

Sensores

Un sensor es todo aquello que tiene una propiedad sensible a una magnitud del medio, y al variar esta magnitud también varía con cierta intensidad la propiedad, es decir, manifiesta la presencia de dicha magnitud, y también su medida

Sensores temperatura LM35

```
float leerGradosC() {
  int dato;
  float c;
  //leemos los datos analogicos del sensor
  dato = analogRead(A0);
  //Convertimos el dato de entrada en grados C
  c = (500.0 * dato / 1024);
  return c;
}
```

Sensor ultrasonico HC-SR04

```
int trigPin = 8;
int echoPin = 9;
long duration; //tiempo de ida/vuelta
int cm = 0;
void setup(){
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop(){
```

Sensor ultrasonico HC-SR04 (cont)

```
> digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
duration = duration/2;
cm=duration/29;
Serial.print("Distancia: ");
Serial.println(cm);
delay(100);
}
```

Bluetooth

Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2.4 GHz.

Internet de las cosas

El internet de las cosas es un concepto que se refiere a una interconexión digital de objetos cotidianos con internet.

Ejemplo IoT con sensor de temperatura

```
#include <Ethernet.h>
#include <SPI.h>
// Configuración del Ethernet Shield
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFF, 0xEE}; // Dirección MAC
byte ip[] = { 192,168,1,100 };
// Dirección IP del Arduino
byte server[] = { 192,168,1,6 }; // Dirección IP del servidor
EthernetClient client;
float temperatura;
int analog_pin = 0;
void setup( void ) {
  Ethernet.begin(mac, ip);
  // Inicializamos el Ethernet Shield
  delay(1000); // Esperamos 1 segundo de cortesía
}
void loop(void) {
  temperatura = analogRead(analog_pin);
  temperatura = (temperatura * 5.0) / 1024.0;
  // Display in Serial Monitor
  Serial.print(temperatura); // Return temperature to Monitor
  Serial.println(" °C");
  // Proceso de envío de muestras al servidor
  Serial.println(" Conectando...");
```

Ejemplo IoT con sensor de temperatura (cont)

```
> if (client.connect(server, 80)>0) { //
Conexion con el servidor
  client.print("GET /tutoiot/iot.php?valor=");
// Enviamos los datos por GET
  client.print(temperatura);
  client.println(" HTTP/1.0");
  client.println("User-Agent: Arduino 1.0");
  client.println();
  Serial.println("Conectado");
} else {
  Serial.println("Fallo en la conexion");
}
if (!client.connected()) {
  Serial.println("Disconnected!");
}
client.stop();
client.flush();
delay(60000); // Espero un minuto antes
de tomar otra muestra
}
```

Sensor IR LM393 (detector obstaculos)

```
const int sensorPin = 9;

void setup() {
  Serial.begin(9600);
//iniciar puerto serie
  pinMode(sensorPin,
INPUT); //definir pin como
entrada
}
```

Sensor IR LM393 (detector obstaculos) (cont)

```
> void loop(){
  int value = 0;
  value = digitalRead(sensorPin); //lectura
digital de pin

  if (value == HIGH) {
    Serial.println("Detectado obstaculo");
  }
  delay(1000);
}
```

Sensor de humedad FC-28

```
const int sensorPin = 10;

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

void loop()
{
  int humedad = digitalRead(
sensorPin);

  //mandar mensaje a puerto
serie en función del valor leído
  if (humedad == HIGH)
  {
    Serial.println(" Encendido ");
    //aquí se ejecutarían las acciones
  }
  delay(1000);
}
```

Sensor de humedad FC-28 (cont)

```
> }
```

Fotoreistencia LDR

```
int valorLDR = 0;
int pinLDR = A0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  valorLDR = analogRead(pinLDR);
  Serial.println(valorLDR);
}
```

Pasos para IoT en Arduino

- * Conseguir una Ethernet shield compatible con Arduino uno.
- * Crear una base de datos.
- * Programar un servidor.
- * Programar la tarjeta Arduino.

Datos

Nombre	Alvarado Anaya Adrian Javier
Matriculo	1253317
Fecha	23 de Mayo de 2020

Sensor de movimiento PIR

```
const int LEDPin= 13;
const int PIRPin= 2;

void setup()
{
    pinMode(LEDPin, OUTPUT);
    pinMode(PIRPin, INPUT);
}

void loop()
{
    int value= digitalRead(PIRPin);

    if (value == HIGH)
    {
        digitalWrite(LEDPin, HIGH);
        delay(50);
        digitalWrite(LEDPin, LOW);
        delay(50);
    }
    else
    {
        digitalWrite(LEDPin, LOW);
    }
}
```

Potenciómetro

```
const int analogPin = A0;
int value; //variable que
almacena la lectura analógica
raw

int position; //posicion del
potenciómetro en tanto por
ciento

void setup() {

}

void loop() {
    value = analogRead(analogPin); // realizar la
    lectura analógica raw
    position = map(value, 0,
    1023, 0, 100); // convertir a
    porcentaje

    //...hacer lo que se
    quiera, con el valor de posición
    medido

    delay(1000);
}
```

Conexion Bluetooth (cont)

```
> char option = Serial.read();
//si la opcion esta entre '1' y '9'
if (option >= '1' && option <= '9')
{
    //restamos el valor '0' para obtener el
    numero enviado
    option -= '0';
    for(int i=0;i<option;i++){
        digitalWrite(led, HIGH);
        delay(100);
        digitalWrite(led, LOW);
        delay(200);
    }
}
}
```

Conexion Bluetooth

```
const int led = 13;

int option;

void setup(){
    Serial.begin(9600);
    pinMode(led, OUTPUT);
}

void loop(){
    //si existe información
    pendiente
    if (Serial.available() > 0){
        //leemos la opcion
```