Commodore WiFi Modem

Contents

About the C64 MicroView and Wi-Fi Modem 2

Configurations 3

Default Configuration 3

Firmware 3

Baud rate selection 3

C64 Directly Connected to Wi-Fi Module 4

Standalone 5

MicroView as an Interface Device to the C64 5

MicroView Pinout 6

Schematic 8

Software Support 9

UP9600 Support 9

CommodoreServer Support 9

QuantumLink RELOADED 10

Hayes Commands 11

About the C64 MicroView and Wi-Fi Modem

This is a multipurpose board that allows for experimentation with the Commodore 64 (or other Commodore computer with a compatible User Port like the C128 or VIC20), the MicroView Arduino-compatible device with OLED display, and a serial-based Wi-Fi Module such as the Roving Networks RN-XV Module or the XBee® Wi-Fi Module, in a variety of configurations.

All I/O pins on the MicroView are broken out along the top of the board to allow for interfacing to external devices. Refer to the MicroView documentation and **Table 1** for details.

Current-limiting resistors are recommended between all user port lines and the board. A socket is provided for this purpose at U1, these can be either individual 100-330 Ohm resistors, or a resistor network DIP such as the Bourns Inc. 4116R-1-101LF (provided) can be used. The Commodore SX-64 requires 100 Ohm resisters while most other models will work with 330 Ohm.

A separate VIN connector is provided for powering the board for standalone use, this is <u>not</u> required when the board is connected to a computer.

Reset buttons are provided for resetting each of the C64, MicroView, and Wi-Fi Module.

All parts are through-hole for ease of soldering.

Configurations

Default Configuration

The **default configuration** allows the MicroView to arbitrate between the C64 and the RN-XV, acting like a simple virtual modem, assuming a sketch is loaded into the MicroView to perform this function.

1. TX and RX of the C64 User Port connected to pins D6 and D5 of the MicroView (Software Serial RX/TX #1).

2. TX and RX of the Wi-Fi Module connected to pins D1 and D0 of the MicroView (Hardware Serial)

3. No jumpers are installed on the "TX" and "RX" jumpers.

4. Power is supplied to the MicroView and RN-XV from the C64 User Port.

5. All RS-232 signal lines are connected between the C64 and MicroView for enhanced modem emulation, through current-limiting resistors provided in the resistor network DIP at U1.

Firmware

Two firmware options are available:

Menu (WiFiModem-Menu-*.hex):

Simple menu driven which allows you to connect to:

1. Connect to remote hosts
2. Dial from a phone book and configure an auto-connect entry
3. Accept incoming calls
4. Configure WiFi settings

WiFiModem-Hayes-*.hex

Hayes emulation. Should work with most terminal programs. See the Hayes Command section for supported AT commands.

Baud rate selection

Immediately after bootup, 'Baud Detection' will be displayed on the MicroView. Pressing a key on the keyboard within three seconds will detect the baud rate and store it in flash for future bootups. Some keys such as space may cause an incorrect baud rate to be detected. The most reliable key to press is the letter u.

C64 Directly Connected to Wi-Fi Module

This configuration connects the TX and RX from the User Port directly to the Wi-Fi Module’s RX and TX pins. Ensure the Wi-Fi Module’s serial port is configured to match the baud rate of the C64.   Refer to the Wi-Fi Module’s documentation for configuration and command set details.

1. Remove the MicroView from the board.

2. Place jumpers on the jumper pins marked “TX” and “RX”.

Standalone

This configuration allows the board to be used standalone, without being connected to a computer.

1. Remove the cartridge from the User Port.

2. Supply +3.3V to +16V to the board on the VIN connector. The connector is a standard 2.1 mm barrel connector, center-pin positive. This connects to the MicroView's VIN pin which regulates the input voltage. The board requires about 200 mA @ 5V.

Note: Do not connect the external VIN when the board is connected to the User Port!

MicroView as an Interface Device to the C64

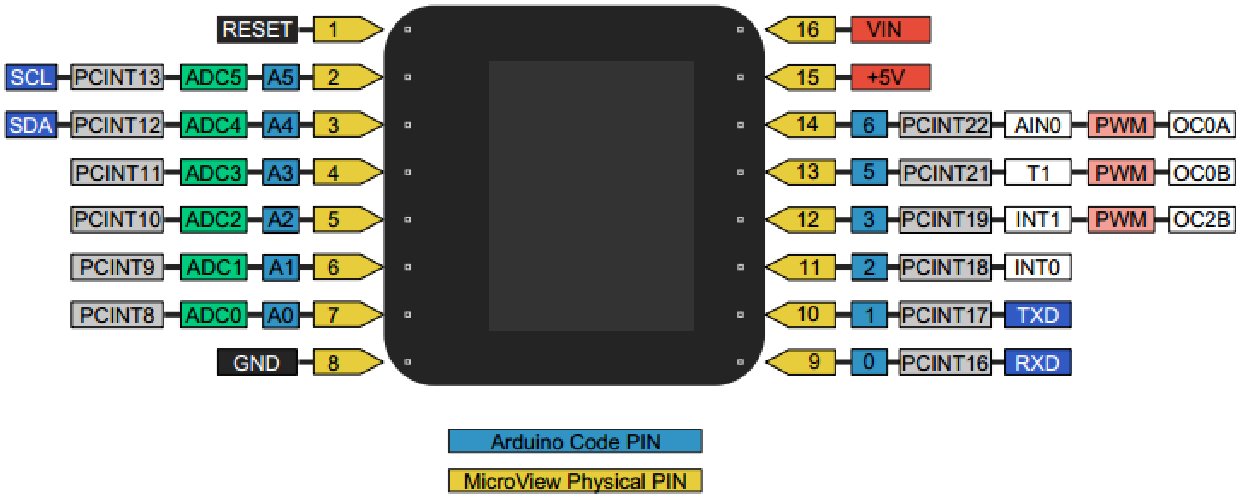
This configuration allows the MicroView's GPIO (including analog input, PWM output, digitial I/O and even I^2C) to be used to interface to the outside world, with data exchanged to the C64 via serial port. A sketch is required inside the MicroView to perform this function.

1. Remove the Wi-Fi Module from the board.

2. Note that some of the MicroView pins are shared with the User Port pins – see **Table 1**. Therefore it is recommended to remove the resistor network and provide standalone resistors only between TxD and RxD at U1. This frees up the Analog GPIO A0 to A5.

MicroView Pinout

At the heart of the C64 Wifi Modem is a “Microview”, an a Arduino Uno-compatible device with built-in OLED display. The pinout is as follows:



All the MicroView pins are broken out along the top of the cart, to allow interfacing to the outside world.   Note that some pins are connected to the C64 User Port, as per **Table 1** below.

Table 1 – MicroView Pinout

Pin#	MicroView	C64 User Port	Wi-Fi	Notes
1	RESET	-	-	-
2	A5 SCL	Pin D (RTS)	-	For extended RS-232 support
3	A4 SDA	Pin E (DTR)	-	For extended RS-232 support
4	A3	Pin F (RI)	-	For extended RS-232 support
5	A2	Pin H (DCD)	-	For extended RS-232 support
6	A1	Pin K (CTS)	-	For extended RS-232 support
7	A0	Pin L (DSR) Pin 6 (CNT2)*	-	For extended RS-232 support (*Connected to Pin 6 for UP9600 mode when jumper is in place)
8	GND	Pins 1,12,A,N (GND)	Pin 10 (GND)	Ground
9	D0 RxD	-	Pin 2 (TxD)	Hardware Serial RxD.
10	D1 TxD	-	Pin 3 (RxD)	Hardware Serial TxD.
11	D2	-	Pin 12 (RTS)	Serial RTS from Wi-Fi
12	D3	-	Pin 16 (CTS)	Serial CTS to Wi-Fi
13	D5	Pin M (TxD) Pin 5 (SP1)	-	Software Serial RxD (Pin 5 is for UP9600 mode)
14	D6	Pins B,C (RxD) Pin 7 (SP2)	Pins B,C (RxD) Pin 7 (SP2)	Software Serial TxD (Pin 7 is for UP9600 mode)
15	+5V (from C64)	Pin 2 (+5V to MicroView)	Pin 2 (+5V to MicroView)	+5V Power from C64 to MicroView In standalone mode, +5V available here
16	VIN	-	-	External VIN **for standalone use** +3.3V to +16V

Schematic

Software Support

UP9600 Support

The UP9600 driver for 9600 baud support in Novaterm 9.6 is supported. Add the jumper marked UP9600 Enable to enable it. Note that this prevents Pin L (DSR) from being used.

The UP9600 driver is also supported in Striketerm 2014 (based on Novaterm 9.6), but a workaround is required: The driver must be selected and the configuration saved with the Wi-Fi Modem unplugged. Plug in the modem (with the C64 power off) and restart Striketerm and it will then work normally.

CommodoreServer Support

www.commodoreserver.com is supported at 2400 baud. The recommended way to connect is to configure the WiFi modem to auto-connect to the commodoreserver server. To configure, follow these steps:

1. Make sure the modem is set for 2400 baud. During power-up, the MicroView will display the baud rate. If it is not set to 2400 baud, set the baud rate using a terminal program such as NovaTerm. See the section for Baud Rate Selection.
2. Configure auto-connect to connect to www.commodoreserver.com:1541 at power-up
    a. Menu firmware:
        i. Set one of the phone book entries to www.commodoreserver.com:1541
        ii. Configure auto-connect for the phone book entry.
    b. Hayes firmware:
        i. Set one of the phone book entries to www.commodoreserver.com:1541 using at&pb*x*=www.commodoreserver.com:1541 where x is a phone book entry from 1-9.
        ii. Configure auto-connect using at&pbauto=*x* where x matches the phone book entry above.
3. Reboot the computer and WiFi modem
4. load “v-1541”,8,1
5. sys 49152
6. new
7. V-1541 and documentation can be found at http://www.commodoreserver.com/Downloads.asp

QuantumLink RELOADED

Currently not supported

Hayes Commands

Defaults in **bold**.

Command	Description	Comments
at&pass=	Set WiFi passphrase, if required	Example: at&pass=Commodore64
at&key=	Set WiFi key, if required	
at&ssid=	Set WiFi SSID	Configures SSID and attemps to connect. Configure pass or key before ssid.
at&port=	Set TCP listening port for answering calls	If set to 0, incoming calls will be disabled. Modifying the port requires a reboot of the WiFly. At 2400 baud, the reboot will be handled by the MicroView but at other baud rates the end-user must reboot both the MicroView and the WiFly module. Port settings is saved immediately. at&w is not required.
at&pb?	List phone book entries	
at&pbclear	Clear all phone book entries	Also pre-populates phone book entries 8 and 9 with bbs.jammingsignal.com:23 and www.commodoreserver.com:1541 Phone book settings are saved immediately. at&w is not required.
at&pb#=*address*	Modify a phone book entry	Example: at&pb1=bbs.jammingsignal.com:23 Phone book settings are saved immediately. at&w is not required.
at&pbauto=	Set phone book entry for auto-connect at bootup	Set to 0 to disable. Phone book settings are saved immediately. at&w is not required.
ati	Display modem configuration information	Displays MAC address, IP address, SSID, Firmware version, listen port and AT init string.
ata	Answer incoming call	If an inbound connection has been created by the WiFly, RING will be sent to the computer. ata will answer the call.
ate / ate0	Disable echo	
ate1	**Enable echo**	
atf	Load modem defaults	Same at atz.
ath / ath0	Force modem on-hook (hang-up)	+++ during outbound calls currently drops the call. Future versions will drop to command mode which will then allow you to use ath / ath0 to drop the call. +++ during inbound calls goes to AT command mode. ath/ath0 will then disconnect the call.
ath1	Not implemented	Not implemented
atq0	Display AT command results	
atq1	**Suppress AT command results**	
atv0	Display numeric result codes	
atv1	**Display textual result codes**	

atz	Load modem defaults	Same as atf.
at&c0	**Force data carrier detect (DCD) on at all times**	
at&c1	Data carrier detect (DCD) follows remote carrier / connection	DCD will turn on when an outbound or inbound connection is made
at&k0	**Disable RTS/CTS (RTR/CTS) flow control**	
at&k1	Enable RTS/CTS (RTR/CTS) flow control	
at&w	Save settings to flash	
atr	Enter RAW terminal mode which allows you to communicate directly with the WiFly module	
ats0=0	**Disable auto-answer**	
ats0=1	Enable auto-answer	
atd *address* atdt *address* atdp *address*	Connect to a remote host. Format is hostname:port	Defaults to port 23. Examples: atd bbs.jammingsignal.com:23 atdp bbs.jammingsignal.com atdtbbs.jammingsignal.com
atd# atdt# atdp#	Connect to a remote host from the phone book.	Example: atd#1 atdp#4 atdt#9

Appendix A: Setup after Factory Reset

After a factory reset, the following settings are recommended for normal use.

set wlan join 1	*(Auto- associate with access point)*
set ip dhcp 1	*(Turn on DHCP client)*
set uart baud 2400	*(Default baud rate)*
set uart flow 1	*(Turn on hardware flow control, needed for UP9600 mode)*
set ip tcp-mode 0x10	*(Turn off remote configuration via $$$)*

Plus the SSID and key/passphrase need to be set. This can be done through the Arduino interface, or manually as follows:

set wlan key <key>	(for WEP)
-or-	
set wlan passphrase <phrase>	(for WPA)
set wlan ssid <ssid>	*(Set the SSID)*
set comm time 20	(Default is 5. Setting this to 20 fixes 1200 baud issues)

Then **save** and **reboot** to make the changes permanent.