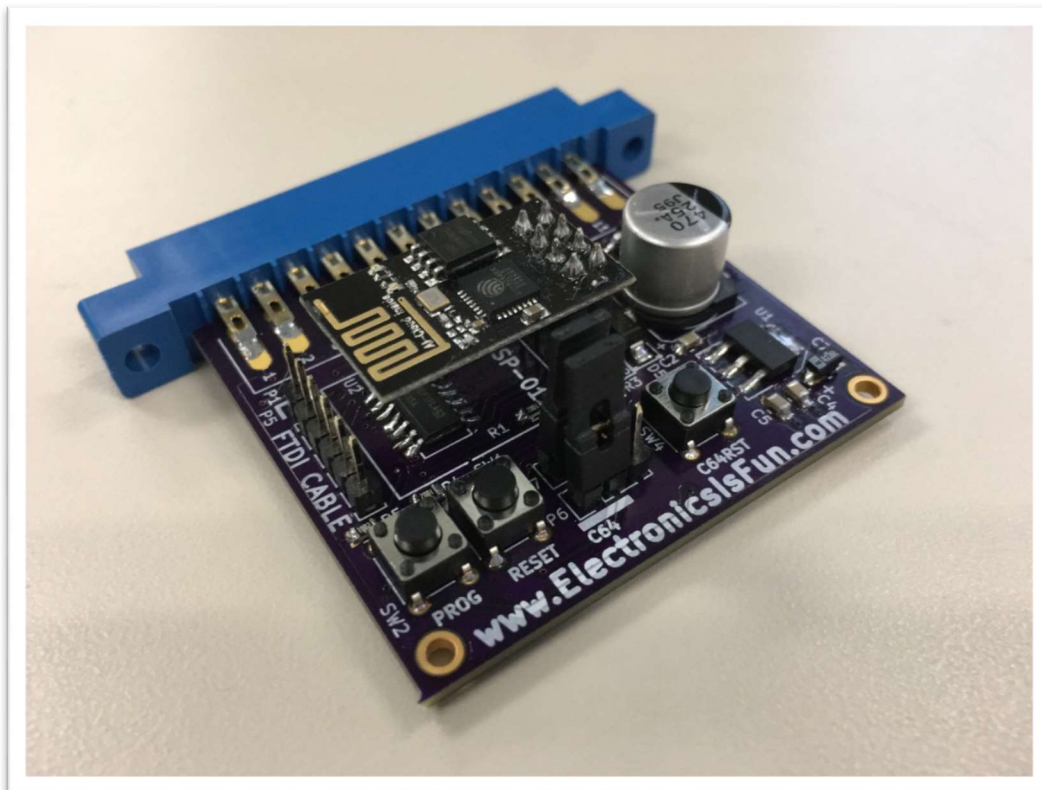


# C64NET WIFI

Internet modem card for Commodore 64  
and Commodore 128 Computers



From Electronics Is Fun

[www.ElectronicsIsFun.com](http://www.ElectronicsIsFun.com)

August 12, 2017

# TABLE OF CONTENTS

<b>C64NET WIFI CARD INTRODUCTION .....</b>	<b>1</b>
Getting Started .....	3
Bootstrapping .....	4
Using the C64Net WiFi MODEM Card .....	7
C64Net Programs .....	10
CONFIGURE64-128.....	11
FTP64-128 .....	12
WGET64-128.....	14
IRC64-128.....	15
TELNET64-128.....	15
CBMBBS64-128.....	16
D64WGET64-128 .....	17
TELNETD64.....	18
WEATHER.....	19
PML64.BIN and PML128.BIN .....	19
UP9600.BIN .....	22
Advanced topics.....	26
FTDI Cable Download.....	26
Over the Air Download.....	27

AT Command Reference .....	28
Schematics .....	36
Board Layout .....	37
<b>GLOSSARY .....</b>	<b>38</b>
<b>INDEX .....</b>	<b>39</b>

# C64NET WIFI CARD INTRODUCTION

The C64Net Wifi card adds modern communications capabilities to your Commodore 64 or 128 computer. C64Net provides wireless connectivity to any 802.11g network, Users can connect to the internet using the provided applications (on the supplied software disk). Additionally, the user can also connect to other computers and BBSs. The C64Net can be programmed using AT commands that are similar to the Hayes modem command set used with phone line modems. Libraries are also provided for developer to create their own communications applications.

The C64Net card connects to the computer's user I/O port. No external power or cables are required. The computer talks to the C64Net card over a TTL serial interface at speeds up to 9600 baud\* with the appropriate software. Other models of commodore computers are supported. However, they may only operate at baudrates of 2400 and below.

- *Commodore 128 computer models will not support 9600 baud.*

This manual provides information on how to install and use the C64Net card.

It can be downloaded from:

<http://ElectronicsIsFun.com>

Additional information is provided on the software disk and at Electronics Is Fun. If you purchased only the board, software is provided online at:

[http://coffeemud.net:8080/ctcug/index.php/Inexpensive Wireless Internet Modem](http://coffeemud.net:8080/ctcug/index.php/Inexpensive_Wireless_Internet_Modem)

There are two way to get started using the C64Net.

First, you can install the software provided on disk:

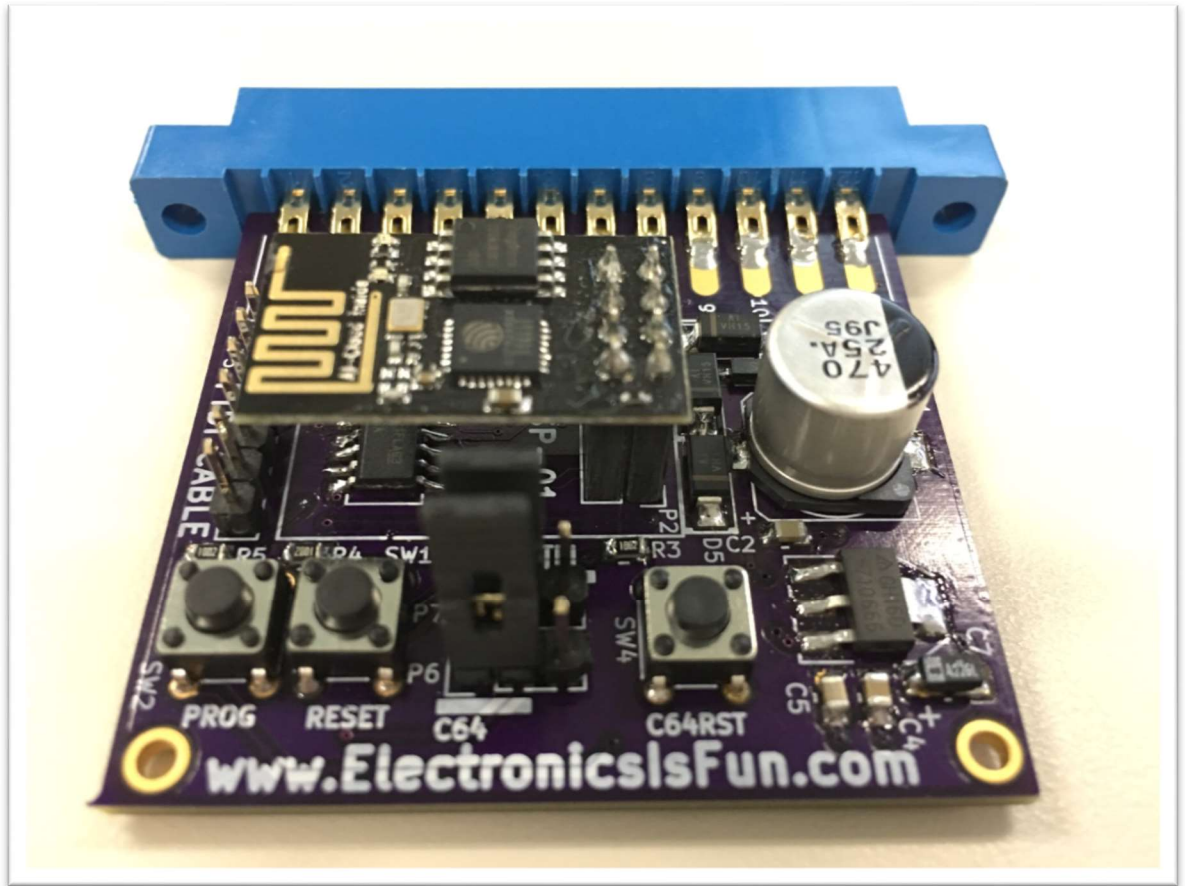
1. *Insert the software disk in you drive and load the CONFIGURE64-128 program.*
2. *Follow the instructions to configure the C64Net.*
3. *Once configured, you will be ready to use your favorite communications software to get on line.*

The second way is to read the GETTING STARTED SECTION. This section assumes that you did not receive a software disk with your board. It provides two basic programs that you can enter to bootstrap your computer and download all the software provided on the software disk.

### ***System Requirements***

1. *A Commodore 64 computer (other compatible models such as the C128)*
2. *(1) Floppy Drive.*
3. *C64Net card*

#### 4. *Communications Software.*



## GETTING STARTED

Installing the C64Net Wifi card simple and only takes a few minutes.

1. *Make sure the computer is off.*
2. *Find the user I/O port on the rear of the computer.*
3. *Install the C64Net Wifi card as shown in the figure.*
4. *Turn the power on.*

The C64Net Wifi card is now ready to use. It can be used with off the shelf software such as StrikeTerm, CCGMS, and others. It can also be used with basic and assembly programs you create.

***NOTE: Additional information is required for configuration. For example, you will need to provide your network SSID (service set identifier), password, an IP address and a port number.***

If you do not have a serial communication program, it's ok. You can boot strap your system using a small basic program.

Software is available on line to allow the C64Net card to communicate with other computers with the TelNet protocol and FTP. Additional tools and utilities are available to configure the modem and to improve communications over the internet.

If you received a disk with your C64Wifi card, it will contain the software to get you started. See the section on running the supplied software.

## **BOOTSTRAPPING**

For those without the disk, a small basic program can be used to get started. It is a dumb terminal program that can be used to configure the modem for your wifi network.

This process of getting the modem configured is known as bootstrapping. First the dumb terminal program will be used to get your modem connected to the network, then another program called the bootstrap loader will allow you to download all the other tools and software for the modem.

Enter the dumb terminal basic program below and save it to disk.

```
10 open5,2,0,chr$(8):poke665,73-(peek(678)*30)
15 print#5,"at&ple1":print#5,"at&ple1":printchr$(14);
20 get#5,a$:ifa$<>" thenprinta$;
30 geta$:ifa$<>" thenprint#5,a$;
40 goto20
```

Run the dumb terminal basic program, with it you will be able to configure your network settings.

To display the available wifi networks enter ATW and press <return>. You should see several wifi network SSID's (sometimes referred to as the hotspot name) displayed.

Once you have identified your SSID, enter the following command to configure it.

```
ATW"<your SSID>,<password>"
```

This will take a few seconds, an "OK" will be displayed if you were successful. If an "ERROR" is displayed, check your SSID/password and try again.

To store your wifi network settings issue the following command.

```
ATZ&W<enter>
```

Once your modem is connected to wifi, switch your C64 to lowercase mode by pressing shift-C= (<shift><commodore key>) and then type in the following program and save it to a disk:

```
10 open5,2,0,chr$(8):pa=peek(678)*30:poke665,73-pa
15 a3$=chr$(3):print#5,a3$;
20 input"enter filename";f$:iff$=""thenstop
30 input"enter drive";dv;ifdv<8thenstop
35 print#5,a3$;
40 get#5,a$:ifa$<>""then40
50 dl$=mid$(str$(99+pa),2):print#5,"atze0r0&p1f1s45=2s44="+dl$
60 input#5,a$:ifa$<>"OK"anda$<>"ok"then60
70 print#5,"at&g";chr$(34);"www.zimmers.net:80/otherprojs/";f$;chr$(34)
75 print#5,chr$(19);
80 input#5,b$:ifb$="["then100
90 printb$:print"reset computer and check wifi":close5:stop
100 input#5,ch,ln,c8,b$:ifb$<>"]"thenprint#5,a3$:goto90
110 n8=0:open8,dv,8,"@0:"+f$+"",p,w":print"reading...";
120 m=49152:ni=i+(ln/20):print#5,chr$(17);:fori=1toln
130 input#5,a:n8=n8+a:ifn8>255thenn8=n8-256
140 pokem,a:m=m+1:ifm>53240thengosub200:m=49152
145 ifi>nithenni=i+(ln/20):printint(i/ln*100);"%",
150 next:print#5,a3$;"atz":ifn8<>c8thenprint"checksum fail. try again."
160 print"100%",:gosub200:close8:close5:print"done":poke49152,0:poke51200,0
199 end
200 print#5,chr$(19);:print:print"writing...":ti$="000000":tt=ti+200
205 ifti<ttthen205
210 forx=49152tom-1:print#8,chr$(peek(x));:nextx
220 ifi<lnthenprint"reading...":tt=ti+200
225 ifti<ttthen225
230 print#5,chr$(17);:return
```

This program will be used to download additional program that are needed to allow the modem to run at 9600 baud and run the FTP (file transfer protocol) protocol. You will need to run the program three times to download the following files:

1. pml64.bin (that's pee em el, not pee em one)
2. up9600.bin
3. ftp64-128

If you have any problems or errors here, such as the program doing nothing after the "enter drive" prompt, or it freezing at a certain download percentage, or getting a "file data" error or some other error, then do the following: First, hold down the RUN/STOP key and sharply strike the RESTORE key at the same time. You should see a "READY." prompt. Now enter: "close8" and hit RETURN. This should properly close the file if it was open, and turn the light on your disk drive off. Next, you should turn your computer off and turn it back on to reset it. Then re-load the little downloader program above and run it again. Don't get frustrated. This dumb little program is only to bootstrap you. Once you have the files "ftp64-128", "pml64.bin", and "up9600.bin", things will go much smoother and sweeter.



If you have successfully downloaded the three files, you can now speed things up by running FTP at 9600 baud . If you are using a commodore 128, it will run at 1200 baud.

Load and run the ftp64-128 and connect to [ftp.zimmers.net](http://ftp.zimmers.net) anonymously. Once you are logged-in, enter the following commands.

1. cd pub/projects/c64wifi
2. ls
3. get pml128.bin,p
4. get telnet64-128,p
5. ... repeat for all the files you saw in step 2.
6. ... repeat step 2 if you forget which files remain

The bootstrapping process is now complete. You should now have all the software for the C64Net Wifi modem on your disk.

## USING THE C64NET WIFI MODEM CARD

Using the DUMBTERM basic program or another communications program, you can issue commands to the C64Net to change parameters and configuration. This section describes some of the more common commands and how they can be used to get the most out of the C64Net card. Additional commands can be found in the AT Command Reference

### *Set the baud rate*

The default baud rate is set to 1200 baud. If you want to change this, you need to understand that once the baud rate is changed, characters will no longer be displayed properly. This can be corrected by changing the settings in you communications program after you have set a new baud rate with the following command.

```
atb9600
```

This command will change the baud rate to 9600 baud. Other baud rates are available. Here is a list of the baud rates available:

```
1200, 2400, 4800, 9600.
```

### *Configuring the SSID and password for your network*

Start the communications program and issue the following command:

```
atw"your SSID, your password"
```

### *Saving your SSID and password*

Since most users frequently connect to the same WiFi network, it is best to save the SSID and password that has been configured. Otherwise, the setting will be lost when you power down the system.

```
at&w
```

Once you issue this command, the C64Net card will remember the SSID and password to your network.

## ***PETSCII vs. ASCII***

The C64Net card can understand PETSCII and ASCII. However, under certain situations sending PETSCII to the C64Net can result in an error. This occurs because the C64Net can only interpret commands in ASCII. For example, when setting the SSID and password, if the computer is in graphics (PETSCII) mode an error will be generated. It is for this reason that the following command was created.

```
atwp"your SSID, your password"
```

## ***Getting help***

After you configure the C64Net, you display help information by issuing the following command.

```
at&h
```

Note: this command will only work if you have configured your SSID and password. This is because the command will access the help information from the internet. After this help has been access once, it will remain resident. Issuing the command again will display a local copy.

## ***Display available hotspots/networks***

You can scan for available networks with the C64Net card by issuing the following command.

```
atw
```

This will display the available hotspots along with their signal strength. This command is helpful when trying to connect to a different network.

Note: this command only works if you have already configured your SSID and password.

### ***Reset to factory settings***

You can reset the C64Net card back to factory setting by issuing the following command.

```
at&f
```

This command will erase any settings that were configured by the user back to the default settings that were set at the factory.

### ***BBS's***

With the C64Net card, you can access BBS's all over the world. Access is simple, just find your favorite BBS. Then use the atd command to connect.

The following site contains a list of BBS's dedicated to the commodore computer. To access a BBS, you need the name of the website and the port number.

```
http://cbbsoutpost.servebbs.com/
```

For example, for CottonWood BBS, the website is cottonwoodbbs.dyndns.org and the port number is 6502. Do to connect, you would issue the following command:

```
atd" cottonwoodbbs.dyndns.org:6502"
```

### ***UP9600***

The C64Net can operate at 9600 baud using the UP9600 driver. This driver is supplied with the software disk or is available for download. To enable 9600 baud support SW3 is provided. When the (2) switches are in the ON position, UP9600 support is enabled. On some systems, including the commodore 128, the UP9600 support must be disabled by setting the (2) switches to the off position. Other platforms may also require UP9600 to be disabled.

## C64NET PROGRAMS

A set of easy to use program are available for using your C64Net Wifi modem. These programs are described in this section. Additional information on using each of these program can be found in the following sections of this manual.

CONFIGURE64-128 – Setup Wifi settings

[FTP64-128](#) – Used to access files on FTP servers

WGET64-128 – Used to download files and pages using HTTP web addresses (URL)

IRC64-128 – For chatting using internet relay chat

TELNET64-128 – Used for communicating with TELNET servers

CBMBBS64-128 – Used for communicating with Commodore Graphics BBS servers

D64WGET64-128 – Used for downloading full disk images

TELNETD64 – Used to control your Commodore 64 remotely over the internet

WEATHER – a two player online game

PLM64.BIN – Used to perform functions from basic on the commodore 64

PML128.BIN – Used to perform functions from basic on the commodore 128

## CONFIGURE64-128

The Configure program is for setting up or altering the WiFi Hotspot, also known as a Wireless Internet Router. Your modem uses this router to communicate with the internet. It also allows you to modify the modems built in phone book. It is written in BASIC, but uses the PML64.BIN or PML128.BIN libraries for speed (see below). This program requires that your C64Net WiFi Modem be configured for 1200 baud on reset, which is the default factory configuration.

When the program runs, it will initialize the C64Net modem with certain commands. If you receive an error at this point, or if the program hangs for longer than a minute, try turning your computer off and on again to clear memory and reset the modem, and then load and run it again.

After initialization, the program will scan the area for WiFi Hotspots and provide you a list to select from. If you do not see your own wireless router listed, or do not know the name, consult your ISP's internet equipment documentation. After you have selected the appropriate hotspot, you will be prompted for the password. After the password is entered, the program will test the settings and, if successful, save the settings to the flash memory in your modem it will be remembered even after the computer is turned off. When wifi configuration is complete, and the modem passes the connection test, you can modify the modems internal phone book.

The modems internal phone book maintains a list of numbers and associated internet addresses and configurations. These numbers can then be used by third party software to 'dial' to those sites using the modems ATD command. A phone number, in this case, is any large number, preferably 7 digits or more, that is assigned to a particular internet host and port. Each host can then be configured for ASCII-PETSCII translation, TELNET code translation, XON/XOFF flow control, and/or local ECHO (half vs full duplex). Every modification you make is saved to the modem and available after reboots. This system can be used to connect to the online Q-Link service online using the existing Q-Link software.

## **FTP64-128**

File Transfer Protocol program supports a minimal number of features to connect to FTP Servers all over the internet in order to download and/or upload programs. It is written in BASIC, but uses the UP9600.BIN and PML64.BIN or PML128.BIN libraries for speed (see below).

This program requires that your C64Net WiFi Modem be configured for 1200 baud on reset, which is the default factory configuration. If you run it on a breadbox Commodore 64 or 64C with UP9600 enabled on the modem (default), it will operate at 9600 baud during your session. If you run it on a Commodore 128 in C128 mode and 80 columns, it will operate at 2400 baud during your session. If you run it on a Commodore 128 in C64 mode, or C128 mode and 40 columns, it will remain at 1200 baud speed.

When you start the program, it will initialize your C64Net modem with certain commands. If you receive an error at this point, or if the program hangs for longer than a minute, try turning your computer off and on again to clear memory and reset the modem, and then load and run it again.

After initialization, you will be given a menu where you can enter the host name of the ftp server to connect to, and the login and password to use. If you plan to login with the name "anonymous", you should use your email address as your password. The last option on the menu is the drive number to upload files from and download files to. When all options are correct, hit RETURN to connect.

At the FTP prompt, you may enter one of the following commands:

DIR : list the files in the current directory

CD [DIRECTORY OR PATH] : change to a new directory

QUIT : exit the ftp program

HELP : show the commands supported by the client and the server

GET [FILENAME,P] : download the program with the name filename as a PRG. Use ,S for SEQ.

PUT [FILENAME,P] : upload the program with the name filename which is a PRG file. Use ,S for SEQ.

DEL [FILENAME] : delete the given file from the server

LCD [DEVICE NUMBER or CMD Hard Drive PATH] : change the active local upload/download drive, or change which directory you are using if the active drive is a UIEC or CMD Hard Drive compatible.

LDIR [MASK] : get a directory of the active local drive.

LDEL [FILENAME] : delete the given local file from the active drive.



## WGET64-128

WGET program is for downloading files and pages from a standard HTTP web address (URL). It is written in BASIC, but uses the UP9600.BIN and PML64.BIN or PML128.BIN libraries for speed (see below).

This program requires that your C64Net WiFi Modem be configured for 1200 baud on reset, which is the default factory configuration. If you run it on a breadbox Commodore 64 or 64C with UP9600 enabled on the modem (default), it will operate at 9600 baud during your session. If you run it on a Commodore 128 in C128 mode and 80 columns, it will operate at 2400 baud during your session. If you run it on a Commodore 128 in C64 mode, or C128 mode and 40 columns, it will remain at 1200 baud speed.

When you start the program, it will initialize your C64Net modem with certain commands. If you receive an error at this point, or if the program hangs for longer than a minute, try turning your computer off and on again to clear memory and reset the modem, and then load and run it again.

After initialization, you will be given a menu where you can enter the url of the file or page to download, the unit number of the disk drive to download the file or page to, and the name of the file to write to the disk. Your filename should end with ,P to create a PRG file, and ,S to create a SEQ file. For example, a filename of "mypage,s" will create a file called "mypage" of type SEQ. When all options are correct, hit RETURN to connect and begin the download. When the download completes, the program will exit.

## **IRC64-128**

IRC program is for chatting with people on free internet IRC Servers. It is written in BASIC, but uses the PML64.BIN or PML128.BIN libraries for speed (see below). This program requires that your C64Net WiFi Modem be configured for 1200 baud on reset, which is the default factory configuration. If you run it on a breadbox Commodore 64 or 64C with UP9600 enabled on the modem (default), it will operate at 9600 baud during your session. If you run it on a Commodore 128 in C128 mode and 80 columns, it will operate at 2400 baud during your session. If you run it on a Commodore 128 in C64 mode, or C128 mode and 40 columns, it will remain at 1200 baud speed.

When you start the program, it will initialize your C64Net modem with certain commands. If you receive an error at this point, or if the program hangs for longer than a minute, try turning your computer off and on again to clear memory and reset the modem, and then load and run it again.

After initialization, you will be asked to enter the host name of the IRC Server to connect to. Next you will be prompted for the port number that the server listens on. The program will then attempt to connect to the host and port. If successful, you will be prompted to enter a nickname, which is the name you will be known by during chatting. If your nickname is already in use by someone else, you will eventually be prompted to enter a new one. Either way, the program will display the servers MOTD and other information. This could take several minutes. It's best to just wait until you see the message "End of MOTD" to begin typing.

After the MOTD message, you may type at any time. If you have joined a chat channel, you will automatically see messages as they appear in the channel. When you begin typing, all messages from the server are suspended until you hit RETURN. Since the IRC client will appear to the server to be idle while you are typing, it is best to spend as little time as necessary in message and command entry mode. You may enter ? followed by RETURN to see the IRC commands supported by this client.

Commands include:

/QUIT : exit the IRC chat program

/JOIN #channelname : join the new channel named "#channelname"

\* Anything else entered will be assumed to be a message to sent to the channel you have already joined.

## **TELNET64-128**

TELNET program is for communicating with a TELNET server as a terminal, or with a TELNET-like server such as a Multi-User Dungeon game (a MUD). It is written in BASIC, but uses the PML64.BIN or PML128.BIN libraries for speed (see below). This program

requires that your C64Net WiFi Modem be configured for 1200 baud on reset, which is the default factory configuration.

When you start the program, it will initialize your C64Net modem with certain commands. If you receive an error at this point, or if the program hangs for longer than a minute, try turning your computer off and on again to clear memory and reset the modem, and then load and run it again.

After initialization, you will be shown a menu with the following options:

Dial from Phonebook:

This will let you connect to a server saved in your phonebook on disk. See below for information about terminal mode while connected to a server

Modify Phonebook:

This will let you add or modify servers to save in your phonebook on disk. These can be then "Dialed" to quickly later on.

Quick Connect:

This will let you manually enter a server host name and port to immediately connect to. See below for information about terminal mode while connected to a server.

Terminal Mode:

This will put you into terminal mode without connecting to any server. In this case, you would be in the modem firmware's Command Mode. You should consult the documentation for the C64Net WiFi Firmware for more information on this mode. Use the F1 key or the C128's ESC key to exit terminal mode.

Quit:

Exit the TELNET program

When you have connected to a server, you will be in ANSI terminal mode. You can exit the server and disconnect at any time by hitting the F1 key on your C64, or the ESC key on your C128. When connected to a server, the modem will translate as much ASCII/ANSI into mostly viewable PETSCII for your Commodore computer. However, you will notice that underscores look like back-arrows. ANSI colors will be translated to Commodore PETSCII colors wherever possible. Your own keystrokes will also be translated to ASCII/ANSI as you type.

## **CBMBBS64-128**

C= BBS program is for communicating with a telnet-style Commodore Graphics BBS server as a terminal. It is written in BASIC, but uses the PML64.BIN or PML128.BIN libraries for speed (see below). This program requires that your C64Net WiFi Modem be configured for 1200 baud on reset, which is the default factory configuration.

When you start the program, it will initialize your C64Net modem with certain commands. If you receive an error at this point, or if the program hangs for longer than a minute, try turning your computer off and on again to clear memory and reset the modem, and then load and run it again.

After initialization, you will be shown a menu with the following options:

Dial from Phonebook:

This will let you connect to a server saved in your phonebook on disk. See below for information about terminal mode while connected to a server

Modify Phonebook:

This will let you add or modify servers to save in your phonebook on disk. These can be then "Dialed" to quickly later on. You will receive an option for a Host Name, a Port Number, and Local Echo. Consult your online BBS list for hosts and ports of the BBSes you wish to connect to. The Local Echo option should usually be answered with NO, as most BBSes will echo for you, and having the modem do local echo will corrupt X-Modem transfers.

Quick Connect:

This will let you manually enter a server host name and port to immediately connect to. See below for information about terminal mode while connected to a server.

Terminal Mode:

This will put you into terminal mode without connecting to any server. In this case, you would be in the modem firmware's Command Mode. You will be prompted as to whether you would like Local Echo turned on. If you plan to remain in Command Mode and not connect to anything, you can say Yes. However, if you plan to connect to a BBS, or especially if you plan to do any file transfers, you should definitely answer NO. You should consult the documentation for the C64Net WiFi Firmware for more information on this mode. Use the F1 key or the C128's ESC key to exit terminal mode. See below for more you can do after exiting terminal mode.

Quit:

Exit the CBMBBS program

When you have connected to a server, you will be in CBM PETSCII/Graphics mode. You can exit the terminal at any time by hitting the F1 key on your C64, or the ESC key on your C128. If you are offline, you will immediately return to the main menu. However, if you are still online, you will have the option to hang-up and return to the main menu, upload a file using X-Modem CRC, download a file, or continue back to terminal mode.

## **D64WGET64-128**

D64WGET program is for downloading and writing full emulator disk images (such as .D64, .D71, .D81, .DNP, etc) files from a standard HTTP web address (URL) and write

them to a formatted, blank diskette. It is written in BASIC, but uses the UP9600.BIN and PML64.BIN or PML128.BIN libraries for speed (see below).

This program requires that your C64Net WiFi Modem be configured for 1200 baud on reset, which is the default factory configuration. If you run it on a breadbox Commodore 64 or 64C with UP9600 enabled on the modem (default), it will operate at 9600 baud during your session. If you run it on a Commodore 128 in C128 mode and 80 columns, it will operate at 2400 baud during your session. If you run it on a Commodore 128 in C64 mode, or C128 mode and 40 columns, it will remain at 1200 baud speed.

When you start the program, it will initialize your C64Net modem with certain commands. If you receive an error at this point, or if the program hangs for longer than a minute, try turning your computer off and on again to clear memory and reset the modem, and then load and run it again.

After initialization, you will be given a menu where you can enter the url of the disk image file to download and write, and the unit number of the disk drive containing the formatted blank disk to write the image to. When all options are correct, hit RETURN and answer Y to the confirmation to connect and begin the download and disk write. When the download completes, which could take awhile, the program will exit.

## **TELNETD64**

TELNETD program is for controlling your C64 remotely over the internet. It is written mostly in machine language which is contained in the file "RDS64.BIN", with the source code in "RDS64.BAS". The program TELNETD64 is written in BASIC, and is for configuring and launching your server.

This program requires that your C64Net WiFi Modem be configured for 1200 baud on reset, which is the default factory configuration.

When you start the program, it will load "rds64.bin" and prompt you for certain server settings, such as how much time to allow users to remain on your computer in a single session (Time Limit), the amount of time to allow users to remain online when they haven't been typing anything (the Idle Time), remain online when they aren't typing anything, the security flag, the port to listen on, and the modem initialization string. You should be cautious with modifying that last, as it may affect the servers functionality.

After the prompts, the c64 will start up the server itself. The server will initialize your C64Net modem with the above modem initialization string. If you receive an error at this point, or if the program hangs for longer than a minute, try turning your computer off and on again to clear memory and reset the modem, and then load and run it again.

Once the server is started, you will be right back to the normal BASIC "READY." prompt. When a user connects, they will be able to type in commands, which you can watch them do. If the security flag was turned on, then certain BASIC commands, such as "POKE" and

"SYS" will be disabled, as will the ability to OPEN channel 2 (the modem). Should the user violate the idle or session time limit, they will be immediately disconnected.

You should be careful with what disk drives you leave connected and which disks you leave in those drives, as online users may be able to modify them. You should also be aware that this program is in an Alpha state. For example, a C64 with JiffyDOS will work with the TELNETD software, but any attempts to use JiffyDos "@" commands will hang up the server. It is also possible there are other ways to hang the server up, even with security turned on. If you discover any new ways, please email the author.

To connect to your server from the internet, you should set your ISP router to forward connections to the TELNETD64 port (default 6400) to the ip address of your modem.

The only way to shut down your server is either to reset the computer, or turn it off and back on again.

## **WEATHER**

Weather is a two player online game originally written for the Commodore PET. This program requires that both players have C64Net WiFi Modems configured for 1200 baud on reset, which is the default factory configuration.

When running the program, the system will ask for one player to act as the host, and the other as the client. The host computer then waits for a connection from the client computer, after which the game can commence. The host player must be familiar with opening firewall ports in order to allow the client to connect to their computer.

In the game, both players compete against each other in this artillery like contest that involves attacking each other's houses with the forces of nature.

## **PML64.BIN and PML128.BIN**

These files, for the C64 and C128 respectively, are machine language support files for performing certain common C64Net modem functions from BASIC. PML64.BIN loads to \$C000 (49152), and PML128.BIN to \$1300 (4864). They are functionally identical.

To use them from BASIC, make sure to follow two important rules: 1. Load the appropriate binary into memory (pml64.bin or pml128.bin). and 2. Initialize the variable p\$, by giving it some value, as early in your program as possible. For example: P\$="" would be fine.

Throughout this section, I will refer to the variable ML to refer to the starting address of the library. For the C64, this is always 49152, and for the C128, this is 4864. So, when I say ML+3, I mean 49155 on the C64, and 4867 on the C128.

## ***BUFILIN***

Usage:  
sys ML

Pre-requisites:  
P\$ must be initialized in BASIC  
Channel 5 must be OPEN to the modem

Outputs:  
P\$ contains the string line read from the modem, if any  
ML + 19 is 0 to denote success, or 255 for a timeout  
ML + 26 contains the CRC8 value of the received string

Description:  
Reads from the open modem channel until a carriage return (13) is seen, or until 255 characters are read, or until a timeout occurs. The characters are returned in the BASIC variable P\$, although character values 10 and 0 are always ignored. The timeout is about 12 seconds.

## ***BUFXLIN***

Usage:  
sys ML+3

Pre-requisites:  
P\$ must be initialized in BASIC  
Channel 5 must be OPEN to the modem  
ML + 18 contains the number of bytes to read

Outputs:  
P\$ contains the X bytes read from the modem if any  
ML + 18 contains the number of bytes NOT read (0 on success)  
ML + 19 is 255 for a timeout, or any other value for the index of the last carriage return (13) in the returned buffer.  
ML + 26 contains the CRC8 value of the received bytes

Reads from the open modem channel until the number of bytes in ML+18 have been read, or until 253 bytes are read, or until a timeout occurs. The bytes are returned in the BASIC variable P\$. The timeout is about 12 seconds.

## ***GETPACKET***

Usage:  
sys ML+6

Pre-requisites:  
P\$ must be initialized in BASIC  
Channel 5 must be OPEN to the modem

Outputs:  
ML + 18 contains the number of bytes read  
ML + 19 is 255 for a timeout or packet formatting error  
ML + 20 is the packet/channel ID that sent the data (0=none)  
ML + 22 is the size of the packet received  
ML + 24 is the EXPECTED CRC8 of the packet data sent by the modem  
ML + 26 contains the CRC8 value of the received string

Reads a special C64Net WiFi Firmware formatted Packet from modem channel. This function will block until it times out, or a packet is received, so it's best used with the ATF3 flow control to ensure that every time XON is sent to the modem, a single packet header is guaranteed to be sent to the computer. When this method returns, ML+19 should be checked for an error, ML+22 for a non-0 to see if any data was received, or ML+20 to make sure it matches the expected channel (this can be 0 for a non-response when using ATF3). You can also optionally compare ML+26 to ML+24 to check for RS232 errors.

## ***CRCP***

Usage:  
sys ML+9

Pre-requisites:  
P\$ contains the string to calculate the CRC of

Outputs:  
ML + 26 contains the CRC8 value of the received string

Description:  
Calculates the CRC8 of P\$ and returns it.



## ***BUFAP***

Usage:  
sys ML+12

Pre-requisites:  
P\$ must be initialized in BASIC  
ML + 32 contains the channel number to read from  
The above channel must be open

Outputs:  
P\$ contains the bytes read from the channel, if any  
ML + 26 contains the CRC8 value of the received bytes

Description:  
Reads from the open channel whose number is contained in ML+32 until no more bytes are available, or until 254 bytes are read. The bytes are returned in the BASIC variable P\$.

## **UP9600.BIN**

This file for the Commodore 64 (not the Commodore 128 in 64 mode) is a machine language kernel wedge that supports the 9600 baud UP9600 feature of the C64Net WiFi Modem. It resides in memory at \$C800 (51200), and must be initialized before use by calling SYS 51200.

Once initialized, the normal BASIC commands for using the modem will be modified to support UP9600 modems. The OPEN command can now be opened for 9600 baud use by entering OPEN 5,2,0,CHR\$(14), as 14 is the baud rate code for 9600 baud. GET, PRINT, INPUT and other commands can then be used to read and write to the modem at speed. If the modem channel is opened at any other baud rate, the wedge will remain inactive and the normal CBM Kernal will operate.

Machine language KERNAL routines are likewise wedged to support the UP9600 baud rates. OPEN, CLOSE, CHKIN, CKOUT, CHRIN, GETIN, and CHROUT are all supported. As with BASIC, the wedge only operates when the OPEN command has been issued with a baud rate code of 14. Obviously, the modem must also already be set to 9600 baud for any of this to work.

It should be noted that, due to the way the driver works, IEC/Disk access cannot occur at the exact same time as the modem is receiving data. The driver tries to prevent disk issues by disabling itself whenever an IEC/Disk channel is selected, but this should be kept in mind in your application development.

Below is a slightly more detailed explanation of the routines in UP9600.BIN. Throughout this section, I will refer to the variable UM to refer to the starting address of the library, which is 51200 or \$C800. So, if you see UM+3, it means 51203.

### ***INIT***

Usage:  
sys UM

Pre-requisites:  
None

Outputs:  
None

Description:  
Initializes the wedge by replacing the standard KERNAL vectors for OPEN, CLOSE, CHKIN, CKOUT, CHRIN, GETIN, and CHROUT with its own routines.

### ***INSTALL***

Usage:  
sys UM+3

Pre-requisites:  
Memory locations 247 and 248 point to a 256 byte input buffer

Outputs:  
Memory location 789 has something other than the value 234 if this call is successful.

Post-requisites:  
You should put a non-zero value into memory location UM+19, such as 1, after calling this function, for the KERNAL wedge to recognize that the driver is waiting for 9600 baud data from the modem.

Description:  
Checks for the presence of UP9600 hardware, and if found, Installs and Enables the UP9600 baud driver. This should only be called once, ever. It is called by the OPEN BASIC and KERNAL command automatically. If the driver is disabled with the DISABLE call, then call ENABLE below to re-enable it.

## ***ENABLE***

Usage:  
sys UM+6

Pre-requisites:  
INSTALL must have been called  
Memory location 247 and 248 point to a 256 byte input buffer

Outputs:  
Memory location 789 has something other than the value 234 if this call is successful.

Description:  
Enables or Re-Enables the UP9600 baud driver. It is called by INSTALL automatically.

## ***DISABLE***

Usage:  
sys UM+9

Pre-requisites:  
ENABLE or INSTALL must have been called

Outputs:  
Memory location 789 has the value 234 when this completes.

Description:  
Disables the UP9600 baud driver, allowing IEC/Disk access to be performed.

## ***RSOUT***

Usage:  
sys UM+12

Pre-requisites:  
INSTALL or ENABLE must have called  
The accumulator .A contains the byte to transmit

Outputs:  
None

Description:  
Transmits a byte at 9600 baud, in real time.

## ***RSIN***

Usage:  
sys UM+15

Pre-requisites:  
INSTALL or ENABLE must have called

Outputs:  
The accumulator .A contains the byte received, if any.  
The Carry flag is Set if there was an error, Clear if a byte is in the accumulator.

Description:  
Reads a byte from the 9600 baud input buffer.

## ***RDS64.BIN***

This file is the machine language portion of the TELNETD server. It resides as \$c000 and can be activated with SYS49152.

## ***X-XFER64.BIN***

This file is the X-Modem protocol used by CBMBBS64-128. It resides at \$c800 with the following jump table:  
\$c800 SendFile (upload)  
\$c803 Receive Checksum (download)  
\$c806 SendFile (upload)  
\$c809 Receive CRC (download)  
All functions require the modem opened to channel 5, and the appropriate file on channel 2.  
The C= key can be used to abort.

## ***X-XFER128.BIN***

This file is the X-Modem protocol used by CBMBBS64-128. It resides at \$c800 in bank 0 with the following jump table:  
\$c800 SendFile (upload)  
\$c803 Receive Checksum (download)  
\$c806 SendFile (upload)  
\$c809 Receive CRC (download)  
All functions require the modem opened to channel 5, and the appropriate file on channel 2.  
Also be aware that it will copy a small 16 byte bit of bank-switch code to \$1BE0. During transfers, the C= key can be used to abort.

## ADVANCED TOPICS

The C64Net Wifi card is fully programmable. The tools used to create the firmware include the Arduino IDE and the ESP8266 libraries. Additionally, the card can be programmed (i.e. firmware downloaded) by using an FTDI cable or over the air using WiFi.

Developing new firmware or modifying the current firmware will be left as an exercise for the user. However, this section will provide details on what is required to configure the hardware and downloading a new version of the firmware.

The C64Net card is based on the ESP8266 by Expressif. This is a feature rich WiFi module that contains 1MB of flash onboard. This flash can be updated using the Arduino IDE or the ESP8266 Flash Downloader. It can also be updated over the air using an AT command. To perform the updates, the following instructions are provided.

### FTDI Cable Download

To perform the firmware update from the Arduino IDE or using the ESP8266 Flash Downloader, an FTDI cable is required. The FTDI model TTL-232R-5.0V cable is required.

Configuring the hardware for firmware downloading

The C64Net card must not be installed in the computer. It must be configured as standalone with the jumpers set to allow programming over the FTDI Cable. **Attaching the FTDI cable to the card while it is installed on the computer will not work and may result in damage.**

1. *Build or acquire the firmware file you wish to download to the C64Net card.*
2. *Attach the FTDI cable to P1 on the C64Net card.*
3. *Connect the USB end of the cable to your PC.*
4. *Set jumpers P6 and P7 to the FTDI position.*
5. *Start the firmware download.*
6. *Disconnect the FTDI cable*
7. *Set the jumpers P6 and P7 to the C64 position.*
8. *Install the C64Net card on the computer.*

## Over the Air Download

To perform the over the air download, the C64Net card must be configured and up and running on your C64 or other commodore computer with a communications program.

Issuing the following command will perform the update from the default server.

<b>AT&amp;U6502</b>	<b>Will update the firmware from the home page on the web.</b>
<b>AT&amp;U=x</b>	Will update the firmware from the web to custom version x.

The over the air update restricts the size of the update to 512K bytes or less. If you are building a custom firmware that is larger, then the FTDI Cable download method must be used.

## AT COMMAND REFERENCE

Command Set:

-----

The command set is as follows (not case sensitive):

COMMAND	Description
<b>ATZ</b>	closes all open socket connections (preserving the Access Point connection), stops all listeners, and resets the state of the Command processor to the saved configuration, preserving the current baud rate and wifi connection.
<b>ATI</b>	re-shows the startup message, including wifi connection information.
<b>ATI0</b>	same as ATI
<b>ATI1</b>	Shows the current common variable settings, common 'S' registers.
<b>ATI2</b>	Shows the modem's current IP address
<b>ATI3</b>	Shows the modem's current Wireless Router connection
<b>ATI4</b>	Shows only the firmware current version
<b>ATI5</b>	Shows all the current variable settings, all 'S' registers.
<b>ATA</b>	If a server listener has generated a RING, then ATA will switch the last rung connection to Stream mode (see ATD). ATA/ : Repeats the previous command (no idea why...)
<b>ATAn</b>	Causes the modem to create a server listening on port n. When a connection is received, the terminal will generate 1 or more RINGs according to the ATS0 register, followed by a normal CONNECT response. At this point, all other commands related to connections may be used normally, unless ATS41 is > 0, in which case incoming connections are automatically sent to Stream mode as per ATD or ATA. Listeners are listed along with other connections using ATC0.
<b>ATAPn" [HOSTNAME] : [PORT] "</b>	Adding a P modifier causes all incoming connection input to be translated to PETSCII
<b>ATATn" [HOSTNAME] : [PORT] "</b>	Adding a T modifier causes connection streaming input to be translated per TELNET when the changed to Stream mode

--	--

<b>COMMAND</b>	<b>Description</b>
<b>ATAEn</b> " [HOSTNAME] : [PORT] "	Adding a E modifier causes connection terminal echo to be enabled when the changed to Stream mode
<b>:</b> <b>ATAXn</b> " [HOSTNAME] : [PORT] "	Adding a X modifier causes connection XON/XOFF flow control to be enabled when the changed to Stream mode
<b>ATN0</b>	Shuts down all listeners, leaving client connections open ATNn : if n > 0 then same as ATAn
<b>ATE0</b>	Turns serial terminal echo off for command mode.
<b>ATE1</b>	Turns serial terminal echo on for command mode.
<b>ATV0</b>	Turns off verbose responses mode (Uses Terse Numeric response mode)
<b>ATV1</b>	Turns on verbose responses mode (Uses Word response mode)
<b>ATX0</b>	Turns off extended response codes (1/CONNECT instead of 5/CONNECT 2, etc..)
<b>ATX1</b>	Turns on extended response codes (5/CONNECT 2 instead of 1/CONNECT, etc..)
<b>ATF0</b>	Turns on rts/cts flow control.
<b>ATF1</b>	Turns on xon/xoff flow control.
<b>ATF2</b>	Turns on xon/xoff flow control, sets XON mode (if necessary), and, in command mode, will immediately go to XOFF when a single connection packet is received. This is very useful when the client wants to ensure it only receives one packet to process. You can think of this as an alternative way to use xon/xoff by having XOFF automatic between packets.
<b>ATF3</b>	Similar to ATF2 except that the default is XOFF, and, in command mode, a XON code from the user will immediately trigger either an empty packet response [ 0 0 0 ], or a real packet if one is available. After this, as in ATF2, XOFF is automatically set. ATF4 : Turns off flow control for command mode
<b>ATQ0</b>	Turns off quiet mode (Sends response codes)
<b>ATQ1</b>	Turns on quiet mode (Stops sending response codes)



<b>ATRO</b>	Suppresses linefeed (\n \$0a) in end of lines. Will only send carriage return (\r \$0d).
-------------	--

<b>COMMAND</b>	<b>Description</b>
<b>ATR1</b>	Sends \r\n (\$0d0a) as end of line string.
<b>ATR2</b>	Sends \n\r (\$0a0d) as end of line string.
<b>ATR3</b>	Suppresses carriage return (\r \$0d) in end of lines. Will only send linefeed (\n \$0a).
<b>ATBn</b>	Sets a new serial Baud Rate. Takes effect immediately.
<b>ATB"n,xYz"</b>	Sets baud rate n, bits x, parity (E,O,M, or N) for Y, and stop bits z.
<b>ATW</b>	List all wireless network access points scanned within range. The response for each entry is the SSID, following by the RSSI, followed by an * character is the connection is encrypted.
<b>ATWn</b>	Where n > 0, this lists up to n wireless network access points scanned within range. The response for each entry is the SSID, following by the RSSI, followed by an * character is the connection is encrypted.
<b>ATW" [SSID] , [PASSWORD] "</b>	Connects to the wireless access point with the given SSID, using the given password.
<b>ATWP</b>	Adding a P modifier is the same as all forms of ATW, with both arguments and results presented in PETSCII.
<b>ATD</b>	Start a streaming connection between the current opened connection. Use "+++" to exit back to Command mode.
<b>ATDn</b>	Where n > 0, this will start a streaming connection between the previously opened connection with an id the same as n. Use "+++" to exit back to Command mode.
<b>ATD" [HOSTNAME] : [PORT] "</b>	This opens a streaming connection between the terminal and the given host/port. Use "+++" to disconnect and exit back to command mode.
<b>ATDP" [HOSTNAME] : [PORT] "</b>	Adding a P modifier causes connection input to be translated to PETSCII during the streaming session.
<b>ATDT" [HOSTNAME] : [PORT] "</b>	Adding a T modifier causes connection input to be translated per TELNET during the streaming session.
<b>ATDE" [HOSTNAME] : [PORT] "</b>	Adding a E modifier causes terminal echo to be enabled that streaming session.
<b>ATDX" [HOSTNAME] : [PORT] "</b>	Adding a X modifier causes XON/XOFF flow control to be enabled that streaming session.
<b>ATDnnnnnnn</b>	Where n=0-9, if the digits exist in the phonebook (see ATP), it will try connect to that host, with those modifiers, from the phonebook.

<b>ATC</b>	Shows information about the current network connection in the following format "[CONNECTION STATE] [CONNECTION ID] [CONNECTED TO HOST]:[CONNECTED TO PORT]"
------------	---

<b>COMMAND</b>	<b>Description</b>
<b>ATC0</b>	Lists information about all of the network connections in the following format "[CONNECTION STATE] [CONNECTION ID] [CONNECTED TO HOST]:[CONNECTED TO PORT]", including any Server (ATA) listeners.
<b>ATCn</b>	Where n > 0, this changes the Current connection to the one with the given ID. If no connection exists with the given id, ERROR is returned.
<b>ATC" [HOSTNAME] : [PORT] "</b>	Creates a new connection to the given host and port, assigning a new id if the connection is successful, and making this connection the new Current connection. The quotes and colon are required.
<b>ATCP" [HOSTNAME] : [PORT] "</b>	Adding a P modifier causes all connection input to be translated to PETSCII.
<b>ATCT" [HOSTNAME] : [PORT] "</b>	Adding a T modifier causes streaming input to be translated per TELNET when the changed to Stream mode.
<b>ATCE" [HOSTNAME] : [PORT] "</b>	Adding an E modifier causes terminal echo to be enabled when the changed to Stream mode.
<b>ATCX" [HOSTNAME] : [PORT] "</b>	Adding a X modifier causes XON/XOFF flow control to be enabled when the changed to Stream mode
<b>ATH</b>	Hangs up (disconnects and deletes) all open connections. Does not close Server listeners.
<b>ATH0</b>	Hangs up (disconnects and deletes) the current opened connection.
<b>ATHn</b>	Hangs up (disconnects and deletes) the open connection with the given id. Closing a Server (ATA) listener does not close any connections received from that listener.
<b>ATO</b>	Re-enters a Streaming session (see ATD) under the previous settings, with the current (previous) connection.
<b>ATP</b>	Lists all existing phonebook entries, with the format phone number followed by ATD modifiers, followed by the host and port.

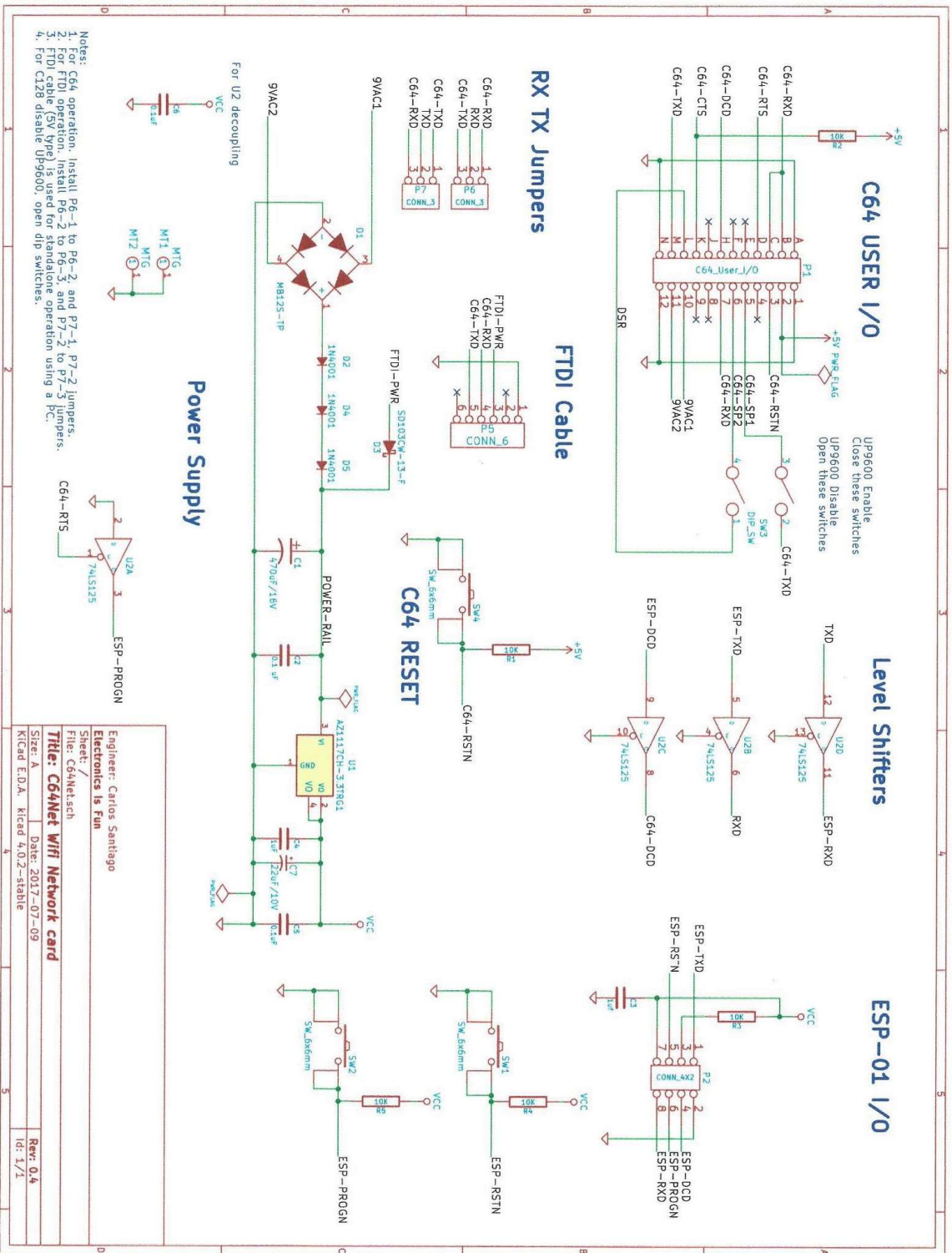
<b>COMMAND</b>	<b>Description</b>
<b>ATP" [NUMBER]=[HOSTNAME] : [PORT] "</b>	Adds or Modifies an entry to the phonebook with the given 7 digit number, host, and port. Use ATDnnnnn.. to connect.
<b>ATPP" [NUMBER]=[HOSTNAME] : [PORT] "</b>	Adding a P modifier causes connection input to be translated to PETSII when connected to that entry.
<b>ATPT" [NUMBER]=[HOSTNAME] : [PORT] "</b>	Adding a T modifier causes connection input to be translated per TELNET when connected to that entry.
<b>ATPE" [NUMBER]=[HOSTNAME] : [PORT] "</b>	Adding a E modifier causes terminal echo to be enabled when connected to that entry.
<b>ATPX" [NUMBER]=[HOSTNAME] : [PORT] "</b>	Adding a X modifier causes XON/XOFF flow control to be enabled when connected to that entry.
<b>ATP" [NUMBER]=DELETE"</b>	Removes the phonebook entry with the given number.
<b>ATS0=n</b>	Changes the number of RING messages received before a CONNECT response is sent, on incoming Server listeners.
<b>ATS1=n</b>	Unimplemented, always returns OK
<b>ATS2=n</b>	Change the escape character (n = 0-255), Defaults to ASCII decimal 43 ("+")
<b>ATS3=n</b>	Change the Carriage Return Character (n = 0-127), Defaults to ASCII decimal 13 (Carriage Return)
<b>ATS4=n</b>	Change the Line Feed Character (0-127), Defaults ASCII decimal 10 (Line Feed)
<b>ATS5=n</b>	Change the Backspace Character (0-32), ASCII decimal 8 (Backspace)
<b>ATS6 ... 39=n</b>	Unimplemented, always returns OK
<b>ATS40=n</b>	Change the size of the connection packets (n > 0), Defaults to 127 bytes
<b>ATS41=n</b>	When n > 0, all incoming Server listener connections are immediately sent to Stream mode. If n=0, connections remain in normal command mode (default).
<b>ATS42=n</b>	Set the CRC8 for an expected incoming Transmit command. e.g. ATS42=123T"[MESSAGE]" returns error unless 123 is CRC8 of "[MESSAGE]".
<b>ATS43=n</b>	Sets a standby baud rate n for the next incoming or outgoing connection only. ATZ clears.

<b>COMMAND</b>	<b>Description</b>
<b>ATS44=n</b>	Sets an automatic delay of n milliseconds after most bytes written to the Serial port. This is for computers that support a baud rate, but can't really keep up, and you don't want to use flow control.
<b>ATS45=n</b>	Changes how packet and at&g data is delivered. 0 is normal binary with normal headers, 1 is 78 char HEX digit streams followed by EOLN with hex digit headers, 2 is decimal digits followed by EOLN, with decimal digit headers.
<b>ATS46=n</b>	Changes DCD status. n=0 is default DCD=HIGH=online. n=1 is DCD=LOW=online
<b>ATS47=n</b>	Changes CTS status. n=0 is default CTS=HIGH=active. n=1 is CTS=LOW=active
<b>ATS48=n</b>	Changes RTS status. n=0 is default RTS=HIGH=active. n=1 is RTS=LOW=active (N/A on ESP01)
<b>ATS49=n</b>	Changes DCD pin number, n=2 is default
<b>ATS50=n</b>	Changes CTS pin number, n=0 is default on ESP01, and default is 5 otherwise
<b>ATS51=n</b>	Changes RTS pin number, n=4 is default (N/A on ESP01) +++ : With a 1 second pause with no other characters afterwards, this will disconnect the current opened connection.
<b>ATT" [MESSAGE] "</b>	Transmit the given text string, with \r\n at the end, on the current connection.
<b>ATTn</b>	Where n > 0, Starts a transmission of exactly n bytes to the current connection. The \n from entering this command must be followed by the n bytes to transmit.
<b>ATTP" [MESSAGE] "</b>	Transmit the given text string, translating petSCII to ascii, with \r\n at the end, on the current connection.
<b>ATTPn</b>	Where n > 0, this will start the transmission of exactly n bytes to the current connection, translating PETSCII to ASCII. The \n from entering this command must be followed by the n bytes to transmit.
<b>COMMAND</b>	<b>Description</b>

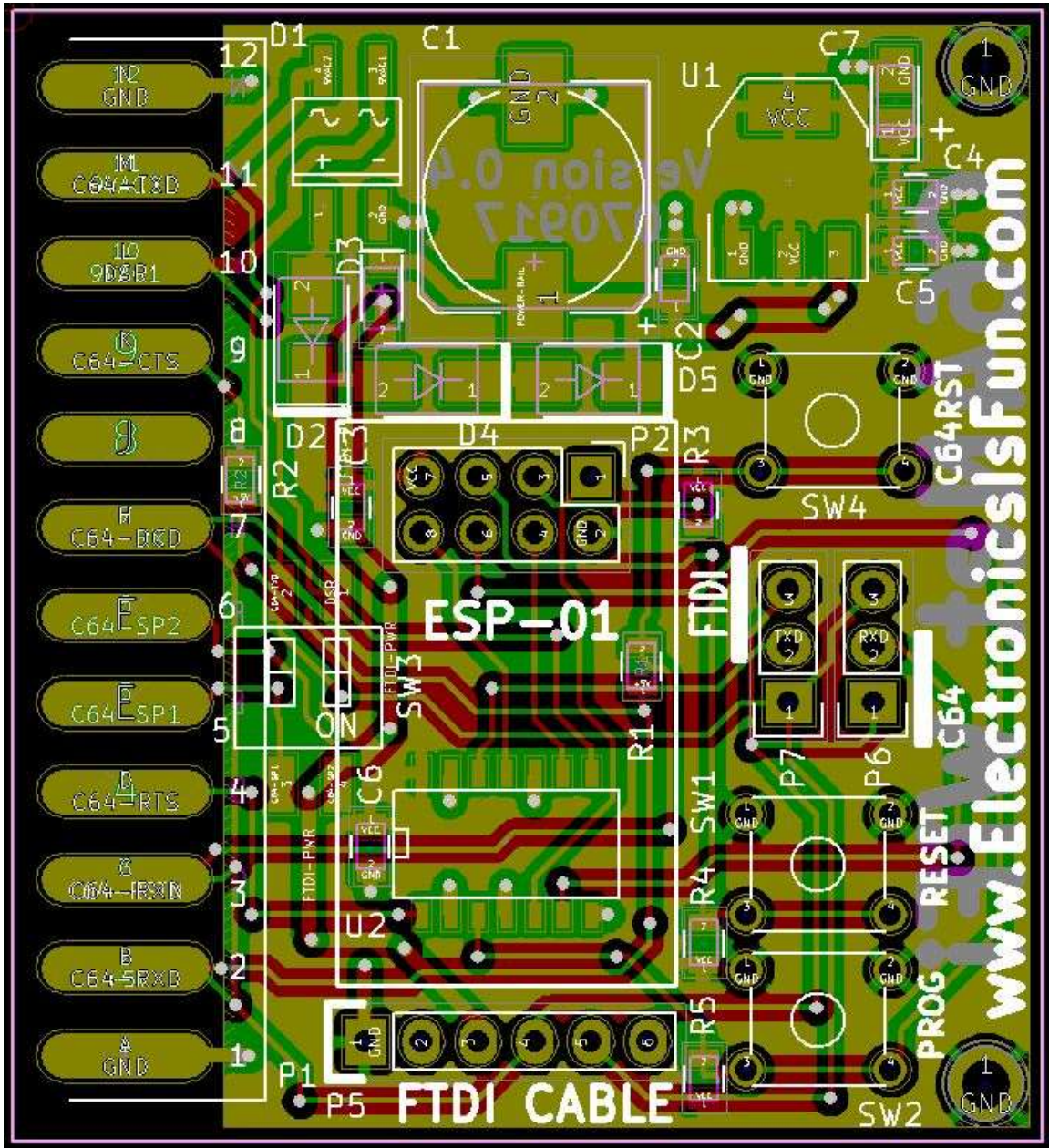
<b>AT&amp;L</b>	Reloads the saved configuration.
<b>AT&amp;W</b>	Saves the current configuration: WiFi settings(ATW), baud rate (ATB), end of line (ATR) settings, flow control (ATF), echo mode (ATE), extended responses (ATX), verbose responses (ATV), quiet responses (ATQ), PETSCII mode (AT&P1), DCD status (ATS46), CTS status (ATS47).
<b>AT&amp;F</b>	Restores the modem to factory default settings.
<b>AT&amp;On</b>	n is 1 to turn on internal serial-reception log, n is 0 to turn off or view a previously turned-on log.
<b>AT&amp;H</b>	Shows a help file from the web, or brief help otherwise.
<b>AT&amp;U</b>	Checks the firmware home page to see if a new version is available.
<b>(BETA) AT&amp;U6502</b>	Will update the firmware from the home page on the web.
<b>(BETA) AT&amp;U=x</b>	Will update the firmware from the web to custom version x.
<b>AT&amp;Pn</b>	Where n > 0, all command mode input and output will be translated to/from PETSCII before internal processing. This will not affect received packet data, or the stream mode.
<b>AT&amp;Nx</b>	Shows the status of ESP8266 I/O pin x
<b>AT&amp;Mn</b>	Adds the byte denoted by n to a list of mask-out bytes. These are bytes that are not transmitted to the serial port in command mode incoming packets. If this command is followed by a C, N, or A command on the SAME LINE, then the setting will apply ONLY to that connection or listener.
<b>AT&amp;M</b>	Resets the mask-out bytes list. No bytes will be masked-out. If this command is followed by a C, N, or A command on the SAME LINE, then the setting will apply ONLY to that connection or listener.

COMMAND	Description
<b>AT&amp;D</b>	Resets the delimiter bytes lis. No bytes will be delimited, and packets will contain as many bytes as are received and allowed by ATS40. If this command is followed by a C, N, or A command on the SAME LINE, then the setting will apply ONLY to that connection or listener.
<b>AT&amp;G" [HOSTNAME] : [PORT] / [FILENAME] "</b>	Streams a file from an HTTP source on the internet. The header contains channel 0, file length, and an 8-bit sum of all the bytes in the forthcoming file, followed by the bytes of the file, all formatted as a normal packet. An ASCII 3 (CNTRL-C) received during the transfer will abort. The S44 register can be used to create artificial delays in this output. XON/XOFF Flow control also remains in effect with, on a byte-by-byte basis for the auto and manual flow control systems.

# SCHEMATICS



# BOARD LAYOUT





# GLOSSARY

<b>Term</b>	<b>Definition</b>
<b>SSID</b>	SSID is the service set identifier. Also known as the network name. It is the name defined in your router.
<b>IP Address</b>	Internet Protocol Address. Every device on the internet is assigned a unique address that is usually (4) bytes long for example 192.168.1.100.
<b>WiFi</b>	The wireless standard defined as 802.11a, b, g, or n.
<b>BBS</b>	Bulletin Board System.
<b>Bootstrap</b>	A small program that starts the process of loading other larger programs, or an operating system.

# INDEX

AT Command Reference, 7, 28  
AT&D, 35  
AT&F, 34  
AT&G, 35  
AT&H, 34  
AT&L, 34  
AT&M, 34  
AT&Mn, 34  
AT&Nx, 34  
AT&On, 34  
AT&Pn, 34  
AT&U, 27, 34  
AT&U6502, 27, 34  
AT&W, 34  
ATA, 28, 31  
ATAEn, 29  
ATAn, 28, 29  
ATAPn, 28  
ATATn, 28  
ATAXn, 29  
ATB, 30, 34  
ATBn, 30  
ATC, 31  
ATC0, 28, 31  
ATCE, 31  
ATCn, 31  
ATCP, 31  
ATCT, 31  
ATCX, 31  
ATD, 11, 28, 30, 31  
ATDE, 30  
ATDnnnnnnn, 30  
ATDP, 30  
ATDT, 30  
ATE0, 29  
ATE1, 29  
ATF0, 29  
ATF1, 29  
ATF2, 29  
ATF3, 21, 29  
ATH, 31  
ATH0, 31  
ATHn, 31  
ATI, 28  
ATI1, 28  
ATI2, 28  
ATI3, 28  
ATI4, 28  
ATI5, 28  
ATN0, 29  
ATO, 31  
ATP, 30, 31, 32  
ATPE, 32  
ATPP, 32

ATPT, 32  
 ATPX, 32  
 ATQ0, 29  
 ATQ1, 29  
 ATR0, 30  
 ATR1, 30  
 ATR2, 30  
 ATR3, 30  
 ATSO, 28, 32  
 ATS1, 32  
 ATS2, 32  
 ATS3, 32  
 ATS4, 32  
 ATS40, 32, 35  
 ATS41, 28, 32  
 ATS42, 32  
 ATS43, 32  
 ATS44, 33  
 ATS45, 33  
 ATS46, 33, 34  
 ATS47, 33, 34  
 ATS48, 33  
 ATS49, 33  
 ATS5, 32  
 ATS50, 33  
 ATS51, 33  
 ATS6, 32  
 ATT, 33  
 ATTn, 33  
 ATTP, 33  
 ATTPn, 33  
 ATV0, 29  
 ATV1, 29  
 ATW, 4, 30, 34  
 ATWn, 30  
 ATWP, 30  
 ATX0, 29  
 ATX1, 29  
 ATZ, 5, 28, 32  
 Board Layout, 37  
 Bootstrapping, 4  
 C64Net card, 1, 2, 4, 7, 8  
 C64Net Programs, 10  
 CBMBBS64-128, 10, 16, 25  
 Commodore 64, i, 1, 2, 10, 12, 14, 15, 18,  
 22  
 CONFIGURE64-128, 2, 10, 11  
 Configuring the SSID and password for  
 your network, 7  
 D64WGET64-128, 10, 17  
 Dial from Phonebook, 16, 17  
 Display available hotspots/networks, 8  
 FTDI Cable Download, 26  
 FTP64-128, 10, 12  
 Getting help, 8  
 Getting Started, 3  
 IRC64-128, 10  
 Modify Phonebook, 16, 17  
 Over the Air Download, 27  
 PETSII vs. ASCII, 8  
 PLM64.BIN, 10  
 PML128.BIN, 10, 11, 12, 14, 15, 16, 18,  
 19  
 Quick Connect, 16, 17  
 Reset to factory settings, 9  
 Saving your SSID and password, 7  
 Schematics, 36  
 Set the baud rate, 7  
 TELNET, 10, 11, 15, 16, 28, 30, 31, 32  
 TELNET64-128, 10, 15  
 TELNETD64, 10, 18, 19  
 Terminal Mode, 16, 17  
 WEATHER, 10, 19  
 WGET64-128, 10, 14