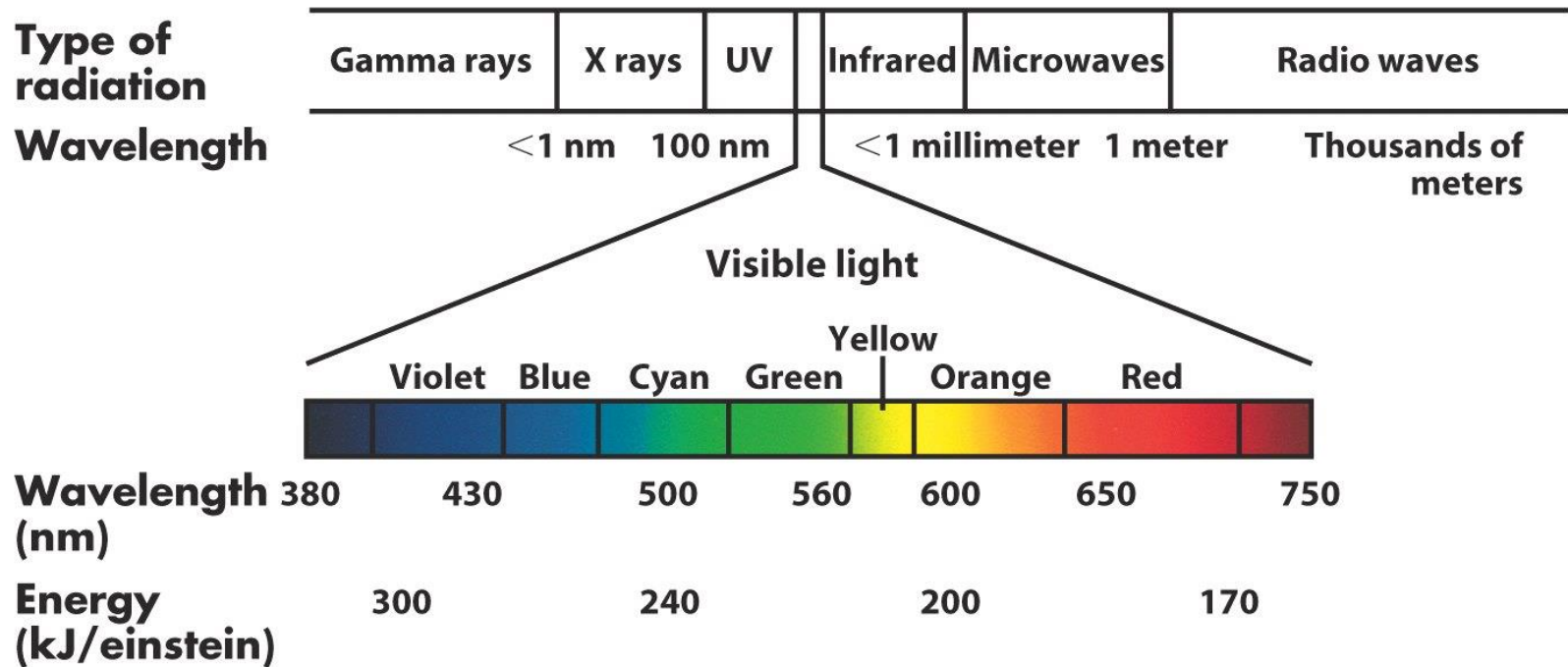


SENSOR INFRARROJO ACTIVO

ALFREDO EL TERS

LUZ INFRARROJA




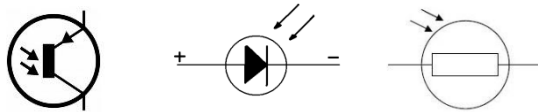
LUZ INFRARROJA

- Filtro polarizado de cámara

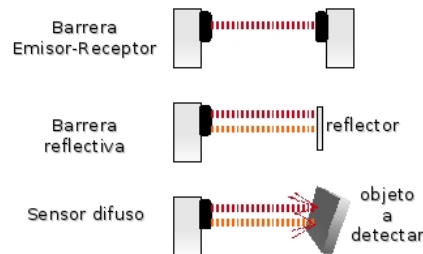


SENSORES INFRARROJO ACTIVO

- Emisor y receptor
- El emisor es un led infrarrojo 
- El receptor puede ser un fototransistor, fotodiodo o fotoresistor infrarrojo.

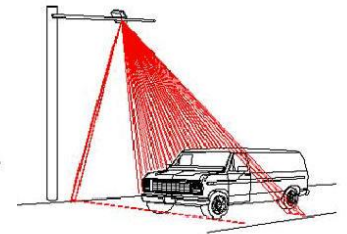
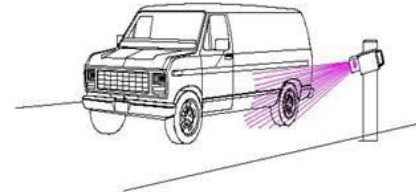
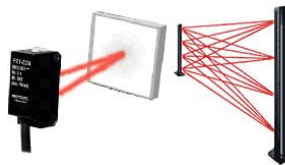
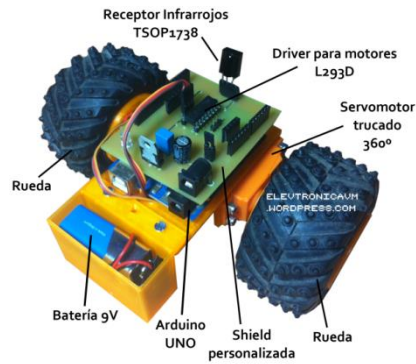


- 940nm normalmente
- Varias configuraciones

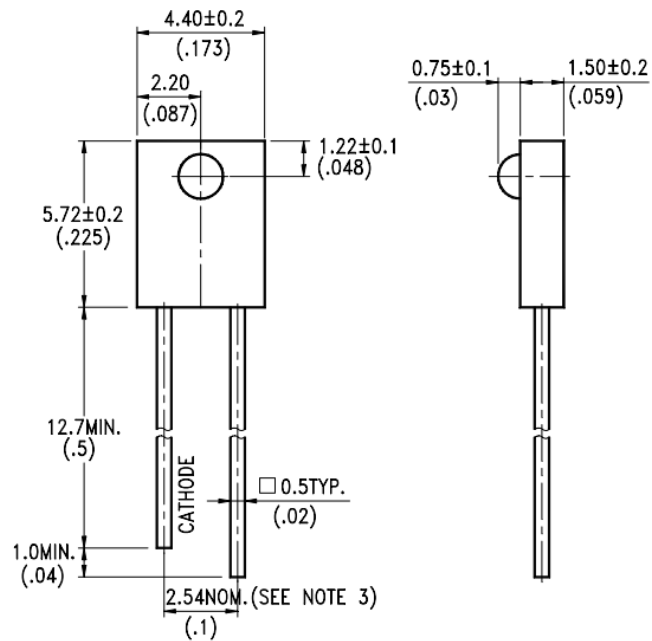


ALGUNOS USOS COTIDIANOS

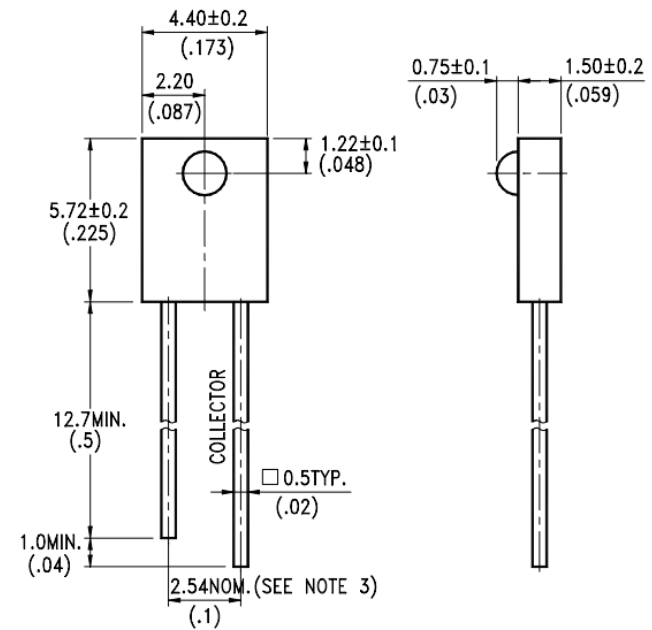
- Controles remotos
- Seguridad (Si bien se usan más los pasivos)
- Domótica
- Robótica
- ¿Carreteras?



EMISOR

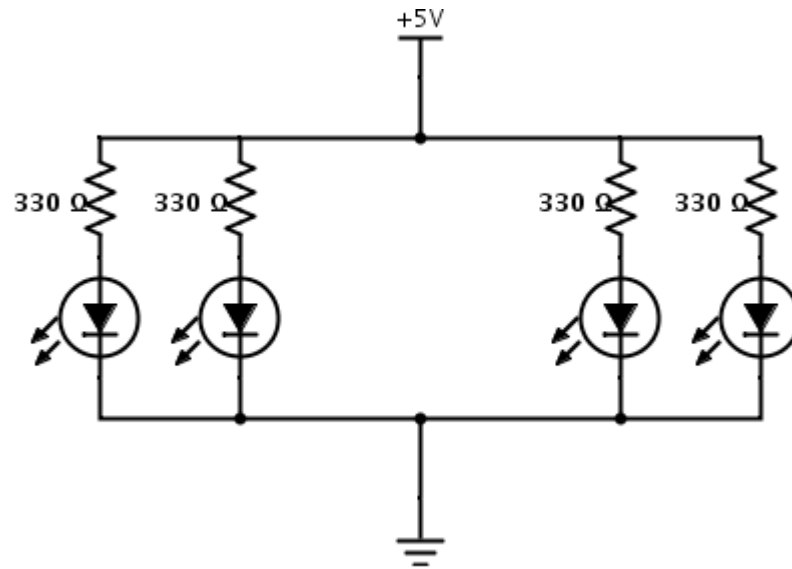


RECEPTOR



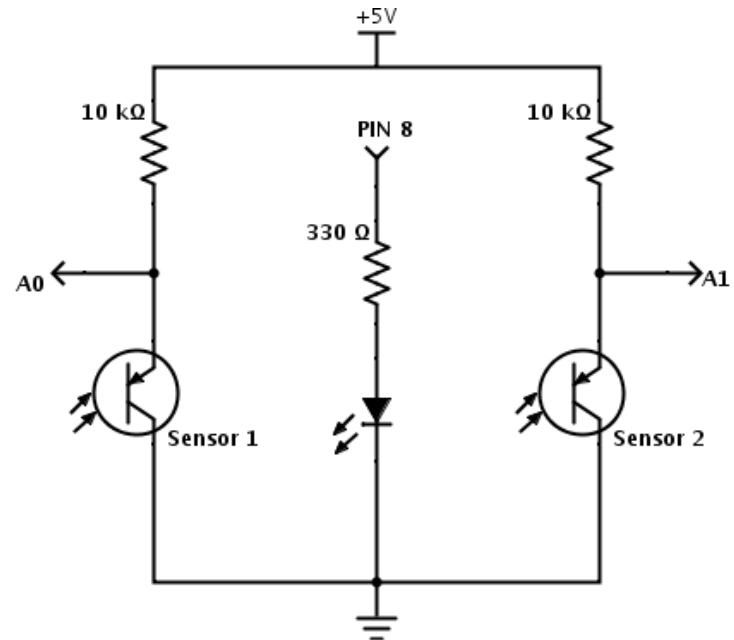
CIRCUITO EMISOR

- 4 leds infrarrojo
- 4 resistencias de $330\ \Omega$
- Cables



CIRCUITO RECEPTOR

- 2 fototransistores infrarrojo
- 2 resistencias de 10 k Ω
- 1 resistencia de 330 Ω
- Cables



CALIBRAR SENSORES

```
calibrar_sensores_laboratorio

int sens1;
int sens2;

void setup()
{
  Serial.begin(9600);
  pinMode (A0, INPUT);
  pinMode (A1, INPUT);
}

void loop()
{
  sens1 = analogRead(A0);
  sens2 = analogRead(A1);
  Serial.println("Primer Sensor");
  Serial.println(sens1);
  Serial.println("Segundo Sensor");
  Serial.println(sens2);
  delay (1000);
}
```

Primer Sensor

827

Segundo Sensor

806

Primer Sensor

827

Segundo Sensor

805

Primer Sensor

986

Segundo Sensor

806

Primer Sensor

992

Segundo Sensor

807

Primer Sensor

827

Segundo Sensor

989

Primer Sensor

828

Segundo Sensor

989

MEDIR VELOCIDADES

medir_velocidad

```
int sens1,sens2;
boolean pasoSensor1, pasoSensor2;
unsigned long tiempol, tiempo2;
double mps,distancia, elap;

void setup()
{
  Serial.begin(9600);
  pinMode (A0, INPUT);
  pinMode (A1, INPUT);
  pinMode (8, OUTPUT);
  pasoSensor1 = false;
  pasoSensor2 = false;
}
```

```
void loop()
{
  while(!pasoSensor1){
    sens1 = analogRead(A0);
    if(sens1>900){
      tiempol = micros();
      pasoSensor1 = true;
    }
  }
  1.04
  0.96
  while(!pasoSensor2){
    sens2 = analogRead(A1);
    if(sens2>900){
      tiempo2 = micros();
      pasoSensor2 = true;
    }
  }
  1.16
  1.77
  0.16
  1.09
  if((tiempo2 - tiempol)>5000){
    elap = (tiempo2 - tiempol)/1000000.0;
    distancia = 6.2/100;
    Serial.println(distancia/elap);
    digitalWrite(8,HIGH);
    delay(200);
    digitalWrite(8,LOW);
  }
  pasoSensor1 = false;
  pasoSensor2 = false;
}
```

PERSISTENCIA DE VELOCIDADES



```
persistirDatosSeriales.py - C:\Users\Alfredo\Documents\Facultad\Proyecto\persistirDatosSeriales.py
File Edit Format Run Options Windows Help

from serial import Serial
import psycpg2
import datetime

def receiving():
    ser = Serial('COM3', 9600) #Establezco la conexión con el puerto
    while True:
        velocidad = ser.readline() #Leo el ultimo serial print que mando el arduino
        print velocidad
        persistir(velocidad) #Persisto el valor obtenido

def persistir(velocidad):
    con = None
    try:
        con = psycpg2.connect(database='LaboratorioProyecto', user='postgres', password = 'root')
        cur = con.cursor()
        now = datetime.datetime.now()
        cur.execute("Insert into velocidades (velocidad, fecha) values(%s,%s);", (velocidad, now.strftime("%Y-%m-%d %H:%M:%S")))
        con.commit()

    except psycpg2.DatabaseError, e:
        print 'Error %s' % e
        sys.exit(1)

    finally:
        if con:
            con.close()

receiving()
```

```
>>>
0.67
0.66
0.61
0.70
0.94
1.59
```

	id [PK] integer	velocidad double precision	fecha timestamp without time zone
1	6	0.72	2014-11-07 10:25:52
2	7	1.5	2014-11-07 10:26:11
3	8	1	2014-11-07 10:26:13
4	9	2.06	2014-11-08 14:06:08
5	10	0.28	2014-11-08 14:06:10
6	11	0.62	2014-11-08 14:06:11
7	12	1.02	2014-11-08 14:06:11
8	98	0.67	2014-11-13 07:53:22
9	99	0.66	2014-11-13 07:53:24
10	100	0.61	2014-11-13 07:53:25
11	101	0.7	2014-11-13 07:53:27
12	102	0.94	2014-11-13 07:53:30
13	103	1.59	2014-11-13 07:53:33
*			

APLICACIÓN WEB



- Java Resources
 - src/main/java
 - com.lab.proy.beans
 - VelocidadesBean.java
 - com.lab.proy.database.daos
 - VelocidadDAO.java
 - com.lab.proy.database.src
 - ConnectionFactory.java
 - DBHelper.java
 - IDBHelper.java
 - com.lab.proy.entities
 - Velocidad.java
 - com.lab.proy.facade
 - FachadaDatos.java
 - com.lab.proy.locale
 - LocaleBean.java
 - com.lab.proy.utils
 - Utils.java

```
public void recargarModelo(){
    velocidades = FachadaDatos.getInstance().selectVelocidades();
    modeloGrafica = new LineChartModel();
    LineChartSeries serie = new LineChartSeries();
    for(Velocidad vel:velocidades){
        serie.set(vel.getFecha(), vel.getVelocidad());
    }
    modeloGrafica.addSeries(serie);
}
```

CONCLUSIONES

- Ventajas:
 - ✓ Pueden detectar presencia, velocidad y distancia.
 - ✓ Sencillos de instalar y leer.
- Desventaja:
 - ❖ Bloqueo de transmisión con casi cualquier objeto.
 - ❖ Corto alcance
 - ❖ Normalmente sensibles a luz y clima



¿DUDAS?

