G2 Development

How do the G2 modules work?

The advanced display modules (G2) run embedded Linux, and (by default) have GUIs written in QML. We provide Qt Creator with custom graphic elements suitable for our display modules. These GUI elements include buttons, sliders, spinners, virtual keyboards. The customer can use the GUI element images we provide, or they may substitute their own image files. The customer may use Qt Creator on Windows, or in a Linux Virtual Machine that we provide. Using the VM, the customer may also write applications in C or C++. The Linux distribution running on the display module is our custom distribution, based on standard releases from the processor vendor (Freescale). Some of these modules can play videos (from files) on their on-board codec, but they are not designed to accept video input from an external source.

Windows or Linux Development

When choosing which development environment to use, it comes down to what language you will be using to write your application.

If you choose to write your whole application in QML, then you can use either Windows or the Linux VM. On the Windows side, we provide the Qt Creator IDE for development, and our G2Link application to transfer your code to the module. In the Linux VM, we also provide the Qt Creator IDE, but there is not a standalone application that transfers your code to the module. You would need to use SSH or Samba. Note: The G2 modules only support Samba 1.0. So, this comes down to personal preference on OS.

If you will be writing your application in C/C++, or a combination of QML/C/C++, then you will have to use the Linux VM. To write C/C++ applications that will run on the display module, the source must be compiled with a cross compiler which is only available in the Linux VM.

I have placed some links below that will provide more information...

Windows Development

G2H2 Windows Quick Start Guide

https://www.reachtech.com/products/touchscreen-display-modules/support/g2-documentation/windows-quick-start-guides/g2h2-windows-quick-start-guide/

Qt Creator

https://www.reachtech.com/products/touchscreen-display-modules/support/g2-documentation/qt-creator-integrated-development-environment/

G2Link

https://www.reachtech.com/products/touchscreen-display-modules/support/g2-documentation/g2link/

G2H v1.8 Software Documentation

https://www.reachtech.com/products/touchscreen-display-modules/support/g2-documentation/g2h-v1-8/

Linux Development

G2H2 Linux Quick Start Guide

https://www.reachtech.com/products/touchscreen-display-modules/support/g2-documentation/linux-quick-start-guides/g2h2/

Linux Virtual Machine Releases

https://www.reachtech.com/products/touchscreen-display-modules/support/g2-documentation/linux-virtual-machine-releases/

G2H v1.8 Software Documentation

https://www.reachtech.com/products/touchscreen-display-modules/support/g2-documentation/g2h-v1-8/

How Does Communication Work Between a Microcontroller and our Modules?

For your microcontroller to communicate with the QML Viewer (a tool for loading QML documents that makes it easy to quickly develop and debug QML applications) messages being sent need to be translated. We have created a set of tools called the Serial Input/Output Agent (SIO) and Translator Input/Output Agent (TIO) to make this microcontroller interface translation happen.

There is an "I/O Demo" in the standard Demo (pre-loaded on the Dev Kits) that shows how to use the serial port. The demo has two buttons that send ASCII strings when pressed, and there are also two text fields that accept ASCII strings to show temperature readings.

The SIO agent is the lowest-level layer that interacts with the serial port. The translate agent is the next layer, and can provide optional translation of the messages, based on runes in the file translate.txt. The QML viewer is the upper-most level, and it accepts messages from the translate agent, looks up the specified object, then property, and if found, sets the property to the given value (string, in the "I/O Demo").

Messages from your host controller would be in the form "object.property=value" by default, or you can use the translate agent to convert messages in the form "x=y" into the previous form. Outgoing messages can either conform to these formats if you need translation or can be free-form if you don't need translation.

For the "I/O Demo", when you press a button, the display module sends out a string such as "b=1" or "b=2" depending on which button is pressed. To set a text field, your host controller (or terminal emulator on the 'application port') would send "tc=45" or "tf=99" to set the Celsius or Fahrenheit fields. You could also send the translated version of the message with "celsius_input.text=45" or "fahrenheit_input.text=99".

You can download the demo(s) at the following link:

https://www.reachtech.com/products/touchscreen-display-modules/support/g2-documentation/demos-and-examples/

I will also list more links below that will help you get more familiar with our I/O Agents.

https://www.reachtech.com/faq-items/how-can-my-microcontroller-talk-to-your-user-interface-widgets-and-code/

https://www.reachtech.com/products/touchscreen-display-modules/support/g2-documentation/reference-materials/io-agent-development/

https://www.reachtech.com/products/touchscreen-display-modules/support/g2-documentation/reference-materials/qml-i-o-agents/

https://www.reachtech.com/products/touchscreen-display-modules/support/g2-documentation/reference-materials/serial-to-qml-communication/

https://www.reachtech.com/products/touchscreen-display-modules/support/g2-documentation/qml-viewer/

If you ever wanted to use a different communication method (CAN, Ethernet, etc.), you would need to replace the standard SIO Agent, with a CAN Agent, Ethernet Agent (EIO-Agent), or you would need to create a custom I/O Agent. We have a few sample applications on our website that demonstrate how to modify the SIO-Agent. You can find those here:

https://www.reachtech.com/products/touchscreen-display-modules/support/g2-documentation/application-notes/

I have also attached a document on the I/O Agents.

QML Components

Due to the variety of platforms that QML runs on, by default, there are very few QML components available. Reach Technology has provided several components to make development easier. You will need to include these in your project to make graphs, sliders, charts, etc.

https://www.reachtech.com/products/touchscreen-display-modules/support/g2-documentation/reference-materials/qml-components/