1. **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 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 330 Ohm resistors, or a resistor network DIP such as the Bourns Inc. 4116R-1-331LF (provided) can be used.

A separate VIN connector is provided for powering the board for standalone use, this is not 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.

1. **Configurations**
   1. **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.  
     
   1. **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”.   
   1. **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!

* 1. **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 I²C) 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.

1. **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 |

1. **Schematic**

# Software Support

* 1. **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.

* 1. **CommodoreServer Support**
  2. **QuantumLink RELOADED**

1. 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 )*

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)*

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