

Datos

Nombre: SOBERANIS
CEDILLO BRYAN FERNANDO

Matrícula: 1253408

Fecha: 25/05/2020

Sensor

Dispositivo que capta magnitudes físicas (variaciones de luz, temperatura, sonido, etc.) u otras alteraciones de su entorno.

Sensor de temperatura LM35

```
float leerGradosC(){
  int dato;
  float c;
  //leemos los datos
  analógicos 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(){
  digitalWrite(trigPin,
  LOW);
  delayMicroseconds(2);
```

Sensor Ultrasonico HC-SR04 (cont)

```
digitalwrite(trigPin,
HIGH);
delayMicroseconds(10);
digitalwrite(trigPin,
LOW);
duration = pulseIn(e-
choPin, HIGH);
duration = duration/2;
cm=duration/29;
Serial.print("Distancia:
");
Serial.println(cm);
delay(100);
}
```

Sensor IR LM393 (detector de obstáculos)

```
const int sensorPin = 9;
void setup(){
  Serial.begin(9600);
  //iniciar puerto serie
  pinMode(sensorPin,
  INPUT); //definir pin
  como entrada
}
void loop(){
  int value = 0;
  value = digitalRead(sen-
  sorPin); //lectura
  digital de pin
  if (value==HIGH){
  Serial.println("se
  detecto el obstáculo");
  }
  delay(1000);
}
```

Sensor de humedad FC-28

```
const int sensorPin =
10;
void setup()
{
  Serial.begin(9600);
  pinMode(sensorPin,-
  INPUT);
}
void loop()
{
  int humedad = digitalRe-
  ad(sensorPin);
  //mandar mensaje a
  puerto serie en función
  del valor leído
  if (humedad == HIGH)
  {
  Serial.println("encend-
  ido");
  //aquí se ejecutarían
  las acciones
  }
  delay(1000);
}
```

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);
```

Sensor de movimiento PIR (cont)

```
delay(50);
digitalWrite(LEDPin,
LOW);
delay(50);
}
else
{
digitalWrite(LEDPin,
LOW);
}
}
```

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.

Conexión Bluetooth

```
const int led = 13;
int option;
void setup(){
  Serial.begin(9600);
  pinMode(led, OUTPUT);
}
void loop(){
  //si existe información
  pendiente
  if (serial.availab-
  le(>0){
  //leemos la opción
  char option = Serial.re-
  ad();
  {
```

Conexión Bluetooth (cont)

```
//si la opción 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<opt-
ion;i++){
digitalWrite(led, HIGH);
delay(100);
digitalWrite(led, LOW);
delay(200);
}
}
}
```

Internet de las cosas

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

Pasos para LoT en Arduino

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

Ejemplo IoT

```
*include <Ethernet.h>
#include <SPI.h>
//configuración del
ethernet shield
byte mac[] = {exDE,
exAD, exBE, exEF, exFF,
exEE}; //dirección MAC
byte ip[] = {192,168,-
1,100}; // Dirección IP
del Arduino
byte server[] = {192,1-
68,1,6}; // 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 = analogRea-
d(analog_pin);
temperatura = S.0temper-
atura100.0/1024.0;
//Display in serial
monitor
Serial.print(temperat-
ura); //Return temper-
ature to Monitor
Serial.println( "oC");
//Proceso de envío de
muestras al servidor
Serial.println("Connect-
ing...");
if (client.connect(se-
rver, s0)>0){
```

Ejemplo IoT (cont)

```
// conexión con el
servidor
client.print("GET
/tutoiot/iot.php?valo-
r="); //Enviamos los
datos por GET
client.print(temperat-
ura);
client.println("HTT-
P/1.0");
client.println("User-A-
gent: Arduino 1.0");
client.println();
Serial.println("Conect-
ado");
} else {
Serial.println("fallo en
la conexión");
}
if (!client.connected())
{
Serial.println("Discon-
nected");
}
client.stop();
client.flush();
delay(6000); //espero un
minuto antes de tomar
otra muestra
}
```