Booting the display to a custom application

Submitted by Joshua Caldwell on Fri, 2015-11-20 21:58

By default, the displays will boot up to a Maximatecc default GUI application. This application can be useful in finding the IP address of your display and changing specific display settings, among other things. There will come a time in your development where you will want to instead have the display boot up to your specific application developed in Qt or C++ (Booting to a CoDeSys application is handled differently). This is easy to achieve by following the process outlined below.

If you are using a Qt4 based application:

- Log into the display either using an application such as Putty for Windows or using our development VM and the terminal window with the 'ssh' command
- Navigate to the '/opt/etc/init.d' directory in the display (once logged in 'cd /opt/etc/init.d')
- Type 'ls'. This will show you the files in this directory. You will see several script files which are used during the display boot process to start specific applications. The script file we are interested in is called 'StartupGui'. We want to modify this script to change it so it will start your custom application instead of the Maximatecc default GUI. To modify the script file type 'nano StartupGui' into the command line
- In the terminal window the StartupGui script file should have opened. The script may look a bit complicated but we have made it easy to change the boot application. We are looking for two lines near the top of the script, the first named 'APPLICATION_PATH' and the second named 'APPLICATION'. To modify the script to start your own application do the following:
  1. Before the words 'APPLICATION_PATH' and 'APPLICATION' put a pound sign ('#'). This will comment out these lines in the script file. The pound sign can be removed later if you ever want to boot back to our default GUI.
  2. Below the two lines above where we just inserted the pound sign type the following:
     APPLICATION_PATH = "/opt/NameOfYourApplication/"
     APPLICATION = "NameOfYourApplication"
     Note: The path and name of your application need to be typed exactly as they appear on the disk. Case Matters! To find them you should be able to navigate to the '/opt' directory of the display and see a folder with the same project name as you have created in Qt.
  3. That's it! Now we need to save the file by pressing CNTR+S. It will ask you if you to press 'y' if you want to save the file. And then you will need to confirm the file name. By default it will be StartupGui. We want to keep this name so press enter here. It should save it. To confirm this you can open it again with 'nano StartupGui' and confirm the changes were saved.
- Now that we have changed the StartupGui script to start our application instead of the default GUI we can restart our display and if done correctly it should now boot up to our custom application! You can restart the display either through the power button or by

typing 'restart' into the command line.

Note: You should have noted down or remember the IP address of the display. After this step the default GUI will no longer start up so you will not see the IP address of the display unless it is shown in your custom application. If the above steps were done incorrectly than the display will boot to a black screen. In this case we can still log into the display via the IP address and try again.

If you are using a Qt5 based application:

If using Qt5 the process is very similar to the process for Qt4 above with the following changes:

- The Qt4 StartupGui script is enabled by default on the displays while the Qt5 startup script is disabled. We will disable the Qt4 startup script by navigating to the '/opt/etc/init.d/' directory as described above. Then disable the Qt4 startup script by typing 'chmod -x StartupGui' into the display terminal. We also need to enable the Qt5 startup script by typing 'chmod +x StartupGuiQt5'.
- We have now enabled the Qt5 startup script and disabled the Qt4 script. Now we need to modify the Qt5 startup script to point to the application we want to start on boot. We can edit the Qt5 startup script file by typing 'nano StartupGuiQt5'. This will open the file for editing. We want to point the 'APPLICATION_PATH' and 'APPLICATION' to the location of the Qt5 application on the display as described in the section above for Qt4.

If you are using an iMX6 display (VS/VI2):

For the iMX6 displays VS and VI2 there is a slightly different startup file than the regular "StartupGuiQt5" script. Use the example script attached to this article and update it with your application PATH and NAME.

If you are using the XM2 display:

In the XM2 display there is a slightly different file structure than presented for the ARM displays above. In the XM2 display there is a single startup file which is used for both Qt4 and Qt5. The file is located in the '/opt/etc/init.d/' directory and is called 'StartupGuiQt5'. We need to make sure this file is enabled by navigating to the '/opt/etc/init.d' directory and typing 'chmod +x StartupGuiQt5' into the terminal. Once enabled we need to modify the file as described in the sections above to point to the Qt application located in the '/opt' directory of the device.

Once that is complete there is one additional script we need to disable which is responsible for launching the Maximatecc default settings application. Navigate to the '/opt/etc/rc3.d/' folder (type "cd /opt/etc/rc3.d") and disable the S50StartupLauncher file by typing 'chmod -x S50StartupLauncher'. Once complete your custom application should boot with the display. If at any time you would like to re-enable the default application you can type 'chmod +x

S50StartupLauncher'.

Attachment	Size
Startup script for iMX6 units [1]	913 bytes

Category:
ARM-Platform [2]
Linux [3]
X86-Platform [4]

Source URL: https://support.crosscontrol.com/kb/booting-display-custom-application

Links
[1] https://support.crosscontrol.com/sites/default/files/kb/startupguiqt5_imx6.txt
[2] https://support.crosscontrol.com/kb/arm-platform
[3] https://support.crosscontrol.com/kb/linux
[4] https://support.crosscontrol.com/kb/x86-platform