SeeedStudio GSM SIM900 Shield with Arduino using Ting.com SMS

03/2016 R.Grokett WA4EFH

Overview

I wanted a way to track activity in building that has power and cell service but no WiFi service, so I turned to a SeeedStudio GSM GPRS SIM900 Arduino shield. I only wanted a single message (email or SMS) sent each morning with a simple report on the movement activity detected over the past 24 hours. The Arduino collects motion sensor data totaling hourly.

```
Activity seen 123 times
01/12 09:00 – 12
01/12 11:00 – 42
01/12 14:00 – 21
01/12 19:00 – 34
01/12 22:00 – 8
01/13 02:00 – 4
01/13 07:00 – 11
```

Example of SMS msg

Network Costs

I could have had DSL or Cable Internet installed, but at typically $40/month just to send a few bytes each day seemed a bit much. So, I decided to look into using cellular data. Most cell providers have yet to adopt "real" pricing for very low volume IoT (Internet of Things) type devices. They tend to think that their only customers are smart phone users with high voice/data requirements and price accordingly.

Since this project only needs about 1 kilobyte per day(!), high speed, high volume is NOT a requirement. But low monthly cost is! I found a very good provider "Ting" that has a bottom end plan for small devices starting as little as $6/month. You can get a SIM card from them for the SIM900 for about $9 one-time with no "activation" fee. If you use the link https://z2q89r47d4e.ting.com/ you (and I) can get a bill credit or discount. See below for steps.

SMS vs HTTP

This project uses SMS messaging instead of HTTP, since I wanted an alert on my phone. It is possible to convert to use HTTP, but note that older SIM900's don't handle HTTP POST or HTTPS very well. HTTP GET works, though.

Hardware Costs

There are a number of different versions of the GSM shields available from several vendors, but any that use the sim900 module are similar in both function and price around $60. But, you can sometimes find the older versions at much lower price. I ended up with the SeeedStudio version 1.0 from Radio Shack for about $20. The cost of the GSM module, even new, is about the same as a DSL or Cable Modem. Check eBay for cheap older versions, if you like.

NOTE: MOST SIM900 UNITS DO NOT COME WITH SUPPORT FOR HTTPS SSL. SO IF YOU NEED THAT, BE WILLING TO TRY TO UPGRADE THE FIRMWARE: http://seeedstudio.com/wiki/images/3/37/SIM_900_AGPS_instructions.zip

THIS PROJECT USES SMS, SO HTTPS IS NOT REQUIRED.

Other Uses

This system can be adapted to other types of sensors, such as temperature, humidity, water level, gas, etc. instead of the motion sensor I used. Just replace the PIR sensor and feel free to modify the code to add your message output.

Requirements

You will need each of the following:

A GSM GPRS Arduino Shield such as from SeeedStudio. I used an older version 1.0 for this project, but others should be fine.
http://www.seeedstudio.com/wiki/GPRS_Shield_V1.0

- An Arduino Uno or similar.

- A PIR Sensor, such as from SeeedStudio, Adafruit, Parallax, etc.

-

A SIM card, such as from TING https://z2q89r47d4e.ting.com/

- Software from GITHUB: https://github.com/rgrokett/GSM_SIM900_Arduino
- USB cable and Arduino IDE version 1.6 or better
- Wires, LED & 220ohm resistor (optional), Case, 5v 1amp power supply
- Arduino IDE from https://www.arduino.cc/en/Main/Software

Setup

You can assemble and run initial tests of your Arduino and GPRS SIM900 hardware without purchasing a SIM card or service. Then once ready, you can order a SIM card and activate it. Ting delivered mine in only a couple days and activation just took a few minutes. (See below)

Plug your GPRS Shield into your Arduino following the card's directions. Note that there are several hardware settings needed for these cards. Since these vary, check the GPRS Shield's web site for details.

Again, you do NOT need a SIM card or battery for this initial setup.

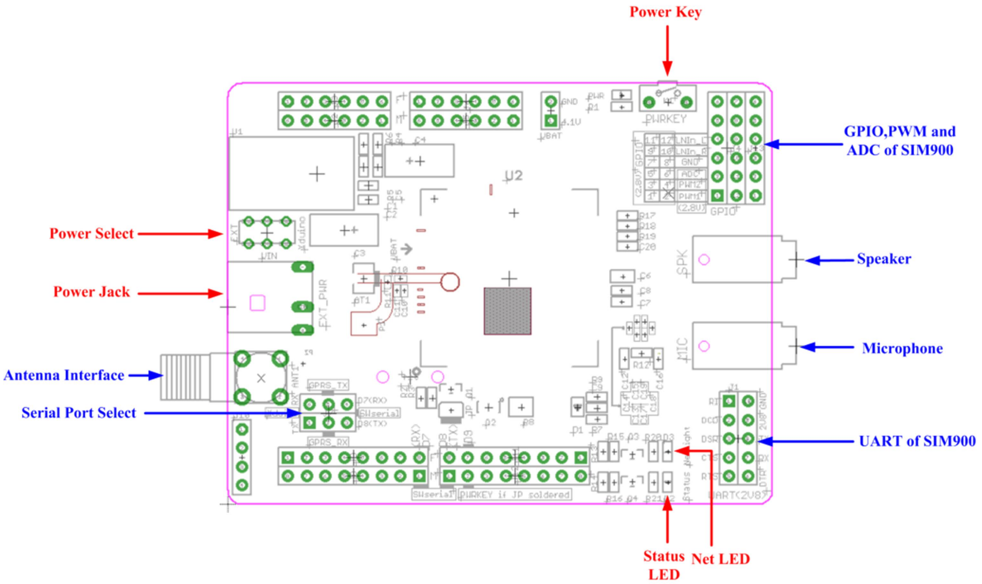
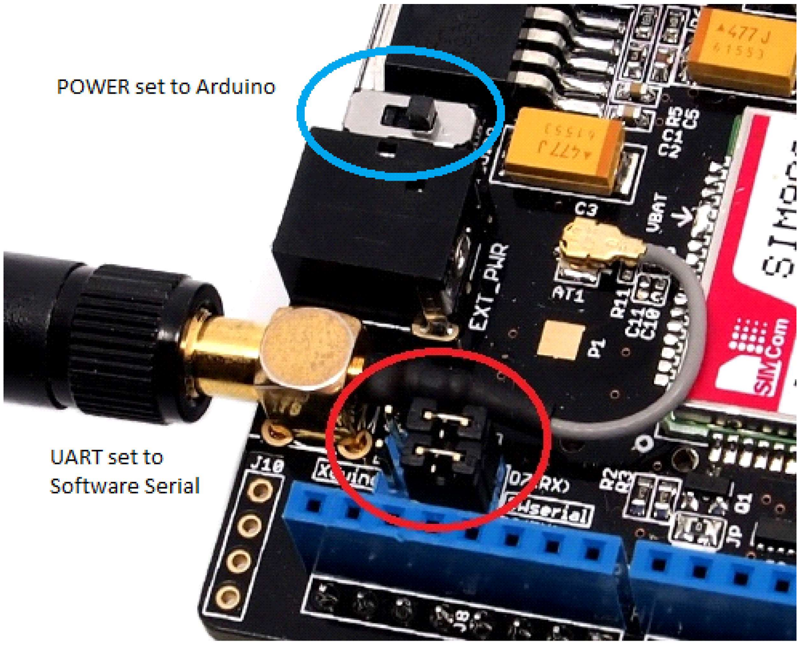


Figure from SeeedStudio Wiki for GPRS Shield V1.0

1. Be sure the POWER SELECT is set to “Arduino”
2. And the Communications UART is set to “Software Serial”



For more information on the SeeedStudio GPRS Shield, try:
http://www.seeedstudio.com/wiki/GPRS_Shield_V1.0
http://www.seeedstudio.com/wiki/GPRS_Shield_V2.0
http://www.seeedstudio.com/wiki/GPRS_Shield_V3.0

3. Attach the Antenna and verify its cable connection.

NOTE: ALL OF THESE HARDWARE STEPS ARE IN THE SEEEDSTUDIO GPRS WIKI PAGES (URLS ABOVE) IN MORE DETAIL, SO FEEL FREE TO CHECK THEM OUT!

4. Next, plug your USB cable into the Arduino and into your PC.

5. Install the Arduino IDE onto your PC (if not already installed).
   https://www.arduino.cc/en/Main/Software

6. Plug in your Arduino into a USB port on your PC and verify you can compile and load any Arduino Uno program. You will need to check the instructions for using the Arduino IDE for this part. But key is to select the correct board (Tools -> Board -> Arduino/Genuino Uno) and the correct serial port (Tools -> Port -> COM…)

   It doesn't matter what .ino program you try, as it won't affect the GSM SIM900 board and will be replaced at the next step. You are just trying to verify that your development set up is good.

Initial Testing

You can do some initial testing to be sure you can talk to the Arduino and GPRS card ok.

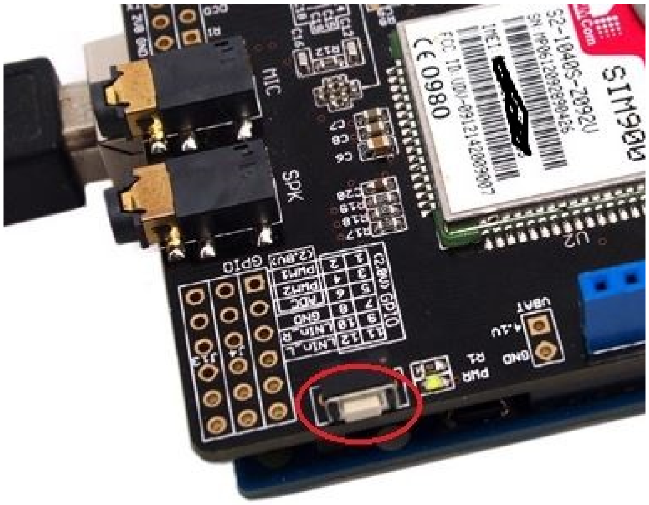
1. Download my GPRS PIR application software from GITHUB (above) and install into a directory in your Documents folder or similar. Navigate to the sketch folder and go to the SerialRelay subfolder and double-click on the SerialRelay.ino program to start the IDE.

   This program (courtesy SeeedStudio) just connects a terminal program to the SIM900 so you can issue commands to it.

2. Compile and install the SerialRelay.ino program on your PC from the IDE.

3. Start a serial monitor; either the one with the Arduino IDE (Tools -> Serial Monitor) or better, download and use their Sscom32E.exe program
   http://www.seeedstudio.com/wiki/images/b/b2/Sscom32E.zip
   SEE SEEEDSTUDIO'S GETTING STARTED SECTION OF THEIR GPRS WIKI AS MENTIONED ABOVE ON HOW TO USE THE SSCOM32E.EXE PROGRAM.

4. Set the serial program to **19200 baud 8-N-1**
5. Turn on the GPRS SIM900 board by pressing its power button for 2 seconds.



At power up, a red and green LED will light and another green LED will start flashing.

6. You should see some output print on your serial terminal screen similar to:
   ```
   RDY +CFUN: 1 +CPIN: READY Call Ready
   ```

7. If you see nothing but gibberish, type **AT+IPR=19200** to set the baud rate of the SIM900 and disconnect/reconnect again.

8. Try typing **AT** and return. You should get an **OK** back.
   You can also try typing **ATI** to see some version info.
   Many other commands are available:
   http://www.seeedstudio.com/wiki/images/7/72/AT_Commands_v1.11.pdf

9. If you are seeing responses to these AT commands, then your hardware is good.
   Next is to get a SIM card and install it.

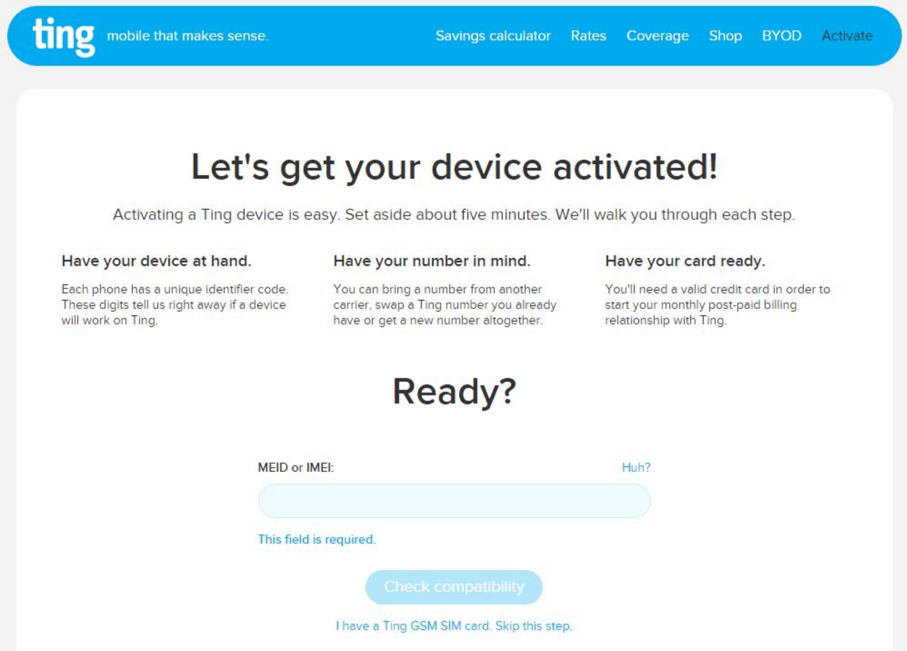
SIM Card

As mentioned earlier, TING is a good low cost provider of the SIM card and GSM service for the SIM900 card.

1. Go to Ting.com (Use the link in Parts list to get some bill credits)
2. Select BYOD as you have your device!
3. Enter the IMEI number as written on the top of your SIM900 chip. This verifies that they support your device in your area. If there's no GSM service… yikes!
4. You can then order the SIM card (currently $9). They use a "universal" card, so it should work just fine for your SIM900.

NOTE THAT THIS DOES NOT ACTIVATE YOUR SERVICE. SO YOU CAN SIT ON THE CARD FOR UP TO A YEAR OR TWO WITH NO MONTHLY CHARGES AND ACTIVATE LATER.

5. Once you get the card and are ready to activate, return to the Ting.com URL and click on Activate menu option.



Let's get your device activated!

Activating a Ting device is easy. Set aside about five minutes. We'll walk you through each step.

Have your device at hand.
Each phone has a unique identifier code. These digits tell us right away if a device will work on Ting.

Have your number in mind.
You can bring a number from another carrier, swap a Ting number you already have or get a new number altogether.

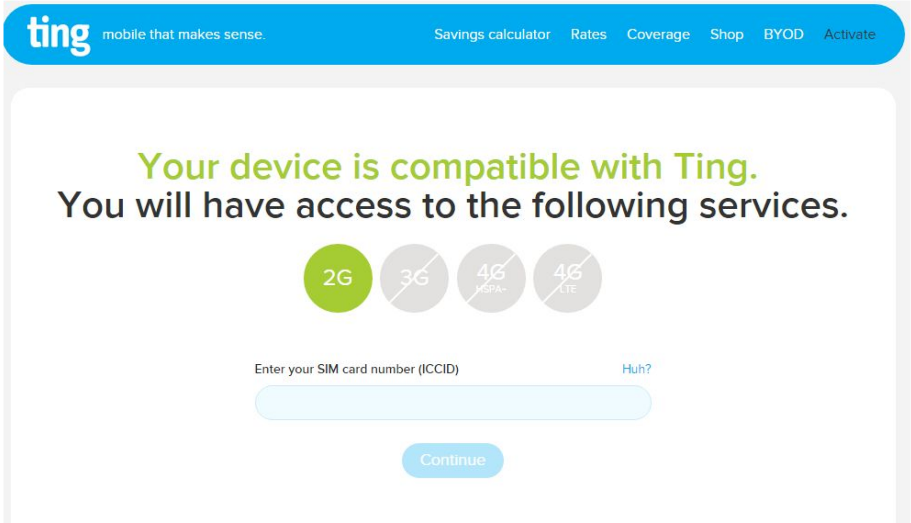
Have your card ready.
You'll need a valid credit card in order to start your monthly post-paid billing relationship with Ting.

Ready?

MEID or IMEI:

This field is required.

6. You will step thru a series of simple screens, again entering your IMEI number. You next may need to enter your SIM card's ICCID number (on the bar code), but it may already be filled in and selectable.

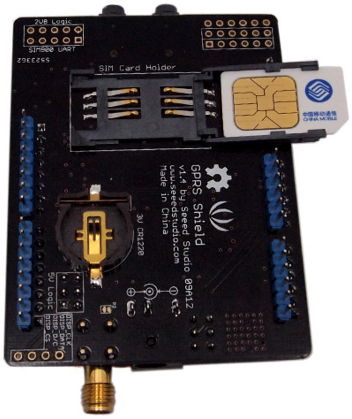


Your device is compatible with Ting.
You will have access to the following services.

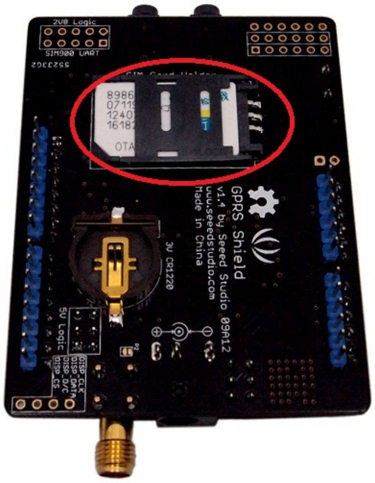
Enter your SIM card number (ICCID)

7. Unplug your Arduino and disassemble your GPRS SIM900 from your Arduino so that you can get to the SIM slot. The slot door slides backwards a tiny bit to unlock it (look close; you will see a "lock/unlock" text on its side.)

8. Swing the door open and slide the SIM card into it as shown below.



9. Close the door and slide it forward a bit to lock it in.



10. If you want to use the Real Time Clock (RTC), I suggest installing a CR1220 battery into it now. This will keep the clock accurate with the power off.
    NOTE that your cell service provider may not support setting the time automatically, so you may need to do so manually. (Described below)

11. Reinstall your SIM900 to your Arduino and plug it back into your PC. It should still have the SerialRelay program so you can just connect to it with your PC serial monitor or sscom32e.exe program again.
12. Turn on the SIM900 by pressing the button for 2 seconds.
13. Again, try the AT commands above.
14. You can check the RTC clock by entering **AT+CCLK?**
    ```
    16/02/14,15:45:00+05
    ```
15. If you see something like 01/01/01, 0:0:0+0 or other incorrect time, you can set it using:
    ```
    AT+CLTS=1
    AT+CCLK="16/02/14,15:45:00+05"   (GMT time + offset to local TZ)
    ```

16. Try **AT+CCLK?** again for hopefully the correct time!
17. The SeeedStudio GPRS wiki page shows how to manually send an SMS message to your mobile phone. Key to remember is to enter the FULL number, like: 19045551212 (country code, area code, local number) no dashes, spaces, etc.
18. If you got to here, you are in business and ready to install your application program.

Setting up your SMS & Sensor Application

Wiring the PIR Sensor

Connect the PIR Sensor to the Arduino using pins:

PIR	**ARDUINO**	**DESC**
VCC +	5V Pin	+ 5v Power
GND	GND Pin	Ground
OUT	Pin 4	Digital I/O

Wiring the external LED (optional)

Connect a 220 ohm resistor to the long lead of the LED (the positive or anode lead) and to Pin 13. Connect the shorter lead (negative or cathode) to an Arduino GND pin. This is only used to show system activity. Note that there are a number of other LEDs on the Arduino and the GSM module, so you really don’t need this unless you are enclosing the entire thing (except for the PIR sensor) in a box and want an external indicator. This LED will light every time there is motion activity detected.

PIR & SMS Software

You can now replace the SerialRelay.ino program with the example **PirGprsSMSAlert.ino** program.

You must edit a field in this program to send to your mobile phone. This program is really just an example use case. You probably want to alter to collect different sensor data and send your own message.

```
// ---- USER MODIFY ----
String MYPHONENUM = "+19045551212";   // Replace with your phone# to send texts to.
SMS needs country code

// Which hour to send message
int timeToSend = 8;    // Sends SMS msg Daily at 08:00 AM  Set between 1 & 23
```

This program detects movement and collects the sensor event counts, totaling for each hour for 24 hours and then sends an SMS message with the last few hours of data. It then resets the counts and starts again. It also sends an initial SMS message when the Arduino is reset or powered up to verify it is functioning correctly.

Compile and Upload the modified .ino script. Open the Serial Monitor. It will display startup messages and then send an initial SMS message. As you move in front of the PIR (should be good up to about 15-20 feet away), the (optional) LED will light and messages print on the serial monitor.

At the "timeToSend" time (default is 08:00 clock time), an SMS message with the date/time hour and number of movements detected for each of the past few hours will be sent.

Between sending SMS messages, the GPS SIM900 module is powered down in order to conserve power.

```
boolean savePower = true;// Conserve power by turning off SIM900 between transmissions
```

I have been using this program for several weeks now and have reliably received an SMS message every morning.

PLEASE FEEL FREE TO MODIFY THE PROGRAM TO GATHER YOUR OWN SENSOR DATA AND SMS MESSAGE!

Extra Info

NOTE THAT THE ARDUINO SOFTWARESERIAL.H FILE LIMITS THE SMS MESSAGE TO NO MORE THAN 64 CHARACTERS! MODIFY IT TO INCREASE.

C:\Program Files (x86)\Arduino\hardware\arduino\avr\libraries\SoftwareSerial\SoftwareSerial.h should increase it for SMS messages longer than 64 bytes. BUT, Arduino Uno has a rather small amount of memory for such variables, so you must be sparing in increasing it!

```
// #define _SS_MAX_RX_BUFF 128 // RX buffer size //BEFORE WAS 64
```

This may not apply to newer versions, so only worry about this if your longer SMS messages are truncated or never even arrive.