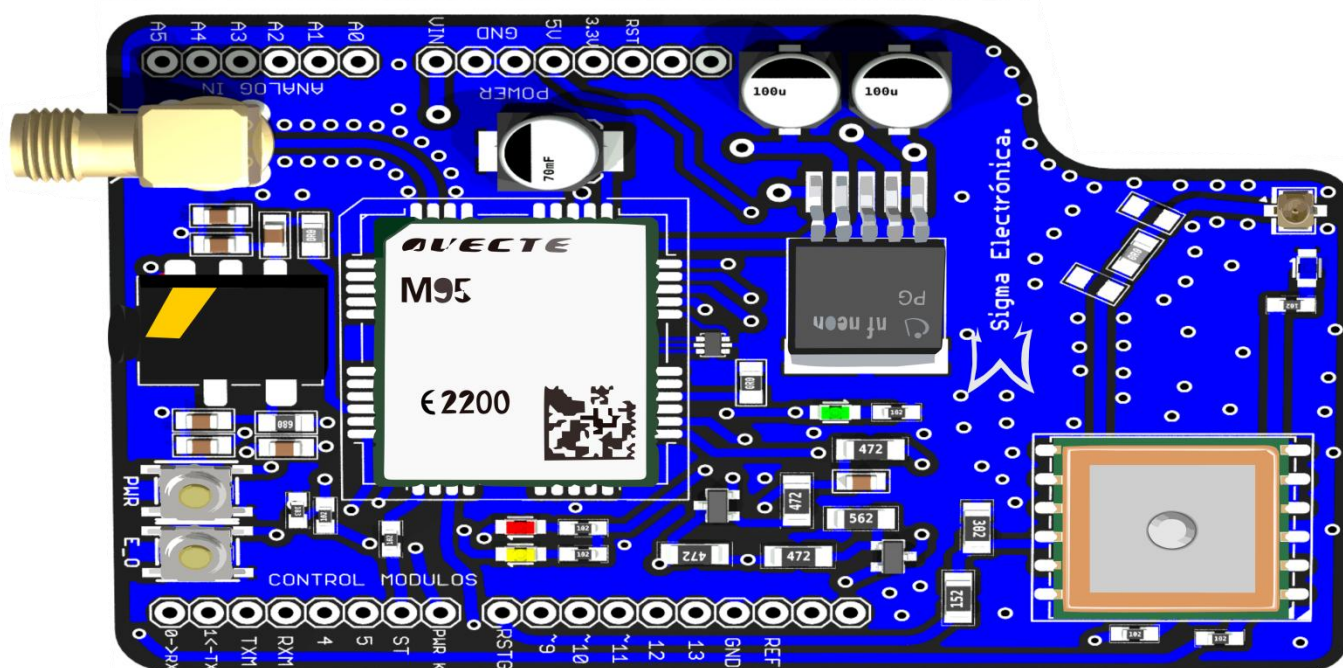


[www.sigmaelectronica.net](http://www.sigmaelectronica.net)

Pbx.3482059

[ventas@sigmaelectronica.net](mailto:ventas@sigmaelectronica.net)

¡ TODO LO QUE NECESITAS PARA DESARROLLAR TUS IDEAS !



**TARJETA M95 L80**  
**ARDUINO**

## TABLA DE CONTENIDO

<b>INTRODUCCIÓN .....</b>	<b>3</b>
<b>DESCRIPCIÓN .....</b>	<b>4</b>
Características Hardware.....	4
<b>ETAPAS Y CARACTERISTICAS ESPECÍFICAS .....</b>	<b>5</b>
1. Entrada de voltaje .....	5
2. Reset .....	5
3. Pines utilizador para Arduino .....	6
4. Conectores de antenas.....	7
5. Socket sim card retráctil .....	9
6. Conector de micrófono y audio.....	9
7. Indicadores.....	10
<b>FUNCIONAMIENTO .....</b>	<b>11</b>
PASO UNO: Conectar el shield a Arduino Uno .....	11
PASO DOS: Realizar la conexión USB y alimentacion del Arduino uno al PC .....	12
PASO CUATRO: Conectamos la antena SMA .....	13
PASO CINCO: Cargar programa de ejemplo .....	13
PASO SEIS: Sentencias de prueba .....	15
<b>WEBGRAFÍA .....</b>	<b>18</b>

## INTRODUCCIÓN

Módulo GPS con antena integrada [L80](#): gracias a las tecnologías EASY (Embedded Assist System) y AlwaysLocate. EASY asegura que el L80 calcule las orbitas automáticamente usando los datos de efemérides (hasta 3 días) almacenados en la memoria flash interna, con esto es capaz de "ubicarse" rápidamente incluso cuando tiene niveles de señal bajos (al interior de un edificio o en entorno urbano denso). AlwaysLocate permite al L80 ajustar los tiempos de ON/OFF automáticamente para obtener balance entre precisión de su ubicación y consumo eléctrico.

El [M95](#) es un módulo GSM/GPRS capaz de operar en 4 bandas. Es uno de los dispositivos más pequeños de su clase en el mundo. Ofrece máxima fiabilidad y robustez. A partir de su diminuto tamaño y facilidad al soldar, puede ser usado en diversidad de aplicaciones tales como voz y datos, SMS, Fax, VTS, PDA, rastreo personal.



Figura 1: Módulo L80 y M95

La integración de estos dos módulos hace una perfecta combinación GSM/GPRS y GPS, puesto que el alcance de sus proyectos no tendrá límite, puesto que por la facilidad de manejo de este shield para Arduino hace una forma muy sencilla de aprender.

## DESCRIPCIÓN

La tarjeta M95 L80 para Arduino es un Shield que permite integrar conectividad GSM/GPRS y GPS en sus aplicaciones de Arduino a través de los Módulos de Quectel.

El M95 es un módulo GSM/GPRS capaz de operar en 4 bandas (850/900/1800/1900 MHz), se controla con comandos AT a través del puerto serial del Arduino. Posee un amplio set de funciones, entre las cuales están GPRS/TCP/UDP/PPP/FTP/HTTP/SMS/Voz/FAX, por esto puede ser usado en diversidad de aplicaciones, adicional a esto el GPS L80 está conectado directamente con el módulo M95, lo que hace posible controlar el GPS por comando AT.

La tarjeta incluye todos los componentes necesarios para operar el Modem, tales como regulador e interfaces de sim-card, UART, antena, audio, botones y leds de control

### Características Hardware

- Regulador de voltaje: ajustado a 4.2 voltios. Este se alimenta del terminal VIN del Arduino, y esta línea regulada, alimenta al M95.
- Interfaz Serial: Circuito nivelador lógico que adapta los niveles de voltaje entre el M95 y Arduino, para ambas líneas del puerto UART (TX y RX).

Interfaz de "Sim Card": Socket para "Sim Card" de carga frontal. La tarjeta SIM debe empujarse horizontalmente para ajustarla y retirarla.

- Conector Antena: GPS (UFL) y GSM (SMA).

## ETAPAS Y CARACTERISTICAS ESPECÍFICAS

### 1. Entrada de voltaje

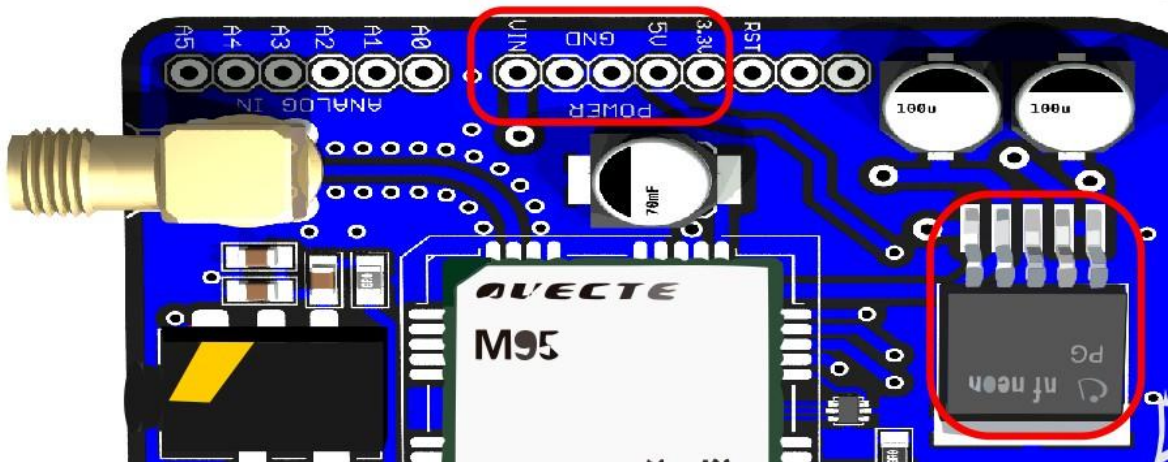


Figura 2: Pines de alimentación.

La alimentación de este shield, se realiza, mediante el pin de VIN de Arduino, es decir que su alimentación se debe apoyar en una fuente externa de 2 A mínimo para garantizar su correcto funcionamiento. Adicional a esto, el módulo GPS, es alimentado por los 3.3V, que posee Arduino.

### 2. Reset

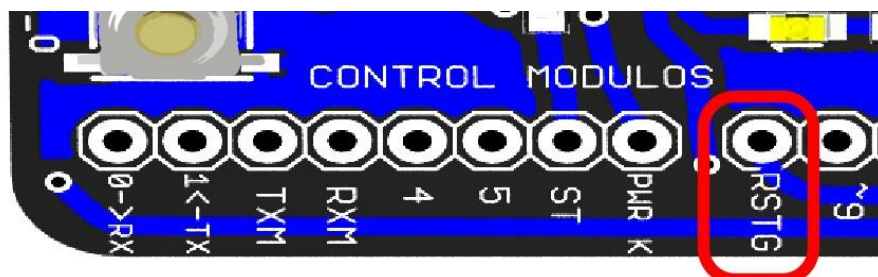


Figura 3: Reset GPS L80



Este pin permite reiniciar el módulo GPS este pin está conectado al pin 8 de los pines Arduino, para reiniciar el módulo, lo hacemos con un 0 lógico o estado bajo, esperamos 1s, y luego cambiamos a 1 lógico o estado alto..

La figura 4 muestra las líneas del código arduino de como se configura este pin, y como se crea una función de reset para dicho módulo.

```
sketch_oct03a $  
  
int rst_gps = 8; //Reset del GPS  
  
void setup(){  
  pinMode(rst_gps, OUTPUT);  digitalWrite(rst_gps, HIGH);  
}  
  
void rest(){  
  digitalWrite(rst_gps, LOW);  
  delay(1000);  
  digitalWrite(rst_gps, HIGH);  
  delay(100);  
}
```

Figura 4: Sentencias de Reset.

### 3. Pines utilizador para Arduino

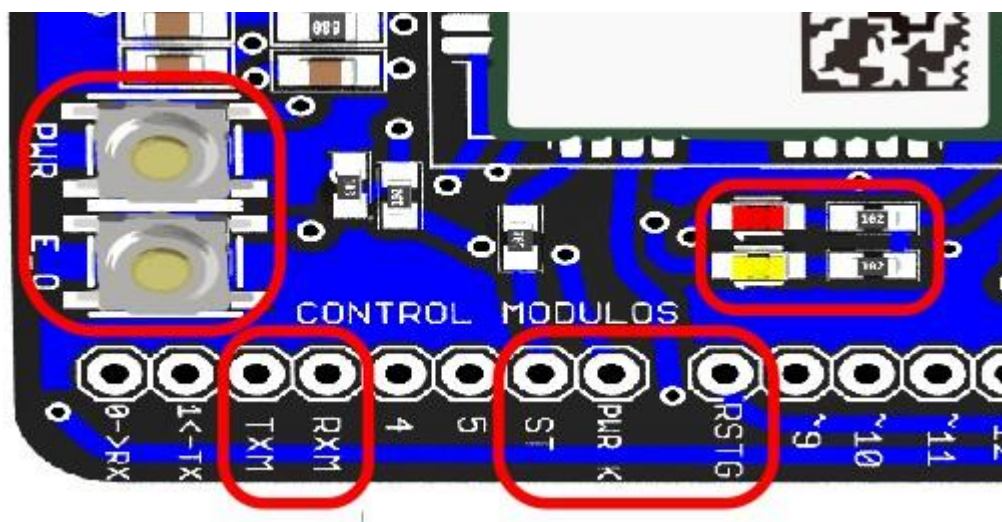


Figura 5: Pines utilizados en Arduino.

Los pines utilizador en Arduino, para controlar este shield son:

PINES UTILIZADOS	
ARDUINO	M95 L80 SHIELD
2	TXM
3	RXM
6	ST
7	PWR K
8	RSTG
VIN	VIN
3.3V	3.3V
GND	GND

- TXM: este pin es el transmisor del módulo M95.
- RXM: este pin es el receptor del módulo M95.
- ST: es el pin de estatus en el M95, que a su vez cuenta con un led amarillo.
- PWR K: es el pin de apagado, que se controla por arduino, o por un pulsador externo integrado en el shield.
- RSTG: es el pin de reset del GPS.
- VIN: voltaje de alimentación del shield.
- 3.3V: voltaje de alimentación del L80.
- GND: tierra de referencia.

Adicional a estos pines, este shield cuenta con un pulsador de apagado de emergencia (E\_O) para el módulo M95.

#### 4. Conectores de antenas

El shield cuenta con dos conectores de antena, en la figura 6 tenemos el conector SMA, para la antena del módulo M95, donde puedes incluir cualquiera de [estas](#) antenas.

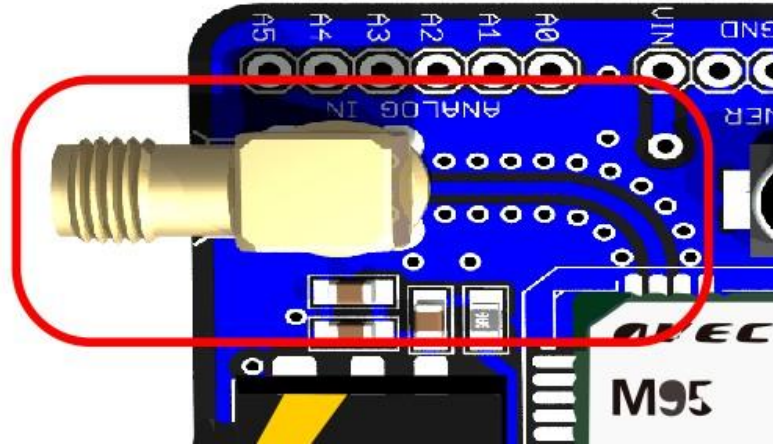


Figura 6: Conector SMA del M95

El conector UFL de la figura 7, es opcional, para conectar una antena al GPS, si las condiciones de funcionamiento son muy complejas en señal, se puede utilizar un [convertidor](#) de UFL a SMA, para utilizar esta [antena](#).

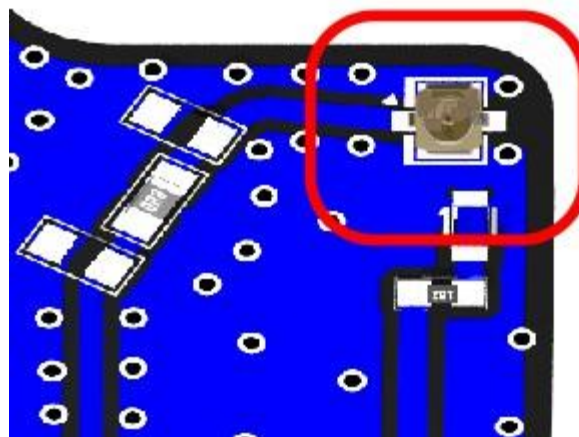


Figura 7: Conector UFL del L80.



## 5. Socket sim card retráctil

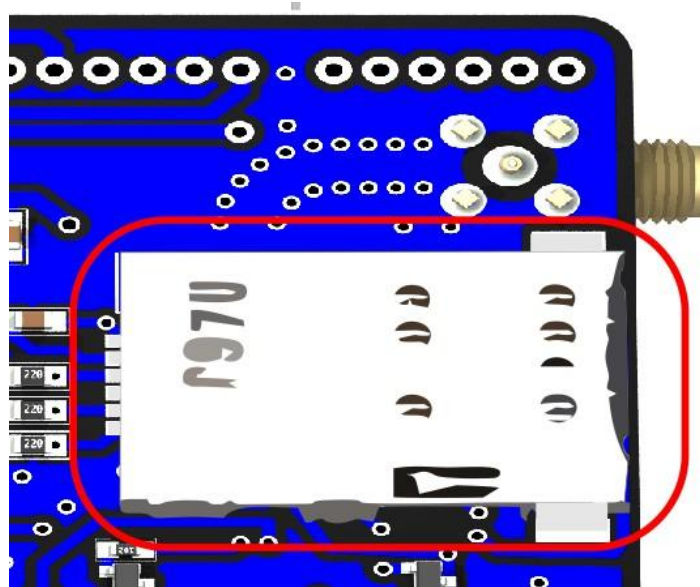


Figura 8: Socket sim card.

Este socket, permite el uso de sim cards de tamaño estándar, y con una gran ventaja, que es retráctil, soporta cualquiera de los tres operadores en Colombia el módulo M95.

## 6. Conector de micrófono y audio.

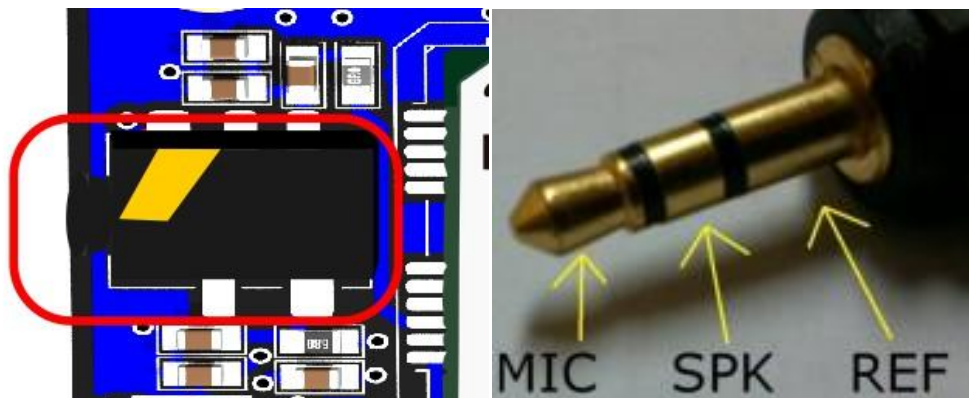


Figura 9: conector micrófono y audio.

El Shield posee un conector de 3.5mm para conexión de audífono y micrófono tipo “manos libres”. El módulo manos libres (no incluido) que debe usarse en este conector,

debe tener la conexión del micrófono ("MIC") en la punta del Plug, la conexión del audífono (o audífonos) en el punto central ("SPK"), y la referencia ("REF" o "GND") en la base del mismo.

## 7. Indicadores

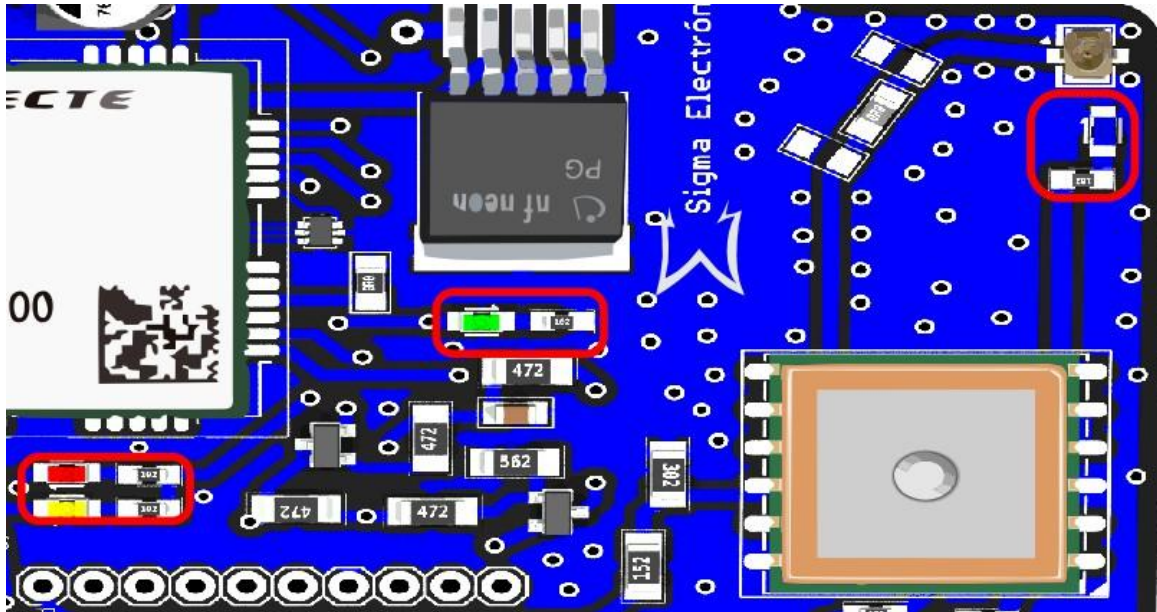


Figura 10: Indicadores led del shield.

- Led Azul, indica si el L80 está o no.
- Led Verde, indica si el shield está alimentado o no.
- Led Amarillo, este led indica el status del módem, es decir si esta encendido el M95 o no.
- Led Rojo, indica el estado de registro en la red de telefonía, es decir, con una sim card insertada, y encendido el M95, el led rojo, comenzara a parpadear rápidamente, pero cuando se registre en la red celular, el led parpadeará más lento.

## FUNCIONAMIENTO

El funcionamiento de este shield se va a mostrar con un ejemplo básico, el cual consta de verificar unos comandos y datos del modulo M95 L80.

Se necesita:

[Arduino UNO](#)

[M95+L80 SHIELD](#)

[ANTENA GSM MAG](#)

Fuente de 12V - 2A

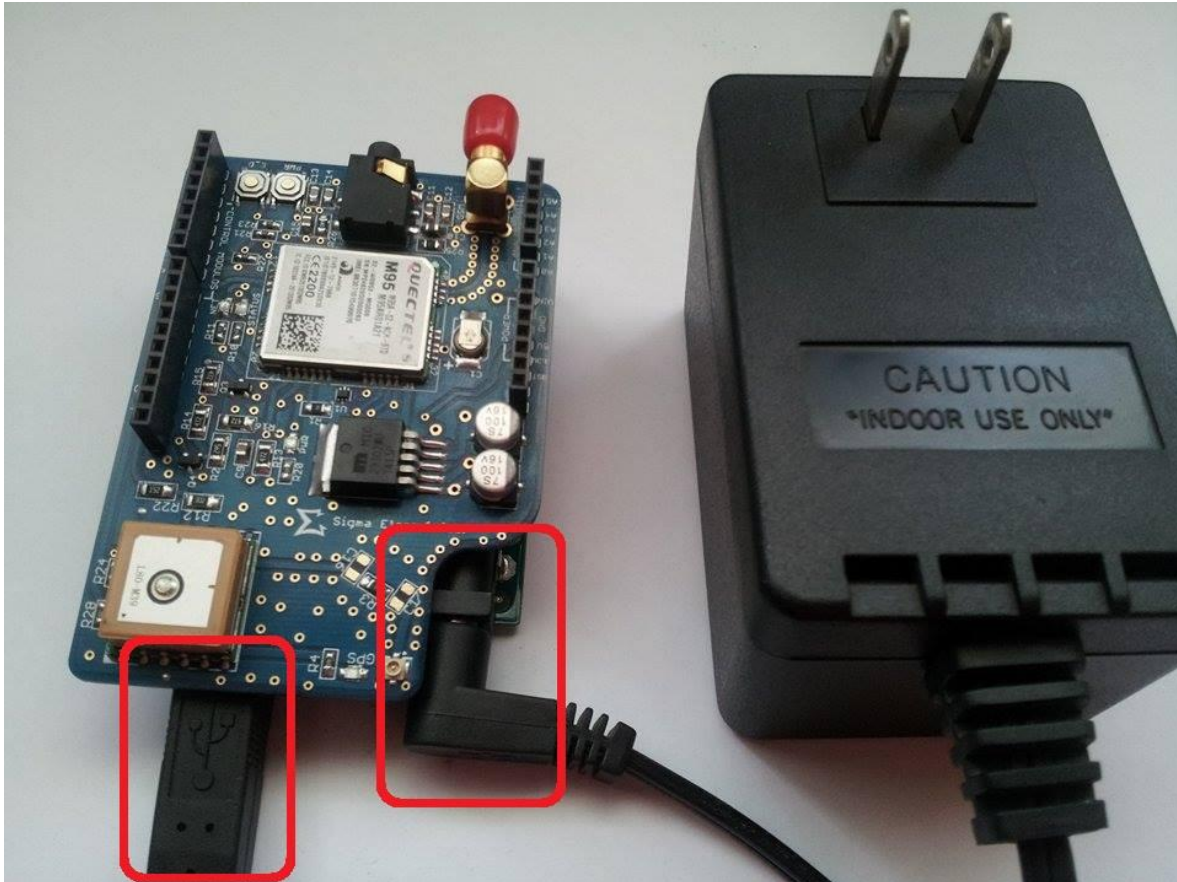
### PASO UNO: Conectar el shield a Arduino Uno



Figura 11: Conexión del Shield a Arduino

Al colocar el Shield en Arduino, vamos a verificar el sentido de la tarjeta, nos podemos guiar con el conector de la sim card que este en el lado al cable USB de Arduino, o sobre los pines ISP del Arduino.

## PASO DOS: Realizar la conexión USB y alimentación del Arduino uno al PC



**Figura 11:** Conexión USB y alimentación.

Nos disponemos a programar nuestro Arduino Uno, así que realizamos la conexión del cable USB, y el cable de alimentación de la fuente externa de 12-17V a 2A (min), lo que alimentará el shield y encenderá el led verde.



#### PASO CUATRO: Conectamos la antena SMA



Figura 11: Conexión de la antena.

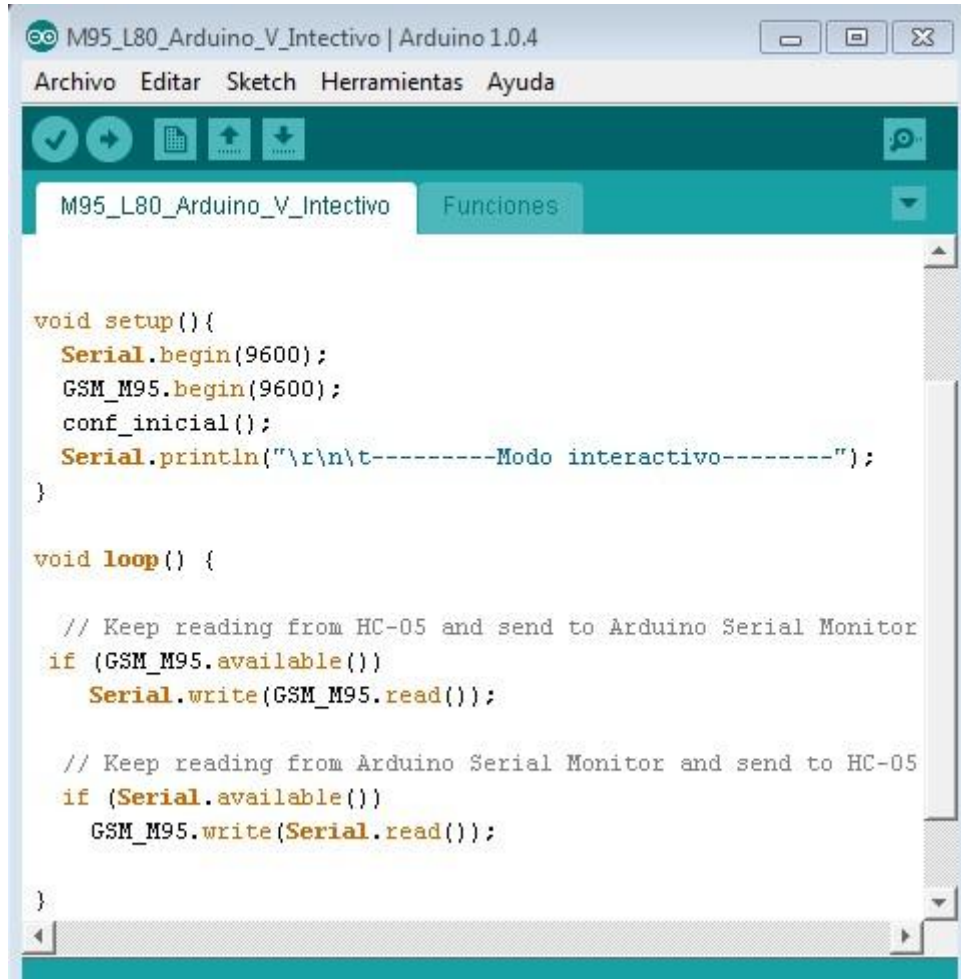
#### PASO CINCO: Cargar programa de ejemplo

Antes de cargar el programa debemos tener en cuenta que es recomendable apagar el módulo M95 antes de programarlo, es decir si previamente ha estado encendido, es decir, si el led rojo esta parpadeando “esta encendido”, entonces oprimimos el pulsador E\_0, y veremos que el led rojo deja de parpadear y se apaga.



Figura 12: Pulsador de apagado E\_0.

Después de abrir nuestro Software de programación, vamos a utilizar el siguiente [código](#) para realizar la programación del Arduino Uno. (Más [códigos Ejemplo](#))



```
void setup(){
  Serial.begin(9600);
  GSM_M95.begin(9600);
  conf_inicial();
  Serial.println("\r\n\t-----Modo interactivo-----");
}

void loop() {

  // Keep reading from HC-05 and send to Arduino Serial Monitor
  if (GSM_M95.available())
    Serial.write(GSM_M95.read());

  // Keep reading from Arduino Serial Monitor and send to HC-05
  if (Serial.available())
    GSM_M95.write(Serial.read());

}
```

**Figura 13:** Interfaz Arduino con programa prueba.

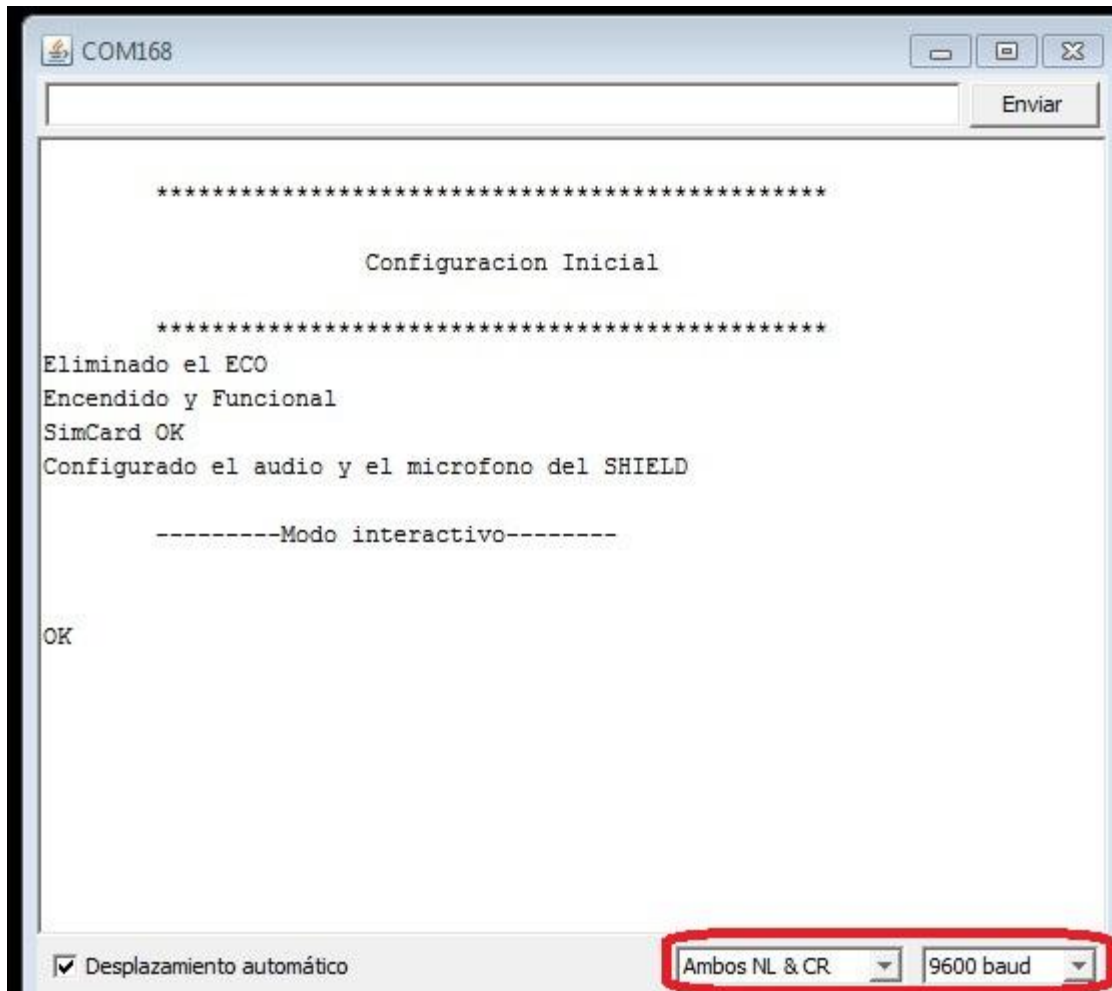
Este código consta de una función adicional, que permite las configuraciones respectivas del modem, previamente a su funcionamiento, este código, no va a servir para comunicarnos con el modulo M95, de manera transparente.

Cargamos el programa a nuestro Arduino Uno, en donde encenderá el led amarillo, y luego parpadea el led rojo.



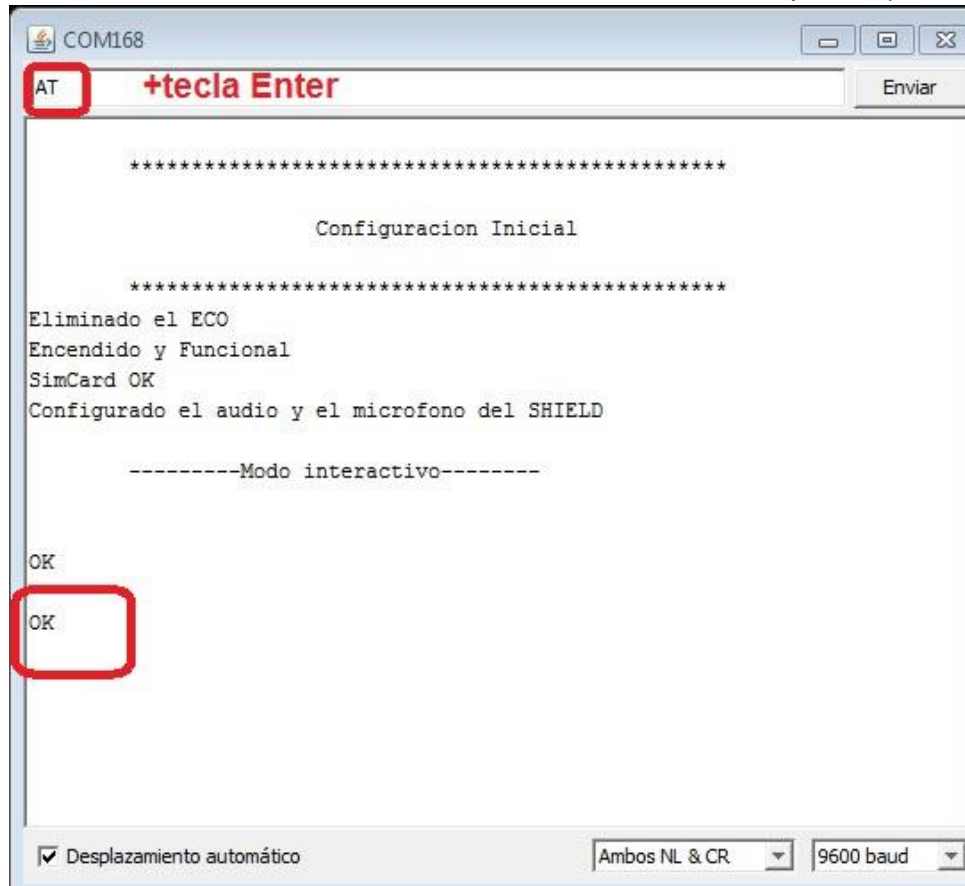
## PASO SEIS: Sentencias de prueba.

Luego de haber cargado el programa en nuestro Arduino Uno, abrimos el monitor de Arduino, y podremos ver la figura 14, donde cargan unos parámetros de configuración básicos, se debe tener en cuenta que la velocidad de transferencia debe ser de 9600.



**Figura 14:** Terminal serial de Arduino

Ahora cuando usemos los [comandos AT](#) del modem M95 damos enter y esperamos una respuesta en el terminal, como en la Figura 15.



**Figura 15:** Terminal serial de Arduino comando AT

Ahora vamos a seguir la siguiente secuencia de comando, para ver el comportamiento del shield.

-----Comandos-----

AT  
 ATI (entrega los datos del Modem)  
 AT+GSN (Entrega el IMEI)  
 ATD\*611; (llama a un numero en este caso el \*611, al final debe ir ;)  
 ATH (Finaliza una llamada)  
 AT+COPS? (entrega el operador de la Sim Card)  
 AT+QGPSCFG? (Verifica el puerto del GPS).  
 AT+QGPSCFG="GPSINIT",1 (Activa el GPS).  
 AT+QGPSRD="NMEA/RMC" (Lee la trama RMC del GPS)

-----

Después de escribir los comandos anteriores tendremos:

```

COM168
SIMCard OK
Configurado el audio y el microfono del SHIELD

-----Modo interactivo-----

OK

OK

Quectel_Ltd
Quectel_M95
Revision: M95AR02A02

OK

863071015499970

OK

OK

OK

OK

+COPS: 0,0,"Claro"

OK

+QGPSCFG: "GPS"PORT", "UART2"
+QGPSCFG: "GPS"BAUD", 9600
+QGPSCFG: "GPS"INIT", 0

OK

OK

+QGPSRD: $GPRMC, 192213.000, A, 0438.9243, N, 07404.3908, W, 0.02, 0.00, 031014, , , D*70

OK
    
```

**Figura 16:** Resultado Trama de comandos

Dudas, inquietudes o sugerencias sobre este producto, contacte a [ingenieria@sigmaelectronica.net](mailto:ingenieria@sigmaelectronica.net)

*Elaboró: Ing. Juan C. Guarnizo B.*

## WEBGRAFÍA

- Hoja de diseño del M95. Visto en:  
[http://www.sigmaelectronica.net/docs/M95\\_HD\\_V1.2.pdf](http://www.sigmaelectronica.net/docs/M95_HD_V1.2.pdf) (Recuperado: 31-06-14)
- Comandos AT M95. Visto en:  
[http://www.sigmaelectronica.net/docs/M95\\_AT\\_Commands\\_Manual\\_V1.2.pdf](http://www.sigmaelectronica.net/docs/M95_AT_Commands_Manual_V1.2.pdf)  
(Recuperado: 31-06-14)
- Comandos del GPS con el M95. Visto en:  
[http://www.sigmaelectronica.net/docs/GSM\\_UART\\_GNSS\\_Application\\_Note\\_V3.0.pdf](http://www.sigmaelectronica.net/docs/GSM_UART_GNSS_Application_Note_V3.0.pdf) (Recuperado: 31-06-14)
- Hoja de diseño del L80 Visto en:  
[http://www.sigmaelectronica.net/docs/L80\\_GPS\\_Protocol\\_Specification\\_V1.0.pdf](http://www.sigmaelectronica.net/docs/L80_GPS_Protocol_Specification_V1.0.pdf)  
(Recuperado: 31-06-14)
- Protocolo del L80 Visto en:  
[http://www.sigmaelectronica.net/docs/L80\\_GPS\\_Protocol\\_Specification\\_V1.0.pdf](http://www.sigmaelectronica.net/docs/L80_GPS_Protocol_Specification_V1.0.pdf)  
(Recuperado: 31-06-14)