



Application Note 5530

Raspberry Pi Interface with Galil Controllers

An HMI or Human Machine Interface is needed in many motion control applications. These devices are used to allow a user to instruct a machine on what to do without the need of a full PC. Many times HMIs are touch enabled allowing them to be placed on the production floor and used by technicians right by the machine. There are many sources for HMIs and many can cost thousands of dollars. With the arrival of the Raspberry Pi Foundation and its Raspberry Pi models, a small but powerful option arrived. Adding to this, the official Raspberry Pi 7" touchscreen display, and the Galil application program interface this powerful trio form the foundation for an inexpensive but fully capable HMI that can be customized for any motion control application.

The following video shows the Raspberry Pi based HMI in action. You can see it connect to a controller, allow the user to control a linear stage, and run the cut to length application.

<http://www.galil.com/raspberrypi>

The strength of this new HMI option can be found in the software that can be developed to run on it. There are free software tools that can allow a person to create just about any kind of interface to run on the Raspberry Pi. The software acts as the glue to take in the input from the user in the touch interface and the Galil gclib API translates those simple instructions into the commands to send to the machine motion controller and PLCs. This lowers the bar of entry to something similar to developing a modern mobile phone app. The online community provides plenty of assistance to get started and allows the user to create interfaces to run on these fast and inexpensive computers. The parts for this project at the time of writing was under \$100.



Figure 1: Photo of Touchscreen and Motion Controller

This example Raspberry Pi based HMI is using the Raspberry Pi 3, the latest version of the Raspbian Linux operating system, Python for its programming language, Kivy for its user interface framework, and Galil gclib for communication with the motion control hardware. The Raspberry Pi 3 is an ARM based 1.2GHz quad core CPU board with 1GB RAM, built in Ethernet and WIFI. It runs a Linux distribution called Raspbian which is highly supported by the online community. Python is a popular programming language with large community support and Kivy is a Python based open source UI framework with built in touch-enabled elements.

The Galil gclib C API can be used on various platforms in many different programming languages. Its function is to communicate with the Galil controller at a higher level and dictate the overall application flow. Because gclib is written in low level C it can be used in other higher level languages such as Python. Galil has provided a Python wrapper and example code to make installing and using the API very easy.

There are two applications on this HMI project, the first being a homing and servo demo application and the second a simple cut-to-length application. The example applications here are provided as a demonstration and only showcase a small part of what is possible.

The first part of this example is the “Controller Selection” screen. The Galil gclib API has the capability to find controllers on a network. The first screen displays any controllers that are found as seen in Figure 2 and lets the user select the controller to use in the application. A controller must be chosen before any application can begin.

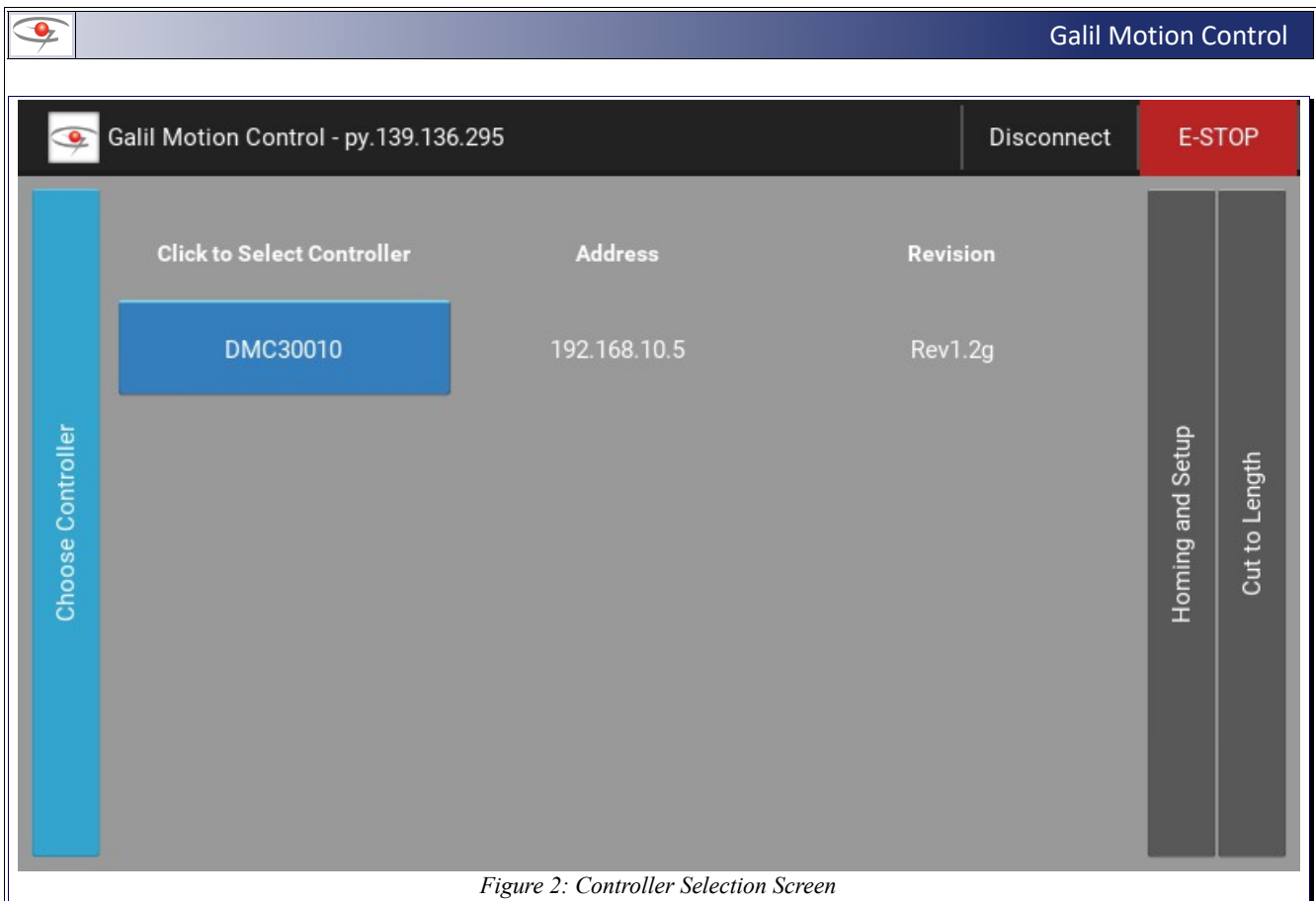


Figure 2: Controller Selection Screen

After the controller is selected, the user may need to check that the hardware is working correctly and home the equipment before beginning the application. The “Homing and Setup” screen displays information about the connected controller and allows the user to perform basic operations on the axis. Figure 3 shows the available operations such as turning the motor off, servoing, stopping, manually moving the motor on the stage, or perform a homing operation.

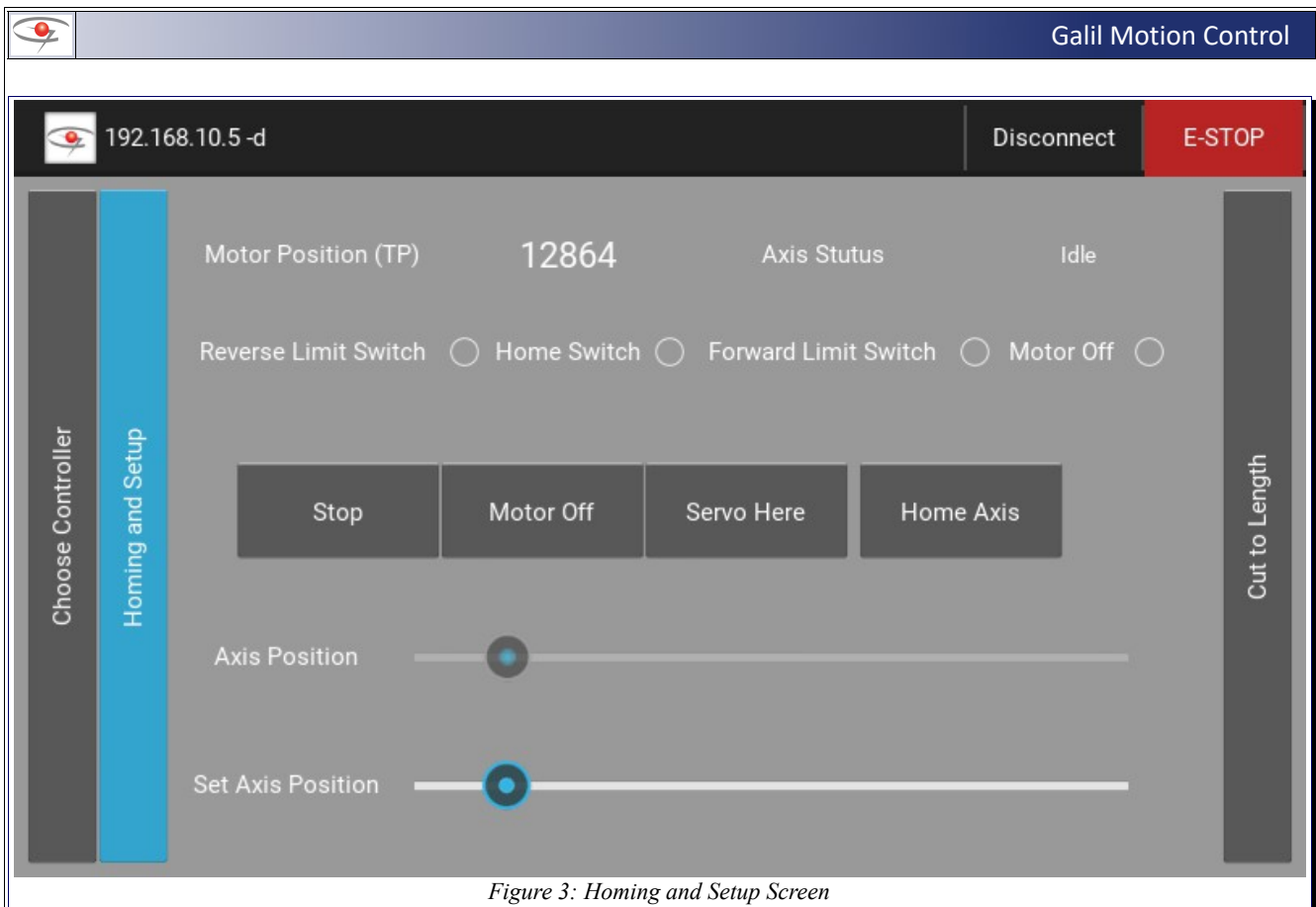


Figure 3: Homing and Setup Screen

The glib API is used to read the motor position and the state of the axis switches. The Python application displays this to the user. A simple program is downloaded to the controller to allow a slider to be used to set the axis position. When the user is ready the Home Axis button can be pressed to put the equipment in the right place to begin the cut to length application.

Most motion applications can be broken down into the following simple flow diagram.

Motion Mechanical System & I/O ↔ **Motion Controller & PLC (Optional)** ↔ **HMI**

In this application a technician enters values into an interface for the material to be processed. The technician enters the desired length of material and the number of pieces to cut. The Python application does the job to take in the commands from the technician and relay the information to the motion control system and/or PLC. Its main job is to handle the overall application and rely on the other systems to report success or failure of their components. The motion controller performs the needed motion elements to physically move the motors and cutting tools. A PLC such as the RIO-47000 series, may be involved to handle the flow of material to and through the machine including loading the bulk material and unloading the cut product.

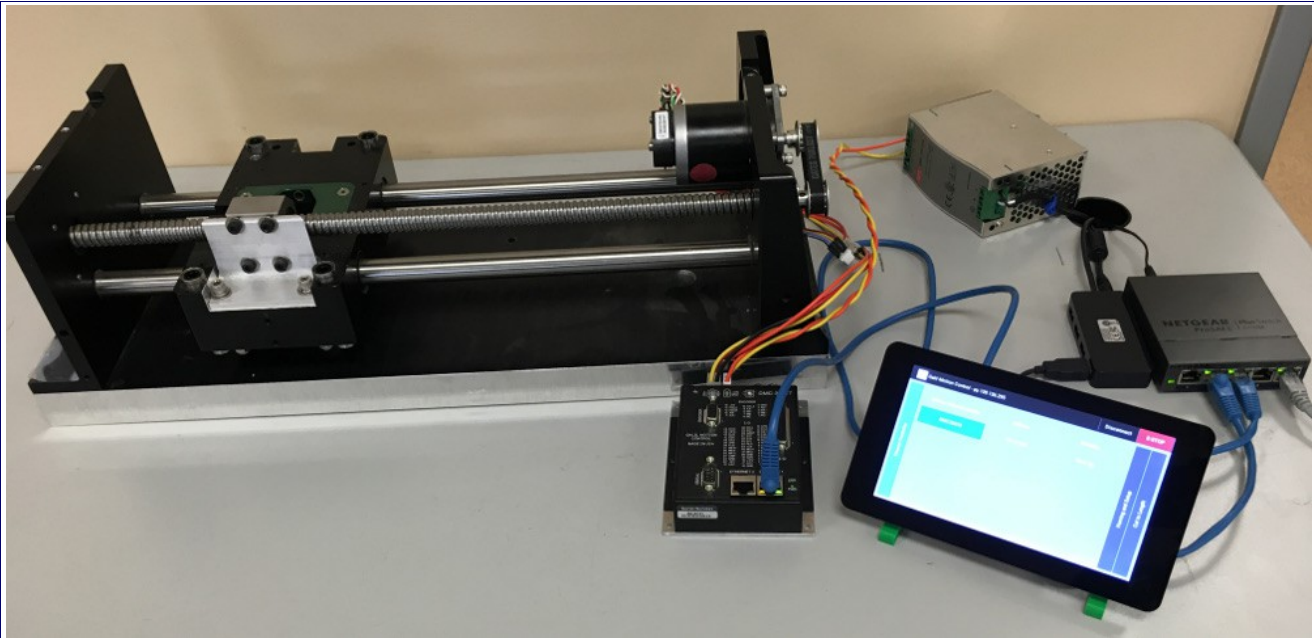


Figure 4: Touchscreen HMI Controlling Linear Stage

The “Cut To Length” screen seen in Figure 5 displays the user interface items needed to modify the cut to length application including entering the number of cuts, the desired length, and the begin button to start the process. The Python application downloads the needed code to the controller and requests information from the controller to display to the user. If a fault occurs the user can cancel the operation.

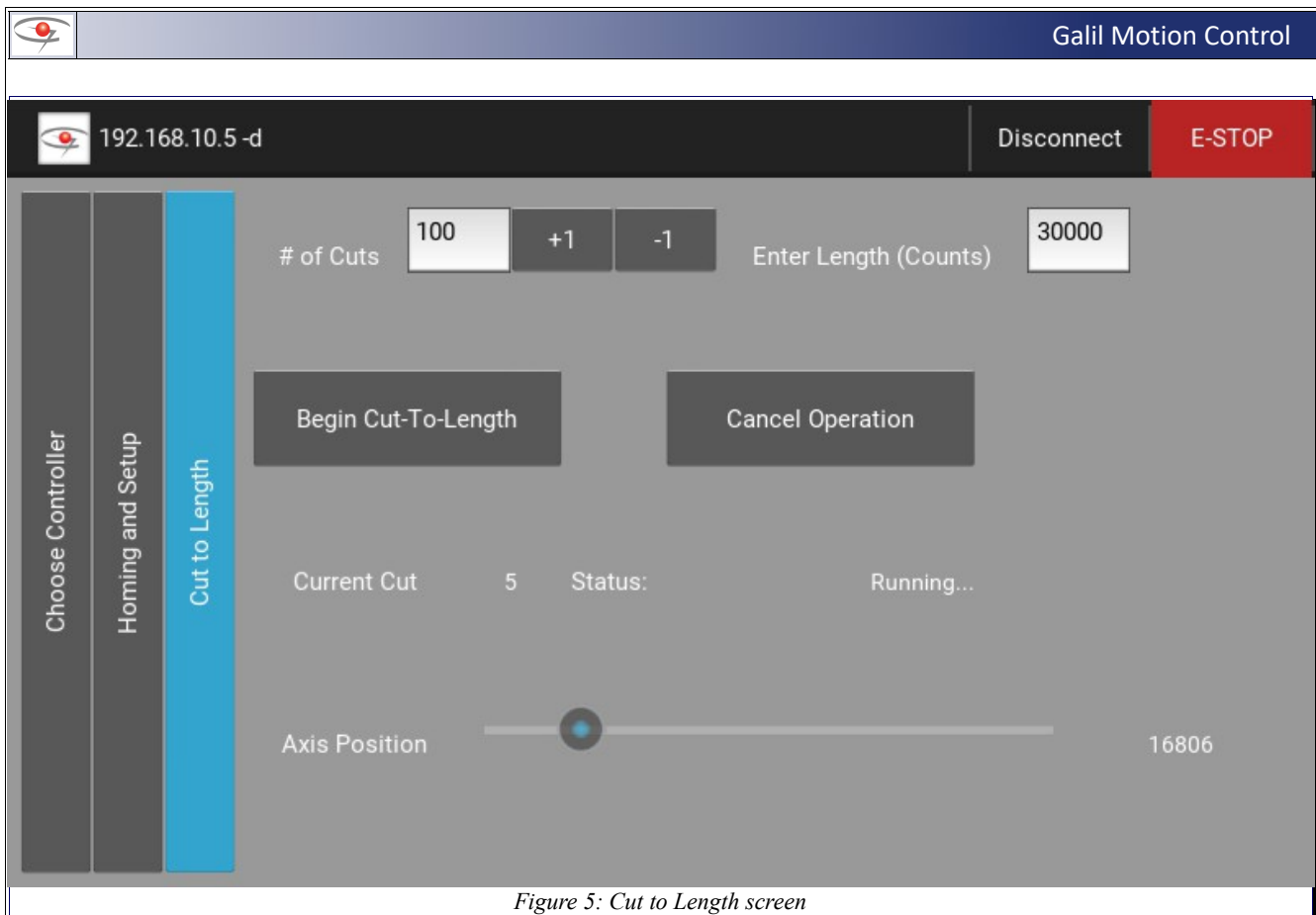


Figure 5: Cut to Length screen

The following video shows the Raspberry Pi based HMI in action. You can see it connect to a controller, allow the user to control a linear stage, and run the cut to length application.

This application is a powerful demonstration of the advancements in computer hardware that have driven down prices but increased performance. Advancements in software have provided easy to use tools with free and open source programming languages and frameworks. By combining these advancements with Galil Motion Control products, interfaces can be easily made for any motion control application. The Raspberry Pi with touch screen display used in this demonstration cost around \$100. All software used in this demonstration is free or open source.

Sources:

<https://www.raspberrypi.org/>

<https://www.raspberrypi.org/products/raspberry-pi-touch-display/>

<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

<https://www.python.org/>

<https://kivy.org/>