FlexSpec1 Manual Arduino Control with PuTTY and Xming on Windows

Wayne Green

Saturday 20th May, 2023 12:39:42 MT

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

As of 2023-05-15 testing with the Stellarmate package (Jasem Mutlaq) has many support features already installed. Jasem Mutlaq, the author, is heavily involved with Kstars and Ekos – so the integration is very strong.

Contents

List of Figures

PuTTY and Xming on Windows

This article addresses executing PuTTY on a Raspberry Pi from a Win1X machine located elsewhere on the Internet. This sidesteps the bandwidth overhead of VNC like environments. The office desktop is presumed to be a Win1X office desktop, but may be a Linux desktop honoring X11 protocols[1], 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[2] package is the only one discussed.

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.

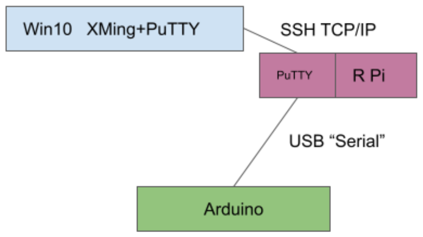


Figure 1-1: Win1X office desktop monitor uses TCP/IP and SSH with the -X switch to open a connection to Raspberry Pi. A PuTTY instance is started on the Raspberry Pi, configured for the USB port for the Arduino displays the PuTTY window on the remote desktop vi SSH -X working with Xming

[1]In X11, a 'program' is a 'Server' 'Serving' packets to an X11-client. The X11-client makes one or more windows/dialogs/etc appear at a remote location. The X-Server supplies the rules for making the window and interacts with the user sending and obtaining information from the remote user. The protocol is very light-weight compared to remote-desktop methods.

[2]Get Xming: https://sourceforge.net/projects/xming/

```
Check the files in the /etc/X11 directory:
   /etc/X11/xinit/xserverrc
   ~/.xinitrc
   ~/.xsession  -- start programs when the login is complete
   ~/.profile

ssh-keygen -t ed25519 -C "your_email@example.com"
```

If there is no /etc/X11

```
xmodmap -pke > ~/.Xmodmap # make a .Xmodmap upon which to hack
```

This varies with new modern keyboards, there is massive confusion over what a key is called and what code corresponds to the key among keyboard vendors.

To use the caps lock key as a spare control key find the code for the caps lock key using the xev program. Xev will show keycodes when a key changes state, like shift down, then shift up etc...

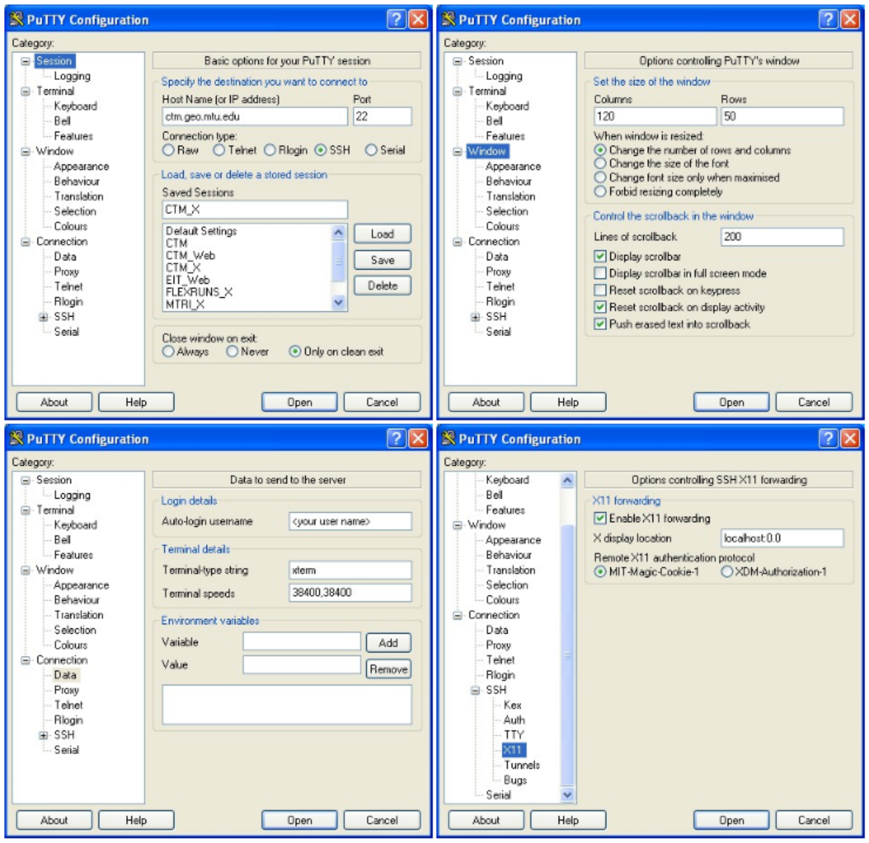
Hit the the “caps” key and note its code. Then edit the /.Xmodmap file and change:

```
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 disipline options:
        i. Local echo to Force on
        ii. Local line editing to Force on
3. Window
    (a) Columns: 120 Rows: 50
    (b) scrollback 200
    (c) display scrollbar
4. Connection → Data
    (a) login name can be set to smate for stellarmate, or the remote user's name.
    (b) Terminal-type string xterm
    (c) Terminal Speed 38400, 38400
5. 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.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): ctm.geo.mtu.edu Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

Saved Sessions: CTM_X

Default Settings
CTM
CTM_Web
CTM_X
EIT_Web
FLEXRUNS_X
MTRI_X

Close window on exit: Always Never [✓ Only on clean exit]

Options controlling PuTTY's window

Set the size of the window

Columns: 120 Rows: 50

When window is resized:
[✓ Change the number of rows and columns]
Change the size of the font
Change font size only when maximised
Forbid resizing completely

Control the scrollback in the window

Lines of scrollback: 200

[✓ Display scrollbar]
Display scrollbar in full screen mode
Reset scrollback on keypress
[✓ Reset scrollback on display activity]
[✓ Push erased text into scrollback]

Data to send to the server

Login details

Auto-login username: <your user name>

Terminal details

Terminal-type string: xterm

Terminal speeds: 38400,38400

Environment variables

Variable:

Value:

Options controlling SSH X11 forwarding

X11 forwarding

[✓ Enable X11 forwarding]

X display location: localhost:0.0

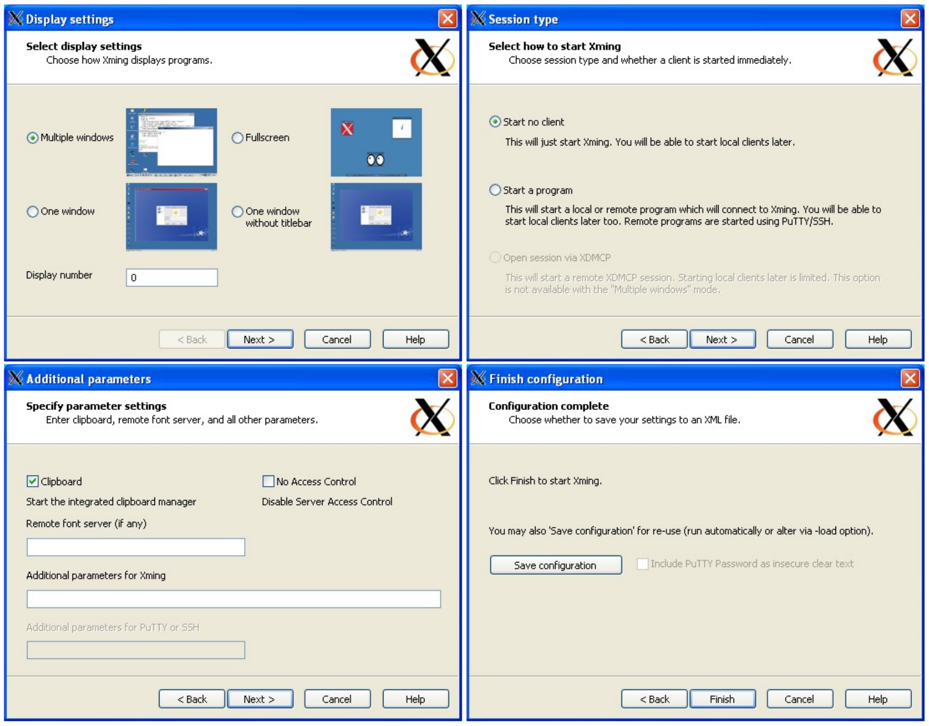
Remote X11 authentication protocol: [✓ MIT-Magic-Cookie-1] XDM-Authorization-1

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.



Display settings

Select display settings
Choose how Xming displays programs.

[✓ Multiple windows] Fullscreen
One window | One window without titlebar

Display number: 0

Session type

Select how to start Xming
Choose session type and whether a client is started immediately.

[✓ Start no client]
This will just start Xming. You will be able to start local clients later.

Start a program
This will start a local or remote program which will connect to Xming. You will be able to start local clients later too. Remote programs are started using PuTTY/SSH.

Open session via XDMCP
This will start a remote XDMCP session. Starting local clients later is limited. This option is not available with the "Multiple windows" mode.

Additional parameters

Specify parameter settings
Enter clipboard, remote font server, and all other parameters.

[✓ Clipboard]
Start the integrated clipboard manager

No Access Control
Disable Server Access Control

Remote font server (if any)

Additional parameters for Xming

Additional parameters for PuTTY or SSH

Finish configuration

Configuration complete
Choose whether to save your settings to an XML file.

Click Finish to start Xming.

You may also 'Save configuration' for re-use (run automatically or alter via -load option).

Include PuTTY Password as insecure clear text

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

2 SSH on the Raspberry Pi

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.
```

2.1 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
```

2.2 StellarMate

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

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

Linux treats devices like serial ports, USB ports etc as special files. Making no distinction makes things very easy but drives Win1X people nuts.

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

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-l 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.