

# FlexSpec1 Manual Arduino Control with PuTTY and Xming on Windows

Wayne Green

Sunday 5<sup>th</sup> November, 2023 12:04:19 MT

This document covers using a Win1X machine to control a Raspberry Pi over the Internet.

As of October 2023 we are leaning to using the StellarMate<sup>1</sup> product together with PuTTY<sup>2,3</sup>. These notes were developed for for our roll-your-own interface.

Stellarmate package has many support features already installed. Jasem Mutlaq, the author, is heavily involved with Kstars, libindi and Ekos – so the integration is very strong. We have successfully used the package in both stand-alone mode on the Raspberry Pi (RPi), with X-Windows Xming<sup>4</sup> support, in its 'split' mode – where a basic interface is provided between a Win1X system and a remote libinbdi-server running on the RPi. As of October 2023 we tested between Boulder CO, Phoenix AZ, Portland OR and Newcastle upon Tyne, UK.

---

<sup>1</sup><https://stellarmate.com/>

<sup>2</sup><https://www.putty.org/>

<sup>3</sup>`sudo apt install putty -y`

<sup>4</sup><https://sourceforge.net/projects/xming/>

| Service          | Port         | WAN/Internet Control        |
|------------------|--------------|-----------------------------|
| SSH              | 5624         | Required for Flexspec       |
| INDI Web Manager | 8624         | Not required                |
| INDI Server      | 7624         | Not required                |
| EkosLive Server  | 3000         | Not required                |
| Web VNC          | 6080         | Not required                |
| VNC              | 5900         | Not required                |
| Serial           | /dev/ttyS<n> | Arduino Serial1 pins on RPi |
| Serial           | /dev/ttyACM0 | Arduino Serial USB          |

Table 1: Stellarmate Wan Firewall Requirements

## Contents

|                   |  |           |
|-------------------|--|-----------|
| 1.1               | Remote Desktop PuTTY                               | 3         |
| 1.2               | Remote Desktop Xming                               | 4         |
| <b>2</b>          | <b>SSH on the Raspberry Pi</b>                     | <b>5</b>  |
| 2.1               | Non-StellarMate Raspberry Pi's                     | 5         |
| 2.2               | Raspberry Pi – Raspbian SSH                        | 6         |
| 2.3               | StellarMate  | 6         |
| <b>3</b>          | <b>Using Putty from Linux Desktop</b>              | <b>6</b>  |
| <b>4</b>          | <b>Linux Ports</b>                                 | <b>6</b>  |
| <b>5</b>          | <b>Running the Arduino CLI on the Raspberry Pi</b> | <b>7</b>  |
| 5.1               | Additional Resources                               | 7         |
| 5.2               | More than One Win1X connection                     | 7         |
| 5.2.1             | Another approach to autostart with Win1X           | 7         |
| <b>6</b>          | <b>Auditing</b>                                    | <b>7</b>  |
| 6.1               | Windows  | 8         |
| 6.2               | Linux  | 8         |
| <b>7</b>          | <b>WAN Firewall</b>                                | <b>9</b>  |
| 7.1               | Handy Network Commands                             | 11        |
| 7.2               | ip vs ifconfig                                     | 12        |
| 7.3               | Modem Configuration                                | 12        |
| <b>8</b>          | <b>RPi Serial Ports</b>                            | <b>12</b> |
| 8.1               | Connecting FlexSpec1 via Serial Ports              | 13        |
| <b>9</b>          | <b>Home Security</b>                               | <b>13</b> |
| <b>References</b> |  | <b>13</b> |

## List of Figures

|     |  |   |
|-----|--|---|
| 1-1 | Win1X use.   | 2 |
| 1-2 | PuTTY screen snaps of the 4 configuration menu panels. | 4 |
| 1-3 | Xming screen snaps of the 4 configuration menu panels. | 5 |

## PuTTY and Xming on Windows

PuTTY running remotely on a Raspberry Pi uses X11 to share its interface with the remote user. This may require Xming for Win11, and XQuartz for Apple machines. It may require including a X11 desktop (in lieu of Wayland) on the remote Linux machines.

This article addresses executing PuTTY on a Raspberry Pi from a Win1X machine located elsewhere on the Internet. It addresses basic tactical aspects of network configuration. Direct X11 and port-to-port TCP methods sidestep the bandwidth overhead of VNC like environments. The office desktop is presumed to be a Win1X office desktop, but may be a Linux desktop with built-in native X11 protocols<sup>56</sup>, or an Apple machine with a proper X11 Server support package like XQuartz. Linux is addressed in section 3.

Win1X requires an X11 Server support program. Here the Xming<sup>7</sup> package is the only one discussed.

The typographic conventions used in these notes.

Here **k** means hit the “k” on the keyboard.

For the Win1X office desktop, it is important that you acquire and install:

1. Win1X office desktop
  - (a) Xming
  - (b) PuTTY
2. Raspberry Pi
  - (a) PuTTY
3. Optional Linux office desktop
  - (a) PuTTY

You need to configure Xming’s XLaunch program. You need to configure the Raspberry Pi’s configuration.

The PuTTY program provides a X11/ssh connection to remote machines. Here the machine’s hostname is **stellarmate.local**, the port is 22.

Xming is a X11 “server” (meaning is backwards from a data center’s client/server sense). X11 “serves” the graphics from a remote machine doing all the heavy computing on a light-weight machine that knows nothing about the remote application.

Grab Xming and install on a Win machine.

First some X11 magic:

You will open a **ssh -X** connection to the Raspberry Pi. This allows linux to display a console window on the Win1X office desktop monitor. In that window (remember will start programs on the Raspberry Pi) you will start a PuTTY program running on the Raspberry Pi that causes its window to appear on the Win1X office desktop.

Check the files in the /etc/X11 directory:  
/etc/X11/xinit/xserverrc

---

<sup>5</sup>In X11, the XClient is the program with all the builtin domain/control logic – like a spreadsheet, that offers up a GUI connection. An XServer, with all the logic for window creation, managing user interaction and ‘serving’ back graphics etc ties the user to the XClient’s logic. The XServer may be elsewhere on the network from the running program. The protocol is very light-weight compared to remote-desktop (VNC) methods.

<sup>6</sup>Note: The Wayland X11-Replacement may introduce X11 issues.

<sup>7</sup>Get Xming: <https://sourceforge.net/projects/xming/>

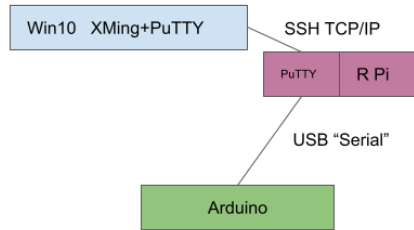


Figure 1-1: Win1X office desktop monitor uses TCP/IP and SSH with the -X switch to open a connection to Raspberry Pi. A PuTTY instance is started on the Raspberry Pi, configured for the USB port for the Arduino displays the PuTTY window on the remote desktop vi SSH -X working with Xming

```

36  ~/.xinitrc
37  ~/.xsession  -- start programs when the login is complete
38  ~/.profile
39
40  ssh-keygen -t ed25519 -C "your_email@example.com"
41
42  If there is no /etc/X11
43  xmodmap -pke > ~/.Xmodmap # make a .Xmodmap upon which to hack
44  This varies with new modern keyboards, there is massive confusion over what a key is called and what code
45  corresponds to the key among keyboard vendors.
46  To use the caps lock key as a spare control key find the code for the caps lock key using the xev program.
47  Xev will show keycodes when a key changes state, like shift down, then shift up etc...
48  Hit the the "caps" key and note its code. Then edit the ~/.Xmodmap file and change:
49  keycode 37 = Control_L NoSymbol Control_L

```

## 1.1 Remote Desktop PuTTY

On the remote desktop, configure the remote PuTTY:

### 1. Session:

- (a) Select the **Serial** radio button
- (b) Serial Line to **/dev/ttyACM0**
- (c) Speed **9600**
- (d) Close window on exit → **Only on clean exit**

### 2. Terminal

- (a) In the **Line discipline options**:
  - i. Local echo to **Force on**
  - ii. Local line editing to **Force on**

### 3. Window

- (a) Columns: **120** Rows: **50**
- (b) scrollbar 200
- (c) display scrollbar

### 4. Connection → Data

- (a) login name can be set to **stellarmate** for stellarmate, or the remote user's name.
- (b) Terminal-type string xterm
- (c) Terminal Speed 38400, 38400

### 5. SSH

- (a) Remote Command to **/home/stellarmate/bin/flex.py**

### 6. SSH→X11

- (a) Enable X11 forwarding
- (b) localhost:0.0
- (c) MIT-Magic-Cookie-1
- (a) Serial
  - i. Baud to 9600
  - ii. Data bits 8
  - iii. Stop bits 1
  - iv. Parity None
  - v. Flow Control None.

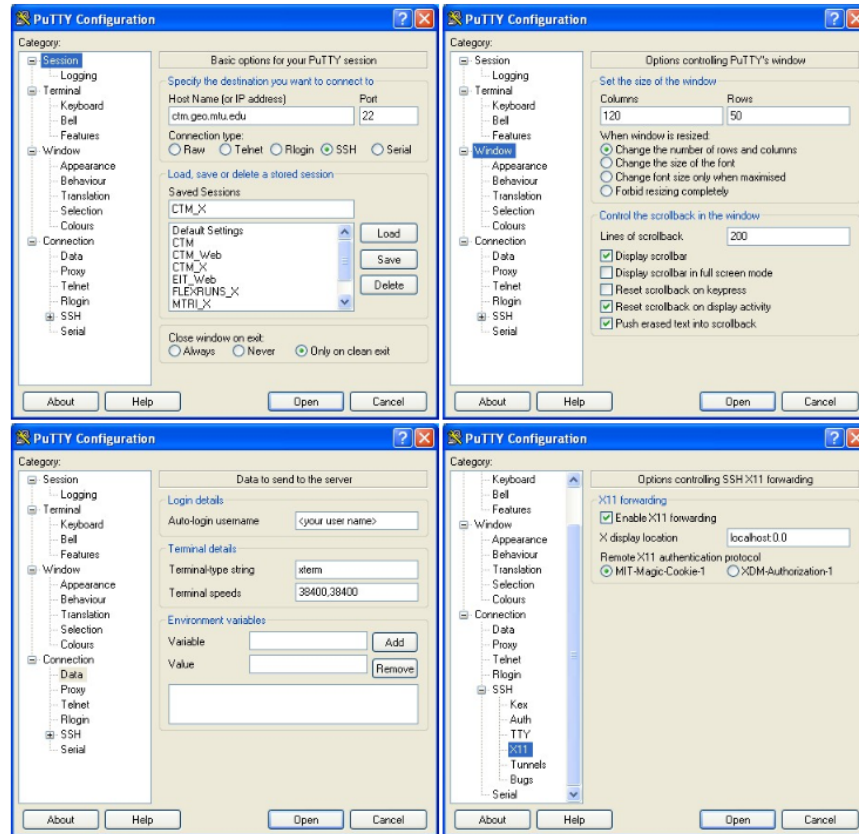


Figure 1-2: PuTTY screen snaps of the 4 configuration menu panels.

## 1.2 Remote Desktop Xming

On the Windows pane, enter a new configuration for this machine.

1. Page 1 - Set Multiple windows (Each remote application gets its own Win1X office desktop window).
2. Page 2 - Start no client. The clients are started with the Win1X PuTTY's SSH -X session.
3. Page 3 - Clipboard (handy)
4. Page 4 - Save the configuration. This sets the policy for all machines, Xming does not care about the remote machines business – any remote may connect if firewall permissions are set.

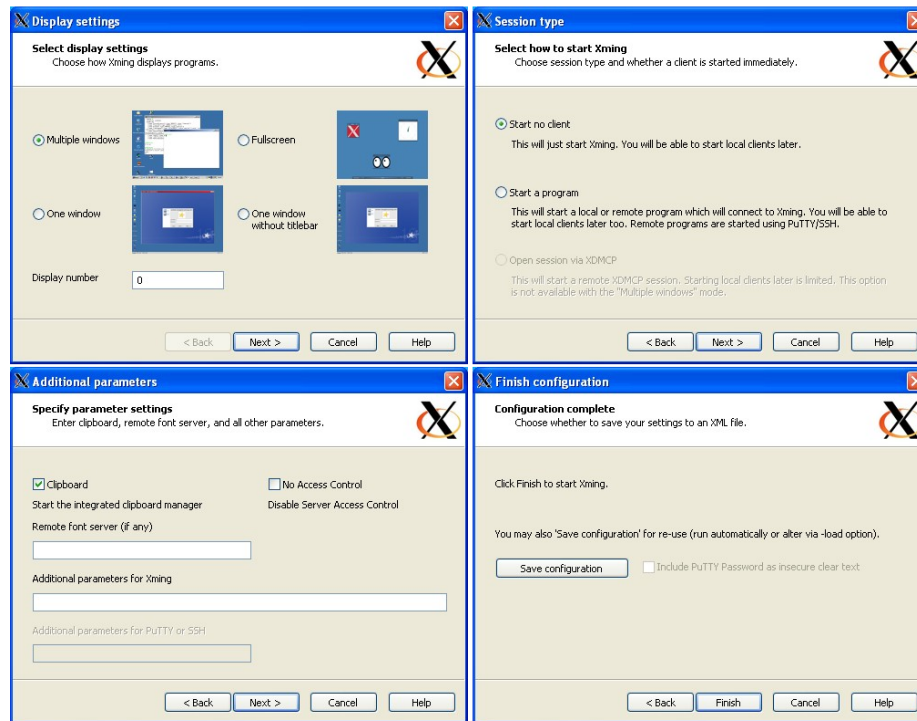


Figure 1-3: Xming screen snaps of the 4 configuration menu panels.

## 2 SSH on the Raspberry Pi

The StellarMate image has all necessary ports open and ready for use. The firewall has been set to pass all relevant ports.

See subsection 2.3 below.

### 2.1 Non-StellarMate Raspberry Pi's

Note: this section is complicated by the details needed to open home firewalls to the internet using the local Telco's interface routers. Each router is different, and the Telco changes its mind frequently.

This command will list the full name for hosts that are on the network. Here we're looking for our machine **pier15** that will appear as **pier15.hsd1.co.comcast.net**:

```
{sudo nmap -sP 10.1.10.0/24 | awk -e '/^Nmap/ {print $5;}}}
```

If ssh issues a:

```
ssh: connect to host <machine>.local port 22: Connection refused
```

use the keyboard/mouse for the raspbian machine and run the command **sudo raspi-config**, then under Interface options, **I2 SSH** **TAB**, enable.

SSH (Secure SHell), originates on port 22. The port needs to be allowed through the Raspbian's "uncomplicated firewall" **ufw** task. Root permissions are required.

```
sudo ufw list      # see all the current ports
sudo ufw allow ssh # allow the port to work.
```



## 2.2 Raspberry Pi – Raspbian SSH

Enable SSH through the Raspberry Pi Configuration menu: **Preferences** → **Raspberry Pi Configuration** Click on “Interfaces”: and select **Enabled** next to SSH.

Use the command `sudo nano /etc/ssh/sshd_config` find and `X11Forwarding` set the value to yes

`X11Forwarding yes.`

Save.

`sudo systemctl restart ssh`

## 2.3 StellarMate

You need to `sudo apt update`, then `sudo apt install putty -y` to install PuTTY.

Add the `x11-utils` package to pick up the rather silly command keys. This displays two white eyeballs and the pupils track the mouse movement. While small and cute, it does a rather deep test of the X11 capability including managing the event manager. This is helpful to debug X11 connections.

`sudo apt install -y x11-utils`

Other programs:

`sudo apt install -y saoimage-ds9`

`sudo apt install -y iraf`

`sudo apt install -y python3-pyraf`

The ssh connect to StellarMate is made to an alternate port:

`ssh -X -p 5624 stellarmate@stellarmate.local`

The username is `stellarmate` and the password is the usual StellarMate password.

## 3 Using Putty from Linux Desktop

PuTTY on a linux desktop machine in the office has all the Xming features built-in and does not require additional X-windows code.

## 4 Linux Ports

devices such as serial ports, USB ports etc appear simply as filenames. For example the Arduino often identifies as `/dev/ttyACM0`. By making no distinction using devices becomes very easy but drives Win1X people nuts.

See section 8 for details about the Raspberry Pi.

The ports you are interested in:

1. `/dev/serial0` - the serial port pins on the Raspberry Pi. This requires special action to enable the GPIO pins. This is recognized by the Arduino on its Serial1 device.
2. `/dev/ttyAMC0` - the USB port recognized by the Arduino on its Serial device.

## 5 Running the Arduino CLI on the Raspberry Pi

The Raspberry Pi will run the Arduino CLI, which in turn may run a serial port connection. Using the magic of X11, discussed above, the Arduino CLI windows will appear on the Win1X office desktop monitor or Linux office desktop monitor.

From the Win1X machine:

1. Use XLaunch with multiple windows.
  2. Then make a PuTTY connection to the RPi. (Call this Win1X/Rpi)
    - (a) this opens a Win1X cmd window, and prompts for the password into the RPi machine.
    - (b) A Rpi prompt will appear.
  3. In the Win1X/Rpi window (now a shell on the RPi)
    - (a) enter the command `putty --display=$DISPLAY` (two dashes)
    - (b) This causes a PuTTY configuration window from the RPi to appear on the Win1X desktop.
    - (c) Configure for the serial connection desired `/dev/ttyAM0` for the Arduino.
- (a) Run
  - (b) This causes a PuTTY serial console (as you configured it) to appear on the Win1X desktop.
  - (c) Interact with the Arduino, at the end of a UART serial connection, with the FlexSpec1 instrument – as though you were at the Rpi.

You may wish to install `xeyes` program on the RPi. Xeyes is a rather simplistic X11 program that causes a little window to pop-up with two “eyeballs” where the “eyes” will watch the mouse move around. This tests mouse and graphics in on go. To do so – in the Win1X/Rpi console enter:

```
sudo apt install x11-utils
```

### 5.1 Additional Resources

[http://www.straightrunning.com/xmingnotes/IDH\\_PROGRAM.htm](http://www.straightrunning.com/xmingnotes/IDH_PROGRAM.htm)

### 5.2 More than One Win1X connection

Make Desktop Icon's for unique connections:

1. Run XLaunch.exe and save the configuration to file `config.xlaunch`.
2. Create a shortcut of XLaunch.exe under Startup directory.
3. Modify the target field of the shortcut to `"C:\ZZH\software\Xming\XLaunch.exe" -run "config.xlaunch"`.

#### 5.2.1 Another approach to autostart with Win1X

1. Search for Xming in the Start Menu
2. Right click on the shortcut and select Open file location
3. Copy the selected Xming shortcut
4. Press Ctrl-I to move the cursor to the Explorer location input
5. type `shell:startup` and then Enter
6. Paste the Xming shortcut in the Startup directory.

## 6 Auditing

It is possible to save the and/or share the configuration parameters for PuTTY.

## 6.1 Windows

PuTTY on Win1X records its configuration(s) into the Registry. It requires the dangerous step of using Regedit to recover. The steps are:

1. Find HKEY\_CURRENT\_USER\Software\SimonTatham\PuTTY Note: The “SimonTatham” is the name of the author of the package not you, the “CURRENT\_USER”.
2. Right click and export, the resulting file is a text file suitable for sharing and archiving.

## 6.2 Linux

The configuration file is in `$HOME/.putty` deep.

## 7 WAN Firewall

Never open a large range of ports. Stellarmate uses ports 5624-8624, that includes (Internet Relay Chat RFC-1469) at 6667.

DRAFT

| Service          | Port         | WAN/Internet Control        |
|------------------|--------------|-----------------------------|
| SSH              | 5624         | Required for Flexspec       |
| INDI Web Manager | 8624         | Not required                |
| INDI Server      | 7624         | Not required                |
| EkosLive Server  | 3000         | Not required                |
| Web VNC          | 6080         | Not required                |
| VNC              | 5900         | Not required                |
| Serial           | /dev/ttyS<n> | Arduino Serial1 pins on RPi |
| Serial           | /dev/ttyACM0 | Arduino Serial USB          |

Table 2: Stellarmate Wan Firewall Requirements

## 7.1 Handy Network Commands

The original program to manage and display information about networks was called `ifconfig` for “interface configuration”. This has been replaced recently with the newer `ip` to “show / manipulate routing, network devices, interfaces and tunnels”.

| command  | action   |
|----------|--|
| arp      | address resolution protocol  |
| dig      | DNS lookup   |
| netstat  | Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships |
| nslookup | lookup DNS information   |
| ping     | see if path to machine exists  |
| tcpdump  | view traffic on local machine  |
| tracert  | tracert to a remote machine: eg: <code>tracert google.com</code>   |

Table 3: A few of many tools for network tracing.

## 7.2 ip vs ifconfig

The mapping from old ifconfig commands to new ip commands.

| ifconfig command                    | New ip command                                       |
|-------------------------------------|--|
| arp -a                              | ip neigh   |
| arp -v                              | ip -s neigh  |
| arp -s 192.168.1.1 1:2:3:4:5:6      | ip neigh add 192.168.1.1 lladdr 1:2:3:4:5:6 dev eth1 |
| arp -i eth1 -d 192.168.1.1          | ip neigh del 192.168.1.1 dev eth1                    |
| ifconfig -a                         | ip addr  |
| ifconfig eth0 down                  | ip link set eth0 down                                |
| ifconfig eth0 up                    | ip link set eth0 up                                  |
| ifconfig eth0 192.168.1.1           | ip addr add 192.168.1.1/24 dev eth0                  |
| ifconfig eth0 netmask 255.255.255.0 | ip addr add 192.168.1.1/24 dev eth0                  |
| ifconfig eth0 mtu 9000              | ip link set eth0 mtu 9000                            |
| ifconfig eth0:0 192.168.1.2         | ip addr add 192.168.1.2/24 dev eth0                  |
| iptables                            | show and manipulate IPTABLES <b>iptables -S</b>      |
| netstat                             | ss   |
| netstat -neopa                      | ss -neopa  |
| netstat -g                          | ip maddr   |
| route                               | ip route   |

Table 4: A ifconfig vs ip command quick summary.

## 7.3 Modem Configuration

Each model will use a different scheme to enable Network Address Translation (NAT) to connect the port from the WAN side to a “server” within the LAN. Information usually includes:

Start Port

End Port

Server IP4 Address

Server IP6 Address

## 8 RPi Serial Ports

The Raspberry Pi has 6 serial ports, available in various ways, including over BlueTooth connections. The ports use GPIO pins, and may be disabled to allow secondary functions for the GPIO pins to be permitted.

| Name  | Type      | Models   | Enabled | GPIO (TX) | GPIO (RX) | CTS/RTS |
|-------|-----------|----------|---------|-----------|-----------|---------|
| UART0 | PL011     | All      | Yes     | 14,32,36  | 15,33,37  |         |
| UART1 | mini-UART | All      | No      | 14,32,40  | 15,33,41  |         |
| UART2 | PL011     | Pi4 Only | No      | 0         | 3         | 2,3     |
| UART3 | PL011     | Pi4 Only | No      | 4         | 7         | 6,7     |
| UART4 | PL011     | Pi4 Only | No      | 8         | 11        | 10,11   |
| UART5 | PL011     | Pi4 Only | No      | 12        | 15        | 14,15   |

Table 5: Stellarmate WAN Firewall Ports

## 8.1 Connecting FlexSpec1 via Serial Ports

The Serial Ports for the RPi are 3.3V and require a level shifter to be on the safeside. The FlexSpec1 design calls for a 5-wire cable between the RPi and the Arduino SBC. These include:

| RPi   | RPi                      |
|-------|--------------------------|
| +5 V  | Power                    |
| GND   | Signal/Electrical Ground |
| Tx    | Commands                 |
| Rx    | Response                 |
| Reset | Reboot FlexSpec1         |

Table 6: The Cable has a footprint on the FlexSpec PCBA. Pins are shown for programming clarity. The pins on the Arduino are lables and NOT traditional pin numbers. Be careful.

## 9 Home Security

Adding MAC address filters to your ISP modem is a decent and easy way to add security for to keep unauthorized people out of the observing sessions!

**ACTION:**  
**MAC Security<sup>1</sup>**