Word Count: 1533

Video, Working with Ultrasonic Sensor & DC Motors

In this video, we will learn the basics of ultrasonic sensors, its applications, its working principle, and later we will learn how to interface it to the Raspberry Pi. After that, we will learn about the DC Motor, its working principle, and about Motor Drivers. Finally, we will learn how to interface a DC Motor to the Raspberry Pi. We end with an activity to use both the ultrasonic sensor and dc motor to create an obstacle avoidance robot.

The HC-SR04 Ultrasonic sensor is the best candidate, as it is low cost, readily available, and can detect the distance between 2 centimeters to 400 centimeters. The HCSR04 ultrasonic sensor uses sonar to determine the distance to an object like bats or dolphins do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy to-use package. The module has two eyes, that project in the front, which form the Ultrasonic transmitter and Receiver. As shown here, the HC-SR04 Ultrasonic sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground, respectively. The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in the air, and when it gets obstructed by any material, it gets reflected back toward the sensor and this reflected wave is observed by the Ultrasonic receiver module. If we measure the time it took for the sound to return and divide it by 2, we will get the time the sound signal took to reach the object. As we know that the speed of sound in air is 340 meters per second, we can easily calculate the distance between the source and the object with the following formula. The sound frequency is set at 40 kHz, and the Transmitter is enabled by swinging the Trig Pin high. We wait for the sound to reflect from any object in front of it and when it reached the Receiver, Output on the Echo Pin swing from low to high. This is the basic principle of working of an ultrasonic sensor.

The Ultrasonic Sensor is used in a wide variety of applications as follows

1. Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, pathfinding robot etc.

2. Can be used to map the objects surrounding the sensor by rotating it

3. Depth of certain places like wells, pits etc can be measured since the waves can penetrate through the water

We can power the Sensor using a regulated +5V through the Vcc and Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the onboard 5V pins of the Raspberry Pi 4. Let’s start interfacing the sensor to the Raspberry Pi 4. Connect the VCC to any 5 V pin of the Pi and GND to any GND pins of the Pi. Next, connect the TRIG pin of the sensor to GPIO 24 pin. Now we need to connect the ECHO pin to the Pi. The problem is that Raspberry Pi GPIO pins can only tolerate a maximum of 3.3 Volts, but the ECHO pin is working in 5 V logic. Thus, we need to use a voltage divider with a combination of 330 Ohm & 470 Ohm Resistors. Fix a jumper at the junction point of the two resistors and wire it up to GPIO 23 pin. The interfacing is now done.

Now open the Distance Senor.py in thonny IDE and run the script. You can see that, when I move my hand horizontally in front of the Sensor, the distance in meters is printed out in the shell. In the code, we first imported the class DistanceSensor from the gpiozero library. Then we imported the sleep class from the time library. When you create the object using the class Distance Sensor, you need to specify the GPIO pin numbers of the Pi where the ECHO and TRIG pins are connected. And you should enter the numbers in that order only: first the gpio number of ECHO then TRIG. This infinite loops check every second, and uses the sensor.distance method to collect and print out the distance in meters on the shell.

Next let's move on to work with motors in the Raspberry Pi 4. The DC motor is a machine that transforms electrical energy into mechanical energy in the form of rotation. Its movement is produced by the physical behavior of electromagnetism. DC motors have inductors inside, which produce the magnetic field used to generate movement. But how does this magnetic field changes if DC current is being used? An electromagnet, which is a piece of iron wrapped with a wire coil that has voltage applied to its terminals. If two fixed magnets are added on both sides of this electromagnet, the repulsive and attractive forces will produce a torque.

But how can a DC Motor be controlled?

DC motors have only two terminals. If you apply a voltage to these terminals, the motor will run, if you invert the position of the terminals the motor will change its direction. If the motor is running and you suddenly disconnect both terminals, the motor will keep rotating but slowing down until stopping. Finally, if the motor is running and you suddenly short-circuit both terminals, the motor will stop.

Motor requires a high amount of current, whereas the Pi 4 works on low current signals. Thus, we need a special chip called Motor Driver controller to provide that extra power. Motor drivers act as an interface between the motors and the control circuits. So, the function of motor drivers is to take a low-current control signal and then turn it into a higher-current signal that can drive a motor. The Motor driver will require a separate source of power to do this. The most commonly used motor driver is the L239D motor driver IC. This IC can provide up to 600mA of current. Even a small 5V DC Motor draws a high initial current of around 300 – 400 m. Thus this IC fits the bill.

Thus, before interfacing a dc motor to your Pi 4 make sure you have an L239D Motor Driver module and a Power supply capable of providing the Motor Driver with more than 4.5 V at 600mA. The MB102 Breadboard power supply is sufficient for driving motors. First, connect the VCC Pin of the motor driver to the 5V pin of the power supply. Then connect the GND pin of the Motor driver to one of the ground pins of the power supply. The next important step is to make sure that the Pi, the motor driver, and the supply shares the same ground. Thus, connect any other ground pin of the power supply to any ground pin of the Pi. We will now interface a single DC motor across the MOTOR1 output pins. To control the first motor’s direction of rotation we are given two input pins IN1 and IN2. First, connect the IN1 to GPIO4, then connect IN2 to GPIO14 on the Raspberry Pi 4. Now the basic interfacing is done. So when the IN1 is high, and IN2 is low, the Motor rotates forwards, and when the IN2 is high, and IN1 is low, it rotates backward.

Open the DC Motor.py script in the repository, and run it. Now you can see that the motor rotates forwards, then backwards, then stops and finally continue rotating backwards. Each operation was given 5 seconds of time, in the code. In the code, you can see that we first imported the Motor Class from the gpiozero library, along with the sleep class from the time library. During object creation, we have entered these two parameters to define the pin functions. We have assigned GPIO pin 4 to the forward variable, and GPIO 14 to the backward variable. This basically tells the Pi that if you set pin 4 high the motor will move forwards, and if the pin 14 is set high it will move backwards. These three methods help to initiates the different motions for the motor. Its either forward, backward or stop.

The Gpiozero library creators have gone even further to create a separate library called Robot that is specially designed to work with 2 motors, and makes it very simple to do all the movements like left, right, forward, backward, and stop very easily. You can see how easy it is from the code shown on the screen. I will now give you an activity. Make a Robot that uses the distance sensor to detect things nearby, and moves away from an obstacle if it is within 20cm of the robot. I was able to do this with just 7 lines of code. What about you?! Do share your code in the assignment section. I have provided the code for the obstacle avoidance robot in the resources section

Summary

In this video, we have covered the following

● Working Principle & Interfacing of Ultrasonic Sensor HCSR04

● Working Principle & Interfacing of DC Motors & Motor Driver

● Activity to create an obstacle avoidance robot

In the next video, we will learn to work with UART, I2C & SPI Protocols in the Raspberry Pi 4