

Instrukcja dla: Icomsat v1.0 SIM900 GSM/GPRS shield for Arduino oraz dla GPRS Shield produkcji Seedstudio.



IComsat jest to shield GSM/GPRS współpracujący z Arduino oparty o moduł SIM900 firmy SIMCOM. Komunikacja z modułem SIM900 oparta jest o komendy AT (GSM 07.07 ,07.05 and SIMCOM enhanced AT Commands), moduł współpracuje z Arduino oraz Arduino Mega.

Dane podstawowe:

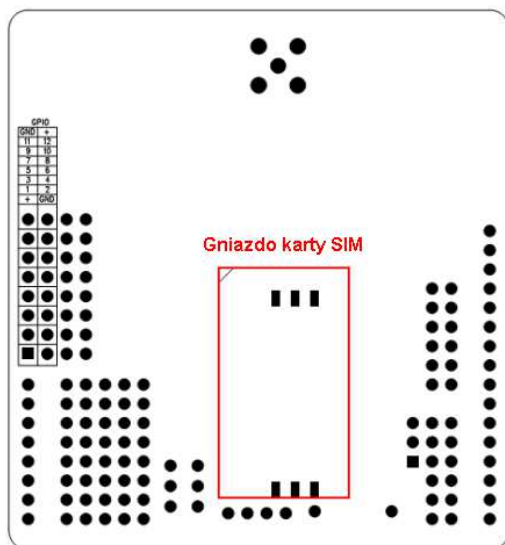
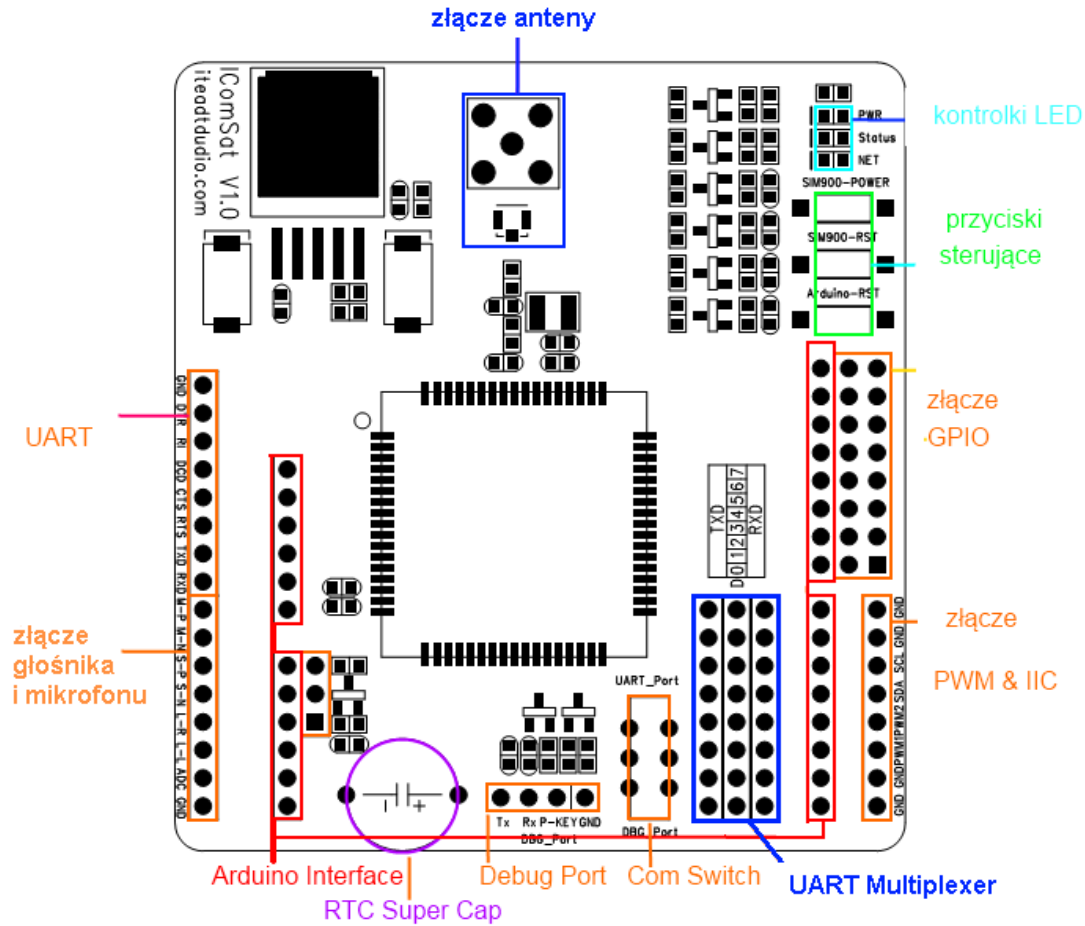
- Quad-Band 850/900/1800/1900MHz
- GPRS multi-slot calss 10/8
- GPRS mobile station class B
- Compliant to GSM phase 2/2+
- Class 4 (2W@850/900MHz)
- Class 1 (1W@1800/1900MHz)
- Control via commands (GSM 07.07, 07.05 and SIMCOM enhanced AT Commands)
- obsługa SMS
- Free serial port selection
- Wszystkie piny SIM900 wyprowadzone na złącze
- Zegar RTC podtrzymywany kondensatorem

- Temperatura pracy: $-40^{\circ}\text{C} \sim +85^{\circ}\text{C}$
- Wymiary płyty: 71,4x66x1,6mm
- Diody LED: PWR, status LED, net status LED
- Napięcie zasilania - pobierane z Arduino
- Protokół komunikacji SIM -> Arduino: UART
- ROHS: tak

Charakterystyka elektryczna:

Specification	Min	Type	Max	Unit
Power Voltage (Vlogic)	4.5	5	5.5	VDC
Power Voltage (Vsupply)	9	-	20	VDC
Input Voltage VH:	4.5	5	5.5	V
Input Voltage VL:	-0.3	0	0.5	V
Current Consumption (pulse)	-	-	2000	mA
Current Consumption (Continues)			500	mA
Baud rate		9600		bps

Opis oraz rozmieszczenie złączy:



Opis kontrolek LED:

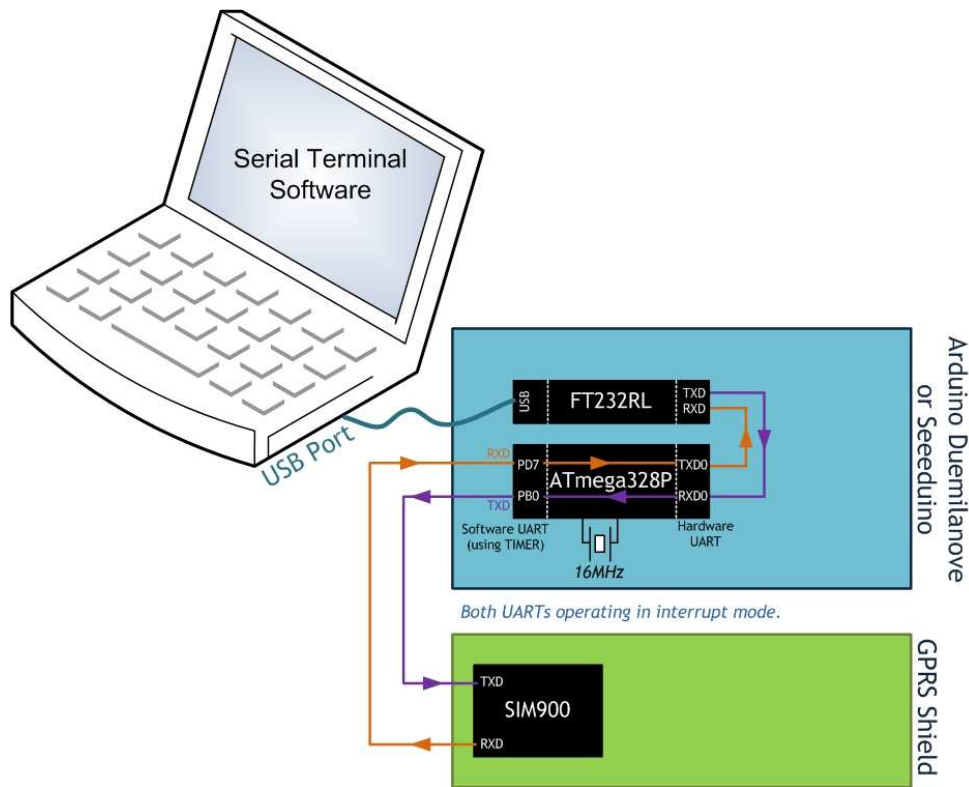
- **PWR** - napięcie zasilania
- **Status** - napięcie zasilania SIM900
- **NET:**
 - wyłączona - moduł SIM900 nie działa
 - 64ms On/800ms Off - SIM900 jest nie podłączony do sieci GSM lub jej szuka
 - 64ms On/3000ms Off - SIM900 jest podłączony do sieci GSM
 - 64ms On/300ms Off - komunikacja GPRS

Opis przycisków:

- **SIM900-POWER** - po wciśnięciu na chwilę włączamy SIM900
- **SIM900-RST** - reset SIM900
- **Arduino-RST**- reset Arduino

Test konfiguracji.

Shieldem możemy sterować za pomocą komputera PC, Arduino będzie pracowało jako "pomost" pomiędzy SIM900 a portem USB. Całość obrazuje rysunek poniżej.



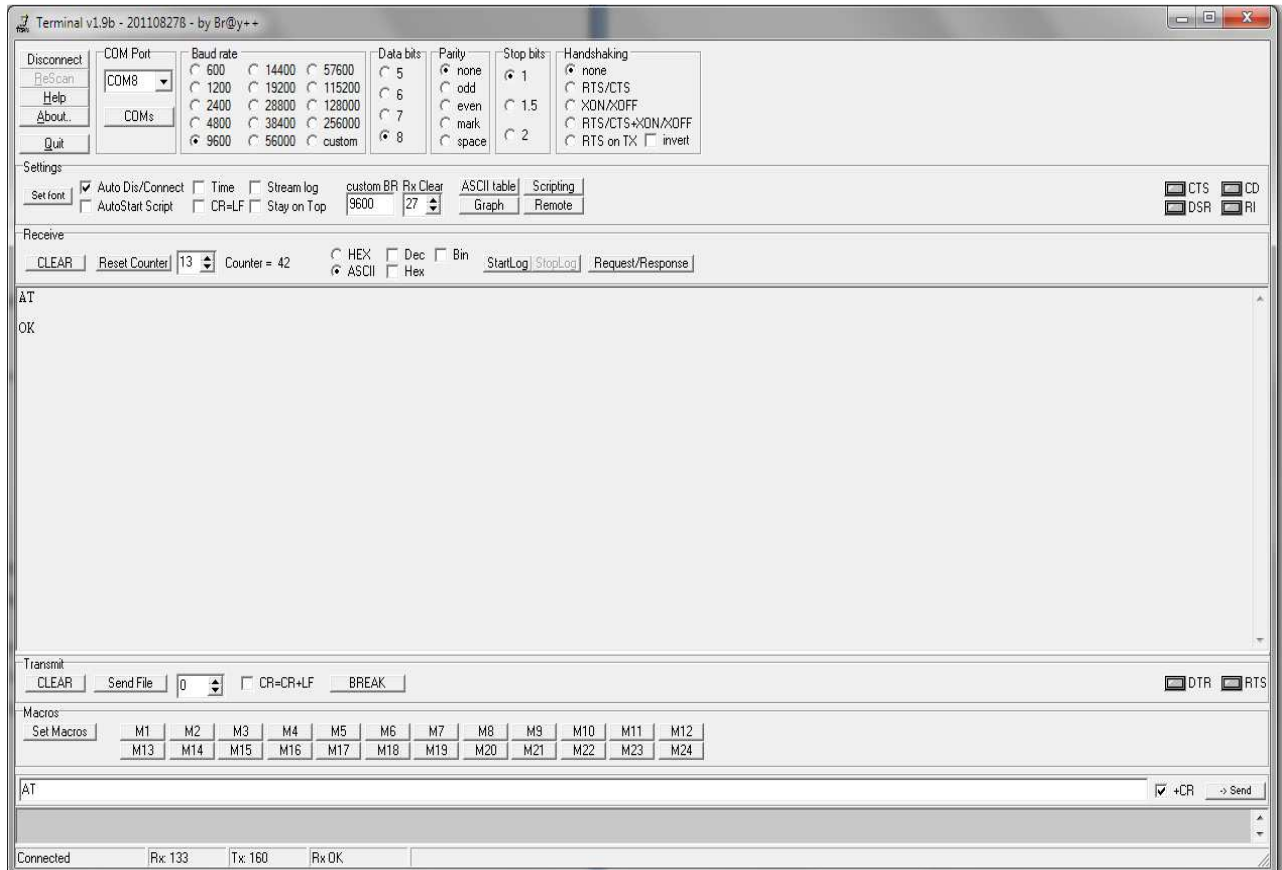
Poniżej przedstawię opis jak wysłać SMS'a.

1. W shieldzie instalujemy kartę SIM - bez kodu PIN,
2. Podłączamy antenę, shielda montujemy razem z Arduino
3. Zworki od UART Multiplexer ustawiamy na TXD 6, RDX 7 czyli tak jak na wcześniejszym rysunku.
4. Musimy mieć zainstalowane biblioteki NewSoftSerial, jeżeli mamy to programujemy Arduino programem:

```
#include <NewSoftSerial.h>
NewSoftSerial mySerial(6, 7);
void setup()
{
  mySerial.begin(9600);      // the GPRS baud rate
  Serial.begin(9600);       // the GPRS baud rate
}
void loop()
{
  if(Serial.available())
  {
    mySerial.print((unsigned char)Serial.read());
  }
  else if(mySerial.available())
  {
    Serial.print((unsigned char)mySerial.read());
  }
}
```

5. Po zaprogramowaniu Arduino wciskamy na chwilę przycisk: SIM900-POWER i czekamy aż moduł się połączy z siecią(dioda LED NET zacznie wolno migać).

6. Uruchamiamy program typu terminal i ustawiamy parametry komunikacji na: 9600-8-N-1 i wysyłamy komendę **AT** potwierdzamy klawiszem Enter. Jeżeli wszystko jest sprawne powinniśmy otrzymać odpowiedź **OK**.



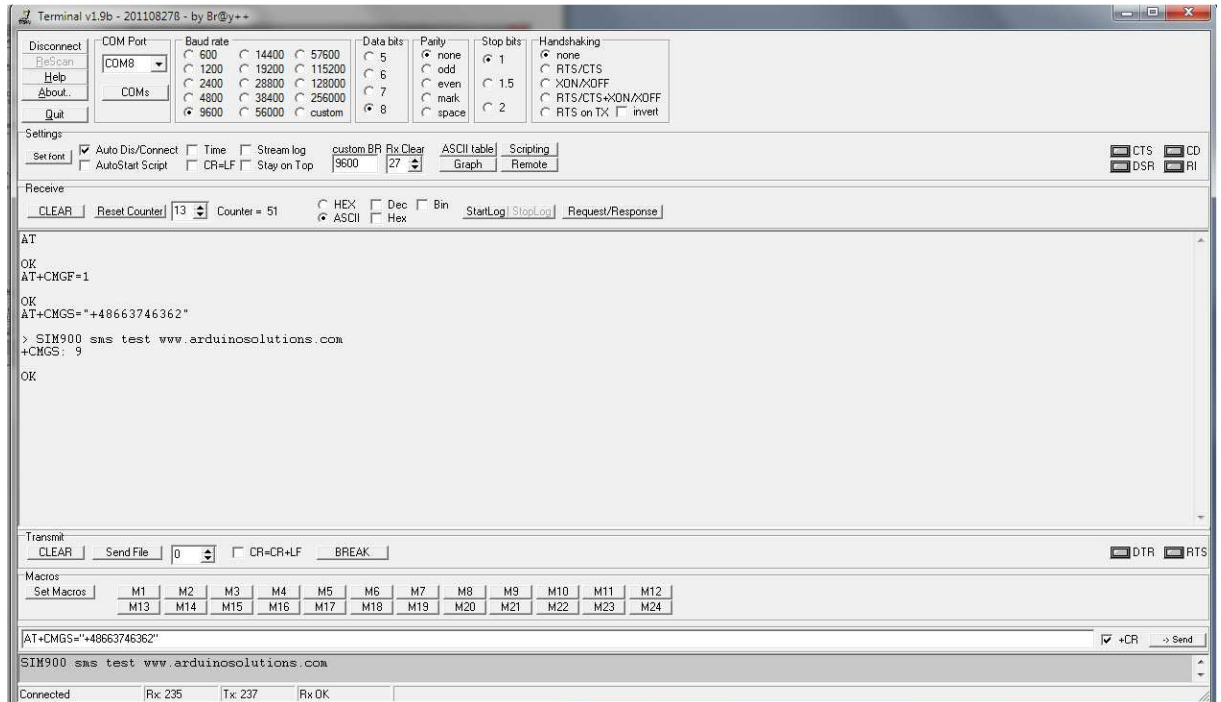
7. Wysyłamy SMS:

- SIM900 należy przełączyć w tryb tekstowy, wysyłamy komendę:

AT+CMGF=1

w odpowiedzi powinniśmy otrzymać OK

- numer telefonu pod który wysyłamy SMS, wysyłamy komendę:
AT+CMGS="+48663746362" oczywiście numer tel. zmieniamy na taki jaki chcemy, w odpowiedzi SIM900 wyśle znak ">"
 następnie wpisujemy treść smsa do wysłania, kończmy wciskając klawisze **CTRL+Z**, po chwili sms powinien dotrzeć.



```

Terminal v1.9b - 201108278 - by Br@y++
Disconnect: CDM Port: CDM8
Baud rate: 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 56000, 57600, 115200, 128000, 256000, custom
Data bits: 5, 6, 7, 8
Parity: none, odd, even, mark, space
Stop bits: 1, 1.5, 2
Handshaking: none, RTS/CTS, XON/XOFF, RTS/CTS+XON/XOFF, RTS on TX, invert

Settings:
Set font: [ ] Auto Dis/Connect [ ] Time [ ] Stream log [ ] custom BR 9600 [ ] Rx Clear [ ] ASCII table [ ] Scripting [ ]
[ ] AutoStart Script [ ] CR=LF [ ] Stay on Top [ ] CTS [ ] CD [ ] DSR [ ] RI

Receive:
CLEAR [ ] Reset Counter [ ] 13 [ ] Counter = 51 [ ] HEX [ ] Dec [ ] Bin [ ] StartLog [ ] StopLog [ ] Request/Response [ ]
[ ] ASCII [ ] Hex

Transmit:
CLEAR [ ] Send File [ ] 0 [ ] CR=CR+LF [ ] BREAK [ ] DTR [ ] RTS

Macros:
Set Macros [ ] M1 [ ] M2 [ ] M3 [ ] M4 [ ] M5 [ ] M6 [ ] M7 [ ] M8 [ ] M9 [ ] M10 [ ] M11 [ ] M12 [ ] M13 [ ] M14 [ ] M15 [ ] M16 [ ] M17 [ ] M18 [ ] M19 [ ] M20 [ ] M21 [ ] M22 [ ] M23 [ ] M24

[AT+CMGS="+48663746362"
SIM900 sms test www.arduinolutions.com
+CMGS: 9
OK

Connected Rx: 235 Tx: 237 Rx OK
  
```

- Jeżeli SMS nie dotarł warto sprawdzić czy w karcie SIM mamy poprawnie zapisany numer "SMS Message Center" w tym celu wysyłamy komendę:
AT+CSCA?
 dla karty SIM PLUS - bo na takiej był testowany shield numer ten był wpisany i otrzymujemy odpowiedź:
+CSCA: "+48601000310",145

GPRS Shield produkcji Seedstudio.



Różnice w porównaniu z Icomsat v1.0:

- posiada wyjścia typu jack służące do podłączenia głośnika oraz mikrofonu
- nie posiada kondensatora podtrzymującego RTC lecz posiada wyjście do podpięcia baterii
- port szeregowy służący do sterowania SIM900 może być obsługiwany tylko przez piny 7 oraz 8 Arduino
- posiada tylko jeden przycisk PWR KEY odpowiednik przycisku PWR w Icomsat v 1.0
- standardowo komunikacja z modułem ustawiona jest na 19200

Przy pisaniu oprogramowania musimy pamiętać o:

- port szeregowy jest obsługiwany przez piny 7 oraz 8 Arduino, więc konfiguracja NewSoftSerial powinna wyglądać następująco:
NewSoftSerial mySerial(7, 8);
- predkość transmisji pomiędzy SIM900 a Arduino wynosi 19200 bound więc konfiguracja połączenia dla NewSoftSerial powinna wyglądać następująco: **mySerial.begin(19200);**

Dla tego modułu rozbudowany opis można znaleźć na stronie producenta:
http://seeedstudio.com/wiki/index.php?title=GPRS_Shield_v0.9b

Gotowe biblioteki

Do obsługi modułów z SIM900 dostępne są biblioteki dla Arduino.
Biblioteki możemy pobrać ze strony:

<http://www.open-electronics.org/arduino-gsm-shield/3/>

Biblioteki te do komunikacji z SIM900 korzystają z pinów 4(RXD) oraz 5(TXD). Dla Icomsat to nie jest problem ponieważ wystarczy odpowiednio ustawić zworki. Dla modułu produkcji Seeedstudio musimy zmienić w pliku GSM_Shield.cpp wywołanie

NewSoftSerial mySerial(4, 5); na

NewSoftSerial mySerial(7, 8);

Przykładowe programy możemy znaleźć na stronie:

http://seeedstudio.com/wiki/index.php?title=GPRS_Shield_v0.9b

Obsługa za pomocą ARDUINO MEGA

Ponieważ mikrokontroler ATMEGA1280(2560) ma więcej niż jeden port szeregowy nie musimy używać programowej emulacji. Do komunikacji z SIM900 możemy wykorzystać jeden ze sprzętowych portów UART, w tym celu będziemy musieli wykonać połączenie przewodami pinów shielta oznaczonymi jako TXD, RXD(ICOMSAT) lub GPRS_TX,GPRS_RX(Seedstudio shield) z pinami Arduino odpowiedzialnymi za komunikację (RX1,TX1,RX2,TX2,RX3,TX3) oczywiście wybieramy tylko jedną parę np. RX3 TX3 łączymy pin TXD z RX3 oraz RXD z TX3, poprawność połączenia możemy sprawdzić za pomocą komputera PC. Przykładowy program "pomostu" pomiędzy SIM900 a PC znajduje się poniżej.

```
void setup()
{
  Serial.begin(9600);          // PC baud rate
  Serial3.begin(9600);        // the GPRS baud rate
}

void loop()
{
  if(Serial.available())
  {
    Serial3.print((unsigned char)Serial.read());
  }
  else if(Serial3.available())
  {
    Serial.print((unsigned char)Serial3.read());
  }
}
```

Jeżeli korzystamy z programów pisanych dla Arduino Duemilanove prosto zmieniamy mySerial. na Serial3.