DESCRIPTION

The Saleng GSM shield adds SMS, GPRS and data call capabilities to your Saleng Uno or Arduino projects. The board greatly simplifies implementation with its Arduino "shield" form factor and integrated supporting circuits. All that is required is an Arduino or any microcontroller with a UART, the Saleng GSM shield and 5-12Vdc/2A source of power. No complicated wiring.

Figure 1: The Saleng GSM Shield

This shield is a culmination of several years of experience working with GSM modules, in particular, the SIMxxx series. The shield is powered by the popular SIM800L GSM module and comes with the necessary power circuit, logic level conversion and SIM card slot. It also comes with an integrated helical spring antenna. There are mini jumpers onboard for other hardware options.

The shield is compatible with any microcontroller or computer with a Serial Port (UART). The serial port, audio in and out, and other signal lines from the SIM800L are exposed via pin headers on the top layer. With an Arduino however, connection is even faster with the stackable pin headers. The Arduino UNO, Mega, Leonardo, Zero and other boards and clones of the same form factor are compatible with the shield connectors. There are also pads that allows a user to quickly connect an Arduino Nano on the top of the board.

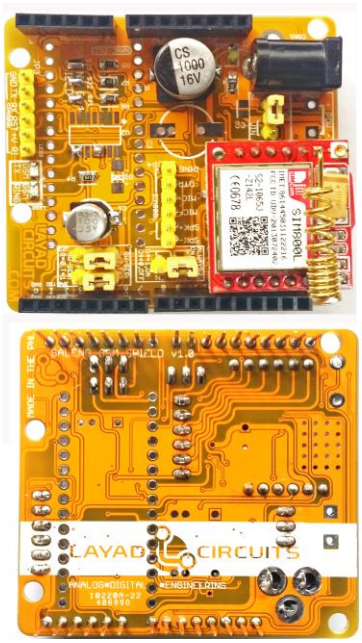


Figure 2: Top and bottom view

FEATURES

- Powered by SIM800L GSM module.
- detachable/ replaceable core SIM800L
- Onboard 3A LDO and Level Shifter circuit
- Arduino Shield form factor with Arduino Nano pin header slots
- Wide external input voltage: 5V ~ 12V
- Onboard DC Jack with slots for Terminal Block option
- Integrated antenna and micro SIM card slot
- Jumper selection for power source, UART pins and Power Enable method
- RI and TXD LED indicators

Revision: v1.1/ 09 Sep 2017 /CDM

Layad Circuits Electronics Engineering Supplies & Services, B314 Lopez Bldg., Session Rd. cor. Assumption Rd., Baguio City, Philippines
General inquiries: info@layadcircuits.com Sales: sales@layadcircuits.com FB: facebook.com/layadcircuits Mobile: +639164428565

- Recommended Power Supply Current = 2A for heavy communications. May be lower current for simpler applications

HARDWARE OVERVIEW

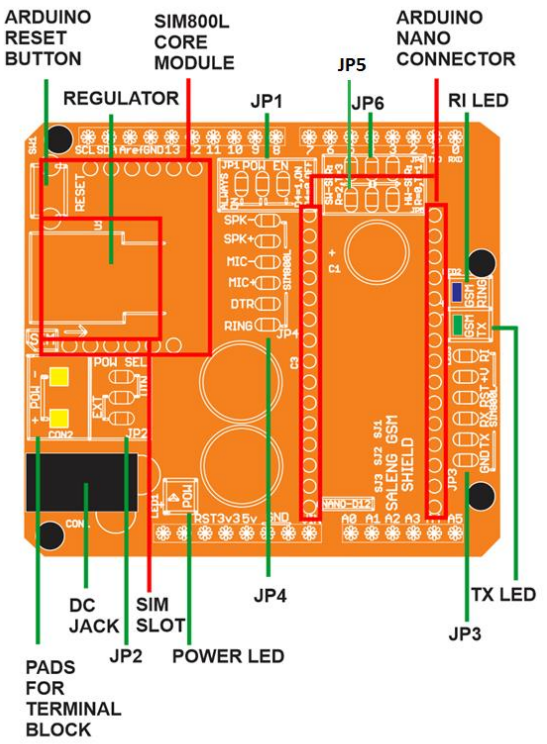


Figure 3: Saleng GSM Shield major components

APPLICATIONS

There are several applications involving GSM communications with computers. Listed below are some applications

- Remote monitoring and notification
- SMS/Data Call/GPRS remote control
- Machine-to-Machine communications
- GPRS modem
- Internet-of-Things (IOT)

POWER REQUIREMENTS

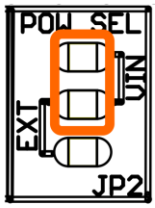
The SIM800L core module alone has a peak current of 2 Amperes during communications. This may be brief but is significant if communications is continuous. The whole shield draws a current below 200mA when the SIM800L is idle. With these we recommend a power supply rated at least 2A with a voltage of 5V up to 12V.

However, because of the shield's design, using a 1A power supply may also work for applications with less active communications.

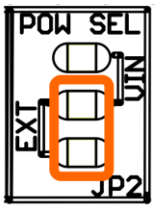
JUMPER OPTIONS

Power Source Selection: Jumper JP2

The 3-pin header labeled POW SEL allows the user to choose where the input power for the shield will be sourced from. It may either come from the power source connected to the DC Jack ("EXT") or from the VIN pin of the Arduino ("VIN").



Jumper set to VIN: Power is taken from the VIN pin of the Arduino



Jumper set to EXT: Power is taken from external supply via the DC Jack

Figure 4: JP2 options

The VIN position is not recommended for heavy communications. Depending on your Arduino board,

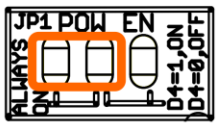
there is a 1A power diode between the DC Jack of the Arduino and the VIN pin. The shield may draw up to 2A during communications, therefore, if the shield is used extensively and continuously, it is recommended to use an external power supply instead and set the jumper to the EXT position. You may also simply split the power source and connect the Arduino's DC plug and the Saleng GSM Shield's DC plug in parallel so long as the common power supply used meets the power requirements of both boards (6-12V). Keep the jumper in the EXT position if this option is desired.

For communications with wider time intervals, the VIN position simplifies power wiring as you only need to connect your power supply at the Arduino's DC Jack. The power is then propagated to the shield via the Vin pin of the Arduino. This means only the DC jack of the Arduino needs to be inserted with power.

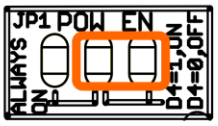
Power Enable Option: Jumper JP1

Jumper JP1 is used to select between having the module always turned on as power is applied or have the microcontroller control the regulator power and hence power to the rest of the circuit. When the microcontroller takes control (micro jumper is set at the side labled"D4=1.."), it may turn off the output of the regulator chip and hence no power is applied to the SIM800L. This is done by applying a LOW on digital pin 4. Power may turned back on by applying a HIGH on pin 4. This scheme provides an actual hardware reset on the SIM800L to occur and is useful in low power applications or applications requiring reliability.

In contrast, when the micro jumper is set at the ALWAYS ON side, the shield is always powered as long as power is applied to the board.



Jumper set to ALWAYS ON: The SIM800L remains powered so long as the power is available



Jumper set to D4: Power to the SIM800L is controlled by the logic state of the Arduino pin D4. If D4 is LOW, power is turned off. If D4 is HIGH, power is turned on.

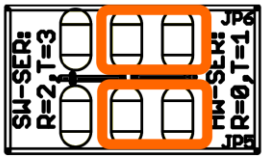
Figure 5: JP1 options

UART Pin Selection: Jumper JP5 & JP6

There are 2 columns representing the TX and RX pins of the SIM800L. Using the 2 jumpers on board, the user may choose to use the UART or "Hardware Serial" of the Arduino at pins D0 and D1. In the Arduino board, pins D0 and D1 are also shared as the PC interface via the USB converter circuit. Because of this, uploading sketches to the Arduino may generate errors while the JP6 micro jumpers are installed. To avoid this, you may want to uninstall the micro jumpers or temporarily remove the SIM800L core board during uploading of sketches.

The user may also want to use a software emulated UART or "Software Serial" set at pins D2 and D3. If this is desired, pin D2 should be set in the code as the receive pin of the Arduino and D3 as the transmit pin of the Arduino e.g. SoftwareSerial mySerial(2,3); .

The middle pins of the header also gives access to 5V-converted serial port of the SIM800L.



Jumper sets the HW Serial as SIM800L interface

Jumper sets the SW Serial on pins D2 and D3 as SIM800L interface

Figure 6: JP5 & JP6 Options

PIN FUNCTIONS

Arduino Shield Headers

The Arduino pin headers are the 4 Arduino standard connectors where the 4 stackable headers are installed making the board as a "shield". Among these pins, the following are used:

Arduino Shield Pins	Function as used on the shield
VIN	Connected to JP2. When selected via JP2, VIN is used as power source of the SIM800L
GND	Connected to the shield's GND
D0 and D1	Connected to JP5 & JP6. Used as SIM800L interface
D2 and D3	Connected to JP5 & JP6. Used as SIM800L interface as SW serial
D4	Used to enable/disable the regulator power when set on JP1

Arduino Nano Headers

The Arduino Nano slots are unpopulated. If a Nano is desired instead of an Uno, Leonardo or Mega, female pin headers with 2.54mm pitch may be installed on these slots. The Nano may then be inserted on the top layer of the shield. The shield has a pin labeled as NANO-D12 to mark the correct orientation of the Arduino Nano board.

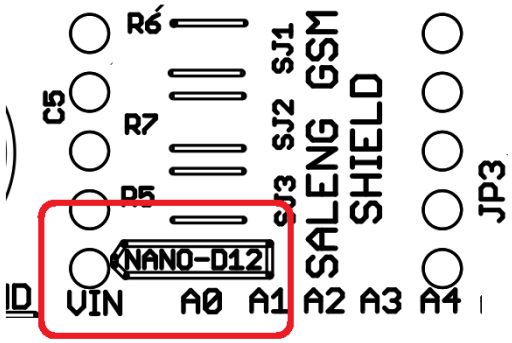


Figure 7: Label to correctly orient the Arduino Nano

The Arduino Nano pins are connected to their corresponding counterpart on the Arduino Shield pin headers. A6 and A7 are unconnected as these are not available on the standard Uno Shield headers.

Header JP3

JP3 are pins coming from the SIM800L core board.

Pin Label	Function as used on the shield
GND	Connects to the common ground point
TX	Data Transmit line of the SIM800L. **This is 3.3V Logic.**
RX	Data Receive line of the SIM800L. **This is 3.3V Logic.**
RST	Reset line of the SIM800L
+V	Power of the SIM800L. This is approximately between 4 and 4.1V. This is tied to the output of the regulator.
RI	Ring Indicator of the SIM800L. Connected to RING pin of JP4. This may be used to detect an incoming call or SMS without using AT commands.

Revision: v1.1/ 09 Sep 2017 /CDM

Layad Circuits Electronics Engineering Supplies & Services, B314 Lopez Bldg., Session Rd. cor. Assumption Rd., Baguio City, Philippines
General inquiries: info@layadcircuits.com Sales: sales@layadcircuits.com FB: facebook.com/layadcircuits Mobile: +639164428565

Header JP4

JP4 are pins coming from the SIM800L core board.

Pin Label	Function as used on the shield
-SPK	Negative Terminal, Audio Output
+SPK	Positive Terminal, Audio Output
MIC-	Negative Terminal, Audio Input
MIC+	Positive Terminal, Audio Input
DTR	DTR control signal of the SIM800L UART
RING	Ring Indicator of the SIM800L. Connected to RI pin of JP3. This may be used to detect an incoming call or SMS without using AT commands.

LED Indicators

LED Label	Behavior
POW LED	This is ON when the power regulator is enabled. When the micr jumper at JP1 is on the ALWAYS ON side, this LED should be ON when power is applied. When the micro jumpers is at the D4 side, this LED is ON only when D4 is HIGH. It remains OFF when the power regulator is disabled or no power is available
GSM TX	This is connected directly at the TX line of the SIM800L and hence represents the actual status of that signal. This is normally turned ON since the TX is normally HIGH. When data is sent out byt eh SIM800L, this LED may flicker if the baud rate is slow enough and the data stream is continuous. At higher baud rate, the flicker may not be obvious.
GSM RING	This LED represents the logic state of the Ring Indicator (available from JP3 or JP4). The LED is normally HIGH. When a call is received, the LED is turned OFF while ringing. When an incoming SMS is received, the LED
	momentarily goes OFF then goes back ON.

There is also an LED on board the SIM800L core module as shown the figure below.

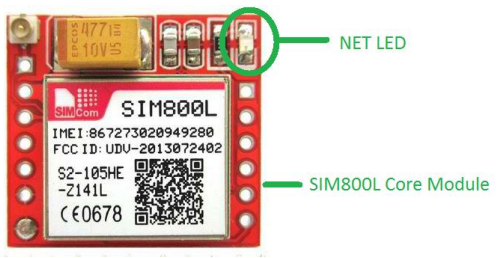


Figure 8: NET LED on the SIM800L core module

This LED shall blink fast at around once eery second when it has yet to establish connection to a GSM network either due to the non-availability of GSM signal, if there is no SIM card inserted, or if the module is still busy connecting. This LED will shift to a slow blink of around once every 3 seconds when the module has completed registration to the GSM network.

APPLICATION NOTES

First Test

Before proceeding, ensure a microSIM is inserted in the SIM card slot of the Shield. A proper power source should also be prepared. Move in an are where there is a strong GSM signal. To communicate with the GSM module, AT-Hayes Commands are used. In a nutshell, these are simple text commands sent from the host microcontroller to the SIM800L. The SIM800L will analyze and then execute the command if it is executable. It shall then give back a response, if

enabled, to the host as to the result of command. By default, the host UART must be set to the following parameters:

Baud rate: 9600
Number of Data Bits: 8
Parity: None
Number of Stop Bits: 1

This is easily done in the Arduino side with the begin(9600) of the Serial class declared in the setup() function.

The very basic command is the AT. From the Arduino side, send the chars 'A' then 'T' followed by Carriage Return '\r' and Line Feed '\n'. Or simply use the println() function. When this is received by the SIM800L, it replies with 'O', 'K','\r','\n'. This is a simple test command to check if the host microcontroller is able to communicate correctly with the SIM800L. See the example code below.

For this example, we will not be communicating with the GSM network, therefore we could use the same power as that of the Arduino.

Procedures for the First Test

1. Mate the Saleng GSM shield to the Arduino UNO/Leonardo or similar board.
2. Set the micro jumer at JP2 to VIN.
3. Connect a 1A / 6-12V power supply to the Arduino's DC jack.
4. Set JP1 in the ALWAYS ON position
5. Set JP5 & JP6 in the SW-SER position.
6. Next, Upload the code below:

```
#include <SoftwareSerial.h>
SoftwareSerial salengGsm(2,3);

void setup() {
  Serial.begin(9600);
  salengGsm.begin(9600);
}

void loop() {
  if(Serial.available())
    salengGsm.write(Serial.read());
  if(salengGsm.available())
    Serial.write(salengGsm.read());
}
```

7. Once uploaded, go to Tools>Serial Monitor.
8. Set the Serial Monitor window as in the image below.

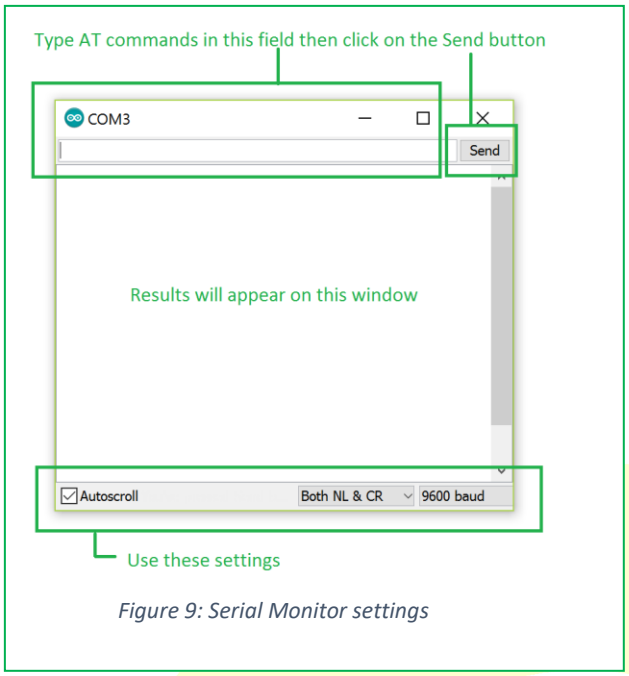


Figure 9: Serial Monitor settings

9. On the send field at the top of the serial monitor, type AT. CR and NL are automatically appended to AT since this was enabled in the serial monitor options.

10. You should get an “OK” reply on screen as show below.

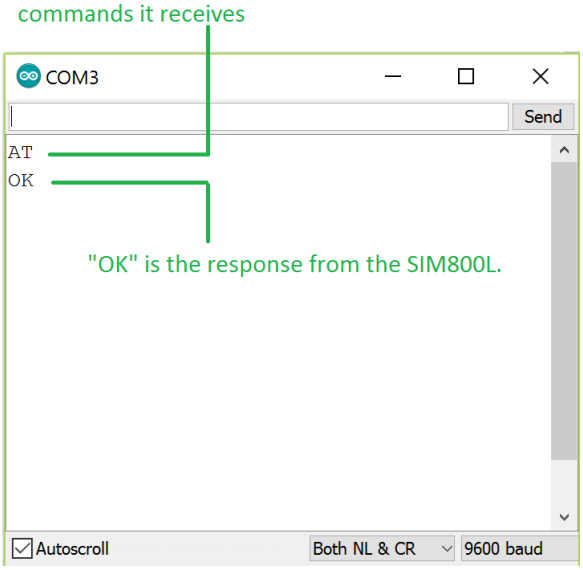


Figure 10: AT with an OK response

11. If you receive the “OK” then everything should be in order. You are now ready for your application.
12. Do not forget to return the JP2 jumper to the EXT position should you wish to use an external power supply.

First test with Arduino Mega 2560

The difference with the mega is that it has 4 UART / “hardware serial” ports available for use. Unless you have a compelling reason to use the Software serial on the mega, then it is best to use hardware serial. Do note that pin 2 on the mega can not be used with serial port. Only certain pins may be used.
To connect serial1, to the Saleng GSM shield, remove the micro jumpers on JP5 and JP6. Use 2 male-female connector to connect the middle pin of JP6 to RX1 of the Arduino Mega and middle pin of JP5 to TX1 of the Arduino Mega.

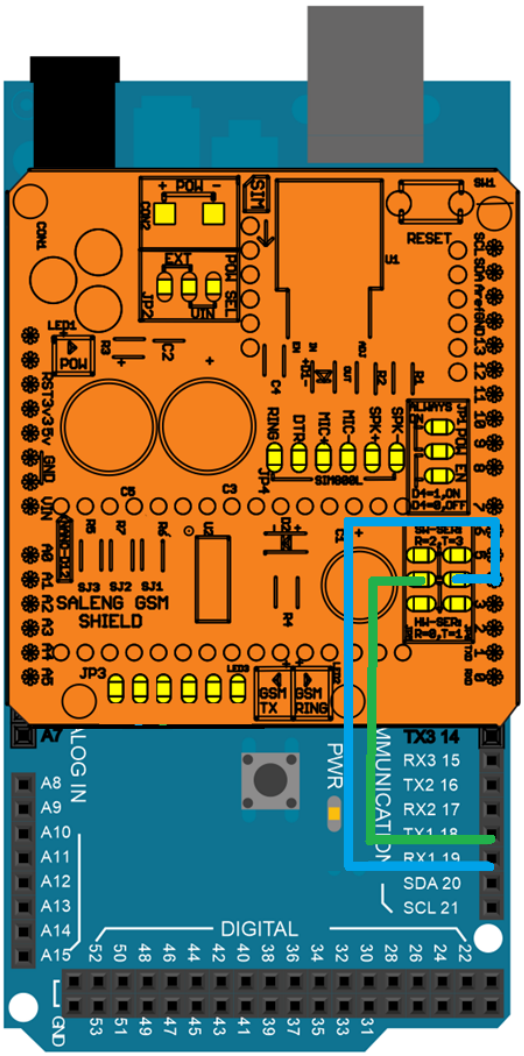


Figure 11: First test with the Arduino Mega 2560

The test code will slightly change since a hardware serial port will now be used. Upload the following code and then refer to the previous section for the procedures in performing the first test on the Arduino Mega.

```
void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
}

void loop() {
  if(Serial.available())
    Serial1.write(Serial.read());
  if(Serial1.available())
    Serial.write(Serial1.read());
}
```

The Saleng GSM Library

SMS applications may be better off using the Saleng GSM Library. It works specifically for the Saleng GSM Shield and was created to simplify SMS transmission and reception.

Download the library from the Layad Circuits Github page: **https://github.com/layadcircuits/Saleng-GSM**

Once the **LayadCircuits_SalengGSM.zip** file is downloaded, open the Arduino IDE and go to Sketch>Include Library>Add .ZIP library and browse over the .zip file you downloaded. Click Open. Close the Arduino IDE and relaunch it. Now you area ready to use the library.

Included in the library is a basic example of how to send and receive an SMS. Below is a copy of the example code. The code requires that the jumpers at JP5 & JP6 are set to the SW-SER position and the jumper at JP1 at the ALWAYS ON position. JP2 may be at any setting. Because this code does not communicate frequently, it may work with JP2 in VIN and with a 7-12V / 1A power supply at the Arduino DC jack.

```
#include <LayadCircuits_SalengGSM.h>
#include <SoftwareSerial.h>

SoftwareSerial gsmSerial(2,3);
// if you are using Arduino Mega or if you plan to use Serial 0 (pins D0 and D1), use the next line instead
// #define mySerial Serial1 // define as Serial, Serial1, Serial2 or Serial3
LayadCircuits_SalengGSM salengGSM = LayadCircuits_SalengGSM(&gsmSerial);

void setup()
{
  salengGSM.begin(9600); // this is the default baud rate
  Serial.begin(9600);
  Serial.print(F("Preparing Saleng GSM Shield.Pls wait for 10 seconds..."));
  delay(10000); // allow 10 seconds for modem to boot up and register
  salengGSM.initSalengGSM();
  Serial.println(F("Done"));
  salengGSM.sendSMS("09164428565","Hi, this is a test SMS from the Layad Circuits' Saleng GSM Shield. Have a
nice day!");
  Serial.println(F("An SMS has been sent out."));
  Serial.println(F("Send an SMS to the phone number of the SIM card and see the message on screen."));
}

void loop()
{
  salengGSM.smsMachine(); // we need to pass here as fast as we can. this allows for non-blocking SMS
transmission
  if(salengGSM.isSMSavailable()) // we also need to pass here as frequent as possible to check for incoming
messages
  {
     salengGSM.readSMS(); // updates the read flag
     Serial.print("Sender=");
     Serial.println(salengGSM.smsSender);
     Serial.print("Whole Message=");
     Serial.println(salengGSM.smsRxMsg); // if we receive an SMS, print the contents of the receive buffer
  }
}
```

The code above sends an SMS to the number specified in the first parameter of `salengGSM.sendSMS()`. This SMS is sent around 10 seconds from powerup. This 10 second delay is needed to allow the SIM800 module to fully bootup and register into the GSM network defined by the SIM card.

After it sends the test SMS, it will display any message it receives on the Serial Monitor (set to 9600 baud). All the data fields received will be buffered into the `.smsRxMsg` array and is unprocessed by the library. The user may then extract the information needed from the array. A comma separated message should be seen on screen. It shall contain the following fields:

- Phone number of the sender
- The Date and Time from the Network
- The contents of the SMS message
- Others

The phone number of sender has been extracted within the library. The last sender's phone number is stored in the variable `.smsRxMsg` .

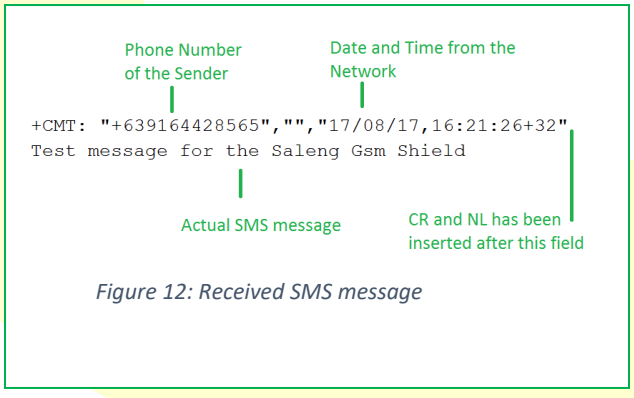


Figure 12: Received SMS message

Mini Project: An SMS remote control

Objectives:
This tiny project aims to demonstrate how a user can utilize the Saleng GSM shield to control an AC load such as a light bulb.

Hardware:

- Saleng UNO or Arduino UNO
- Saleng GSM Shield
- 7-12V / 2A power supply adaptor with 2.1mm DC plug
- A single channel relay module
- A 220V light bulb 100W or less
- A bulb receptacle
- An AC plug
- 1-2m of AC wire
- 3 pieces Male-Female connecting wires

Saleng GSM shield Jumper Settings:
Set the microjumpers jumpers to the following configurations

JUMPER LABEL	JUMPER SETTING
JP2	ALWAYS ON
JP1	VIN
JP5 & JP6	SW-SER (D2,D3)

Wiring:

Mate the Saleng GSM shield and the Arduino/Saleng UNO. They shall be powered from the same power supply adaptor. The lightbulb should be wired up as follows using the relay module's NO (Normally Open) and COM (Common) terminals. The control pin of the relay (IN) shall be controlled by the Arduino's pin D8. Refer to the following wiring diagram.

For safety, use appropriate wiring for the AC part of the circuit. A standard flat AC cable or single solid/stranded cables of gauge 18 or close to this may be used. The input side of the relay may use smaller wires such as gauge 22-24 connecting wires. Care should be exercised in handling the relay and AC section, the exposed pins of the relay on its terminal block and at the bottom side of the relay module are live and should be insulated properly.

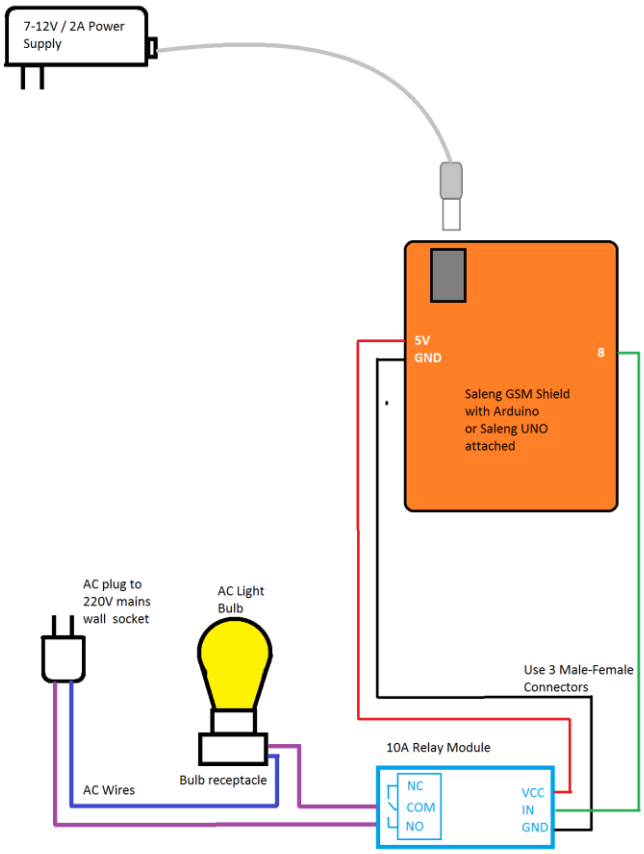


Figure 12: SMS remote control wiring diagram

Operation:

Once the hardware has been properly wired and the code uploaded, power up the device and wait for 10 seconds. Then send the words LIGHT ON to the SIM card on the Saleng GSM Shield via SMS. When the message is received, the Arduino checks for the keywords and then energizes the relay and therefore turns on the relay. Send the words LIGHT OFF to turn the light bulb off.

Application Hints:

The Saleng GSM Shield exposes the RI signal on the pin headers JP3 and JP4. One may use this pin to detect if an incoming call is received which will in turn toggle the bulb on and off. This will avoid additional costs of sending SMS. Instead, a call is initiated by the remote controlling phone.

Arduino Sketch:

The code for this project is rather simple. Our concept is to turn ON the bulb via the relay when the words LIGHT ON are received via SMS. We turn the bulb OFF with the words LIGHT OFF. We shall utilize the library and modify its example code as in the following code.

```
#include <LayadCircuits_SalengGSM.h>
#include <SoftwareSerial.h>
SoftwareSerial gsmSerial(2,3);
LayadCircuits_SalengGSM salengGSM = LayadCircuits_SalengGSM(&gsmSerial);

byte relayPin = 8;

void setup()
{
  pinMode(relayPin,OUTPUT);
  digitalWrite(relayPin,HIGH); //initially Off
  salengGSM.begin(9600); // this is the default baud rate
  Serial.begin(9600);
  Serial.print(F("Preparing Saleng GSM Shield.Pls wait 10s..."));
  delay(10000); // allow 10 seconds for modem to boot up and register
  salengGSM.initSalengGSM();
  Serial.println(F("Done"));
  Serial.println(F("Send LIGHT ON to turn on the bulb"));
  Serial.println(F("Send LIGHT OFF to turn off the bulb"));
}

void loop()
{
  salengGSM.smsMachine();
  if(salengGSM.receiveSMS())
  {
     if(strstr(salengGSM.smsRxMsg,"LIGHT ON")) digitalWrite(relayPin,LOW);
     else if(strstr(salengGSM.smsRxMsg,"LIGHT OFF")) digitalWrite(relayPin,HIGH);
  }
}
```

WHY SALENG GSM SHIELD?

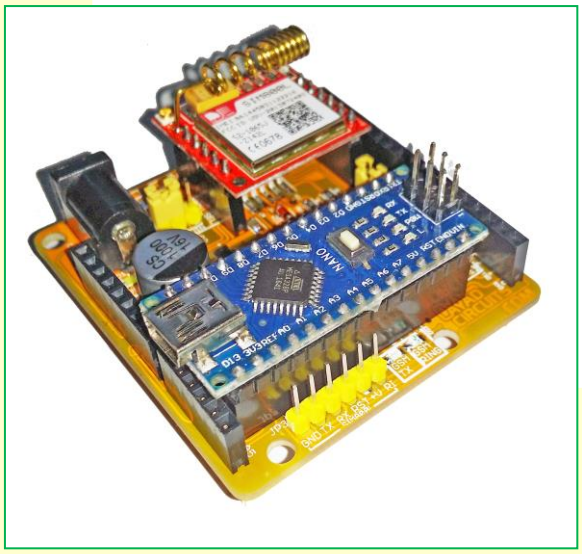
Several projects and years of experience in the usage of GSM modules in both hobby and professional applications has provided Layad Circuits the motivation to design a very easy to use GSM shield with set of features that cater to both novice and advanced users alike. Here are some of the unique features of the Saleng GSM shield that makes it one of the best GSM shield available to date.

- **Selectable Power Source** – JP2 allows the user to change between an external power and internal power.
- **May share the same power source for both Arduino and Shield** - Selecting the internal option (VIN) of the JP2 allows the sharing of the same power supply for the Arduino and GSM shield
- **Extra Pads for Power** – The PCB has a pads labeled CON2 beside the DC jack to give users the option to use a 5mm terminal block as power connector instead of the 2.1mm DC jack. This may be handy when using power supplies without readily available DC plugs (e.g. batteries, bench power supplies, etc..)

Figure 13: Add a terminal block as power connector

- **Uses the latest SIM800L module** – the SIM300 and SIM900 are older products and have been replaced by the SIM800. The shield uses the latest version.
- **Large buffer capacitors** – these capacitors have been selected to help sustain power during sudden current surges. Some other shields and modules use smaller values ,or none at all, that may cause unwanted resets of the module during heavy communications of high current drains.
- **Complete circuit** – while the SIM800L core module, and other similar modules, is a nice little board that may be used directly, it requires a non-standard power source of between 3.3V and 4.3V / 2A , logic level shifters for the module's UART and protection for the circuit. We have fried several of these modules while working with them. A simple reverse polarity can damage the module. These issues are eliminated with the Saleng GSM shield's complete circuitry. All that is need is to apply power and start coding.
- **True Hardware Reset** – The SIM800L may be awaken with the PWR_KEY pin internal to the chip. However, this requires a specifically timed pulse. Sending the same pulse again puts the module back to sleep mode. This may cause confusion when the host starts generating the pulse upon reset/power up which happens often during development (e.g. when uploading an Arduino code, the pulse generated in setup() may put the module in sleep mode when the GSM module is initially awake). The Saleng GSM shield provides a true reset, that is, an actual removal of power, by simply toggling the Arduino pin D4 with JP1 in the D4 position. This is also useful in completely resetting the GSM chip when an unwanted condition occurs (e.g. the SIM800L locks up)
- **Option to boot up immediately upon power** – Most SIMxxx shields have a power button routed to the PWR KEY pin of the GSM module. To wake the module up, apply power and then manually press and hold the button for a period of time before it turns on. This may be a problem when the device is deployed on the field without the benefit of human intervention. Simply set the JP1 to the ALWAYS ON position.
- **SIM800L is completely replaceable** – In the unlikely event of damage to the SIM800L or if it is wanted for testing, the SIM800L core module is completely replaceable. Simple insert a new SIM800L core board on the shield.
- **Selectable Software or Hardware Serial** – JP5 & JP6 easily allows the user to use either the hardware serial on pins D0 and D1 or software serial using D2 and D3
- **Exposed SIM800L serial (UART)** pins – the SIM800L has its communications interface routed on the header JP3. This allows the user to disable HW or SW serial by removing the microjumpers on JP5 & JP6 and instead wiring TX and RX on JP3 manually to the desired connection. This is useful in the case of the Arduino Mega serial1 – serial3 which are accessed by wires. This is also helpful when using another host microcontroller besides the Arduino.
- **Exposed Audio signals** – both microphone and speaker outputs are available in this Shield via JP4. This makes it optional to use or not use the audio signals as most projects use SMS/GPRS which do not necessarily use the audio signals.

- **Ring Indicator Signal** – the RI signal is available from either JP3 or JP4. This makes it easy for the user to access this and use it. It even has its own LED labeled GSM RING so that the signal is instantly visualized.
- **Small Form Factor** – The shield has a smaller form factor than the standard Arduino board. The shield only connects all the headers and is almost 1cm shorter than the Arduino board installed at the bottom. This gives access to the Arduino reset button and avoids contact with the USB connector metal casing of the Arduino which is grounded.
- **Arduino Nano Compatible** – Most other shields only work with R3 boards such as the Arduino Uno, Mega, Leonardo, Due, Zero, and similar boards of the same form factor. Unique to the Saleng GSM Shield is the availability of connection pads to be able to accept the Arduino Nano. Just install two sets of 15pin



female headers into the pads provided. To properly orient the Nano, one of the pads is labeled as NANO-D12. The pins of the Nano available to the Uno shield standard headers are connected. Hence, the user may still access the Nano pins from the Uno headers.

- **Designed and Made in the Philippines** – specifically, in Baguio City. The Saleng GSM Shield is Filipino designed and made. By purchasing this shield, you help promote embedded systems technology awareness in the Philippines and bring support for your local innovators.

IMPORTANT NOTICE

Layad Circuits Electronics Engineering Supplies & Services (Layad Circuits) reserves the right to make corrections, enhancements, improvements and other changes to its products, services and documentations, and to discontinue any product or service. Buyers or clients should obtain the latest relevant information before placing orders and should verify that such information is current and complete. Additional terms may apply to the use or sale of Layad Circuits products and services.

Reproduction of significant portions of Layad Circuits information in Layad Circuits datasheets or user guides is permissible only if reproduction is without alteration, displays the Layad Circuits logo and is accompanied by all associated warranties, conditions, limitations, and notices. Layad Circuits is not responsible or liable for such reproduced documentation. Information of third parties may be subject to additional restrictions. Resale of Layad Circuits products or services with statements different from or beyond the parameters stated by Layad Circuits for that product or service voids all express and any implied warranties for the associated Layad Circuits product or service. Layad Circuits is not responsible or liable for any such statements.

Buyers and others who are developing systems that incorporate Layad Circuits products (collectively, "Designers") understand and agree that Designers remain responsible for using their independent analysis, evaluation and judgment in designing their applications and that Designers have full and exclusive responsibility to assure the safety of Designers' applications and compliance of their applications (and of all Layad Circuits products used in or for Designers' applications) with all applicable regulations, laws and other applicable requirements. Designer represents that, with respect to their applications, Designer has all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. Designer agrees that prior to using or distributing any applications that include Layad Circuits products, Designer will thoroughly test such applications and the functionality of such Layad Circuits products as used in such applications. Layad Circuits' provision of technical, application or other design advice, quality characterization, reliability data or other services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "Layad Circuits Resources") are intended to assist designers who are developing applications that incorporate Layad Circuits products; by downloading, accessing or using Layad Circuits Resources in any way, Designer (individually or, if Designer is acting on behalf of a company, Designer's company) agrees to use any particular Layad Circuits Resource solely for this purpose and subject to the terms of this Notice.

Layad Circuits' provision of Layad Circuits Resources does not expand or otherwise alter Layad Circuits' applicable published warranties or warranty disclaimers for Layad Circuits products, and no additional obligations or liabilities arise from Layad Circuits providing such Layad Circuits Resources.

Layad Circuits reserves the right to make corrections, enhancements, improvements and other changes to its Layad Circuits Resources. Layad Circuits has not conducted any testing other than that specifically described in the published documentation for a particular Layad Circuits Resource.

NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER LAYAD CIRCUITS INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF LAYAD CIRCUITS OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which Layad Circuits products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of Layad Circuits Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from Layad Circuits under the patents or other intellectual property of Layad Circuits . Layad Circuits RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. LAYAD CIRCUITS DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS. LAYAD CIRCUITS SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY DESIGNER AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN LAYAD CIRCUITS RESOURCES OR OTHERWISE. IN NO EVENT SHALL LAYAD CIRCUITS BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF LAYAD CIRCUITS RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER LAYAD CIRCUITS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Unless Layad Circuits has explicitly designated an individual product as meeting the requirements of a particular industry standard , Layad Circuits is not responsible for any failure to meet such industry standard requirements. Where Layad Circuits specifically promotes products as facilitating functional safety or as compliant with industry functional safety standards, such products are intended to help enable customers to design and create their own applications that meet applicable functional safety standards and requirements. Using products in an application does not by itself establish any safety features in the application. Designers must ensure compliance with safety-related requirements and standards applicable to their applications. Designer may NOT use any Layad Circuits products in life-critical applications. Life-critical medical equipment is medical equipment where failure of such equipment would cause serious bodily injury or death (e.g., life support, pacemakers, defibrillators, heart pumps, neurostimulators, and implantables). Designers agree that it has the necessary expertise to select the product with the appropriate qualification designation for their applications and that proper product selection is at Designers' own risk. Designers are solely responsible for compliance with all legal and regulatory requirements in connection with such selection. Designer will fully indemnify Layad Circuits and its representatives against any damages, costs, losses, and/or liabilities arising out of Designer's noncompliance with the terms and provisions of this Notice.

Revision: v1.1/ 09 Sep 2017 /CDM

Layad Circuits Electronics Engineering Supplies & Services, B314 Lopez Bldg., Session Rd. cor. Assumption Rd., Baguio City, Philippines
General inquiries: info@layadcircuits.com Sales: sales@layadcircuits.com FB: facebook.com/layadcircuits Mobile: +639164428565

Revision: v1.1/ 09 Sep 2017 /CDM

Layad Circuits Electronics Engineering Supplies & Services, B314 Lopez Bldg., Session Rd. cor. Assumption Rd., Baguio City, Philippines
General inquiries: info@layadcircuits.com Sales: sales@layadcircuits.com FB: facebook.com/layadcircuits Mobile: +639164428565