

## M95 SHIELD PARA ARDUINO. SIGMA ELECTRONICA



Imagen 1: Tarjeta M95 ARD.<sup>1</sup>

### 1. DESCRIPCION:

M95 Shield para Arduino. Permite integrar conectividad GSM/GPRS en sus aplicaciones de Arduino a través del Modem [Quectel M95](#).

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.

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 (imagen 2).

---

<sup>1</sup> Sigma Electrónica Ltda.

## 2. CARACTERISTICAS:

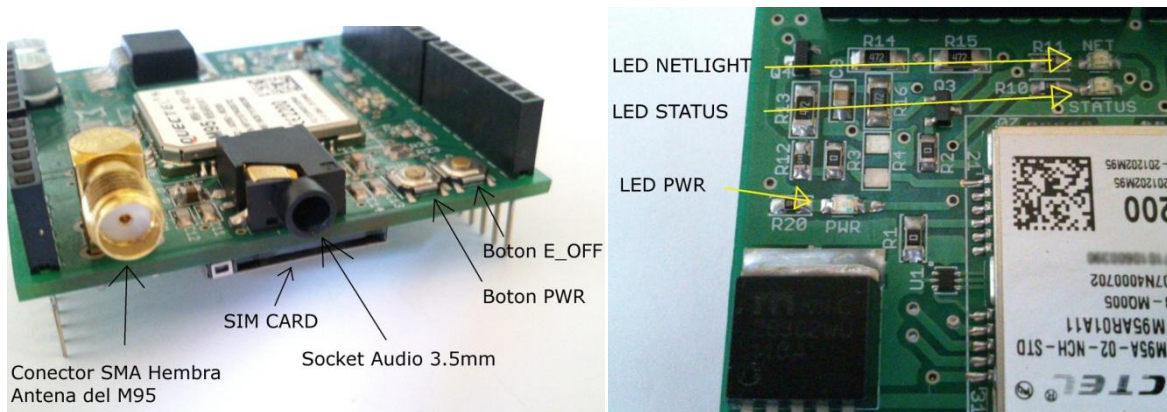


Imagen 2: Accesorios de la tarjeta.

- **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 Antena:**  
Conector SMA hembra para conectar la [antena del M95](#) (se vende por separado).
- **Interfaz de "Sim Card":**  
Socket para "Sim Card" de carga frontal. La tarjeta SIM debe empujarse horizontalmente para ajustarla y retirarla.



Imagen 3: Terminales de conexión del Plug de audio.

- **Plug de Audio:**  
El Shield posee un socket de 3.5mm para conexión de audífono y micrófono tipo "manos libres". El módulo manos libres (no incluido) que debe usarse en este socket, 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 (imagen 3).

- Botones de control:

El botón “PWR” controla el terminal “PWRKEY” del M95 el cual permite prenderlo o apagarlo “manualmente”.

El botón “E\_O” controla el terminal “EMERG\_OFF” del M95, para accionar el apagado de emergencia “manualmente”.

- Leds de estado:

Led PWR indica que la tarjeta está siendo alimentada (conectado a la línea de salida del regulador de 4.2 voltios).

Led STATUS: Indicador del terminal Status del M95. Permite conocer a simple vista, si el M95 está prendido o apagado.

Led NET: Indicador del terminal Netlight del M95. Permite conocer el estado en la red del M95. [Refiérase al capítulo 3.12. Network Status Indication.](#)

### 3. CONEXIONES CON ARDUINO:

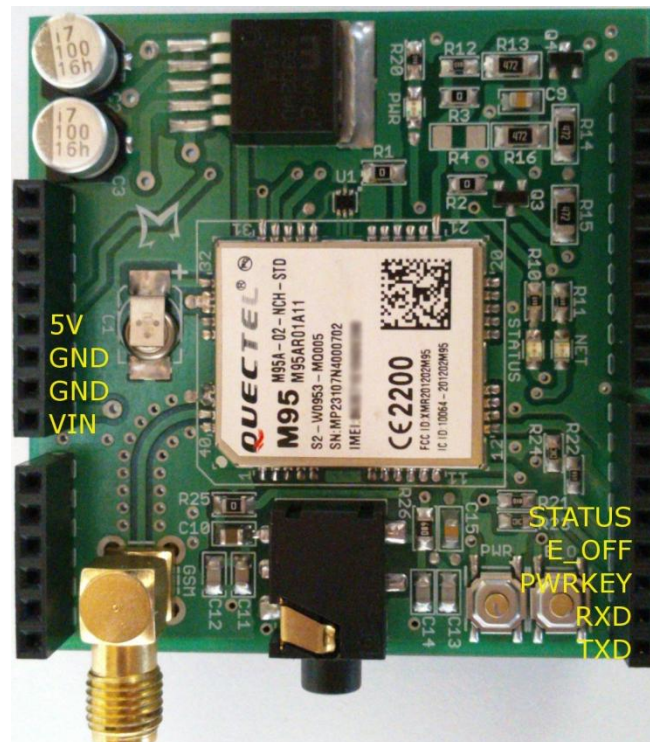


Imagen 4: Pines usados por el Shield.

- **Cuatro terminales de conexión:** (1 de diez pines, 2 de ocho pines y 1 de seis pines), conectan todos los pines de la tarjeta Arduino al Shield. Los circuitos de la Sim-Card, de la antena, y del socket de audio, no intervienen de forma alguna con la tarjeta Arduino. Los puntos de conexión de energía (terminal de conexión de 8 pines al lado superior izquierdo de la imagen 4), pueden usarse para alimentar otros componentes que puedan necesitarlos en su circuito, teniendo en cuenta que el terminal “VIN” presentara el voltaje conectado en el “jack” de alimentación externa de la tarjeta Arduino.

- **Energía:** Este Shield, se alimenta del terminal “VIN” del Arduino. El voltaje presente en este pin, alimenta el regulador del cual se alimenta el M95. Por esto, se recomienda no usar este Shield cuando la tarjeta Arduino esta alimentándose por USB. Debe usar un regulador o fuente externa y alimentar la tarjeta Arduino a través de su jack de energía (imagen 5). En este plug, se recomienda usar una fuente que soporte al menos 2 amperios, y un voltaje entre 9 y 15 voltios. El Shield también usa la línea de 5 voltios de la tarjeta Arduino, pero solo se usa para el circuito de nivelación lógica, por lo cual no se exige corriente de este.

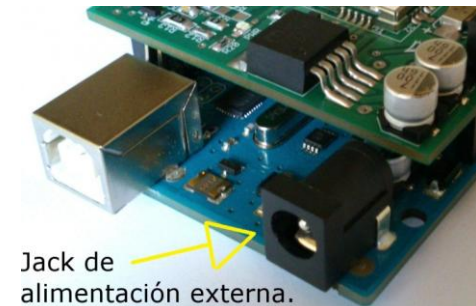


Imagen 5: Alimentación de la tarjeta Arduino.

- **UART:** El puerto serial de M95, va conectado al puerto serial del Arduino. Por esto, los terminales digitales 0 y 1 del Arduino estarán siempre ocupados por el Shield. En estos dos terminales, es donde se presenta el circuito de nivelación lógica entre al M95 y la tarjeta Arduino, para proveer al RX del Arduino con niveles lógicos de 0-5v, y al RX del M95 con niveles lógico de 0-4.2v. El puerto UART del M95 está configurado en modo “auto-baud”, como configuración de fabrica (soporta velocidades entre 4800 y 115200). [Refiérase al capítulo 3.8 Serial Interface.](#)
- **STATUS:** Este terminal del M95, se encuentra conectado al pin digital 4 del Arduino. Este pin del Arduino debe ser configurado como entrada con pull-up. Esta conexión se presenta con el fin de que se pueda conocer el estado del M95 (ON/OFF) en el software, realizando lectura del pin. Paralelamente, el STATUS del M95 está conectado a un led para visualizar su estado lógico (LED STATUS en Imagen 2).
- **PWRKEY:** Este terminal del M95, se encuentra conectado al pin digital 2 del Arduino. Esta conexión se tiene con el fin de controlar el encendido o pagado del M95 a través de software, donde debe estar en “0” constantemente, y en “1” al momento de activarlo (este 0 y 1 se controla en el programa de la tarjeta Arduino). Para prender o apagar el M95, debe accionarse el pin pwrkey (2), colocando un “1” lógico, hasta que el terminal STATUS cambie a “1”; en ese momento se puede bajar el pwrkey a “0”. El terminal PWRKEY del M95 puede ser operado por el pulsador destinado a esto (“PWR”), de forma paralela al control por software.

- **EMERG\_OFF:** Este terminal del M95, se encuentra conectado al pin digital 3 del Arduino. Esta conexión tiene como fin, controlar el apagado de emergencia a través de software, donde el pin debe estar en “0” constantemente y en “1” al momento de activarlo (controlar este 0 y 1 en el programa de la tarjeta Arduino). El terminal EMERG\_OFF del M95 puede operarse paralelo a esto, a través del pulsador destinado a esto (“E\_O”).

#### 4. MODO DE OPERACIÓN:

**Software:** los pines listados a continuación, son usados por el Shield y por ello no puede conectar alguna otra cosa en ellos. Los pines y su configuración son (Esta numeración de pines, tiene como referencia la numeración de los pines digitales en el Arduino UNO, donde RX es el pin 0 y TX es el pin 1 del terminal de conexión de 8 puntos, contiguo al terminal de 10 pines):

PIN 0: Entrada digital (RX del UART del Arduino)  
 PIN 1: Salida digital (TX del UART del Arduino).  
 PIN 2: Salida digital. Estado inicial en 0 (PWRKEY).  
 PIN 3: Salida digital. Estado inicial en 0 (EMERG\_OFF).  
 PIN 4: Entrada digital con pull-up (STATUS).  
 PIN 13: Salida digital (OPCIONAL, para operar el led en la tarjeta Arduino).

Encender el M95: Realice lectura del terminal STATUS. Si el resultado es un “LOW”, indica que el M95 está apagado y para prenderlo debe operarse el terminal PWRKEY. Accione el terminal PWRKEY con un “1”, y haga un ciclo de lectura continua del terminal STATUS. Cuando STATUS sea “1”, puede cambiar PWRKEY a 0, con esto el M95 está activo y comienza a conectarse a la red celular automáticamente.

Abrir puerto UART: Inicie el puerto UART del Arduino a través de la función Serial.begin(). Puede colocar cualquier velocidad entre 4800 y 115200 bps, cuando el M95 tiene los ajustes de fabrica (entre estos su puerto serial se encuentra en modo “auto-baud”, el cual auto detecta la velocidad del puerto cuando se le envía algún dato). Normalmente, puede operar todas las funciones del M95, con una velocidad de 115200bps. Después de abrir el puerto serial, debe enviar al comando “AT” algunas veces, hasta que el M95 responda “OK”. Esto puede hacerse, con solo enviar “AT” continuamente hasta que el buffer del puerto UART deje de estar vacío (con la función Serial.available()), o revisando que los datos que recibe el Arduino coincida con “OK”. Después del OK, es posible enviar comandos de configuración para el M95, sea para cambiar la velocidad del puerto, para cambiar el canal de audio, o para leer datos de configuración de algún registro.

Con esta configuración, es posible mandar a hacer algo al M95 en cualquier momento del programa (realizado por el usuario), por ejemplo conectar un pulsador en algún pin digital y al accionarlo, mandar al M95 a que envíe un SMS o



que realice una llamada, etc..; para esto, refiérase a la [hoja de comandos AT del M95](#).

En el lenguaje de programación de Arduino, es muy simple enviarle los comandos AT al M95, basta con usar la función `Serial.println()`; , donde el paréntesis contiene el comando AT entre comilla doble así: `Serial.println("AT+QAUDCH=2")`; La función `Serial.println()`; contiene el salto de línea y retorno de carro incluido, lo que normalmente se realizaría con un "enter" de teclado.

**PROGRAMA DE EJEMPLO:** Este ejemplo, configura los pines adecuadamente para operar el Shield, inicia el puerto UART del Arduino y se quede censando un pin en el cual hay un pulsador. Al accionar el pulsador, el Arduino manda al M95 a realizar una llamada de voz a un número predeterminado en programa (debe cambiar ese número por uno donde pueda responder para realizar esta prueba).

## Hardware:

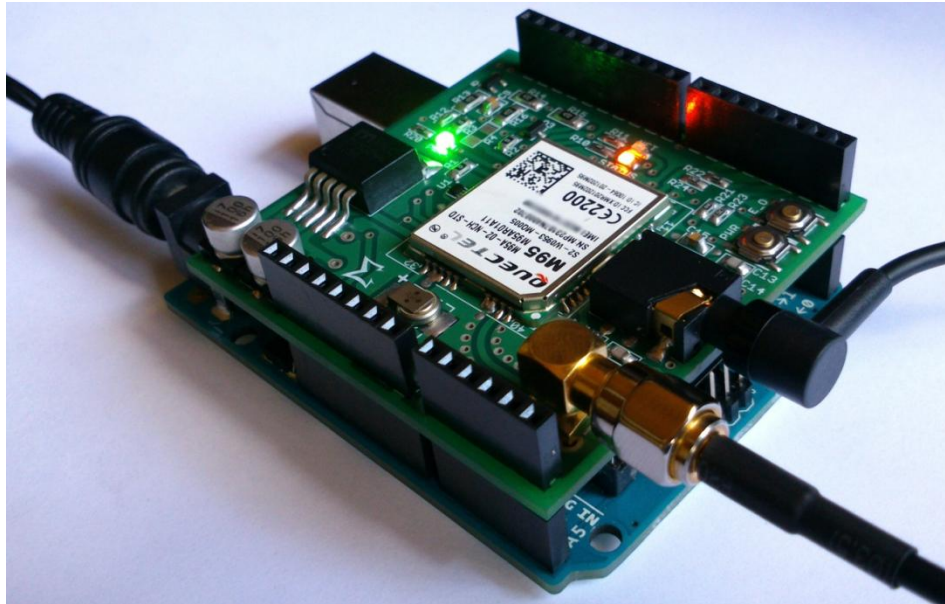


Imagen 6: Conexión del Shield con un Arduino UNO.

- Realice el programa para su tarjeta Arduino, configurando los pines para el Shield adecuadamente, luego descargue el programa a la tarjeta a través de USB (cable USB no incluido).
- Desconecte el cable USB y Conecte el Shield sobre su tarjeta Arduino (imagen 6, imagen de referencia con una Arduino UNO).
- Con esto listo, alimente la tarjeta Arduino a través de su jack para alimentación externa, y listo.
- **Precaución:** Si va a realizar cambios en el programa de su tarjeta Arduino, desconecte la alimentación externa (imagen 5) y el Shield; luego conecte el cable USB únicamente.

## 5. RESUMEN:

- Tarjeta M95-ARD basada en el Módulo GSM/GPRS Quectel M95.
- Voltaje y corriente de operación: El Shield se alimenta de los terminales VIN y 5V de la tarjeta Arduino sobre la que se conecte. Alimente su tarjeta Arduino a través del jack de energía, con una fuente entre 9 y 15 voltios, que soporte al menos 2 Amperios.
- Tipo de conector: Cuatro terminales de conexión para ensamble sobre su tarjeta Arduino.
- Dimensiones de la tarjeta: 55mm x 54mm x 14mm.

## 6. PRECAUCIONES:



Imagen 7: Límite de inserción de los terminales de conexión.

- Al ensamblar el Shield sobre su tarjeta Arduino, no presione los terminales de conexión hasta el fondo. Es posible que algún (o algunos) componentes en la parte inferior del Shield hagan contacto con algún componente de la tarjeta Arduino (imagen7). Por esto, presione el Shield hasta que los pines de los terminales de conexión queden con una separación de unos 3mm (insertando los pines hasta este punto, hay conexión adecuada con los pines del Arduino).



Imagen 8: M95 Shield operando con la alimentación externa de la tarjeta Arduino.

- Use siempre el jack de alimentación externa de su tarjeta Arduino, al momento de usar el Shield (imagen 8). De lo contrario, al prender el M95, en el momento de conectarse a la red, este va a pedirle 1.6 Amperios a la línea de energía del puerto USB, lo cual puede ocasionar daños.

## 7. IMÁGENES DE REFERENCIA:

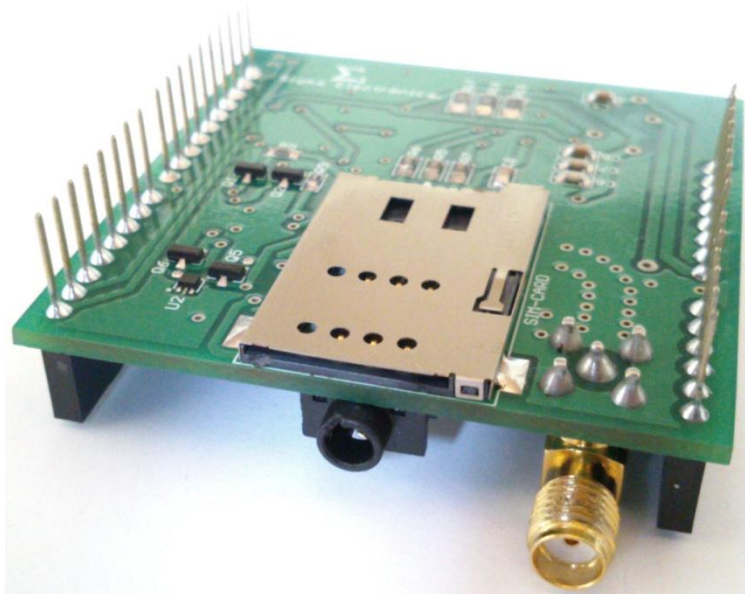


Imagen 9: imagen de referencia.





Imagen 10: imagen de referencia.



Imagen 11: imagen de referencia.

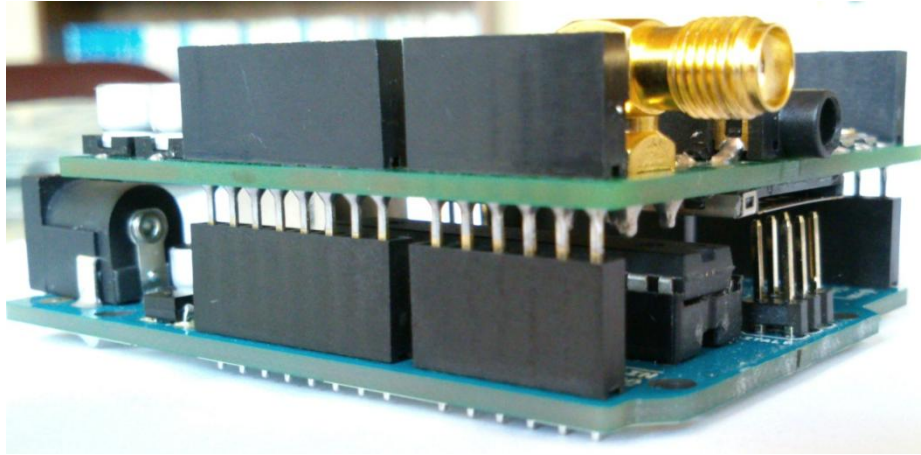


Imagen 12: Imagen de referencia.

---

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