For everyone who contributed to the publication of this book

Table of Contents

Introduction to Book

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

mBlock is a STEAM programming software tool designed for programming for kids. It is developed based on Scratch 3.0 and Arduino code. It supports block-based and text-based programming languages.

mBlock also provides software programming services, software design services and maintenance of computer software services in the education of programming for those who want to promote their programming abilities.

With mBlock, children can not only create games and animations by dragging blocks or using Python code, but can also code robots or boards to do anything they can imagine. And mBlock exposes children to cutting-edge technologies, allowing children to create projects with technologies like AI and IoT. Moreover, in the mBlock Community, children are able to share projects and learn from the like-minded.

In this book, Arduino will be programmed using Scratch using **mblock**. In this book, it is useful for those who want to program Arduino by using simple programming that is simpler than ordinary programming languages, using codes, but here blocks will be used, which are easier.

INTRODUCTION

Introduction

Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Why Arduino?

- Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove

chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

- There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:
- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
- **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical

details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

- **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

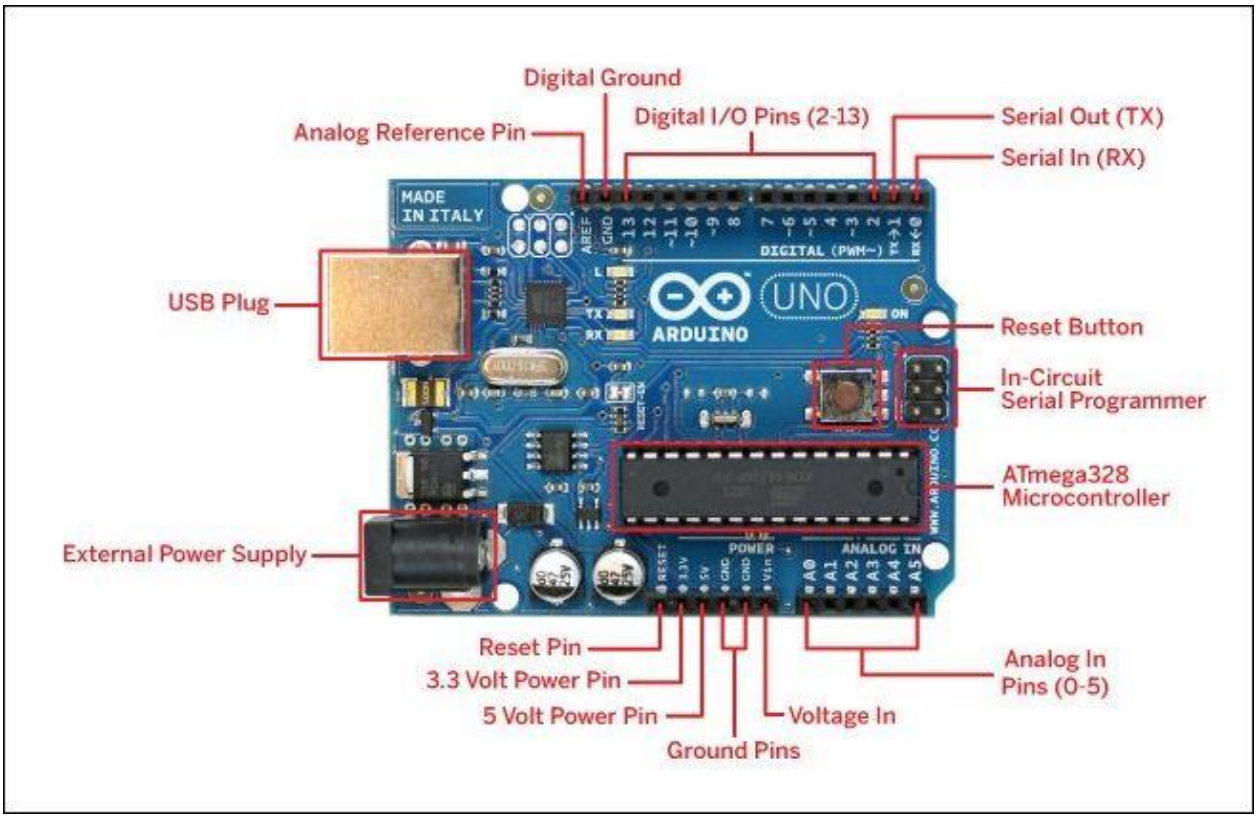
Types of Arduino Boards

There are different Arduino boards which are following

- Arduino UNO (R3)
- LilyPad Arduino
- Red Board
- Arduino Mega (R3)
- Arduino Leonardo

In this book we use the Arduino Uno (R3) type.

Introduction



Specifications and Features:

- Model Type: UNO Rev R3
- Microcontroller Chip: ATmega328
- Color: Blue
- Input Voltage(Recommended): 7-12V
- Input Voltage (limit): 6-20V
- Operating Voltage: 5
- Analog I/O Pins: 6
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- PWM Digital I/O Pins: 6
- DC Current per I/O Pin: DC Current per I/O Pin: 40 mA; DC Current for 3.3V Pin: 50 mA
- Clock Speed: 16 MHz
- EEPROM: 1 KB (ATmega328)
- Flash Memory: 32 KB (ATmega328) of which 0.5 KB used by bootloader
- PWM Output Pins: 6
- SRAM: 2 KB (ATmega328)

Introduction

- Power Supply Option: DC Jack or USB
- On Board LEDs: Yes
- Operating Temperature (°C): -40 to +90
- Dimensions in mm (LxWxH): 75 x 54 x 12
- Weight (gm): 26

Introduction

Scratch

Scratch is a block-based visual programming language and website targeted primarily at children. Users of the site can create online projects using a block-like interface. The service is developed by the MIT Media Lab, has been translated into 70+ languages, and is used in most parts of the world. Scratch is taught and used in after-school centers, schools, and colleges, as well as other public knowledge institutions. As of April 2020, community statistics on the language's official website show more than 52 million projects shared by over 54 million users, and almost 55 million monthly website visits.

Scratch takes its name from a technique used by disk jockeys called "scratching", where vinyl records are clipped together and manipulated on a turntable to produce different sound effects and music. Like scratching, the website lets users mix together different media (including graphics, sound, and other programs) in creative ways by creating and remixing projects, like video games and animations.

You can download the scratch program from the scratch website https://scratch.mit.edu/

In figure 1 that shown the scratch program.

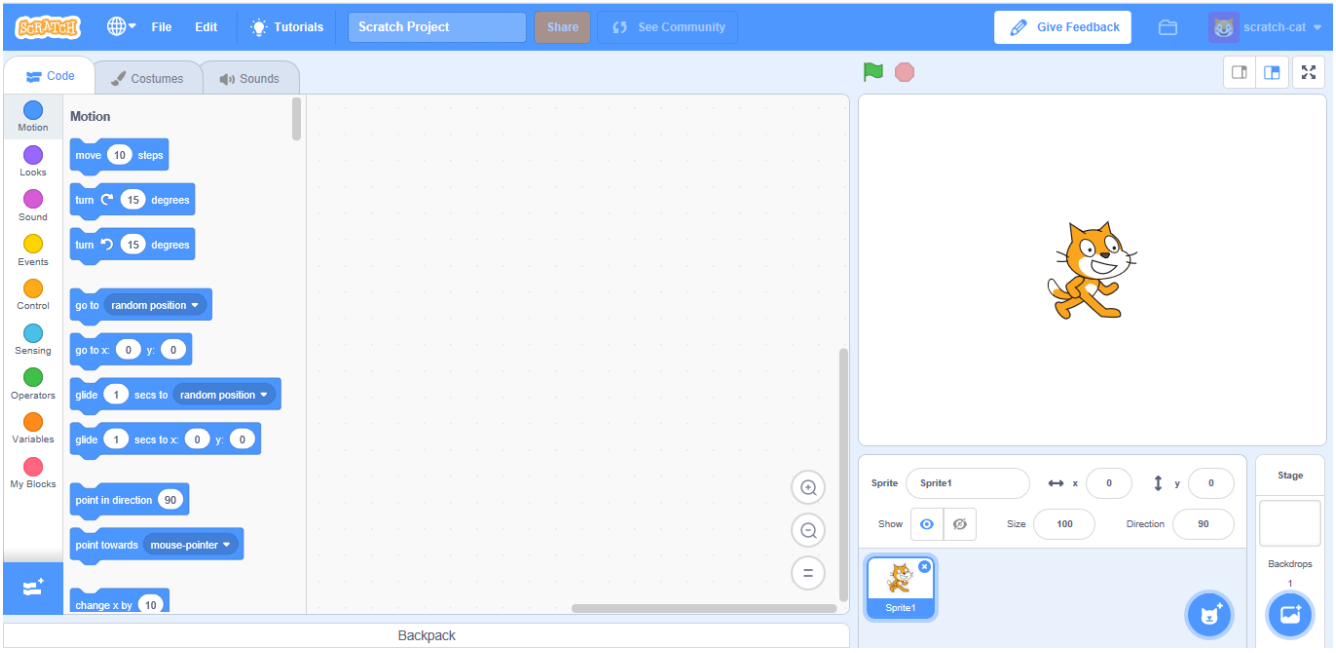


Figure 1: scratch program

Introduction

To program Arduino by Scratch, we will use a program called **mblock V3**.

What is mBlock?

mBlock is a STEAM programming software tool designed for programming for kids. It is developed based on Scratch 3.0 and Arduino code. It supports block-based and text-based programming languages.

mBlock also provides software programming services, software design services and maintenance of computer software services in the education of programming for those who want to promote their programming abilities.

With mBlock, children can not only create games and animations by dragging blocks or using Python code, but can also code robots or boards to do anything they can imagine. And mBlock exposes children to cutting-edge technologies, allowing children to create projects with technologies like AI and IoT. Moreover, in the mBlock Community, children are able to share projects and learn from the like-minded.

1.**Based on Scratch 3.0,start programming easily.**

Scratch is a programming tool developed by MIT and is credited as the most globally influential programming language for children. Based on Scratch 3.0 and Arduino code, mBlock is versatile and user-friendly enough to offer you whatever Scratch can give. And it's quite easy to pick up because you can code simply by dragging and dropping blocks.

2. **Go to Python with one-click.**

With **mBlock** , users can easily switch to Python with one-click. After students grasp how to program with blocks, they can effortlessly move on to Python. You don't have to switch between software because mBlock allows you to write code directly in its Python editor. The transition is just seamless.

3.**A mix of software and robots makes it fun to learn coding.**

mBlock allows users to program robots to do anything they can imagine. By showing the outcomes of coding in the physical world, we attempt to keep students engaged in learning

code and bring them senses of fulfillment. Also, mBlock brings diversity into classrooms because it has the power that enables educators and students to turn different ideas into reality.

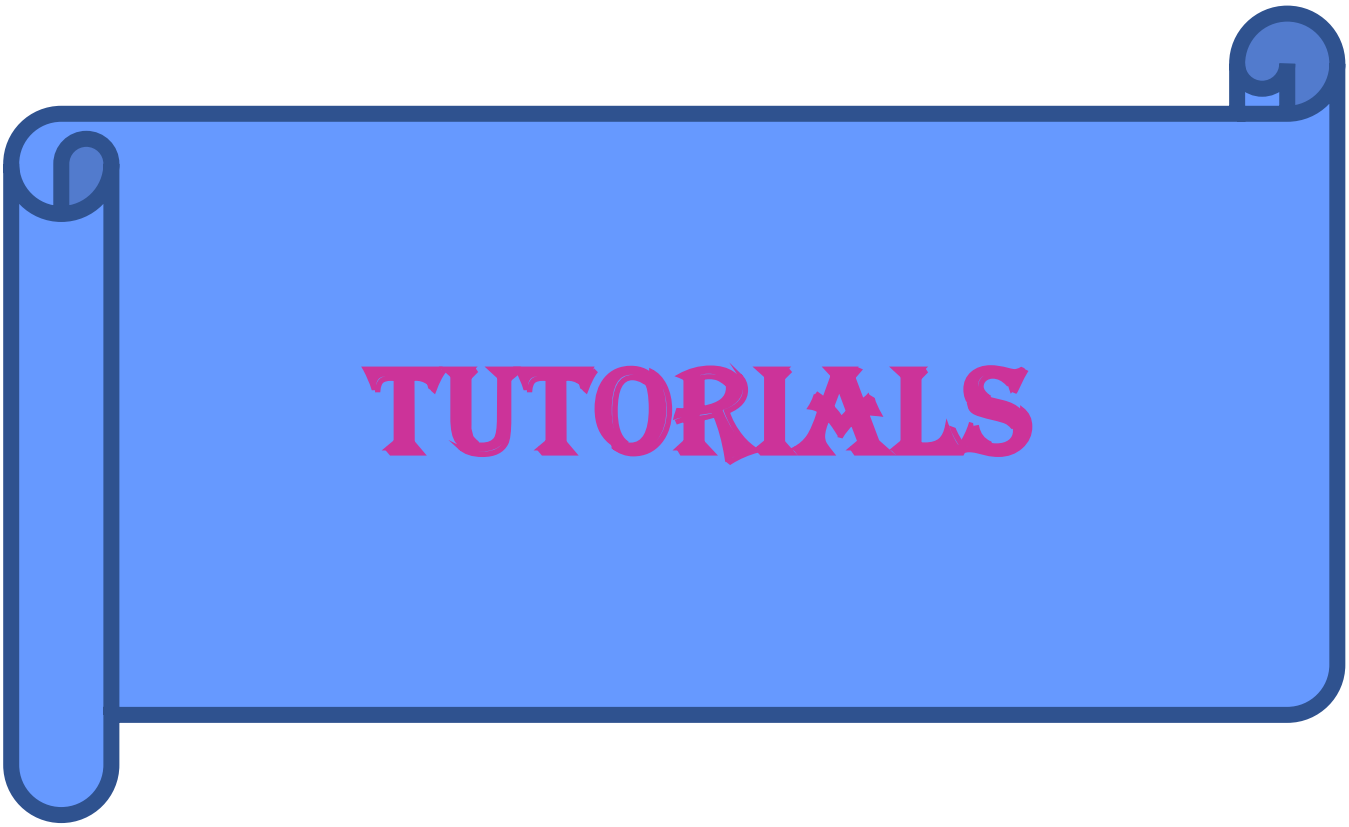
4. **Give your creativity an edge with cutting-edge technologies, like AI .**

mBlock integrates Microsoft cognitive services and Google deep learning into one tool. With these features, children can program to add more capabilities to **mBlock**, like age guessing or playing rock-paper-scissors games. We just hope to make it easier for children to master the fundamentals of **AI**.

5. **Create in a physical world with IoT applications.**

mBlock comes with the cloud service that is designed for IoT teaching. By working with robots or electronic modules, you can take advantage of the feature to create fun projects, like Weather Report, Autonomous Plant Watering Robot and Smart Lighting. For students, the best way to learn about IoT is to see how it works in real life. In figure 2 that shown the **mBlock V3** program.

Figure 2: mBlock V3


TUTORIALS

Tutorials

In this book simple projects will be made using the Arduino with sensors, drivers and displays.

Therefore, you must have prior knowledge of Arduino, sensors, drivers, etc. in order to understand these tutorials.

Tutorials

Tutorial1# LED Flasher

Experiment Object

In this experiment learn how to program the led flasher by using scratch language.

Theory

Led

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor.[5] White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device.

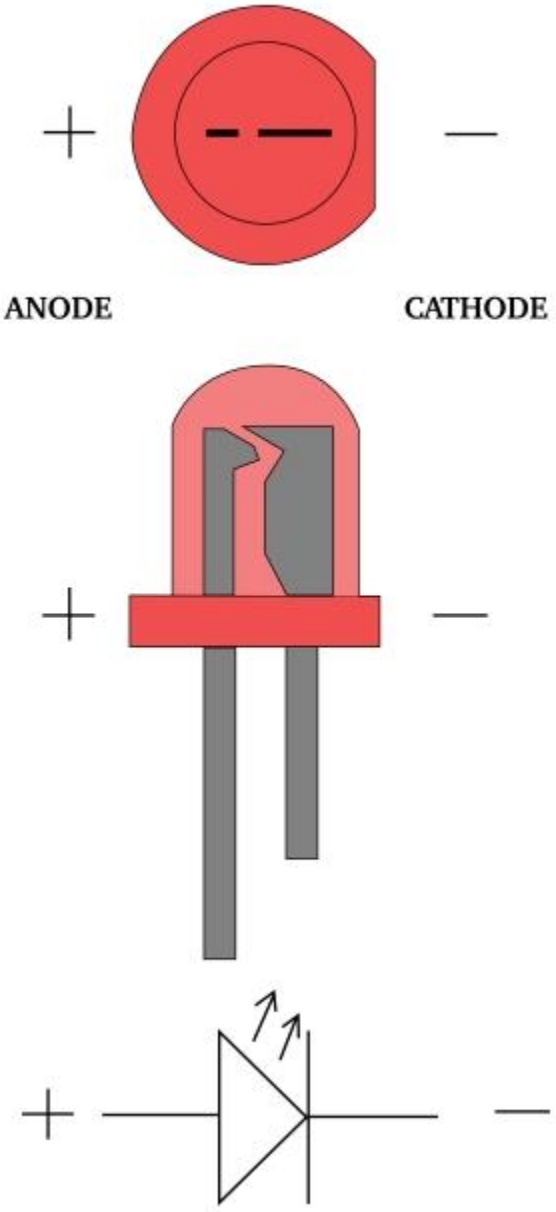
Appearing as practical electronic components in 1962, the earliest LEDs emitted **low-intensity infrared (IR) light**. Infrared LEDs are used in remote-control circuits, such as those used with a wide variety of consumer electronics. The first visible-light LEDs were of low intensity and limited to red. Modern LEDs are available across the visible, ultraviolet (UV), and infrared wavelengths, with high light output.

Early LEDs were often used as indicator lamps, replacing small incandescent bulbs, and in seven-segment displays. Recent developments have produced high-output white light LEDs suitable for room and outdoor area lighting. LEDs have led to new displays and sensors, while their high switching rates are useful in advanced communications technology.

LEDs have many advantages over incandescent light sources, including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. LEDs are used in applications as diverse as aviation lighting, automotive

headlamps, advertising, general lighting, traffic signals, camera flashes, lighted wallpaper, horticultural grow lights, and medical devices.

Unlike a laser, the light emitted from an LED is neither spectrally coherent nor even highly monochromatic. However, its spectrum is sufficiently narrow that it appears to the human eye as a pure (saturated) color. Also, unlike most lasers, its radiation is not spatially coherent, so it cannot approach the very high brightness's characteristic of lasers.



Tutorials

Components

1- Arduino UNO R3.

2- Led.

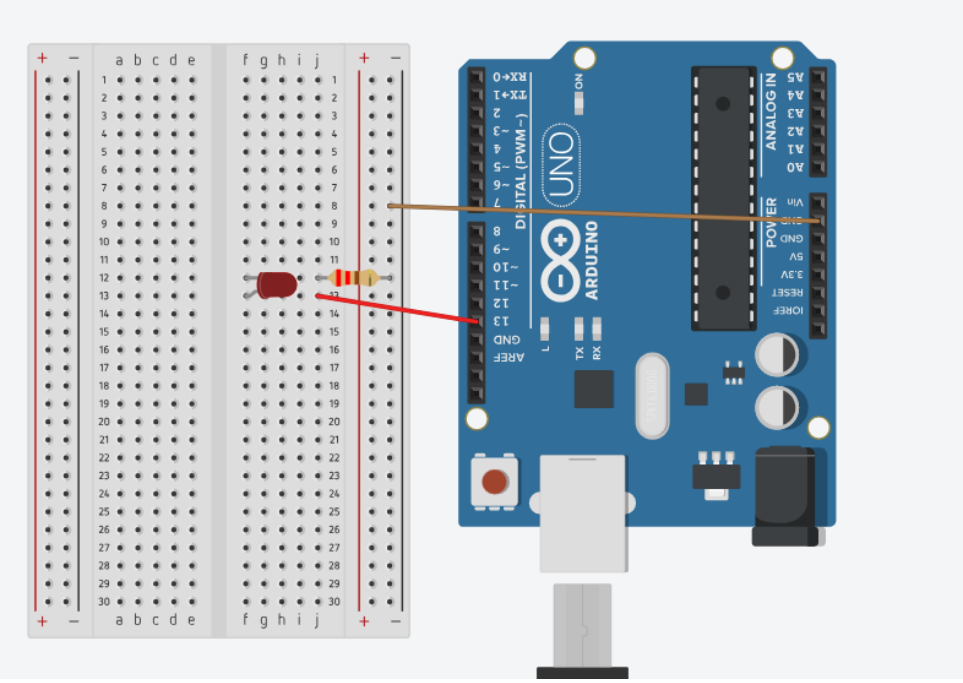
3- Resistor 330 ohm.

4- Wires.

5- Breadboard.

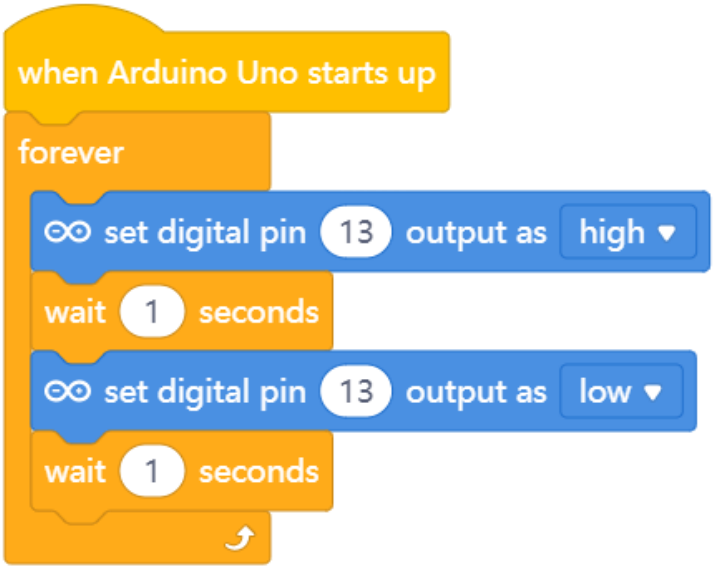
Procedure

CIRCUIT DESIGN



Tutorials

CODE



Tutorials

Tutorial 2# Five LEDS Flasher

Experiment Object

In this experiment learn how to program the five leds flasher by using scratch language.

Components

1- Arduino UNO R3.

2- Led x5.

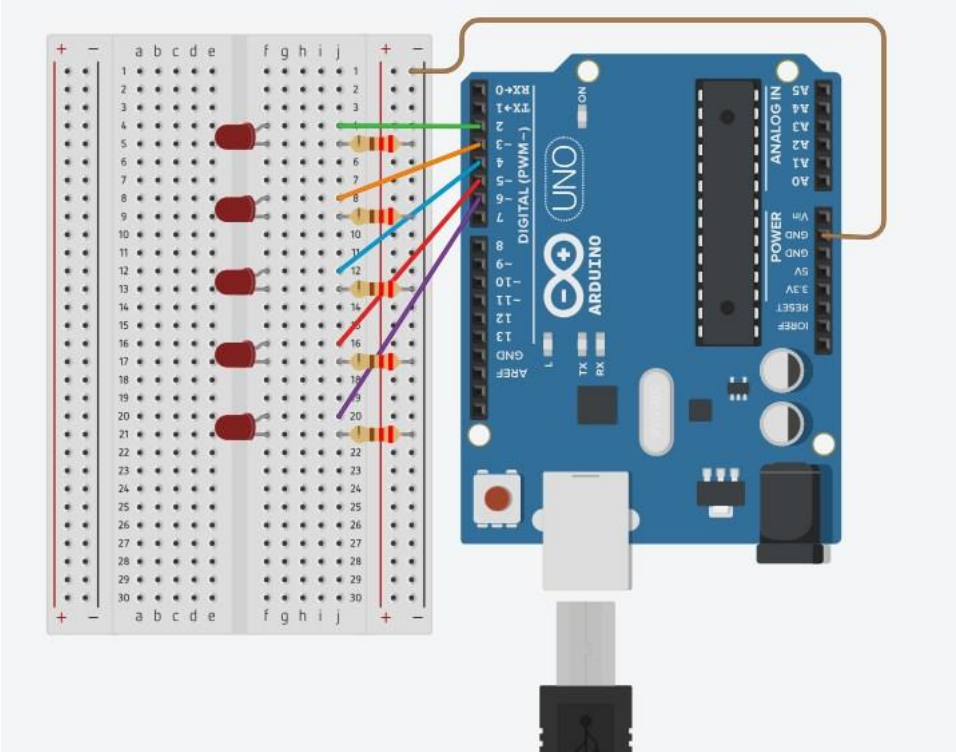
3- Resistors 330-ohm x5.

4- Wires.

5- Breadboard.

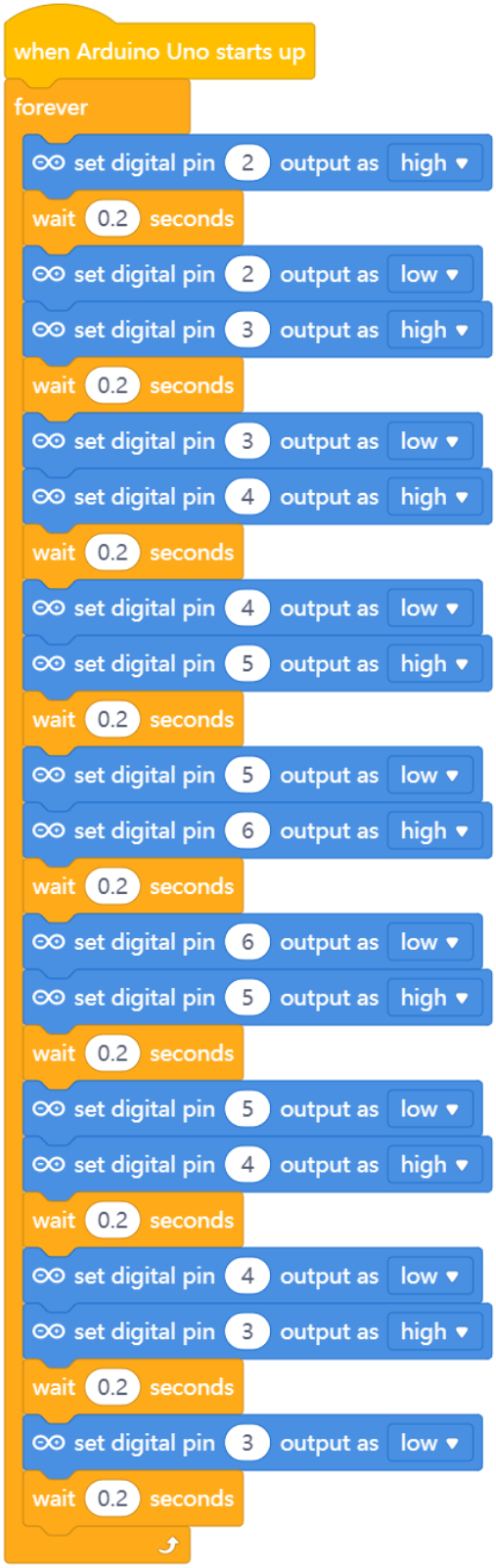
Procedure

CIRCUIT DESIGN



Tutorials

CODE



Tutorials

Tutorial 3# Buzzer

Experiment Object

In this experiment learn how to program the **buzzer** by using scratch language.

Theory

Piezoelectric Sounders / Buzzers

Piezoelectric Sounders / Buzzers are sound components prepared by incorporating a **piezoelectric vibration** plate in a plastic case (resonator).

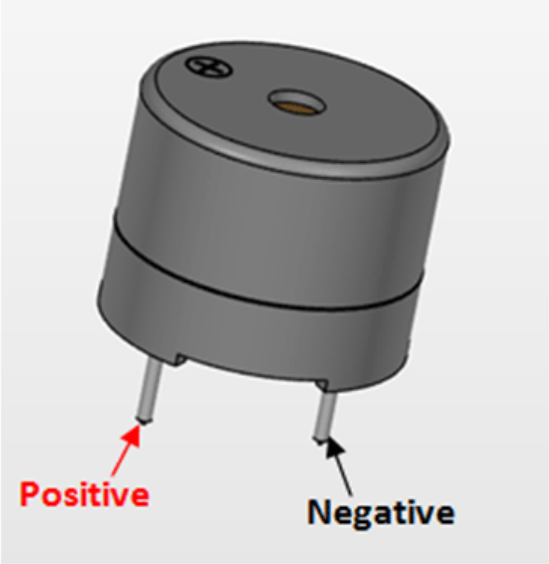
Piezoelectric sounders are sound components which generate sound suitable for use as input signals (including multi-tone, melody and so forth) without built-in oscillation circuits.

This characteristic allows them to be used in a wide range of applications. They come as the **SMD** type, which is optimal for small, **high-density** mounting and the pin type, which can be used for general purposes.

Piezoelectric buzzers are sound components which generate a monotone using a built-in oscillation circuit.

Tutorials

Buzzer Pins



Components

1- Arduino UNO R3.

2- Buzzer.

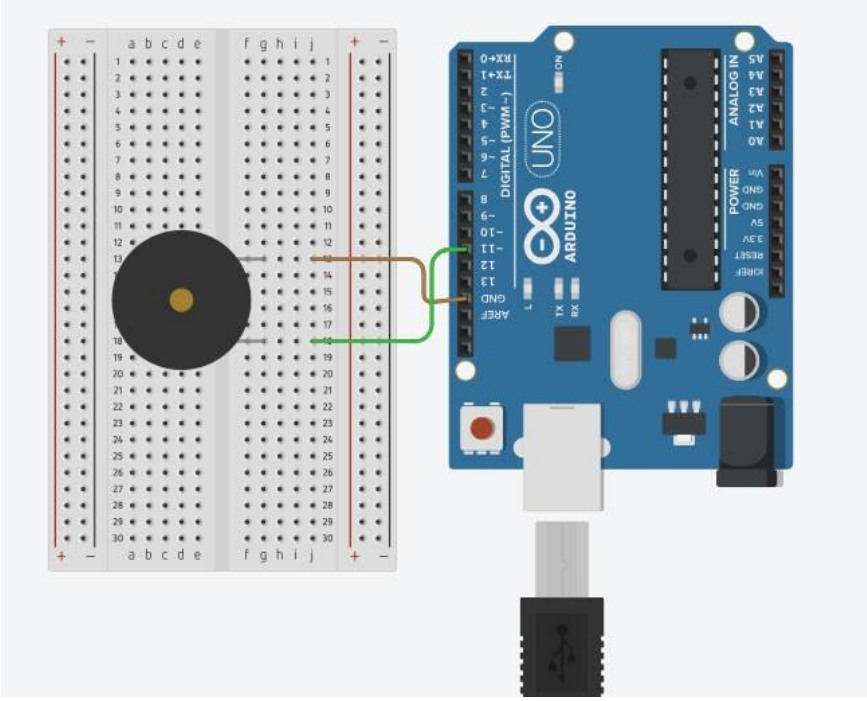
3- Wires.

4- Breadboard.

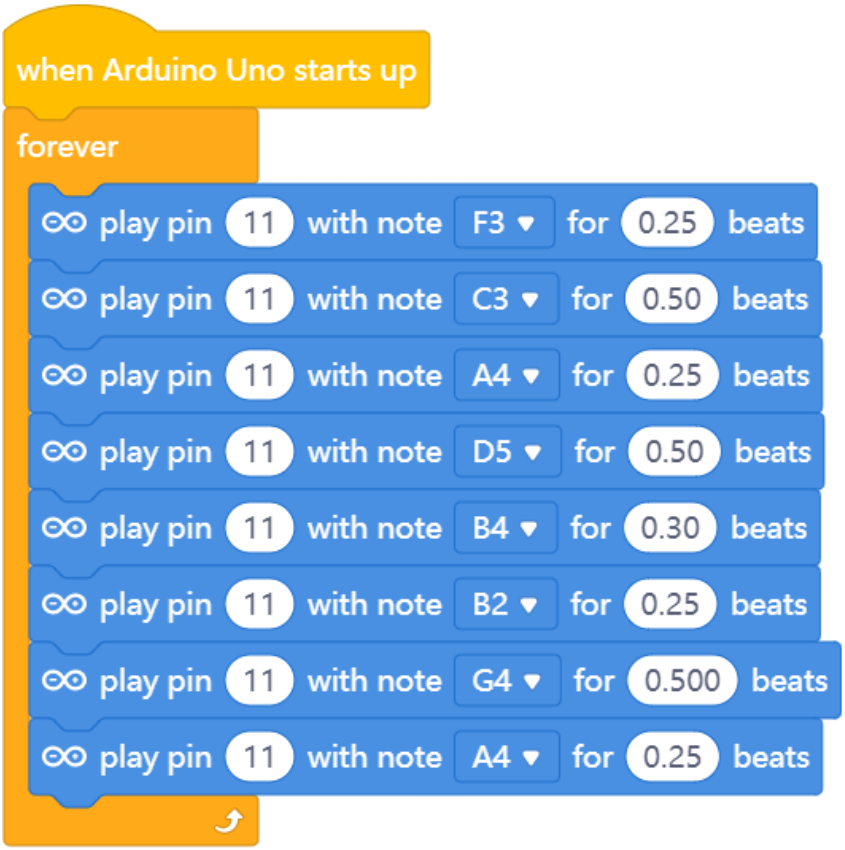
Procedure

CIRCUIT DESIGN

Tutorials



CODE



Tutorials

Tutorial 4# Button

Experiment Object

In this experiment learn how to program the button by using scratch language.

Theory

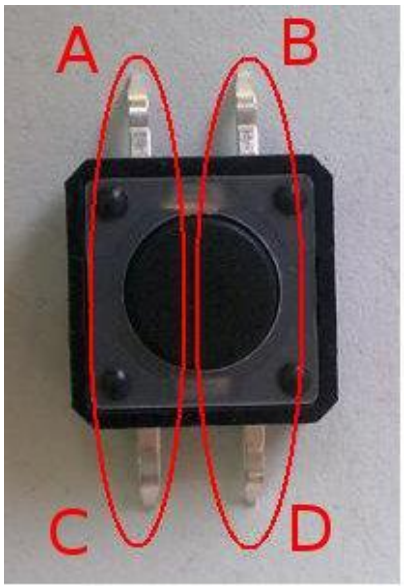
Button

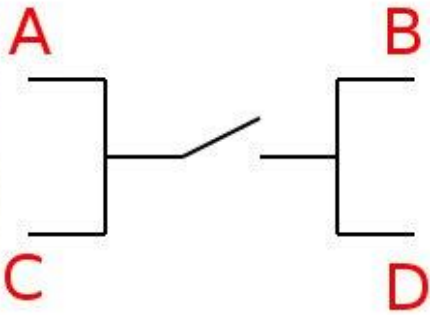
The pushbutton is a component that connects two points in a circuit when you press it. The example turns on an LED when you press the button.

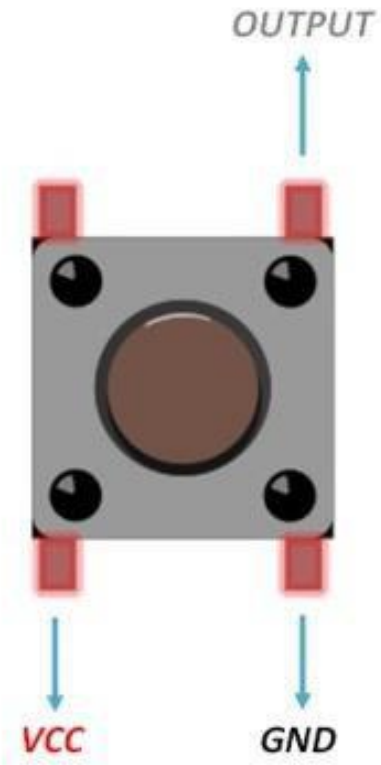
We connect three wires to the Arduino board. The first goes from one leg of the pushbutton through a pull-up resistor (here **2.2 KOhms)** to the **5 volt** supply. The second goes from the corresponding leg of the pushbutton to ground. The third connects to a digital i/o pin which reads the button's state.

When the **pushbutton** is open (impressed) there is no connection between the two legs of the pushbutton, so the pin is connected to **5 volts** (through the pull-up resistor) and we read a HIGH. When the button is closed (pressed), it makes a connection between its two legs, connecting the pin to ground, so that we read a LOW. (The pin is still connected to **5 volts**, but the resistor in-between them means that the pin is "closer" to ground.)

Tutorials







Components

1- Arduino UNO R3.

2- Button.

3- Resistors 330 ohm.

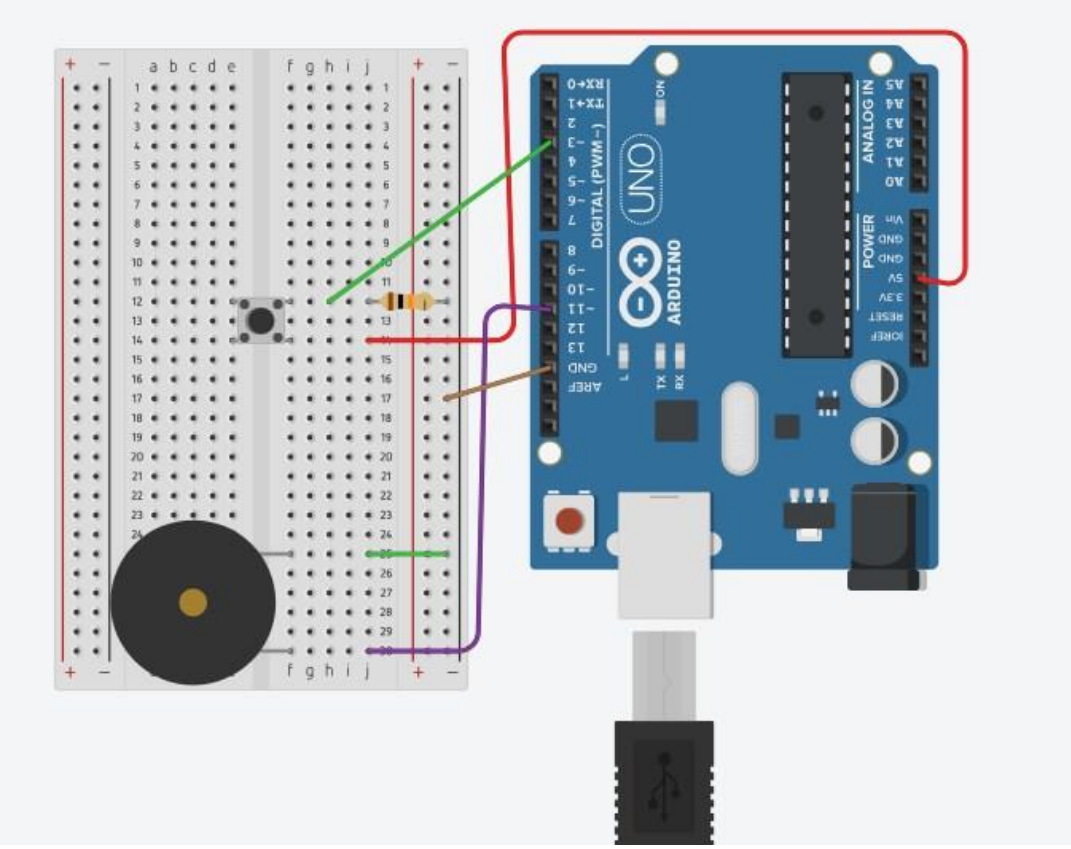
4- Buzzer.

5- Wires.

6- Breadboard.

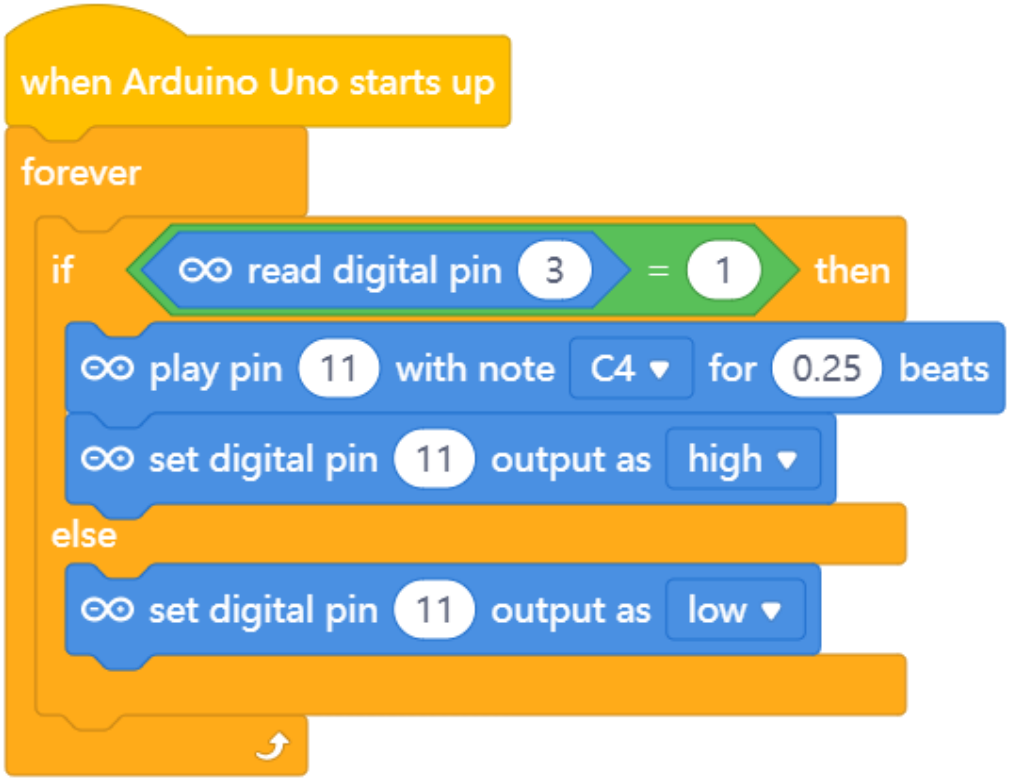
Procedure

CIRCUIT DESIGN



Tutorials

CODE



Tutorials

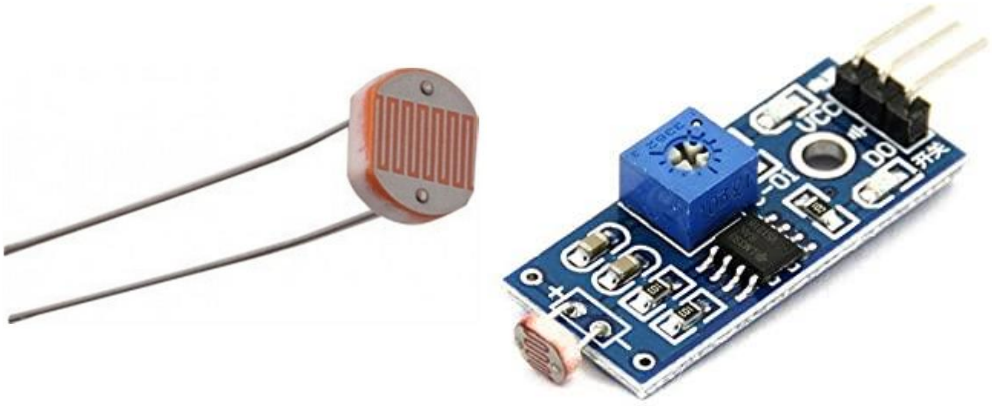
Tutorial 5# LDR

Experiment Object

In this experiment learn how to program the **LDR** by using scratch language.

Theory

LDR Sensor



A **Light Sensor** generates an output signal indicating the intensity of light by measuring the radiant energy that exists in a very narrow range of frequencies basically called “light”, and which ranges in frequency from “Infra-red” to “Visible” up to “Ultraviolet” light spectrum.

The **light sensor** is a passive device that convert this “light energy” whether visible or in the infra-red parts of the spectrum into an electrical signal output. Light sensors are more commonly known as “Photoelectric Devices” or “Photo Sensors” because the convert light energy (photons) into electricity (electrons).

Photoelectric devices can be grouped into two main categories, those which generate electricity when illuminated, such as **Photo-voltaics** or **Photo-emissives** etc, and those

which change their electrical properties in some way such as Photo-resistors or Photo-conductors. This leads to the following classification of devices.

• **Photo-emissive Cells** – These are photodevices which release free electrons from a light sensitive material such as cesium when struck by a photon of sufficient energy. The amount of energy the photons have depends on the frequency of the light and the higher the frequency, the more energy the photons have converting light energy into electrical energy.

• **Photo-conductive Cells** – These photodevices vary their electrical resistance when subjected to light. Photoconductivity results from light hitting a semiconductor material which controls the current flow through it. Thus, more light increases the current for a given applied voltage. The most common photoconductive material is Cadmium Sulphide used in LDR photocells.

• **Photo-voltaic Cells** – These photodevices generate an emf in proportion to the radiant light energy received and is similar in effect to photoconductivity. Light energy falls on to two semiconductor materials sandwiched together creating a voltage of approximately 0.5V. The most common photovoltaic material is Selenium used in solar cells.

• **Photo-junction Devices** – These photodevices are mainly true semiconductor devices such as the photodiode or phototransistor which use light to control the flow of electrons and holes across their PN-junction. Photojunction devices are specifically designed for detector application and light penetration with their spectral response tuned to the wavelength of incident light.

The Photoconductive Cell

A **Photoconductive** light sensor does not produce electricity but simply changes its physical properties when subjected to light energy. The most common type of photoconductive device is the ***Photoresistor*** which changes its electrical resistance in response to changes in the light intensity.

Tutorials

Photoresistors are Semiconductor devices that use light energy to control the flow of electrons, and hence the current flowing through them. The commonly used ***Photoconductive Cell*** is called the **Light Dependent Resistor** or **LDR**.

The Light Dependent Resistor

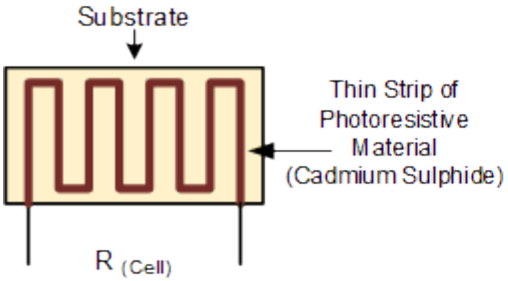
As its name implies, the **Light Dependent Resistor (LDR)** is made from a piece of exposed semiconductor material such as cadmium sulphide that changes its electrical resistance from several thousand Ohms in the dark to only a few hundred Ohms when light falls upon it by creating hole-electron pairs in the material.

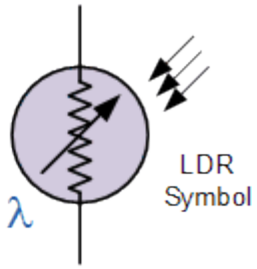
The net effect is an improvement in its conductivity with a decrease in resistance for an increase in illumination. Also, **photoresistive** cells have a long response time requiring many seconds to respond to a change in the light intensity.

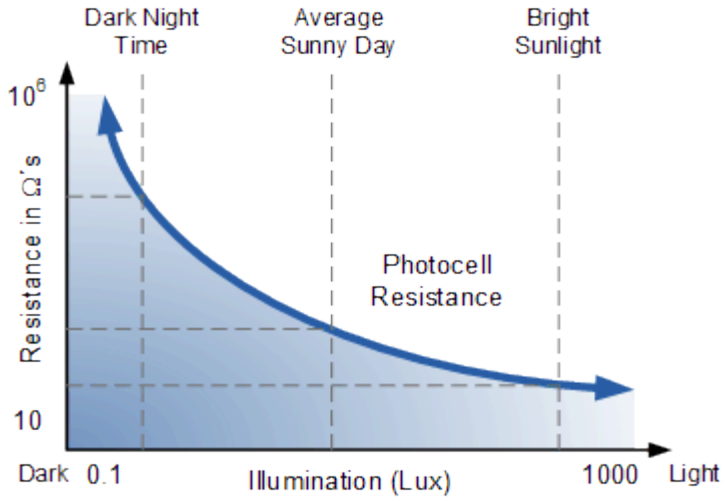
Materials used as the semiconductor substrate include, lead sulphide (PbS), lead selenide (PbSe), indium antimonide (InSb) which detect light in the infra-red range with the most commonly used of all photoresistive light sensors being Cadmium Sulphide (Cds).

Cadmium sulphide is used in the manufacture of photoconductive cells because its spectral response curve closely matches that of the human eye and can even be controlled using a simple torch as a light source. Typically, then, it has a peak sensitivity wavelength (**λp**) of about 560nm to 600nm in the visible spectral range.

The Light Dependent Resistor Cell



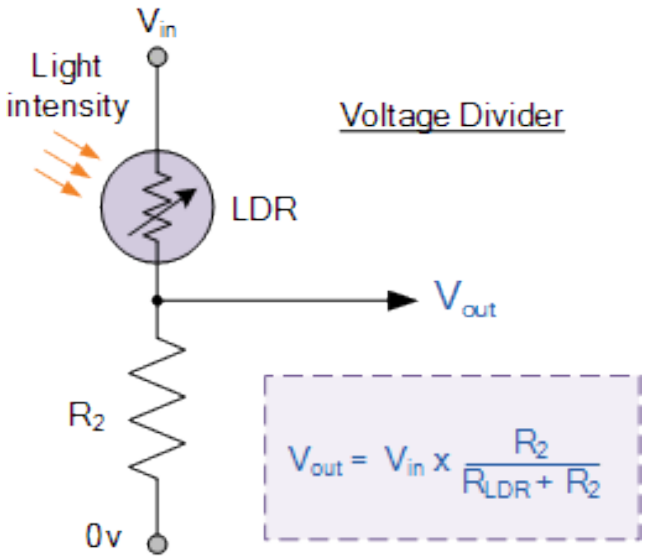




The most commonly used **photoresistive** light sensor is the ORP12 Cadmium Sulphide photoconductive cell. This **light dependent resistor** has a spectral response of about 610nm in the yellow to orange region of light. The resistance of the cell when unilluminated (dark resistance) is very high at about 10MΩ's which falls to about 100Ω's when fully illuminated (lit resistance).

To increase the dark resistance and therefore reduce the dark current, the resistive path forms a zigzag pattern across the ceramic substrate. The CdS photocell is a **very low-cost** device often used in auto dimming, darkness or twilight detection for turning the street lights "**ON**" and "**OFF**", and for photographic exposure meter type applications.

Tutorials



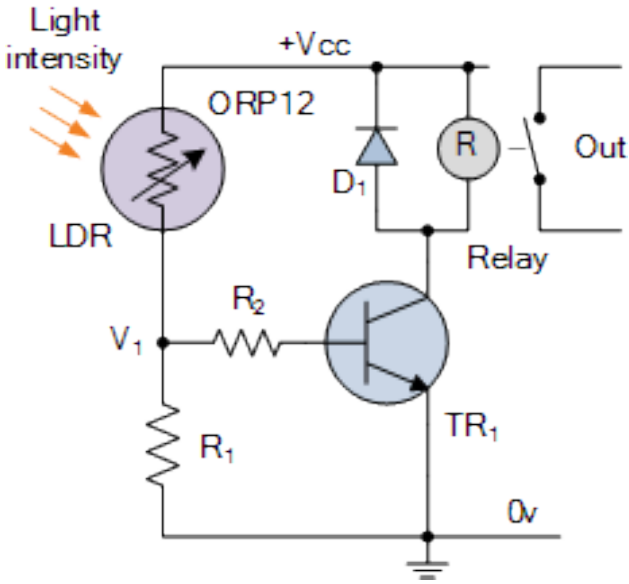
Connecting a light dependant resistor in series with a standard resistor like this across a single DC supply voltage has one major advantage, a different voltage will appear at their junction for different levels of light.

The amount of voltage drop across series resistor, R2 is determined by the resistive value of the light dependant resistor, RLDR. This ability to generate different voltages produces a very handy circuit called a “Potential Divider” or Voltage Divider Network.

As we know, the current through a series circuit is common and as the LDR changes its resistive value due to the light intensity, the voltage present at VOUT will be determined by the voltage divider formula. An LDR’s resistance, RLDR can vary from about 100Ω in the sun light, to over 10MΩ in absolute darkness with this variation of resistance being converted into a voltage variation at VOUT as shown.

One simple use of a Light Dependent Resistor, is as a light sensitive switch as shown below.

Tutorials



Components

1- Arduino UNO R3.

2- LDR.

3- Resistors 330 ohm.

4- LED.

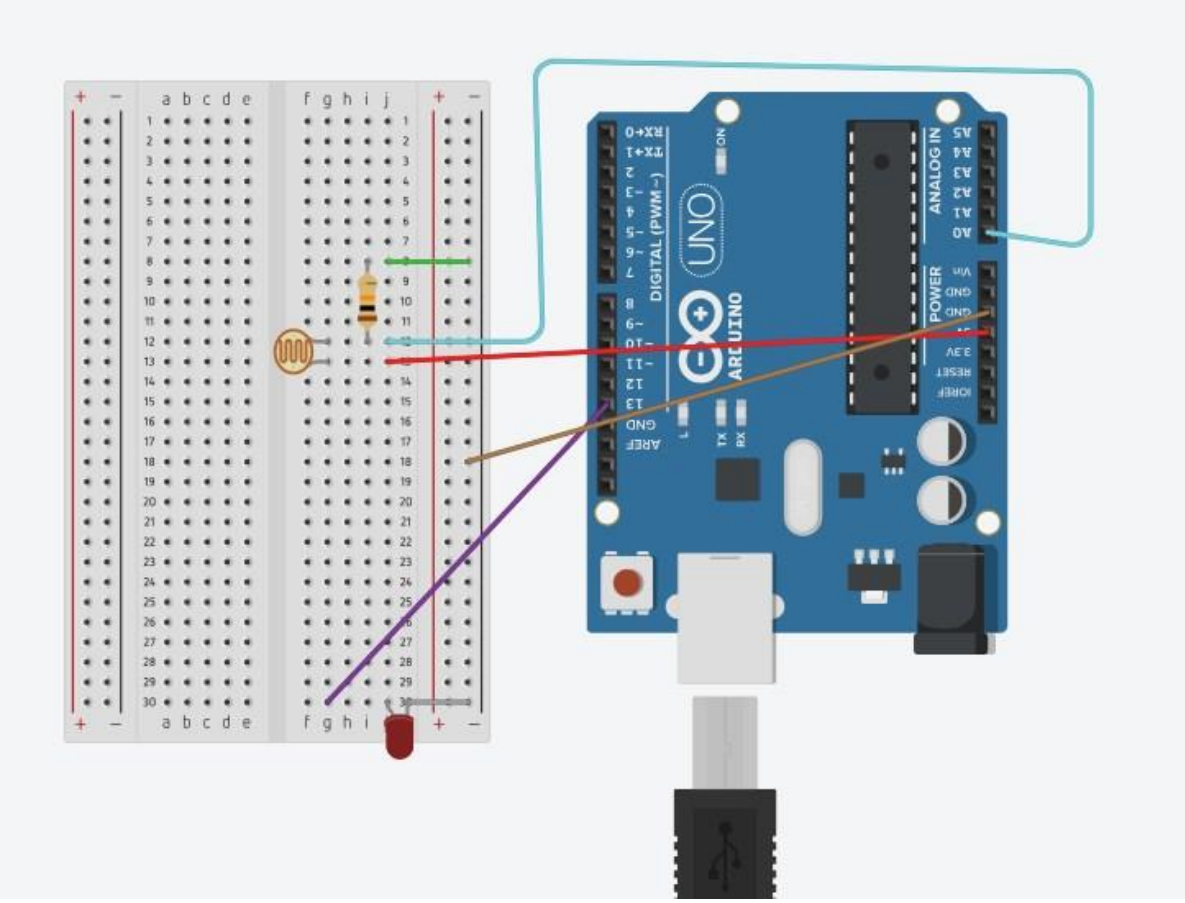
5- Wires.

6- Breadboard.

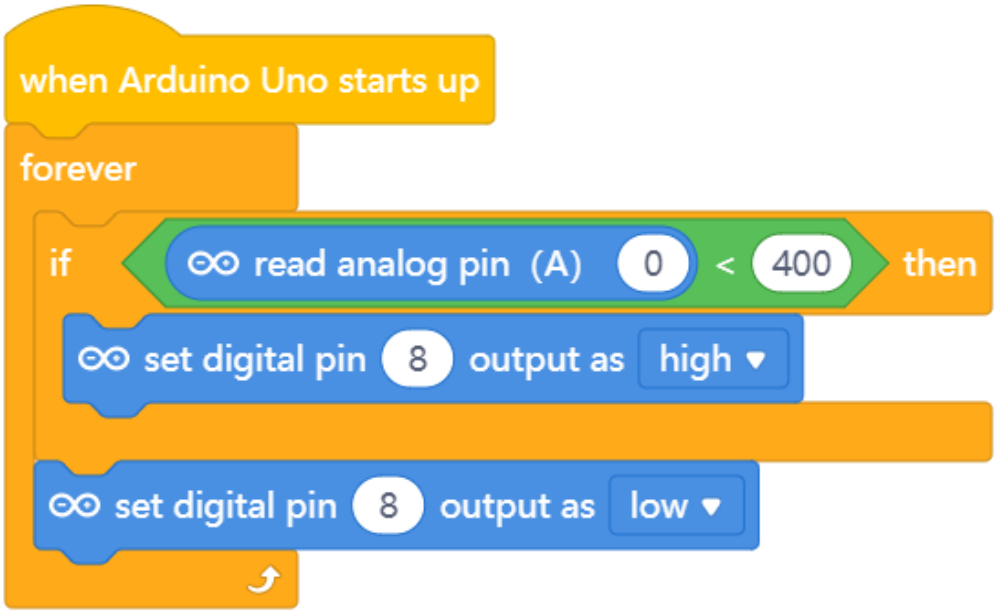
Procedure

CIRCUIT DESIGN

Tutorials



CODE



Tutorials

Tutorial 6# Police Leds Flasher

Experiment Object

In this experiment learn how to program the **police leds** flasher by using scratch language.

Components

1- Arduino UNO R3.

2- Resistors 330-ohm x2.

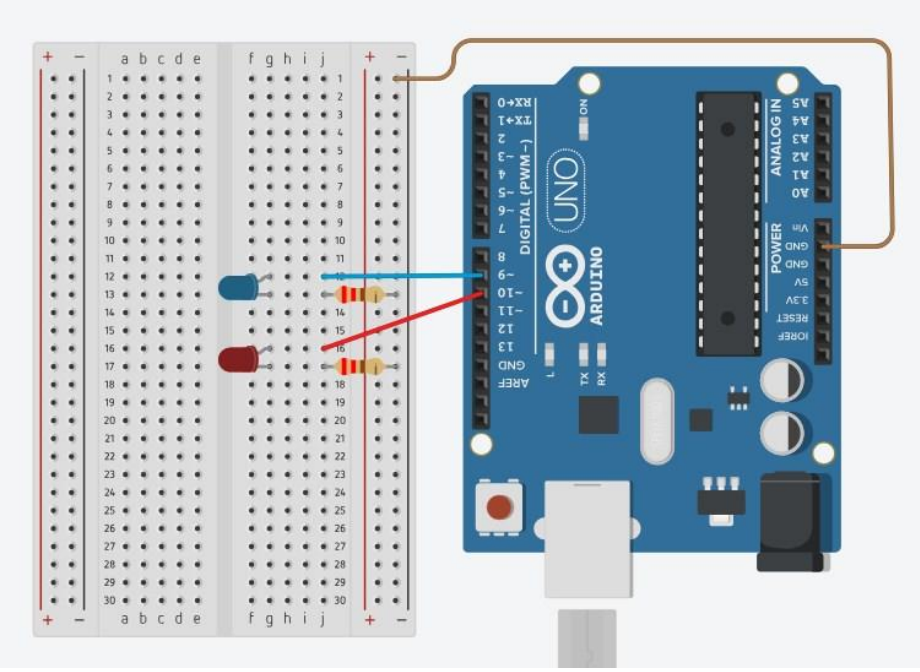
3- LED x2.

4- Wires.

5- Breadboard.

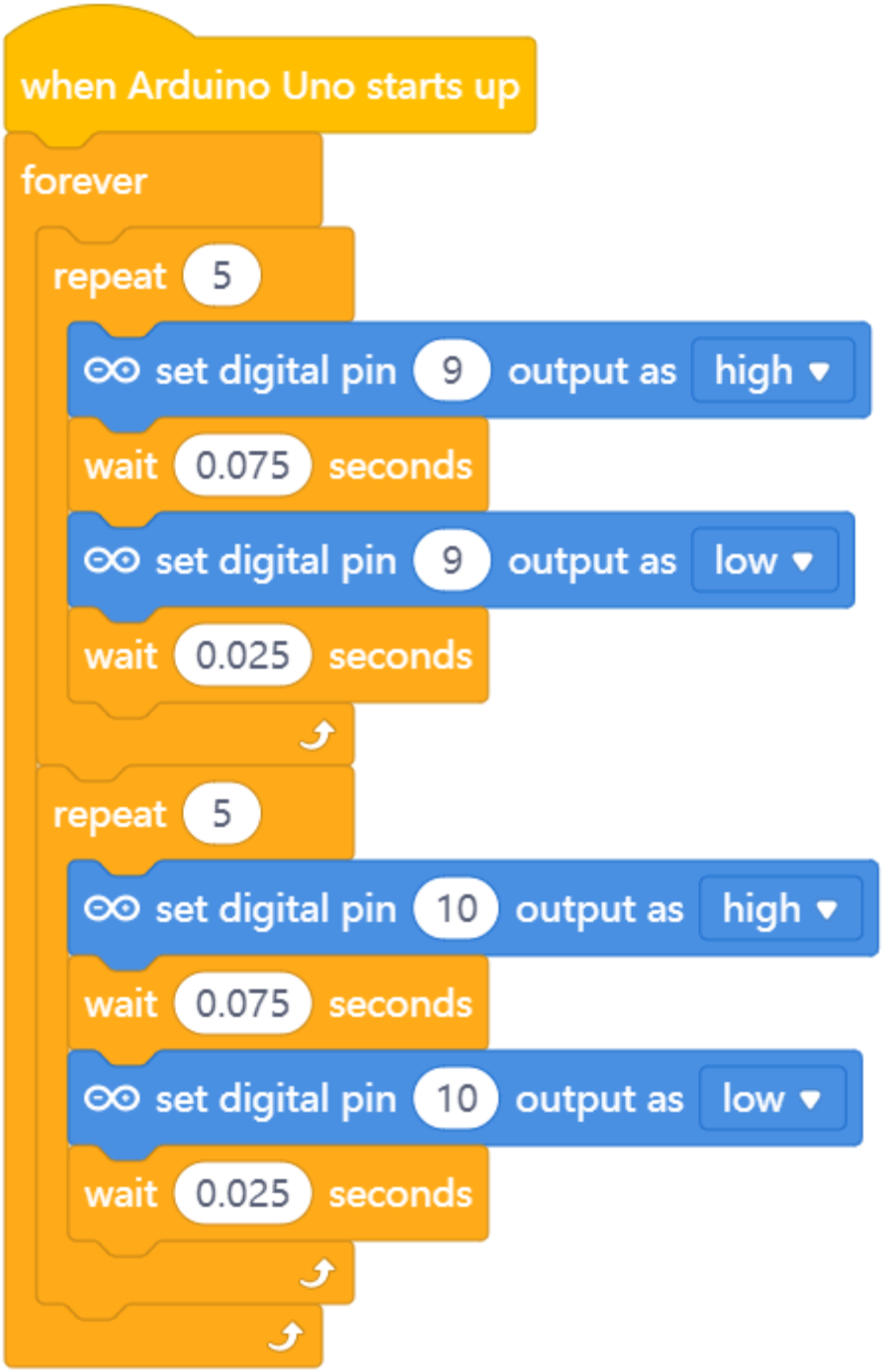
Procedure

CIRCUIT DESIGN



CODE

Tutorials


when Arduino Uno starts up
forever
repeat 5
set digital pin 9 output as high
wait 0.075 seconds
set digital pin 9 output as low
wait 0.025 seconds
repeat 5
set digital pin 10 output as high
wait 0.075 seconds
set digital pin 10 output as low
wait 0.025 seconds

Tutorials

Tutorial 7# Servo Motor

Experiment Object

In this experiment learn how to program the **servo motor** by using scratch language.

Theory

Servo Motor

What is a 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.

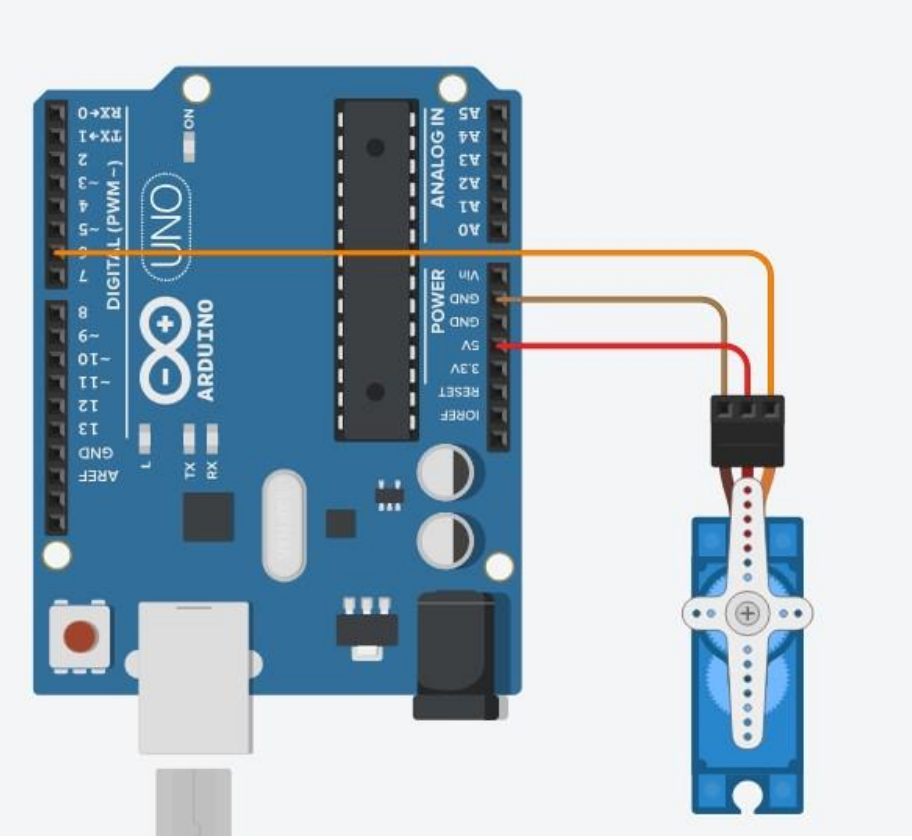
Components

1- Arduino UNO R3.
2- Servo Motor SG90.
3- Wires.
4- Breadboard.

Procedure

CIRCUIT DESIGN

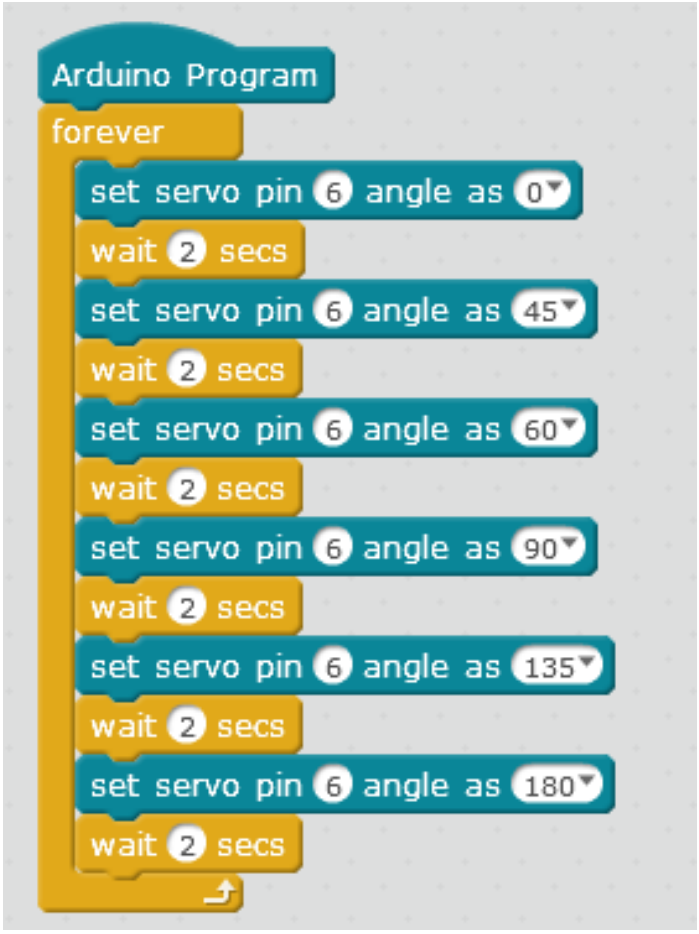
Tutorials



CODE:

This code for move the servo in 0,45, 60, 90, 135, and 180 degrees.

Tutorials


Arduino Program
forever
set servo pin 6 angle as 0
wait 2 secs
set servo pin 6 angle as 45
wait 2 secs
set servo pin 6 angle as 60
wait 2 secs
set servo pin 6 angle as 90
wait 2 secs
set servo pin 6 angle as 135
wait 2 secs
set servo pin 6 angle as 180
wait 2 secs

Tutorials

Tutorial 8# Potentiometer

Experiment Object

In this experiment learn how to program the **potentiometer** by using scratch language.

Theory

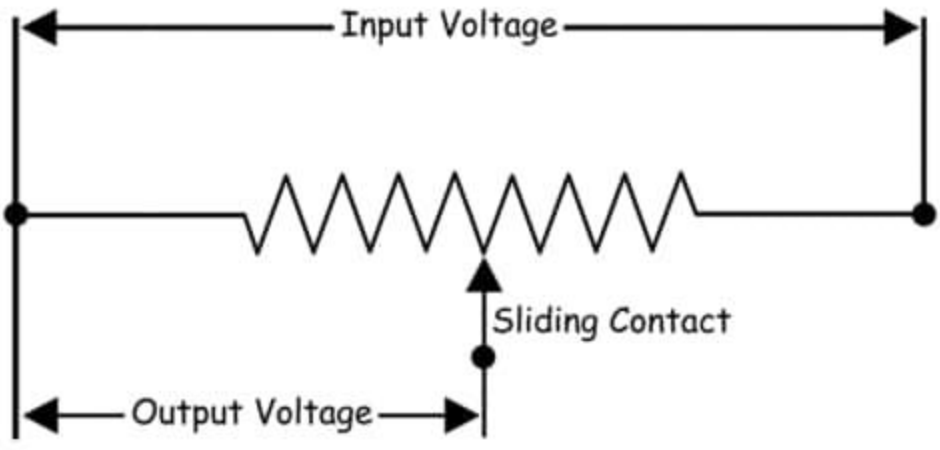
Potentiometer

What is a Potentiometer?

A **potentiometer** (also known as a pot or potmeter) is defined as a 3 terminal variable resistor in which the resistance is manually varied to control the flow of electric current. A potentiometer acts as an adjustable voltage divider.

How Does a Potentiometer Work?

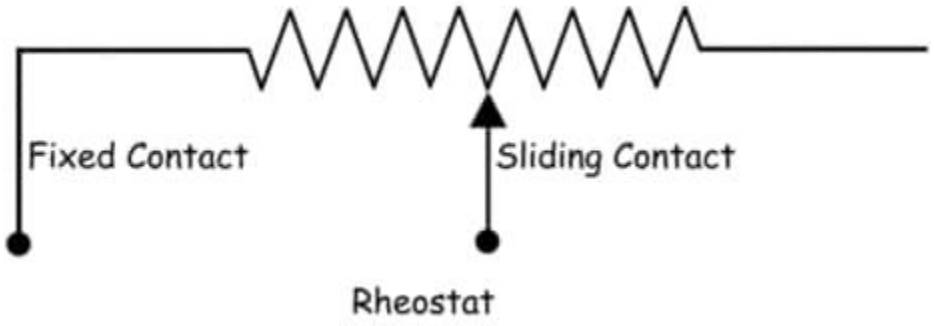
A potentiometer is a passive electronic component. Potentiometers work by varying the position of a sliding contact across a uniform resistance. In a potentiometer, the entire input voltage is applied across the whole length of the resistor, and the output voltage is the voltage drop between the fixed and sliding contact as shown below.



Tutorials

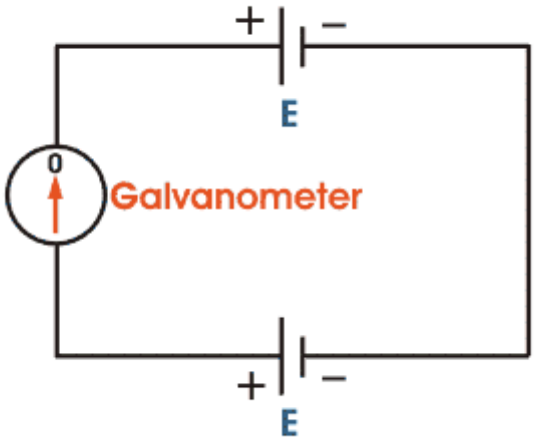
A **potentiometer** has the two terminals of the input source fixed to the end of the resistor. To adjust the output voltage the sliding contact gets moved along the resistor on the output side.

This is different to a rheostat, where here one end is fixed and the sliding terminal is connected to the circuit, as shown below.



Rheostat

This is a very basic instrument used for comparing the emf of two cells and for calibrating ammeter, voltmeter, and watt-meter. The basic working principle of a potentiometer is quite simple. Suppose we have connected two batteries in parallel through a galvanometer. The negative battery terminals are connected together and positive battery terminals are also connected together through a galvanometer as shown in the figure below.



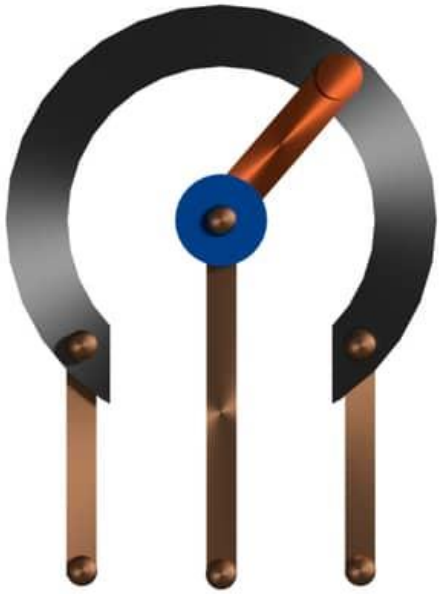
Here, if the electric potential of both battery cells is exactly the same, there is no circulating current in the circuit and hence the galvanometer shows null deflection. The working principle of potentiometer depends upon this phenomenon.

Potentiometer Types

There are two main types of potentiometers:

- ### Rotary potentiometer

  The **rotary** type potentiometers are used mainly for obtaining adjustable supply voltage to a part of electronic circuits and electrical circuits. The volume controller of a radio transistor is a popular example of a rotary potentiometer where the rotary knob of the potentiometer controls the supply to the amplifier.



  This type of potentiometer has two terminal contacts between which a uniform resistance is placed in a semi-circular pattern. The device also has a middle terminal which is connected to the resistance through a sliding contact attached with a rotary knob. By rotating the knob one can move the sliding contact on the semi-circular resistance. The voltage is taken between a resistance end contact and the sliding contact. The potentiometer is also named as the POT in short. POT is also used in substation battery chargers to adjust the charging voltage of a battery. There are

many more uses of rotary type potentiometer where smooth voltage control is required.

- **Linear potentiometer**

  The **linear potentiometer** is basically the same but the only difference is that here instead of rotary movement the sliding contact gets moved on the resistor linearly. Here two ends of a straight resistor are connected across the source voltage. A sliding contact can be slide on the resistor through a track attached along with the resistor. The terminal connected to the sliding is connected to one end of the output circuit and one of the terminals of the resistor is connected to the other end of the output circuit.

This type of potentiometer is mainly used to measure the voltage across a branch of a circuit, for measuring the internal resistance of a battery cell, for comparing a battery cell with a standard cell and in our daily life, it is commonly used in the equalizer of music and sound mixing systems.

Components

1- Arduino UNO R3.

2- Servo Motor SG90.

Tutorials

3- LED.

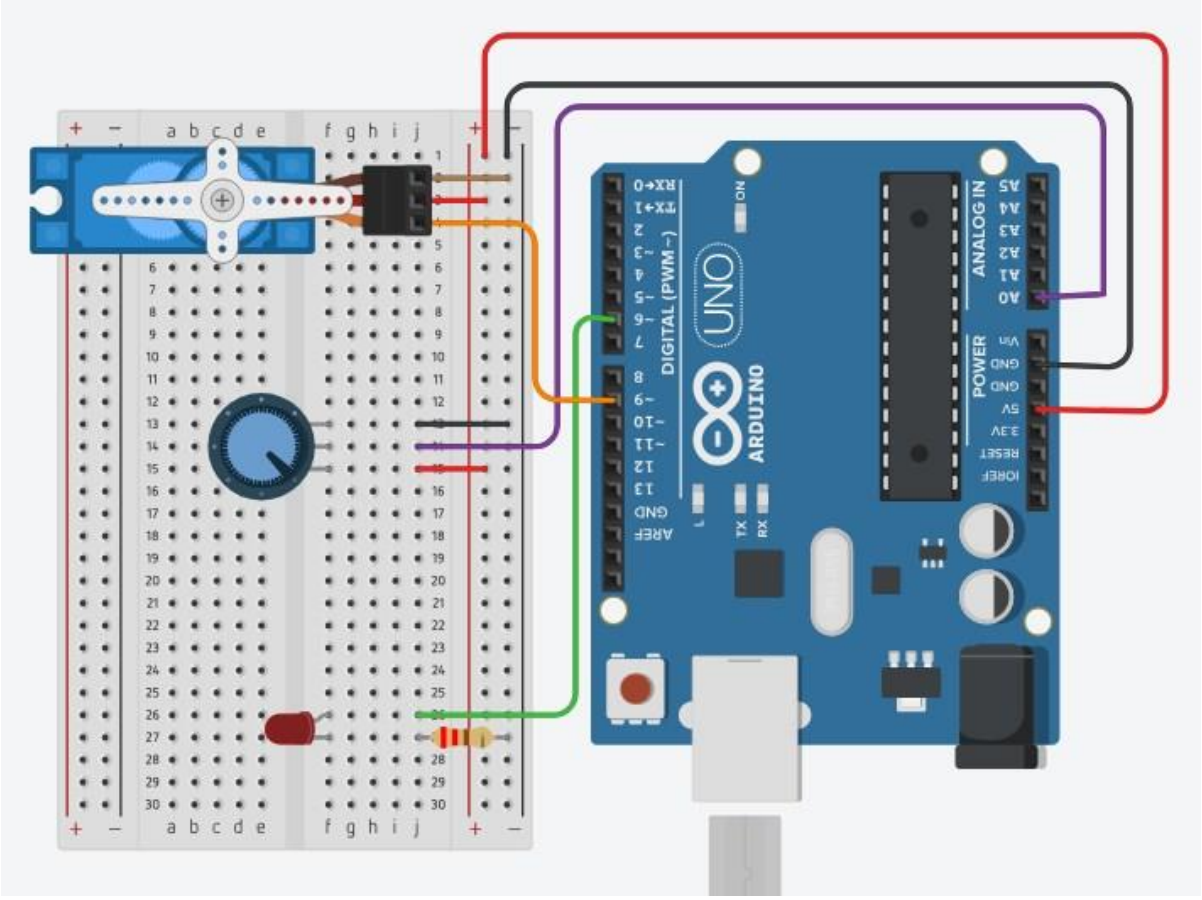
4- Potentiometer 10k.

5- Wires.

6- Breadboard.

Procedure

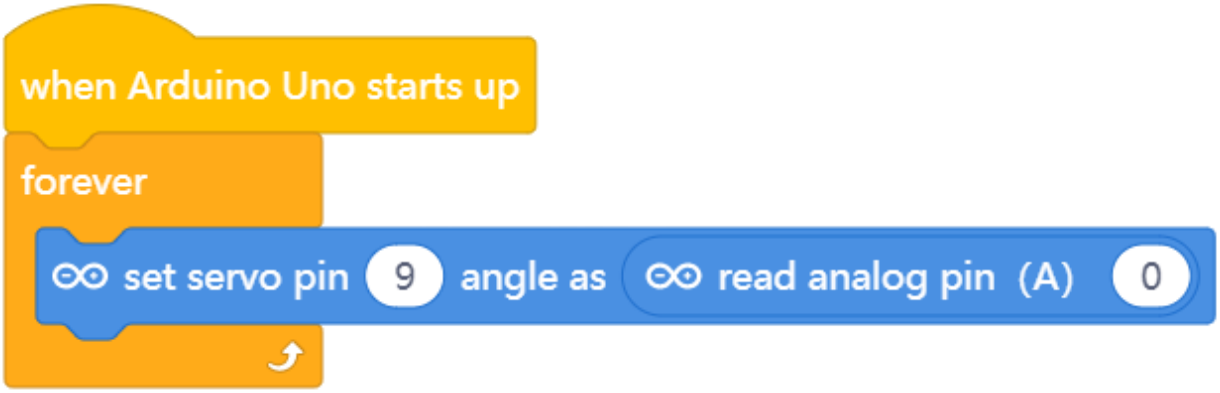
CIRCUIT DESIGN



Tutorials

CODE1:

CODE2:



Tutorials

Tutorial 9# RGB Led

Experiment Object

In this experiment learn how to program the **RGB led** by using scratch language.

Theory

RGB LED

How do RGB LEDs work?

An RGB LED is a combination of 3 LEDs in just one package:

- 1x Red LED
- 1x Green LED
- 1x Blue LED

You can produce almost any color by combining those three colors. An RGB LED is shown in the following figure:

How to create different colors?

With an **RGB LED** you can, of course, produce red, green, and blue light, and by configuring the intensity of each **LED**, you can produce other colors as well.

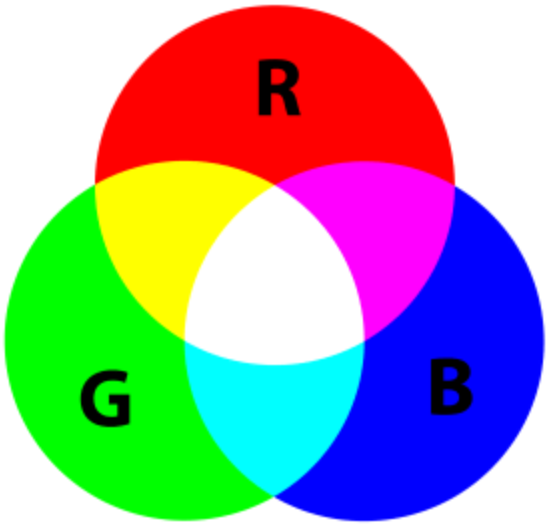
For example, to produce purely blue light, you'd set the **blue LED** to the highest intensity and the green and red LEDs to the lowest intensity. For a white light, you'd set all three LEDs to the highest intensity.

Mixing colors

To produce other colors, you can combine the three colors in different intensities. To adjust the intensity of each LED you can use a **PWM signal**.

Because the LEDs are very close to each other, our eyes see the result of the combination of colors, rather than the three colors individually.

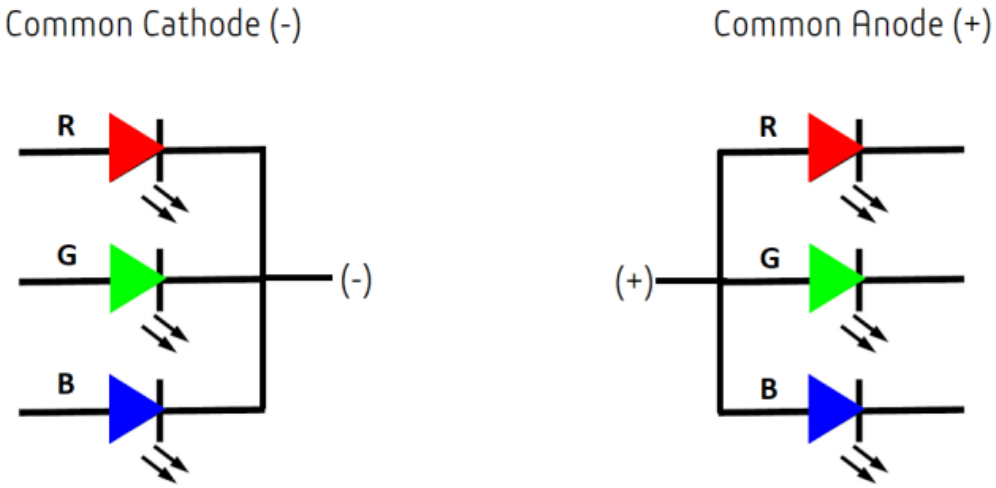
To have an idea on how to combine the colors, take a look at the following chart. This is the simplest color mixing chart, but gives you an idea how it works and how to produce different colors.



Common Anode and Common Cathode RGB LEDs

There are two kinds of RGB LEDs: common anode LED and common cathode LED. The figure below illustrates a common anode and a common cathode LED.

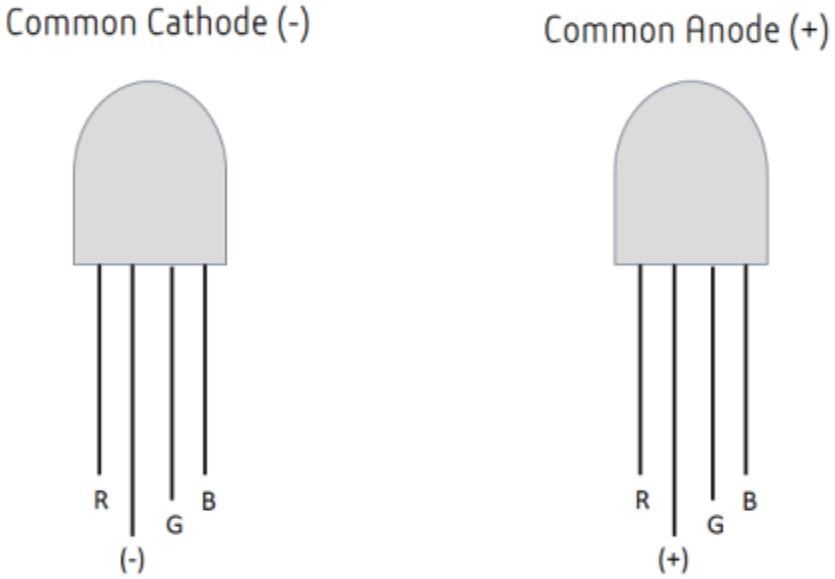
Tutorials



In a common cathode RGB LED, all three LEDs share a negative connection (cathode). In a common anode RGB LED, the three LEDs share a positive connection (anode). This results in an LED that has 4 pins, one for each LED, and one common cathode or one common anode.

RGB LED Pins

RGB LEDs have four leads—one for each LED and another for the common anode or cathode. You can identify each lead by its length, as shown in the following figure.



Tutorials

With the LED facing you so the anode or cathode (the longest lead) is second from the left, the leads should be in the following order: red, anode or cathode, green, and blue.

Components

1- Arduino UNO R3.

2- RGB LED.

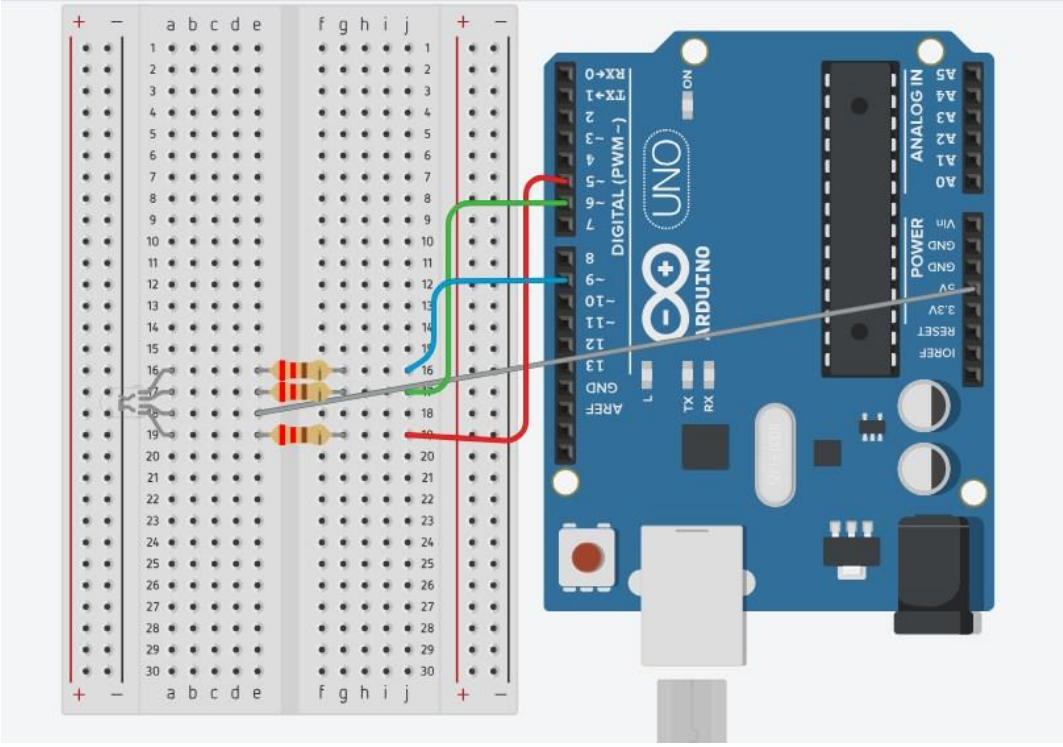
3- Resistor 330ohm x3.

4- Wires.

5- Breadboard.

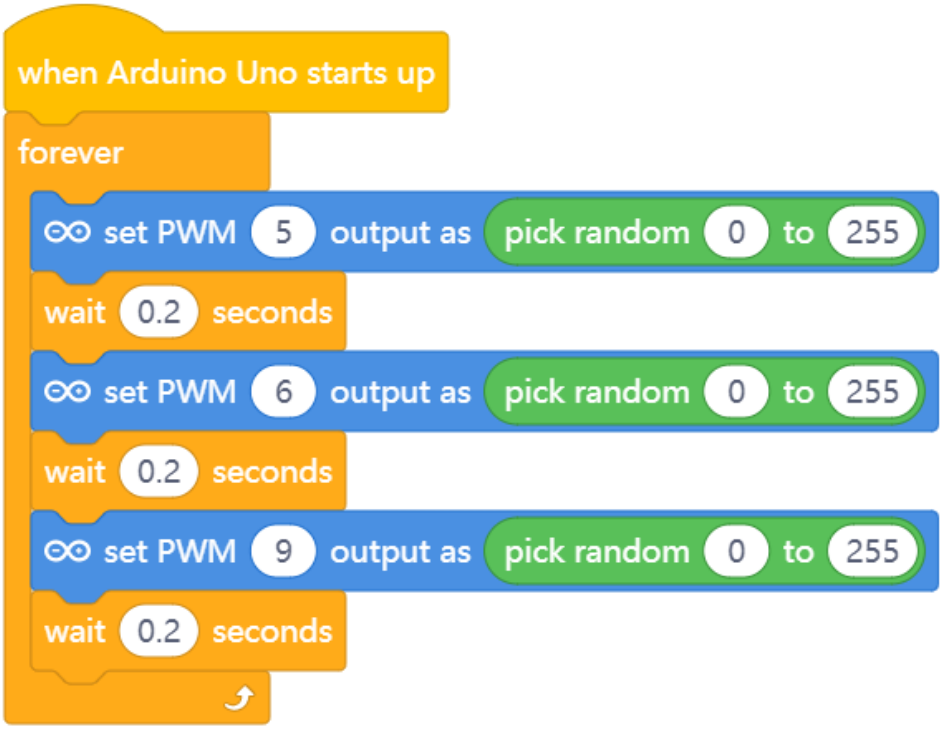
Procedure

CIRCUIT DESIGN



Tutorials

CODE:



Tutorials

Tutorial 10# 7 Segments

Experiment Object

In this experiment learn how to program the **7 segments common anode** by using scratch language.

Theory

7-segment Display

The **7-segment display**, also written as "seven segment display", consists of seven LEDs (hence its name) arranged in a rectangular fashion as shown. Each of the seven LEDs is called a segment because when illuminated the segment forms part of a numerical digit (both Decimal and Hex) to be displayed. An additional 8th LED is sometimes used within the same package thus allowing the indication of a decimal point, (DP) when two or more 7-segment displays are connected together to display numbers greater than ten.

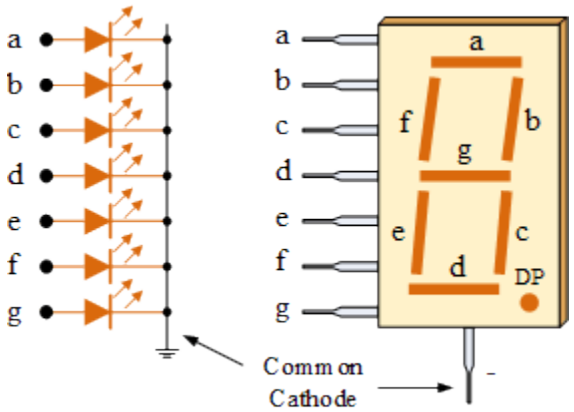
Each one of the seven LEDs in the display is given a positional segment with one of its connection pins being brought straight out of the rectangular plastic package. These individually LED pins are labelled from a through to g representing each individual LED. The other LED pins are connected together and wired to form a common pin.

So, by forward biasing the appropriate pins of the LED segments in a particular order, some segments will be light and others will be dark allowing the desired character pattern of the number to be generated on the display. This then allows us to display each of the ten decimal digits 0 through to 9 on the same 7-segment display.

The displays common pin is generally used to identify which type of 7-segment display it is. As each LED has two connecting pins, one called the "Anode" and the other called the "Cathode", there are therefore two types of LED 7-segment display called: Common Cathode (CC) and Common Anode (CA).

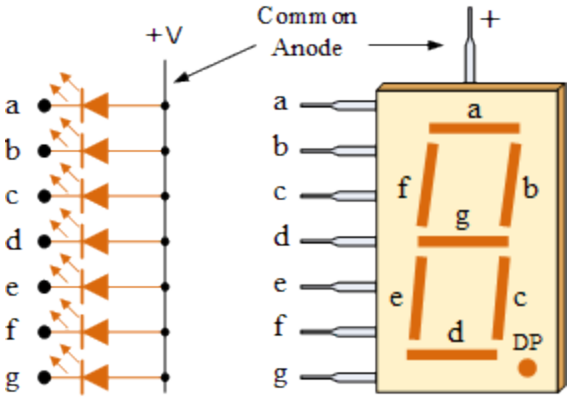
The difference between the two displays, as their name suggests, is that the common cathode has all the cathodes of the 7-segments connected directly together and the common anode has all the anodes of the 7-segments connected together and is illuminated as follows.

1. **The Common Cathode (CC)** – In the common cathode display, all the cathode connections of the LED segments are joined together to logic "0" or ground. The individual segments are illuminated by application of a "HIGH", or logic "1" signal via a current limiting resistor to forward bias the individual Anode terminals (a-g).



2. **The Common Anode (CA)** – In the common anode display, all the anode connections of the LED segments are joined together to logic "1". The individual segments are illuminated by applying a ground, logic "0" or "LOW" signal via a suitable current limiting resistor to the Cathode of the particular segment (a-g).

Tutorials



Components

1- Arduino UNO R3.

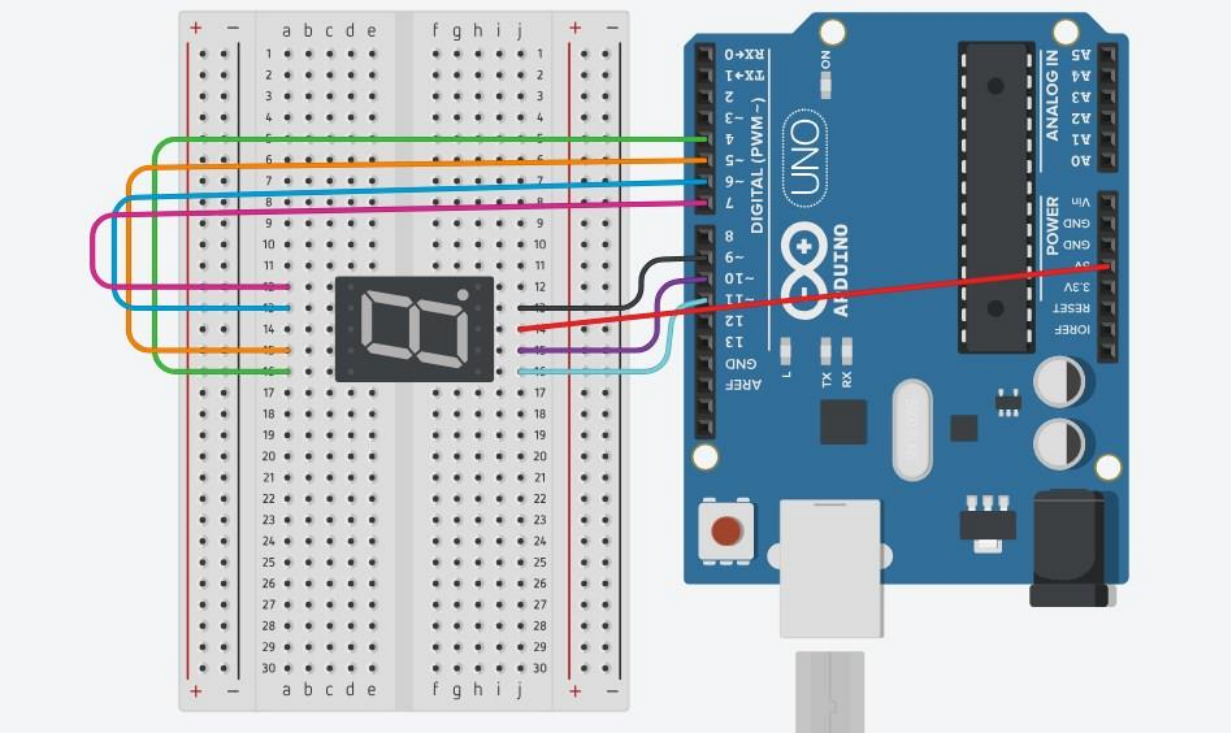
2- 7 Segments (Common Anode).

3- Wires.

4- Breadboard.

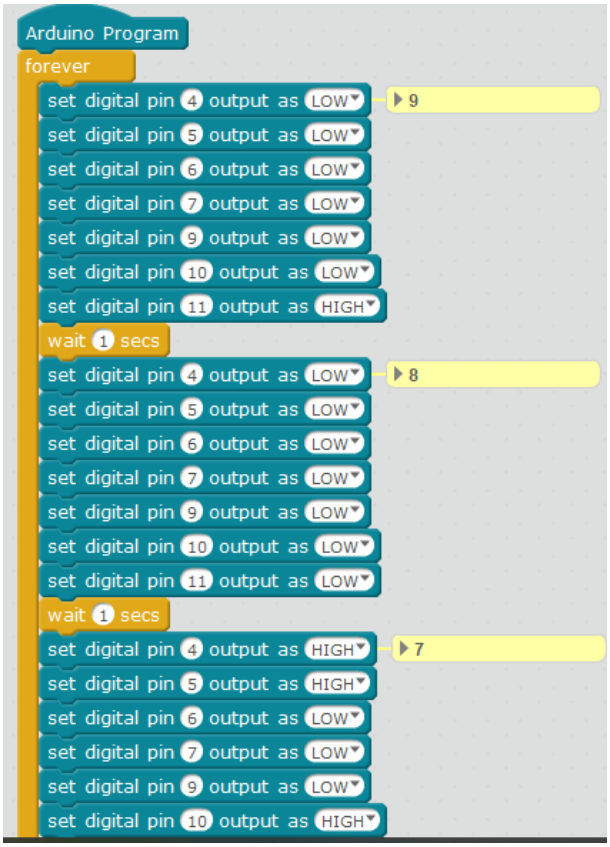
Procedure

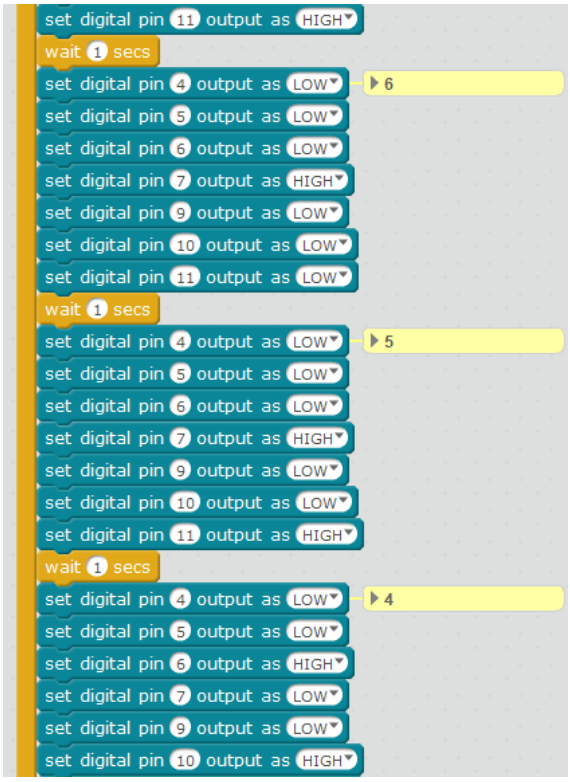
CIRCUIT DESIGN



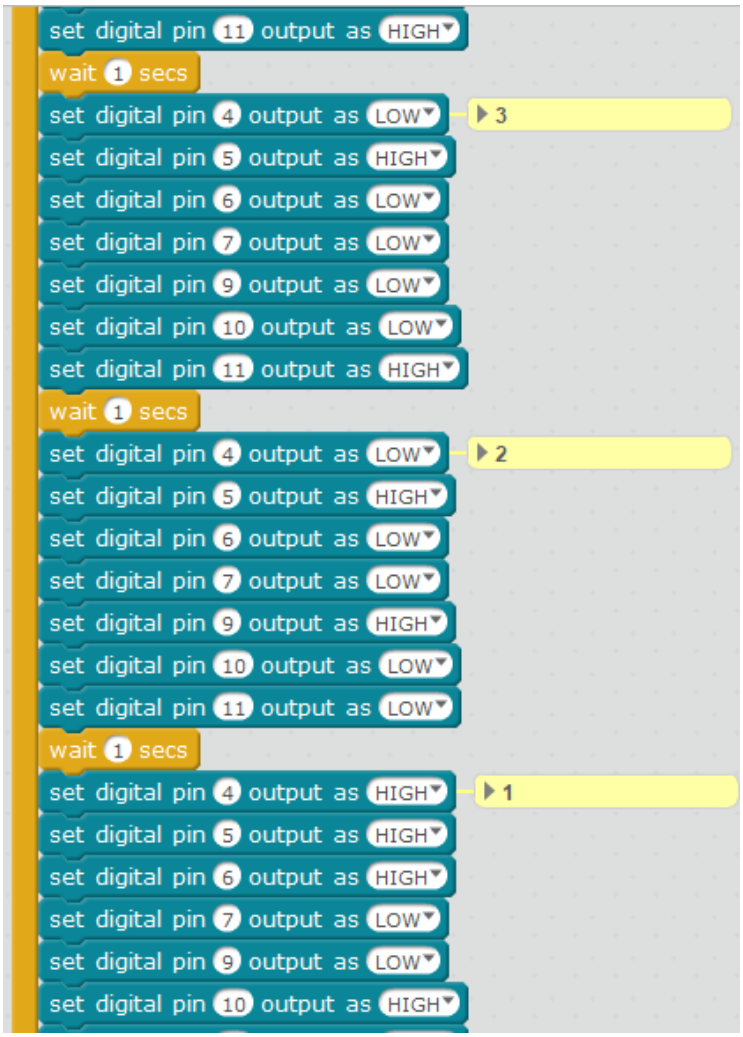
Tutorials

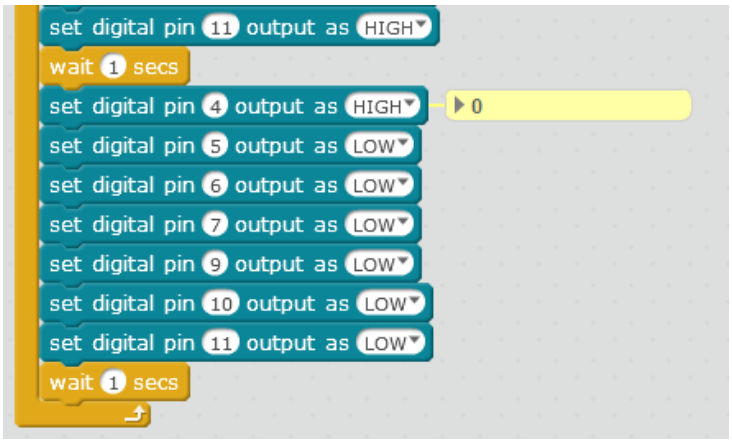
CODE:





Tutorials


set digital pin 11 output as HIGH
wait 1 secs
set digital pin 4 output as LOW
3
set digital pin 5 output as HIGH
set digital pin 6 output as LOW
set digital pin 7 output as LOW
set digital pin 9 output as LOW
set digital pin 10 output as LOW
set digital pin 11 output as HIGH
wait 1 secs
set digital pin 4 output as LOW
2
set digital pin 5 output as HIGH
set digital pin 6 output as LOW
set digital pin 7 output as LOW
set digital pin 9 output as HIGH
set digital pin 10 output as LOW
set digital pin 11 output as LOW
wait 1 secs
set digital pin 4 output as HIGH
1
set digital pin 5 output as HIGH
set digital pin 6 output as HIGH
set digital pin 7 output as LOW
set digital pin 9 output as LOW
set digital pin 10 output as HIGH


set digital pin 11 output as HIGH
wait 1 secs
set digital pin 4 output as HIGH
0
set digital pin 5 output as LOW
set digital pin 6 output as LOW
set digital pin 7 output as LOW
set digital pin 9 output as LOW
set digital pin 10 output as LOW
set digital pin 11 output as LOW
wait 1 secs

Tutorials

Tutorial 11# DC Motor

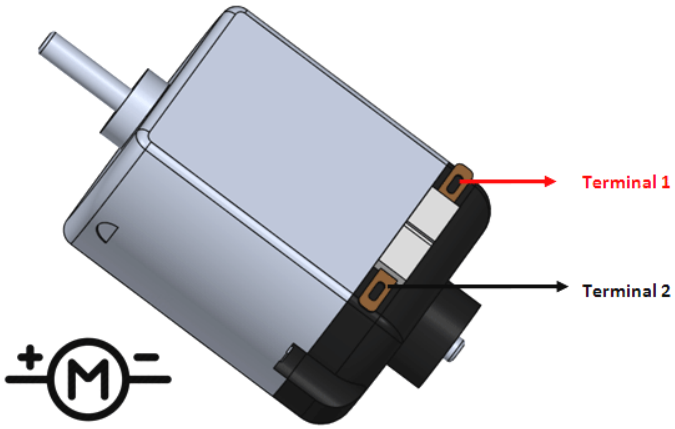
Experiment Object

In this experiment learn how to program the **DC motor** by using scratch language.

Theory

DC Motor

A **direct current** or **DC** motor, converts electrical energy into mechanical energy. It is one of two basic types of motors: the other type is the alternating current or AC motor. Among DC motors, there are shunt-wound, series-wound, compound-wound and permanent magnet motors.



Tutorials

Function

A DC motor consists of a stator, an armature, a rotor and a commutator with brushes. Opposite polarity between the two magnetic fields inside the motor cause it to turn. DC motors are the simplest type of motor and are used in household appliances, such as electric razors, and in electric windows in cars.

Basic DC Motor Operation

A **DC motor** is equipped with magnets, either permanent magnets or electromagnetic windings, that produce a magnetic field. When current passes through the armature, also known as the coil or wire, placed between the north and south poles of the magnet, the field generated by the armature interacts with the field from the magnet and applies torque. In a DC motor, the magnet forms the stator, the armature is placed on the rotor and a commutator switches the current flow from one coil to the other. The commutator connects the stationary power source to the armature through the use of brushes or conductive rods. Furthermore, DC motors operate at a fixed speed for a fixed voltage and there is no slip.

DC Motor Types

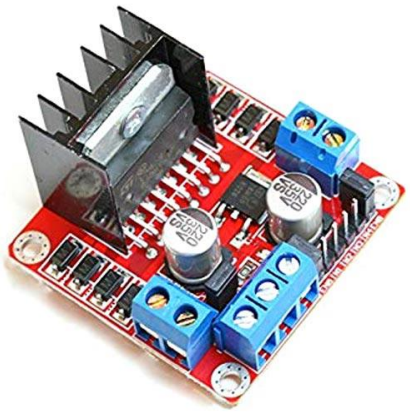
The field coils and the armature in a shunt-wound motor are connected in parallel, also known as shunt, formation, causing the field current to be proportional to the load on the motor.

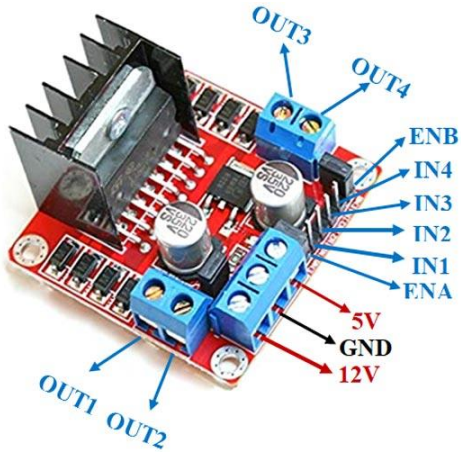
In series-wound motors, the field coils and armature are connected in a series and the current flows through the field coils only.

A compound-wound motor is hybrid of both the shunt-wound and series-wound types and features both configurations. The field in a permanent magnet motor is created by permanent magnets as the name allows.

Tutorials

L298N Driver





This L298N Motor Driver Module is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors, or 2 DC motors with directional and speed control.

L298N Module Pin Configuration:

Pin Name	Description
IN1 & IN2	Motor A input pins. Used to control the spinning direction of Motor A
IN3 & IN4	Motor B input pins. Used to control the spinning direction of Motor B
ENA	Enables PWM signal for Motor A
ENB	Enables PWM signal for Motor B
OUT1 & OUT2	Output pins of Motor A
OUT3 & OUT4	Output pins of Motor B

12V	12V input from DC power Source
5V	Supplies power for the switching logic circuitry inside L298N IC
GND	Ground pin

L298 Module Features & Specifications:

- Driver Model: L298N 2A
- Driver Chip: Double H Bridge L298N
- Motor Supply Voltage (Maximum): 46V
- Motor Supply Current (Maximum): 2A
- Logic Voltage: 5V
- Driver Voltage: 5-35V
- Driver Current:2A
- Logical Current:0-36mA
- Maximum Power (W): 25W
- Current Sense for each motor
- Heatsink for better performance
- Power-On LED indicator

Components

1- Arduino UNO R3.

2- DC Motor.

3- L298N Driver.

4- Battery.

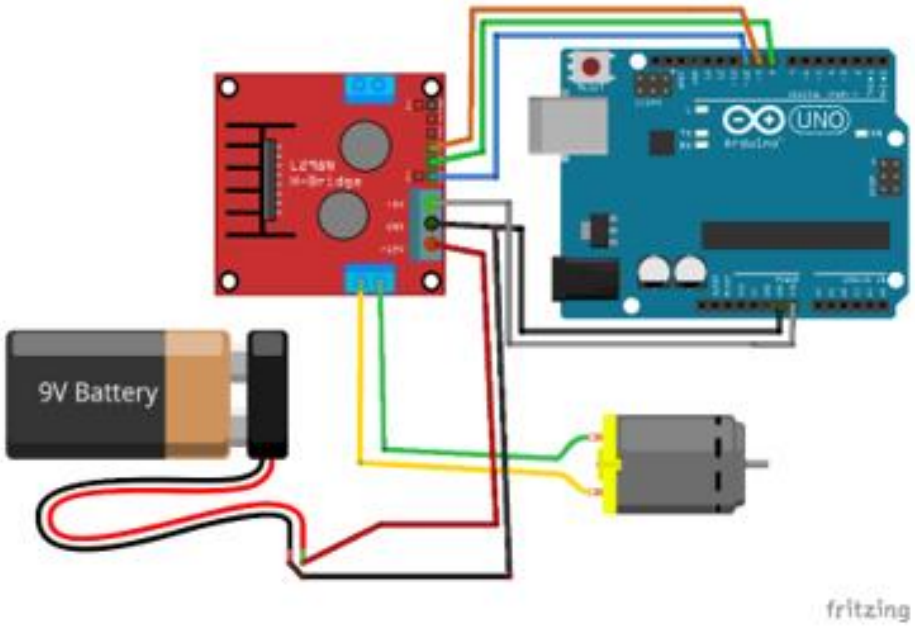
5- Wires.

6- Breadboard.

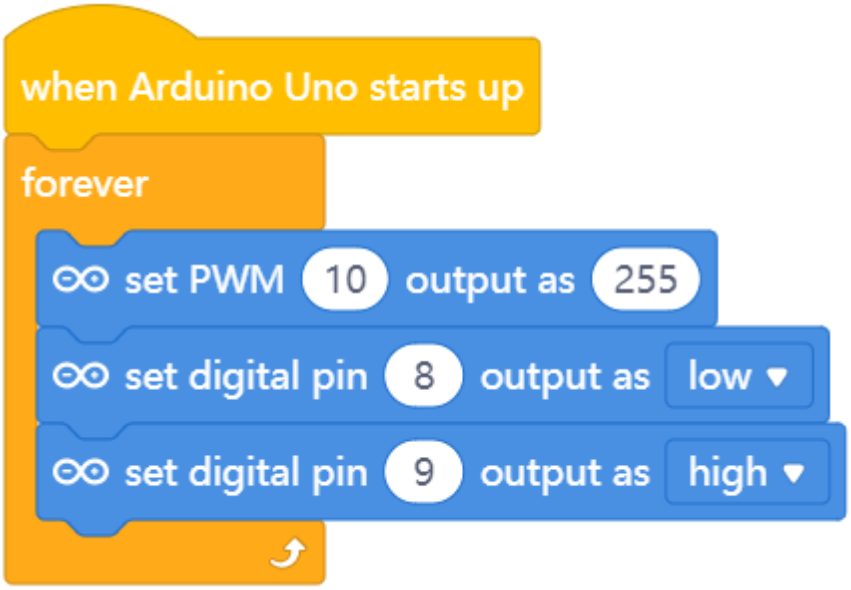
Procedure

CIRCUIT DESIGN

Tutorials



CODE:



Tutorials

Tutorial 12# PIR Sensor

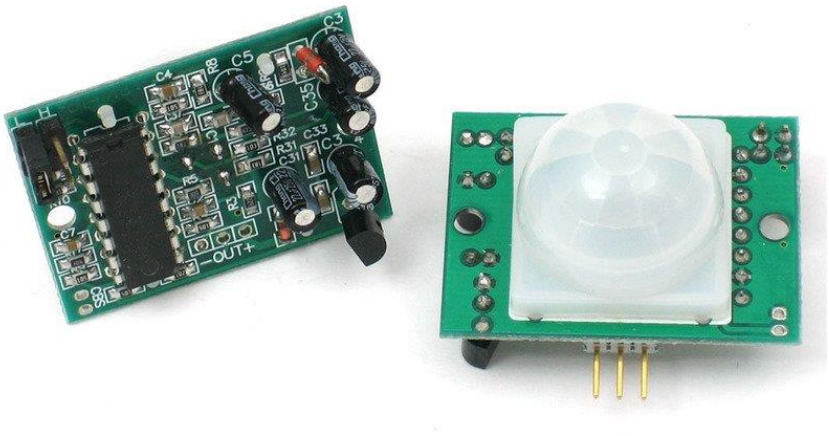
Experiment Object

In this experiment learn how to program the **PIR sensor** by using scratch language.

Theory

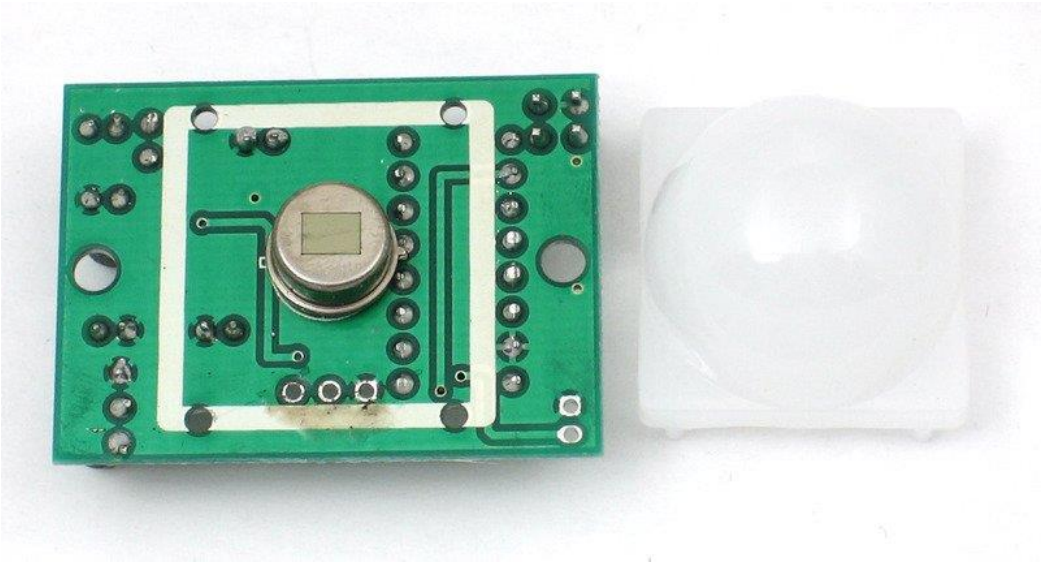
PIR Sensor

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason, they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "**Passive Infrared**", "**Pyroelectric**", or "**IR motion**" sensors.



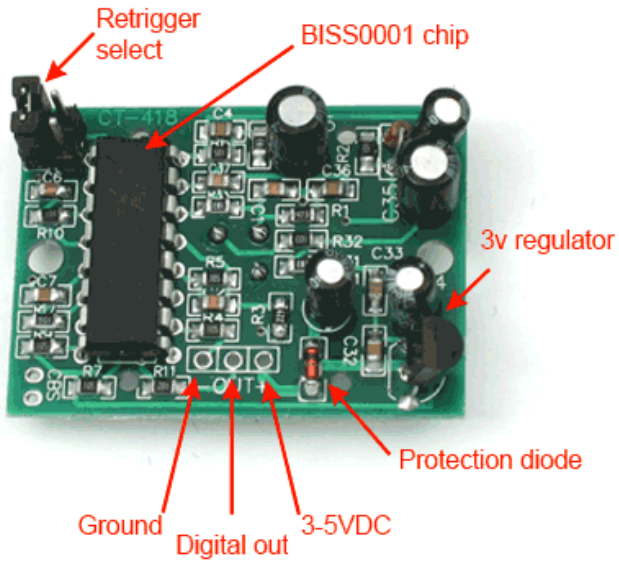
PIRs are basically made of a pyroelectric sensor (which you can see below as the round metal can with a rectangular crystal in the center), which can detect levels of infrared radiation. Everything emits some low-level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

Tutorials



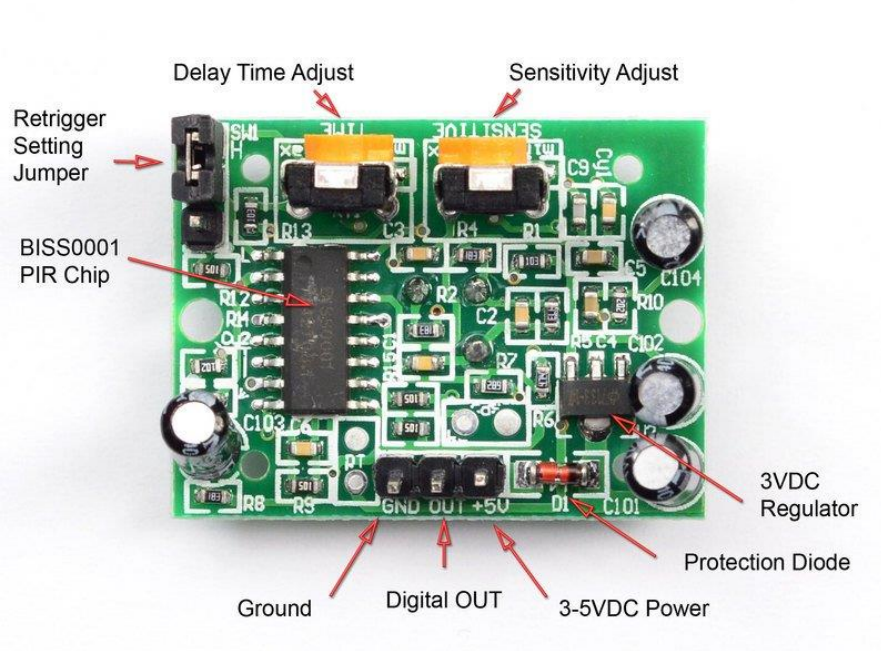
Along with the **pyroelectric sensor** is a bunch of supporting circuitry, resistors and capacitors. It seems that most small hobbyist sensors use the BISS0001 ("**Micro Power PIR Motion Detector IC**"), undoubtedly a very inexpensive chip. This chip takes the output of the sensor and does some minor processing on it to emit a digital output pulse from the analog sensor.

Our older PIRs looked like this:



Our new PIRs have more adjustable settings and have a header installed in the 3-pin ground/out/power pads.

Tutorials



Components

1- Arduino UNO R3.

2- Buzzer.

3- PIR Motion Sensor.

4- LED.

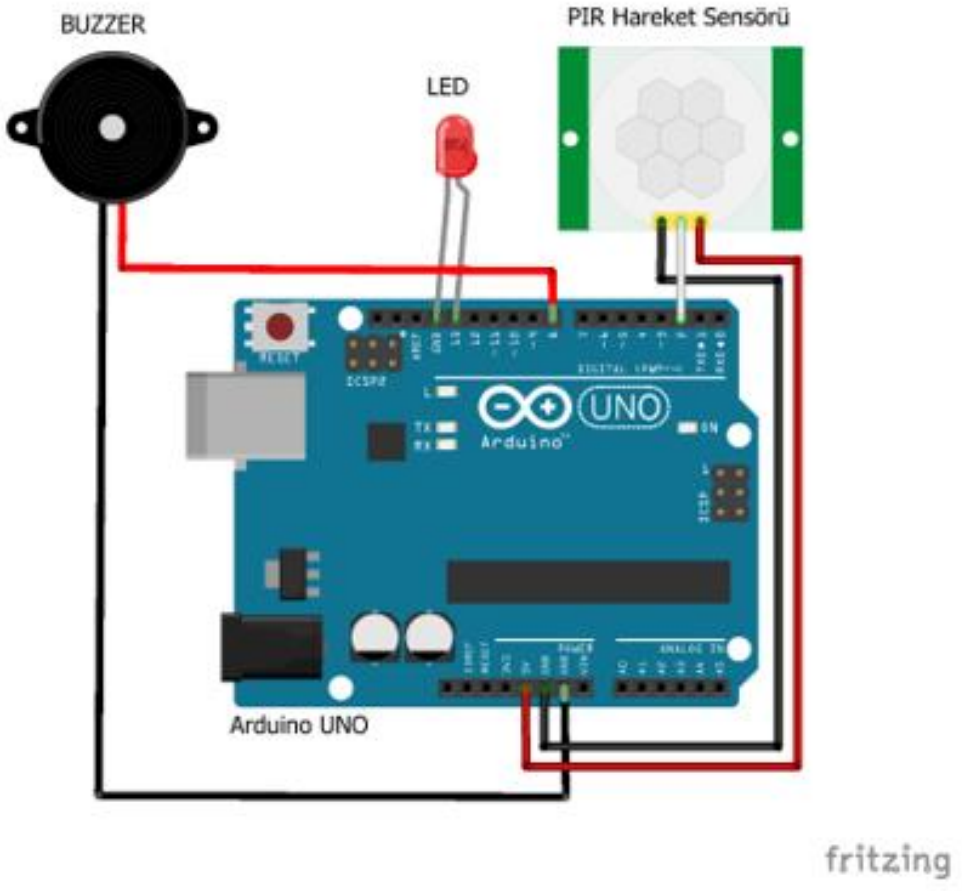
5- Wires.

6- Breadboard.

Procedure

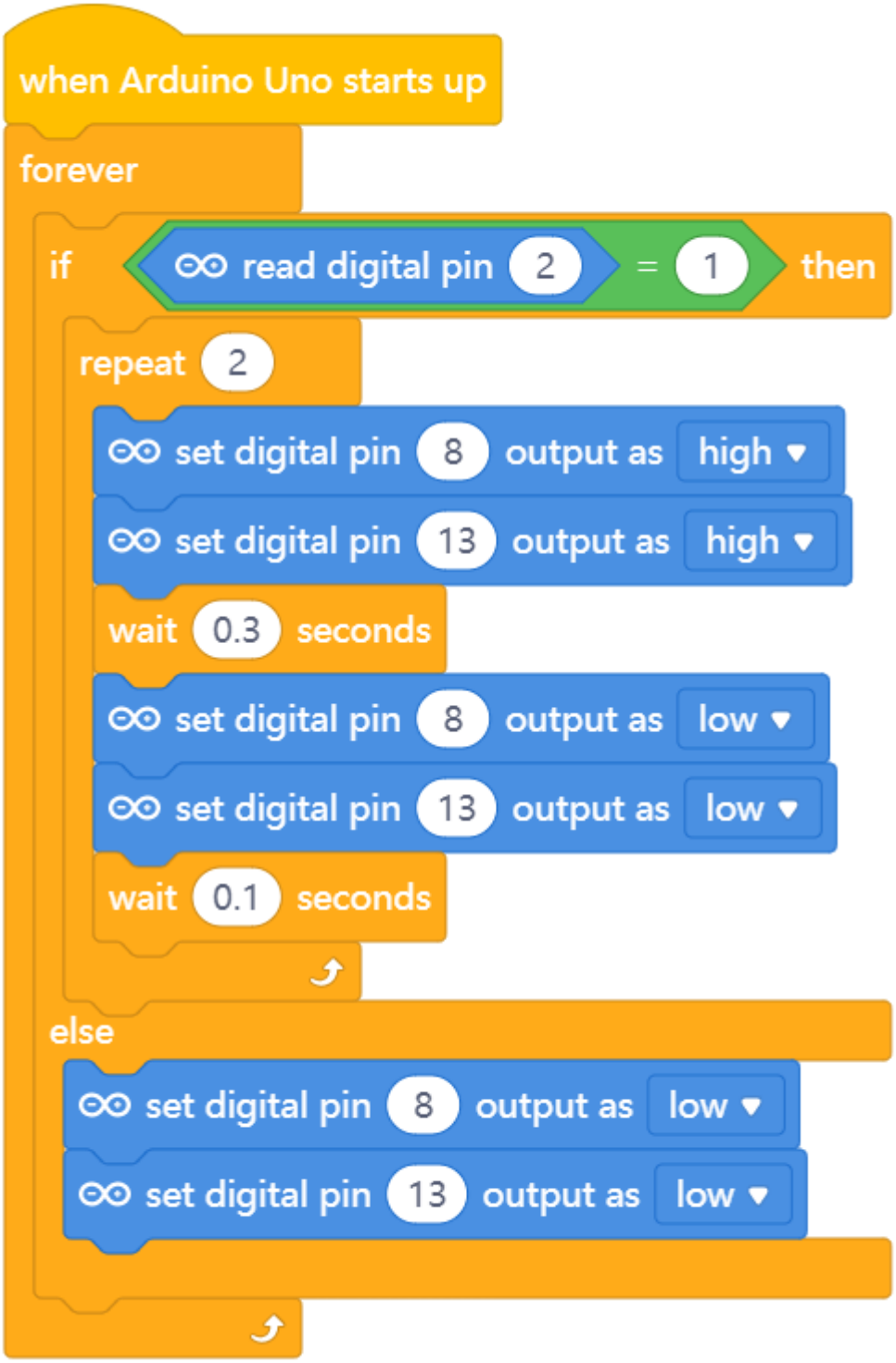
CIRCUIT DESIGN

Tutorials



CODE:

Tutorials


when Arduino Uno starts up
forever
if
read digital pin 2
=
1
then
repeat
2
set digital pin 8 output as high
set digital pin 13 output as high
wait 0.3 seconds
set digital pin 8 output as low
set digital pin 13 output as low
wait 0.1 seconds
else
set digital pin 8 output as low
set digital pin 13 output as low

Tutorials

Tutorial 13# Bluetooth Module

Experiment Object

In this experiment learn how to program the **Bluetooth module** display by using scratch language.

Theory

Bluetooth HC-06

HC-06 is a Bluetooth module designed for establishing short range wireless data communication between two microcontrollers or systems. The module works on Bluetooth 2.0 communication protocol and it can only act as a slave device. This is cheapest method for wireless data transmission and more flexible compared to other methods and it even can transmit files at speed up to 2.1Mb/s.

HC-06 uses frequency hopping spread spectrum technique (FHSS) to avoid interference with other devices and to have full duplex transmission. The device works on the frequency range from 2.402 GHz to 2.480GHz.

Tutorials

Pin configuration

HC-06 module has six pins as shown in the pinout. In them we only need to use four for successfully interfacing the module. Some breakout boards will only leave four output pins only because of this reason.

Pin	Name	Function
1	Key	The pin state determines whether the module works in AT command mode or normal mode [High=AT commands receiving mode(Commands response mode), Low or NC= Bluetooth module normally working]
2	Vcc	+5V Positive supply needs to be given to this pin for powering the module
3	Gnd	Connect to ground
4	TXD	Serial data is transmitted by module through this pin (at 9600bps by default), 3.3V logic
5	RXD	Serial data is received by module through this pin (at 9600bps by default),3.3V logic
6	State	The pin is connected to the LED on the board to represent the state of the module

HC-06 Features and Electrical characteristics

Bluetooth protocol: Bluetooth V2.0 protocol standard

Power Level: Class2(+6dBm)

Band: 2.40GHz—2.48GHz, ISM Band

Receiver sensitivity: -85dBm

USB protocol: USB v1.1/2.0

Modulation mode: Gauss frequency Shift Keying

Safety feature: Authentication and encryption

Operating voltage range:+3.3V to +6V

Operating temperature range: -20ºC to +55ºC

Operating Current: 40mA

Similar Bluetooth Modules

HC-04, HC-02, HC-05, HC-03

HC-06 Bluetooth Module Advantages

HC-06 is best option when short distance wireless communication is needed. The module is used for wireless communications of less than 100 meters.

The module is very easy to interface and to communicate.

The module is one of the cheapest solutions for wireless communication of all types present in the market.

The module consumes very less power to function and can be used on battery operated mobile systems.

The module can be interfaced with almost all controllers or processors as it uses UART interface.

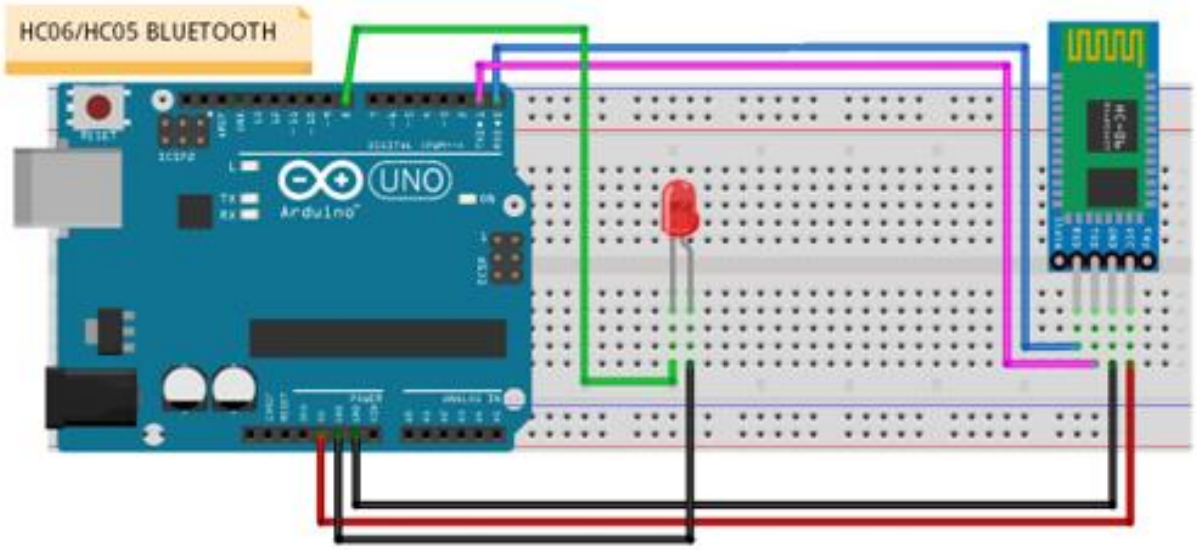
Components

1- Arduino UNO R3.
2- Bluetooth.
3- LED.
4- Wires.
5- Breadboard.

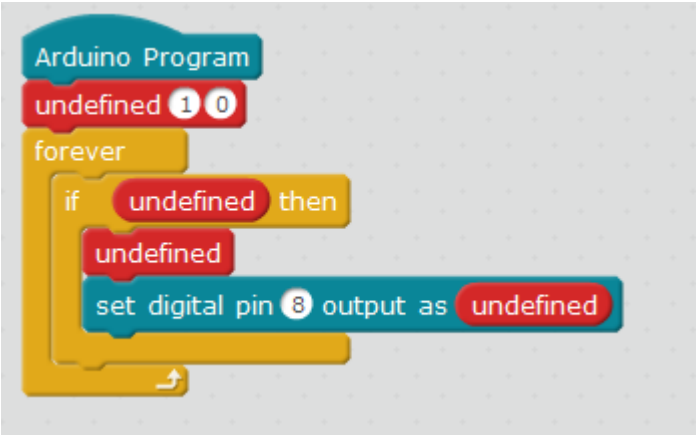
Procedure

CIRCUIT DESIGN

Tutorials



CODE:



Tutorials

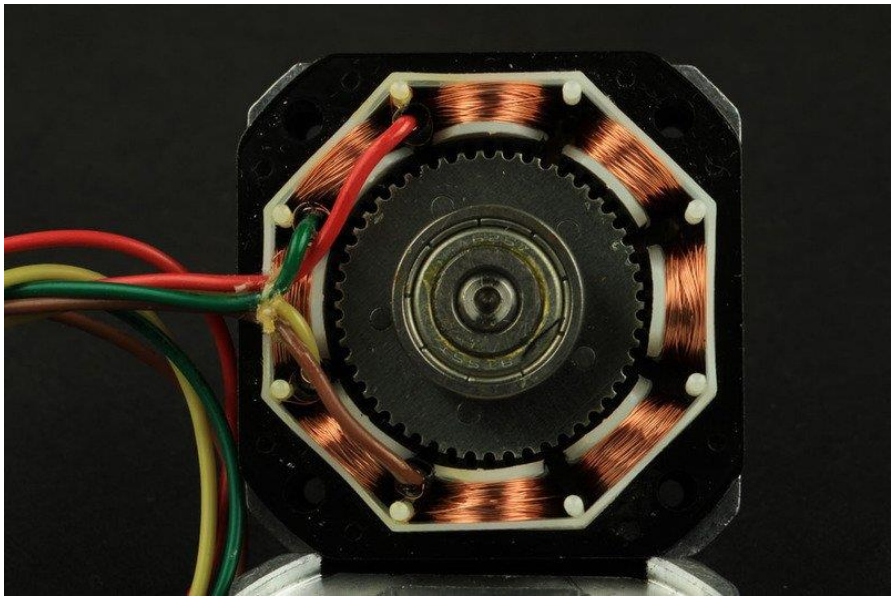
Tutorial 14# Stepper Motor

Experiment Object

In this experiment learn how to program the **stepper motor** by using scratch language.

Theory

Stepper Motor



Stepper motors are DC motors that move in discrete steps. They have multiple coils that are organized in groups called "phases". By energizing each phase in sequence, the motor will rotate, one step at a time.

With a computer controlled stepping you can achieve very precise positioning and/or speed control. For this reason, stepper motors are the motor of choice for many precision motion control applications.

Stepper motors come in many different sizes and styles and electrical characteristics. This guide details what you need to know to pick the right motor for the job.

Types of Stepper Motor:

There are three main types of stepper motors, they are:

- Permanent magnet stepper
- Hybrid synchronous stepper
- Variable reluctance stepper

Permanent Magnet Stepper Motor: Permanent magnet motors use a permanent magnet (PM) in the rotor and operate on the attraction or repulsion between the rotor PM and the stator electromagnets.

Variable Reluctance Stepper Motor: Variable reluctance (VR) motors have a plain iron rotor and operate based on the principle that minimum reluctance occurs with minimum gap, hence the rotor points are attracted toward the stator magnet poles.

Hybrid Synchronous Stepper Motor: Hybrid stepper motors are named because they use a combination of permanent magnet (PM) and variable reluctance (VR) techniques to achieve maximum power in small package size.

Advantages of Stepper Motor:

The rotation angle of the motor is proportional to the input pulse.

The motor has full torque at standstill.

Precise positioning and repeatability of movement since good stepper motors have an accuracy of **3 – 5%** of a step and this error is noncumulative from one step to the next.

Excellent response to starting, stopping, and reversing.

Very reliable since there are no contact brushes in the motor. Therefore, the life of the motor is simply dependent on the life of the bearing.

The motor's response to digital input pulses provides open-loop control, making the motor simpler and less costly to control.

It is possible to achieve very low-speed synchronous rotation with a load that is directly coupled to the shaft.

A wide range of rotational speeds can be realized as the speed is proportional to the frequency of the input pulses.

Applications:

Industrial Machines – Stepper motors are used in automotive gauges and machine tooling automated production equipment.

Security – new surveillance products for the security industry.

Medical – Stepper motors are used inside medical scanners, samplers, and also found inside digital dental photography, fluid pumps, respirators, and blood analysis machinery.

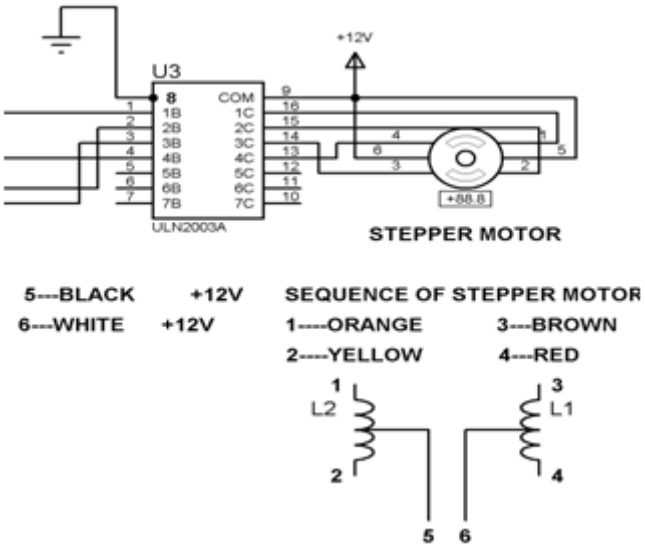
Consumer Electronics – Stepper motors in cameras for automatic digital camera focus and zoom functions.

And also have business machines applications, computer peripherals applications.

Operation of Stepper Motor:

Stepper motors operate differently from DC brush motors, which rotate when voltage is applied to their terminals. Stepper motors, on the other hand, effectively have multiple toothed electromagnets arranged around a central gear-shaped piece of iron. The electromagnets are energized by an external control circuit, for example, a microcontroller.

Tutorials



To make the motor shaft turn, first one electromagnet is given power, which makes the gear's teeth magnetically attracted to the electromagnet's teeth. The point when the gear's teeth are thus aligned to the first electromagnet, they are slightly offset from the next electromagnet. So when the next electromagnet is turned ON and the first is turned OFF, the gear rotates slightly to align with the next one and from there the process is repeated. Each of those slight rotations is called a step, with an integer number of steps making a full rotation. In that way, the motor can be turned by a precise. Stepper motor doesn't rotate continuously, they rotate in steps. There are 4 coils with a 90o angle between each other fixed on the stator. The stepper motor connections are determined by the way the coils are interconnected. In a stepper motor, the coils are not connected. The motor has a 90o rotation step with the coils being energized in a cyclic order, determining the shaft rotation direction. The working of this motor is shown by operating the switch. The coils are activated in series in 1-sec intervals. The shaft rotates 90o each time the next coil is activated. Its low-speed torque will vary directly with current.

Components

1- Arduino UNO R3.

2- Bluetooth.

Tutorials

3- Stepper Motor.

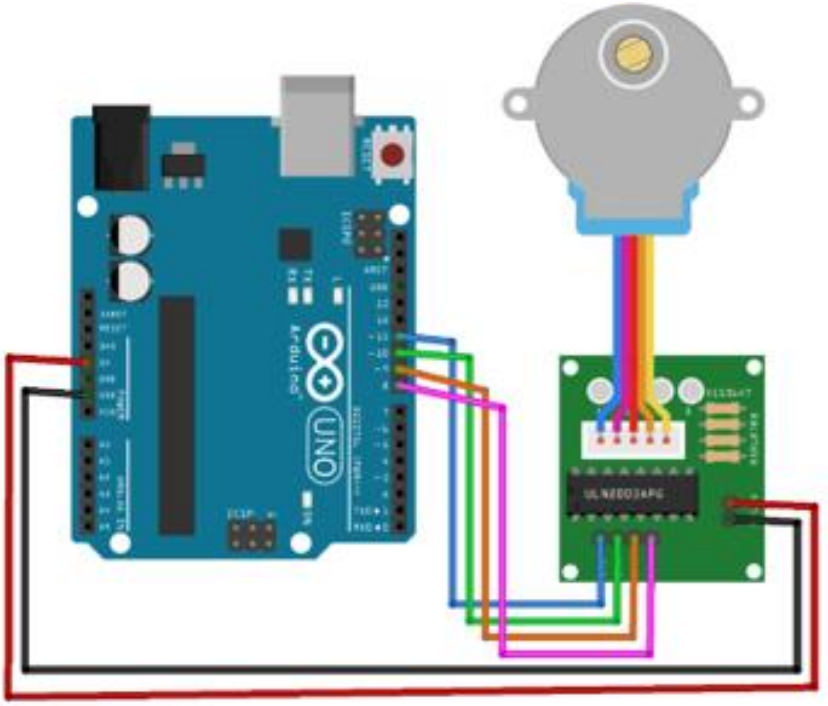
4- ULN2003 Driver.

5- Wires.

6- Breadboard.

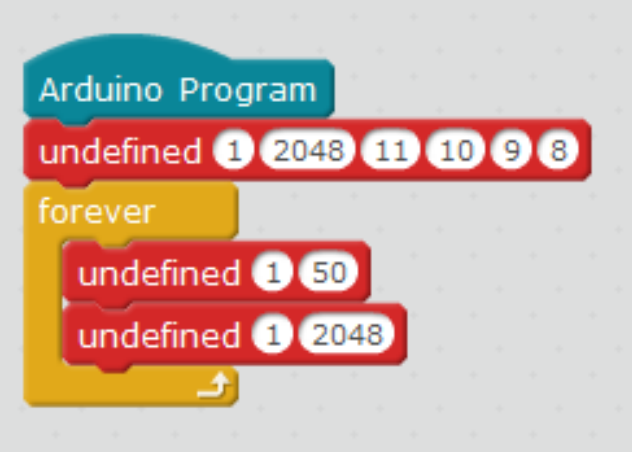
Procedure

CIRCUIT DESIGN



Tutorials

CODE1:



CODE2:

Reference

1- Resnick, Mitchel, et al. "Scratch: programming for all." Communications of the ACM 52.11 (2009): 60-67.

2- Maloney, John, et al. "The scratch programming language and environment." ACM Transactions on Computing Education (TOCE) 10.4 (2010): 1-15.

3- Gunbatar, Mustafa Serkan, and Halit Karalar. "Gender Differences in Middle School Students' Attitudes and Self-Efficacy Perceptions towards mBlock Programming." European Journal of Educational Research 7.4 (2018): 925-933.

4- Arduino, Store Arduino. "Arduino." Arduino LLC (2015).

5- Badamasi, Yusuf Abdullahi. "The working principle of an Arduino." 2014 11th international conference on electronics, computer and computation (ICECCO). IEEE, 2014.