

MODULO SIM900 GSM / GPRS P/ARDUINO

OKY2223-2



Productos evaluados por **ingenieros calificados**



Garantía y seguridad en cada producto



Experiencia de compra en la **calidad** como sello distintivo

Descripción

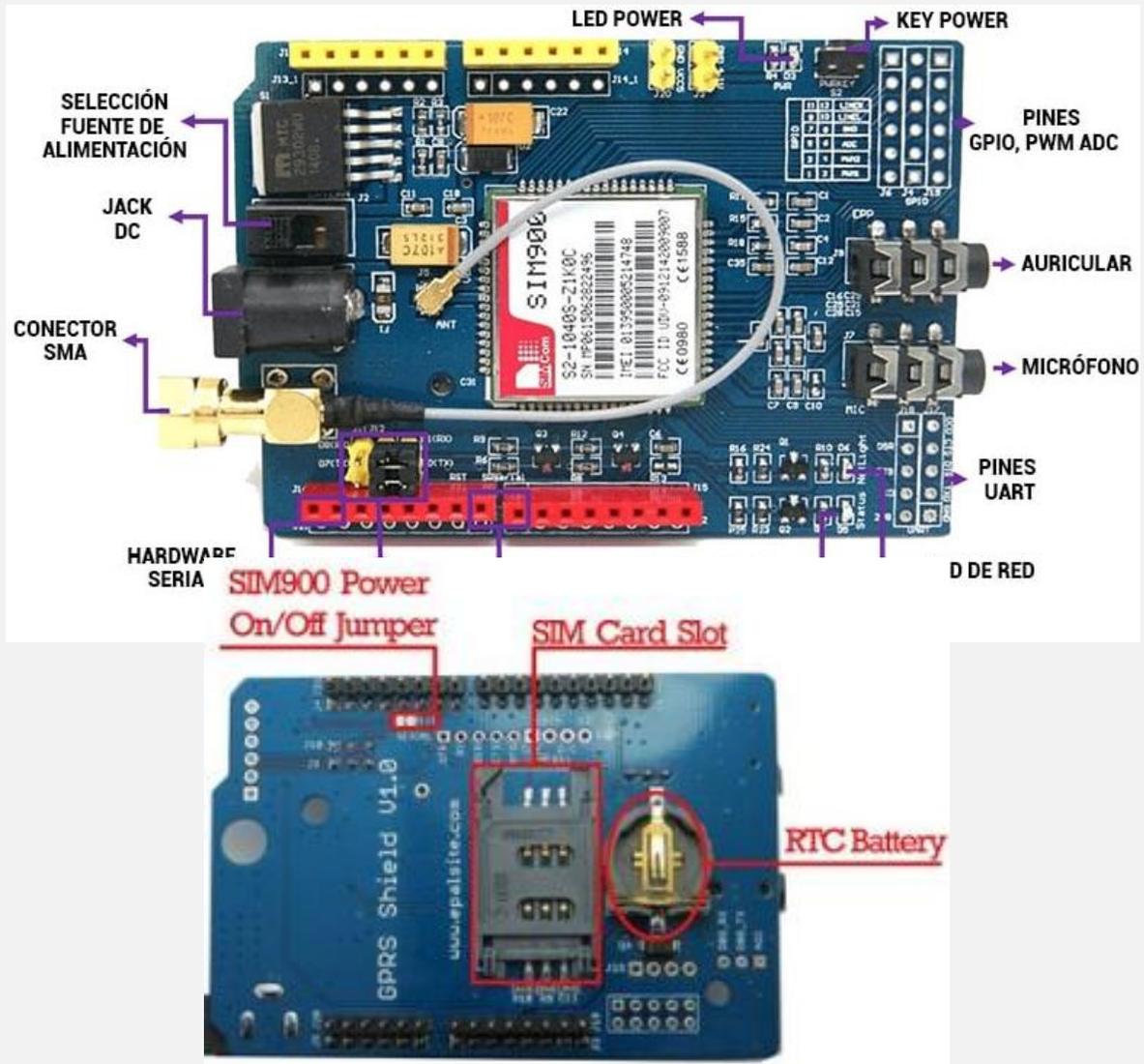
El SIM900 es una tarjeta GSM /GPRS de comunicación inalámbrica. La tarjeta es compatible con todos los modelos de Arduino UNO, además de poder ser controlada con otros microcontroladores también. Puede trabajar en frecuencias GSM/GPRS de 850/900/1800/1900 MHz, para realizar llamadas de voz, envío de SMS. Se controla y configura mediante protocolo UART, usando comandos AT. Por default la velocidad UART está establecida en 19200 baudios.

Características

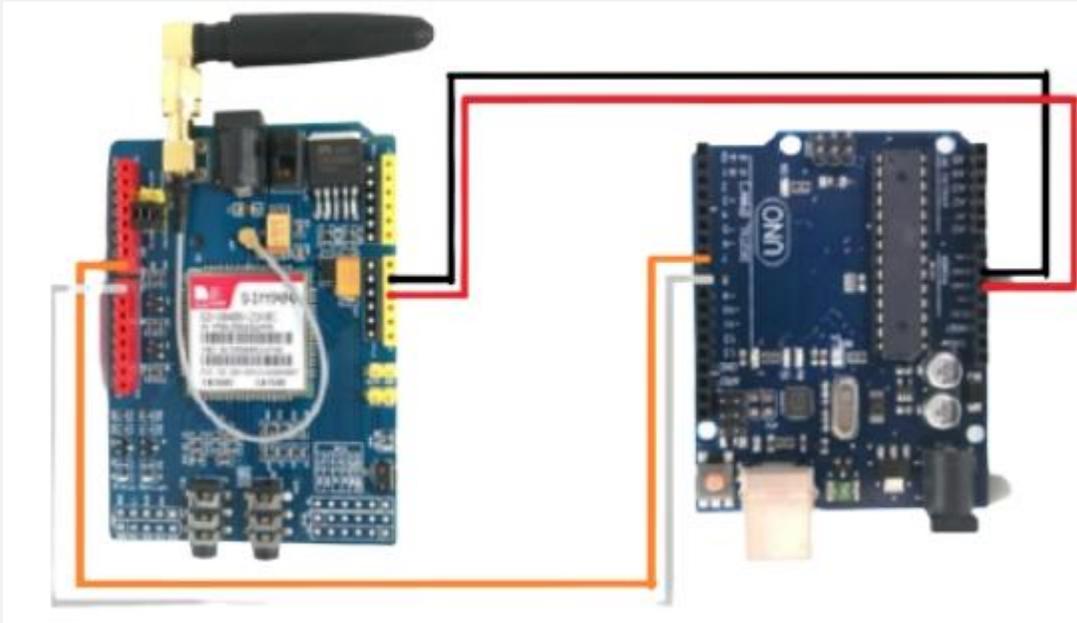
- Voltaje de alimentación: 5 -10 V.
- Corriente: 1.5 mA
- Comunicación: UART
- Bandas de frecuencia: 850/900/1800/1900MHz
- Estación móvil: GPRS Clase B
- Fase: GSM 2/2+
- Clase 1 1W a: 1800/1900 MHz
- Clase 4 2W a: 850/900 MHz
- Servicio de mensajería
- Pila embebida TCP/UDP: Carga de datos a un servidor web
- Puerto serie: Libre selección
- Altavoz y tomas de auriculares: 2 conectores Jack 3.5 mm
- Temperatura de operación mínima: -40 °C

- Temperatura de operación máxima: 85 °C
- Dimensiones: 75 mm X 55 mm X 10 mm

Asignación de pines



1. Programa y conexión para enviar un mensaje de texto a un teléfono celular por medio de la SIM900 y Arduino

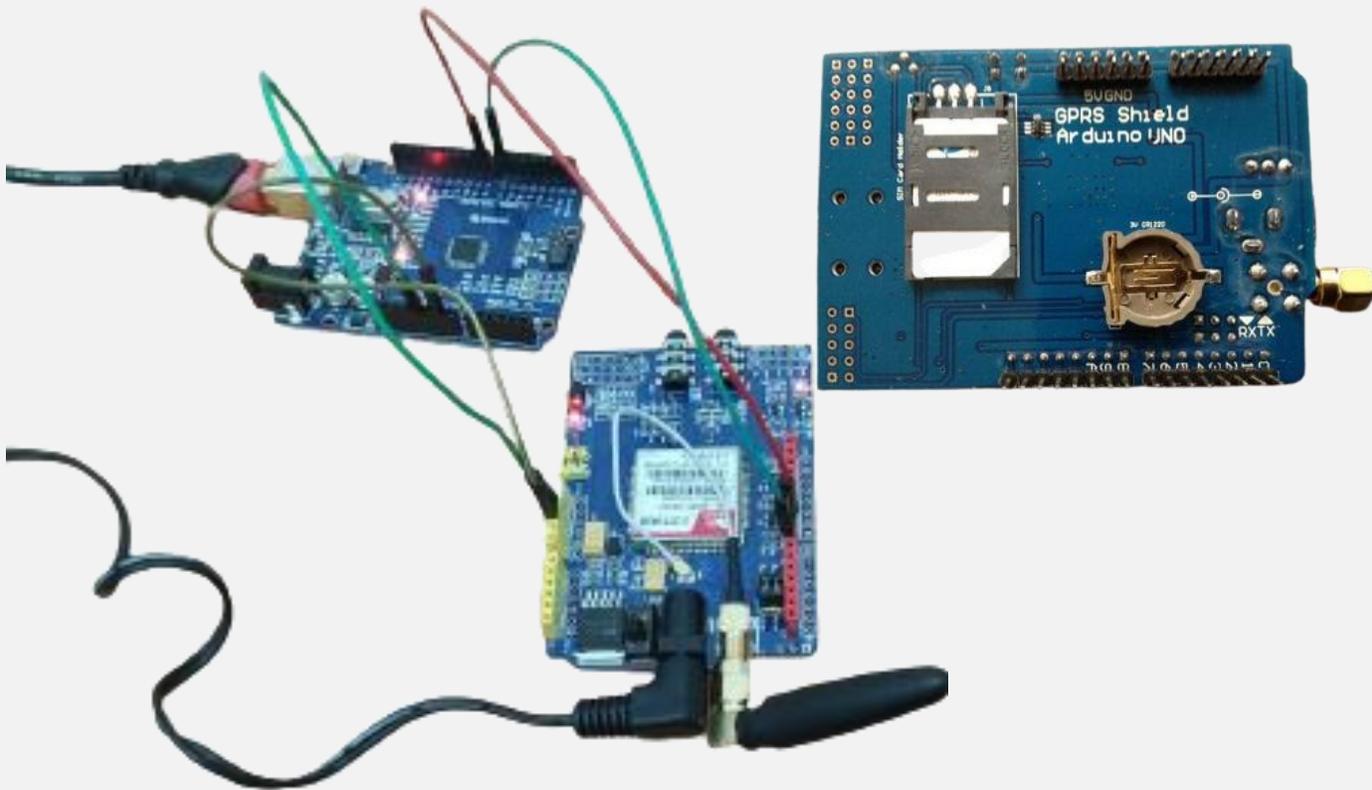


Componentes

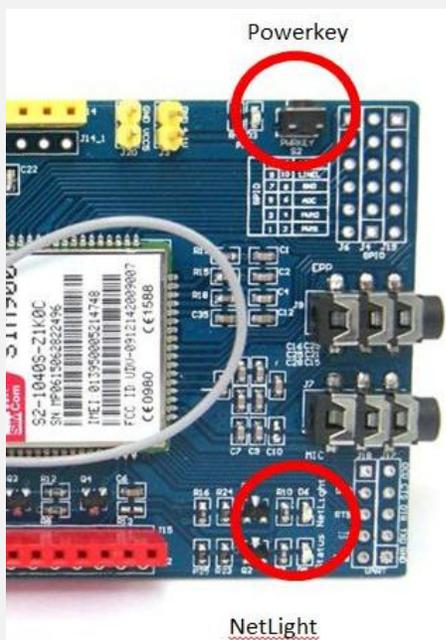
- Tarjeta Arduino UNO
- SIM900

Para programar el SIM900 te recomendamos hacerlo con el IDE de Arduino ya que solo se utiliza la librería SoftwareSerial para comunicarnos con el módulo mediante comando reservados para que se envíen mensajes SMS, realizar llamadas o se conecte el módulo a Internet, estos comandos se llaman AT.

1. Conectar los pines de pines de arduino TX y RX con los RX y Tx del SIM900,conectamos el USB de Arduino a la computadora y tambien usamos un cargador externo de 5V que va al Jack de carga del SIM900, después colocar cuidadosamente el “SIM” en la parte trasera del Modulo SIM900.



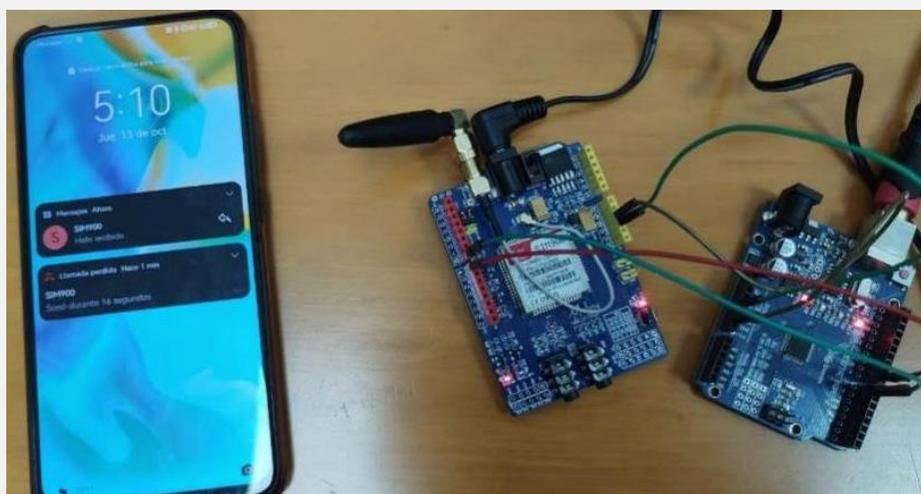
2. Encender el módulo SIM900 con el interruptor de Powerkey presionando el pulsador durante uno o dos segundos y se encenderán los LEDs NetLight (leds de estado y de conexión a red). Después el led de conexión empezara a parpadear una vez por segundo, este parpadeo indica que está buscando la red. Cuando la encuentre y se conecte cambiará la frecuencia de parpadeo y lo hará cada 3 o 4 segundos. Así comprobamos que se está detectando el acceso a la red de la tarjeta SIM que se le colocó.



3. Por último compilamos el código en el IDE de Arduino Uno, abre el monitor serie y selecciona 9600 baudios.

```
#include <SoftwareSerial.h> //Se incluye la librería de "SoftwareSerial"
SoftwareSerial sim900(7, 8);
void setup() {
  sim900.begin(9600); // velocidad en baudios del módulo SIM900
  Serial.begin(9600);
}
void loop() {
  Serial.println("Enviado sms...");
  sim900.print("AT+CMGF=1\r"); // comando AT para envió de sms
  delay(1000);
  sim900.print("AT+CNMI=2,2,0,0,0\r");
  delay(1000);
  sim900.println("AT + CMGS = \"55XXXXXXX\"); //cambiar el número telefonico al que
  desee enviar
  delay(1000);
  sim900.println("Holis AG");
  delay(1000);
  sim900.println((char)26);
  delay(1000);
  sim900.println();
  delay(500);
  Serial.println("SMS Enviado...");
}
```

Pruebas realizadas



```

llamada
void loop()
{
  if (Serial.available() > 0) // Verifico si hay datos en el puerto serie de Arduino.
  {
    switch (Serial.read())
    {
      case 'a': // Si hay algo en el monitor serie, si es una letra "a" se ejecutara los comando AT que se definen en la función "envio_sms"
      envio_sms();
      break;
      case 'c': // Pero si una letra "c" se ejecutara los comando AT que se definen en la función "hacer_llamada"
      hacer_llamada();
      break;
    }
  }
  if (sim900.available() > 0)
  Serial.write(sim900.read()); // Imprime los datos recopilados del puerto serie
}

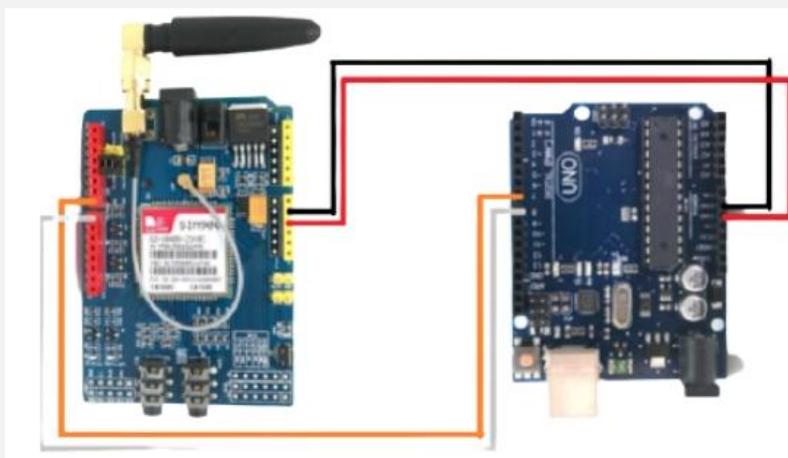
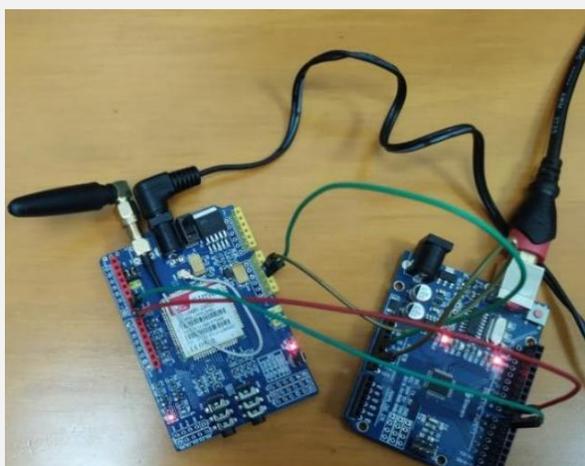
void envio_sms()
{
  Serial.println("Enviado sms...");
  sim900.print("AT+CMGF=1\r"); // Establecer en modo envío de sms
  delay(1000);
  /* De la asigna el numero, debes cambiar la clave de cuenta al país al área de México
  solo agrega el numero al que enviaras el sms */
  sim900.print("AT+CMMS=2,2,0,0,S\r");
  delay(1000);
  sim900.println("Hola"); // Contenido que llevara el sms
  sim900.println("AT + CMMS = '553'");
  delay(1000);
  sim900.println("Hola recibido");
  delay(1000);
  sim900.print("\r");
}

```



2. Programa y conexión para realizar una llamada a un teléfono celular por medio de la SIM900 y Arduino

Utilizamos la misma conexión que se menciono anteriormente y podemos hacer un nuevo código o agregarlo con un if



Código

```
#include <SoftwareSerial.h>
SoftwareSerial sim900(7, 8);
void setup() {
  sim900.begin(9600);
  Serial.begin(9600);
}
void loop() {
void hacer_llamada()
{
  Serial.println("Realizando llamada...");
  sim900.println("ATD5537489655;"); //Comando AT para realizar una llamada
  delay(20000); // Espera 30 segundos mientras realiza la llamada
  sim900.println("ATH"); // Cuelga la llamada
  delay(1000);
  Serial.println("Llamada finalizada");
}
}
```

Código con if para opción de llamada y SMS

```
#include <SoftwareSerial.h>
SoftwareSerial sim900(7, 8);
void setup() {
  sim900.begin(9600);
  Serial.begin(9600);
}

void loop() {
  if (Serial.available()>0)
    switch(Serial.read())
    {
  case 's':
envio_sms(); //para enviar SMS
break;
  case 'c': //para realizar llamada
hacer_llamada();
break;
}
  if (sim900.available()>0)
  Serial.write(sim900.read());
}

void envio_sms()
{
("Llamada finalizada");
}
```

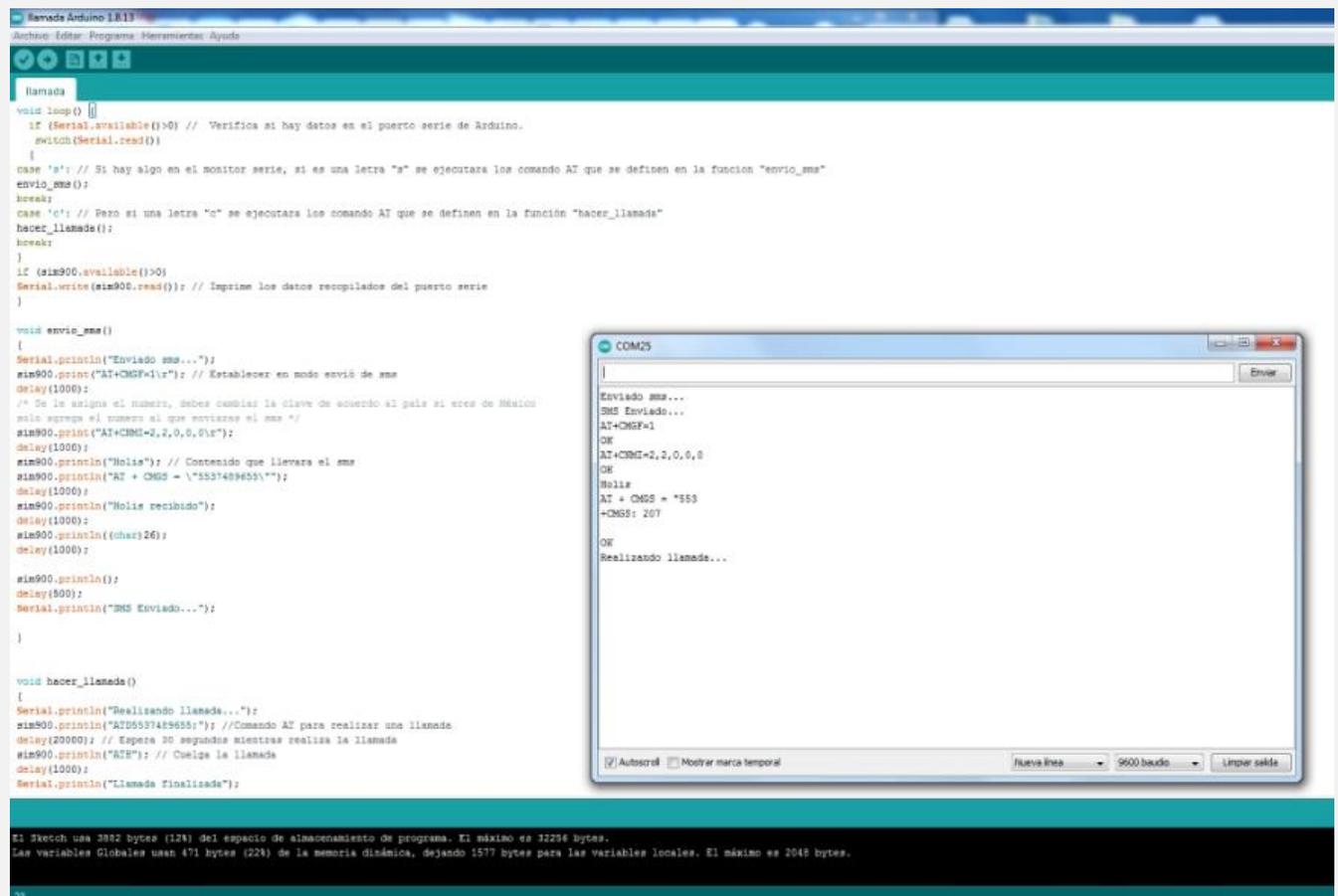
```

Serial.println("Enviado sms...");
sim900.print("AT+CMGF=1\r");
delay(1000);
sim900.print("AT+CNMI=2,2,0,0,0\r");
delay(1000);
sim900.println("AT + CMGS = \"5537xxxx\"");
delay(1000);
sim900.println("Holis AG");
delay(1000);
sim900.println((char)26);
delay(1000);

sim900.println();
delay(500);
Serial.println("SMS Enviado...");
}
void hacer_llamada()
{
Serial.println("Realizando llamada...");
sim900.println("ATD5537489655");
delay(20000); // Espera 30 segundos mientras realiza la llamada
sim900.println("ATH"); // Cuelga la llamada
delay(1000);
Serial.println

```

Pruebas



The screenshot shows the Arduino IDE interface with the code from the previous block and the serial monitor output. The serial monitor displays the following sequence of messages:

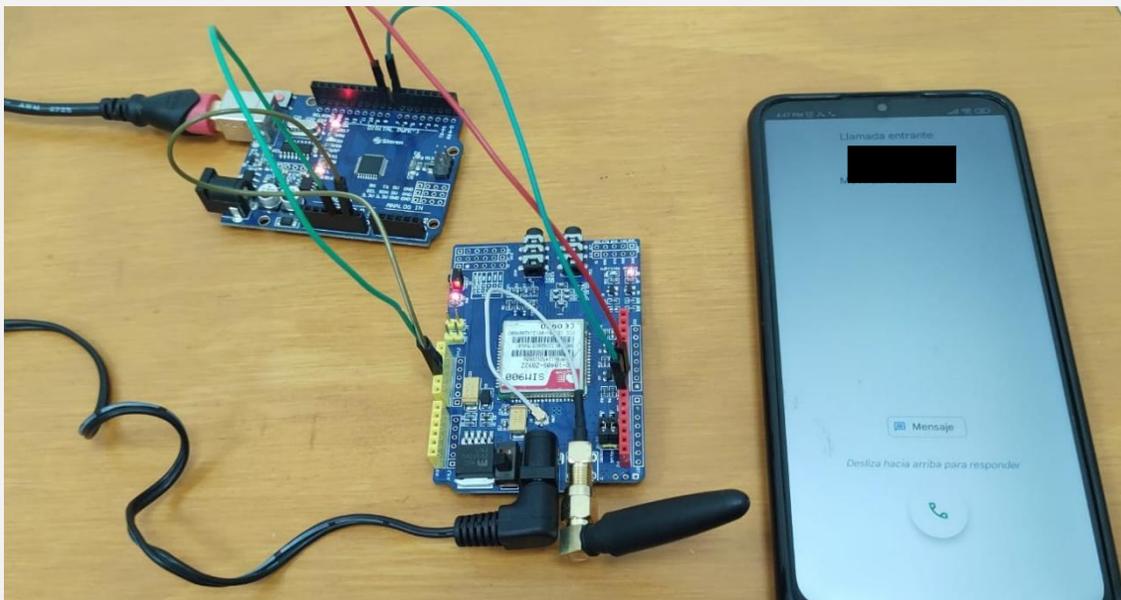
```

Enviado sms...
SMS Enviado...
AT+CMGF=1
OK
AT+CNMI=2,2,0,0,0
OK
Holis
AT + CMGS = "553
+CMGS: 207
OK
Realizando llamada...

```

At the bottom of the IDE, there is a status bar indicating memory usage: "El sketch usa 3982 bytes (12%) del espacio de almacenamiento de programa. El máximo es 32768 bytes. Las variables Globales usan 471 bytes (22%) de la memoria dinámica, dejando 1577 bytes para las variables locales. El máximo es 2048 bytes."

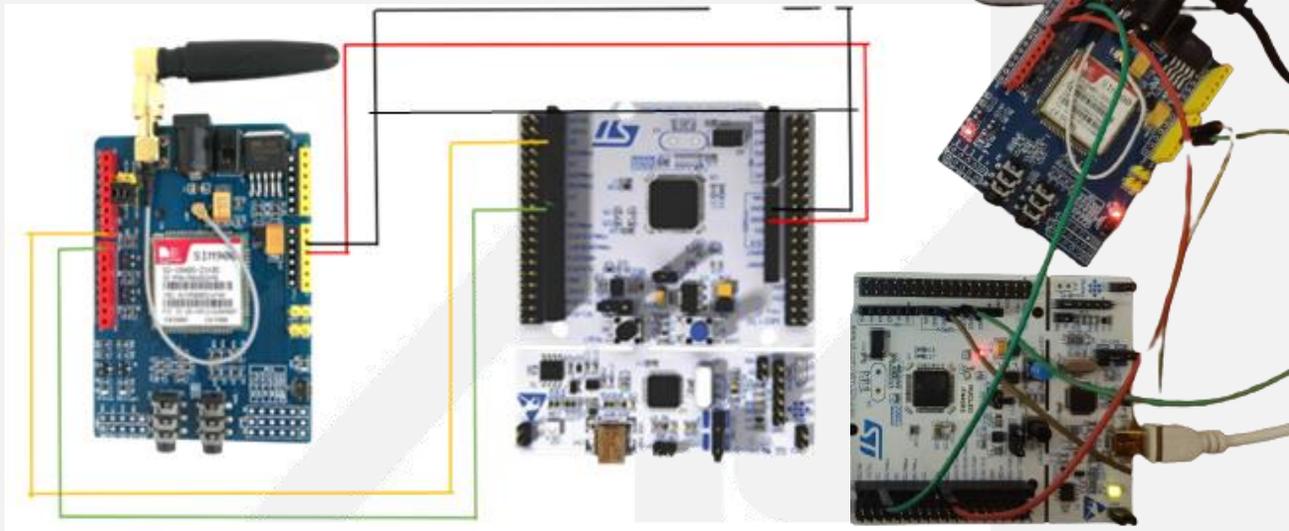
Telefono 1 y Telefono 2



3. Programa y conexión para realizar una llamada a un telefono celular y un mensaje de texto por medio de la SIM900 y STM32F446RE

Comó se menciona se puede utilizar con distintos microcontroladores , se hizo la conexión y pruebas con el Núcleo 64 STM32F446RE

Conexión:



Código:

```
#include <SoftwareSerial.h>
SoftwareSerial sim900(7, 8);
void setup() {
  sim900.begin(19200);
  Serial.begin(19200);
}
void loop() {
  if (Serial.available()>0)
    switch(Serial.read())
    {
  case 's': // "s" se ejecutara los comando AT que se definen en la funcion "envio_sms"
    envio_sms();
    break;
  case 'c': // "c" se ejecutara los comando AT que se definen en la función "hacer_llamada"
    hacer_llamada();
    break;
  }
  if (sim900.available()>0)
    Serial.write(sim900.read());}
void envio_sms()
{
  Serial.println("Enviado sms...");
  sim900.print("AT+CMGF=1\r"); /
  delay(100);
  sim900.print("AT+CMGS=\"+5255.....\"\r");
  delay(200);
  sim900.print("Hola, AG SIM900");
  sim900.print("\r");
  delay(500);
  sim900.print((char)26);
  delay(100);
  sim900.println();
  Serial.println("SMS Enviado...");
```

```

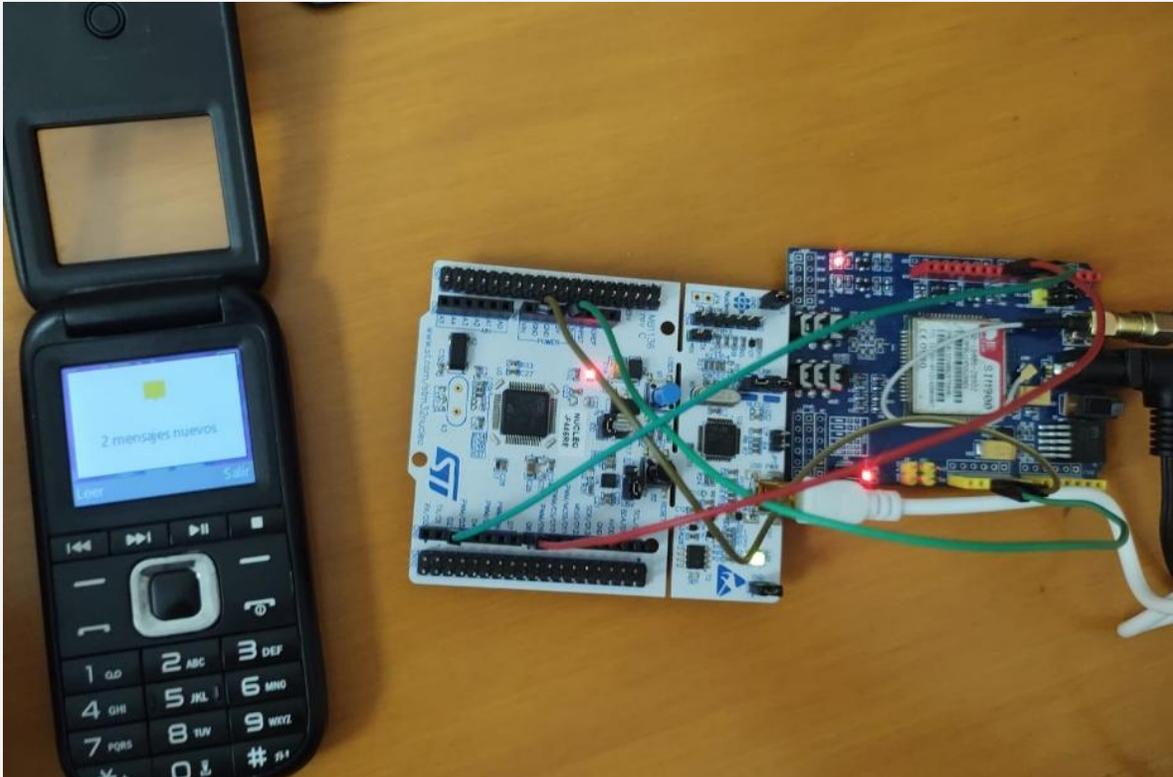
delay(500);
}
void hacer_llamada()
{
sim900.println("ATD+52155.....;");
delay(100);
sim900.println();
}

```

Pruebas: Llamada



Mensaje:



ANEXOS

Como anexo se incluye en código que sirve para comprobar si estas teniendo conexión con los comandos AT

```

oct13a Arduino 1.8.13
PRODUCTOS Transistores
Archivo Editar Programa Herramientas Ayuda
oct13a
#include <SoftwareSerial.h> //Se incluye la siguiente libreria para permitir la comunicación serial
SoftwareSerial SIM900(7, 8); //Seleccionamos los pines 0 como (Rx) y 1 como (Tx)

void setup()
{
  SIM900.begin(19200); //Habilitamos el sim900 a 19200 bps
  Serial.begin(19200); //Habilitamos el puerto serie de Arduino IDE a 19200 bps
  delay(1000); //Esperamos 1 segundo
  //delay(25000);
}

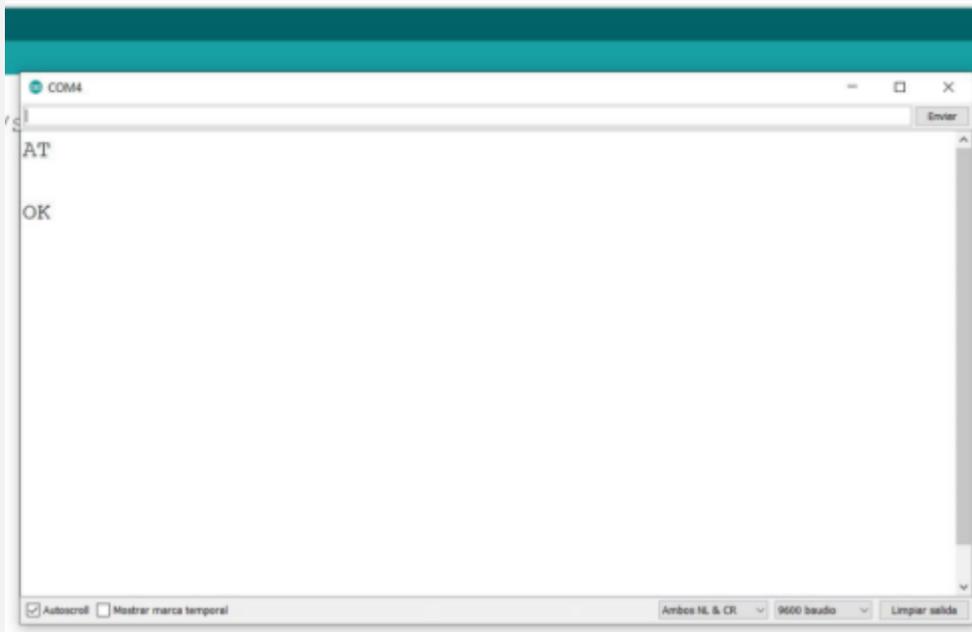
void loop()
{
  //Enviamos y recibimos datos
  if (Serial.available() > 0)
  SIM900.write(Serial.read());
  if (SIM900.available() > 0)
  Serial.write(SIM900.read());
}

El Sketch usa 3348 bytes (10%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables Globales usan 301 bytes (1%) de la memoria dinámica, dejando 1747 bytes para las variables lo
13 Arduino Uno en COM25
  
```

Código:

```
#include <SoftwareSerial.h> //Se incluye la siguiente librería para permitir la comunicación
serial
SoftwareSerial SIM900(0, 1); //Seleccionamos los pines 7,8 comoRX y TX
void setup()
{
  SIM900.begin(19200); //Habilitamos el sim900 mini a 9600 bps
  Serial.begin(19200); //Habilitamos el puerto serie de Arduino IDE a 9600 bps
  delay(1000); //Esperamos 1 segundo
  //delay(25000);
}
void loop()
{
  //Envíanos y recibimos datos
  if (Serial.available() > 0)
  SIM900.write(Serial.read());
  if (SIM900.available() > 0)
  Serial.write(SIM900.read());
}
```

Abrimos monitor serie y tecleamos "AT" nos debe devolver un OK



ANEXO 2: se incluyen algunos comandos AT que se pueden utilizar en aplicaciones para SIM900

a) Enviar SMS:

AT+CMGF=1

AT+CMGS="número de teléfono"

Leer SMS: AT+CMGR=número de mensaje

Borrar SMS: AT+CMGD=número de mensaje

Comandos del servicio de red:

b) Llamada : "ATD+ número de teléfono"

c) AT+CSQ: Obtener calidad de la señal

d) Comandos para conexión a internet

AT+CGDCONT=1,"IP","gprs.movistar.com.ar"

AT+CSTT="gprs.movistar.com.ar"

AT+CIICR<CR> Inicia la conexión

AG Electrónica SAPI de CV República de El Salvador 20 Piso 2, Centro Histórico, Centro, 06000 Ciudad de México, CDMX Teléfono: 55 5130 7210	Realizó	Valeria Zarate	 
	Revisó	Ing. Jessica López Morales	
	Fecha	14/04/2025	