ASSIGNMENT 1 - Galil Use Instructions

MTRN3500 Computing Applications in Mechatronic Systems - 2023

Alexander Cunio, James Stevens, & Jay Katupitiya

1 Introduction

This assignment requires you to interface with a Galil PLC. These are physical devices set up in room 212 of Willis Annex, for you to connect and test your software during laboratory classes. However, to facilitate you to develop your software when you are not in the lab (i.e. when you are not able to connect to a physical Galil PLC), a simulator is available for you. You can use the installer provided on Moodle by downloading and extracting the zip file. The starter code you use to begin developing your software is designed in such a manner that you can easily switch between the two system and continue your development work. In your software, at most you need is typing the word “true” to switch to simulator (for example when working at home), and replacing it with “false” or deleting the word “true” to switch to the physical PLC (when working in the lab). These changes need to be made in the provided ‘EmbeddedFunctions‘ class, which is explained later in this document.

2 Running on a physical Galil

Following the structure prescribed in the assignment spec and starter code, your implementation of the Galil class can be tested by instantiating a version of it within a `main()` function. There are two versions of the Galil constructor that must be written; use use the overloaded constructor to modify working with physical or simulated Galil.

```
Galil(EmbeddedFunctions* Funcs, GCStringIn address);
```

As you can see, it takes two arguments. The second is a string containing the IP address of the physical Galil itself (provided in assignment spec). The first requires you to pass in a version of the EmbeddedFunctions class, which must first be instantiated appropriately. When running on the physical Galil, default construction of the ’EmbeddedFunctions’ class can be used. An example of how you might setup your variables in your main function in order to run on the physical Galil in the lab is as below.

```
EmbeddedFunctions funcs;
Galil myGalil = Galil(&funcs, "192.168.0.120 -d");
```

In order to actually run your code on the physical Galil, you will need to be logged into the lab computers (within the lab class in Willis Annexe 212) and open up the solution you are working on within Visual Studio. The recommended method for transferring the code from your own computer to the lab one would be to add the project to a private Git repository (as described in the setup steps of the assignment and here) and clone it onto your account on the lab computer. You can then modify, test, and push any changes you make.

3 Running on the simulator

The Galil simulator can be used when developing remotely (away from the lab). It graphically replicates the functionality of the physical Galil that you have access to in the lab. However, please note that not every command listed in the command reference manual of Galil has been implemented, only those that may be useful for the assignment.

If you feel you have found a bug with the simulator please post it in the 'Assignments->Simulators' channel on the Edstem forum. If these require an update, we will release a new version for you to download and announce this in the same channel. This should be your first port of call for all problems related to the simulator.

Switching between working on a physical Galil and the simulator is carried out in your main function during the creation of your 'EmbeddedFunctions' object. Within the EmbeddedFunctions.h header file, there are two possible constructors provided:

```
EmbeddedFunctions();
EmbeddedFunctions(bool use_simulator);
```

Previously, the first was used for normal construction (working with a physical Galil) and the second is for specifying that a simulator is being used (passing in an argument of `true` into the constructor.

Overall, your construction when running with the simulator would hvae the form

```
EmbeddedFunctions funcs(true);
Galil myGalil = Galil(&funcs, "192.168.0.120␣-d");
```

Note that the IP address of the second argument of the Galil constructor when running on the simulator is ignored, however a string must still be supplied.

The simulator can be run on your computer using the provided zip file (located under the assignment files on Moodle). Please watch for any announcements of updates made to this within the 'Assignments->Simulator' Edstem forum channel which may require you to install a new version. To use the simulator, run the executable 'Animation.exe' from within the extracted folder (all the other files must remain in the same folder as the executable for it to function correctly).

Note: if you have problems running the simulator after installation, you may need to first run the supplied executable 'VC_redistx86.exe'.

When testing code with the simulator, ensure you open the simulator first then run your code and you should be able to see the result displayed on the simulator.

Please remember, your in-person assessment will be conducted with the physical Galils so make sure you test on them during your designated lab classes.

4 Digital input configuration

The digital inputs on the physical Galil can be configured in one of two ways. They may be either 'direct' inputs or inverting. This affects the active level of the input. The configuration of each individual digital input can be configured as either state '0' or state '1'. When bit configuration is set to 0, current flowing through it will return a 0 (inverting state). When the bit is configured as 1, current flow returns a 1 (direct state).

The simulator is established such that the digital inputs are always direct. However, this can be configured manually on the physical Galil using the "IQ" command.

To switch to direct inputs send the command:

```
IQ 65535;
```

To switch to inverted inputs send the command:

```
IQ 0;
```

For simplicity, these commands can be sent to the physical Galil using the online interface found by typing '192.168.0.120' into a search browser (this can only be accessed from the lab machines with the physical Galils).