CENTER FOR URBAN SCIENCE+PROGRESS

Rapid Prototyping of Urban Sensors

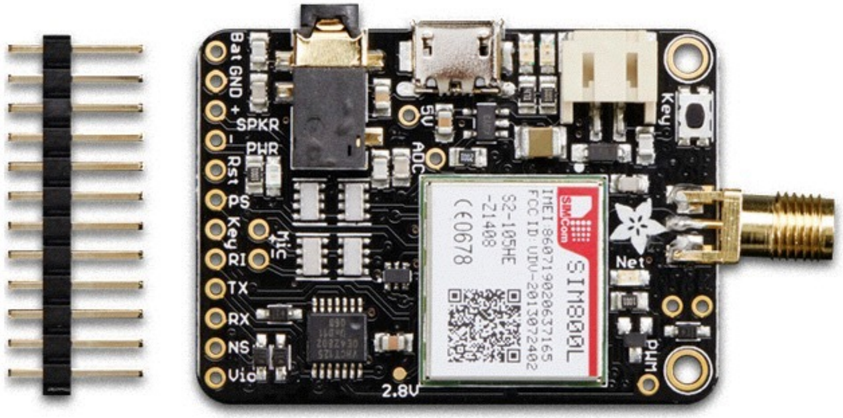
Wireless Communication with GSM

GSM MODEMS


SIMCom
SIM900
S2-1040S
FCC ID:UDV-0912142009007
IMEI:012207000003738
SN:P10609520001373
CE 0980


94V0
E99006
TYPE2 B
0951
RoHS
SIM900_V2.03

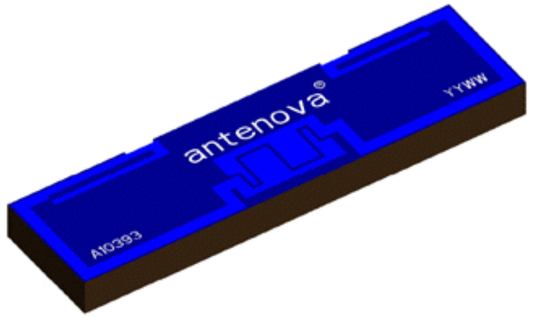

antenova
GSM SHIELD
CE0700


Bat
GND
+
SPKR
PHR
Rst
PS
Key
RI
TX
RX
NS
Vio
5V
ADC
Mic
Net
Key
PWR
2.8V
SIM800L


sparkfun.com

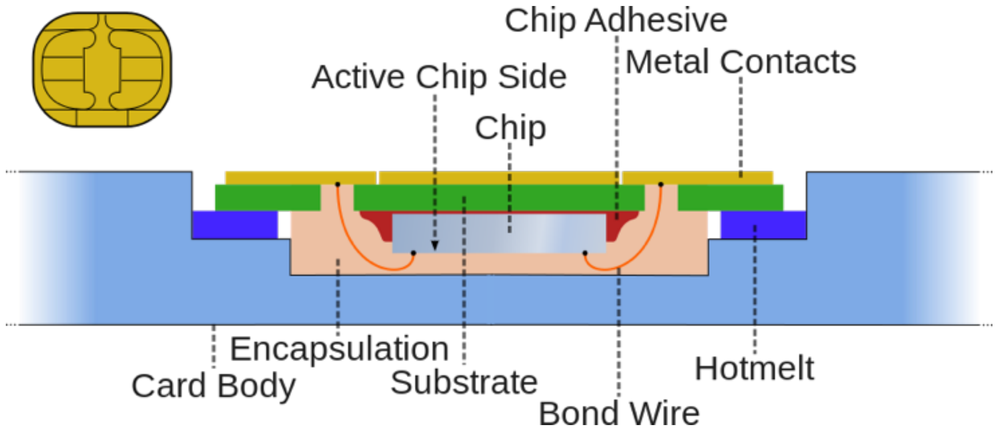
ANTENNAS

- embedded, PCB and external antennas



SUBSCRIBER IDENTITY MODULE

- **I**nternational **M**obile **S**ubscriber **I**dentifier(**IMSI**) - is the number for the account
- **I**nternational **M**obile **E**quipment **I**dentifier(**IMEI**) - a number stored on the hardware device describing it - unique to each phone. (like a serial number)



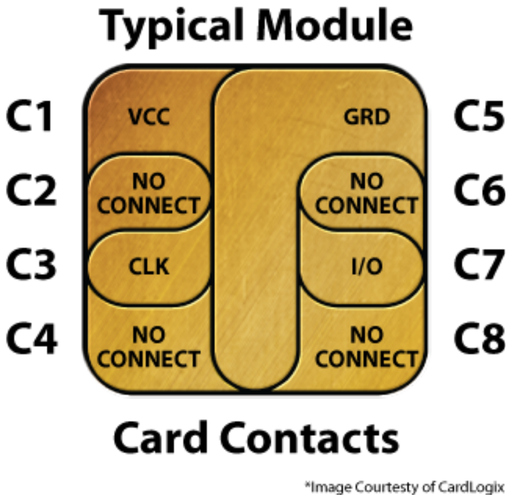


*Image Courtesty of CardLogix

EXAMPLES

Temp & Humidity Data Logger

Liquid Level Monitor

Lion Tracking Collar

Network Smoke Alarm

Netorked Utility Monitor

Bird Tracking

GETTING STARTED W/ FONA

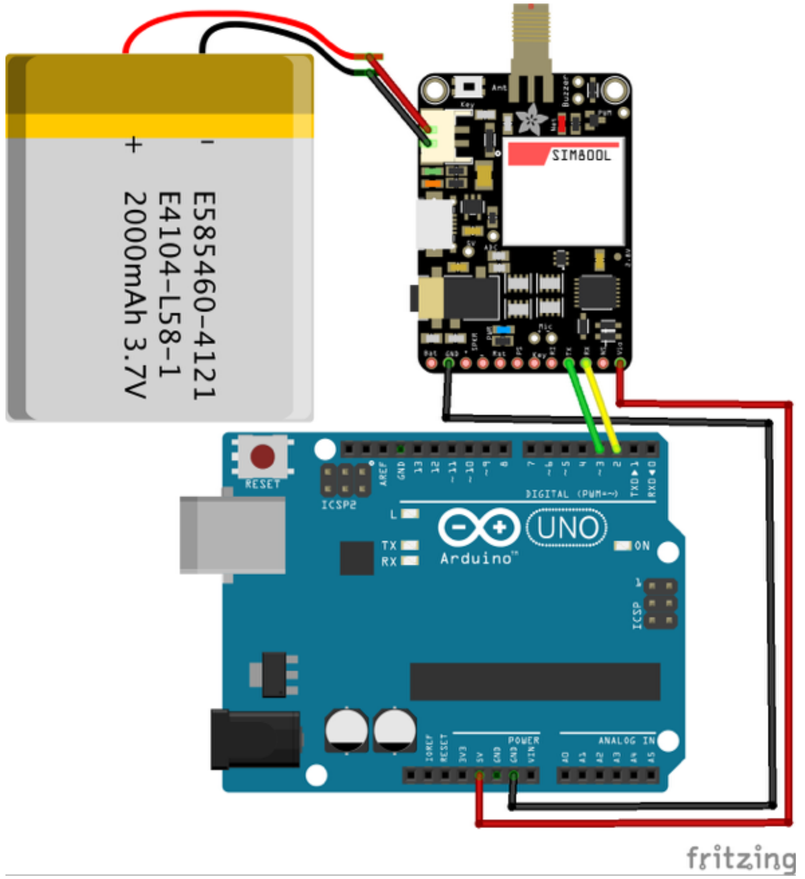
- connecting to the Fona module using the Arduino Software Serial library
- this example allows you to send commands directly to the module as well as listen to the response
- does not use the Fona GSM library

```
#include <SoftwareSerial.h>

SoftwareSerial fona = SoftwareSerial(3,2);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  fona.begin(9600);
  Serial.println("GSM Started");
}

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



AT COMMANDS

- **AT Commands** are the standard protocol to communicate with the GSM module
- Commands start with **AT+(some command)**
- These commands let you:
    - set module settings and functionality
    - getting information from the module
    - interactive with the network
- There is a standard set of commands though individual modules often have their own proprietary commands extending functionality

AT COMMAND EXAMPLES

- Start by verifying connection. Type: > **AT <return>** which should return with > **OK**

- **AT** Check to see if the module is active. Should return 'OK'
- **AT+CREG?** Is the module registered to the network?
- **AT+COPS?** What network is the module registered?
- **AT+CMGF=1** This puts the module into text mode so messages can be sent/received
- **AT+CMGS="number",129, <body of message> <hex return character '1A'>** Send a text message. When using CoolTerm, enter this command, the use 'Command+T' to bring up another window. This will let you type ASCII and Hex. Type your message, and the add '1A' as Hex. Note: When sending a message through OpenBTS, you do not need to use the 129.
- **AT+CMGL="ALL"** Lists all text messages that are on the device (or network)
- **AT+CMGR=<index>** Read SMS message at index number
- **AT+QBAND?** What band am I on?
- **AT+CIMI** Get the IMSI number from the module
- **AT+CSQ** Check the signal strength

AT COMMAND SEND SMS

Using CoolTerm, we first need to set the module into text mode:

AT+CMGF=1

We then use the command:

AT+CMGS="number", 129 <return>

You should then see a response:

> type body of the message here <return>

Finally, you need to send a control sequence to tell the module you have finished typing. This will be in HEX. To do this, simply hit **Control-Z** and then <return>. You should receive:

+CMGS: 211 OK

GO BUILD WIRELESS SENSORS