



ElectronicWings (/)
Platforms
(/explore)

Projects
(/projects)

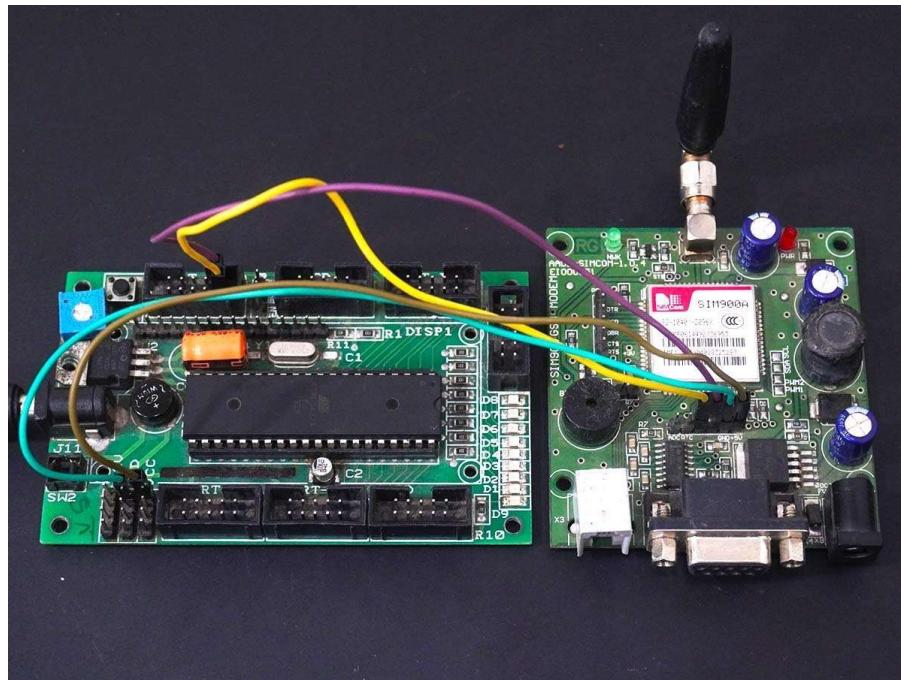
Search

+ Project (/publish/project) BorisDmitrenko

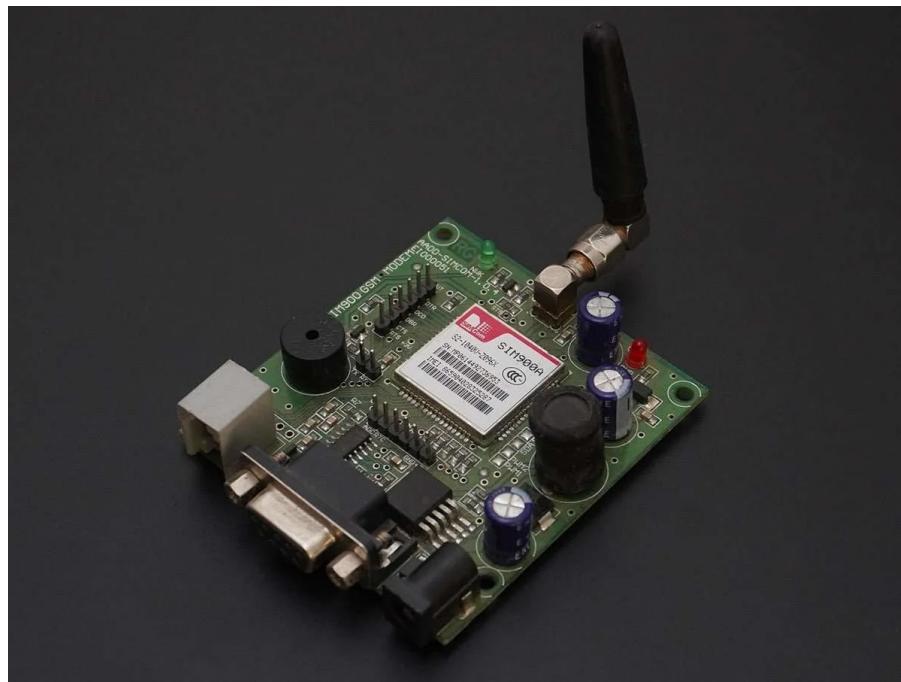


Contests
(/contests)

TCP Client using SIM900A GPRS and AVR ATmega16



Introduction





SIM900

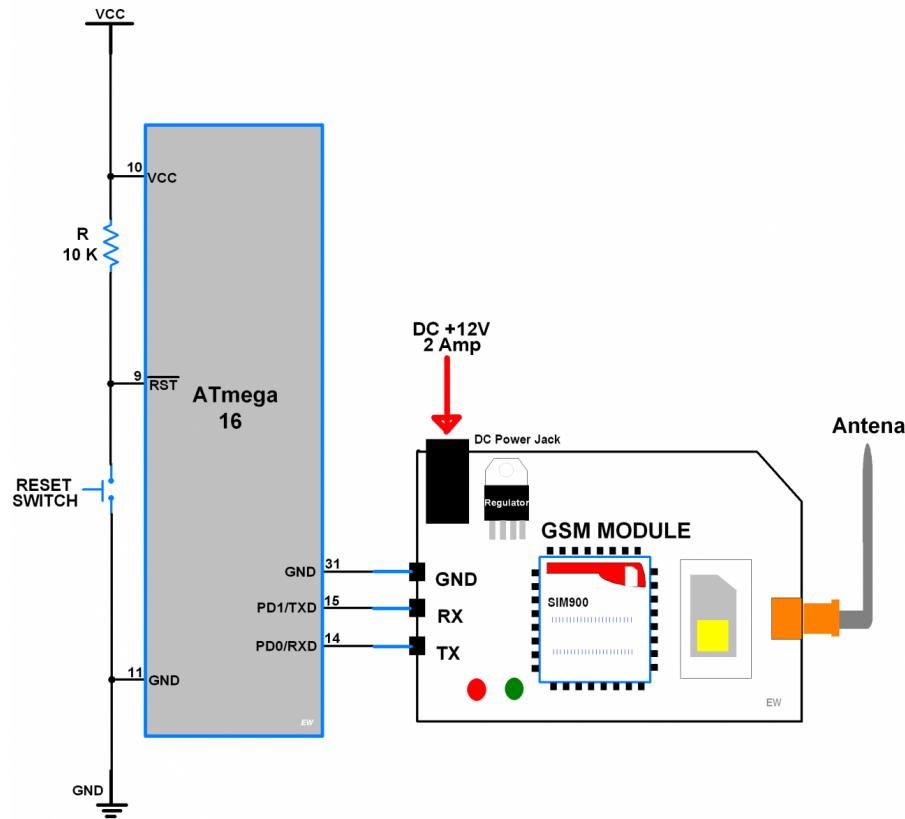
SIM900 enables GPRS to embedded applications. We can implement **TCP** (https://en.wikipedia.org/wiki/Transmission_Control_Protocol) Client protocol using SIM900 TCP function AT Commands.

The Transmission Control Protocol (TCP) is a standard transport layer internet protocol which used in establishing and maintaining communication between server and client.

It is widely used in IoT (Internet of Things) embedded applications, where every sensor is connected to a server and we have access to control them over the internet.

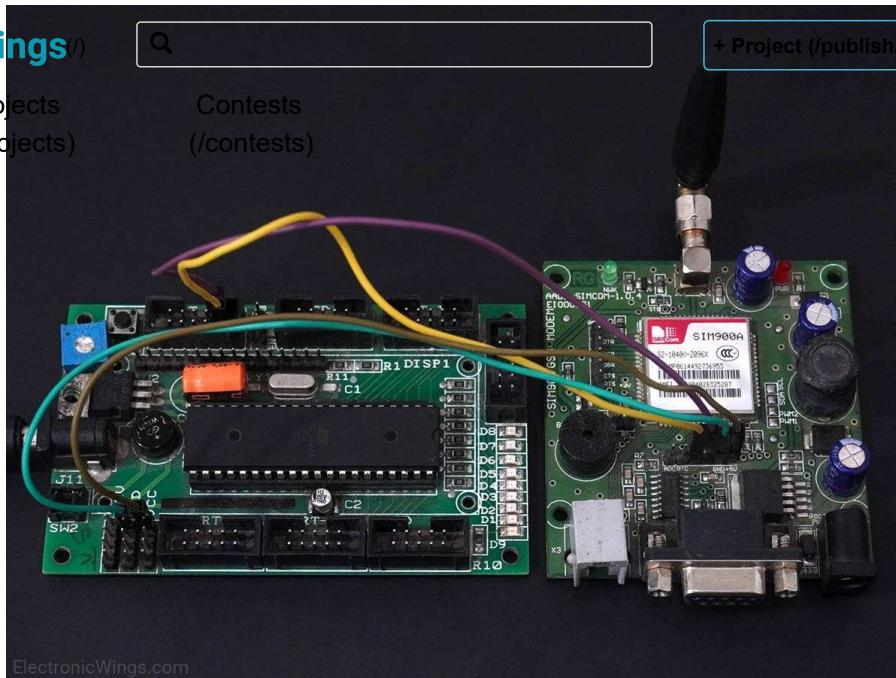
To know about SIM900 GSM/GPRS Module refer to **SIM900** (<http://electronicwings.com/sensors-modules/sim900a-gsmgprs-module>)

Interfacing Diagram



Atmega16 interface with SIM900


[+ Project \(/publish/project\)](#)

[Platforms
\(/explore\)](#)
[Projects
\(/projects\)](#)
[Contests
\(/contests\)](#)


TCP Client over GPRS

Let's program AVR ATmega16 to configure SIM900A as TCP Client and Receive/Send data to/from the server using GPRS.

Here, we are using the Thingspeak server for TCP Client demo purposes.

Thingspeak is an open IOT platform where anyone can visualize and analyze live data from their sensor devices. Also, we can perform data analysis on data posted by remote devices with Matlab code in Thingspeak. To learn more about Thingspeak refer link [\(https://thingspeak.com/pages/learn_more\)](https://thingspeak.com/pages/learn_more)

Just sign up and create a channel. We have below the channel and write key on Thingspeak for data send and receive.

- channel ID is = 119922
- Write Key is = C7JFHZY54GLCJY38

Note: Do not forget to tick the **Make Public** field in the **channel setting** option on your Thingspeak channel. It makes channels available to use as public.

For **TCP RECEIVE** method use below AT command steps shown in the screenshot of RealTerm Serial Terminal.

The below screenshot consists of AT commands (Green) and Responses (Yellow).



```

RealTerm: Serial Capture Program 2.0.0.70
Project (/public/project)

OK\r\n
AT+CIPMODE=0\r\n
OK\r\n
AT+CIPMUX=0\r\n
OK\r\n
AT+CGATT=1\r\n
OK\r\n
AT+CREG?\r\n
+CREG: 0,1\r\n
OK\r\n
AT+CGATT?\r\n
+CGATT: 1\r\n
OK\r\n
AT+CSTT="internet", "", ""\r\n
OK\r\n
AT+CIICR\r\n
OK\r\n
AT+CIFSR\r\n
10.108.207.100\r\n
AT+CIPSTART="TCP", "api.thingspeak.com", "80"\r\n
OK\r\n
CONNECT OK\r\n
AT+CIPSEND\r\n
> GET /channels/119922/feeds/last.txt\r\n
SEND OK\r\n
{"created_at": "2017-04-29T12:22:08Z", "entry_id": 1076, "field1": "1"}\r\n
CLOSED

```

For the **TCP SEND** method use below AT command steps shown in the screenshot of RealTerm Serial Terminal.

```

RealTerm: Serial Capture Program 2.0.0.70
AT+CGPWR=0\r\n
OK\r\n
AT+CIPMODE=0\r\n
OK\r\n
AT+CIPMUX=0\r\n
OK\r\n
AT+CGATT=1\r\n
OK\r\n
AT+CREG?\r\n
+CREG: 0,1\r\n
OK\r\n
AT+CGATT?\r\n
+CGATT: 1\r\n
OK\r\n
AT+CSTT="internet", "", ""\r\n
OK\r\n
AT+CIICR\r\n
OK\r\n
AT+CIFSR\r\n
10.106.22.175\r\n
AT+CIPSTART="TCP", "api.thingspeak.com", "80"\r\n
OK\r\n
CONNECT OK\r\n
AT+CIPSEND\r\n
> GET /update?api_key=C2JFHZY54GLCJY38&field1=1\r\n
SEND OK\r\n
1076\r\n
CLOSED

```

In the below program of TCP Client, do the following

For TCP Client RECEIVE demo

```

#define RECEIVE_DEMO          /* Define RECEIVE demo */
#ifndef SEND_DEMO           /* Define SEND demo */

```

For TCP Client SEND demo

```

#ifndef RECEIVE_DEMO        /* Define RECEIVE demo */
#define SEND_DEMO            /* Define SEND demo */

```



Edit Fields below with respective data



+ Project (/publish/project)



Platforms
(/explore)

Projects Contests
 (/projects) (/contests)
 /* Define Required fields shown below */
 #define DOMAIN "api.thingspeak.com"
 #define PORT "80"
 #define API_WRITE_KEY "C7JFHZY54GLCJY38"
 #define CHANNEL_ID "119922"
 #define APN "internet"
 #define USERNAME ""
 #define PASSWORD ""

In the below program, we are using response-based functions to get the better status if things deviate from normal.

Program for TCP Client

```
/*
 * ATmega16_GPRS_TCPClient
 * http://www.electronicwings.com
 */

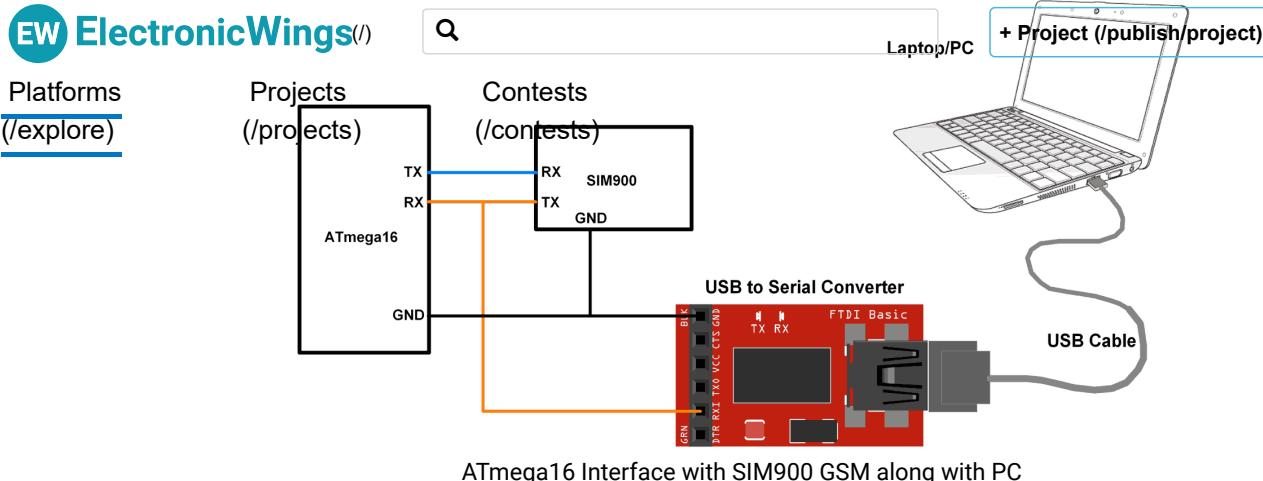
#define F_CPU 8000000UL      /* Define CPU clock Frequency e.g. here */
#include <avr/io.h>          /* Include AVR std. library file */
#include <string.h>          /* Include string library */
#include <stdio.h>           /* Include standard IO library */
#include <stdlib.h>           /* Include standard library */
#include <stdbool.h>          /* Include standard boolean library */
#include <util/delay.h>        /* Include delay header file */
#include <avr/interrupt.h>      /* Include avr interrupt header file */
#include "USART_RS232_H_file.h" /* Include USART header file */

#define SREG _SFR_IO8(0x3F)

#define DEFAULT_BUFFER_SIZE    200
#define DEFAULT_TIMEOUT        20000
#define DEFAULT_CPLF_COUNT     2
```

SIM900 Response

At the client end, we need to check SIM900 responses. We can check it on the serial terminal of the PC/Laptop. Connect SIM900 transmit pin (TX) to the receive pin (RX) of Atmega16 Microcontroller and to the receive pin (RX) of USB to serial converter as shown in the below figure. connect USB to serial converter to PC/Laptop. Open the serial terminal on the PC/Laptop to see the SIM900 responses for the AT command sent from the Atmega16 microcontroller.



ATmega16 Interface with SIM900 GSM along with PC

Now for **TCP SEND** commands (sent from ATmega16 Microcontroller), we can see the below response from SIM900 on the serial terminal for the Thingspeak server.

```
+CIPMUX: 0
OK
OK
CONNECT OK
>
SEND OK
976
CLOSE OK
+CIPMUX: 0
OK
OK
CONNECT OK
>
SEND OK
977
CLOSE OK
```

Thingspeak Responds with entry id for data send success

In response to **TCP SEND** we get the data entry no. as shown in the above figure i.e. 976, 977, and so on.

For **TCP RECEIVE** commands (sent from ATmega16 Microcontroller), we can see the below response from SIM900 on the serial terminal for the Thingspeak server.

```
+CIPMUX: 0
OK
OK
CONNECT OK
>
SEND OK
{"created_at": "2017-04-29T07:08:47Z", "entry_id": 975, "field1": "2"}
CLOSE OK
+CIPMUX: 0
OK
OK
CONNECT OK
>
SEND OK
{"created_at": "2017-04-29T07:08:47Z", "entry_id": 975, "field1": "2"}
CLOSE OK
```

Thingspeak Response for last updated data

In response to **TCP RECEIVE** we get the last entry data for field1 on Thingspeak as shown in the above figure.



Platforms
[\(/explore\)](#)

Projects
[\(/projects\)](#)

Contests
[\(/contests\)](#)

Note: here we are retrieving the last entry data on field1 of Thingspeak Project (publish/project) server hence we get the last updated data of field1 from the server as shown in the above figure i.e. "field1": "2". In the program, we used GET /channels/2/feeds/last.txt" to receive the last updated data only.



Updates at Thingspeak server on TCP SEND

For **TCP SEND** we can see the output at the server end. Here we are using the Thingspeak server and sending the incremented count at field1 on the server. We get incremented count at field1 of Thingspeak server as shown in below figure.





+ Project Mouser/project



Platforms
(/explore)

Projects
(/projects)

Contests
(/contests)

(https://www.
mouser.in?
utm_source=el
ectronicswing
s&utm_medium
m=display&ut
m_campaign=
mouser-
componentsli
sting&utm_co
ntent=0x0)

(https://www.mouser.i
n/ProductDetail/Micro
chip-Technology-
Atmel/ATMEGA16L-
8PU?
qs=%2Fha2pyFaduiGC
JtTvs2wv8fVZbVAalLu
7Iq%2FgITS0tALAx6f
MenLvg%3D%3D&utm_
source=electronicswin
gs&utm_medium=displ
ay&utm_campaign=m
ouser-
componentslisting&ut
m_content=0x0)

Datasheet (/componen
ts/atmega-
16/1/datashe
et)

ATmega 16
ATmega 16

X 1

(https://www.mouser.i
n/ProductDetail/Micro
chip-Technology-
Atmel/ATMEGA32-
16PU?
qs=aqrBurbvGdpkmjg
7RWmsQ%3D%3D&ut
m_source=electronics
wings&utm_medium=d
isplay&utm_campaign
=mouser-
componentslisting&ut
m_content=0x0)

Datasheet (/componen
ts/atmega3
2/1/datashe
et)

Atmega32
Atmega32

X 1



Platforms
(/explore)

Projects
(/projects)

Contests
(/contests)

Components Used

Powered By
+ Project: MouserELECTRONICS/project
(https://www.mouser.in?utm_source=electronicswing&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

SIM900A GSM GPRS Module

SIM900A is dual band GSM/GPRS 900/1800MHz module...

X 1

(https://www.mouser.com/ProductDetail/M5Stack/M031-D?qs=%2Fha2pyFadugEKx3cUjc5DGB4hxkc1iwAkV7YxEmv6c%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

Datasheet (</components/sim900a-gsm-gprs-module/1/datasheet>)

CP2103 USB TO UART BRIDGE

CP2103 is single chip USB to UART Bridge. It su...

X 1

(https://www.mouser.com/ProductDetail/Silicon-Labs/CP2103-GMR?qs=Zq62GxwlckYrXEgTuxpNRg%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

Datasheet (</components/cp2103-usb-to-uart-bridge/1/datasheet>)



Platforms
(/explore)

Projects
(/projects)

Contests
(/contests)

ATmega16 GPRS TCP Client Project file

+ Project (/publish/project)



Dow (/api/download/platf
nloa orm-attachment/377)
d

Dow (/api/download/platf
nloa orm-attachment/378)
d

Dow (/api/download/platf
nloa orm-attachment/379)
d

Comments



Comment



qadirdu12

(/users/qadirdu12/profile)
2020-06-26 11:44:09

:

Will that code work fine on Proteus?

Reply Like

[About Us \(/about\)](#)
[Business Offering \(/business-services\)](#)
[Host Platform \(/launch-platform\)](#)
[Contact Us \(/contactus\)](#)

[Terms of Service \(/terms-of-service\)](#)
[Cookies Policy \(/cookie-policy\)](#)
[Privacy Policy \(/privacy-policy\)](#)

Connect On:

Facebook(<https://www.facebook.com/electronicwings>)
 LinkedIn(<https://www.linkedin.com/company/electronicwin>)
 YouTube(<https://www.youtube.com/channel/UCNdqkukBtk4>)
 Instagram (https://www.instagram.com/electronicwings_coligshid=1cip10jiittko)

ElectronicWings
© 2023