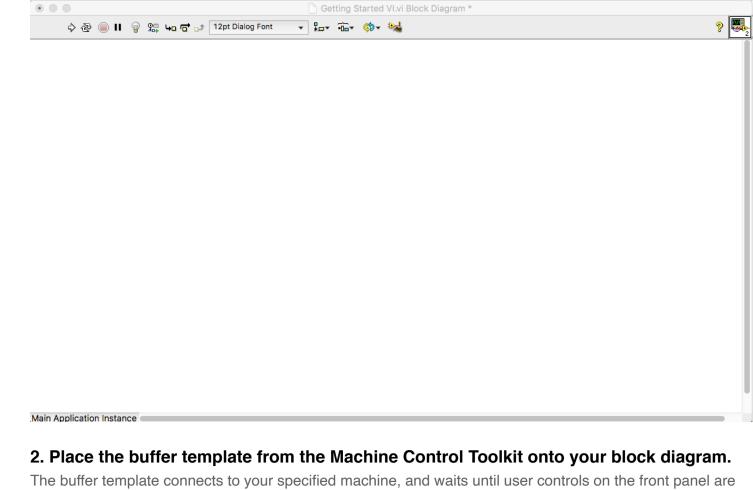
Getting Started

Get started with the Machine Control Toolkit by building a simple machine interface (step-by-step instructions below).

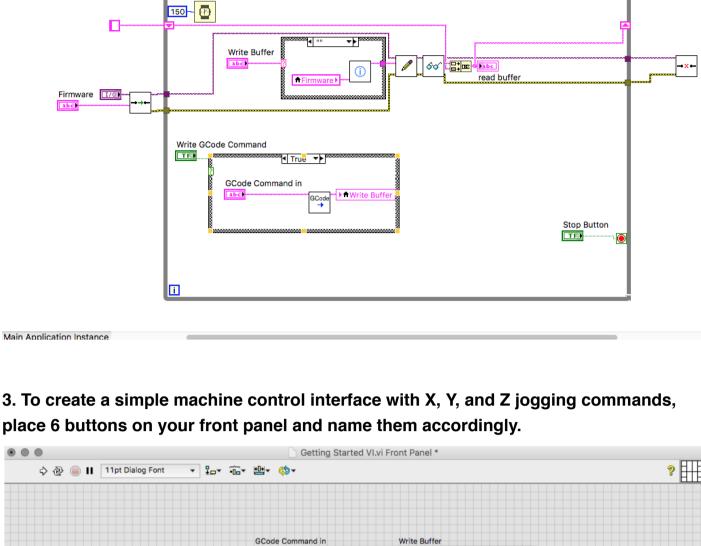
1. Open up a blank LabVIEW VI

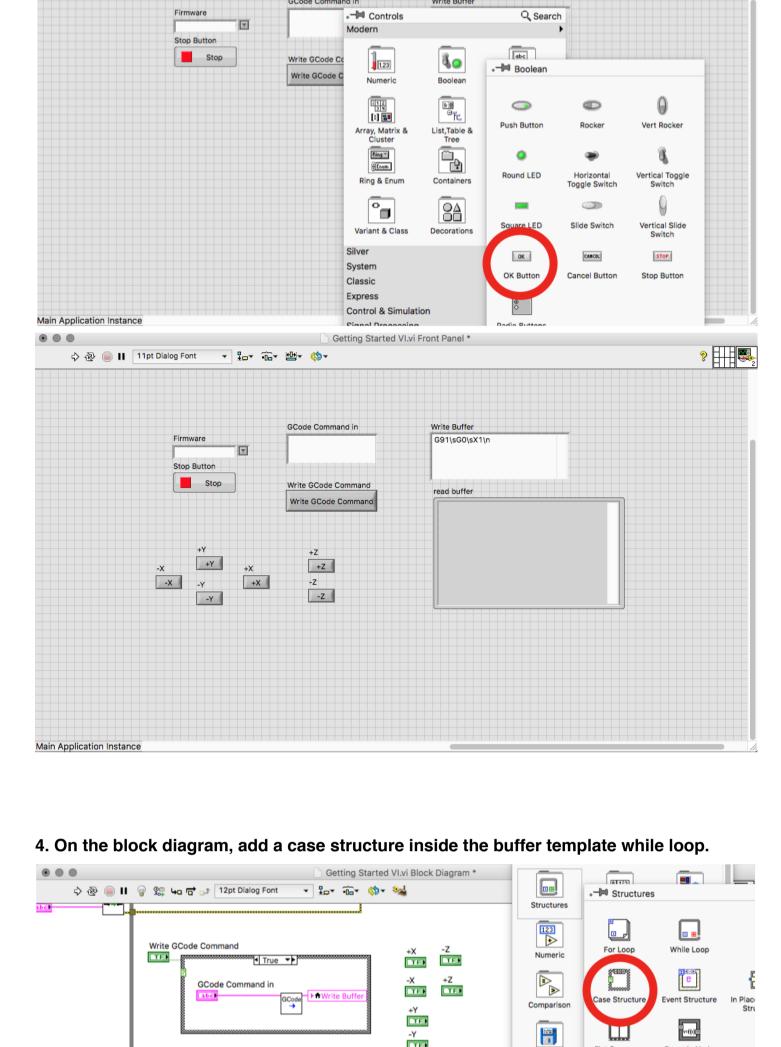


the template is a control to type in a G-Code command and have it written to the machine. Functions Q Search Programming Getting Started VI R 1 2 6 3 4 19 💠 🐼 🧓 Ⅱ 💡 😘 👆 🗗 🚅 12pt Dialog Font - Lor Tor 🕸 - 🤏

actuated, and then writes the appropriate command to your machine until you stop the program. Already in

Cluster, Class, & Variant Array obc a A Numeric Boolean String at m **O** .→ Machine Control Toolkit 60 Write to Machine .vi Disconnect from Machine.vi *** Ŷ **4** X Х⇔ Jog +X.vi Jog.vi Jog -X.vi Jog +Y.vi **₽** Z B Ž (married Jog -Y.vi Jog -Z.vi Set Units.vi Jog +Z.vi trol Toolkit **(i) ७**▶ GCode Preview.vi Get Log.vi Spindle On.vi Spindle Off.vi NIWorx.mnu Bot Speak GCode → GCode File to Array.vi G-Code Array to File.vi Send GCode Command.vi BotSpeak Main Application Instance **** Buffer grbl Text to GCoo Buffer Grbl uArm Swift Pro Machine Control Example Interfaces nScope Keep TuftsAthletics Box ZRent OnShape Google Sites SIS Tufts Trunk AmazonSmi Getting Started VI.vi Block Diagram * **-** ╬⇔+ ┅⊶ औ+ 🦦 💠 🚱 🥚 💶 💡 🔐 👆 🔂 12pt Dialog Font 150 - 🕐 Write Buffer EZ+BZ Dabo **∅** 60 →×read buffer





6 Measurement I/O Instrument I/O Vision and Motion Mathematics Signal Processing

-Y □ T F ¥

Flat Sequence

Structure

Diagram Disable Structure

File I/O

Synchronization

VI Analyzer

Data Communication

Formula Node

Conditional Disable ...

▶ ♠

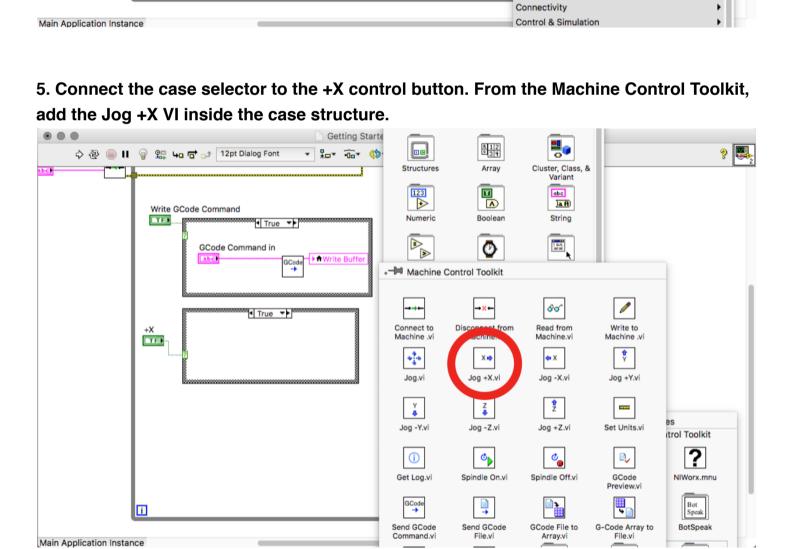
Local Variable

١

E

Global

?



Write Buffer SO DE DE Firmware 1701

-Z

Stop Button

TF

[®]◀ True ▼▶

*⁴ True ▼▶

-X +Z

+Y

6. Create a control for the jogging step size so you can adjust it from the front panel.

▼ ╬_ल┰ ╦┰ భ>┰ 🦦

String Out to the Write Buffer also using a local variable.

[®]◀ True ▼▶

GCode Command in

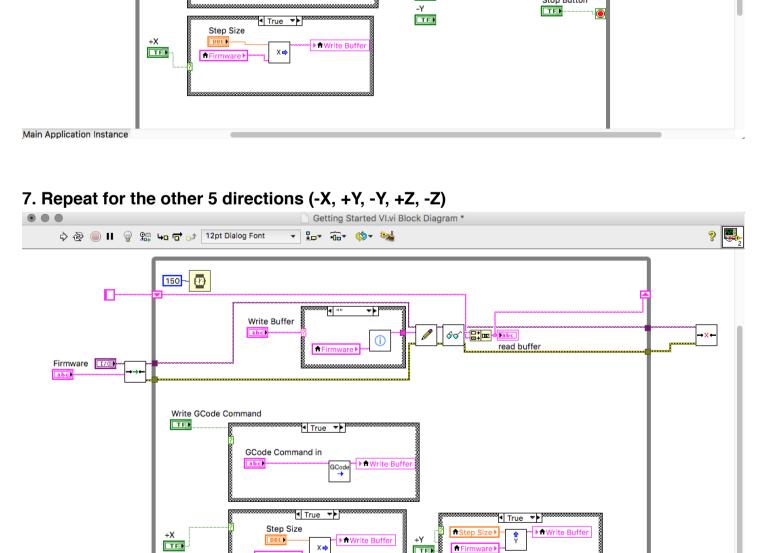
💠 🚳 📗 🛘 💡 😭 👆 📅 🚅 12pt Dialog Font

150 -

Write GCode Command

Wire the firmware into the firmware input using a local variable, and connect the G-Code

Getting Started VI.vi Block Diagram *



TF)

-Z

First, add another case structure, and create a control on the Units input. Wire that control to the case selector, and change the cases to "in" and "mm" instead of "true" and "false". Then create a local variable for the units and wire it to the units input, and wire the G-Code String Out to a Write Buffer local variable. Make sure to populate both cases in

Getting Started VI.vi Block Diagram * \bullet \bullet \bullet 💠 🚱 🥚 🛮 💡 🐫 👆 🖶 🖒 12pt Dialog Font ▼ ‱ •‰ • ♦♦ • 🖦

√ сомз

Grbl Stop Button ▼

▣

Write Buffer

-X TFI

+Z

8. Add control over the machine's units.

150 -

Main Application Instance

the case structure.

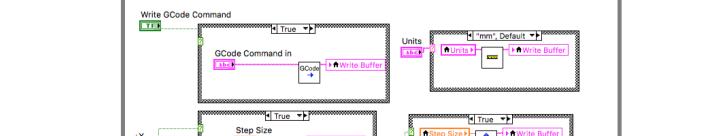
Firmware 1701~

Main Application Instance

[∞]◀ True ▼▶

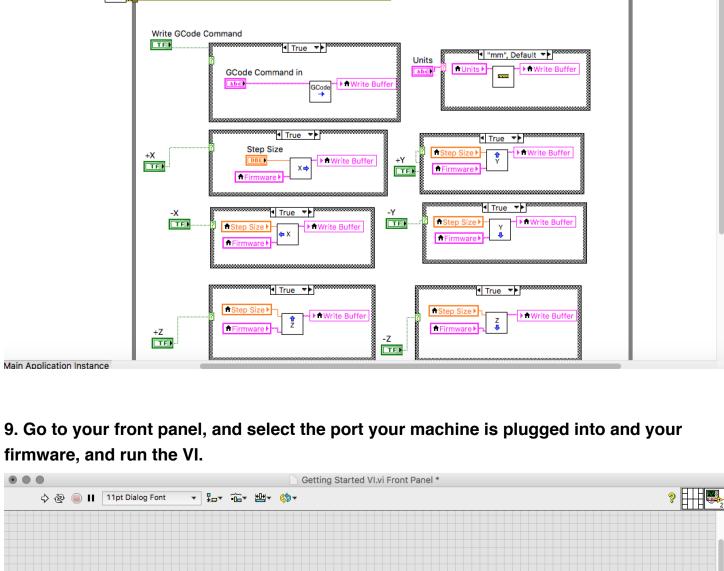
*⁴ True ▼▶

û Z



1

SO BE BLOC



Stop read buffer Write GCode Comman Step Size

GCode Command in

Write Buffer G91\sG0\sX1\n