Introduction

This document provides an introduction to software that controls bioreactors of the Bioflo product family. The functionality of the software includes setpoint control, data logging, and protocol execution. Simple experimental protocols can be entered as a CSV scripting file, and a Python-based protocol execution model is included for more demanding conditional experimental control. This document will also introduce the plugin interface, and how it is used to create software for the integration of auxiliary hardware without altering the existing manger software. This framework was designed to be a flexible and free alternative to commercially available solutions. The software package, source code, and various auxiliary plugins are publicly available for download from https://github.com/LibourelLab/BiofloSoftware. Plugins (and the un-compiled actor VI's) depend on the plugin interface framework, and cannot be executed as standalone programs.

Software distribution

- Download repository as a ZIP file by navigating to https://github.com/LibourelLab/BiofloSoftware and selecting the "Download zip" button
- Once downloaded, unzip contents to a local PC directory
- Folder contents overview:
  - **Manager Installer:** contains compiled manager software and an installer for 32 and 64 bit windows users (XP SP3 or newer)
  - **Manager Source Files:** contains manager source code which can be opened in Labview 2013
  - **Plugins:** contains plugins, which are pre-complied actors in the form of Packed Project Libraries (.PPL)
  - **Plugin Source Files:** contains actor source code which can be opened in Labview 2013
  - **SineWave.csv:** example protocol file used by the Time Protocol plugin to sinusoidally oscillate the temperature of a Bioflo reactor
  - **SineWave.txt:** example protocol file used by the Python Protocol plugin to sinusoidally oscillate the temperature of a Bioflo reactor
  - **Data Viewer:** contains a stand-alone executable file that graphical represents data in TDMS files

Software installation

- Navigate to BiofloSoftware\Manager Installer\Installer\Volume\Setup.exe to launch the windows installer
- The windows installer will download necessary run-time engines which have system requirements listed at http://www.ni.com/labview/requirements/
- Follow the on screen instructions to finish the installation

Bioflo hardware settings

Bioflo 3000 machines can operate in "multidrop" or single unit mode. The operation mode is specified in by a dipswitch (see user manual). The Bioflo 110 always operates in multidrop mode. Reactors that are run in multidrop mode can share a single communication bus, and are identified by a unique multidrop ID. This multidrop ID is assigned a value between 0 and 15 through dipswitches for the Bioflo 3000, or in the communication setup screen (Bioflo 110). For the Bioflo 110, a single base address is set in the communication setup screen of the primary control unit, and each individual reactor (up to four) are given an address that is incremented by a value between 0 and 3, corresponding to reactors 1-4. The default address for the Bioflo 3000 is 0. The bioreactor plugin detects whether a reactor is operating in a multidrop mode. If so, it will use a multidrop ID address specified at startup, otherwise is will ignore the address provided. Therefore, specifying a multidrop ID of 0 for a Bioflo 3000 machine with factory default settings should always work. If no address is provided, the unit will be addressed as a standalone 3000 Bioflo machine, which will only work if the reactor was configured through dipswitches as standalone.

Using the software

The software has the following hierarchical structure: The "Manager" is the main program that is started from the operating system. The manager contains at least one, but can contain several control systems. Each control system is independent, and does not share information (but can share hardware) with other control systems. A control system contains at least one, but can contain several plugins. Each plugin interact with a specific piece of hardware or conveys some other kind of other functionality, such as the "protocol" plugins. All plugins within the same control system share the same database, and can therefore interact with each other. The protocol plugins take advantage of this by adding programmable logic to the otherwise autonomous hardware control plugins (such as the bioreactor plugin). The same plugin root folder can be used for multiple control systems, but the data path needs to be unique for each control system.

Hot starts: If the data folder contains previously recorded data of a compatible control system, the control system performs a hot start and resumes execution using all previously used settings, and append data to the existing logs. If no such data exists, the plugins will read the current setpoint values. Note that the folder browser that is used to locate the data path does not show files. Use windows explorer to verify that a data directory is empty instead.

- The manager software can be launched from the desktop or start menu shortcuts created by the installer
- Once launched, the user will see the manager's configuration screen for the first control system. This screen requires the user to name the control system, to select a plugins root folder that contains all plugin folders used for the control system, and a data logging folder, which is the folder that contains the data logs produced by the plugins of that control system. Each control system needs to have its own unique data folder, even if the same plugins are being used for several control systems.
- Secondary configuration popup screens will appear for some of the plugins that are loaded for the control system. If the reactor plugin is part of the plugin root folder (default), it will generate a configuration screen prompting for a com port address, a multidrop ID, and a sampling interval. Specify the com port address and the multidrop ID (if any, see Bioflo hardware settings above), as well as a sampling interval in milliseconds. Note that the time interval should be chosen to be slow enough for the program to be able to handle the communication load. If the selected time interval is too short, the message queue will fill up more quickly than it can be processed. This can result in very long shutdown periods, because the manager will only shutdown once all messages in the queue have been processed. A sampling rate of 5 seconds (5000), if one Bioflo unit is attached to a multidrop bus, is approximately the minimum sampling interval. If more units are attached, the minimum sampling interval should be chosen to be 5 seconds times the number of units. Regardless, fast sampling rates will lead to very large data files.
- Additional reactors can be added by creating additional control systems.

Navigating the Manager VI

- **Add control system:** launches "Add control system" prompt to add an new control system
- **Control system menu:** Selects the control system displayed in the subpanel
- **VI menu:** Selects one of the plugins of the control system for data display in the main subpanel
- **Remove control system:** Removes individual control systems, stopping the included plugins. If only one control system existed, the manager closes as well

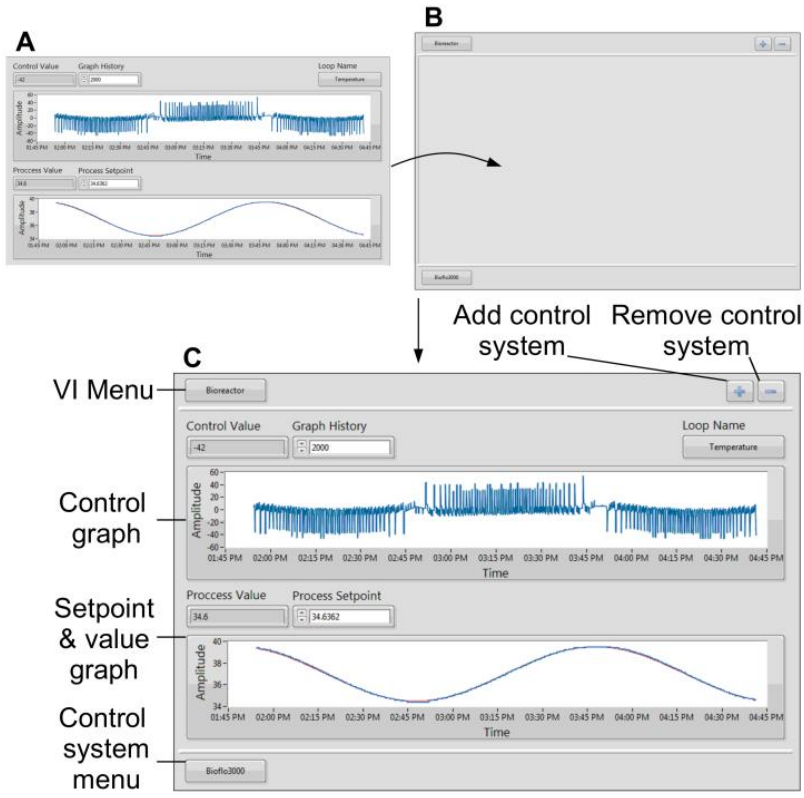

Figure 1 Bioreactor & Manager VI

Navigating the Bioreactor VI control system

- **Loop Name:** this menu control is auto populated by the Bioflo hardware at start up to lists all control loops. It selects which control loop the Bioreactor VI is currently displaying on its front panel.
- **Control value:** displays the value of the control output that is used to regulate the process. This value is either set manually, or automatically determined by PID control (see below)
- **Control graph:** displays the control output value over time a period of time
- **Process value:** displays the most current process value
- **Process setpoint:** displays the current process setpoint
- **Setpoint & value graph:** displays the process setpoint and value over a period of time up to the current process value
- **Graph history:** selects the number of data points to include in the graphs. No data is shown if the selected period is much longer than the amount of data that has been gathered up to the current time point.

Navigating the Time Protocol VI

- **Protocol path:** displays the path of the current protocol
- **Open protocol:** Opens a CSV protocol file
- **Start status:** starts protocol execution
- **Pause status:** suspends protocol execution
- **Stop status:** resets and suspends protocol execution
- **Command list:** displays the protocol commands remaining
- **Command history:** displayed protocol commands executed

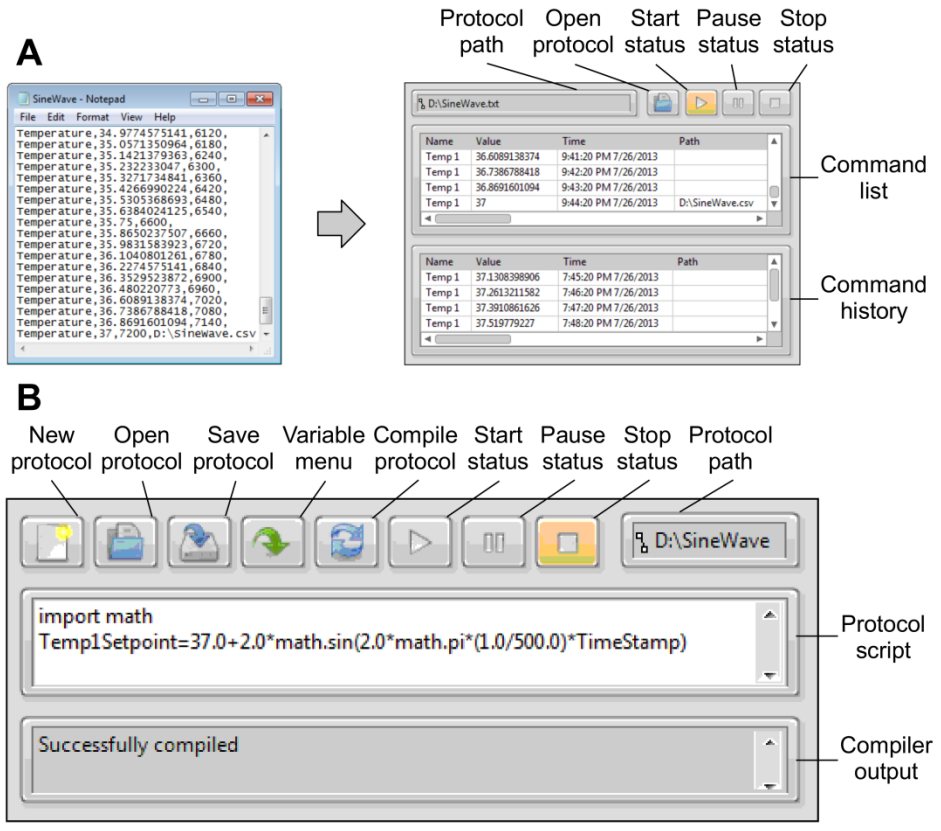

Figure 2 Time & Python Protocol

Navigating the Python Protocol VI

- **New protocol:** clears protocol script, compiler output, protocol path, and transitions to protocol execution stopped status
- **Open protocol:** opens a text protocol file
- **Save protocol:** saves the current protocol to PC and updates protocol path
- **Variable menu:** displays the variable names from the shared data source
- **Compile protocol:** checks the protocol script for compiler errors
- **Start status:** starts protocol execution
- **Pause status:** suspends protocol execution
- **Stop status:** resets and suspends protocol execution
- **Protocol path:** displays the path of the current protocol
- **Protocol script:** contains the Python code that executes the script
- **Compiler output:** displays errors encountered when compiling the protocol script

Developing Protocols

Time Protocol

The time protocol VI modifies control loop setpoints at specific times specified in the protocol script. Protocol scripts are command lines that contain: 1) execution time; 2) variable name; and 3) setpoint value. Protocol files can be opened from protocol command lines as well, allowing for the recurrent operation of a protocol, or the concurrent execution of additional protocols. Each Bioflo control setpoint that is available through the VISA interface can be modified using the protocol VI. The protocol scripts are stored in comma separated value (CSV) files which contain one protocol command in each line. The CSV file format was chosen to facilitate protocol editing in Microsoft Excel. As an example of a time protocol that implements a cyclic operation the SineWave.csv file was created (Supplementary material). The SineWave.csv file contains a series of commands that sinusoidally oscillates the temperature of the bioreactor over one period. The last command in the protocol contains the load command for itself. This prompts the protocol VI to load another period of the oscillation into memory.

Python Protocol VI

The python protocol VI uses an open source library called LabPython to interface with the Python scripting language. Python is an open source object orient code with a syntax that is designed to be highly readable. By utilizing the Python language conditional logic can be implemented in protocols within the same familiar, stable environment. As an example of a Python protocol, the SineWave.txt file was created (Supplementary material). This protocol duplicates the functionality of the SineWave.csv time protocol. For more information on the Python language and its syntax see http://docs.python.org/2.6/tutorial/.

Developing Plugins

Plugin development for auxiliary hardware integration begins with the use of a template actor. There are three types of template actors: Sensor, Control, and Sensor & Control. A sensor actor periodically reads values from the hardware, and does not contain any control capabilities (for example a thermometer). A control actor writes setpoints to the hardware when changed by user or a protocol, and contains no sensing capabilities (for example a heating element). A sensor & control actor contains the functionality of both previous template actors (for example a temperature controlled water bath). Choose which template actor is appropriate for use according to the capabilities of the hardware the plugin is being created for. Modification of template actors for plugin development consists of formatting serial communications according to the hardware communications protocol, and compiling the modified template actor into a packed project library. Detailed instructions of how to modify the template actors and compilation into packed project libraries are given bellow.

Sensor & Control Template

- Navigate to **BiofloSoftware\Plugin Source Files** and create a copy of the sensor folder. Rename this folder to a unique plugin name.
- Open Sensor & Control.lvproj located with the uniquely named plugin folder
    - Format hardware configuration

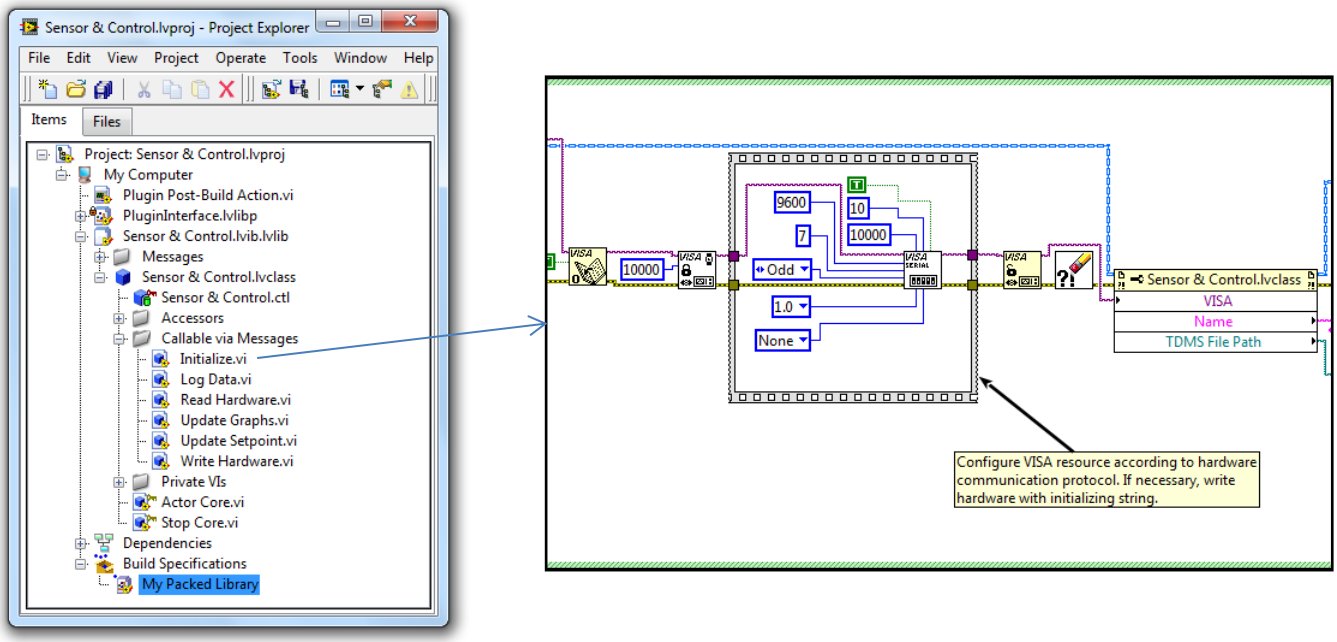

Figure 3: Initialize.vi

- Open Initialize.vi
- Enter “Control-E” to view block diagram
- Configure VISA resource according to hardware communication protocol. If necessary,

format VISA with initializing string.

- Enter “Control-S” to save changes
- Close Initialize.vi front panel and block diagram

◦ Format hardware communication

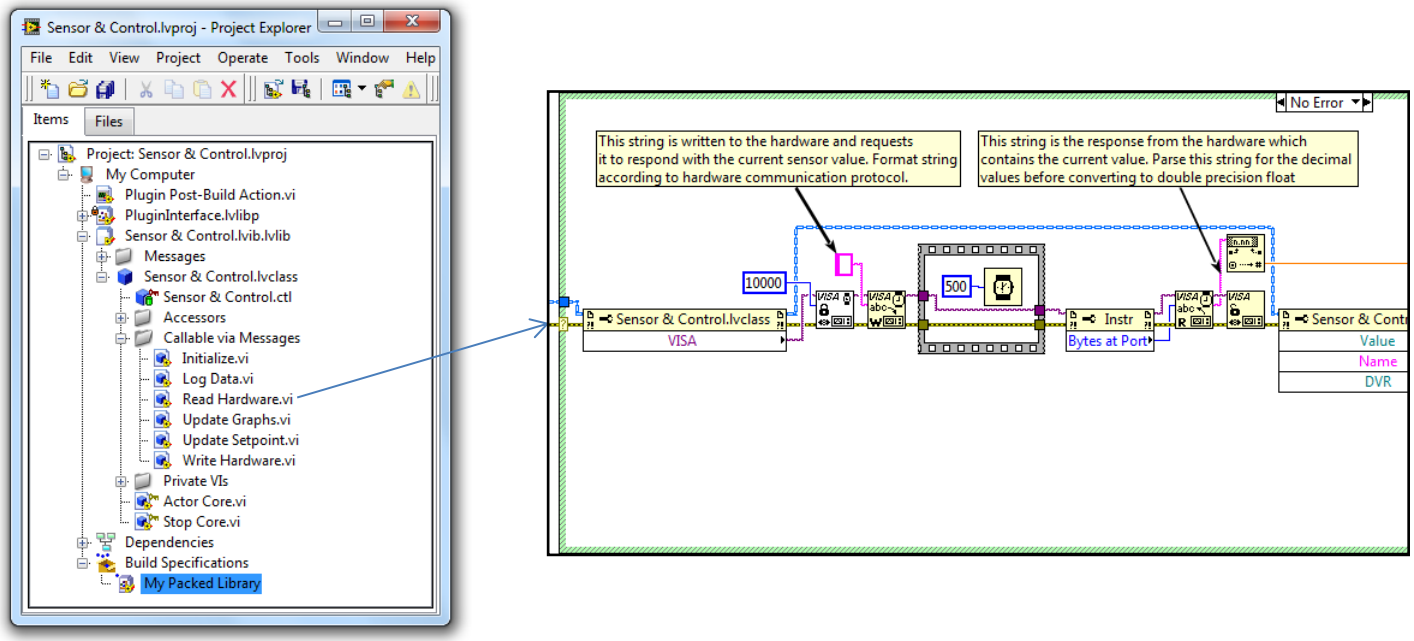

Figure 4: Read Hardware.vi

- Open the Read Hardware.vi
- Enter “Control-E” to view the block diagram
- Format the string written to the hardware that requests the sensor value.
- If necessary, parse the string received from the hardware for numerical values.
- Enter “Control-S” to save changes
- Close the Read Hardware.vi front panel and block diagram

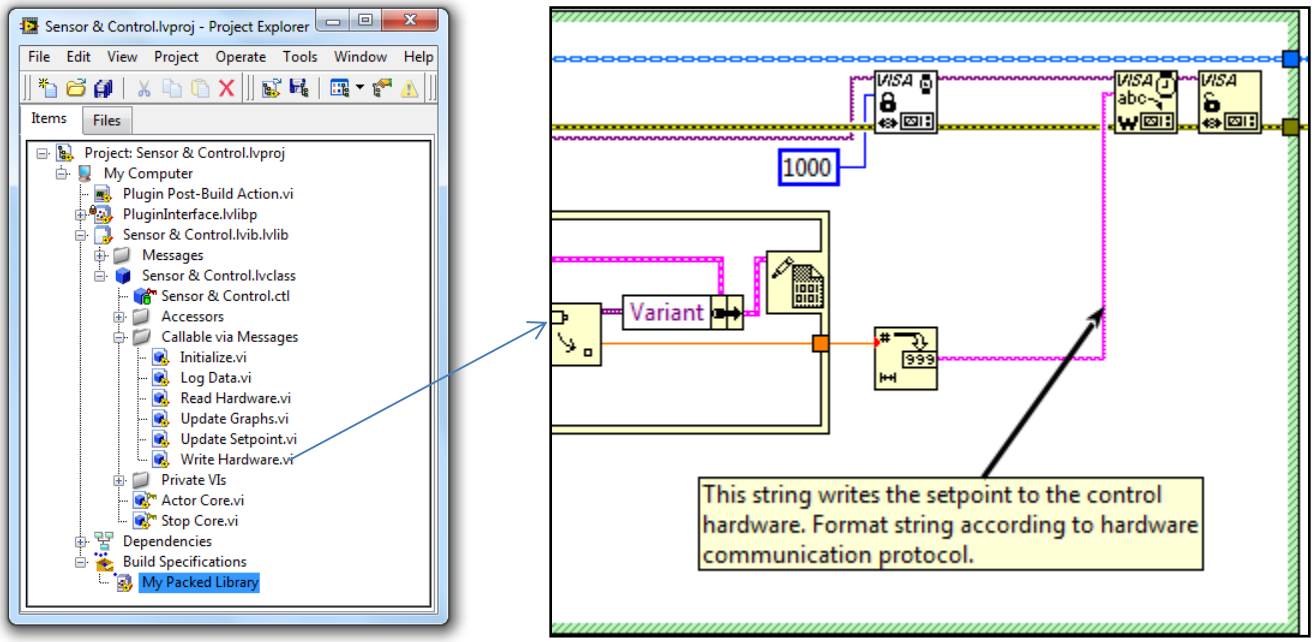

Figure 5: Write Hardware.vi

- Open the Write Hardware.vi
- Enter “Control-E” to view the block diagram
- Format the string written to the hardware that writes the setpoint to the control hardware.
- Enter “Control-S” to save changes
- Close the Write Hardware.vi front panel and block diagram

◦ Compile actor into packed project library.

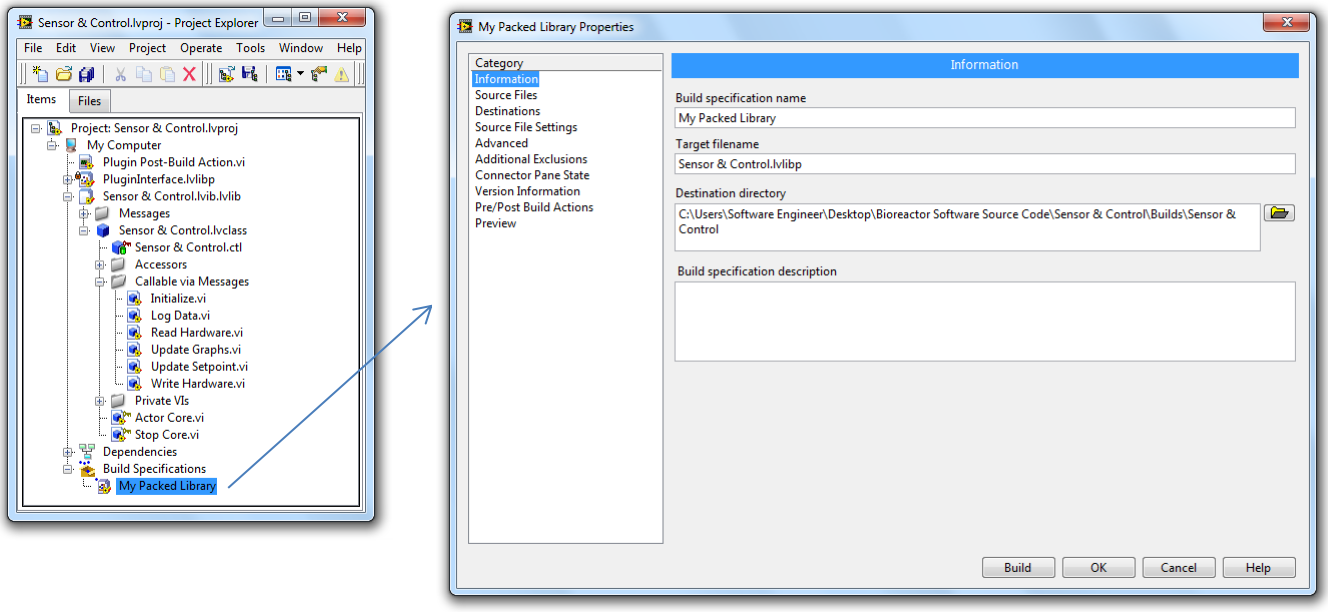

Figure 6: Compiling template actor

- Open “My packed Library”
- Modify the destination directory to a folder with the same unique plugin name used in the first step.
- Select Build
- Save project changes and close the project explorer

◦ Add plugin to a control system

- Move the plugin destination folder to C:\Program Files\Manager\Plugins\Default, or to another control system folder located within C:\Program Files\Manager\Plugins. This plugin will now be launched with the other plugins in the control system by the manager VI when control system folder is selected.

Control Template

The steps for developing plugins from the control template are the same as those for the sensor & control template, with the exception of formatting hardware communication for reading values. Because the control template does not read hardware, but rather writes to it, formatting hardware communication occurs in the Write Hardware.vi and Initialize.vi only. To develop plugins with the control template, follow the all instructions given with the exception of formatting hardware communication in the Read Hardware.vi.

Sensor Template

The steps for developing plugins from the sensor template are the same as those for the sensor & control template, with the exception of formatting hardware communication for writing setpoints. Because the sensor template does not write hardware, but rather reads to it, formatting hardware

communication occurs in the Read Hardware.vi and Initialize.vi only. To develop plugins with the control template, follow the all instructions given with the exception of formatting hardware communication in the Write Hardware.vi.