ELECTRONICS HUB

PROJECTS | TUTORIALS | REVIEWS | KITS

HOME	PROJECTS	MINI PROJECTS	ARDUINO	FREE CIRCUITS	TUTORIALS
SYMBOLS	DIY	REVIEWS	CONTACT US		

YOU ARE HERE: HOME / RASPBERRY PI / RASPBERRY PI SERVO MOTOR INTERFACE | HOW TO CONTROL A SERVO MOTOR USING RASPBERRY PI?

Raspberry Pi Servo Motor Interface | How to Control a Servo Motor using Raspberry Pi?

FEBRUARY 14, 2018 BY RAVI — 1 COMMENT

In this project, we will learn about Servo Motors, Raspberry Pi Servo Motor Interface and How to Control a Servo Motor using Raspberry Pi and Python. For this project, we will be using the Raspberry Pi 3 and Tower Pro SG90 Servo Motor.

Table of Contents

Overview

Interfacing a Servo Motor with Raspberry Pi is an interesting topic as Servo Motors are the main components of a Robot and with the help of Raspberry Pi, you can have endless opportunities in Robotics.

A Servo Motor is a simple device that consists of a DC Motor, Gears and a Feed Back based Position Control System. The main advantage of a Servo Motor is its ability to hold the angular position of its shaft.

There are several varieties of Servo Motors you can choose from. Two of the most common Servo Motors are Tower Pro SG90 and Tower Pro MG90S.

SG90 is a plastic gear motor whereas MG90S is a metal gear motor. In this project, I'll be using the SG90 Servo Motor.
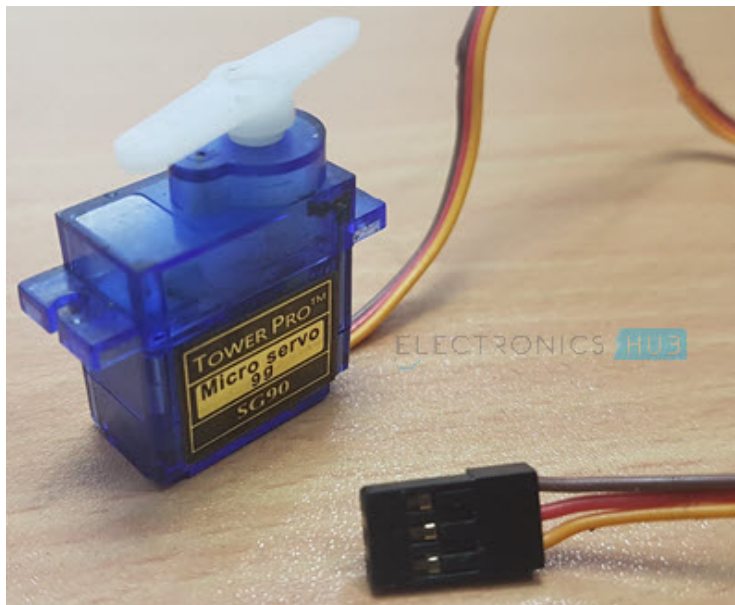
The Tower Pro SG90 Servo Motor Consists of three Pins: PWM (Orange or Yellow), V_{CC} (Red) and GND (Brown). The V_{CC} and GND pins must be connected to +5V and GND of the power supply.

PWM or Signal Pin of the Servo Motor must be connected to the PWM Output of the Controller (Raspberry Pi). Based on the width of the Pulses from the PWM signal, the angle of the Servo Motor's shaft will vary.

How to Control a Servo Motor?

In order to control a Servo Motor, you need to use a technique called Pulse Width Modulation or PWM. In PWM technique, you will be sending a pulse of variable width and the position of the Servo Motor's shaft will be set by the width or length of the Pulse.

The frequency of the PWM Signal is a fixed value and is dependent on the type of the Servo Motor. In our case, both SG90 and MG90S Servo Motors have a PWM Frequency of 50Hz.

At 50Hz i.e. a period of 20ms, the minimum pulse width is 1ms and the maximum pulse width is 2ms. Most servo motors can have a sweep area of 180 degrees i.e. 90 degrees on either side of the neutral position.
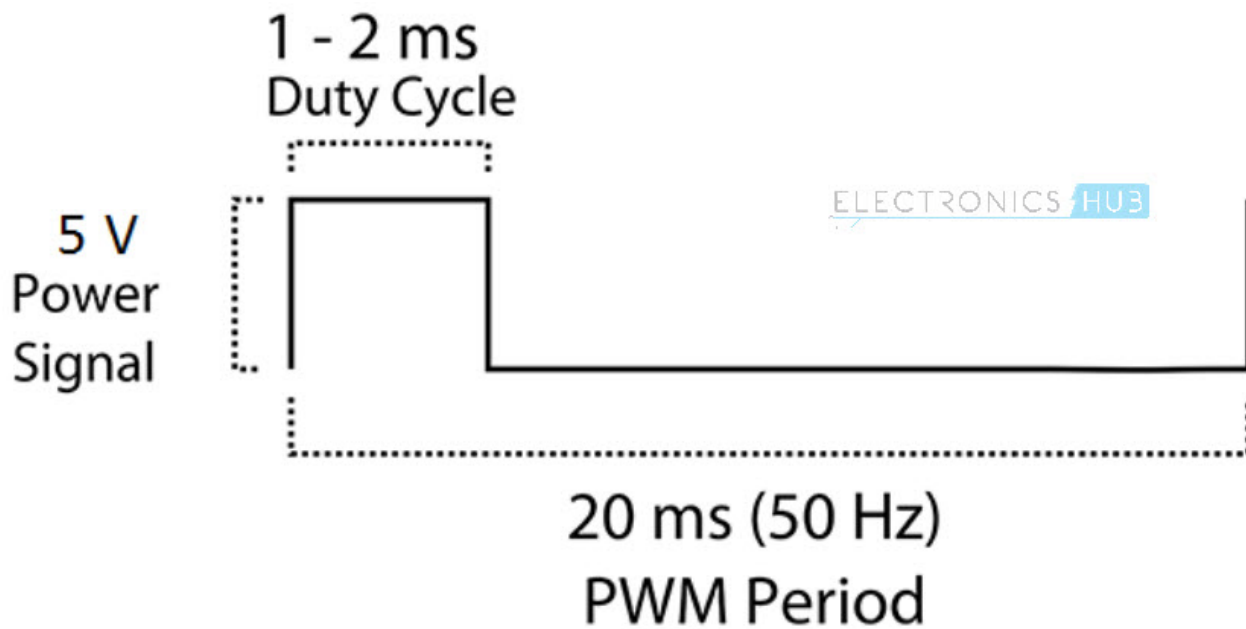
When the pulse width of the PWM Signal is 1ms, the position of the servo is all the way to the LEFT. The Duty Cycle of this position is (1ms/20ms) x 100 = 5%.

Similarly, for pulse widths of 1.5ms and 2ms, the position of the servo is MIDDLE (with duty cycle of 7.5%) and far RIGHT (with duty cycle of 10%).
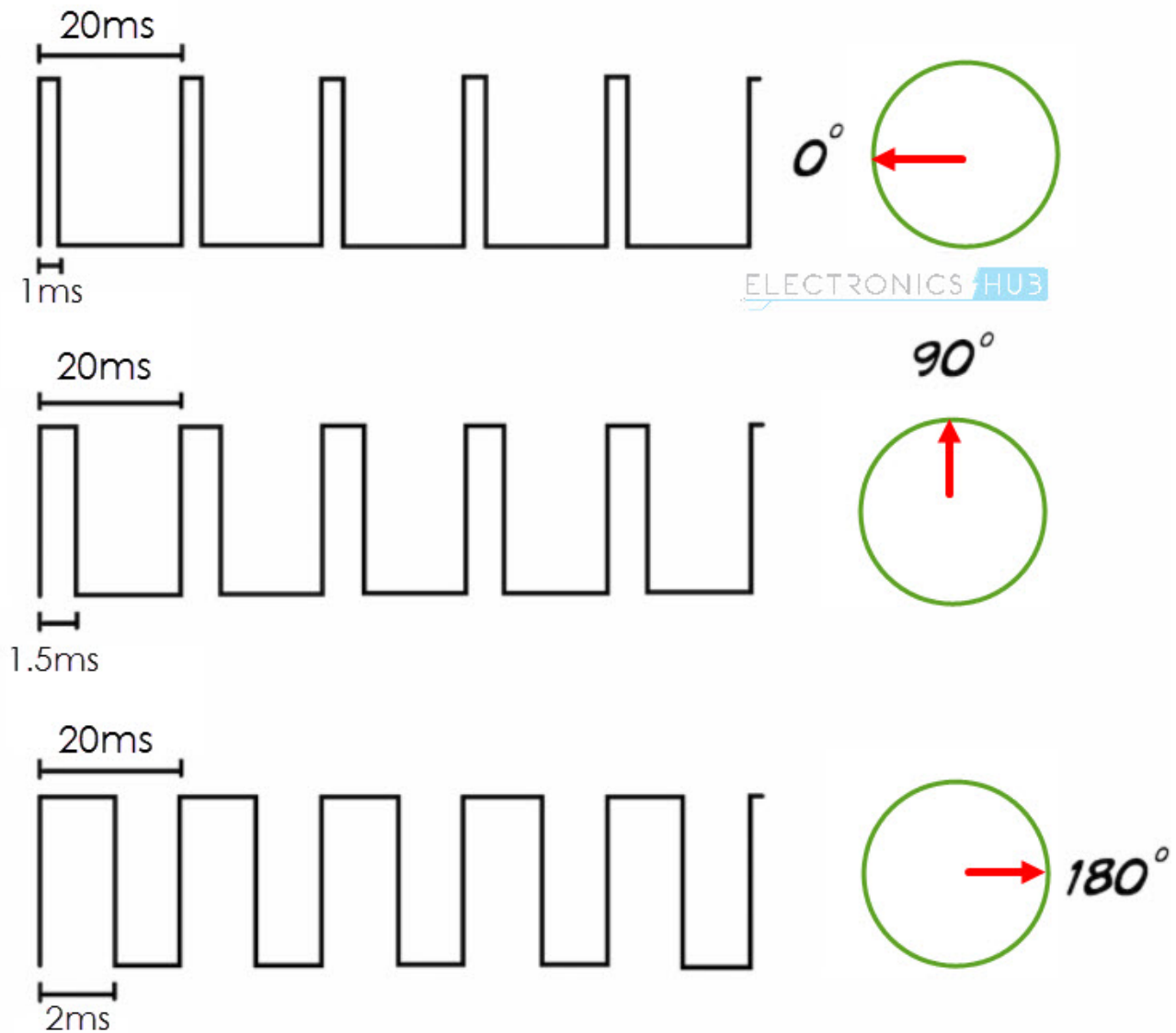
Raspberry Pi Servo Motor (SG90) Interface

Interfacing a Servo Motor with Raspberry Pi will help you in implementing complex projects like Web Controlled Servo, RC Robot, IoT Servo Control etc.

Also, the first thing you need to learn while working on Robotics is how to control a Servo Motor.

So, in this project, I've implemented a simple control of Servo Motor with the help of Raspberry Pi. Now, Raspberry Pi has only one pin that is capable of producing PWM Pulse. It is Physical Pin 12 i.e. GPIO18.

But I have connected the PWM pin of Servo to GPIO25 (Physical Pin 22) of Raspberry Pi, in order to check whether the project will work or not. It worked without any hassle.

Circuit Diagram of Raspberry Pi Servo Motor Interface

The following Fritzing image shows the circuit diagram of the Raspberry Pi Servo Motor Interface.
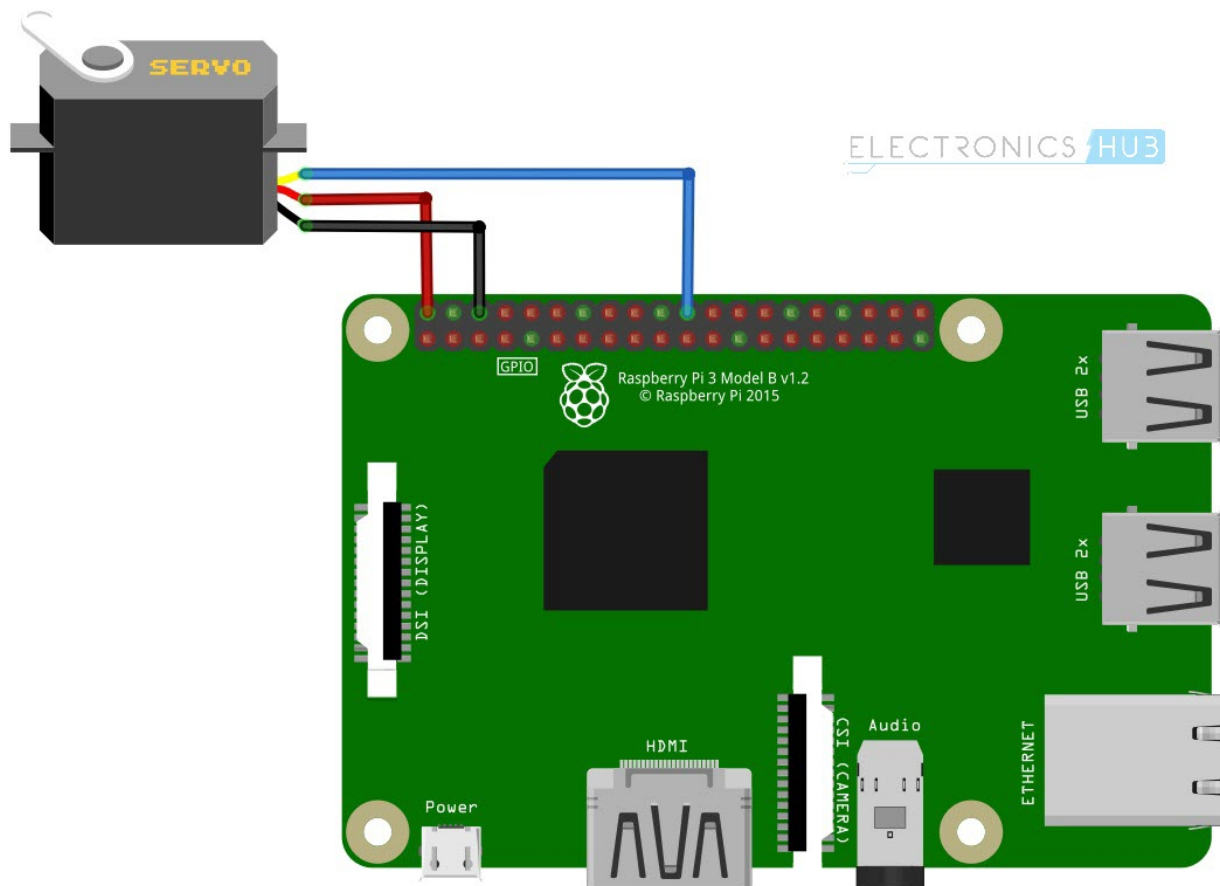
To understand more clearly about the connections, the following wiring diagram might be helpful.
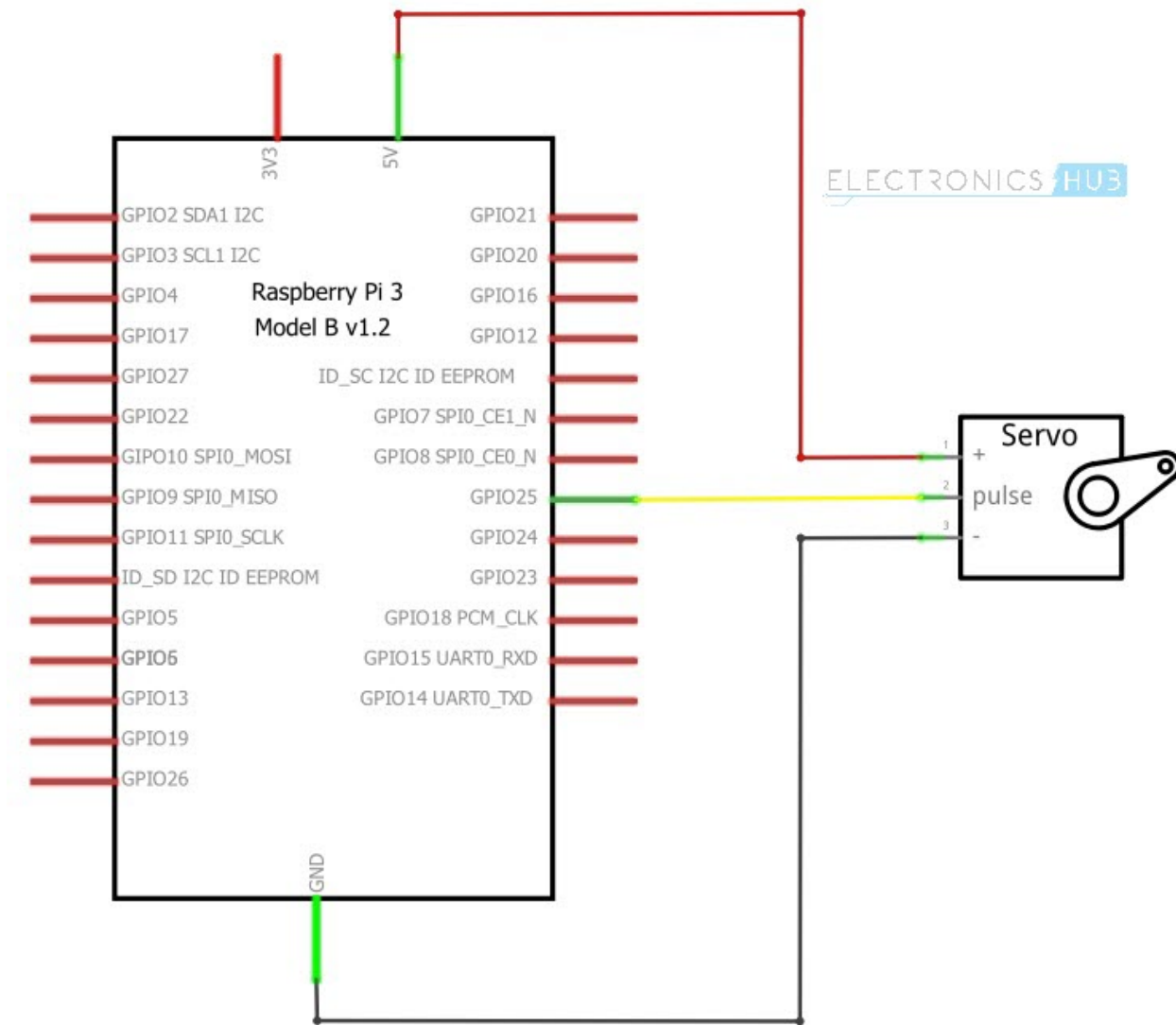
Components Required

- Raspberry Pi
- Tower Pro SG90 Servo Motor (any Servo Motor can be used)
- Connecting Wires
- Power Supply
- Computer

Circuit Design

Connect the V_{CC} and GND of the Tower Pro SG90 Servo Motor to +5V and GND pins of the power supply. Then connect the PWM Pin of the Servo Motor to Physical Pin 22 of Raspberry Pi i.e. GPIO25.

Make the ground common between Raspberry Pi and the Power Supply of the Servo Motor.

NOTE: I've connected the Servo Motor directly to the +5V pin of the Raspberry Pi. I would not recommend you to do the same.

Code

The code for Controlling a Servo Motor with Raspberry Pi is written in Python. The Python Script for the project is given below.

```python
import RPi.GPIO as GPIO
import time

control = [5,5.5,6,6.5,7,7.5,8,8.5,9,9.5,10]

servo = 22

GPIO.setmode(GPIO.BOARD)

GPIO.setup(servo,GPIO.OUT)
# in servo motor,
# 1ms pulse for 0 degree (LEFT)
# 1.5ms pulse for 90 degree (MIDDLE)
# 2ms pulse for 180 degree (RIGHT)

# so for 50hz, one frequency is 20ms
# duty cycle for 0 degree = (1/20)*100 = 5%
# duty cycle for 90 degree = (1.5/20)*100 = 7.5%
# duty cycle for 180 degree = (2/20)*100 = 10%

p=GPIO.PWM(servo,50)# 50hz frequency

p.start(2.5)# starting duty cycle ( it set the servo to 0 degree )


try:
        while True:
            for x in range(11):
                p.ChangeDutyCycle(control[x])
                time.sleep(0.03)
                print x

```

```
33              for x in range(9,0,-1):
34                  p.ChangeDutyCycle(control[x])
35                  time.sleep(0.03)
36                  print x
37
38      except KeyboardInterrupt:
39          GPIO.cleanup()
40
```

Raspberry_Pi_Servo_Motor.py hosted with ♥ by **GitHub**　　　　　　　**view raw**

Working

The aim of this project is to understand the Raspberry Pi Servo Motor Interface and how to control a simple Servo Motor using Raspberry Pi.

Now, coming to the working of the project, we are going to use the PWM feature of the Raspberry Pi. As mentioned earlies, based on the Duty Cycle of the PWM Signal from the Raspberry Pi, the position of the Servo Motor will vary.

Since, 5% Duty Cycle of the PWM signal corresponds to extreme left position and 10% Duty Cycle corresponds to extreme right position, we need to vary the Duty Cycle between 5 and 10% to get a sweeping effect from the Servo Motor.

If you observe in the code, the duty cycle gradually rises from 5 to 10% with an increment of 0.5% in every step. Once it reaches the 10% mark, the reverse action will begin.

Applications

Interfacing a Servo Motor with Raspberry Pi and controlling the angle of rotation of the Servo can be helpful in several applications like:

- RC Car
- RC Plane
- Robot
- Quadcopter

Related Articles

- **Raspberry Pi – Basic Setup without Monitor and Keyboard**
- **First Python Program on the Raspberry Pi**
- **How to Blink an LED using Raspberry Pi and Python?**
- **Controlling a DC Motor with Raspberry Pi**

Related Posts:

- Controlling a Servo Motor with STM32F103C8T6 Blue Pill
- Bluetooth Controlled Servo Motor using Arduino,…
- How to make a Simple Servo Motor Tester Circuit?
- Web Controlled Servo using ESP8266
- Raspberry Pi Stepper Motor Control using L298N
- Arduino Joystick Interface - Control Servo using…

FILED UNDER: DIY PROJECTS, RASPBERRY PI

Comments

Bonzadog says

FEBRUARY 14, 2018 AT 10:53 AM

Not a big fan of this method. It ties the RPi up just dealing with the software. My approach is to buy a PWM/Servo Hat/Board from Adafruit or Pimoroni and use that. The board take care of all the PWM side. One sends a command to the board to position the servo/PWM. Ok it costs more but I feel that the extra cost is justified.

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

POST COMMENT

Arduino Projects (200+)

Electronics Projects (250+)

Mini Project Circuits (160+)

Mini Project Ideas (150+)

ECE Projects (150+)

EEE Projects (150+)

8051 Projects (110+)

Raspberry Pi Projects (101+)

Electrical Project Ideas (100+)

Embedded Projects (100+)

Latest Electronics Ideas (100+)

Microcontroller Mini Projects (100+)

Robotics Projects (100+)

VLSI Projects (100+)

Solar Projects (100+)

IOT Projects (100+)

Communication Projects (70+)

LED Projects (70+)

Power Electronics Projects (60+)

RFID Projects (60+)

Home Automation Projects (50+)

Matlab Projects (50+)

EIE Projects (50+)

Wireless Projects (50+)

LabView Projects (45+)

Zigbee Projects (45+)

GSM Projects (40+)

555 Timer Circuits (40+)

Sensor Projects (40+)

ARM Projects (60+)

DTMF Projects (30+)

PIC Projects (30+)

Electrical Mini Projects (25)

ESP8266 Projects (15)

KITS

Best Rgb Led Strip Light Kits

Arduino Starter Kit

Electronics Books Beginners

Breadboard Kits Beginners

Best Arduino Books

Diy Digital Clock Kits

Drone Kits Beginners

Best Brushless Motors

Raspberry Pi Books

Electronics Component Kits Beginners

Soldering Stations

Electronics Repair Tool Kit Beginners

Raspberry Pi Starter Kits

Best Waveform Generators

Arduino Robot Kits

Oscilloscope Kits Beginners

Raspberry Pi LCD Display Kits

Robot Cat Toys

FM Radio Kit Buy Online

Best Resistor Kits

Soldering Iron Kits

Best Power Supplies

Best Capacitor Kits

Arduino Sensors

Best Function Generator Kits

Led Christmas Lights

Best Iot Starter Kits

Best Gaming Headsets

Best Python Books

Best Robot Dog Toys

Best Robot Kits Kids

Best Solar Panel Kits

Led Strip Light Kits Buy Online

Top Robot Vacuum Cleaners

Digital Multimeter Kit Reviews

Solar Light Kits Beginners

Best Jumper Wire Kits

Best Gaming Earbuds

Best Wireless Routers

3d Printer Kits Buy Online

Best Gaming Mouse

Electric Lawn Mowers

Best Gaming Monitors

Symbols

Courses

Calculator

Contact

HomeZene

Best Arduino Kits

TechZene

Electronics

Embedded

Power

Robotics

ARM

IOT

Microcontroller

Arduino

Solar

Free circuits

Home Automation

Seminar Topics

Electronics Questions

TUTORIALS

Capacitors

Resistors

Filters

Diodes

Transistors

TUTORIALS

Amplifiers

IO Devices

Thyristors

DC Circuits

Number System

TS EAMCET 2019

FOLLOW US

Instagram

Youtube

Facebook

Google Plus

Twitter

PROJECTS BY CATEGORY

SUBSCRIBE FOR UPDATES

Enter your email address:

SUBSCRIBE

GENERAL

PROJECTS

PROJECTS

Tutorials

Electrical

Mini projects