ELECTRONIC SIGNALS SYSTEM FOR RAILWAYS SAFETY

TEAM WORK

- تحت اشراف /

دكتور / حسين ابراهيم
مهندس / مصطفى الحاجري

أمير حسن أبو المجد	محمد سعد حلمي
علي عباس رمضان	احمد سعيد محمد
هاجر محمد فوزى	يوسف محمد سيد
فاطمه جمال	علي محمود خلاف
محمد جمال	ابانوب رشدي بشري
عمر عبد العليم	كريم صبحي عبد الرحمن

BOOK CONTENT

INTRODUCTION

INTRODUCTION

الهيئة القومية لسكك حديد مصر تعرف اختصاراً باسم (س ح م) تم تاسيسها في عام ١٨٥٣ في خمسنيات القرن التاسع عشر و أشرف علي بنائها الخدويوي اسماعيل وهي شركة قطاع عام تمتلكها الحكومة المصرية بالكامل و تشغّل خطوط السكك الحديد المصرية والتي هي الثانية في العالم حيث تأسست بعد تأسيس السكك الحديد البريطانية بفترة وجيزة تعاني اليوم من خلل في منظومة الصيانة و تداعيات إهمال أصابها على مدى عقود فاتت مما يجعل معدل الحوادث عليها أعلى بكثير من نسب باقي دول العالم و بعد حادث قطار منفلوط في أواخر سنة ٢٠١٢ وضعت الحكومة المصرية خطة لتطوير المزلقانات و التي مخطط تطوير جميعها قبل نهاية سنة ٢٠١٣. و تحد المزلقانات الجديدة و التي افتتح أولها في بداية ٢٠١٣ على العامل البشري الذي كان حاضرا في السابق و تعتمد بوابات صلبة بدلا من السلاسل الحديدية تعد سكك حديد مصر هي أول خطوط سكك حديد يتم إنشاؤها في أفريقيا والشرق الأوسط، والثانية على مستوى العالم بعد المملكة المتحدة حيث بدأ إنشاؤها في ١٨٣٤ إذ مدت قضبان خطوط السكة الحديد فعلا وقتها في خط السويس الإسكندرية إلا أن العمل ما لبث أن توقف بسبب اعتراض فرنسا لأسباب سياسية ثم أحييت الفكرة مرة أخرى بعد ١٧ عاما في ١٨٥١ في خمسينيات القرن التاسع عشر حيث تمتد عبر محافظات مصر من شمالها إلى جنوبها بدأ إنشاء أول خط حديدي في مصر يوم ١٢ يوليو عام ١٨٥١ و بدأ التشغيل في ١٨٥٤ و مما يذكر أن المشرف على مشروع بناء السكك الحديدية المصرية

INTRODUCTION

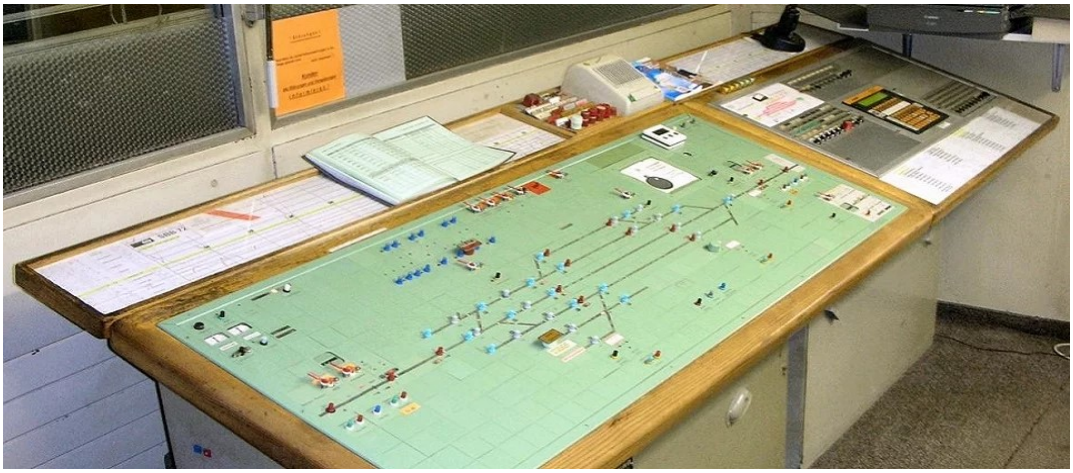
آنذاك كان المهندس الإنجليزي روبرت ستيفنسون وهو ابن مخترع القاطرة الشهير جورج ستيفنسون . ويبلغ طول السكه الحديدية حوالي ٩٠٠٠ كم منها حوالي ٤٨٧٢ كم تشكل مجموع أطوال الخطوط الطوالى و يبلغ عدد المحطات والمواقف و نقط البلوك على الشبكة أكثر من ٧٠٥ محطة منها عشرون محطة رئيسية في عواصم المحافظات في الدلتا والقناة و الوجه القبلي وبذلك تربط شبكة السكك الحديدية الوادي من أقصاة إلى أدناة على طول أكثر من ١٠٠٠ كيلو متر و تصل الخطوط الحديدية بين معظم المراكز العمرانية و الاقتصادية في البلاد مثل الموانئ على البحرين الأحمر و المتوسط و مراكز الشحن الخام و المصانع فضلا عن جميع المدن و المراكز الأخرى في شبكة هائلة و متسعة من الطرق الحديدية الحديثة المجهزة لتشغيل جميع أنواع القطارات ذات السرعات العالية عليها. لتأمين السلامة ولمنع الحوادث نحتاج إلى أنظمة تحكم بمسار القطار، أي بالسكك الحديدية، مهمتها معرفة بداية ونهاية طريق القطار التأكد من فراغ الطريق من أي قطار آخر - تحريك كل التحويلات الحديدية في اتجاه حركة السير المطلوبة يمكن اختصار عمل نظام التحكم في السكك الحديدية في جملة واحدة فقط، ألا وهي في آن واحد لايُسمح بتواجد أكثر من قطار بين اشارتين تطورت أنظمة التحكم في السكك الحديدية مع مرور الزمن، فمن التحكم اليدوي عن قرب إلى التحكم الميكانيكي عن بعد، ومنه إلى التحكم الكهربائي والتي تقوم بتحويل قوانين سير القطارات إلى دارات كهربائية .في العصر الحالي تعتبر أنظمة التحكم الإلكتروني في السكك الحديدية، والتي تستخدم أجهزة الحاسوب، أكثر الأنظمة تطوراً في هذا المجال

نظام التحكم الميكانيكي /

نظام التحكم الميكانيكي في السكك الحديدية هو نظام للتحكم عن بعد في التحويلات والإشارات المعدنية وكل العناصر المتحركة الأخرى التحريك عن بعد يتم بنقل القوة العضلية إلى العناصر المعنية عن طريق استخدام الأذرعة الحديدية المرتبطة ببكرات ومسننات وكابلات معدنية

نظام التحكم الكهربائي /

من مميزاته القدرة على تحويل قوانين سير القطارات إلى دارات كهربائية باستخدام الريلاي كما أن من الميزات الشاشة المعدنية وقد رسم عليها مخطط للسكك الحديدية في المحطة وزودت بمصابيح صغيرة مختلفة الألوان تساعد على تتبع حركة القطارات ومراقبة الإشارات الضوئية .تشييد طريق للقطار يتم بالضغط على مفتاحين كهربائيين، الأول يحدد بداية الطريق والثاني يحدد نهايته.



نظام التحكم الالكتروني /

في منتصف الثمانينات من القرن العشرين دخل الحاسوب عالم القطارات وظهر مايسمى بالنظام الإلكتروني للتحكم في السكك الحديدية. يتكون هذا النظام من ثلاث طوابق

طابق المراقبة -- طابق التحكم المركزي -- طابق التحكم بالعناصر الخارجية

من مزايا النظام الإلكتروني للتحكم في السكك الحديدية:

صغر الحجم وأناقة التصميم

سهولة تطوير وتعديل النظام

القدرة الفائقة على تزويد المراقب بالمعلومات الإضافية

ويوجد حوادث كثيره علي خطوط السكه الحديديه من بدايه التسعينات حتي الآن ومن أشهر الحوادث (حادثه قطار العياط ٢٠٠٩ - حادثه قطار دهشور ٢٠١٢ – حادثه قطار منفلوط ٢٠١٣ – حادثه قطار محطه مصر ٢٠١٩).

- في ٣ فبراير ٢٠٢١ أعلنت الهيئة القومية لسكك حديد مصر أنه أثناء مرور قطار ٢٠١٠ «القاهرة / أسوان» على مزلقان فزارة المطور؛ بين محطتي ديروط والقوصية اقتحمت عربة نصف نقل مجهزة بونش، الحارة المخالفة للمزلقان وهو مغلق مما أدى إلى اختراق غراب الونش الموجود على العربة كابينة جرار القطار؛ ووفاة مساعد سائق القطار و٢ من عمال هندسة سكة متواجدين بالجرار.

- في ٢٦ مارس ٢٠٢١ اصطدام قطار مكيف بقطار ركاب من الخلف بدائرة مركز طهطا بسوهاج ما نتج عنه خروج ٣ عربات عن القضبان راح ضحية الحادث ٣٢ مواطنًا وأصيب ١٠٨ مواطن.

- في ١٨ أبريل ٢٠٢١ خرج قطار متجه من القاهرة الي المنصورة عن القضبان قرب محطة سكة حديد طوخ وراح ضحية الحادث عدد من المواطنين بالإضافة إلى إصابة ما يقرب من ١٠٣ أشخاص .

و بعد كل هذه الحوادث فكرنا في تطوير شبكه السكه الحديد وعمل بعض وسائل الأمان .

1- Rf WIRELESS

This Device Works To Send And Receive Signals, Meaning That This Device Is Placed At The Beginning And End Of Each Train, So That If A Qatar Stops During Its Track Or Stops And A Second Train Moves Behind It On The Same Itinerary Quickly, This Device Sends A Signal To The Moving Diameter Towards It With Its Eyes At A Certain Distance Which Is Defined As About 500 Meters Separate The Moving Train So That No Two Trains Run At The Same Distance.

2 - Load cell sensor

This is a weight sensor when there is a broken vehicle, winch, tractor, or something at a crossroads. It sends a signal to the towing that something on the pedestrian crossing at a certain distance needs to be determined, which, in the driver's eyes, slows down or stops the train.

3 - Servo motor & IR

This is for car lane opening and closing signals at intersections. At the beginning of the train's arrival at the crossroads or the beginning of its entry at a certain distance, a signal is sent to (servo motor) via an infrared sensor to lock the road on cars and not overtake them until the train passes completely with all its vehicles and with the passing of the last vehicle on the train sends another signal to Servo drive to open the way for cars to pass.

4 - LEDs

The wires can also be connected to each car in parallel, and the end of these wires is connected (with a light bulb) in the driver's room so that if the car breaks, the lamp lights up and gives a warning to the driver who has stopped the train

CHAPTER 1
SLIPWAY

IR SENSOR &SERVO MOTOR CIRCUIT

Circuit Idea :

When The Train Passes In Front Of The First Sensor, The Servo Motor Rotates 90 Degrees And An Alarm Bell Alerts Pedestrians At The Crossing.

When The Train Passes Completely And The Second Sensor Receives A Physical Signal From The Movement Of The Train, The Engine Will Return To Its Normal Position And The Alarm Bell Will Stop Working. And The Component ARE

CONTROL UNIT ARDUINO UNO

SERVO MOTOR

2 SENSOR IR

AN ALARM

IR SENSOR

The sensor is affected by the light surrounding it, so the designated part is covered by receiving the reflected rays from the object falling on it.

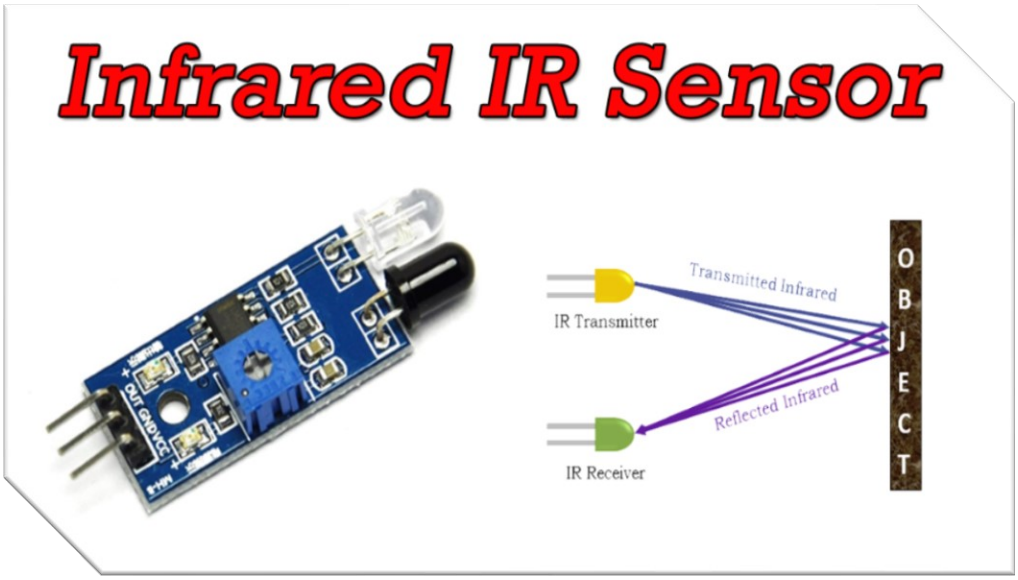
This light cannot be seen with the naked eye but can be seen by the camera

The IR sensor (Infrared) is an infrared optical sensor in the wavelength range 780 nm to 50,000 nm.

The rays are sent from the IR LED to hit the object opposite it and are received by the photo diode.

Infrared sensors are now widely used in motion detectors, which are used in building services to power lights or in alarm systems.

It can detect colors and can be used digitally or analogue. (This is why they are also used in night vision cameras.)



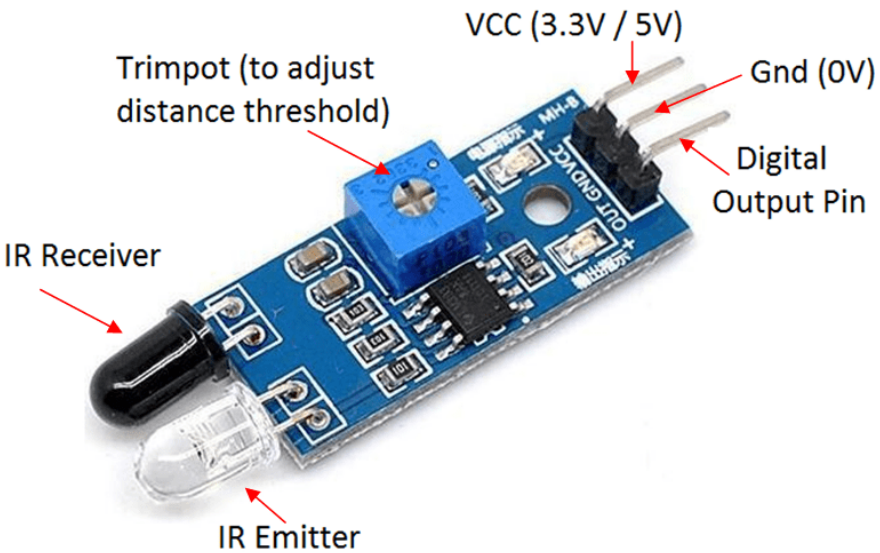
For the application of colour detection: the amount of reflected light depends on the colour of the surface from which it is reflected, the reflection varies with different coloured surfaces. This makes it a colour detector.

The photo diode gives a quick response in terms of the change in resistance when light falls on it, and this change is measured in terms of voltage.

The distance within which the sensor is required to operate is set by the variable resistance on the sensor chip.

The amount of reflection and reception varies with distance.

This difference causes the input voltage to change through the infrared input. This difference in the input voltage is used for proximity detection.



SERVO MOTOR

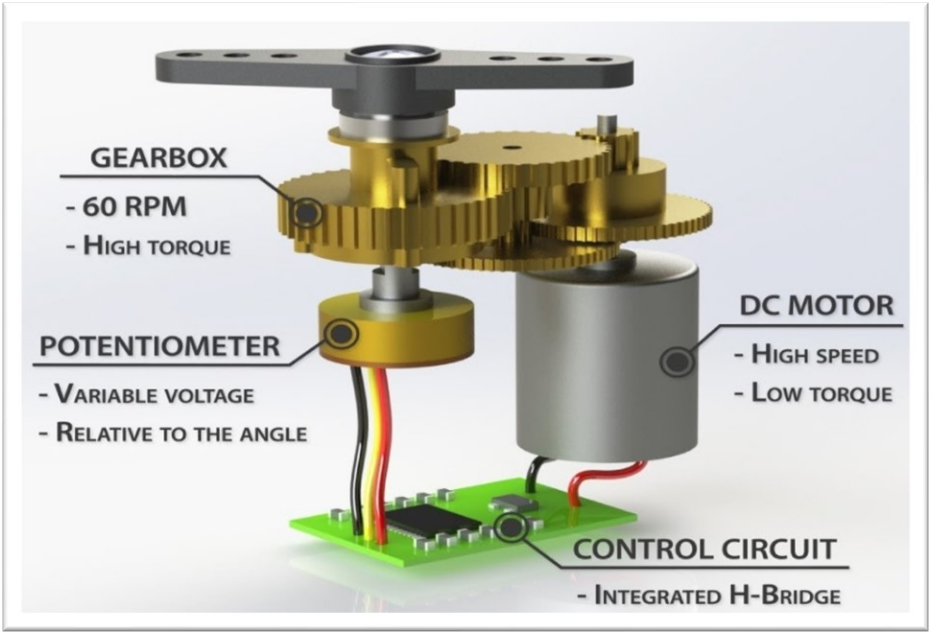
The servo motor is made up of four parts: a normal DC motor, a gear reducer, a position sensor, and a control circuit.

A servo motor is a motor that allows precise control of angle, acceleration and speed, capabilities that an ordinary motor does not have. It uses a normal motor and is connected to a sensor to take a look at its status. The control unit is the most complex part of a servo motor, as it is specially designed for this purpose.

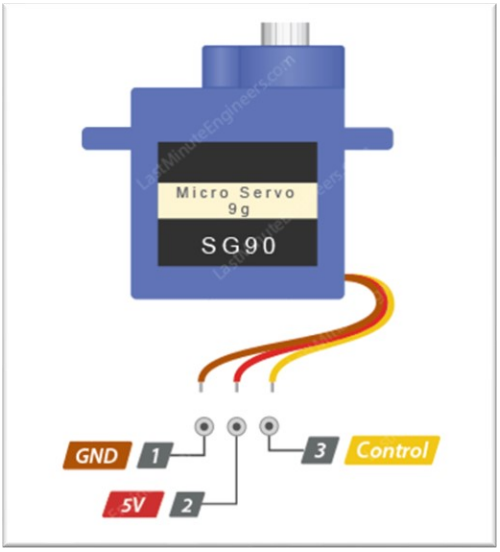
A servo motor is a simple electric motor, which is controlled with the help of a mechanism.

It is a DC motor whose speed is slowly decreased by gears. The angle at which it rotates in servo motors is usually from 90° to 180°.

Most servo motors rotate between specific angles, not all 360 degrees.



The terminals of the servo motor are three wires:



- ✓ A wire (black/brown) is connected to ground, a wire (white/yellow) is connected to the console, and a red wire is connected to the power supply.
- ✓ The motor works when receiving a signal from the shaft control circuit and stops when the required angle is reached.
- ✓ The position sensor is used to find out where the shaft is rotating, so it knows which direction the motor must rotate to move the shaft to the desired position. Usually the shaft does not rotate freely around a DC motor, but instead can only rotate a specific angle

Features of the servo motor:

- ✓ Superfast response, meaning that the engine speed reaches the required value in a short time.
- ✓ The engine accepts repeated disconnection and connection operations, no matter how many.
- ✓ High power for the size and power of the engine.

CONNECTIONS

✓ Connecting the sensor:

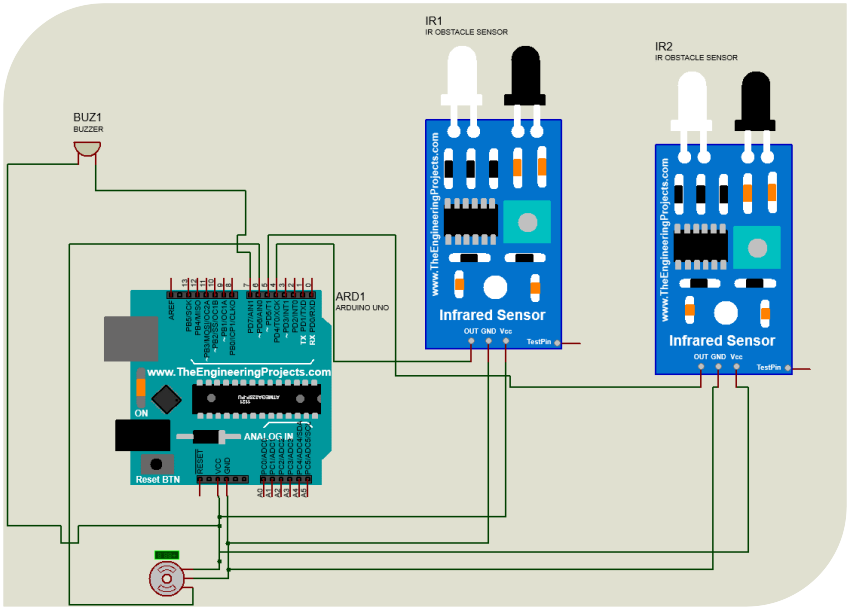
The two terminals, GRD, 5V are connected to the output of the GRD, 5V on the arduino. The control end is connected to an output from the digital outputs on the arduino.

✓ Connecting the servo motor:

The terminals GRD, 5V, are connected to the output of the GRD, 5V on the arduino and The control terminal is output from the digital outputs in the arduino.

✓ Connecting the alarm bell:

The positive terminal is connected to the output of the Arduino control outputs, and the negative terminal to GND.



PROGRAMMING CODE

And write the code so that the engine works and rotates 90 degrees and the alarm bell works when receiving a signal from the first sensor and does not return to its original position and the bell stops only when receiving a signal from the second sensor.

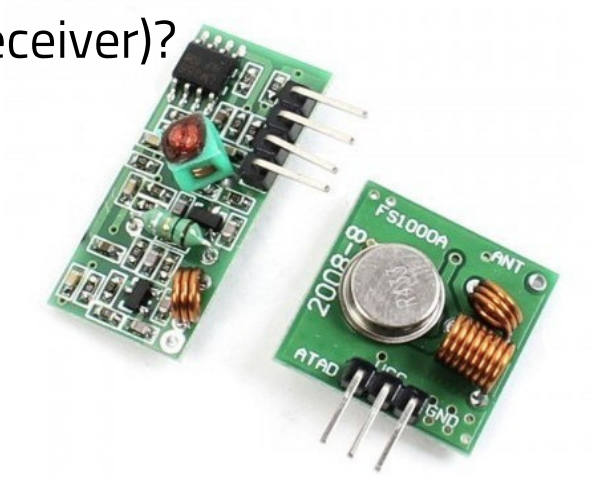
```
#include <Servo.h>
Servo tap1_servo;
Servo tap2_servo;
int IR_sensor1=4;
int IR_sensor2=3;
int val;
int val_1;
void setup() {
  pinMode(IR_sensor1,INPUT);
  pinMode(IR_sensor2,INPUT);
  tap1_servo.attach(2);
  tap2_servo.attach(5);
  pinMode ( 7 , OUTPUT); }
void loop() {
  val=digitalRead(IR_sensor1);
   if(val==0)
{ tap1_servo.write(0);
    tap2_servo.write(0);  }
    val_1= digitalRead
(IR_sensor2);
if(val_1==0)
  { tap1_servo.write(90);
    tap2_servo.write(90); }
if( val==0)
{ tone (7,700,200);
    delay(200);
    tone ( 7,350,150);
    delay(200);
     digitalWrite (7 , HIGH); }
if (val_1==0){
     digitalWrite (7 , LOW);}}
```

CHAPTER 2
RF WIRELESS TO AVOID TRAIN ACCIDENTS

RF TRANSMTTER - RECEIVER

- What is the RF (Transmtter & receiver)?

A small electronic device Used to transmit and/or receive radio signals between two devices.



- How the RF Sensor work?

The sensor functions by detecting a change in electromagnetic properties of the volume forming its 'sensitive region, which is located just outside the mouth of the coil and forms an approximate cube with sides of a similar diameter to the antenna coil. An object moving through the sensitive region, or the presence of a different material can cause such a change. 'Because the sensor uses an electromagnetic field rather than either a purely magnetic or electric field,' explains Adams, 'it is sensitive to both magnetic and electric properties of materials intruding into the sensitive volume at the same time, for example, the teeth of a rotating metallic cog moving through the sensitive region.' Other examples include homogeneous liquid flowing within a pipe located in the sensitive region, which contains small bubbles or pieces of solid debris carried along in the flow. As the liquid passes through the sensor a signal is produced that is proportional to the size or mass of the debris and for which the signal type is characteristic of that particular material.

The feature of the rf sensor in our project

In our project we use the RF senor as a way to protect against the collision of two trains if they are on the same direction or rail

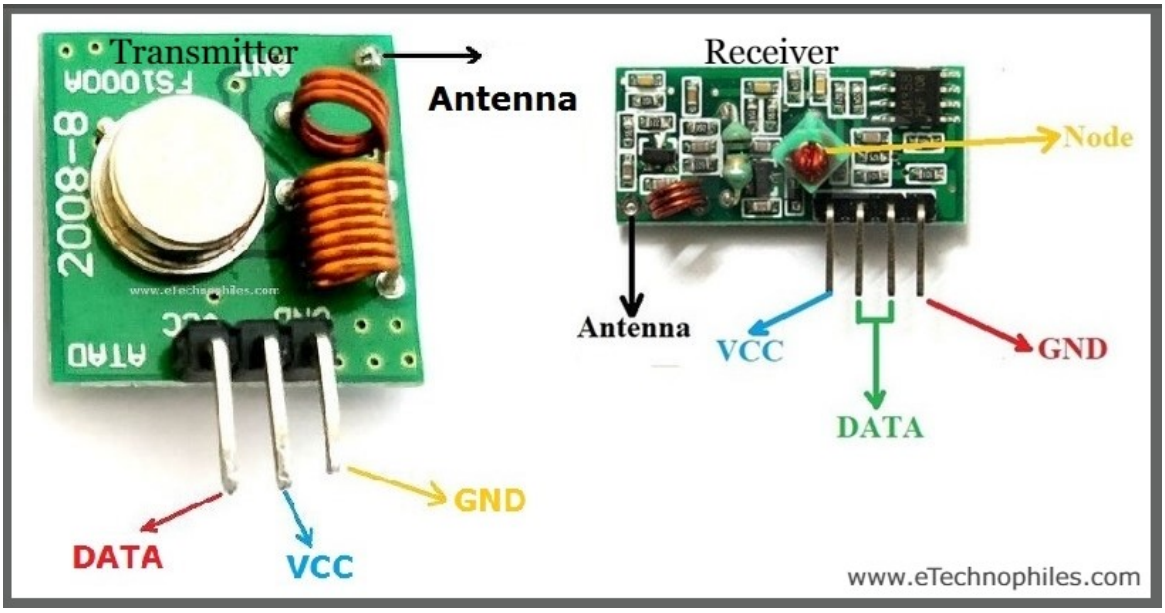
The transmitting part of the signal will be placed at the end of the front train, and we will place the receiving part of the signal at the beginning of the next train from the back, which will be connected to a relay controlled by the signal coming out of the receiving part of the RF sensor. When a train comes from behind and is on the same road The first train, the sending part will send a signal received by the receiving part and then give an electrical signal to the relay to disconnect the motor controlling the running of the coming train from behind and stop the train

Advantages of the RF Sensor

The sensor has advantages over competing devices in that:

- It requires no additional components such as magnets, coils, or magnetic circuits.
- It can be very small and the electronics may be accommodated on 100µm square of silicon wafer.
- It has a high signal-to-noise ratio, which can be maximized by optimizing the sensor design for a particular application.
- Unlike inductive sensors, its output is speed-independent

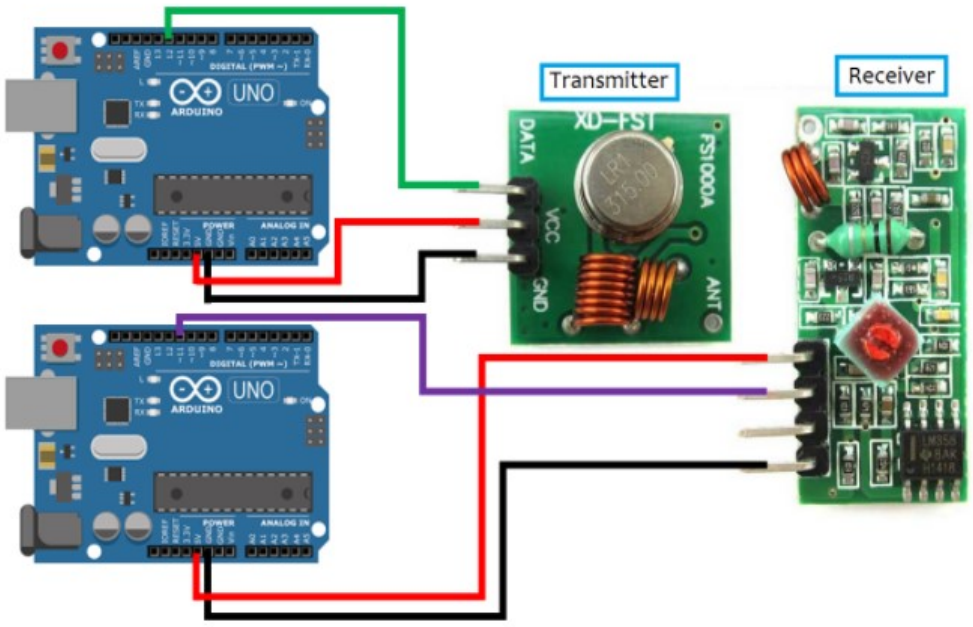
RF Sensor PINS



How do I connect the sensor to the Arduino?

The wiring for the transmitter is fairly simple. It has only three connections. Connect the VCC pin to 5V pin and GND to ground on the Arduino. The Data-In pin should be connected to Arduino's digital pin #12. You should try and use pin 12 as by default the library we'll be using in our sketch uses this pin for data input.

The wiring for the receiver is just as easy as the transmitter was. Once again there are only three connections to make. Connect the VCC pin to 5V pin and GND to ground on the Arduino. Any of the middle two Data-Out pins should be connected to digital pin #11 on the Arduino



1-433 MHZ TX Pinout

DATA pin accepts digital data to be transmitted.

VCC supplies power for the transmitter. This can be any positive DC voltage between 3.5V to 12V. Note that the RF output is proportional to the supply voltage i.e. the higher the Voltage, the greater the range will be.

GND is a ground pin.

Antenna is a pin for external antenna. As discussed earlier, you will want to solder a 17.3 cm piece of solid wire to this pin for the improved range.

2-433 MHZ RX Pinout

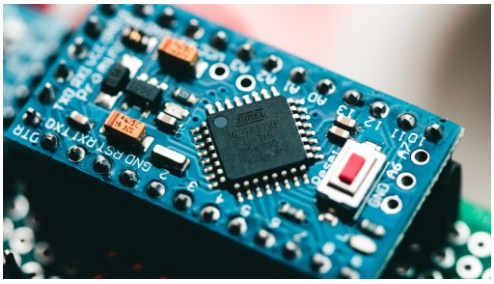
- VCC supplies power for the receiver. Unlike the transmitter, supply voltage for receiver needs to be 5V.
- DATA pins output the digital data received. The two center pins are internally tied together, so you can use either one for data out.
- GND is a ground pin.
- Antenna is a pin for external antenna which is often unmarked. It is the pad in the lower left of the module, right next to the small coil. Again, you will want to solder a 17.3 cm piece of solid wire to this pin for the improved range.

How do you check if RF transmitter is working?

- Test the circuit by putting the antenna near a source of noise, like a computer monitor, or a motor, or your wireless access point, or your laptop's WiFi antenna, or any other radio transmitter you have that you know works.
- Then, put it near your RF transmitter. Try transmitting something. If the ammeter moves, it's transmitting something.

CIRCUIT COMPONENTS

2x Arduino Nano



3x Push Buttons

4 Green Led

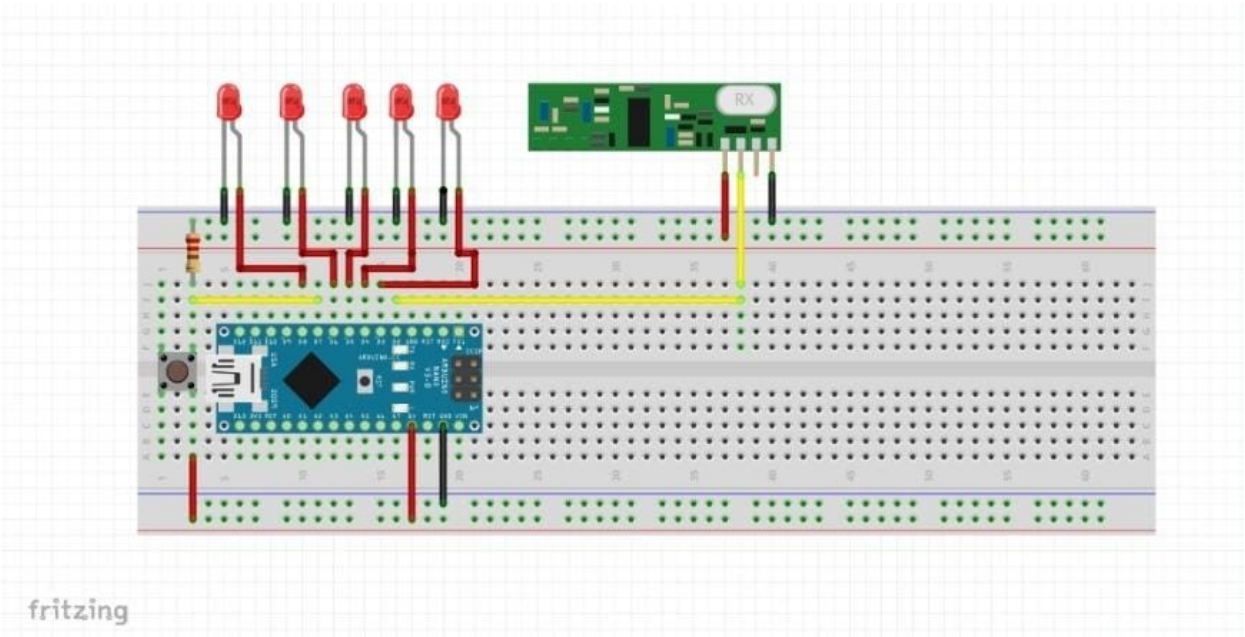
4 Yellow Led

Transmitter

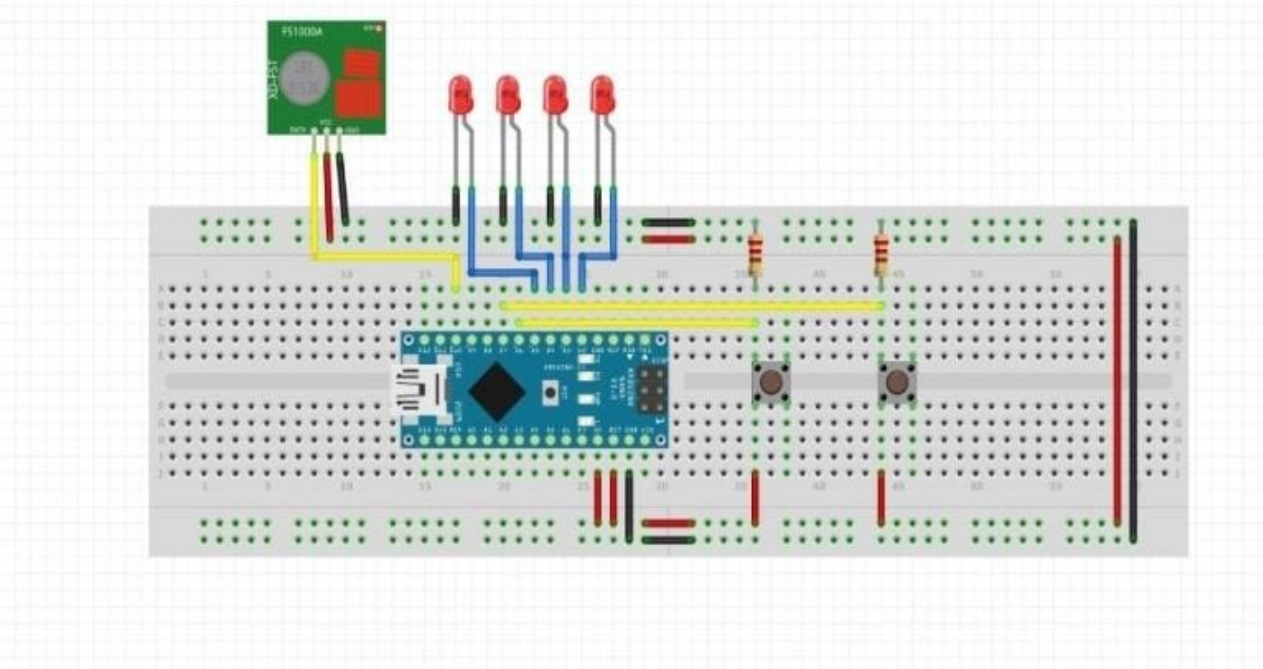
Receiver

CONNECTIONS

- Receiver /



- Transmitter /



PROGRAMMING CODE FOR RECEIVER

```
#include <RCSwitch.h>

const int btnSelectLinePin   = 7;

const int ledNormalLineAPin = 3;
const int ledNormalLineBPin = 4;

const int ledFaultLineAPin  = 5;
const int ledFaultLineBPin  = 6;

const int trailRelayPin     = 8;

int lineState    = 0;

bool lineAFlag  = true ;
bool lineBFlag  = false ;

bool statusFaultlineAFlag   = false ;
bool statusFaultlineBFlag   = false ;

RCSwitch mySwitch = RCSwitch();
 Serial.begin(115200);

  mySwitch.enableReceive(0);
  // Arduino Uno Receiver on interrupt 0 => that is pin #2
```

```
 pinMode(ledNormalLineAPin  , OUTPUT );
  pinMode(ledNormalLineBPin  , OUTPUT );

  pinMode(ledFaultLineAPin  , OUTPUT );
  pinMode(ledFaultLineBPin  , OUTPUT );

  pinMode(trailRelayPin     , OUTPUT );

  pinMode(btnSelectLinePin    , INPUT  );

  lineState    = 0;

  Serial.println("Hello Trains Receiver");

}

void loop() {
  lineState   = digitalRead(btnSelectLinePin);

  if ( lineState == HIGH ) {
     Serial.println("lineState Pressed");

     if ( lineAFlag  ) {
          lineAFlag  = false ;
          lineBFlag  = true ;
     } else {
          lineAFlag  = true ;
          lineBFlag  = false ;
     }
  }
```

```
if ( lineAFlag ) {

//     Serial.println("Train on Line A");
     digitalWrite(ledNormalLineAPin    ,HIGH );
     digitalWrite(ledNormalLineBPin    ,LOW );
//     mySwitch.send("A", 24);
//     mySwitch.send(1, 24);

  } else {
//    Serial.println("Train Not Stopped on Line A");
    digitalWrite(ledNormalLineAPin    ,LOW );

  }

  if ( lineBFlag ) {

//     Serial.println("Train on Line B");
     digitalWrite(ledNormalLineBPin    ,HIGH );
     digitalWrite(ledNormalLineAPin    ,LOW );
//     mySwitch.send("A", 24);
//     mySwitch.send(1, 24);

  } else {
//    Serial.println("Train Not Stopped on Line A");
    digitalWrite(ledNormalLineBPin    ,LOW );

  }
```

```
if (mySwitch.available()) {

  int value = mySwitch.getReceivedValue();
  Serial.print("value = ");
  Serial.println(value);

  switch (value){
   case 1:
    Serial.println("Fault on Line A");
    digitalWrite(ledFaultLineAPin  ,HIGH );
    digitalWrite(ledFaultLineBPin  ,LOW );
    statusFaultlineAFlag  = true ;
    statusFaultlineBFlag  = false ;
   break;

   case 2:
    Serial.println("Fault on Line B");
    digitalWrite(ledFaultLineAPin  ,LOW );
    digitalWrite(ledFaultLineBPin  ,HIGH );
    statusFaultlineAFlag  = false ;
    statusFaultlineBFlag  = true ;
   break;

   case 3:
    Serial.println("Normal Status");
    digitalWrite(ledFaultLineAPin  ,LOW );
    digitalWrite(ledFaultLineBPin  ,LOW );
    statusFaultlineAFlag  = false ;
    statusFaultlineBFlag  = false ;
   break;  }
```

```
If ( statusFaultlineAFlag && lineAFlag ) {
    Serial.println("Relay Stopped A ");
     digitalWrite(trailRelayPin  ,LOW );
 }
//  else {
//        Serial.println("Relay Not Stopped A ");
//      digitalWrite(trailRelayPin  ,HIGH );
//  }

 if ( statusFaultlineBFlag && lineBFlag ) {
       Serial.println("Relay Stopped B ");
     digitalWrite(trailRelayPin  ,LOW );
 }
//  else {
//     Serial.println("Relay Not Stopped B ");
//      digitalWrite(trailRelayPin  ,HIGH );
//  }

 if ( !statusFaultlineAFlag && !statusFaultlineBFlag ) {
    Serial.println("Not Not");
     digitalWrite(trailRelayPin  ,HIGH );
 }

  mySwitch.resetAvailable();

  Serial.println("  ");
  }
 delay(250);
}
```

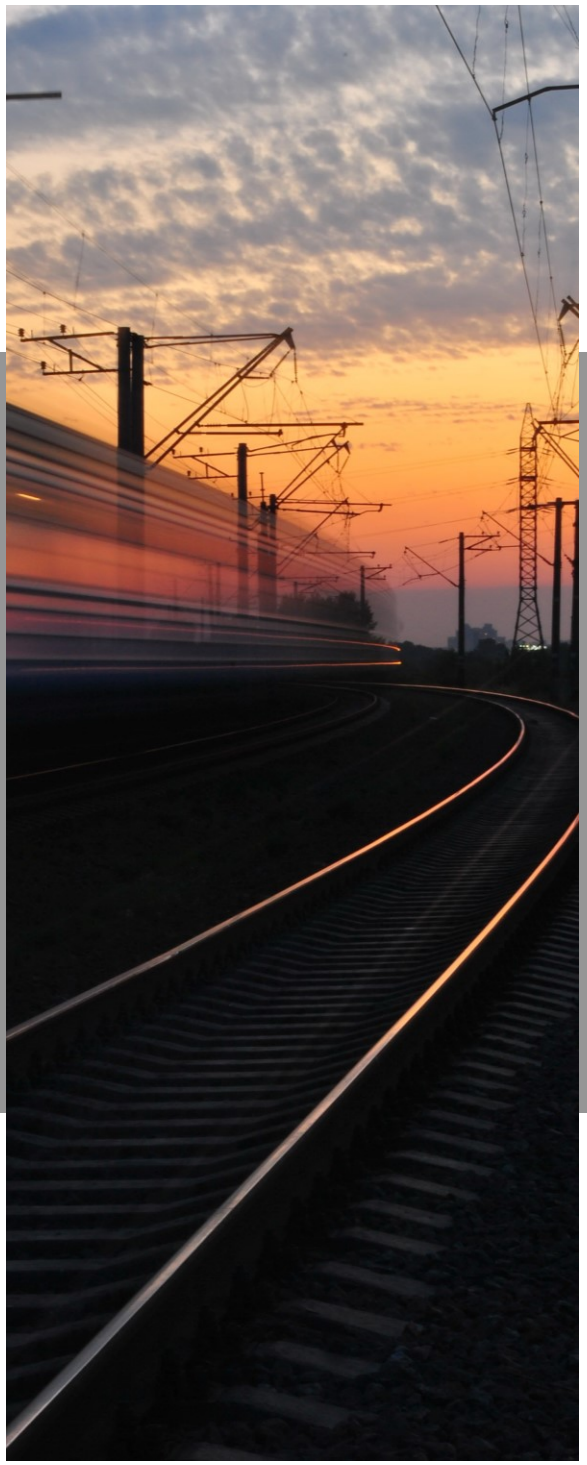
PROGRAMMING CODE FOR TRANSMITTER

```
#include <RCSwitch.h>

const int btnSelectLinePin   = 6;
const int btnSelectStatusPin = 7;

const int ledNormalLineAPin = 2;
const int ledNormalLineBPin = 3;

const int ledFaultLineAPin  = 4;
const int ledFaultLineBPin  = 5;

int lineState    = 0;
int statusState  = 0;

bool lineAFlag  = true ;
bool lineBFlag  = false ;

bool statusNormalFlag  = true ;
bool statusFaultFlag   = false ;

RCSwitch mySwitch = RCSwitch();

void setup() {

  Serial.begin(115200)
```

```
 pinMode(ledNormalLineAPin  , OUTPUT );
  pinMode(ledNormalLineBPin  , OUTPUT );

  pinMode(ledFaultLineAPin  , OUTPUT );
  pinMode(ledFaultLineBPin  , OUTPUT );

  pinMode(btnSelectLinePin    , INPUT  );
  pinMode(btnSelectStatusPin  , INPUT  );

 //Arduino Uno Transmitter is connected to Arduino
Pin # 10
   mySwitch.enableTransmit;(10)
  lineState    = ;0
  statusState  = ;0
  Serial.println("Hello Trains Transmitter");

{
void loop} ()

  lineState   = digitalRead(btnSelectLinePin);
  statusState = digitalRead(btnSelectStatusPin);
 if ( lineState == HIGH ) {
     Serial.println("lineState Pressed");
     if ( lineAFlag  ) {
         lineAFlag  = false ;
         lineBFlag  = true ;
} else {
         lineAFlag  = true ;
         lineBFlag  = false ;
     }
  }
```

```
if (statusState == HIGH) {
    Serial.println("statusState Pressed");

    if ( statusNormalFlag  ) {
      statusNormalFlag  = false ;
      statusFaultFlag  = true ;
    } else {
      statusNormalFlag  = true ;
      statusFaultFlag  = false ;
    }
 }

 if ( lineAFlag ) {

//    Serial.println("Train on Line A");
   digitalWrite(ledNormalLineAPin    ,HIGH );
   digitalWrite(ledNormalLineBPin    ,LOW );
//    mySwitch.send("A", 24);
//    mySwitch.send(1, 24);

 } else {
//   Serial.println("Train Not Stopped on Line A");
  digitalWrite(ledNormalLineAPin    ,LOW );
 }
if ( lineBFlag ) {
//    Serial.println("Train on Line B");
   digitalWrite(ledNormalLineBPin    ,HIGH );
   digitalWrite(ledNormalLineAPin    ,LOW );
```

```
//     mySwitch.send("A", 24);
//     mySwitch.send(1, 24);
  } else {
//    Serial.println("Train Not Stopped on Line A");
    digitalWrite(ledNormalLineBPin    ,LOW );
  }
  if ( statusFaultFlag ) {

    if ( lineAFlag ) {
        Serial.println("Train Stopped on Line A");
        mySwitch.send(1, 24);
        digitalWrite(ledFaultLineAPin    ,HIGH );
    } else {
        digitalWrite(ledFaultLineAPin    ,LOW );
    }
 if ( lineBFlag ) {
        Serial.println("Train Stopped on Line B");
        mySwitch.send(2, 24);
        digitalWrite(ledFaultLineBPin    ,HIGH );
    } else {
        digitalWrite(ledFaultLineBPin    ,LOW );
    }
  } else {

//    Serial.println("Train Not Stopped on Line B");
    digitalWrite(ledFaultLineAPin    ,LOW );
    digitalWrite(ledFaultLineBPin    ,LOW );
    mySwitch.send(3, 24);
  }
  delay(250); }
```



CHAPTER 3
LOAD CELL TO PROTECT TRAINS AND SLIPWAY

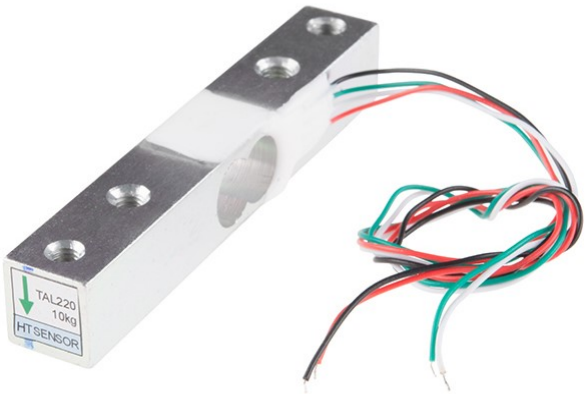
LOAD CELL

A load cell is a type of transducer which is used to convert mechanical force into a measurable electrical output, anything which needs to be weighed probably uses a load cell to do so. Load cells come in many different shapes and sizes so that they can be added to different machinery and weighing equipment.

مستشعر الحمل هو نوع من المحولات التي تستخدم لتحويل القوة الميكانيكية إلى ناتج كهربائي قابل للقياس ، وأي شيء يحتاج إلى وزنه ربما يستخدم مستشعر الوزن للقيام بذلك. يأتى مستشعر الحمل بأشكال وأحجام مختلفة بحيث يمكن إضافتها إلى آلات ومعدات وزن مختلفة

- The feature of the load cell in our project

The cause of Assuit accident was while a hydraulic conveyor was working next to the railway and while crossing the road while the gates were closed and the train crossing the train collided with the carrier, which led to the death of the train driver, his assistant and some people.



From this Accident, we decided to use a weight sensor at the railroad tracks. When it senses a truck, crane, or anything large, it sends a signal to the train informing it that there is something in front of it, so that the driver can stop the train to prevent any accident.

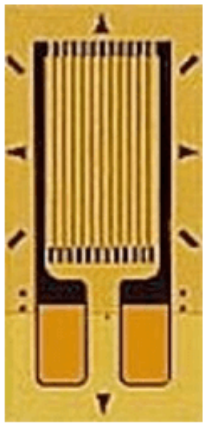
✓ Specifications of the load cell

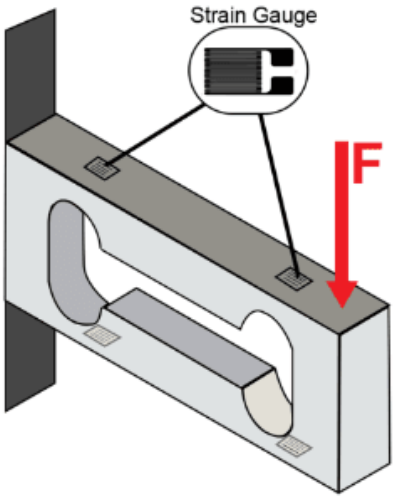
- Rated load: 1000g
- Output: ± 0.05mv / v
- Temperature zero drift: 0.1% F.S
- Output sensitivity: 1.0 ± 0.1 mv / v
- Temperature sensitivity: 0.05% F.S
- Repeatability: 0.03% F.S
- Insulation resistance: ≥ 2000MΩ
- Excitation voltage: 5-10VDC
- Dimensions : 75x12.7x12.7mm

✓ How does a resistive load cell works ?

➢ A load cell is made by using an elastic member (with very highly repeatable deflection pattern) to which a number of strain gauges are attached.

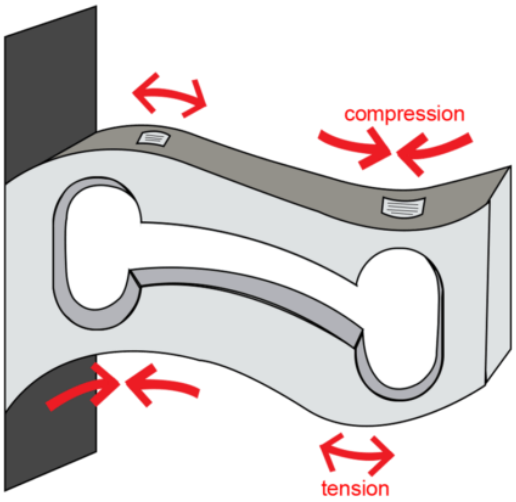
Resistive Load cell Principle



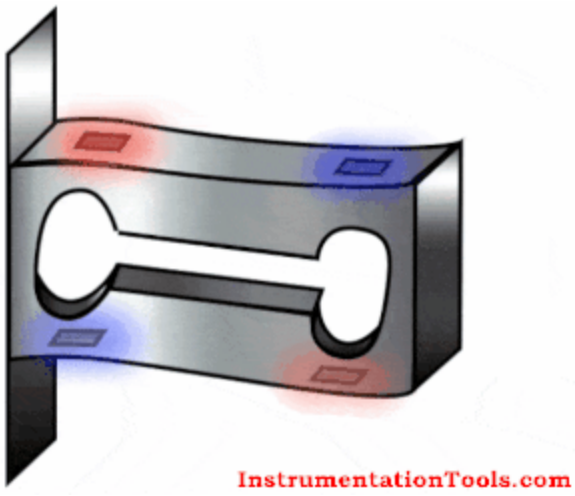


InstrumentationTools.com

- In this particular load cell shown in above figure, there are a total of four strain gauges that are bonded to the upper and lower surfaces of the load cell.



- when the load is applied to the body of a resistive load cell as shown above, the elastic member, deflects as shown and creates a strain at those locations due to the stress applied. As a result, two of the strain gauges are in compression, whereas the other two are in tension as shown in below animation.



- During a measurement, weight acts on the load cell's metal spring element and causes elastic deformation.
- This strain (positive or negative) is converted into an electrical signal by a strain gauge (SG) installed on the spring element. The simplest type of load cell is a bending beam with a strain gauge.
- We use Wheatstone bridge circuit to convert this change in strain/resistance into voltage which is proportional to the load.

➢ Types of load cells :

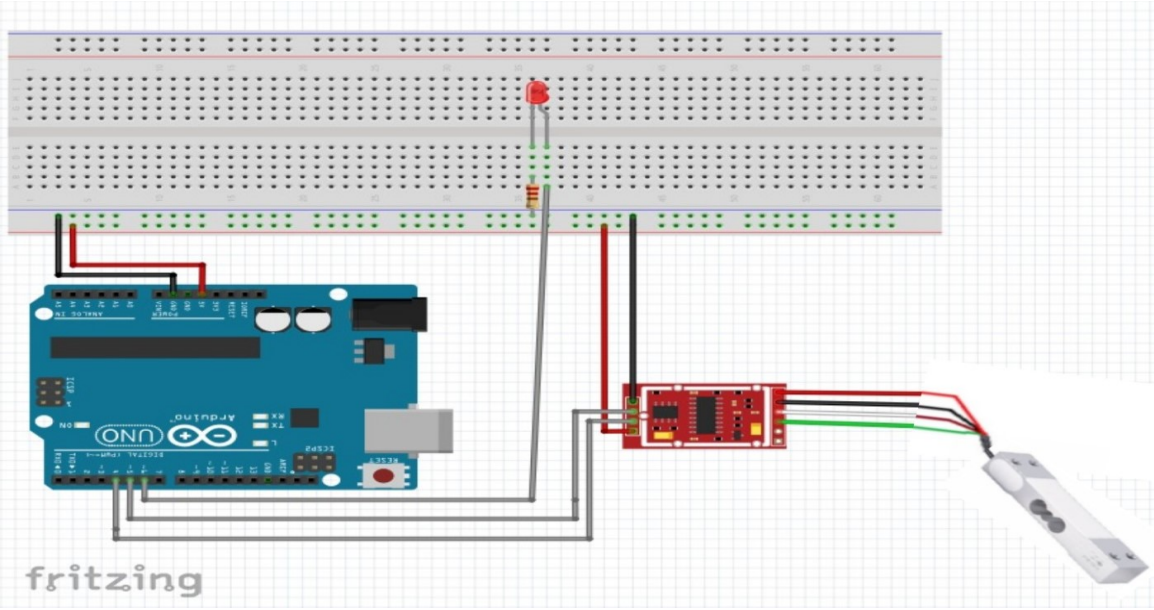
- There are many different types of load cells which have different characteristics for different applications.

- ❑ Single point - Shear beam
- ❑ Compression - Pancake
- ❑ Bending beam load cells - Dual shear beam
- ❑ Bending beam - Miniature

➢ load cells used for- industrial specific applications :

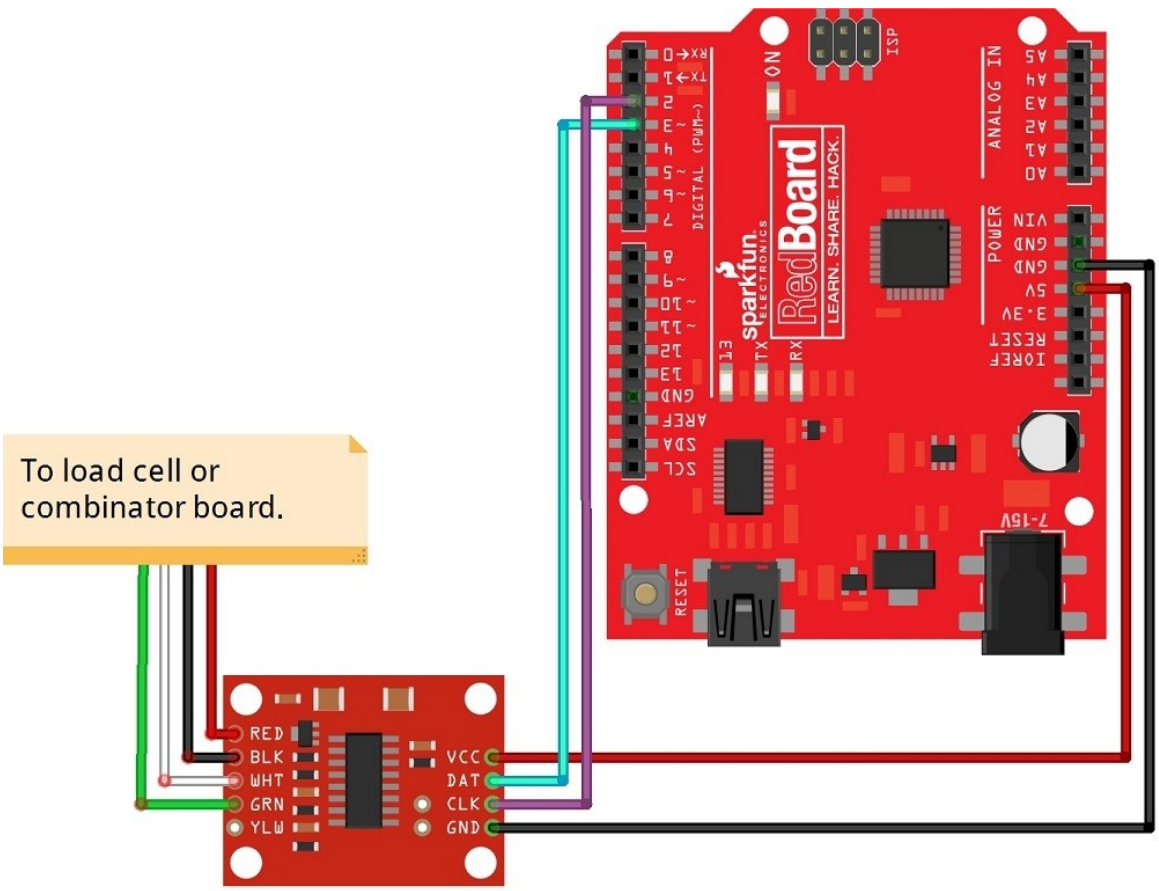
- ❑ On-board weighing
- ❑ Platform weighing
- ❑ Belt scales
- ❑ Overhead track scales
- ❑ Hopper scales
- ❑ Weighing agricultural produce
- ❑ Medical equipment
- ❑ Weighbridges

CONNECTION WITH ARDUINO



LOAD CELL AMPLIFIER

- ✓ A load cell is a transducer or sensor, that converts the kinetic energy of a force into a quantifiable electrical signal. This output signal is very small for strain gauge load cells, in the order of millivolts. On the other hand, most displays need a larger, noise free input to convert that signal for human interpretation. Therefore, the load cell output signal must go through signal conditioners or load cell amplifiers to improve signal quality before use by displays or control systems.



PROGRAMMING CODE

```
#include <HX711_ADC.h>
#include <EEPROM.h>
//pins:
const int HX711_dout = 4; //mcu > HX711 dout pin
const int HX711_sck = 5; //mcu > HX711 sck pin
const int ledPin = 13;
//HX711 constructor:
HX711_ADC LoadCell(HX711_dout, HX711_sck);
const int calVal_eepromAdress = 0;
long t;
int max_weight = 200 ;
void setup() {
  Serial.begin(115200);
  delay(10);
  Serial.println();
  Serial.println("Starting...");
  pinMode( ledPin , OUTPUT );
 LoadCell.begin();
  long stabilizingtime = 2000; // preciscion right after power-up
can be improved by adding a few seconds of stabilizing time
  boolean _tare = false; //set this to false if you don't want tare
to be performed in the next step
  LoadCell.start(stabilizingtime, _tare);
  if (LoadCell.getTareTimeoutFlag() ||
LoadCell.getSignalTimeoutFlag()) {
    Serial.println("Timeout, check MCU>HX711 wiring and pin
designations");
```

```
while (1); }
  else {
    LoadCell.setCalFactor(1.0); // user set calibration value (float),
initial value 1.0 may be used for this sketch
    Serial.println("Startup is complete");
  }
  while (!LoadCell.update());
  calibrate(); //start calibration procedure
}
void loop() {
  static boolean newDataReady = 0;
  const int serialPrintInterval = 0; //increase value to slow down
serial print activity

  // check for new data/start next conversion:
  if (LoadCell.update()) newDataReady = true;
// get smoothed value from the dataset:
if (newDataReady) {
    if (millis() > t + serialPrintInterval) {
      float i = LoadCell.getData();
      Serial.print("Load_cell output val: ");
      Serial.println(i);

      if ( i > max_weight ) {
        Serial.print("Exceeded");
        digitalWrite ( ledPin , HIGH );
      } else {
        digitalWrite ( ledPin , LOW );
      }
```

```
      newDataReady = 0;
      t = millis();
    }
  }
  // receive command from serial terminal
  if (Serial.available() > 0) {
    float i;
    char inByte = Serial.read();
    if (inByte == 't') LoadCell.tareNoDelay(); //tare
    else if (inByte == 'r') calibrate(); //calibrate
    else if (inByte == 'c') changeSavedCalFactor(); //edit
calibration value manually
  } // check if last tare operation is complete
  if (LoadCell.getTareStatus() == true) {
    Serial.println("Tare complete");}}
void calibrate() {
  Serial.println("***");
  Serial.println("Start calibration:");
  Serial.println("Place the load cell an a level stable
surface.");
  Serial.println("Remove any load applied to the load
cell.");
  Serial.println("Send 't' from serial monitor to set the tare
offset.");
  boolean _resume = false;
  while (_resume == false) {
    LoadCell.update();
    if (Serial.available() > 0) {
      if (Serial.available() > 0) {
```

```
float i;
        char inByte = Serial.read();
        if (inByte == 't') LoadCell.tareNoDelay();
      }
    }
    if (LoadCell.getTareStatus() == true) {
      Serial.println("Tare complete");
      _resume = true;
    } }
  Serial.println("Now, place your known mass on the loadcell.");
  Serial.println("Then send the weight of this mass (i.e. 100.0)
from serial monitor.");

  float known_mass = 0;
  _resume = false;
  while (_resume == false) {
    LoadCell.update();
if (Serial.available() > 0) {
      known_mass = Serial.parseFloat();
      if (known_mass != 0) {
        Serial.print("Known mass is: ");
        Serial.println(known_mass);
        _resume = true; }}}
  LoadCell.refreshDataSet(); //refresh the dataset to be sure that
the known mass is measured correct
  float newCalibrationValue =
LoadCell.getNewCalibration(known_mass); //get the new
calibration value
```

```
Serial.print("New calibration value has been set to: ");
  Serial.print(newCalibrationValue);
  Serial.println(", use this as calibration value (calFactor) in your
project sketch.");
  Serial.print("Save this value to EEPROM adress ");
  Serial.print(calVal_eepromAdress);
  Serial.println("? y/n");
  _resume = false;
  while (_resume == false) {
    if (Serial.available() > 0) {
      char inByte = Serial.read();
      if (inByte == 'y') {
#if defined(ESP8266)|| defined(ESP32)
        EEPROM.begin(512);
#endif
        EEPROM.put(calVal_eepromAdress, newCalibrationValue);
#if defined(ESP8266)|| defined(ESP32)
        EEPROM.commit();
#endif
EEPROM.get(calVal_eepromAdress, newCalibrationValue);
        Serial.print("Value ");
        Serial.print(newCalibrationValue);
        Serial.print(" saved to EEPROM address: ");
        Serial.println(calVal_eepromAdress);
        _resume = true;
      }
      else if (inByte == 'n') {
        Serial.println("Value not saved to EEPROM");
        _resume = true;
      }}}
```

```
Serial.println("End calibration");
  Serial.println("***");
  Serial.println("To re-calibrate, send 'r' from serial monitor.");
  Serial.println("For manual edit of the calibration value, send
'c' from serial monitor.");
  Serial.println("***");}
void changeSavedCalFactor() {
  float oldCalibrationValue = LoadCell.getCalFactor();
  boolean _resume = false;
  Serial.println("***");
  Serial.print("Current value is: ");
  Serial.println(oldCalibrationValue);
  Serial.println("Now, send the new value from serial monitor,
i.e. 696.0");
  float newCalibrationValue;
while (_resume == false) {
    if (Serial.available() > 0) {
      newCalibrationValue = Serial.parseFloat();
      if (newCalibrationValue != 0) {
        Serial.print("New calibration value is: ");
        Serial.println(newCalibrationValue);
        LoadCell.setCalFactor(newCalibrationValue);
        _resume = true;
      }
    }
  }
  _resume = false;
  Serial.print("Save this value to EEPROM adress ");
```

```
Serial.print(calVal_eepromAdress);
  Serial.println("? y/n");
  while (_resume == false) {
    if (Serial.available() > 0) {
      char inByte = Serial.read();
      if (inByte == 'y') {
#if defined(ESP8266)|| defined(ESP32)
        EEPROM.begin(512);
#endif
        EEPROM.put(calVal_eepromAdress,
newCalibrationValue);
#if defined(ESP8266)|| defined(ESP32)
        EEPROM.commit();
#endif
        EEPROM.get(calVal_eepromAdress,
newCalibrationValue);
        Serial.print("Value ");
        Serial.print(newCalibrationValue);
        Serial.print(" saved to EEPROM address: ");
        Serial.println(calVal_eepromAdress);
        _resume = true;}
      else if (inByte == 'n') {
        Serial.println("Value not saved to EEPROM");
        _resume = true;
      }
    }
  }
  Serial.println("End change calibration value");
  Serial.println("***");
}
```

✓ CHAPTER 4 NOTIFY THE DRIVER OF THE SEPARATION OF RAIL CARS

CIRCUIT IDEA

- Knowing the communication between the cars and if a car was separated from the train.
- We can control whether the cars on the train are on or off
- By placing a LED in each car, and when the car is separated, the LED is Turned off .

❑ CONNECTION FOR THE LED

- And we control the LED by connecting it to a battery with the train car

❑ The Components For 3 Cars

✓ 6 LED -- 1 battery -- Display for 6 LED

CONNECTION

- Connecting the LED positive to the battery positive and the LED negative to the battery negative from the car .
- Each LED of a car and with the battery.
- Then we connect the LED from the car on the train to the LED that the driver can see it in the driving car, Each LED from the cars is connected to the LED in the display in the driving car via positive with positive and negative .

❑ WHY WE USE IT /

- to reduce accidents and to know more about the cars of the train for the driver . it also facilitates handling and makes it safer for everyone.
- And this to avoid accidents like detachment of train car from Abu Qir train .

RECOMMENDATIONS AND CONCLUSION

- ✓ Passenger and freight trains derailments are sadly more common than we assume. Thousands of people die each year as a result of dozens of significant aberrations around the world. And, because there were numerous accidents last year that resulted in the loss of many lives and the injury of many others, we are working on a plan to reduce these incidents.
- ✓ New rules and regulations have been implemented by governments all over the world in an effort to decrease rail accidents. Implementation of innovative technologies to improve rail administration and safety is one of the new initiatives.
- ✓ Positive Train Control
    - ✓ For this reason, it requires the installation of a new technological technology for positive control of trains, which is called (PTC).
    - ✓ This (PTC) technology must be installed and implemented on the key railway lines where goods and passenger transit activities take place.
    - ✓ These systems must be able to reliably avert accidents (train collisions, excessive speed deflections, incursions into a defined work area, untimely movement of the main line switch).

- ✓ But there are nothing without challenges or obstacles.. This type of system requires very complex techniques capable of analyzing and integrating the huge number of variables that affect train operations, and most importantly, the technology must be able to interpret information automatically and reliably and stop the train safely.

- ✓ Opponents of PTC question whether the technology is better than skilled train operators and engineers with decades of experience. Some even question whether there is a sufficiently advanced and sophisticated technology to do the job effectively and efficiently

Intelligent Databases

Railroad owners and railroad cars take advantage of the information provided by big data to improve railroad safety.

Derailment Detection and Rail Fracture Devices

India has some of the busiest and most widely used railways in the world. It also had some of the worst train accidents. This year alone has seen several major incidents, including a terrorist attack in March where a bomb went off.

- ✓ However, derailment cases remain a huge problem in India, often resulting in a large number of deaths. In November 2016, a passenger train derailed killing 150 and injuring more than 150. It was the deadliest accident in the country since 1999. As a result, the Indian government has invested more resources in developing technologies to help prevent such accidents in the future.

- ✓ Currently, the Indian Institute of Technology Kanpur (IIT Kanpur) is developing skew detectors, which will be in the form of on-board equipment. The device will integrate with existing brake mechanisms to reduce losses from towing a derailed vehicle, according to IIT Kapur. There are currently no devices on the Indian Railways to detect deviation potentials.

- ✓ The Indian Institute of Technology, Madras (IIT Madras) is developing a crack detection system on railway tracks; So far there is no automatic system for this. Currently, rail operators rely on ultrasound tests conducted by engineers every two months to check track conditions, as well as reports from train drivers. IIT Madras hopes to develop a system that digitizes the ultrasound testing process.

Continued Advancements for Safer Railways

As it stands, there are already many systems and technologies in place to help prevent railroad accidents; However, as technology improves, so do these systems. While these systems cannot prevent every accident, nor can they predict accidents caused by spontaneous or planned human events, implementing and improving these systems will certainly save lives and will continue to make railroads safer.

✓ REFERENCES

- https://ar.wikipedia.org/wiki/%D8%B3%D9%83%D9%83_%D8%AD%D8%AF%D9%8A%D8%AF_%D9%85%D8%B5%D8%B1
- https://www.azom.com/article.aspx?ArticleID=1717
- https://ar.wikipedia.org/wiki/%D9%82%D8%A7%D8%A6%D9%85%D8%A9_%D8%AD%D9%88%D8%A7%D8%AF%D8%AB_%D8%B3%D9%83%D9%83_%D8%AD%D8%AF%D9%8A%D8%AF_%D9%85%D8%B5%D8%B1
- https://m.elwatannews.com/news/details/5278142
- Sensors and actuators : Engineering System Instrumentation 2015 by De silva & Clarence W
- https://www.instructables.com/Arduino-Servo-Motors/
- https://www.asmag.com/showpost/24085.aspx