team led by Yoshio Nishi in 1991.

In the batteries lithium ions move from the negative electrode through an electrolyte to the positive electrode during discharge, and back when charging. Li-ion batteries use an intercalated lithium compound as the material at the positive electrode and typically graphite at the negative electrode. The batteries have a high energy density, no memory effect (other than LFP cells)and low self-discharge. They can however be a safety hazard since they contain a flammable electrolyte, and if damaged or incorrectly charged can lead to explosions and fires. Samsung were forced to recall Galaxy Note 7 handsets following lithium-ion fires,and there have been several incidents involving batteries on Boeing 787s.

Chemistry, performance, cost and safety characteristics vary across LIB types. Handheld electronics mostly use lithium polymer batteries (with a polymer gel as electrolyte) with lithium cobalt oxide ($LiCoO_2$) as cathode material, which offers high energy density but presents safety risks, especially when damaged. Lithium iron phosphate ($LiFePO_4$), lithium ion manganese oxide battery ($LiMn_2O_4$, Li_2MnO_3, or LMO), and lithium nickel manganese cobalt oxide ($LiNiMnCoO_2$ or NMC) offer lower energy density but longer lives and less likelihood of fire or explosion. Such batteries are widely used for electric tools, medical equipment, and other roles. NMC in particular is a leading contender for automotive applications.

Research areas for lithium-ion batteries include life extension, energy density, safety, cost reduction, and charging speed, among others. Research has been under way in the area of non-flammable electrolytes as a pathway to increased safety based on the flammability and volatility of the organic solvents used in the typical electrolyte. Strategies include aqueous lithium-ion batteries, ceramic solid electrolytes, polymer electrolytes, ionic liquids, and heavily fluorinated systems.

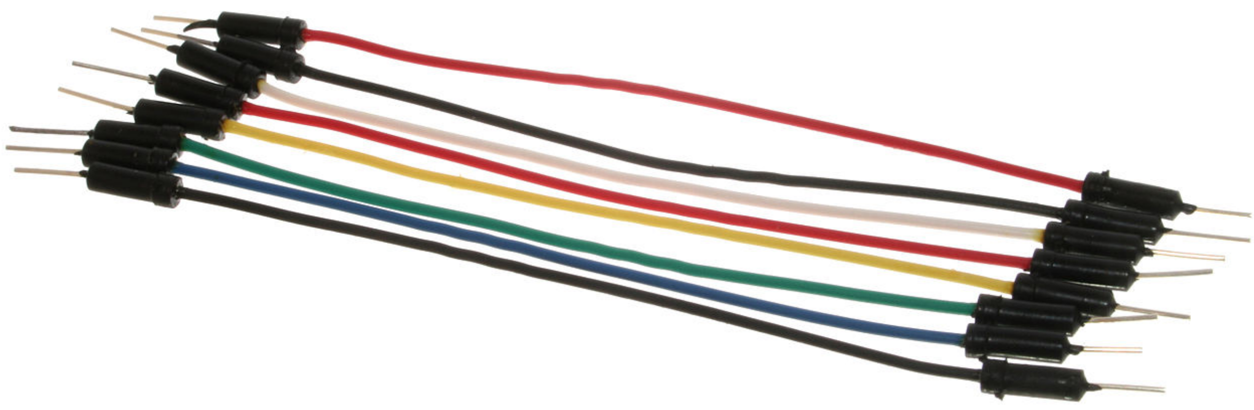
What is 18650 Battery Holder?

A holder which holds two Li-on Battery.

What is Jumper Wire?

A jump wire (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them– simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.



There are different types of jumper wires. Some have the same type of electrical connector at both ends, while others have different connectors. Some common connectors are:

- Solid tips – are used to connect on/with a breadboard or female header connector. The arrangement of the elements and ease of insertion on a breadboard allows increasing the mounting density of both components and jump wires without fear of short-circuits. The jump wires vary in size and color to distinguish the different working signals.
- Crocodile clips – are used, among other applications, to temporarily bridge sensors, buttons and other elements of prototypes with components or equipment that have arbitrary connectors, wires, screw terminals, etc.

- Banana connectors – are commonly used on test equipment for DC and low-frequency AC signals.
- Registered jack (RJnn) – are commonly used in telephone (RJ11) and computer networking (RJ45).
- RCA connectors – are often used for audio, low-resolution composite video signals, or other low-frequency applications requiring a shielded cable.
- RF connectors – are used to carry radio frequency signals between circuits, test equipment, and antennas.
- RF jumper cables – Jumper cables is a smaller and more bendable corrugated cable which is used to connect antennas and other components to network cabling. Jumpers are also used in base stations to connect antennas to radio units. Usually the most bendable jumper cable diameter is 1/2".

What is DC Power Switch?

The most familiar form of switch is a manually operated electromechanical device with one or more sets of electrical contacts, which are connected to external circuits. Each set of contacts can be in one of two states: either "closed" meaning the contacts are touching and electricity can flow between them, or "open", meaning the contacts are separated and the switch is nonconducting. The mechanism actuating the transition between these two states (open or closed) are usually (there are other types of actions) either an "*alternate action*" (flip the switch for continuous "on" or "off") or "*momentary*" (push for "on" and release for "off") type.

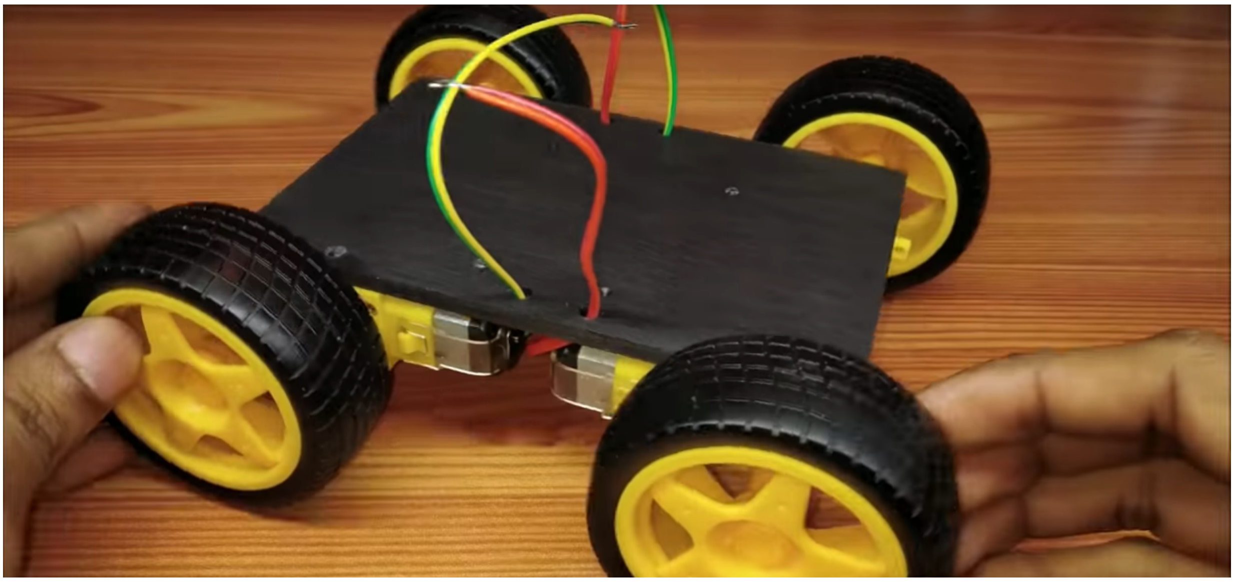
A switch may be directly manipulated by a human as a control signal to a system, such as a computer keyboard button, or to control power flow in a circuit, such as a light switch. Automatically operated switches can be used to control the motions of machines, for example, to indicate that a garage door has reached its full open position or that a machine tool is in a position to accept another work piece. Switches may be operated by process variables such as pressure, temperature, flow, current, voltage, and force, acting as sensors in a process and used to automatically control a system. For example, a thermostat is a temperature-operated switch used to control a heating process. A switch that is operated by another electrical circuit is called a relay. Large switches may be remotely operated by a motor drive mechanism. Some switches are used to isolate electric power from a system, providing a visible point of isolation that can be padlocked if necessary to prevent accidental operation of a machine during maintenance, or to prevent electric shock.

An ideal switch would have no voltage drop when closed, and would have no limits on voltage or current rating. It would have zero rise time and fall time during state changes, and would change state without "bouncing" between on and off positions.

Practical switches fall short of this ideal; as the result of roughness and oxide films, they exhibit contact resistance, limits on the current and voltage they can handle, finite switching time, etc. The ideal switch is often used in circuit analysis as it greatly simplifies the system of equations to be solved, but this can lead to a less accurate solution. Theoretical treatment of the effects of non-ideal properties is required in the design of large networks of switches, as for example used in telephone exchanges.

How to Build Avoid Obstacle Robot?

Step 1: A Chassis!



The first step and the base of any robot is a chassis. i got mine by cutting PVC. if you prefer, you can also make your own using cardboard and DC/Servo motors. Either way, the chassis has to include a body, four motors and sometimes even a battery holder and a switch.

Step 2: Brain

Now we need something that will orchestrate our motors, a micro-controller. From the title of this instructable you probably have already understood that we are going to use Arduino.

<u>Why?</u>

Well, the answer is simple. Arduino (or, specifically, Arduino Uno) is a compact, comfortable and relatively cheap micro-controller. In addition, it is very common and you can get it literally everywhere online.

Step 3: Attaching the Arduino to the Chassis

Now simply use some screws to attach the Arduino to the chassis. If you've made your own chassis/ can't find any screws, you can use a small amount of double-sided tape. If you do choose to use double-sided tape then search for an area under the Arduino that doesn't have many pins/solder above it.

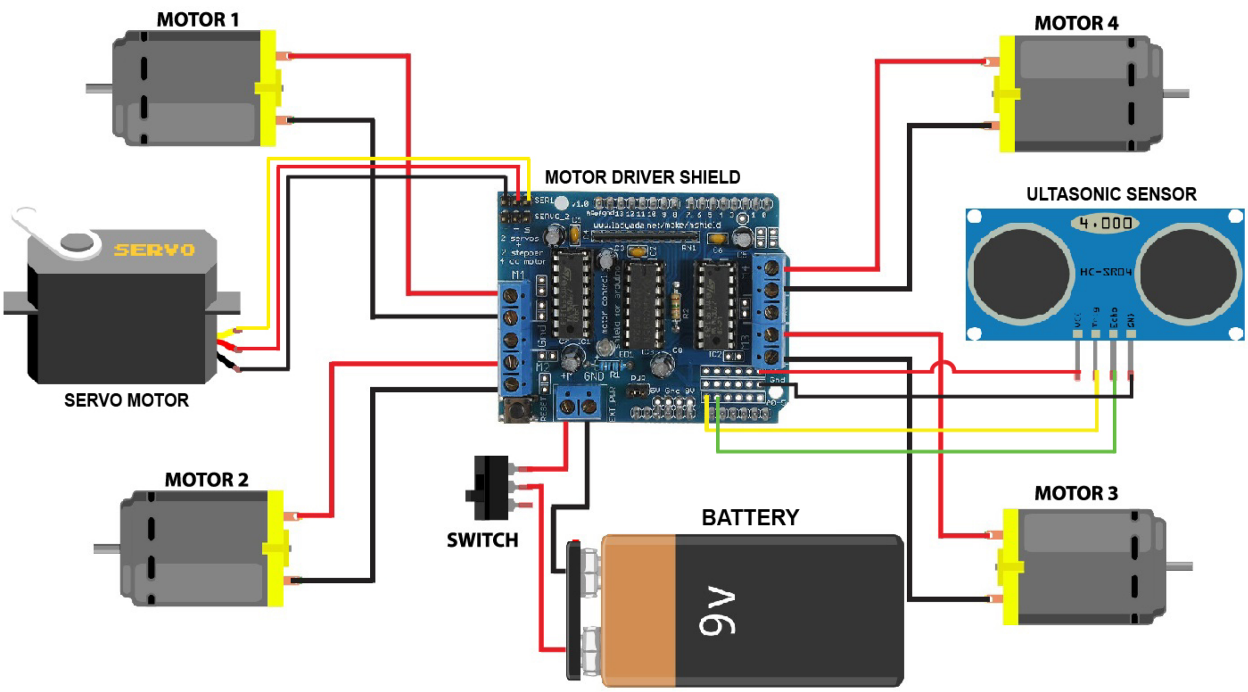
Step 4: Connecting and Controlling the Motors With the Arduino

Arduino boards can't control dc motors by their own, because the currents they are generating are too low. Moreover, the currents they are generating cannot be reversed, so you can't change the direction of the motor. To solve that we will use a motor driver, which helps the Arduino control dc motors. The most comfortable way of using motor drivers is through shields. Motor shields are very cheap. Take your motor shield and simply plug it into the Arduino with the sensor's wires crunching inside.

Step 5: Connecting the Motors to the Shield

Every Motor Shield has (at least) two channels, one for the motors, and one for a power source. Locate the channels and plug your chassis' motors to the motors' channel, and whatever power source you are using (e.g. AA batteries, Lithium Battery) to the power source's channel.

Step 6: Connect Servo Motor & Ultrasonic Sensor Robot



CONNECTION:

1.Connect Left Motors with Motor Shield M1 and M2 ports

2.Connect Right Motors with Motor Shield M3 and M4 ports

3.Connect Battery Holder with Motor Shield M+ and GND ports

4.Connect Servo Motor with Motor Shield Ser1 port

5.Connect Ultrasonic Sensor GND with Motor Shield GND port

6.Connect Ultrasonic Sensor VCC with Motor Shield 5V port

7.Connect Ultrasonic Sensor ECHO with Motor Shield A1 port

8.Connect Ultrasonic Sensor TRIG with Motor Shield A0 port

Step 7: How to Upload Code into the robot

Arduino Coding Requirements:

For Arduino Coding we need Arduino IDE and Library.

1. Arduino IDE

2. Adafruit Motor Shield Library - https://learn.adafruit.com/adafruit-motor-shield/library-install

3. Arduino New Ping Library - https://github.com/livetronic/Arduino-NewPing

4. Arduino Servo Library - https://github.com/arduino-libraries/Servo.git

5. To Install the libraries go to sketch >> Include Library >> Add .ZIP File >> Select the Downloaded ZIP files From the Above links

Raw Code:

```
#include <AFMotor.h>

#include <NewPing.h>

#include <Servo.h>


#define TRIG_PIN A0

#define ECHO_PIN A1

#define MAX_DISTANCE 200

#define MAX_SPEED 190 // sets speed of DC  motors

#define MAX_SPEED_OFFSET 20


NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);
```

```
AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);
Servo myservo;

boolean goesForward=false;
int distance = 100;
int speedSet = 0;

void setup() {

  myservo.attach(10);
  myservo.write(115);
  delay(2000);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
}

void loop() {
 int distanceR = 0;
 int distanceL =  0;
```

```
delay(40);

if(distance<=15)
{
 moveStop();
 delay(100);
 moveBackward();
 delay(300);
 moveStop();
 delay(200);
 distanceR = lookRight();
 delay(200);
 distanceL = lookLeft();
 delay(200);

 if(distanceR>=distanceL)
 {
   turnRight();
   moveStop();
 }else
 {
   turnLeft();
   moveStop();
 }
}else
{
 moveForward();
}
```

```
 distance = readPing();
}

int lookRight()
{
    myservo.write(50);
    delay(500);
    int distance = readPing();
    delay(100);
    myservo.write(115);
    return distance;
}

int lookLeft()
{
    myservo.write(170);
    delay(500);
    int distance = readPing();
    delay(100);
    myservo.write(115);
    return distance;
    delay(100);
}

int readPing() {
  delay(70);
  int cm = sonar.ping_cm();
  if(cm==0)
```

```
  {
    cm = 250;
  }
  return cm;
}

void moveStop() {
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
  }

void moveForward() {

 if(!goesForward)
  {
    goesForward=true;
    motor1.run(FORWARD);
    motor2.run(FORWARD);
    motor3.run(FORWARD);
    motor4.run(FORWARD);
   for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly bring the
speed up to avoid loading down the batteries too quickly
   {
    motor1.setSpeed(speedSet);
    motor2.setSpeed(speedSet);
    motor3.setSpeed(speedSet);
```

```
    motor4.setSpeed(speedSet);
    delay(5);
   }
  }
}

void moveBackward() {
    goesForward=false;
    motor1.run(BACKWARD);
    motor2.run(BACKWARD);
    motor3.run(BACKWARD);
    motor4.run(BACKWARD);
  for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly bring the
speed up to avoid loading down the batteries too quickly
  {
    motor1.setSpeed(speedSet);
    motor2.setSpeed(speedSet);
    motor3.setSpeed(speedSet);
    motor4.setSpeed(speedSet);
    delay(5);
  }
}

void turnRight() {
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
```

```
  delay(500);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}

void turnLeft() {
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
  delay(500);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}
```

Arduino Coding Upload:

1. Connect the Robot with PC via Arduino Wire

2. Click Upload In the Arduino IDE

Discussion:

1. Rechargeable Battery is Preferable

2. Ultrasonic Sensor is not perfect

3. Use More Solid Board

4. Use Safety Gadgets

ADVANTAGES:

1. Reduce Accidents

2. Helpful for auto Driving

3. Helpful for old drivers whose nerve is weak for instant reaction

4. Helpful for voice Control Driving

REFERENCE:

1. Arduino Official Site-
https://create.arduino.cc/projecthub/chandankumarcmsn/obstacle-avoiding-robot-using-arduino-with-ultrasonic-sensor-af66f1

2. Electronics HUB- https://www.electronicshub.org/obstacle-avoiding-robot-arduino/

3. Instructables Site- https://www.instructables.com/id/Obstacle-Avoiding-Robot-Arduino-1/