


# GUIA DE INICIO

**Gracias por adquirir nuestro producto, este manual te ayudara a configurar tu dispositivo.**

1-En una PC ingresa a la pagina oficial de Arduino para poder modificar el número de contacto.

<https://www.arduino.cc/en/software/>, Descarga la última versión de Arduino IDE y asegúrate en descargar la versión disponible para tu sistema operativo.



## Arduino IDE 2.3.6

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

**SOURCE CODE**

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

### DOWNLOAD OPTIONS

**Windows** Win 10 and newer, 64 bits

**Windows** MSI installer

**Windows** ZIP file

**Linux** AppImage 64 bits (X86-64)

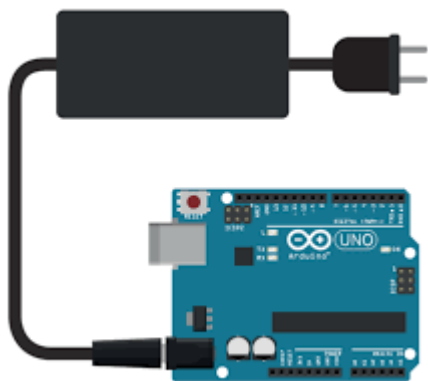
**Linux** ZIP file 64 bits (X86-64)

**macOS** Intel, 10.15: "Catalina" or newer, 64 bits

**macOS** Apple Silicon, 11: "Big Sur" or newer, 64 bits

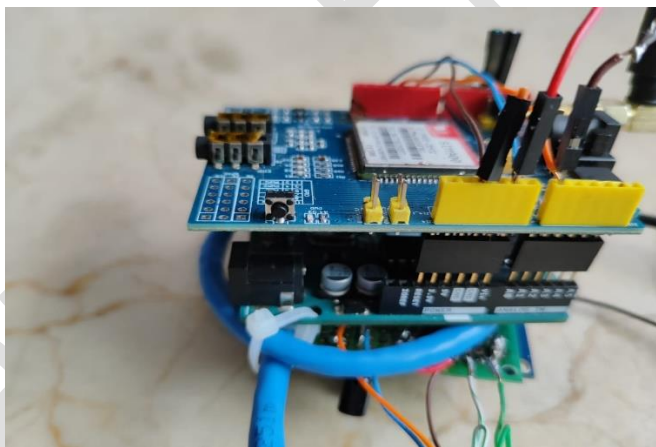
[Release Notes](#)

2-Conecta el producto “Haz Paro” con su cable incluido a la energía eléctrica, o con la batería de 9V/2A.

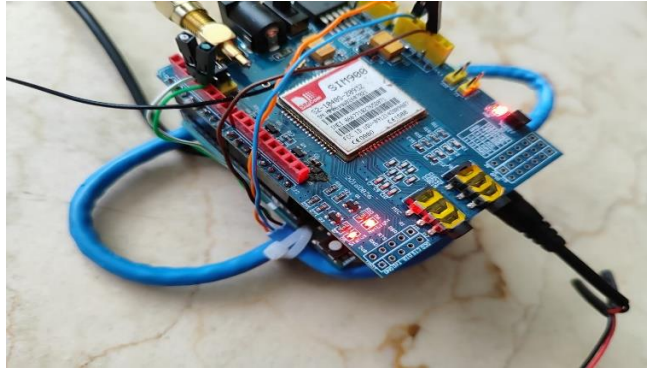


3- Coloca la tarjeta SIM en el modulo GSM

4-Enciende el modulo GSM presionando el boton lateral.



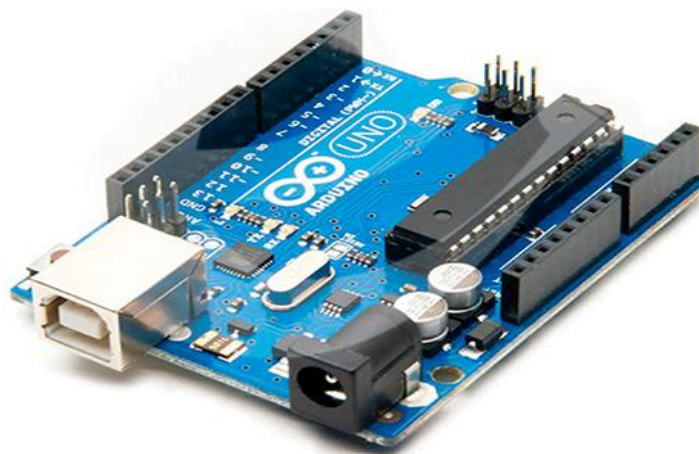
(si un led rojo enciende cada segundo el modulo GSM no tiene señal, pero si prende cada 3 segundos ya esta conectado), asegurarse que el modulo GPS encienda un led azul.



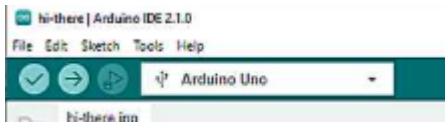
EN CASO DE CUALQUIER PROBLEMA QUE TUVIERA CON EL PRODUCTO CONTACTAR A SERVICIO AL CLIENTE PARA DAR UNA SOLUCIÓN.

SERVICIO AL CLIENTE: 3313047483 (Horario 24H)

5-Ya instalado ARDUINO IDE en tu pc, conecta el producto HAZ PARO mediante un cable de datos a la computadora.



6-Al conectarlo asegúrate que este conectado al ARDUINO UNO R3 en la parte superior izquierda.



7-Copea el siguiente código en el programa:

```
#include <TinyGPS++.h> // Librería para decodificar datos NMEA del GPS
#include <SoftwareSerial.h> // Permite comunicación serial en pines digitales

// Configuración de puertos seriales por software
SoftwareSerial SIM900(7, 8); // RX=7, TX=8 para módulo GSM
static const int RXPin = 4, TXPin = 3; // Pines para GPS (RX=4, TX=3)
TinyGPSPlus gps; // Objeto principal para procesar datos GPS
SoftwareSerial serialGPS(RXPin, TXPin); // Puerto serial virtual para GPS

// Variables de control
int boton = A0; // Pin analógico A0 conectado al botón
int apagado = LOW; // Estado anterior del botón (para detección de flanco)

void setup() {
  pinMode(boton, INPUT); // Configura el pin del botón como entrada digital

  // Inicialización de comunicaciones seriales:
  SIM900.begin(19200); // Módulo GSM a 19200 baudios
  Serial.begin(19200); // Monitor serial para depuración
  serialGPS.begin(9600); // GPS a 9600 baudios (velocidad estándar NMEA)
}

void loop() {
  // Procesamiento continuo de datos GPS:
```

```
while (serialGPS.available() > 0) {  
    gps.encode(serialGPS.read()); // Decodifica cada byte recibido del GPS  
}
```

```
// Lógica del botón (detección de flanco ascendente):
```

```
int presionado = digitalRead(boton);
```

```
if (presionado != apagado && presionado == HIGH) {
```

```
    Serial.println("Verificando ubicación GPS...");
```

```
    delay(1000);
```

```
    mostrarDatosGPS(); // Muestra información en monitor serial
```

```
    delay(1000);
```

```
    if (gps.location.isValid()) { // Si hay coordenadas válidas
```

```
        Serial.println("Enviando posicion!!!");
```

```
        delay(100);
```

```
        EnvioTexto(); // Envía SMS con ubicación
```

```
    } else {
```

```
        Serial.println("Ubicación no localizada.");
```

```
    }
```

```
}
```

```
apagado = presionado; // Actualiza estado anterior del botón
```

```
delay(1000);
```

```
}
```

```
// =====
```

```
// Función: mostrarDatosGPS()
```

```
// Descripción: Muestra toda la información disponible del GPS
```

```
// =====
```

```
void mostrarDatosGPS() {  
    // Sección 1: Coordenadas geográficas  
    Serial.print("Latitud: ");  
    Serial.print(gps.location.lat(), 6); // 6 decimales (~11 cm precisión)  
    Serial.print(" | Longitud: ");  
    Serial.println(gps.location.lng(), 6);  
  
    // Sección 2: Altitud  
    Serial.print("Altitud: ");  
    if (gps.altitude.isValid()) {  
        Serial.print(gps.altitude.meters());  
        Serial.println(" metros");  
    } else {  
        Serial.println("No disponible");  
    }  
  
    // Sección 3: Velocidad  
    Serial.print("Velocidad: ");  
    if (gps.speed.isValid()) {  
        Serial.print(gps.speed.kmph());  
        Serial.println(" km/h");  
    } else {  
        Serial.println("No disponible");  
    }  
  
    // Sección 4: Fecha UTC original  
    Serial.print("Fecha: ");  
    if (gps.date.isValid()) {  
        Serial.print(gps.date.day());
```

```
Serial.print("/");  
Serial.print(gps.date.month());  
Serial.print("/");  
Serial.println(gps.date.year());  
} else {  
    Serial.println("No disponible.");  
}
```

```
// Sección 5: Fecha ajustada a zona horaria de México (UTC-6)
```

```
Serial.print("Fecha (México UTC-6): ");
```

```
if (gps.date.isValid()) {  
    int dia = gps.date.day();  
    int mes = gps.date.month();  
    int ano = gps.date.year();
```

```
// Ajuste de fecha cuando la hora UTC es <6:00 AM (en México aún es día anterior)
```

```
if (gps.time.isValid() && gps.time.hour() < 6) {
```

```
    dia--;
```

```
    if (dia < 1) { // Manejo de cambio de mes
```

```
        mes--;
```

```
        if (mes < 1) { // Manejo de cambio de año
```

```
            mes = 12;
```

```
            ano--;
```

```
        }
```

```
        dia = 31; // Simplificación (debería usar tabla de días por mes)
```

```
    }
```

```
}
```

```
Serial.print(dia);
```

```
Serial.print("/");
```

```
Serial.print(mes);  
Serial.print("/");  
Serial.println(ano);  
} else {  
    Serial.println("No disponible.");  
}
```

```
// Sección 6: Horas (UTC y local)
```

```
if (gps.time.isValid()) {  
    int horaUTC = gps.time.hour();  
    int horaLocal = horaUTC - 6; // Ajuste para UTC-6 (México)  
    if (horaLocal < 0) horaLocal += 24; // Corrección si pasa de medianoche
```

```
Serial.print("Hora UTC: ");  
Serial.print(horaUTC);  
Serial.print(":");  
Serial.print(gps.time.minute());  
Serial.print(":");  
Serial.println(gps.time.second());
```

```
Serial.print("Hora México: ");  
Serial.print(horaLocal);  
Serial.print(":");  
Serial.print(gps.time.minute());  
Serial.print(":");  
Serial.println(gps.time.second());  
} else {  
    Serial.println("Hora no localizada.");  
}
```



```
    delay(2000);
}

// =====
// Función: EnvioTexto()
// Descripción: Envía SMS con ubicación Google Maps
// =====

void EnvioTexto() {
    // Configura módulo GSM en modo texto SMS
    SIM900.print("AT+CMGF=1\r");
    delay(500);

    // Establece número destino (cambiar por el número deseado)
    SIM900.println("AT+CMGS=\"3315376846\"");
    delay(500);

    // Construye enlace a Google Maps con coordenadas
    SIM900.print("https://maps.google.com/maps?q=");
    SIM900.print(gps.location.lat(), 6); // Latitud con 6 decimales
    SIM900.print("+");
    SIM900.println(gps.location.lng(), 6); // Longitud con 6 decimales

    // Mensaje de auxilio
    SIM900.print("Necesito ayuda!! Soy Cristian y esta es mi ubicacion.");
    delay(1000);

    // Envía Ctrl+Z (ASCII 26) para finalizar SMS
    SIM900.println((char)26);
    delay(5000);
}
```

```
Serial.println("Mensaje enviado."); // Confirmación por serial  
}
```

8-Busca la línea 158 en el código del Arduino para ingresar tu número de teléfono para que el producto HAZ PARO Funcione.

```
153 | // Establece número destino (cambiar por el número deseado)  
154 | SIM900.println("AT+CMGS=\""3315376846\"");  
155 | delay(500);
```

Listo, ya puede ser utilizado el producto para cuando se presione el botón mande información crucial a tu contacto de emergencia.