LAMPIRAN

12V DC / 2A

pro-ELEC

RoHS Compliant

Input Characteristics

rated output load & 25°C ambient
AC Leakage Current : 0.25mA Max. @ 240V AC input

Regulated AC Power Adapter

Rated Voltage : 100 – 240V AC
Variation Range : 90 – 264V AC
Rated Frequency : 50/60Hz
Variation Frequency : 47 – 63Hz
Input Current : 0.6A Max. @ Any input AC
voltage &
output full load
Inrush Current : 35A Max. Cold start @ 240V
AC input,

Output Characteristics

Voltage	Min. Load	Rated Load	Output Power
12V DC	0.01A	2A	24W

Combined Load / Line Regulation

Voltage	Min. Load	Rated Load	Line Regulation	Load Regulation

12V DC	0.01A	2A	± 1%	± 3%

Ripple and Noise : Under normal voltage & nominal load, the ripple & noise are as follows when measure with Max. Bandwidth of 20MHz & parallel 10μF electrolysis capacitor & 0.1μF ceramic capacitor crossed connect at testing point.
Voltage: +12V DC
Ripple & Noise (Max.): 120mVp-p

Turn on Delay Time : 2S Max. @ 115V AC input and output Max. Load

Rise Time : 40mS Max. @ 115V AC input and output Max. Load

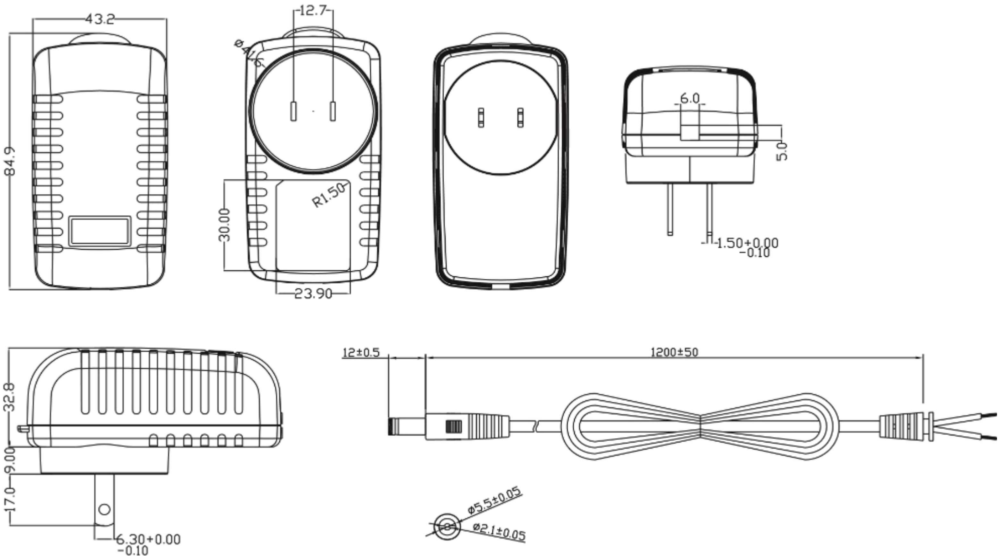
Hold Up Time : 5mS Min. @ 115V AC input and output Max. Load

Efficiency : A verage efficiency 87.4% minimum @ 25%, 50%, 75% & 100% of full-loading and 230V AC input (After warm up 30 minutes). Meet with Energy star level VI.
No load input watt ≤ 0.1W

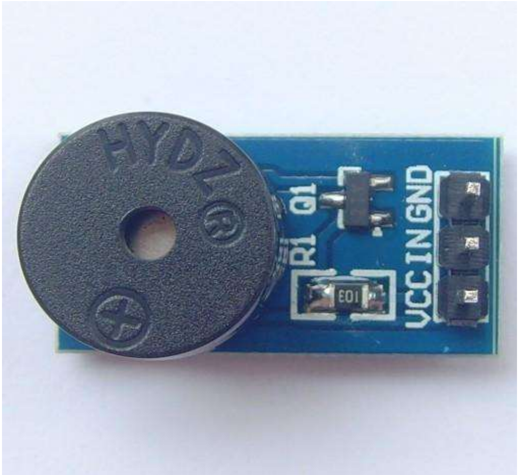
Overshoot : 10% Max. When power supply @ turn on or turn off.

Regulated AC Power Adapter

Diagram



5V Buzzer Module



General Description:

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke.

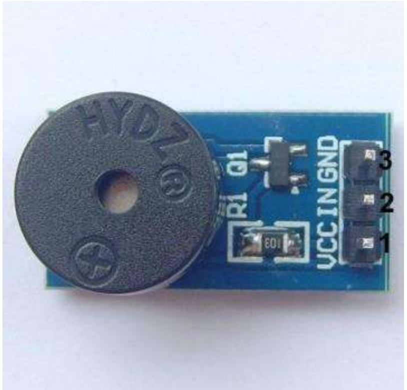
Buzzer is an integrated structure of electronic transducers, DC power supply, widely used in computers, printers, copiers, alarms, electronic toys, automotive electronic equipment, telephones, timers and other electronic products for sound devices. Active buzzer 5V Rated power can be directly connected to a continuous sound, this section dedicated sensor expansion module and the board in combination, can complete a simple circuit design, to "plug and play."

Specifications:

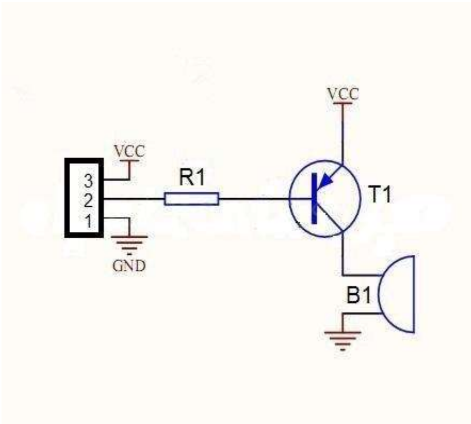
- On-board passive buzzer
- On-board 8550 triode drive
- Can control with single-chip microcontroller IO directly
- Working voltage: 5V
- Board size: 22 (mm) x12 (mm)

Pin Configuration:

1. VCC
2. Input
3. Ground



Schematic Diagram:



How to test:

1. Connect your Arduino microcontroller to the computer.
2. Connect the VCC pin of your module to the to the 5V pin of your Arduino.
3. Connect the GND pin of your module to the GND pin of your Arduino.
4. Connect the Input pin of your module to the pin 13 of your Arduino.
5. Enter this program to your Arduino Integrated Development Environment (IDE):

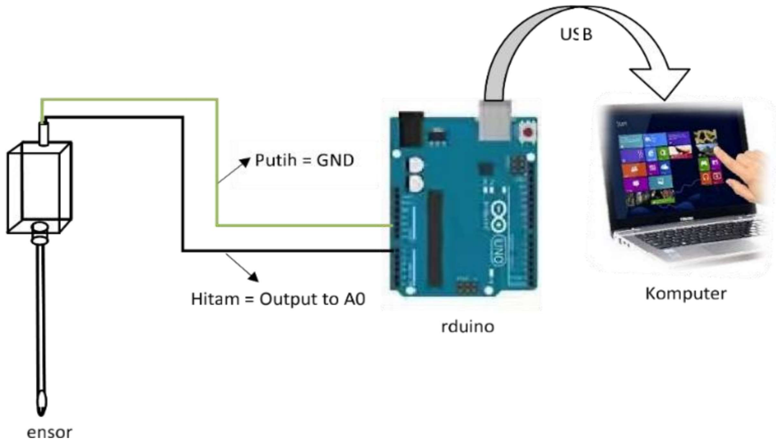
```
int buzzer = 13;
void setup()
{
  pinMode(buzzer,
OUTPUT);
}


void loop()
{
 digitalWrite(buzzer,
HIGH);   delay(1000);
 digitalWrite(buzzer,
LOW);
 delay(1000);
}
```

6. Lastly, click the Upload Button.

Testing Results:

The sample sketch above is a blink which is also applicable for LEDs. The output is the turning on and off of the buzzer every other second. The picture below shows the setup of your module and Arduino:

DIAGRAM KONEKSI



DATA UJI SENSOR

y = -0.0693x + 7.3855, dimana : x = nilai ADC, dan y=pH

LAMPIRAN GAMBAR

ESP32 Datasheet

Espressif Systems

October 8, 2016

About This Guide

This document provides introduction to the specifications of ESP32 *hardware*.

The document structure is as follows:

Chapter	Title	Subject
Chapter 1	Overview	An overview of ESP32, including featured solutions, basic and advanced features, applications and development support
Chapter 2	Pin Definitions	Introduction to the pin layout and descriptions
Chapter 3	Functional Description	Description of the major functional modules
Chapter 4	Peripheral Interface	Description of the peripheral interfaces integrated on ESP32

Chapter 5	Electrical Characteristics	The electrical characteristics and data of ESP32
Chapter 6	Package Information	The package details of ESP32
Chapter 7	Supported Resources	The related documents and community resources for ESP32
Appendix	Touch Sensor	The touch sensor design and layout guidelines

Release Notes

Date	Version	Release notes
2016.08	V1.0	First release

Application

- Generic low power IoT sensor hub
- Generic low power IoT loggers
- Video streaming from camera
- Over The Top (OTT) devices • Music players
  - Internet music players
  - Audio streaming devices
- Wi-Fi enabled toys
  - Loggers
  - Proximity sensing toys
- Wi-Fi enabled speech recognition devices
- Audio headsets
- Smart power plugs

- Home automation
- Mesh network
- Industrial wireless control
- Baby monitors
- Wearable electronics
- Wi-Fi location-aware devices
- Security ID tags
- Healthcare
  - Proximity and movement monitoring trigger devices
  - Temperature sensing loggers

Block Diagram

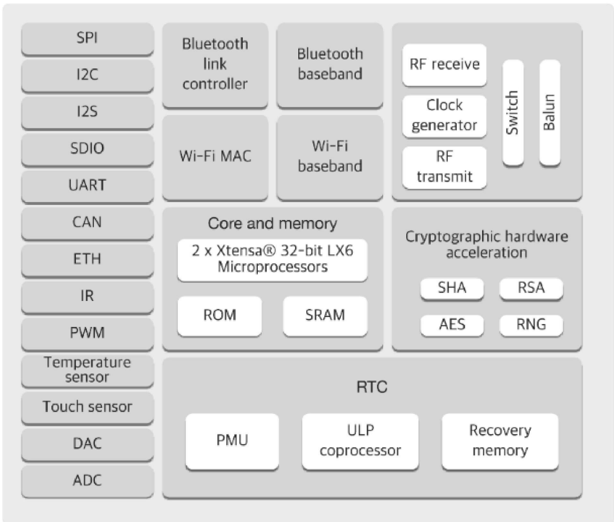
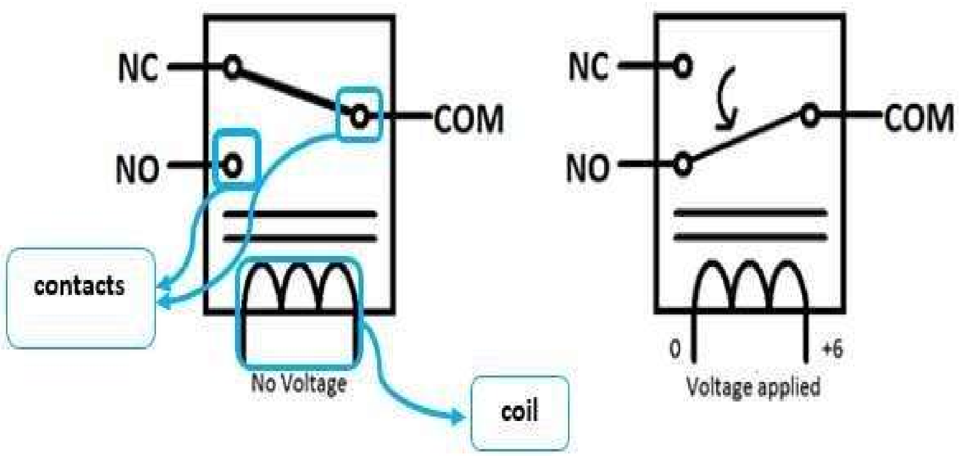


Figure 1: Function Block Diagram

RELAY MODULES

RELAY WORKING IDEA

Relays consist of three pins normaly open pin , normaly closed pin, common pin and coil. When coil powerd on magntic field is generated the contacts connected to each other.

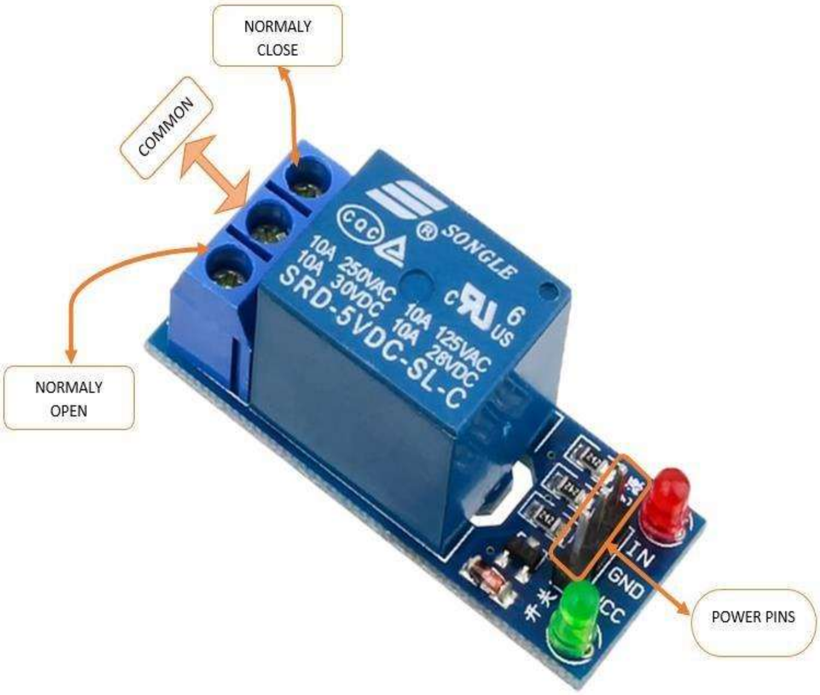


Relay modules 1-channel features

- Contact current 10A and 250V AC or 30V DC.
- Each channel has indication LED.
- Coil voltage 12V per channel.
- Kit operating voltage 5-12 V
- Input signal 3-5 V for each channel.
- Three pins for normally open and closed for each channel.

How to connect relay module with Arduino

As shown in relay working idea it depends on magnetic field generated from the coil so there is power
isolation between the coil and the switching pins so coils can be easily powered from Arduino connecting VCC and GND bins from Arduino kit to the relay module kit after that we choose Arduino
output pins depending on the number of relays needed in project designed and set these pins to output
and make it out high (5 V) to control the coil that allow controlling of switching process.



NOTE : whatever was the relay channels number the pinconfiguration is the same for every channel except the power pins (VCC and GND) are for the board itself. The input signal (IN) pin for every relay.

PIR Sensor (HC-SR501)

Introduction

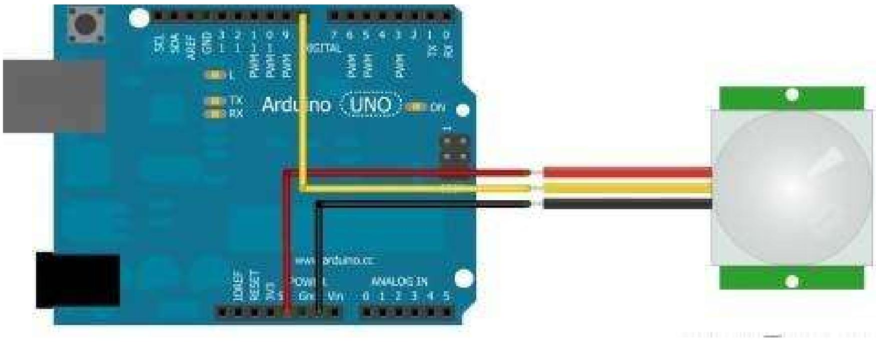
Passive Infra-Red (PIR) sensors are used to detect motion based on the infrared heat in the surrounding area. This makes them a popular choice when building a system to detect potential intruders or people in general. These sensors can take for 10-60 seconds to warm up, so try to avoid motion during that time.

Parts

- Arduino
- PIR Sensor
- Wires

Schematic

Below is the schematic for using a PIR sensor. It is fairly simple.



Code

Adafruit has a really good tutorial for how these sensors are used and various projects for them.

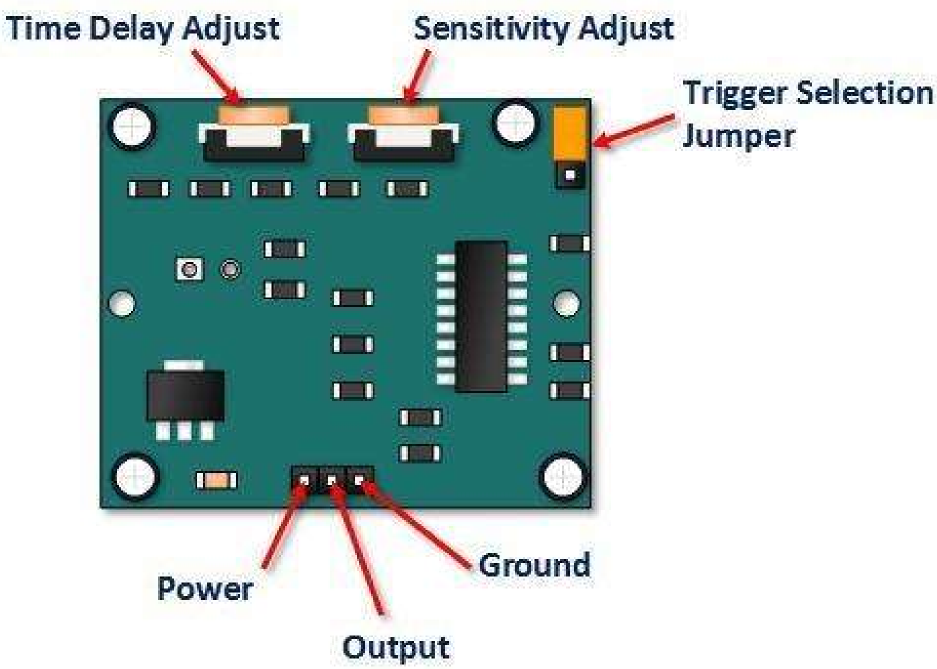
Below is the code for working with a PIR sensor. It should be noted that the PIR sensor does not respond immediately when motion stops. This has to do with the two potentiameters on the sensor.

HC-SR501 Passive Infrared (PIR) Motion Sensor

This motion sensor module uses the LHI778 Passive Infrared Sensor and the BISS0001 IC to control how motion is detected.

The module features adjustable sensitivity that allows for a motion detection range from 3 meters to 7 meters, also includes time delay adjustments and trigger selection that allow for fine tuning within your application.

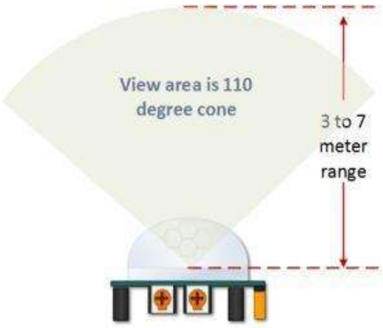
HC-SR501 Pin Outs and Controls



Pin or Control	Function
Time Delay Adjust	Sets how long the output remains high after detecting motion.... Anywhere from 5 seconds to 5 minutes.
Sensitivity Adjust	Sets the detection range.... from 3 meters to 7 meters
Trigger Selection Jumper	Set for single or repeatable triggers.
Ground pin	Ground input

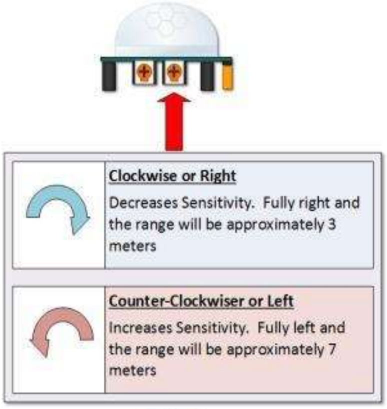
Output Pin Low when no motion is detected.. High when motion is detected. High is 3.3V

Power Pin 5 to 20 VDC Supply input



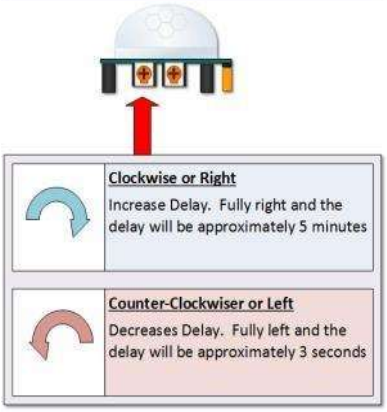
Device Area of Detection

The device will detect motion inside a 110 degree cone with a range of 3 to 7 meters.



PIR Range (Sensitivity) Adjustment

As mentioned, the adjustable range is from approximately 3 to 7 meters.



Time Delay Adjustment

The time delay adjustment determines how long the output of the PIR sensor module will remain high after detection motion.The range is from about 3 seconds to five minutes.

Seconds Off After Time Delay Completes – IMPORTANT

The output of this device will go LOW (or Off) for approximately 3 seconds AFTER the time delay completes. In other words, ALL motion detection is blocked during this three second period.

For Example:

- Imagine you're in the single trigger mode (see below) and your time delay is

set 5 seconds. o The PIR will detect motion and set it high for 5 seconds.
o After five seconds, the PIR will sets its output low for about 3 seconds.

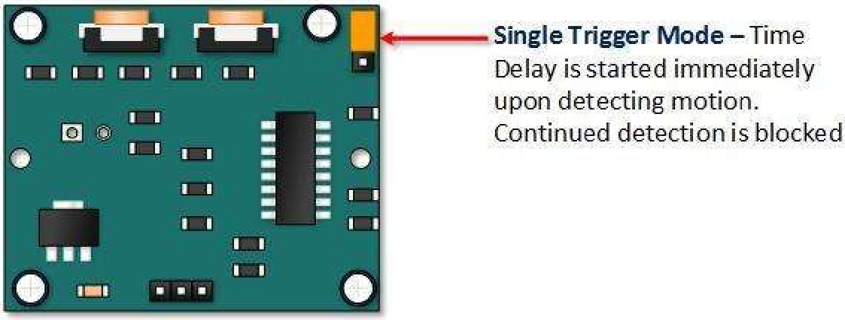
o During the three seconds, the PIR will not detect motion. o After three seconds, the PIR will detect motion again and detected motion will once again set the output high and the output will remain on as dictated by the Time Delay adjustment and trigger mode selection.

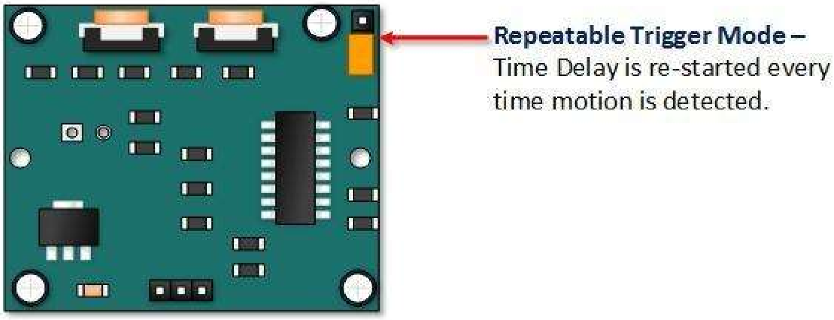
OVERRIDING THE TIME DELAY – If you're connecting your HC-SR501 to an Arduino, it is likely that you are going to take some sort of action when motion is detected. For example, you may wish to brighten lights when motion is detected and dim the lights when motion is no longer connected Simply delay dimming within your sketch.

Trigger Mode Selection Jumper

The trigger mode selection jumper allows you to select between single and repeatable triggers. The affect of this jumper setting is to determine when the time delay begins.

- **SINGLE TRIGGER** – The time delay begins immediately when motion is first detected.
- **REPEATABLE TRIGGER** – Each detected motion resets the time delay. Thus the time delay begins with the last motion detected.





HC-SR501 Dance Floor Application Examples

Imagine that you want to control lighting on a dance floor based upon where the dancers are dancing.

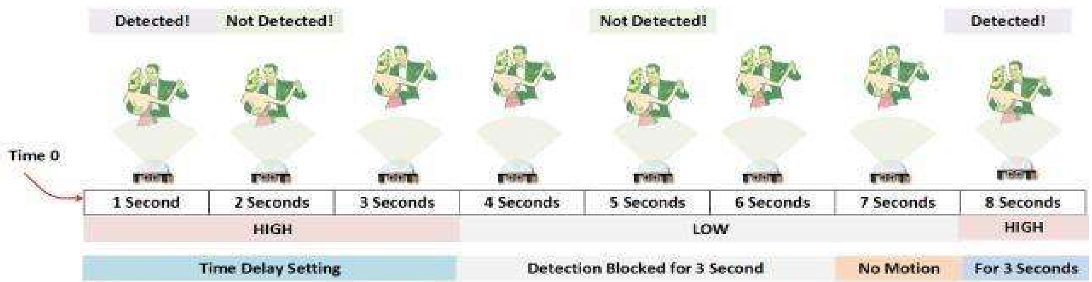
Understanding how the time delay and trigger mode interact will be necessary to controlling that lighting in the manner that you want.

Example One

In this first example, the time delay is set to three seconds and the trigger mode is set to single.

As you can see in the illustration below,the motion is not always detected.

In fact, there is a period of about six seconds where motion can not be detected.



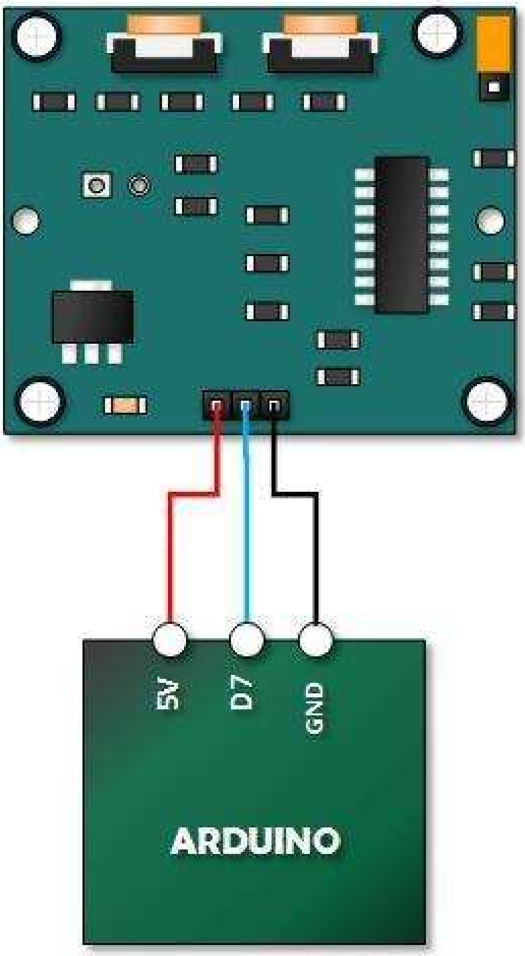
Example Two

In the next example, the time delay is still at three seconds and the trigger is set to repeatable. In the illustration below, you can see that the time delay period is restarted.

However, after that three seconds, detection will still be blocked for three seconds.As I mentioned previously, you could override the 3 second blocking period with some creative code, but do give that consideration.

Some of the electronics you use may not like an on and then off jolt.The three seconds allows for a little rest before starting back up.

Arduino HC-SR501 Motion Sensor Tutorial, connect Your Arduino to the HC-SR501



The sketch simply turns on Your Arduino LED connected to Pin 13 whenever motion is detected.

Be sure to beware of and somehow handle the 1 minute initialization in whatever application you develop.

HC-SR04 Ultrasonic Sensor

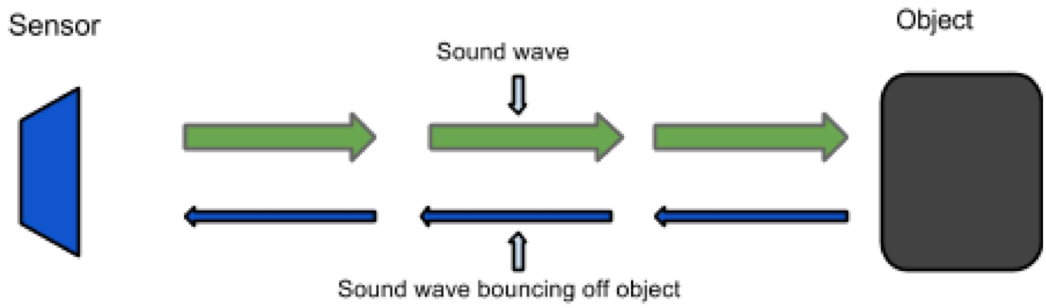
Elijah J. Morgan

Nov. 16 2014

The purpose of this file is to explain how the HC-SR04 works. It will give a brief explanation of how ultrasonic sensors work in general. It will also explain how to wire the sensor up to a microcontroller and how to take/interpret readings. It will also discuss some sources of errors and bad readings.

1. How Ultrasonic Sensors Work

Ultrasonic sensors use sound to determine the distance between the sensor and the closest object in its path. How do ultrasonic sensors do this? Ultrasonic sensors are essentially sound sensors, but they operate at a frequency above human hearing.



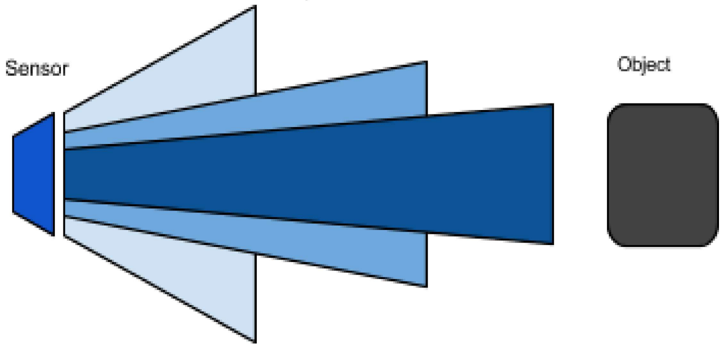
The sensor sends out a sound wave at a specific frequency. It then listens for that specific sound wave to bounce off of an object and come back (Figure 1). The sensor keeps track of the time between sending the sound wave and the sound wave returning. If you know how fast something is going and how long it is traveling you can find the distance traveled with equation 1.

Equation 1. $d = v \times t$

The speed of sound can be calculated based on the a variety of atmospheric conditions, including temperature, humidity and pressure. Actually calculating the distance will be shown later on in this document.

It should be noted that ultrasonic sensors have a cone of detection, the angle of this cone varies with distance, Figure 2 show this relation. The ability of a sensor to detect an object also depends on the objects orientation to the sensor. If an object doesn't present a flat surface to the sensor then it is possible the sound wave will bounce off the object in a way that it does not return to the sensor.

Figure 2



2. HC-SR04 Specifications

The sensor chosen for the Firefighting Drone Project was the HC-SR04. This section contains the specifications and why they are important to the sensor module. The sensor modules requirements are as follows.

- Cost
- Weight
- Community of hobbyists and support
- Accuracy of object detection
- Probability of working in a smoky environment
- Ease of use

The HC-SR04 Specifications are listed below. These specifications are from the

Cytron Technologies HC-SR04 User's Manual (source 1).

- Power Supply: +5V DC
- Quiescent Current: <2mA
- Working current: 15mA
- Effectual Angle: <15º
- Ranging Distance: 2-400 cm
- Resolution: 0.3 cm
- Measuring Angle: 30º
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm
- Weight: approx. 10 g

The HC-SR04's best selling point is its price; it can be purchased at around $2 per unit.

3. Timing Chart and Pin Explanations

The HC-SR04 has four pins, VCC, GND, TRIG and ECHO; these pins all have different functions. The VCC and GND pins are the simplest -- they power the HC-SR04. These pins need to be attached to a +5 volt source and ground respectively. There is a single control pin: the TRIG pin. The TRIG pin is responsible for sending the ultrasonic burst. This pin should be set to HIGH for 10 μs, at which point the HC-SR04 will send out an eight cycle sonic burst at 40 kHZ. After a sonic burst has been sent the ECHO pin will go HIGH. The ECHO pin is the data pin -- it is used in taking distance measurements. After an ultrasonic burst is sent the pin will go HIGH, it will stay high

until an ultrasonic burst is detected back, at which point it will go LOW.

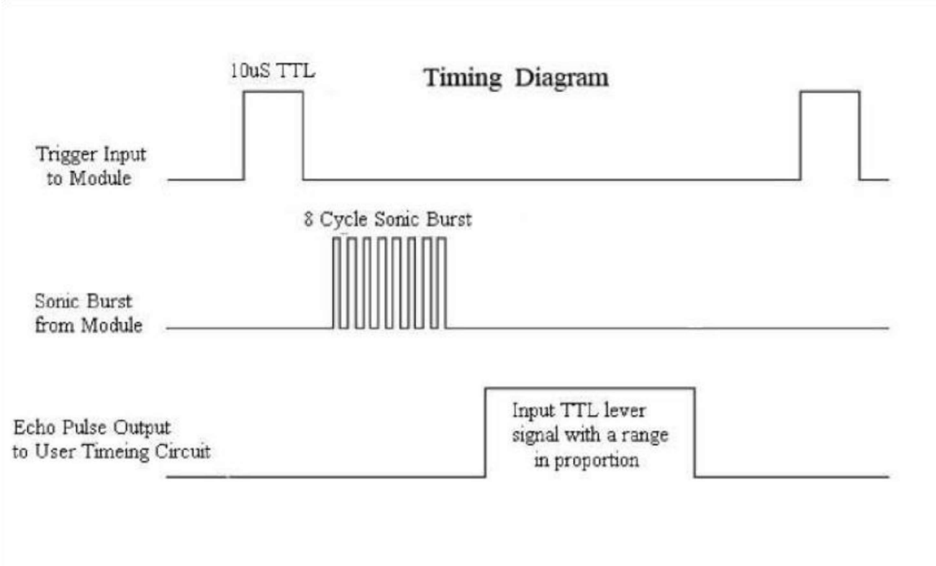
Taking Distance Measurements

The HC-SR04 can be triggered to send out an ultrasonic burst by setting the TRIG pin to HIGH. Once the burst is sent the ECHO pin will automatically go HIGH. This pin will remain HIGH until the the burst hits the sensor again. You can calculate the distance to the object by keeping track of how long the ECHO pin stays HIGH. The time ECHO stays HIGH is the time the burst spent traveling. Using this measurement in equation 1 along with the speed of sound will yield the distance travelled. A summary of this is listed below, along with a visual representation in Figure 2.

1. Set TRIG to HIGH
2. Set a timer when ECHO goes to HIGH

3. Keep the timer running until ECHO goes to LOW
4. Save that time
5. Use equation 1 to determine the distance travelled

Figure 3
Source 2



Source 2

To interpret the time reading into a distance you need to change equation 1. The clock on the device you are using will probably count in microseconds or smaller. To use equation 1 the speed of sound needs to determined,which is 343 meters per second at standard temperature and pressure. To convert this into more useful form use equation 2

to change from meters per second to microseconds per centimeter. Then equation 3 can be used to easily compute the distance in centimeters.

Equation 2. *Distance* = 170.15Speed *m* × Meters100 *cm* × 170.151e6 µS *m* × 58.772*cm* µS

Equation 3. *Distance* = time58 = µsµ/cms__= *cm*

4. Wiring the HC-SR04 to a Microcontroller

This section only covers the hardware side. For information on how to integrate the software side, look at one of the links below or look into the specific microcontroller you are using.

The HC-SR04 has 4 pins: VCC, GND, TRIG and ECHO.

1. VCC is a 5v power supply. This should come from the microcontroller
2. GND is a ground pin. Attach to ground on the microcontroller.
3. TRIG should be attached to a GPIO pin that can be set to HIGH
4. ECHO is a little more difficult. The HC-SR04 outputs 5v, which could destroy many microcontroller GPIO pins (the maximum allowed voltage varies). In order to step down the voltage use a single resistor or a voltage divider circuit. Once again this depends on the specific microcontroller you are using, you will need to find out its GPIO maximum voltage and make sure you are below that.

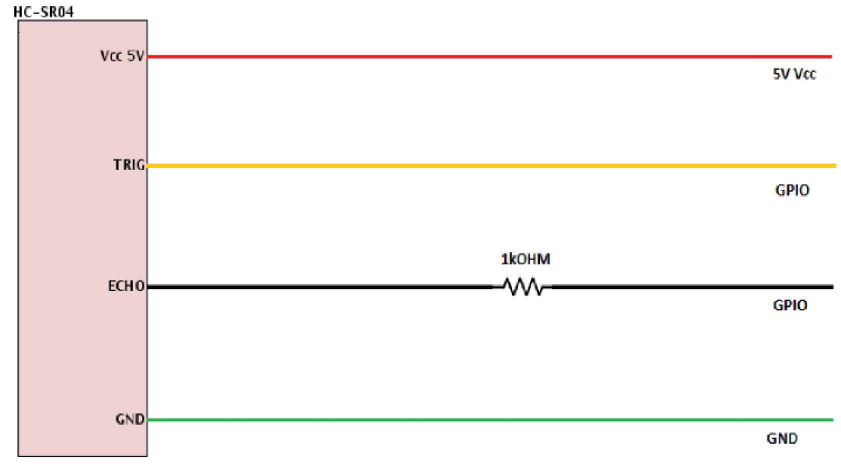
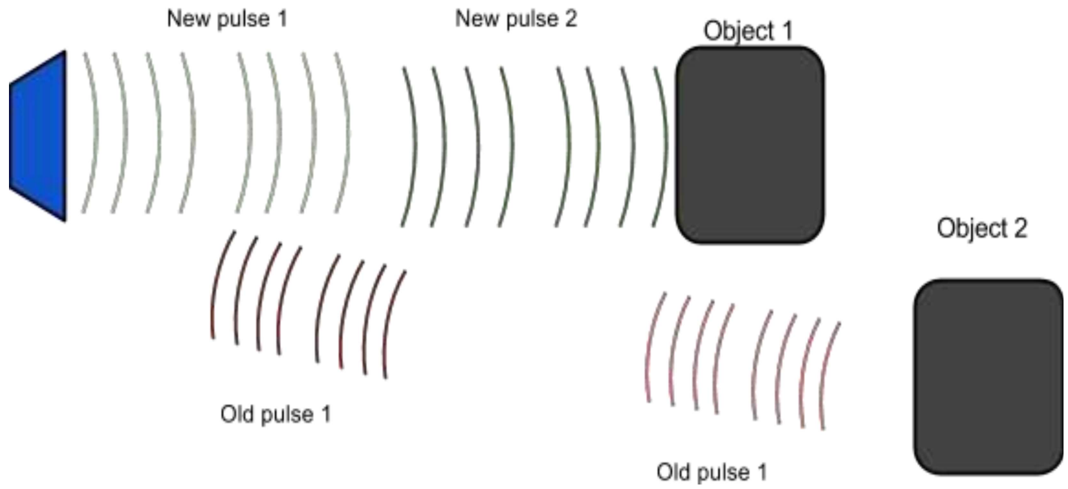


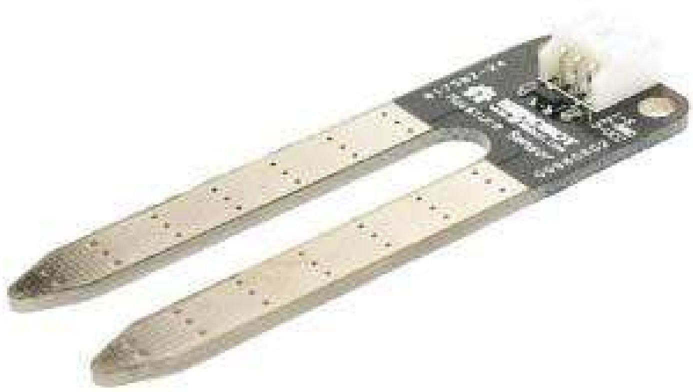
Figure 4

5. Errors and Bad Readings

Ultrasonic sensors are great sensors -- they work well for many applications where other types of sensors fall short. Unfortunately, they do have weaknesses. These weaknesses can be mitigated and worked around, but first they must be understood. The first weakness is that they use sound. There is a limit to how fast ultrasonic sensors can get distance measurements. The longer the distance, the slower they are at reporting the distance. The second weakness comes from the way sound bounces off of objects. In enclosed spaces it is possible, if not probable that there will be unintended echos. The echos can very easily cause false short readings. In Figure 2 a pulse was sent out. It bounced off of object 1 and returned to the sensor. The distance was recorded and then a new pulse was sent. There was another object farther away, so that when the new pulse reaches object 1, the first signal will reach the sensor. This will cause the sensor to think that there is an object closer than is actually true. The old pulse is smaller than the new pulse because it has grown weaker. The longer the pulse exists the weaker it grows until it is negligible. If multiple sensors are being used, the number of echos will increase along with the number of errors. There are two main ways to reduce the number of errors. The first is to provide shielding around the sensor. This prevents echos coming in from angle outside what the sensor should actually pick up. The second is to reduce the frequency at which pulses are sent out. This gives more time for the echos to dissipate.



Moisture Sensor (SKU:SEN0114)



Contents

1. Introduction

This moisture sensor can read the amount of moisture present in the soil surrounding it. It's a low tech sensor, but ideal for monitoring an urban garden, or your pet plant's water level. This is a musthave tool for a connected garden!

This sensor uses the two probes to pass current through the soil, and then it reads that resistance to get the moisture level. More water makes the soil conduct electricity more easily (less resistance), while dry soil conducts electricity poorly (more resistance).

It will be helpful to remind you to water your indoor plants or to monitor the soil moisture in yourgarden.

2. Specification

- Power supply: 3.3v or 5v
- Output voltage signal: 0~4.2v
- Current: 35mA

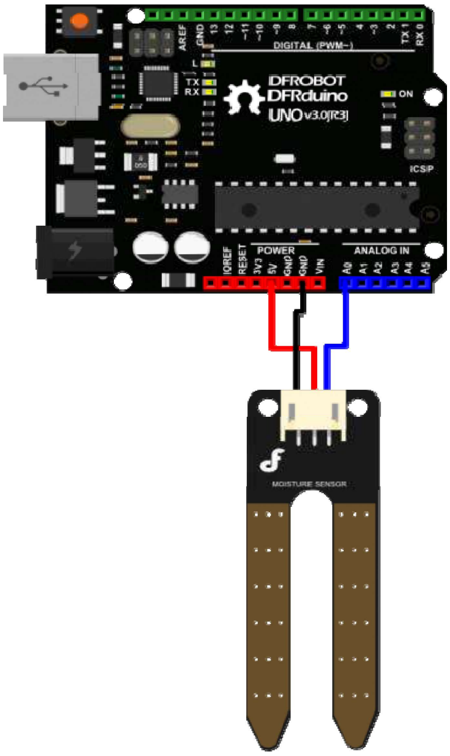
  Pin definition:
- Analog output(Blue wire)
- GND(Black wire)
- Power(Red wire)
- Size: 60x20x5mm

  Value range:
- 0 ~300 : dry soil
- 300~700 : humid soil
- 700~950 : in water

  Specification:
- Power supply: 3.3v or 5v
- Output voltage signal: 0~4.2v
- Current: 35mA
- Pin definition:
- Analog output(Blue wire)
- GND(Black wire)
- Power(Red wire)
- Size: 60x20x5mm

- Value range:
- 0 ~300 : dry soil
- 300~700 : humid soil
- 70050 : in water

3. Usage



Moisture sensor Connection diagram

Open the Arduino Serial Monitor, and choose its baud rate 57600 as set in the code.