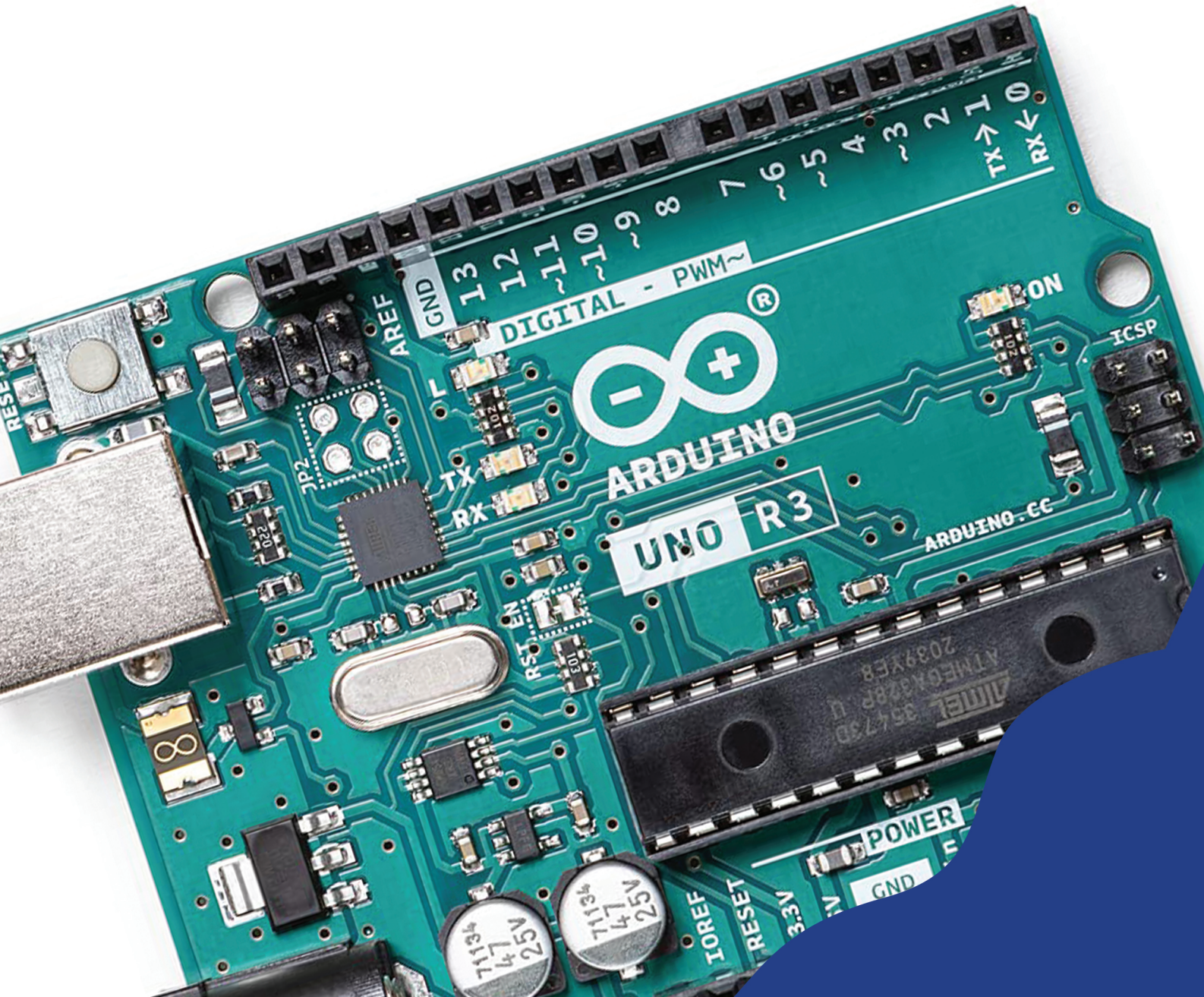
ROBOTROD
by Olatus Systems Private Limited

ARDUINO
CODING/ROBOTICS



Contents

4.TRIPLE AXIS MAGNETOMETER

7.LINE FOLLOWING ROBOT

8.ADVANCED LINE FOLLOWING ROBOT

CHAPTER 1: NATURAL GAS SENSOR(MQ-4)

Introduction

Sensitive material of MQ-4 gas sensor is SnO2, which with lower conductivity in clean air. When the target combustible gas exists, the sensor's conductivity is higher along with the gas concentration rising. Please use simple electro circuit, convert achange of conductivity to the corresponding output signal of gas concentration. The MQ-4 gas sensor has high sensitivity to Methane, also to Propane and Butane. The sensor could be used to detect different combustible gas, especially Methane; it is with low cost and suitable for different application.

A **gas sensor** is a device that detects the presence of gases in an area, often as part of a safety system. This type of equipment is used to detect a gas leak or other emissions and can interface with a control system so a process can be automatically shut down. A gas detector can sound an alarm to operators in the area where the leak is occurring, giving them the opportunity to leave. This type of device is important because there are many gases that can be harmful to organic life, such as humans or animals.

FEATURES

- High sensitivity to CH4, Natural gas.
- Small sensitivity to alcohol, smoke.
- Fast response.
- Stable and long life.
- Simple drive circuit.

MQ4 Gas SensorOverview

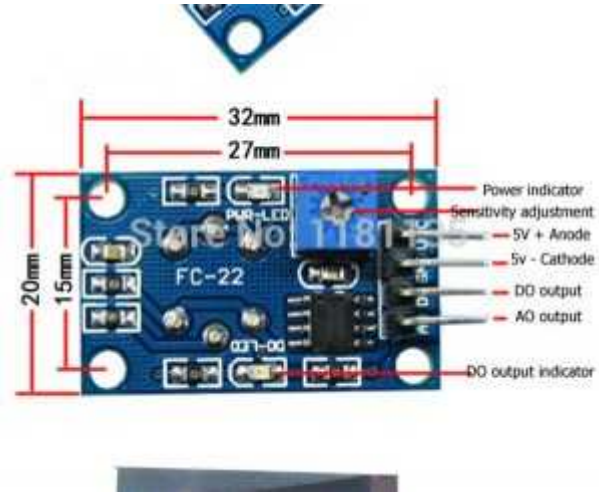
MQ series sensors use a small heater inside with an **electrochemical sensor** in order to measure a different kind of gases combinations. They can be calibrated, but, in order to do that, a known concentration of the measured gas or gasses is needed for that. For the industrialpurpose, calibrations are done in special metrology laboratories with accurate probes and tests. In our case, we will test it as it comes from the producer without any additional calibration or settings. The main idea of ordering this kind of sensor was to build a homemade alarm sensor, which should make some alarming noise or light whenever someone forgot the cooker stove one, or my little kid learned

to play with stove switches, or there is a leak in my gas installation. To accomplish that, I wanted a methane gas, simple to use, and also compatible with Arduino platform. The producer says that MQ4 sensor can sense **methane** / **natural gas** easily with a range sensitivity from **300 to 10000ppm**, costs are **very low**, and can be easily plugged with Arduino boards.

MQ4 sensor pinout connections and power consumption

This model comes with 4 pins:

- 1 pin VCC 5v
- 1 pin GND
- 1 pin DO (digital output) TTL digital 0 and 1 (0.1 and 5V)
- 1 pin AO (analog output) 0.1-0 .3 V (clean), the highest concentration of voltage around 4V.



MQ4 model must be powered with **stable 5v** and needs at least **150mA** (*best to have 250mA*) according to the datasheet declaration, to be able to work properly. Also before getting stable measurements, this model needs at least 1 minute to heat up. **Be aware** that in some datasheets use the term **"preheat"**, which means that some versions need from **12h to 24h to burn-in the sensor**. Only after this, you will be able to get consistent data. Also, this kind of devices, which have an internal heater, are **pretty sensible to ambient influences** like **humidity** or **moisture**.

Application

- Domestic gas leakage detector
- Industrial Combustible gas detector
- Portable gas detector

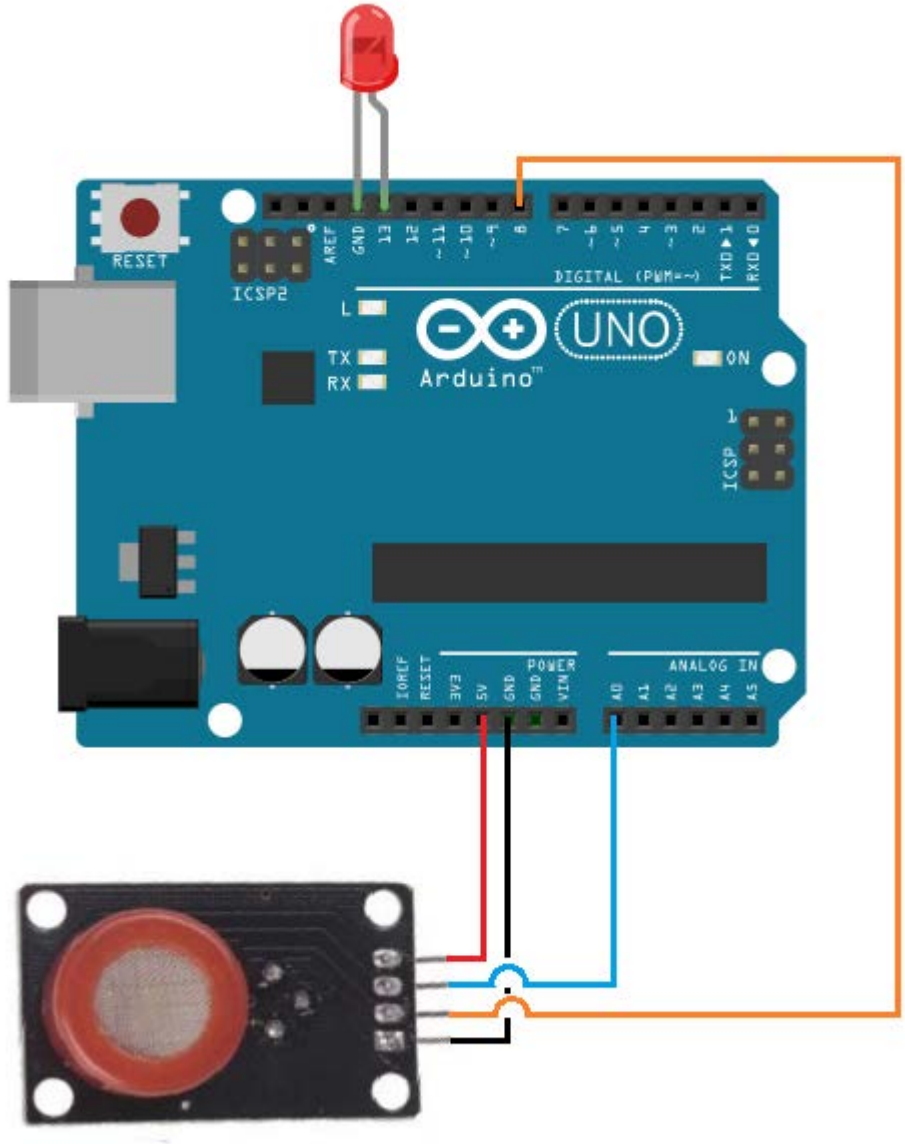
Activity

Write a program read MQ-4 sensor data.

HARDWARE REQUIRED:

- Arduino UNO board
- MQ4 gas sensor
- Jumpers wires/cables

CIRCUIT CONNECTION



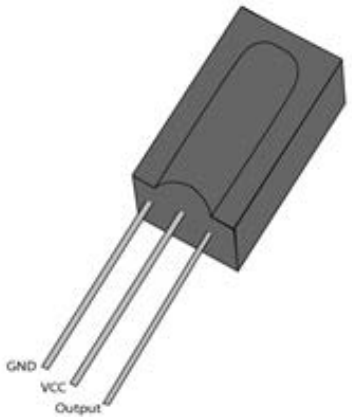
CODE

```
const int gasPin = A0;
void setup()
        {
                Serial.begin(9600);
        }

void loop()
        {
                Serial.println(analogRead(gasPin));
                delay(150);
        }
```

CHAPTER 2: TSOP (1738)-IR RECEIVER

INTRODUCTION

The **TSOP 1738** is a member of **IR remote control receiver** series. This IR sensor module consists of a PIN diode and a preamplifier which are embedded into a single package. The output of **TSOP** is active low and it gives +5V in off state. When IR waves, from a source, with a center frequency of 38 kHz incident on it, its output goes low. Lights coming from sunlight, fluorescent lamps etc. may cause disturbance to it and result in undesirable output even when the source is not transmitting IR signals. A band pass filter, an integrator stage, and an automatic gain control are used to suppress such disturbances.

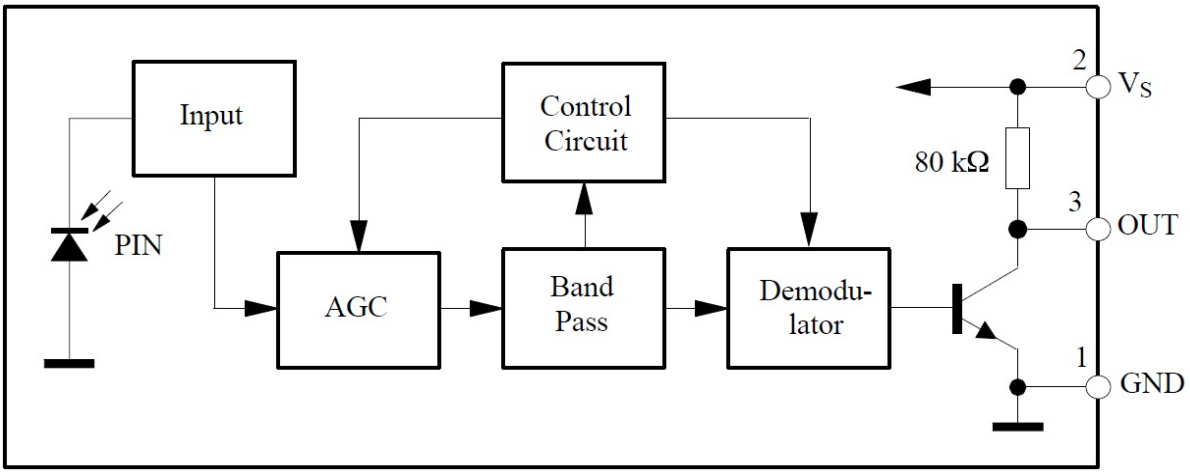


TSOP module has an inbuilt control circuit for amplifying the coded pulses from the IR transmitter. A signal is generated when PIN photodiode receives the signals. This input signal is received by an automatic gain control (AGC). For a range of inputs, the output is fed back to AGC in order to adjust the gain to a suitable level. The signal from AGC is passed to a band pass filter to filter undesired frequencies. After this, the signal goes to a demodulator and this demodulated output drives an NPN transistor. The collector output of the transistor is obtained at pin 3 of TSOP module.

Members of TSOP17xx series are sensitive to different center frequencies of the IR spectrum. For example, TSOP1738 is sensitive to 38 kHz whereas **TSOP1740** to 40 kHz center frequency.

FEATURES

- Photo Detector, IR Filter, Preamplifier and PCM frequency filter in one package
- Shielding against EMI or RFI interference
- CMOS and TTL Compatible
- Active Low Output
- Immunity against ambient light
- Low Power Consumption
- Able to transfer data continuously up to 2400bps
- Supply Voltage: 4.5 – 5.5V



BLOCK DIAGRAM

TSOP17XX are provided with Band Pass Filter, Automatic Gain Control (AGC) and Integrator to avoid changes in output due to noises or other disturbances. Noises and Data signal are distinguished by using carrier frequency, burst length, and duty cycle. Data Signals should satisfy following conditions.The carrier frequency of the received signal should be close the center frequency of the band-pass filter of the device.

For example, Carrier frequency should be close to 38KHz for TSOP1738.
A gap time of at least 14 cycles is necessary after each burst.
A gap of at least same length as theburst is necessary at some point in the data stream for bursts longer than 1.8ms.
It can receive up to 1400 short bursts continuously.

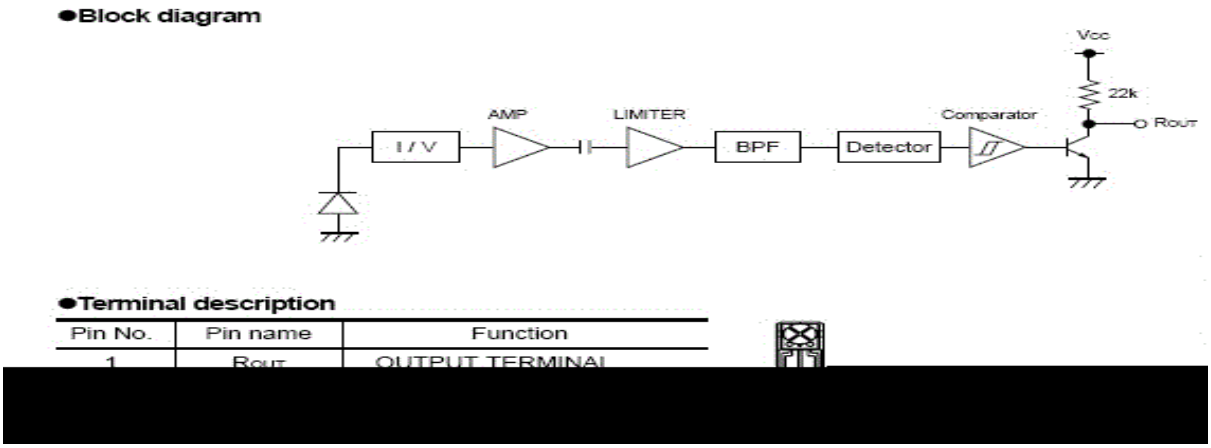
Working with TSOP IR Receiver

The TSOP outputs a constant HIGH signal when idle and as it receives data, it tends to invert the data. i.e. when an IR LED is transmitting data onto the TSOP, every time the IR led goes high, the TSOP will go LOW and vice versa. Remote control signals are often bytes of data that is encoded and transmitted by pulsing (switching ON & OFF the IR LED at a specific frequency) Most TV remote controls work at 32-40 kHz frequency and most receivers can receive this range.

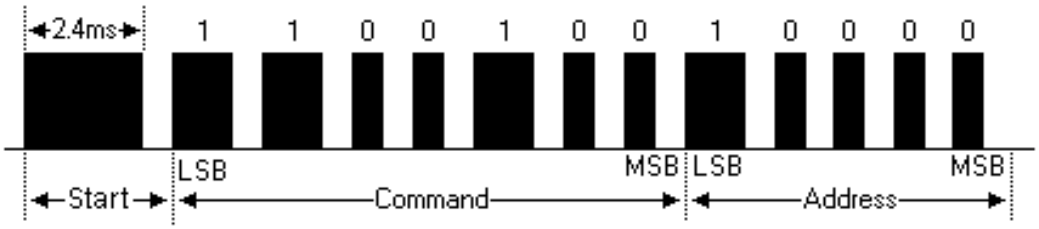
MODULATION

If you look at the image, you can see the 1.2ms high of the Logical '1' has further black lines with spaces in between. These correspond to the ON/OFF cycles. The space between these is what is called the frequency. The frequency of occurrence of an ON/OFF cycle is what it means.

The black bars in the below image correspond to high signals (called marks) and the white spaces in between correspond to low signals (called spaces). The duration of the 'marks' varies according to the bit being transmitted. It is 2.4ms for the start bit, 1.2ms for aHIGH bit and 0.6ms for aLOW bit. The duration of the 'spaces' is a constant 0.6ms. Every mark is followed by a space. Any data can be converted to binary format and transmitted in this manner. In fact, this is the basic form of all types of serial communication



PROTOCOL



LSB=least significant bit

MSB= Most significant bit

The picture above shows a typical pulse train of the SIRC protocol. With this protocol, the LSB is transmitted first. The start burst is always 2.4ms wide, followed by a standard space of 0.6ms. Apart from signaling the start of an SIRC message this start burst is also used to adjust the gain of the IR receiver. Then the 7-bit Command is transmitted, followed by the 5-bit Device address. In this case

Address, 1 and Command 19 are transmitted. Commands are repeated every 45ms (measured from start to start) for as long as the key on the remote control is held down.

Application

- Universal remote control
- Detector with Analog output
- Air conditioner remote
- Car controller

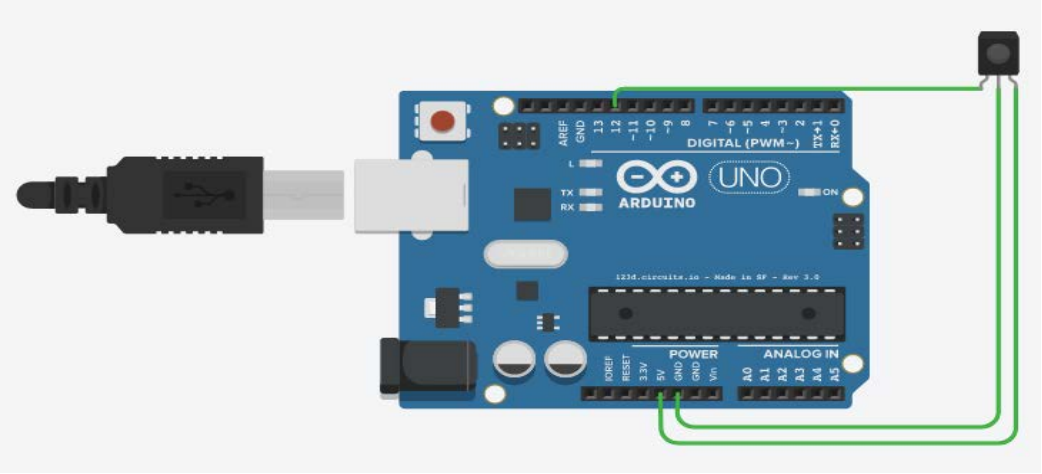
Activity

Write a program to receive values from IR remote using TSOP 1738.

HARDWARE REQUIRED:

- Arduino UNO board
- TSOP-1738 sensor
- Jumpers wires/cables

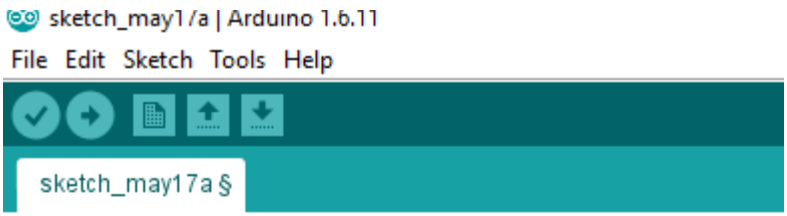
CIRCUIT CONNECTION



Code

```
#include <IRremote.h>
int RECV_PIN = 12;
IRrecv irrecv(RECV_PIN);
decode_results results;
int remote = 0;
```

```
void setup()
        {
                Serial.begin(9600);
                irrecv.enableIRIn(); // Start the receiver
        }

void loop()
        {
                If (irrecv.decode(&results))
                        {
                                remote = results.value;
                                Serial.println(remote);
                                irrecv.resume(); // Receive the next value
                        }
        }
```



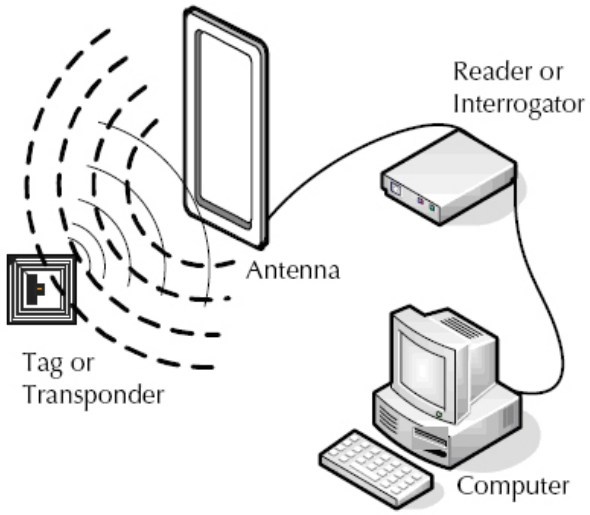

```
#include <IRremote.h>

int RECV_PIN = 12;

IRrecv irrecv(RECV_PIN);

decode_results results;

int remote = 0;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}

void loop() {
  if (irrecv.decode(&results)) {

    remote = results.value;
    Serial.println(remote);
    irrecv.resume(); // Receive the next value
  }
}
```

Chapter 3: Radio Frequency Identification (RFID)

INTRODUCTION TO RFID

RF technology is used in many different applications, such as television, radio, cellular phones, radar, and automatic identification systems. The term RFID (radio frequency identification) describes the use of radio frequency signals to provide automatic identification of items. RFID is used in applications such as:

- electronic toll collection (ETC)
- railway car identification and tracking
- the intermodal container identification
- asset identification and tracking
- item management for retail, health care, and logistics applications
- access control
- animal identification
- fuel dispensing loyalty programs
- automobile immobilizing (security)



Radio frequency (RF) refers to electromagnetic waves that have a wavelength suited for use in radio communication. Radio waves are classified by their frequencies, which are expressed in kilohertz, megahertz, or gigahertz. Radio frequencies range from very low frequency (VLF), which has a range of 10 to 30 kHz, to extremely high frequency (EHF), which has a range of 30 to 300 GHz.

Overview of RFID

ANTENNA

Each RFID system includes at least one antenna to transmit and receive the RF signals. In some systems, a single antenna transmits and receives the signals; in other systems, one antenna transmits and one antenna receives the signals. The quantity and type of antennas used, depends on the application.

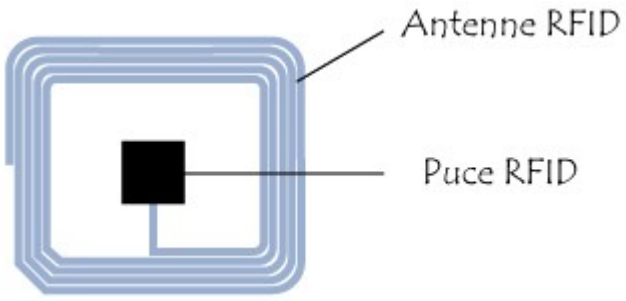
RF TRANSRECEIVER

The RF transceiver is the source of the RF energy used to activate and power the passive RFID tags. The RF transceiver may be enclosed in the same cabinet as the reader or it may be a separate piece of equipment. When provided as a separate piece of equipment, the transceiver is commonly referred to as an RF module. The RF transceiver controls and modulates the radio frequencies that the antenna transmits and receives. The transceiver filters and amplifies the backscatter signal from a passive RFID tag.

READER

The RFID reader directs the RF transceiver to transmit RF signals, receives the encoded signal from the tag through the RF transceiver, decodes the tag's identification, and transmits the identification with any other data from the tag to the host computer. The reader may also provide other functions. For example, ETC applications include accepting data from other input devices such as a vehicle detector and controlling gate and lights. Firmware in the reader controls reader operations. The user can change or customize the reader's operations to suit a specific requirement by issuing commands through the host computer or a local terminal.

RFID TAG



The RFID tag is made up of a microchip attached to an antenna. The tag picks up signals from and sends signals to a reader. The tag contains a unique serial number, but may also contain other information, such as a customer's account number. Tags have lots of different form factors depending on the

application. RFID tags can be active, passive or battery-assisted.

Middleware:

Middleware is software installed on a server connected to interrogators and enterprise applications. This is used to filter data and pass on useful information only to the enterprise applications. Some middleware can also be used to manage readers on a network.

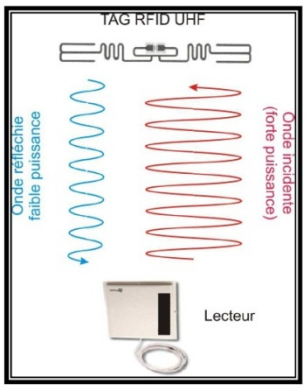
Air interface:

Air interface is the medium used to transfer data and/or energy by magnetic or electromagnetic waves.

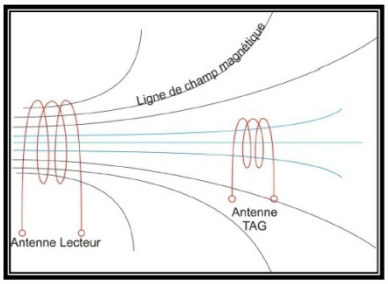
RFID TAG/INTERROGATOR COUPLING

Coupling between tag and interrogator can be either:

- Magnetic - in the case of near field communication (few cm up to 1m). Interrogator uses LF (Low frequencies) or HF (High Frequencies)



Couplage Electrique
UHF et SHF
Champ lointain



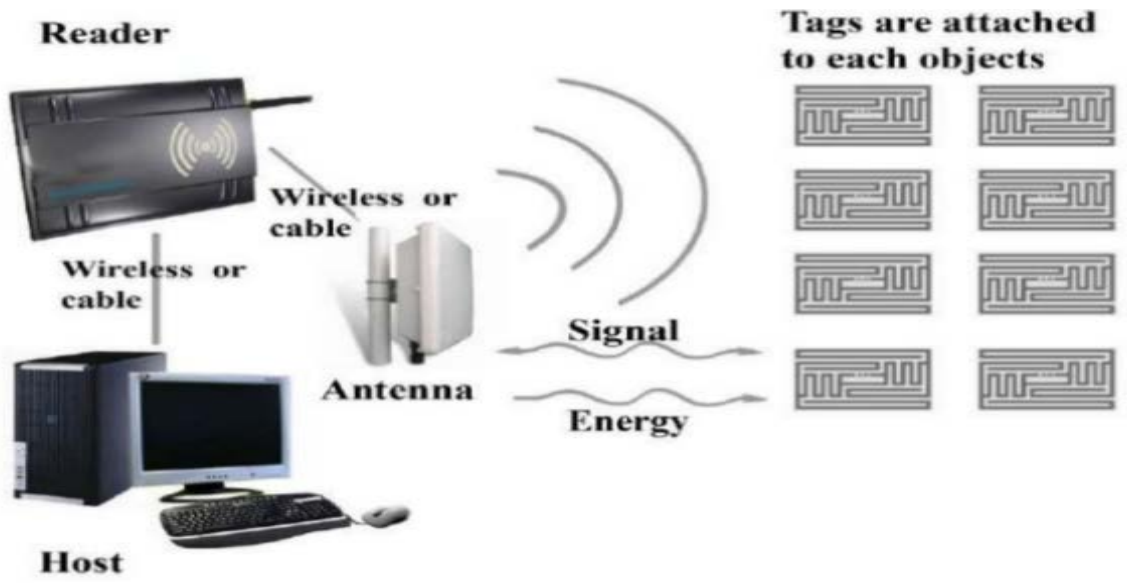
Couplage Magnétique
HF et LF
Champ proche

- Electromagnetic - in the case of far-field communication (up to 6m in theline of sight communication and European regulations). Interrogator uses UHF (Ultra High Frequencies) or SHF (Super High Frequencies)

WORKING

RFID belongs to a group of technologies referred to as Automatic Identification and Data Capture (AIDC). AIDC methods automatically identify objects, collect data about them, and enter those data directly into computer systems with little or no human intervention. FID methods utilize radio waves to accomplish this. At a simple level, RFID systems consist of three components: a RFID tag or smart label, a RFID reader, and an antenna. RFID tags contain an integrated circuit and an antenna, which are used to transmit data to the RFID reader (also called an interrogator). The reader then converts the radio waves to a more usable form of data. Information collected from the tags is then

transferred through a communications interface to a host computer system, where the data can be stored in a database and analyzed at a later time.



TYPES OF RFID TAGS

- ACTIVE TAGS
- PASSIVE TAGS

Passive RFID Tags

Passive tags are comprised of three elements: an integrated circuit or chip, an antenna, and a substrate.

The RFID chip stores data and perform specific tasks. Depending on its design, the chip may be read-only (RO), Write-once, readmany (WORM), or read-write (RW). Typically, RFID chips carry 96 bits of memory but can range from 2-1000 bits.

Attached to the chip is the antenna, whose purpose is to absorb radio-frequency (RF) waves from the reader's signal and to send and receive data. Passive RFID tag performance is strongly dependent on the antenna's size: the larger the antenna, the more energy it can collect and then send back out. Larger antennas, therefore, have higher read ranges (although not as high as those of active tags).

Antenna shape is also important to the performance of the tag. Low- and high-frequency (LF and HF, respectively) antennas are usually coils because these frequencies are predominantly magnetic in nature. Ultrahighfrequency (UHF) antennas, on the other hand, look similar to old-fashioned TV antennas ("rabbit ears") because ultrahigh frequencies are solely electric in nature.

The third component of a passive RFID tag is called a substrate, which is commonly a Mylar or plastic film. Both the antenna and the chip are attached to the substrate, which may be thought of as the "glue" that holds all of the tag's pieces together.

In contrast to active RFID tags, passive tags are usually smaller and less expensive.

Active RFID Tags

Like passive RFID tags, active tags have both a microchip and an antenna. The chips, however, are usually larger in size and have greater capabilities than the RFID chips in passive tags.

Active tags have two additional components that differentiate them from passive tags: an on-board power supply and onboard electronics.

The power supply is usually a battery, although it can also be solar. The built-in power supply allows the tag to transmit data to a reader on its own, without the need to draw power from the reader itself like passive tags do. In addition, active tags can be read from distances of 100 feet or more, whereas passive tags can only be read from up to about 20 feet.

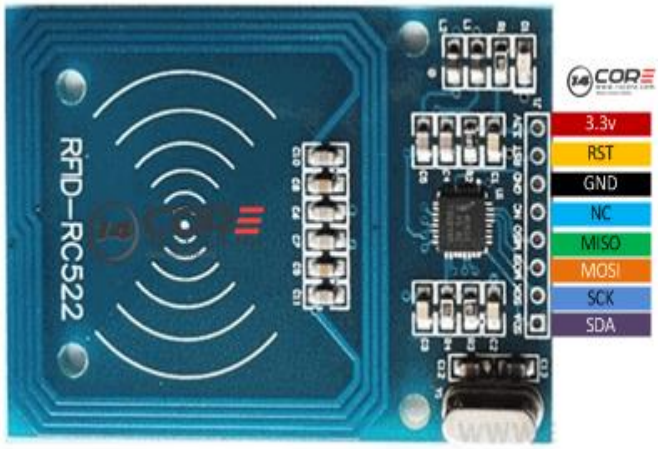
Onboard electronics may consist of sensors, microprocessors, and input/output ports, all of which are powered by the tag's onboard power source. The electronics allow active RFID tags to be used in a wider range of applications than passive tags. For example, perishable food products may be tagged with sensors that collect data that can then be used to determine expiration dates and warn the end user that the item may be spoiled. Even though many products have expiration dates printed on them, these dates are valid only if the product is stored under the optimal conditions (temperature, humidity, exposure to light, etc.) for that type of product. Thus, the product may expire before the printed date if it is not stored properly. A RFID tag equipped with a temperature sensor might be able to predict the actual expiration date of a carton of milk, for example, which may be very different from the printed date.

Whether you choose to use active or passive tags in your RFID system will likely depend on both your particular application and your budget. Simple asset tracking that utilizes barcode technology will become obsolete as RFID proliferates through organizations, making them more efficient and better equipped for accuracy.

APPLICATION

- People Tracking
- Documents tracking
- Metro gate
- Companies
- Tolls
- Schools and college used for attendance
- Industry
- Management

Pin Description



RFID MODULE	ARDUINO
3.3v	Arduino Pin 3.3v
RST (Reset)	Arduino Pin 9
GND (Ground)	Arduino Pin GND
NC	No Connection
MISO	Arduino Pin 13
MOSI	Arduino Pin 11
SCK	Arduino Pin 13
SDA	Arduino Pin 10

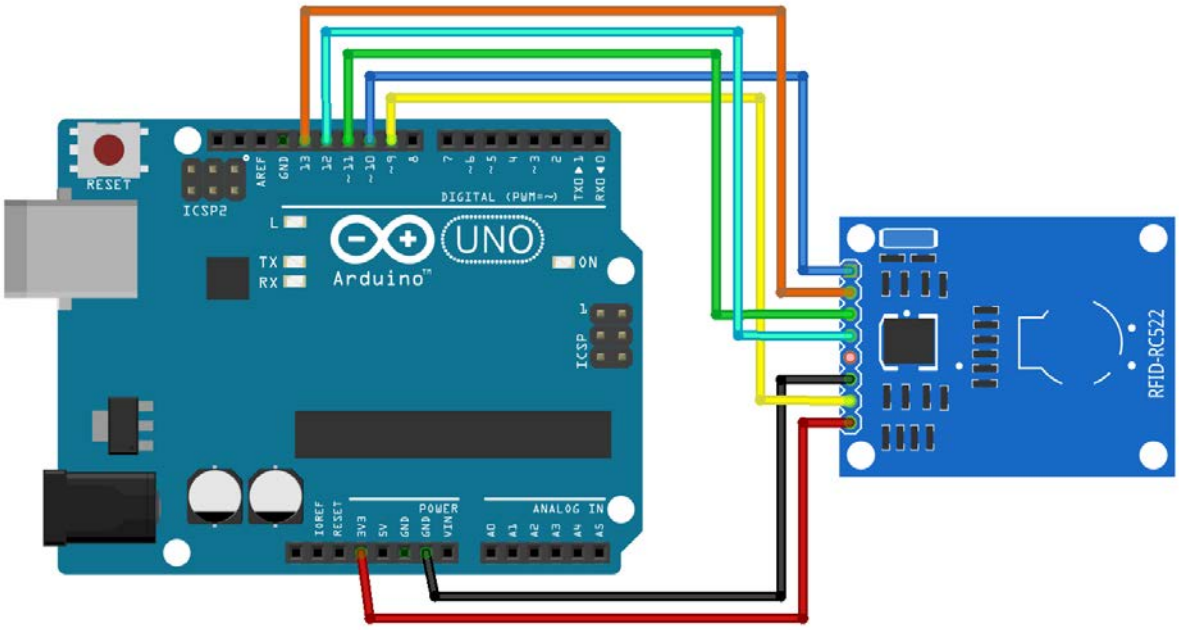
Activity

Write a program to get values of RFID tag using RFID receiver module MFRC-522

HARDWARE REQUIRED:

- Arduino UNO compatible board
- RFID receiver and tags
- Jumpers wires/cables

CIRCUIT CONNECTION



Code

```
#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);
void setup()
        {
                Serial.begin(9600);
                SPI.begin();
                mfrc522.PCD_Init();
                Serial.println("Scan PICC to see UID and type...");
        }

void loop()
        {
        // Look for new cards
```

```
if ( ! mfrc522.PICC_IsNewCardPresent())
        {
                return;
        }


If ( ! mfrc522.PICC_ReadCardSerial())
        {
                return;
        }
mfrc522.PICC_DumpToSerial(&(mfrc522.uid));}
```

sketch_may18a | Arduino 1.6.11

File Edit Sketch Tools Help

sketch_may18a §

```
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);  // Create MFRC522 instance.

void setup() {
  Serial.begin(9600); // Initialize serial communications with the PC
  SPI.begin();      // Init SPI bus
  mfrc522.PCD_Init(); // Init MFRC522 card
  Serial.println("Scan PICC to see UID and type...");
}

void loop() {
  // Look for new cards
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
  }
  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
  }
  // Dump debug info about the card. PICC_HaltA() is automatically called.
  mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
}
```

21

CHAPTER 4: TRIPLE AXIS MAGNETOMETER HMC5883L

INTRODUCTION

A magnetometer is an instrument with a sensor that measures magnetic flux density B (in units of Tesla or As/ m2).

Since magnetic flux density in air is directly proportional to magnetic field strength, a magnetometer is capable of detecting fluctuations in the Earth's field.

Materials that distort magnetic flux lines are known as magnetic, and include materials such as magnetite that possess magnetic fields of their own, as well as very high magnetic conductivity. Such materials create distortions in the Earth's magnetic flux that is flowing around them. Magnetometers detect these distortions.

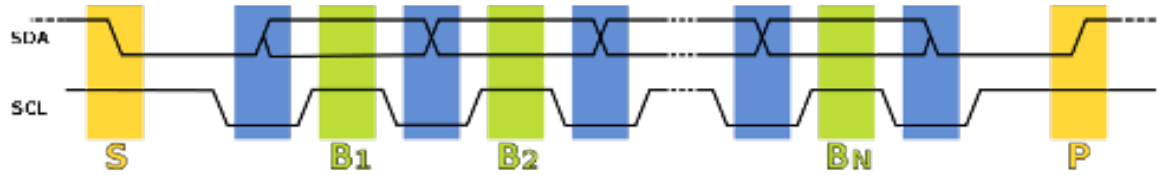
This is a breakout Board HMC5883L, a 3-axis magnetometer. Magnetometers have a wide range of uses. The most common include using the chip as a digital compass to sense direction or using them to detect ferrous (magnetic) metals.

Working

Magnetic fields and current go hand-in-hand. When current flows through a wire, a magnetic field is created. This is the basic principle behind electromagnets. This is also the principle used to measure magnetic fields with a magnetometer. The direction of Earth's magnetic fields affects the flow of electrons in the sensor, and those changes in current can be measured and calculated to derive a compass

FIRMWARE

The HMC5883L speaks a kind of funky serial language called **I^2C**. I^2C requires just two wires for communication – one for a clock (**SCL**), and one for data (**SDA**). Be aware that you probably should avoid using those two pins for anything else.



The Arduino code example we have, HMC5883.pde, will continuously reads data from the magnetometer for each of the magnetometers axes using I^2C. If you need help uploading the sketch to your Arduino, their site has lots of helpful information.

DESCRIPTION

We based this breakout on a popular and well-loved magnetometer, the HMC5883L. This compact sensor uses I2C to communicate and it's very easy to use. Since it's a 3.3V max chip, we added circuitry to make it 5V-safe logic and power, for easy use with either 3 or 5V microcontrollers. Simply connect VCC to +3-5V and ground to ground.Simply download our library and connect the SCL pin to your Arduino's I2C clock pin, and SDA pin to your Arduino's I2C data pin and upload our test example sketch to read out magnetic field data and heading (i.e. which way is north)

SPECIFICATIONS

- I2C interface
- 1-2 degree heading accuracy
- Integrated 12-bit ADC
- 160Hz max data rate
- Range of -8 to +8 Gauss
- APPLICATIONS
- Mobiles
- Notebooks.
- Compass
- Magnetic field detector (only for ferrous material)
- Use for Hall effect sensor

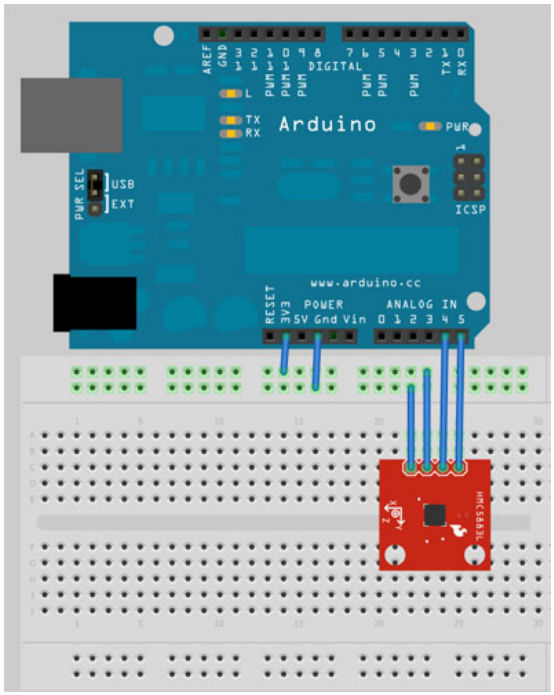
Activity

Write a program to display values of HMC5883L on Serial monitor

HARDWARE REQUIRED:

- Arduino UNO compatible board
- Triple Axis Magnetometer Breakout - HMC5883L
- Jumpers wires / cables

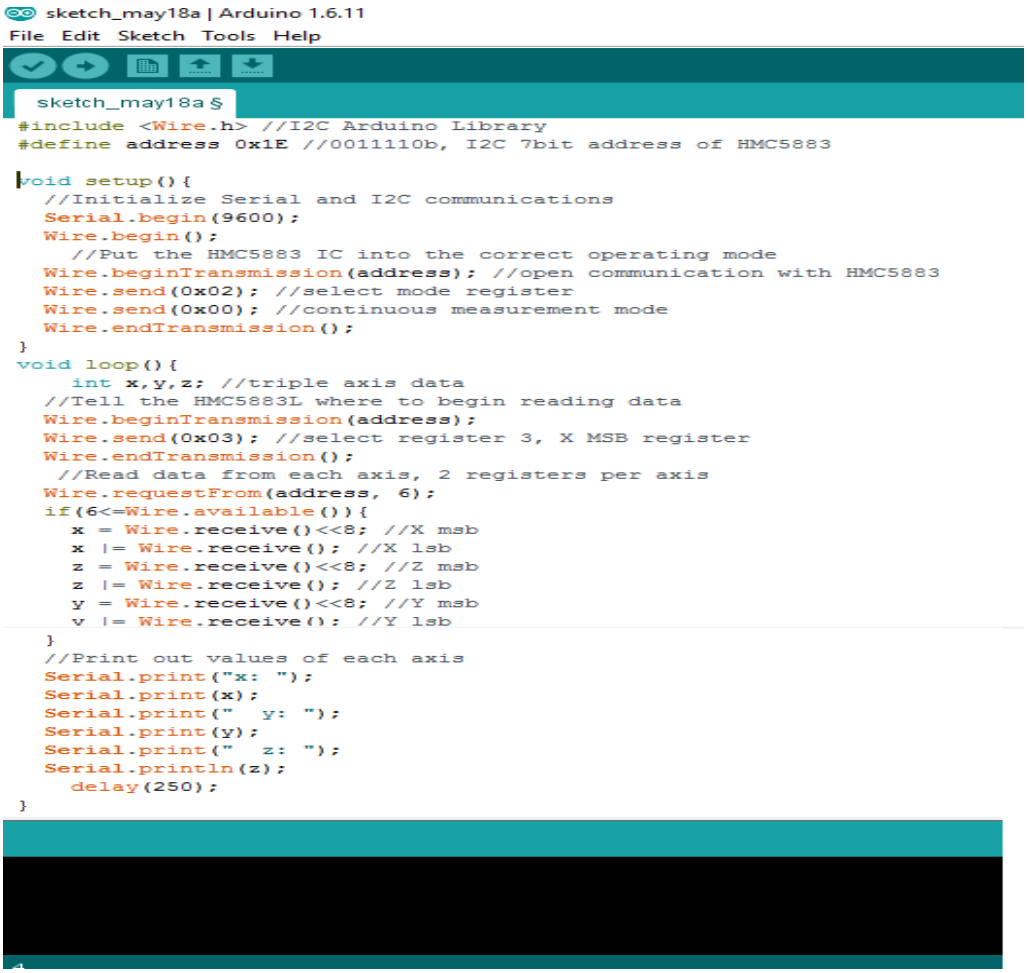
CIRCUIT CONNECTION



CODE

```
#include <Wire.h> //I2C Arduino Library
#define address 0x1E

void setup()
	{
		//Initialize Serial and I2C communications
		Serial.begin(9600);
		Wire.begin();
		Wire.beginTransmission(address);
		Wire.send(0x02);
		Wire.send(0x00);
		Wire.endTransmission();
	}
```

```
void loop()
	{
		int x,y,z;
		Wire.beginTransmission(address);
		Wire.send(0x03);
		Wire.endTransmission();
		Wire.requestFrom(address, 6);
		if(6<=Wire.available()){
			x = Wire.receive()<<8;
			x |= Wire.receive();
			z = Wire.receive()<<8;
			z |= Wire.receive();
			y = Wire.receive()<<8;
			y |= Wire.receive();
		}

		Serial.print("x: ");
		Serial.print(x);
		Serial.print("  y: ");
		Serial.print(y);
		Serial.print("  z: ");
		Serial.println(z);
		delay(250);
	}
```

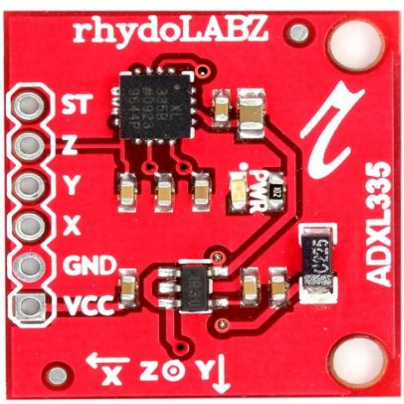


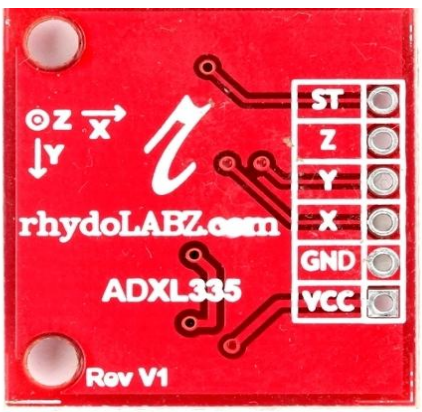
CHAPTER 5: TRIPLE AXIS ACCELEROMETER (ADXL345)

INTRODUCTION

An accelerometer is an electro-mechanical device that will measure acceleration forces. These forces may be static, like the constant force of gravity pulling at your feet, or they could be dynamic – caused by moving or vibrating the accelerometer.

The ADXL345 is a small, thin, low power, 3-axis accelerometer with high resolution (13-bit) measurement at up to ±16 g. Digital output data is





formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I2 C digital interface. The ADXL345 is well suited for mobile device applications. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (4 mg/LSB) enables measurement of inclination changes less than 1.0°. Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion and if the acceleration on any axis exceeds a user-set level. Tap sensing detects single and double taps. Free-fall sensing detects if the device is falling. These functions can be mapped to one of two interrupt output pins. An integrated, patent pending 32-level first in, first out (FIFO) buffer can be used to store data to minimize host processor intervention. Low power modes enable intelligent motion-based power management with threshold sensing and active acceleration measurement at extremely low power dissipation. The ADXL345 is supplied in a small, thin, 3 mm × 5 mm × 1 mm, 14-lead, plastic package.

FEATURES

- Ultralow power: as low as 23µA in mode
- Fixed 10-bit resolution Full resolution, up to 13-bit resolution at ±16 g
- Single tap/double tap detection
- Activity/inactivity monitoring
- Free-fall detection
- Supply voltage range: 2.0 V to 3.6 V
- I/O voltage range: 1.7 V to VS
- SPI (3- and 4-wire) and I2C digital interfaces

Working

Measurement of acceleration can help us in doing some really interesting projects. In embedded systems, this task is performed by electronic devices called accelerometers. An accelerometer does not only give you information about how faster an object is moving but serve as an indicator for the occurrence of natural events like a fall or a tilt.

ADXL335 is a triple axis accelerometer. Tipple axis in the sense that it can measure acceleration along three axes viz x, y and z. The measured values appear as change in voltage at three output pins with respect to a common ground. The sensor measures acceleration with the help of a layer of polysilicon suspended above silicon wafer with the help of polysilicon springs. The motion of this mass is translated into the motion of the plates of a differential capacitor and thereby providing an output proportional to acceleration.

When a small voltage of less than 3.6V is applied on the pin output of the accelerometer is affected due to the generation of electrostatic force.

PIN DESCRIPTION

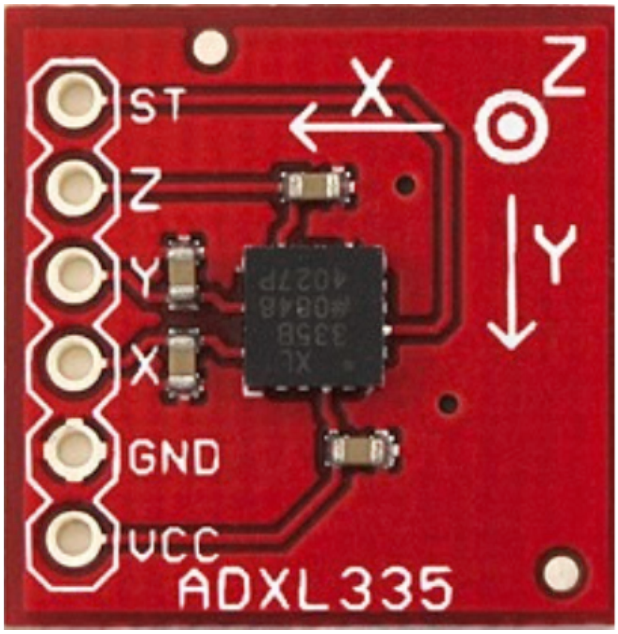
Z-axis acceleration signal

Y-axis acceleration signal

X-axis acceleration signal

Ground

Voltage Supply



Application

- Handsets
- Medical instrumentation
- Gaming and pointing devices
- Industrial instrumentation
- Personal navigation devices
- Fitness equipment

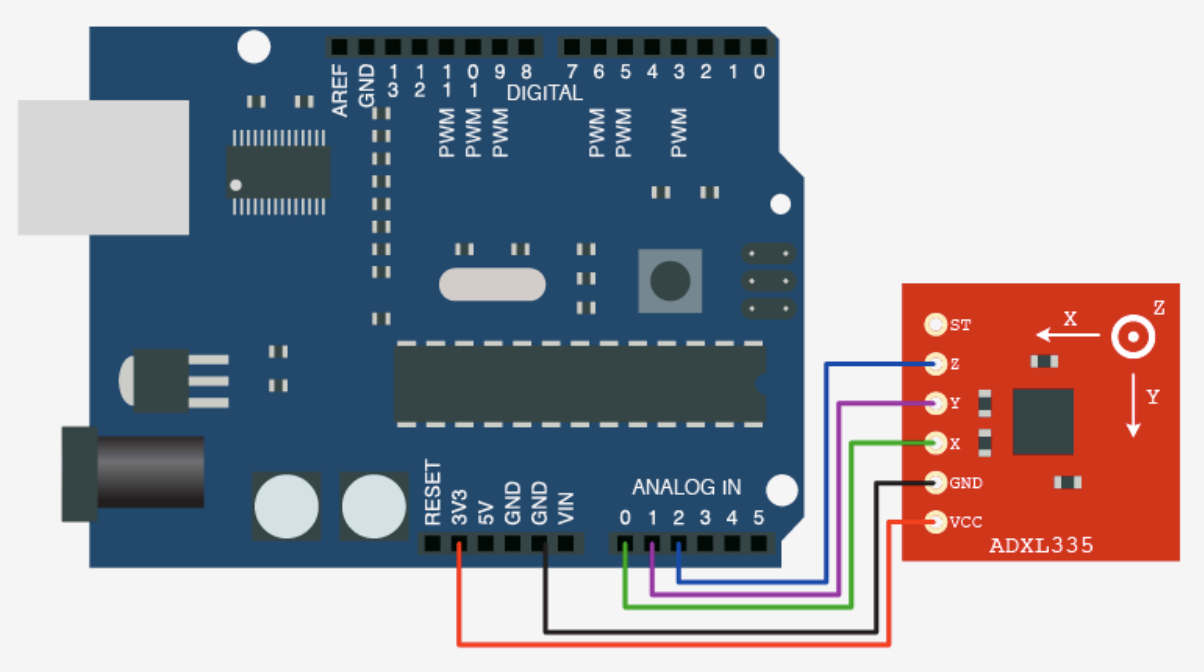
Activity

Write a program to display values from ADXL335 on serial monitor.

HARDWARE REQUIRED

- Arduino
- ADXL335
- Connecting wires (Make sure they are not too short)

CIRCUIT CONNECTION



CODE

```
const int xout = A0;
const int yout = A1;
const int zout = A2;
int out1 = 0;
int sout1 = 0;
int out2 = 0;
int sout2= 0;
int out3 = 0;

int sout3= 0;
void setup() {
                    Serial.begin(9600);
          }
void loop() {
                    analogReference(EXTERNAL);
                    out1 = analogRead(xout);
                    sout1 = map(out1, 0, 1023, 0, 255);
                    delay(2);
                    out2 = analogRead(yout);
                    sout2 = map(out2, 0, 1023, 0, 255);
                    delay(2);
                    out3 = analogRead(zout);
                    sout3 = map(out3, 0, 1023, 0, 255);
                    Serial.print("X = " );
                    Serial.print(out1);
```

```
        Serial.print("\t output1 = ");
        Serial.println(sout1);
        Serial.print("Y = " );
        Serial.print(out2);
        Serial.print("\t output2 = ");
        Serial.println(sout2);
        Serial.print("Y = " );
        Serial.print(out3);
        Serial.print("\t output3 = ");
        Serial.println(sout3);
        delay(3000);
}
```

CHAPTER 6: BUILDING A CHASSIS

Material required

Parts:

Motors

Chassis

Tools:

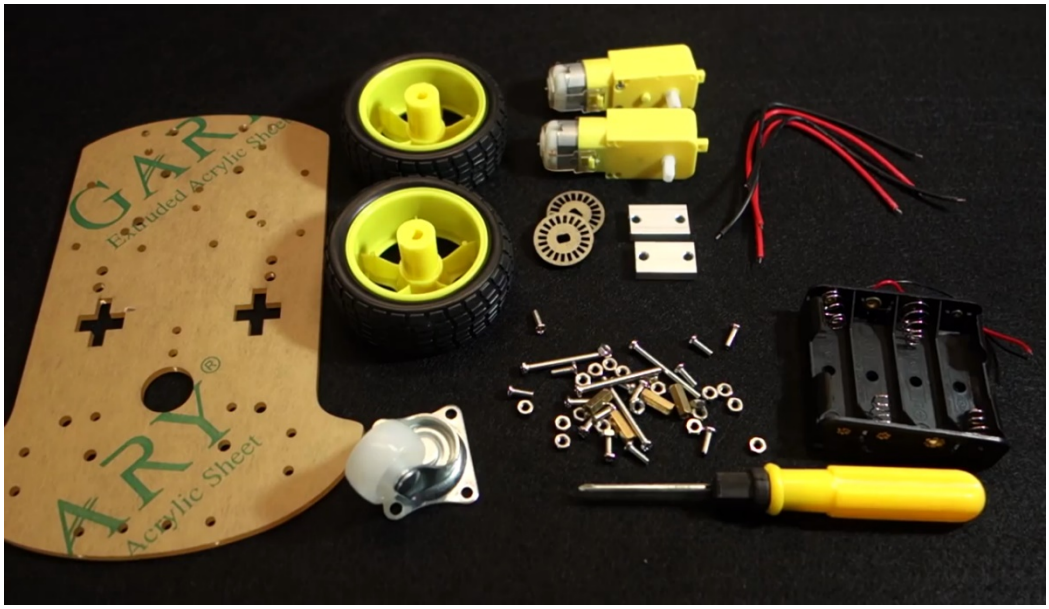
Screws

Screw Driver

Assembly Instructions

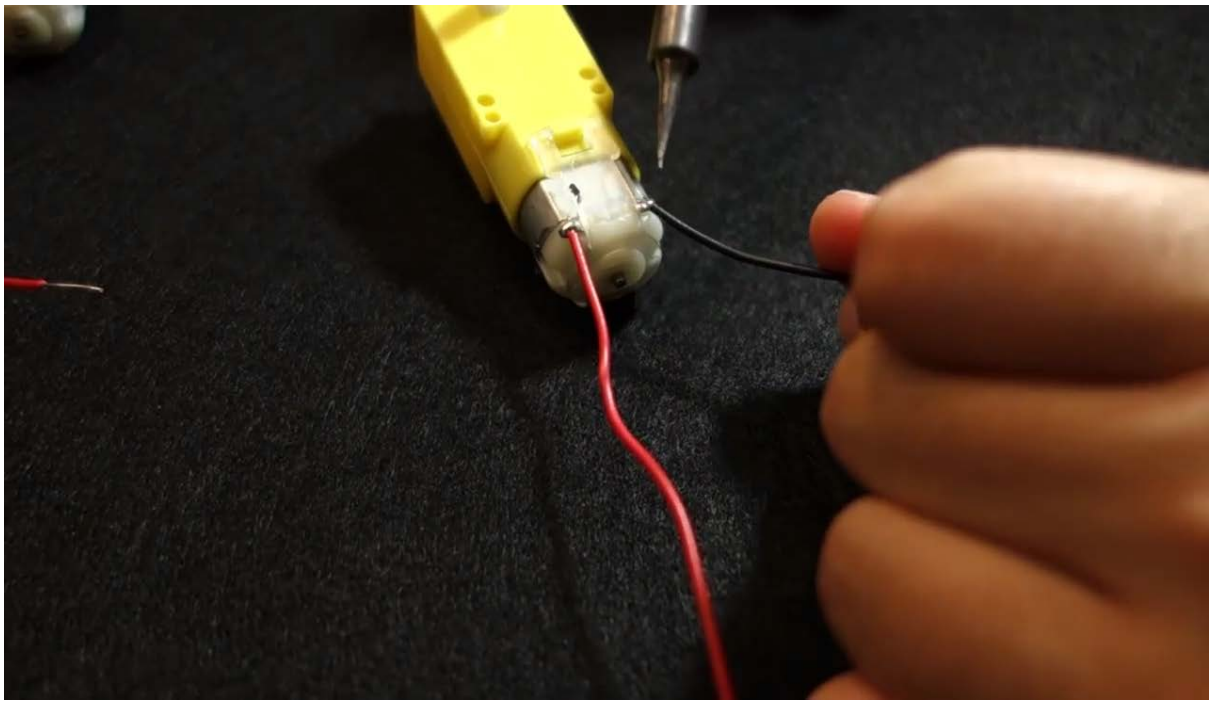
Step 1:

Assemble all the parts



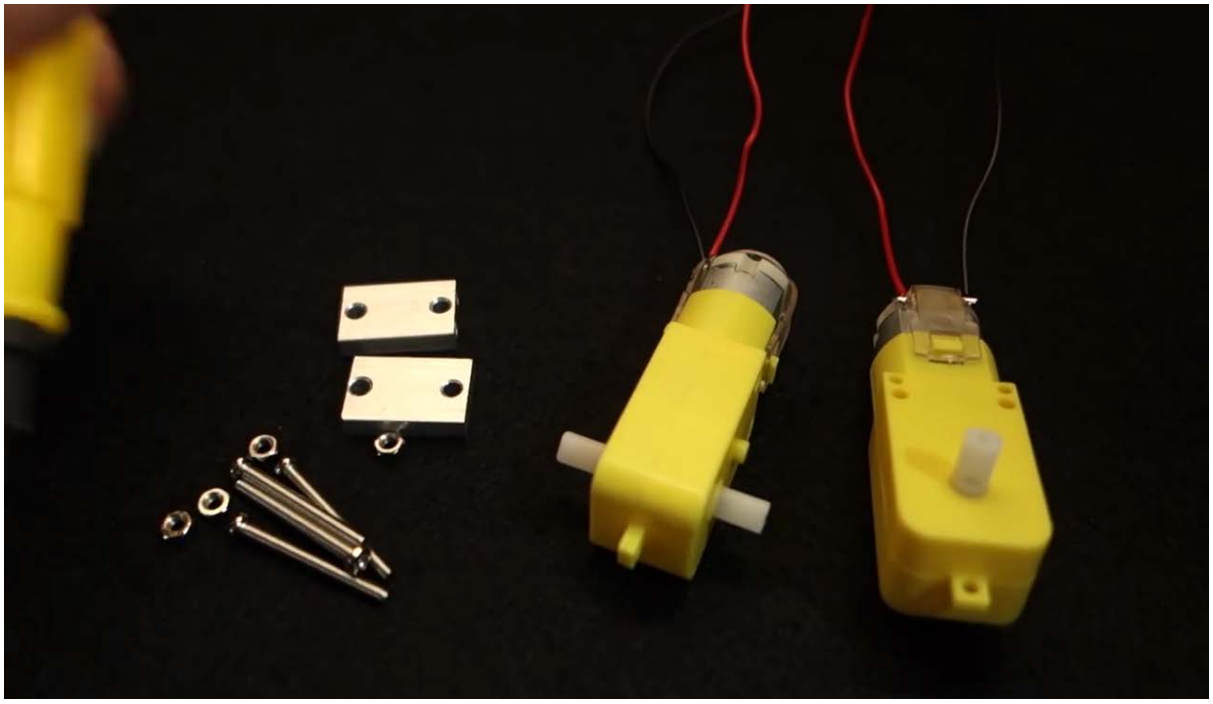
Step 2:

Soldering the cables



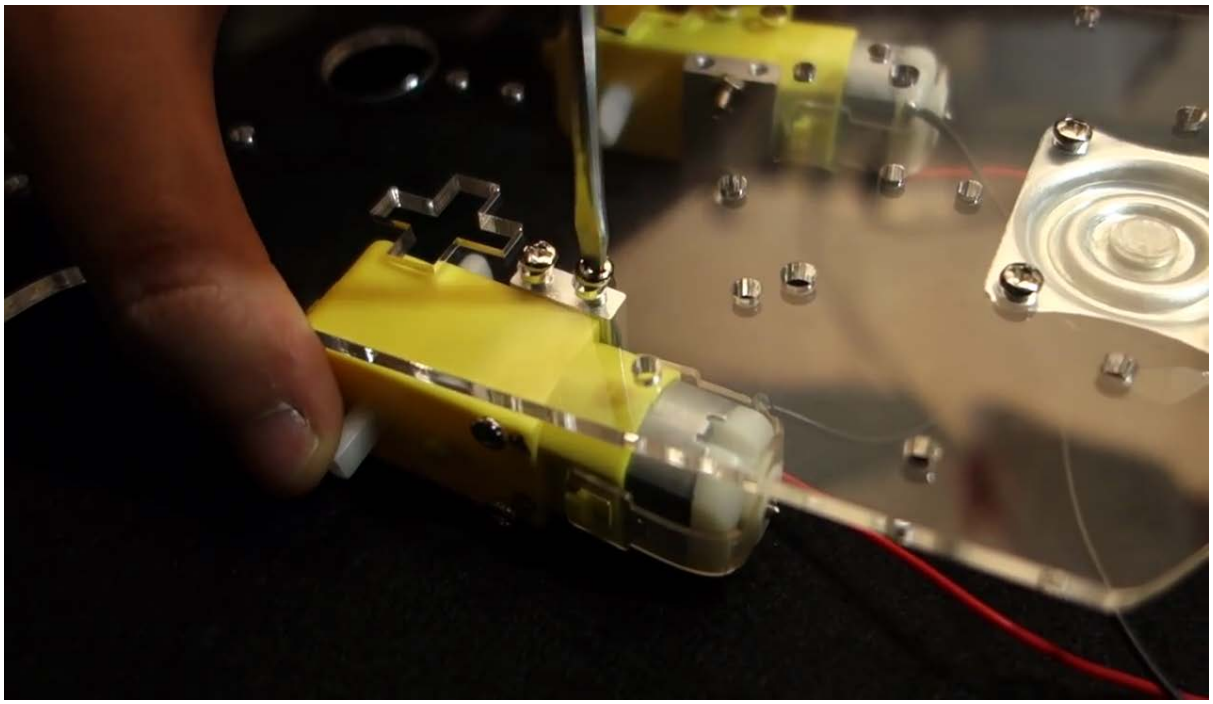
Step 3:

Collect the parts of motor



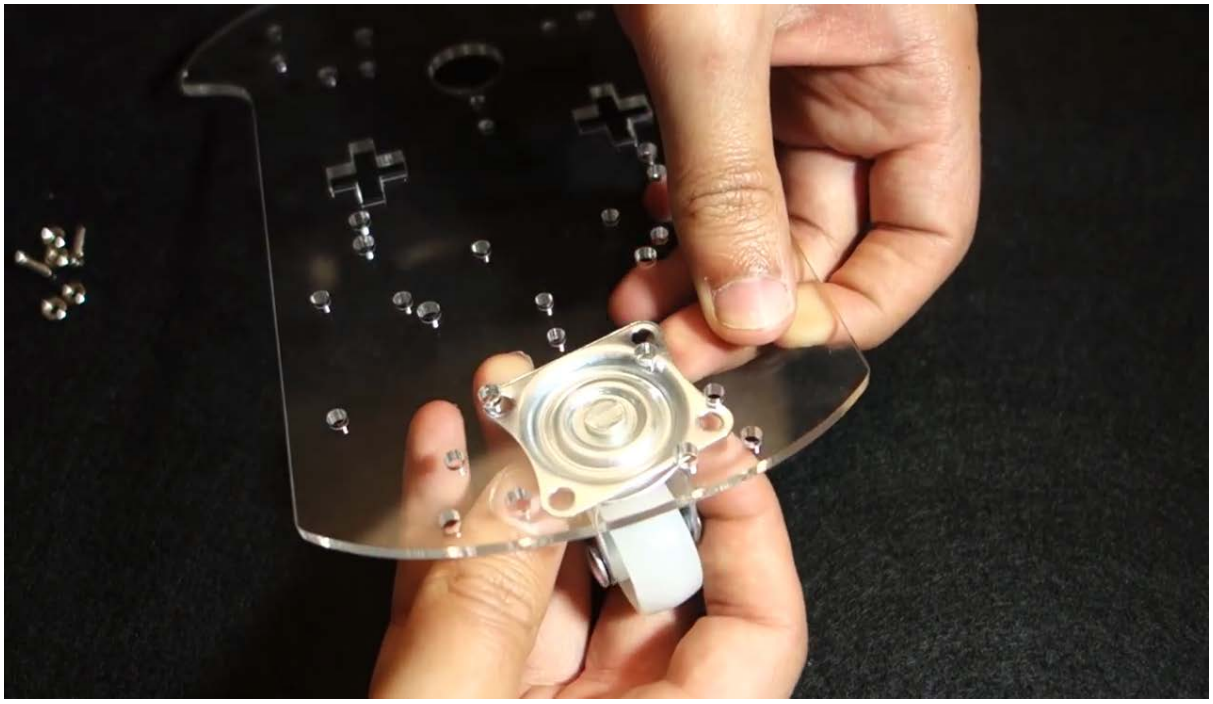
Step 4:

Mount the motor on chassis



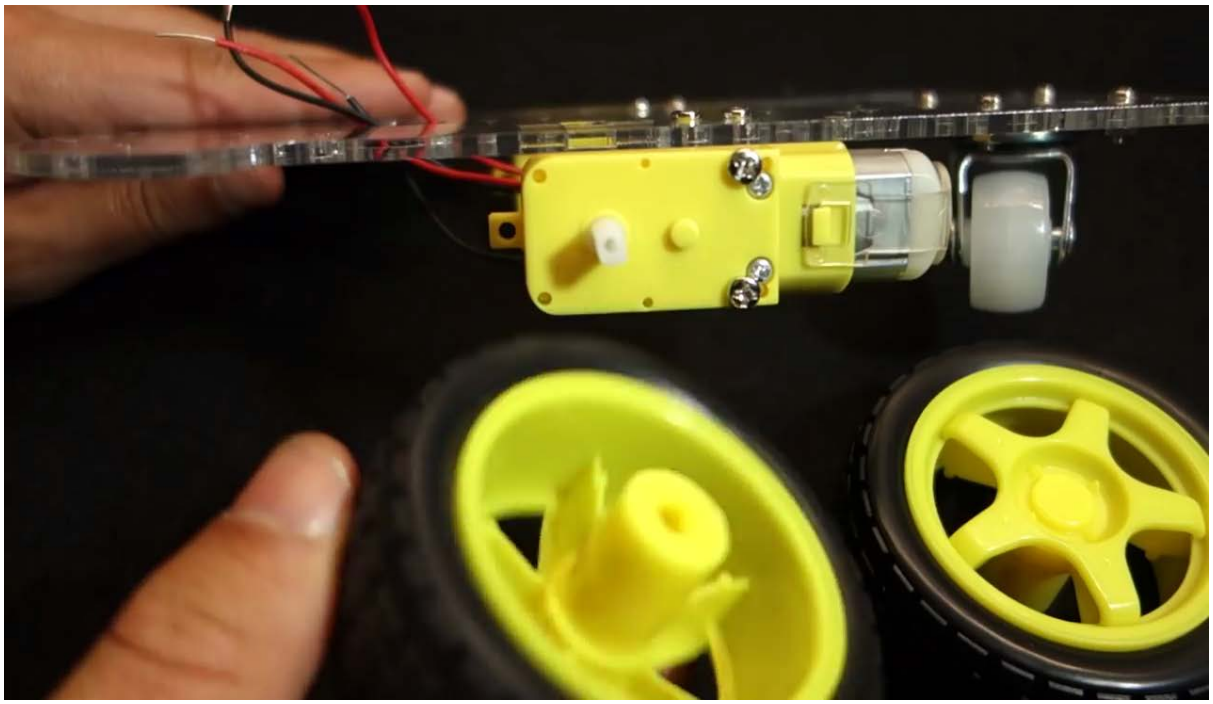
Step 5:

Place the caster wheel



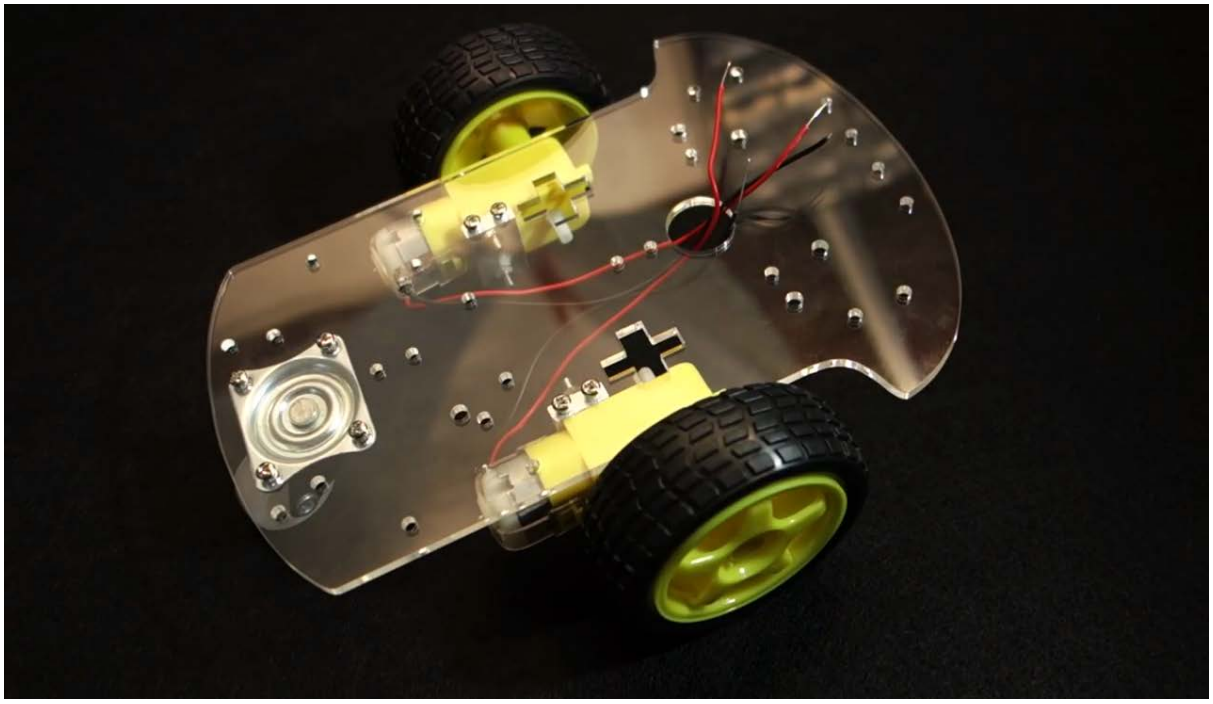
Step 6:

Place wheels on motors



Step 7:

Chassis is done

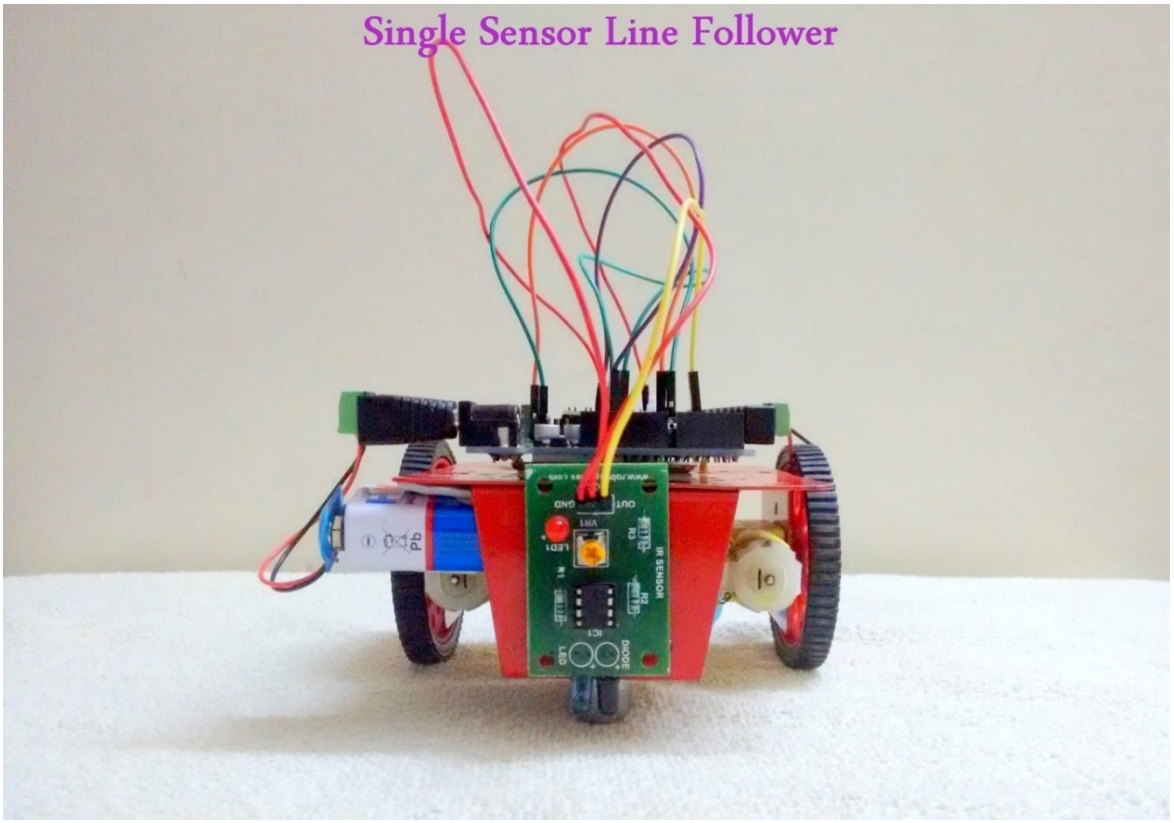


CHAPTER 7: LINE FOLLOWER

Introduction

Line follower is an autonomous robotwhich follows a line, either a black line or white line. Basically. there are two types of line follower robots:

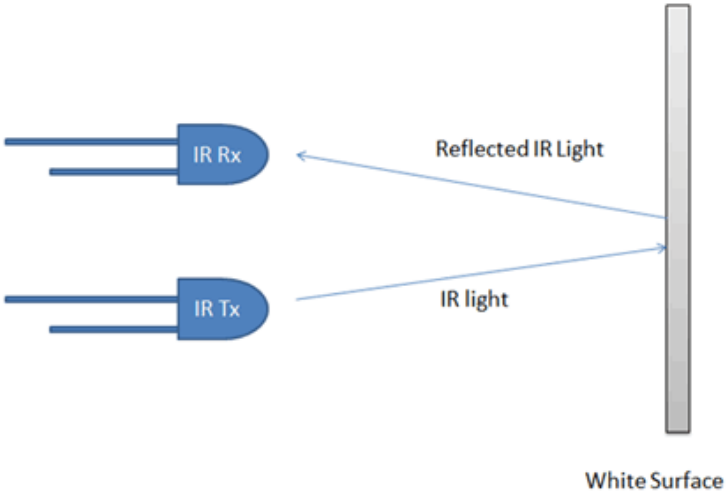
1. Black line follower which follows black line
2. White line follower which follows white line.



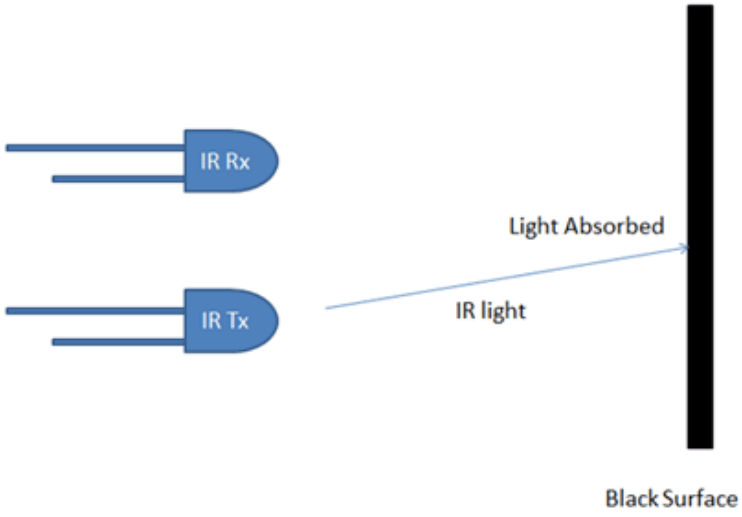
Line follower senses the line and run over it.

Concepts of Line Follower

Concept of working of line follower is related to light. We here the behavior of light at black and white surface. When light fall on a white surface it is almost full reflected and in case of



black surface light is completely absorbed. This behavior of light is used in **building a line follower robot**.



Arduino based line follower robot we have used IR (InfraRed) Transmitters and receivers also called photo diodes. They are used for sendingand receiving light. IR transmits infrared lights. When IR rays falls on white surface, it's reflect and receive by photodiodes which generates some voltage changes. When IR light falls on a black surface, light is absorbed by the black surface and no rays are reflected, thus photo diode does not receive any light or rays.

In line follower robot when sensor senses white surface then Arduino gets 1 as input and when senses black line Arduino gets 0 as input.

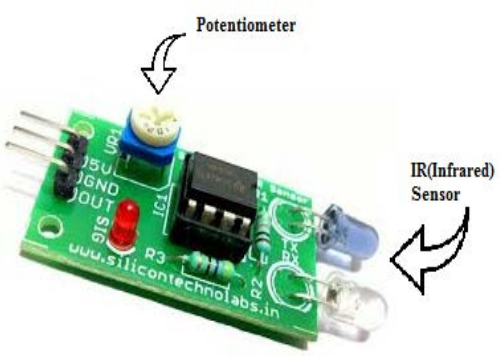
Circuit Explanation

The whole line follower robot can be divided into 3 parts:

1. Sensor
2. Controller
3. Driver shield

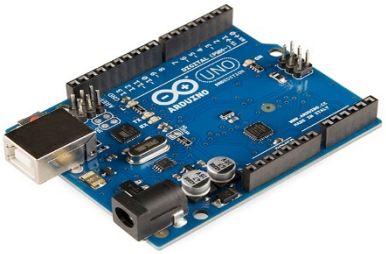
Sensor:

This part contains IR diodes, potentiometer.



- Potentiometer is used for setting reference voltage at comparator's one terminal
- IR sensors are used to sense the line and provide a change in voltage at comparator's second terminal.

Controller

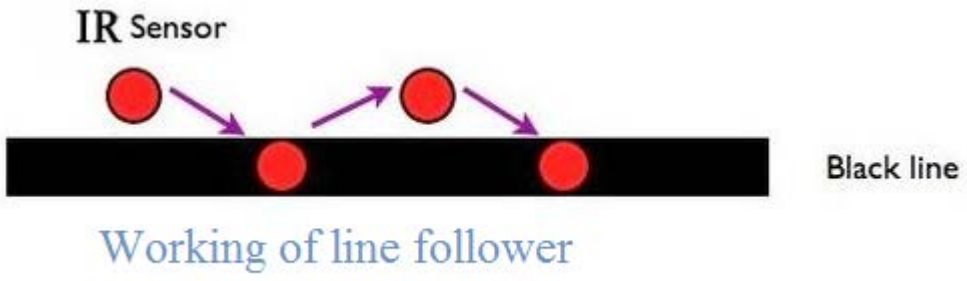


Arduino is used for controlling whole the process of line follower robot. Arduino read these signals and send commands to driver circuit to drive line follower.

Driver shield

Driver consists motor driver and two DC motors. Motor driver is used for driving motors because Arduino does not supply enough voltage and current to motor. So, we add a motor driver circuit to get enough voltage and current for motor. Arduino sends commands to this motor driver and then it drive motors.

Working of Line Follower Robot using Arduino



Working of line follower

Working is as follows.

1. When the IR sensor sees the black line one wheel of the bot rotates and second wheel stop and the bot turns away from the line.
2. When the IR sensors sees the white background the other wheel rotates and first wheel stop and the bot turns slightly towards the line.

The above two steps occur one after other in a repeat mode for very short time which gives us the movement of the bot in the path following the black line.

It just follows the Zig-Zag pattern turning towards the line and turning away from the line.

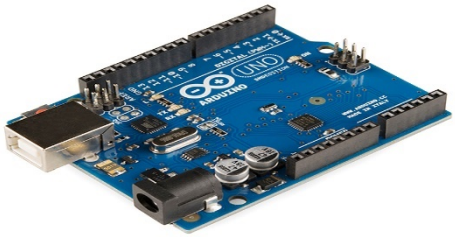
There are two conditions for line following

Sensor	Motor1	Motor 2
White line	Forward	Stop
Black line	Stop	Forward

Required Components

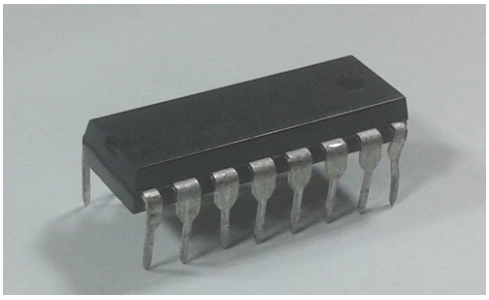
Arduino

In this we have used a microcontroller to control whole the process of system that is ARDUINO.



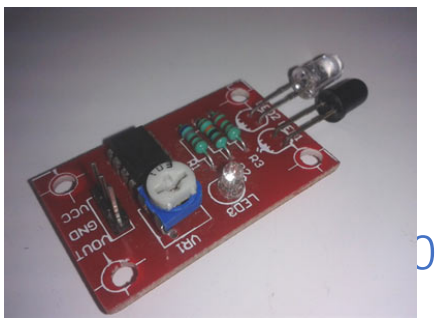
L293D Motor Driver

L293D is a motor driver IC which has two channels for driving two motors. L293D has two inbuilt Transistor Darlington pair for current amplification and a separate power supply pin for giving external supply to motors.



IR Module

IR Module is sensor circuit which consists IR LED/photodiode pair, potentiometer, LM358, resistors



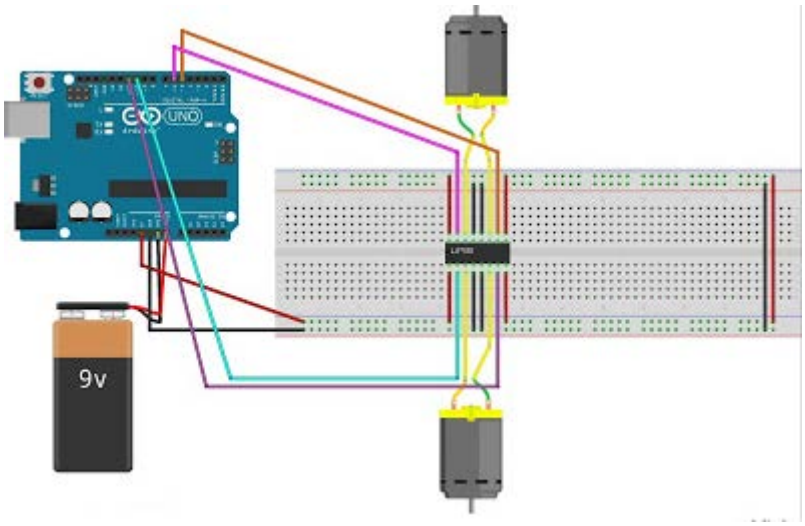
and LED. IR sensor transmits Infrared light and photo diode receives the infrared light.

Power Supply

A 5 to 9 Volt battery is used to power for Arduino and motor driver.

Circuit Diagram

Connect the L293D motor driver with the Arduino and dc motors.



Note: if the robot doesn't turn in the side of curves properly swap the motor pins connections and try it out.

CONNECTIONS:

3,6(l293d) to left motor(output)

11,14(l293d) to right motor(output)

2,7,10,15(l293d) to pins 10,11,5,6 of Arduino(inputs)

Connect your Arduino to your IR sensor.

1. Connect out pin of IR to pin 2 of Arduino
2. Connect vcc of IR to 5v of Arduino
3. Connect gnd of IR to gnd pin of Arduino.

CODE

```
int lmotorpin1=5;

int lmotorpin2=6;
int rmotorpin1=10;
int rmotorpin2=11;
int sensor=2;
int sensorstate=0;

void setup(){
  pinMode(sensor,INPUT);
  pinMode(lmotorpin1,OUTPUT);
  pinMode(lmotorpin2,OUTPUT);
  pinMode(rmotorpin1,OUTPUT);
  pinMode(rmotorpin2,OUTPUT);
}

void loop(){
  sensorstate=digitalRead(sensor);
  if(sensorstate==HIGH){
    digitalWrite(lmotorpin1,HIGH);
    digitalWrite(lmotorpin2,LOW);
    digitalWrite(rmotorpin1,LOW);
    digitalWrite(rmotorpin2,LOW);
  }
  else{
    digitalWrite(rmotorpin1,LOW);
    digitalWrite(rmotorpin2,HIGH);
    digitalWrite(lmotorpin1,LOW);
    digitalWrite(lmotorpin2,LOW);
  }
}
```

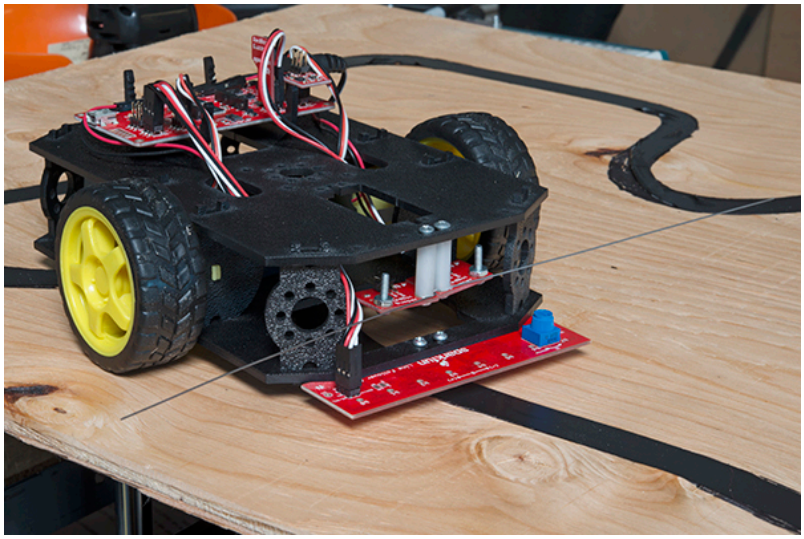
CHAPTER 8: ADVANCED LINE FOLLOWER

Introduction

Line follower is an autonomous robotwhich follows a line, either a black line or white line. Basically. there are two types of line follower robots:

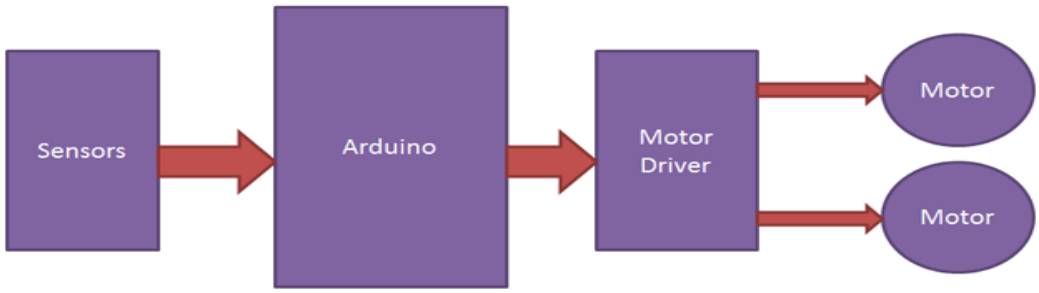
1. Black line follower which follows black line
2. White line follower which follows white line.

Line follower senses the line and run over it.

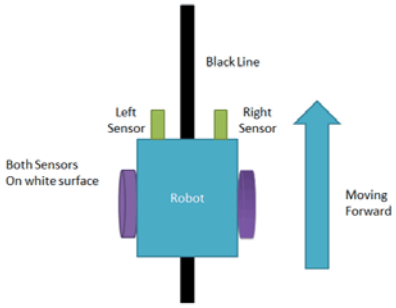


Working of Line Follower Robot using Arduino

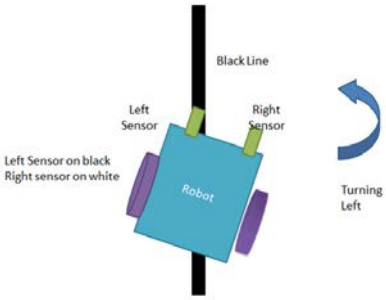
Line follower robot senses the line by using sensor and then sends the signal to Arduino. Then Arduino drives the motor according to sensors' output.



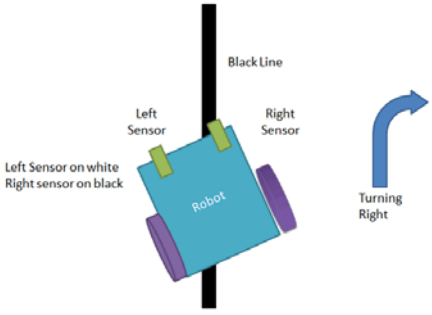
Here in this project we are using two IR sensor modules namely left sensor and right sensor.



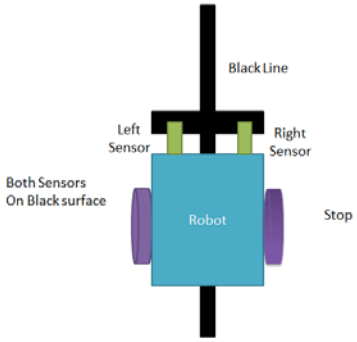
When both left and right sensor senses white then robot move forward.



If left sensor comes on black line then robot turn left side.

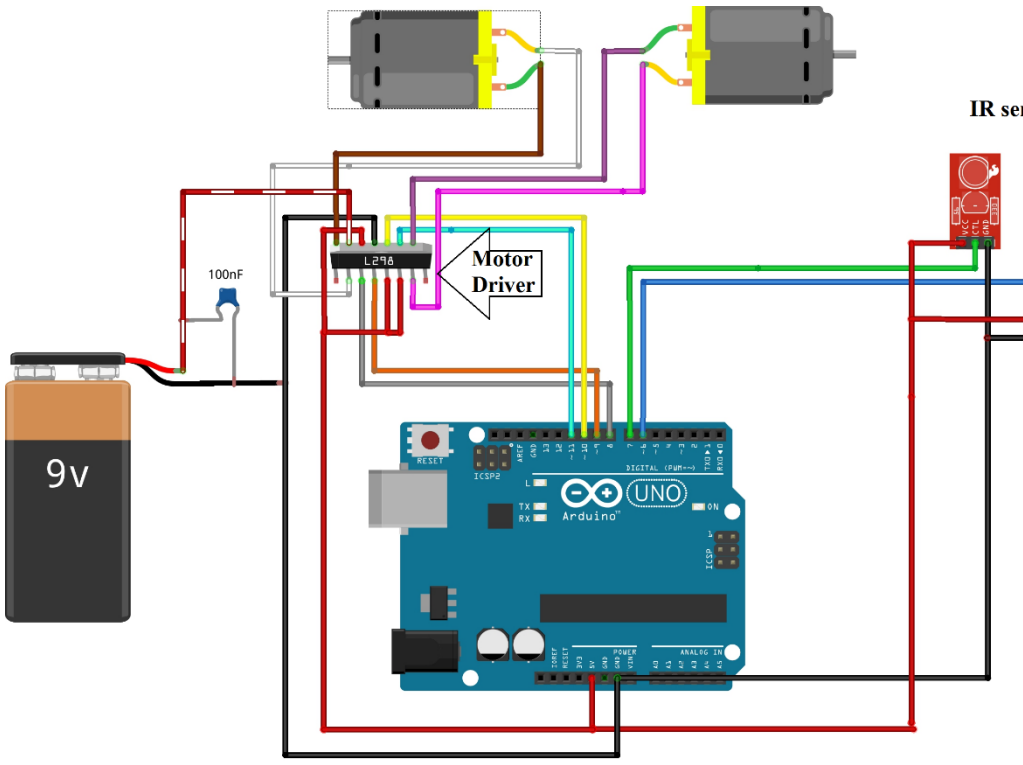


If right sensor sense black line then robot turn right side until both sensor comes at white surface. When white surface comes robot starts moving on forward again.



If both sensors come on black line, robot stops.

Circuit Diagram



Complete Circuit Diagram

IR sensor output pin connected to Arduino digital pin number 2 and 3 and motor driver's input pin 2, 7, 10 and 15 is connected to digital pin number 4, 5, 6 and 7 respectively. And one motor is connected at output pin of motor driver 3 and 6 and another motor is connected at pin 11 and 14.

Program Explanation

First, we defined input and output pin for sensors and motors. If conditions are used to compare IR input data for driving the motors.

```
/*-------definning Inputs------*/
#define LS 2      // left sensor
#define RS 3      // right sensor

/*-------definning Outputs------*/
#define LM1 4       // left motor
#define LM2 5       // left motor
#define RM1 6       // right motor
#define RM2 7       // right motor


if(digitalRead(LS) && digitalRead(RS))      // Move Forward
{
  digitalWrite(LM1, HIGH);
  digitalWrite(LM2, LOW);
```

There are four conditions in this line following robot. We are used two sensor namely left sensor and right sensor.

Input		**Output**				**Movement Of Robot**
Left Sensor	**Right Sensor**	**Left Motor**		**Right Motor**		
LS	RS	LM1	LM2	RM1	RM2	
0	0	0	0	0	0	Stop
0	1	1	0	0	0	Turn Right
1	0	0	0	1	0	Turn Left
1	1	1	0	1	0	Forward

We write program according to the conditions shown in table above.

Code:

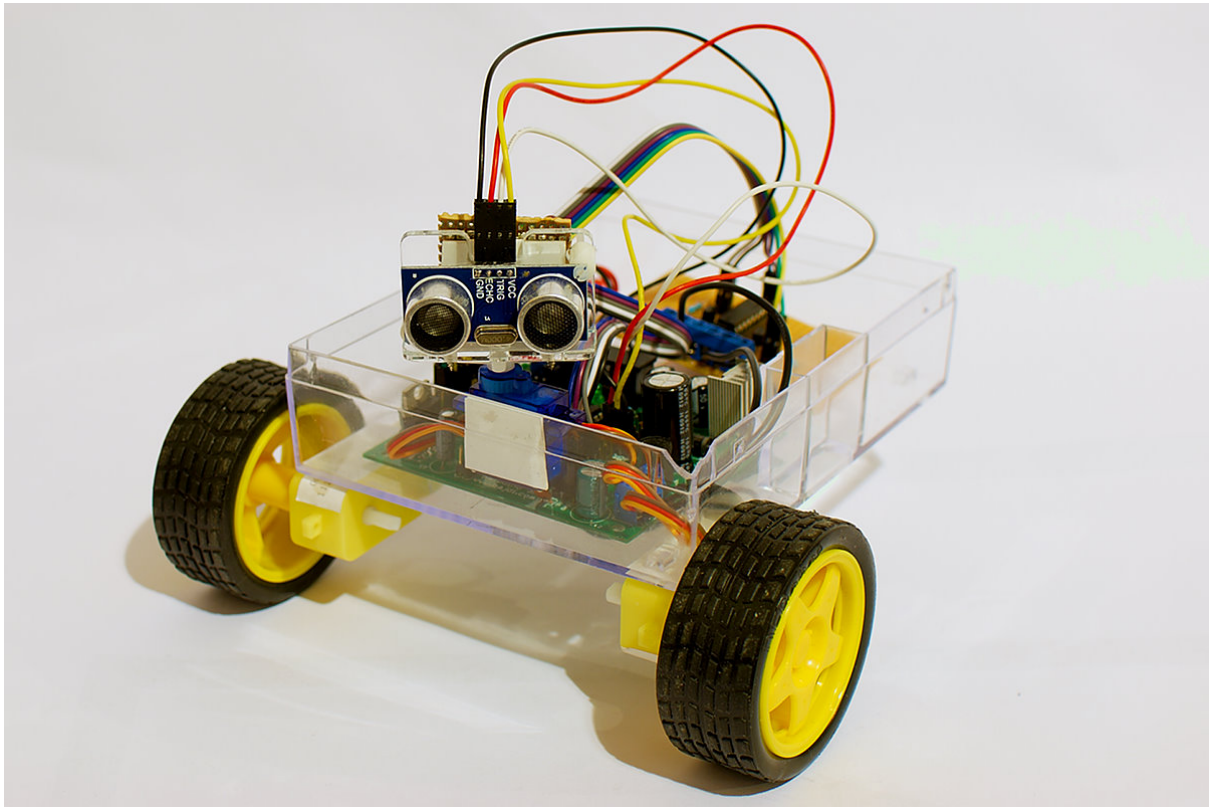
```
/*------ Program for Line Follower Robot using Arduino----- */
/*-------definning Inputs------*/
#define LS 2      // left sensor
#define RS 3      // right sensor
/*-------definning Outputs------*/
#define LM1 4       // left motor
#define LM2 5       // left motor
#define RM1 6       // right motor
#define RM2 7       // right motor
```

```
void setup()
{
 pinMode(LS, INPUT);
 pinMode(RS, INPUT);
 pinMode(LM1, OUTPUT);
 pinMode(LM2, OUTPUT);
 pinMode(RM1, OUTPUT);
 pinMode(RM2, OUTPUT);
}
void loop()
{
 if(digitalRead(LS) && digitalRead(RS))    // Move Forward
 {
  digitalWrite(LM1, HIGH);
  digitalWrite(LM2, LOW);
  digitalWrite(RM1, HIGH);
  digitalWrite(RM2, LOW);
 }

 if(!(digitalRead(LS)) && digitalRead(RS))    // Turn right
 {
  digitalWrite(LM1, LOW);
  digitalWrite(LM2, LOW);
  digitalWrite(RM1, HIGH);
  digitalWrite(RM2, LOW);
 }

 if(digitalRead(LS) && !(digitalRead(RS)))    // turn left
 {
  digitalWrite(LM1, HIGH);
  digitalWrite(LM2, LOW);
  digitalWrite(RM1, LOW);
  digitalWrite(RM2, LOW);
 }

 if(!(digitalRead(LS)) && !(digitalRead(RS)))    // stop
 {
  digitalWrite(LM1, LOW);
  digitalWrite(LM2, LOW);
  digitalWrite(RM1, LOW);
  digitalWrite(RM2, LOW);
 }
}
```

CHAPTER 9: OBSTACLE AVOIDER

Introduction

An Obstacle Avoiding robot detects any obstructions in its path and avoids it by taking deviation from its current path.The robot gets the information from surrounding area through sensors on the robot.



Concepts of Obstacle Avoider

Concept of working of obstacle avoider is related to sound. Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back.

By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.

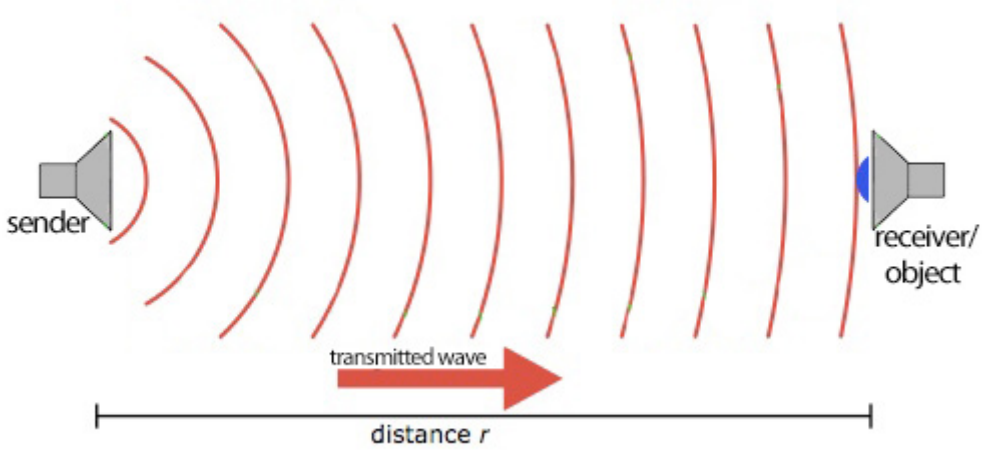


Diagram of the basic ultrasonic sensor operation

Since it is known that sound travels through air at about 344 m/s (1129 ft/s), you can take the time for the sound wave to return and multiply it by 344 meters (or 1129 feet) to find the total round-trip distance of the sound wave. Round-trip means that the sound wave traveled 2 times the distance to the object before it was detected by the sensor; it includes the 'trip' from the sonar sensor to the object AND the 'trip' from the object to the Ultrasonic Sensor (after the sound wave bounced off the object). To find the distance to the object, simply divide the round-trip distance in half.

$$distance = \frac{speed\ of\ sound\ \times time\ taken}{2}$$

Circuit Explanation

The whole circuit can be divided into 3 parts:

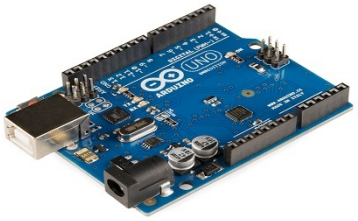
1. Sensor
2. Controller
3. Driver shield

Sensor

Ultrasonic Sensor sends out a high-frequency sound pulse and then times how long it takes for the echo of the sound to reflect. The sensor has 2 openings on its front. One opening transmits

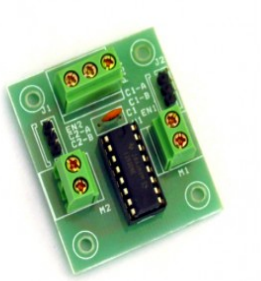
ultrasonic waves, (like a tiny speaker), the other receives them, (like a tiny microphone).

Controller



Arduino is used for controlling whole the process of line follower robot. Arduino read these signals and send commands to driver circuit to drive line follower.

Driver shield



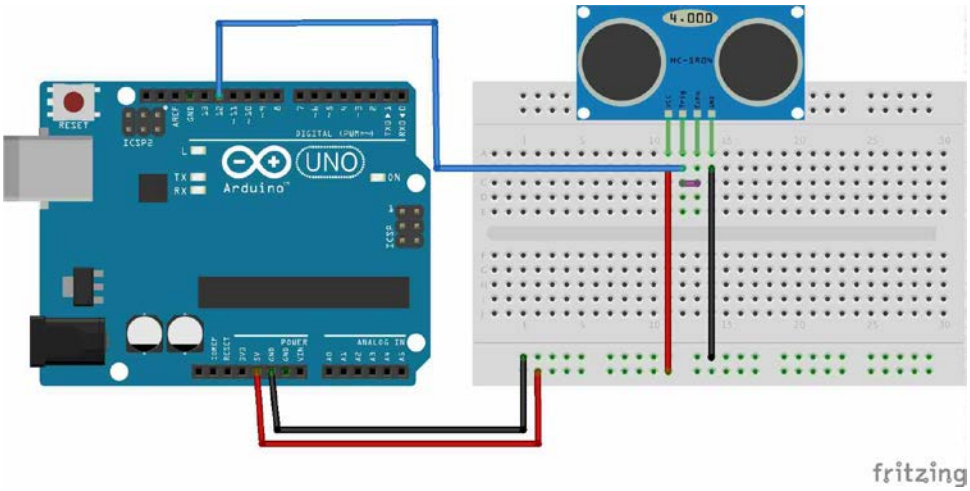
Driver consists motor driver and two DC motors. Motor driver is used for driving motors because Arduino does not supply enough voltage and current to motor. We add a motor driver circuit to get enough voltage and current for motor. Arduino sends commands to this motor driver and then it drive motors.

Working of Obstacle Avoider Robot using Arduino

The obstacle avoidance robotic vehicle uses ultrasonic sensors for its movements. The motors are connected through motor driver IC to microcontroller. The ultrasonic sensor is attached in front of the robot.

Whenever the robot is going on the desired path the ultrasonic sensor transmits the ultrasonic waves continuously from its sensor head. Whenever an obstacle comes ahead of it the ultrasonic waves are reflect from an object and that information is passed to the microcontroller. The microcontroller controls the motors left, right, back, front, based on ultrasonic signals. To control the speed of each motor pulse width modulation is used (PWM).

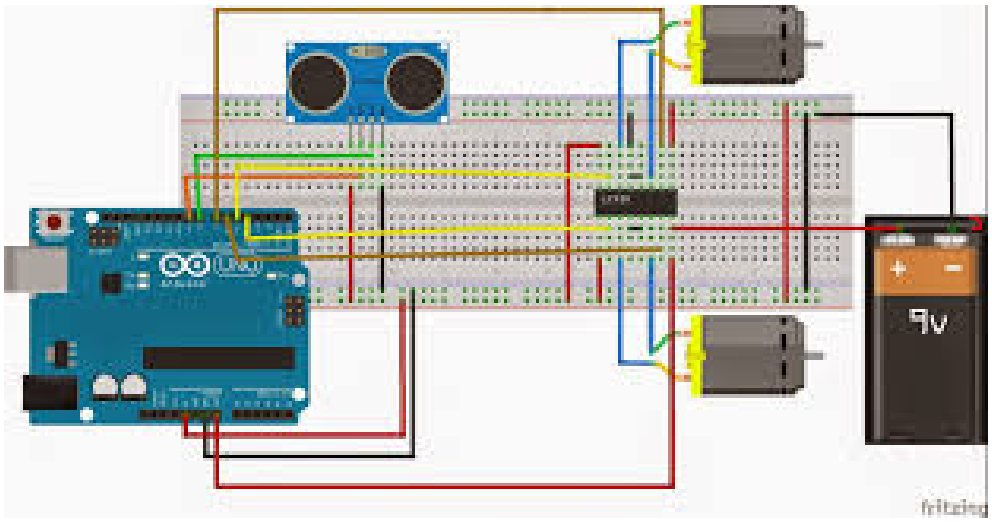
Circuit Diagram



Material

- Arduino UNO
- HC-SR04 Ultra sonic sensor
- Robot chassis
- 2 DC Motors
- 2 Wheels
- 1 Castor Wheel
- L293D motor driver
- Basic electronics kit contains breadboard, connecting wires, battery & other small useful items
- Jumper Wires: Male to Male, Male to Female
- Power source: 5V power bank or 5V battery
- Tools: Screw driver, Scissor / Wire stripper
- Software: Arduino IDE

Connecting the Sensor to the Arduino



Connect the motor driver with Arduino

Code

Motor and Ultrasonic pins connected with Arduino

motorr1 pin to 11
motorr2 pin to 10
motorl1 pin to 9
motorl2 pin to 8
trigger Pin to 3
echo pin to 2

```
#define trigPin 3
#define echoPin 2
#define motorr1 11
#define motorr2 10
#define motorl1 9
#define motorl2 8

void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(motorl1, OUTPUT);
```

```
  pinMode(motorr1, OUTPUT);
  pinMode(motorl2, OUTPUT);
  pinMode(motorr2, OUTPUT);
}

void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW);  // Added this line
  delayMicroseconds(2); // Added this line
  digitalWrite(trigPin, HIGH);
//  delayMicroseconds(1000); - Removed this line
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  if (distance < 20) {  // This is where the LED On/Off happens
    digitalWrite(motorr1,LOW); // When the Red condition is met, the Green LED should turn off
  digitalWrite(motorl1,LOW);
  digitalWrite(motorr2,LOW); // When the Red condition is met, the Green LED should turn off
  digitalWrite(motorl2,LOW);
}
  else {
    digitalWrite(motorr1,HIGH); // When the Red condition is met, the Green LED should turn off
  digitalWrite(motorl1,HIGH);
  digitalWrite(motorr2,LOW); // When the Red condition is met, the Green LED should turn off
  digitalWrite(motorl2,LOW);}

    Serial.print(distance);
    Serial.println(" cm");

  delay(100);

}
```

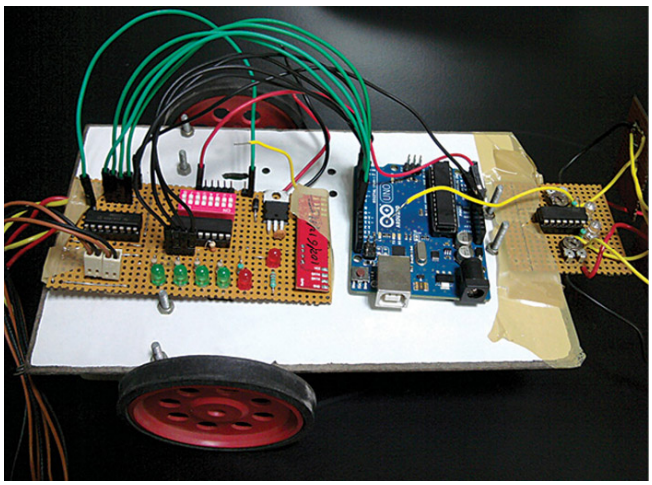
CHAPTER 10: RF CONTROLLED ROBOT

Introduction

These RF communication systems can be used to transmit signals from transmitting end to receiving end. The control signal from transmitter is sent to the receiver which is connected to an object or device or vehicle that is to be remotely controlled. For example, IR remote is used to control TV remotely.

Radio Frequency Technology

The rate of oscillation of frequency of alternating currents and radio waves which carry radio waves in the frequency range of 3 kHz to 300 GHz is called as radio frequency

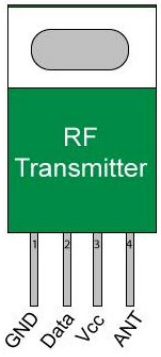


Concepts ofRadio Frequency Remote Controlled Wireless Robot

Radio frequency controlled robotic vehicle is designed using a robotic vehicle that is interfaced with radio frequency remote control. RF transmitter is used by control panel or controlling person and RF receiver is connected to the robotic vehicle that is to be controlled remotely. Radio frequency remote control works over an adequate range (up to 200 meters) by facilitating with proper antenna.

RF Transmitter

RF modules are usually very small in size and operate with voltage range of 3V to 12V. RF transmitter modules are designed to work with 433MHz frequency. If transmitting logic is zero, then no power is drawn by transmitter

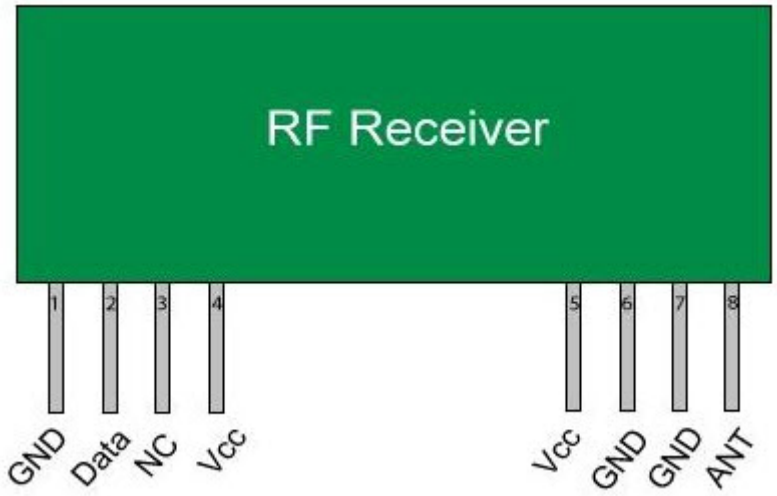


(consumes very low power if carrier frequency is fully suppressed). For transmitting logic one, it consumes power about 4.5mA with 3V.

RF transmitter circuit diagram is shown in the figure that shows different push buttons used to move the robotic vehicle in different directions like forward, backward, left and right. Thus, by pressing the appropriate push button, we can control the robotic vehicle's movement.

RF Receiver

RF receiver modules are also designed to work with 433MHz (that should match with the transmitter frequency for communication purpose to receive signals from the transmitter).



RF receiver has supply current of 3.5mA with an operating voltage of 5V and a frequency of 433MHz. RF transmitter circuit diagram is shown in the figure.

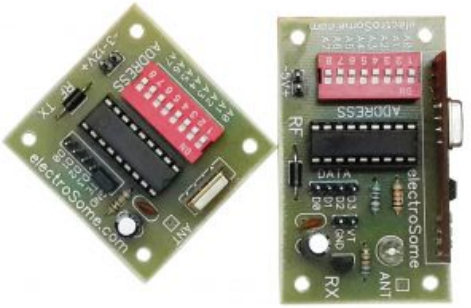
Circuit Explanation

The whole circuit can be divided into 3 parts:

1. RF transmitter and receiver
2. Controller
3. Driver shield

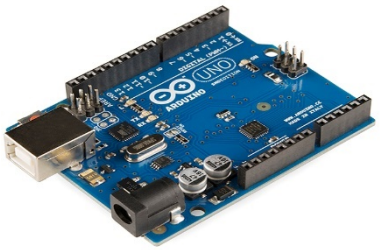
RF transmitter and receiver

A wireless radio frequency (RF) transmitter and receiver can be easily made using HT12D Decoder, HT12E Encoder and ASK RF Module. Wireless transmission can be done by using



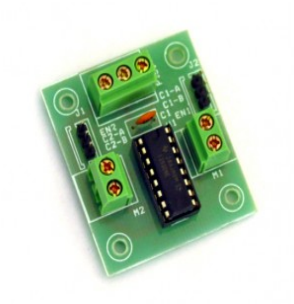
433Mhz or 315MHz ASK RF Transmitter and Receiver modules

Controller



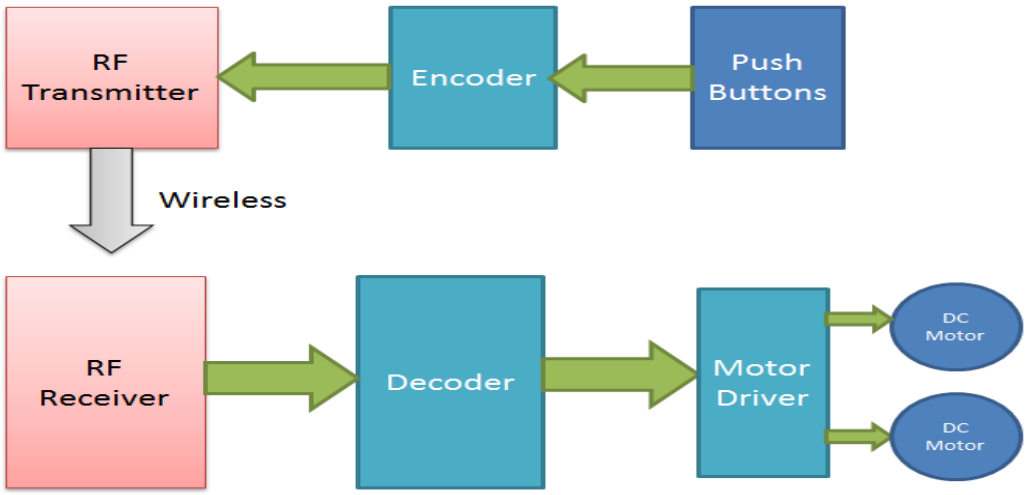
Arduino is used for controlling whole the process of line follower robot. Arduino read these signals and send commands to driver circuit to drive line follower.

Driver shield



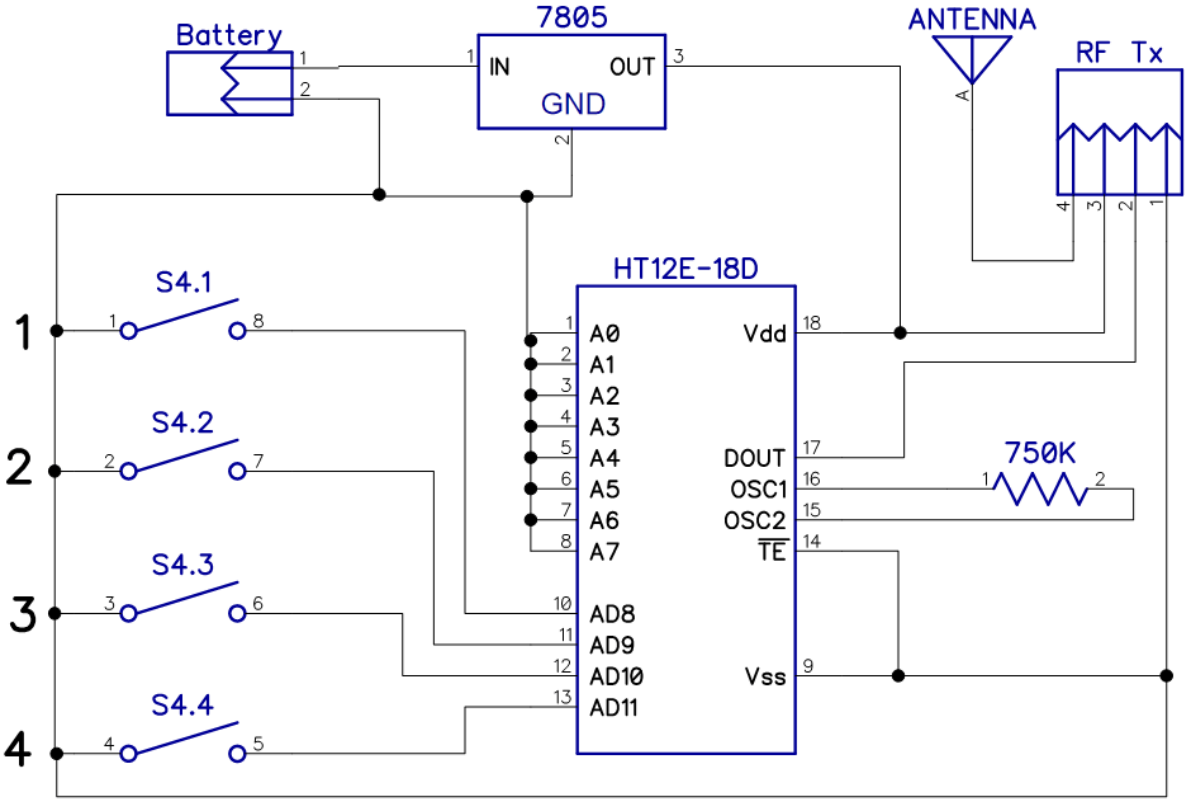
Driver consists motor driver and two DC motors. Motor driver is used for driving motors because Arduino does not supply enough voltage and current to motor. So we add a motor driver circuit to get enough voltage and current for motor. Arduino sends commands to this motor driver and then it drive motors.

Working of RF module

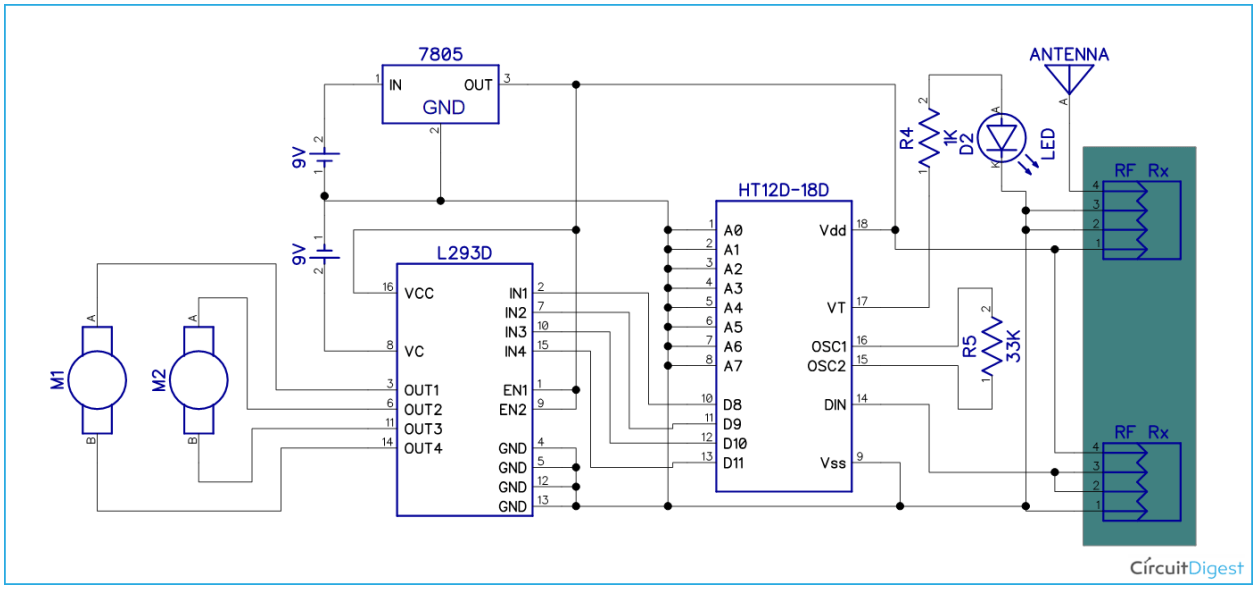


RF controlled robot is controlled by using Four push button placed at transmitter side. Here we only need to push the buttons to control the robot. A transmitting device is used in your hand which also contains a RF Transmitter and a RF Encoder. This transmitter part will transmit command to robot so that it can do the required task like moving forward, reverse, turning left, turning right and stop. All these tasks will perform by using four push buttons that are placed on RF transmitter.

Circuit Diagram for RF Transmitter:



Circuit Diagram for RF Receiver:



As shown in above figures, circuit diagrams for RF controlled robot are quite simple where a RF pair is used for communication. Connections for transmitter and receiver show in circuit diagrams.

RF Controlled Robot has two main parts namely:

1. Transmitter part
2. Receiver part

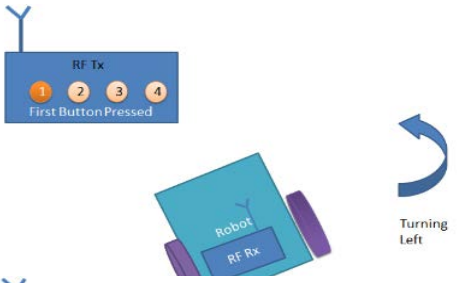
In transmitter part a data Encoder and a RF transmitter is used. As we have already mentioned above that we are using four push buttons to run the robot, these four buttons are connected with Encoder with respect to ground. When we will press any button encoder will get a digital LOW signal and then applied this signal serially to RF transmitter. The Encoder IC HT12E encodes data or signal or converting it into serial form and then sends this signal by using RF transmitter into the environment.

At the receiver end we have used RF receiver to receive data or signal and then applied to HT12D decoder. This decoder IC converts the received serial data to parallel and then send these decoded signals to L293D Motor driver IC. According to the received data robot runs by using two dc motor in forward, reverse, left, right and stop direction.

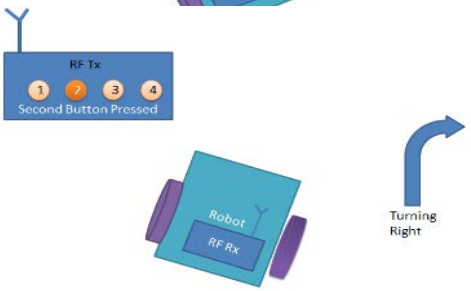
RF controlled robot move according to button pressed at Transmitter.

Button Pressed at Transmitter	**Moving Direction of Robot**
First (1)	Left
Second (2)	Right
First and Second (1 & 2)	Forward
Third and Fourth (3 & 4)	Backward
No Button Pressed	Stop

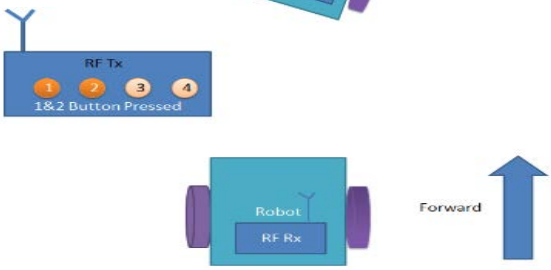
When we press first button (1 mention on circuit and hardware) robot start to moving left side and moving continues until the button is released.



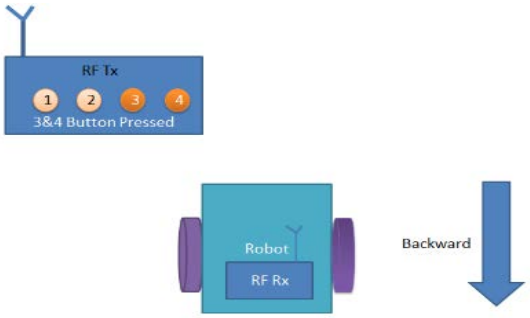
When we press second button at transmitter, robot start moving in right side until button is released.



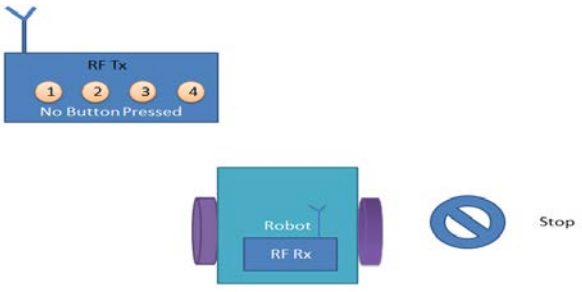
When we press first and second button at the same time, Robot start moving in forward direction until push buttons are released.



When we press third and fourth button at the same time, robot start moving in backward direction and keep going until push buttons are released.



And when no push Button is pressed, robot stops.

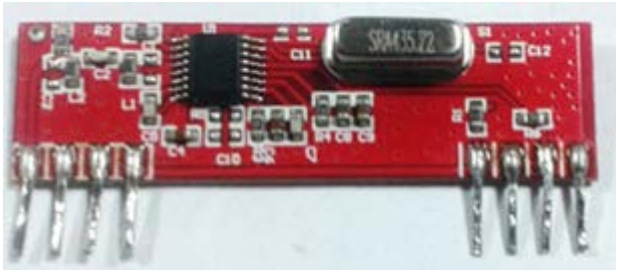


Required Components

- DC Motor - 2
- HT12D - 1
- HT12E - 1
- RF Pair - 1
- Motor Driver L293D - 1
- Robot Chassis – 1
- Arduino

RF Transmitter Features:

- Frequency Range: 433 Mhz
- Output Power: 4-16dBm
- Input supply: 3 to 12 volt dc



RF Receiver Features:

- Sensitivity: -105dBm
- IF Frequency : 1MHz
- Low Power Consumption
- Current 3.5 mA
- Supply voltage: 5 volt

This Module is very cost efficient where long range RF communication is required. This module does not send data using UART communication of PC or microcontroller directly because there is lots of noise at this frequency and its Analog technology. We can use this module with the help of encoder and decoder ICs which extract data from the noise.

The range of transmitter is about 100 meters at maximum supply voltage and for 5 volt the range of transmitter is about 50-60 meter with using a simple wire of single code 17cm length antenna.

Pin Description of RF transmitter

1. GND - Ground supply
2. Data In - This pin accept serial data from encoder
3. Vcc - +5 Volt should be connect to this pin
4. Antenna - A wrapped connect to this pin for proper transmission of data

Pin Description of RF Receiver

1. GND - Ground
2. Data In - This pin give output serial data to Decoder
3. Data In - This pin give output serial data to Decoder
4. Vcc - +5 Volt should be connect to this pin
5. Vcc - +5 Volt should be connect to this pin
6. GND - Ground
7. GND - Ground
8. Antenna

Code

For transmitter

Attach 9V battery positive terminal on RF transmitter VCC and negative terminal on Ground terminal

For receiver

```
int motor1 =3;
int motor2=4;
int motor3=5;
int motor4=6;


void setup() {
pinMode(MOTOR1,OUTPUT);
pinMode(MOTOR2,OUTPUT);
pinMode(MOTOR3,OUTPUT);
pinMode(MOTOR4,OUTPUT);
  pinMode(7,INPUT);
  pinMode(8,INPUT);
  pinMode(9,INPUT);
  pinMode(10,INPUT);
Serial.begin(9600);
}
void loop() {
  // put your main code here, to run repeatedly:
int i=digitalRead(7);
int j=digitalRead(8);
int k=digitalRead(9);
int l=digitalRead(10);
Serial.print(i);
Serial.print(j);
Serial.print(k);
Serial.println(l);
if(i==1)
```

```
{digitalWrite(motor1,HIGH);
digitalWrite(motor2,LOW);
digitalWrite(motor3,HIGH);
digitalWrite(motor4,LOW);
 delay(10);
}
else if(j==1)
{ digitalWrite(motor2,HIGH);
digitalWrite(motor1,LOW);
digitalWrite(motor4,HIGH);
digitalWrite(motor3,LOW);
 delay(10);
}
else if(k==1)
{ digitalWrite(motor1,LOW);
digitalWrite(motor2,LOW);
digitalWrite(motor3,HIGH);
digitalWrite(motor4,LOW);
 delay(10);
}
else if(l==1)
{digitalWrite(motor1,HIGH);
digitalWrite(motor2,LOW);
digitalWrite(motor3,LOW);
digitalWrite(motor4,LOW);
 delay(10);
}
}
```


ARDUINO