C series Controller Communications VI Library

User Manual: Version 1.0
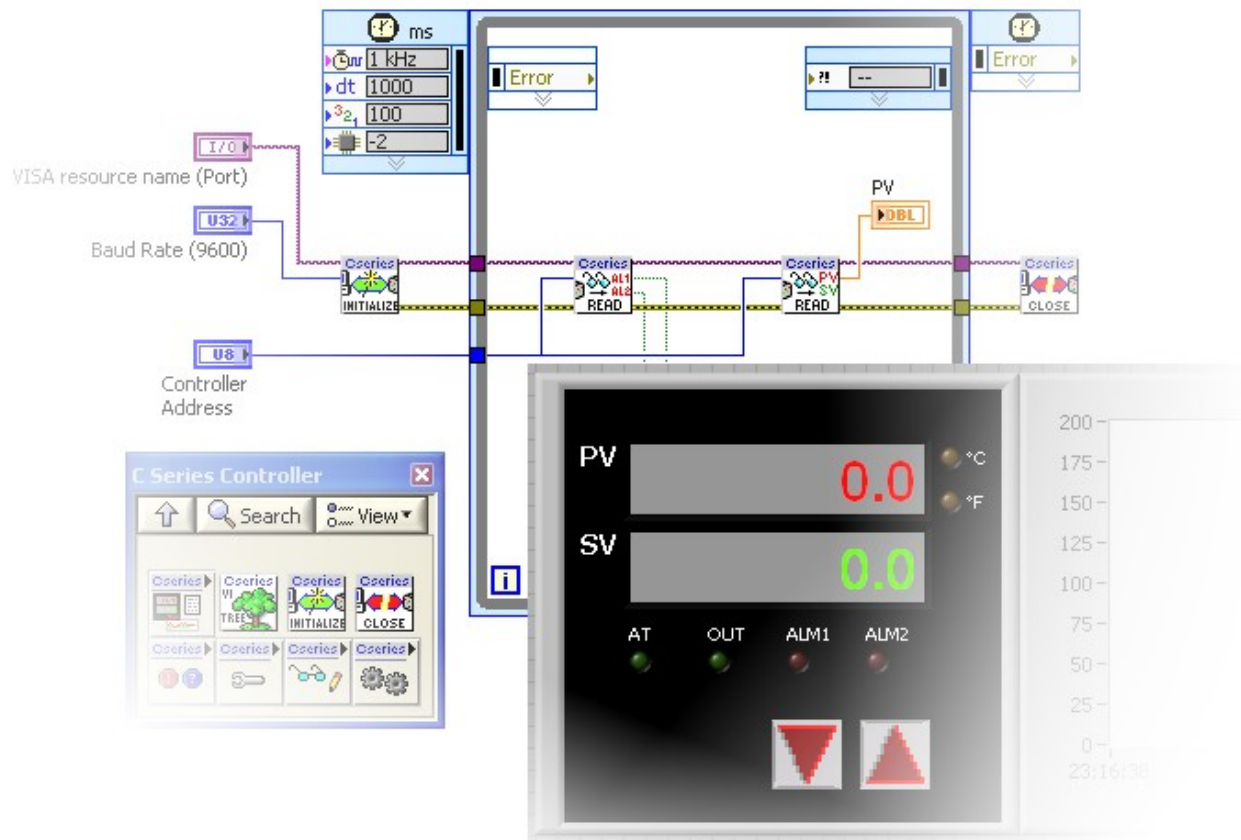
Table of Contents

1. Introduction

This VI library contains VIs and example code for implementing communication between LabVIEW™ from National Instruments and a Love Controls* C Series Controller via RS-485 serial bus. Communication between LabVIEW and Single Loop Controllers (SLCs), such as a C series Controller, allows expanded capabilities such as the ability to log and trend the process value (PV) and setpoint value (SV) or to remotely change the SV or configuration values such as alarm limits. A single instance of LabVIEW running on a PC can access multiple SLCs throughout an installation via the RS-485 bus to provide a central view or coordinated control and data acquisition capability at very low cost.

It is assumed in this document that the reader is familiar with how to program and use LabVIEW. Hardware and wiring are beyond the scope of this document. If you need help setting up the RS-485 serial bus, please consult the user manual for the Controller, your RS-485 serial interface documentation or one of the numerous references on the internet. If you want to hire a consulting engineer, visit our www site: http://www.integratedpro.com

CAUTION!!!! **The VI library can change the configuration, setpoint or other operating parameters of your Controller(s) and/or other devices connected to the RS-485 bus, resulting in changes to Controller behavior and especially if the Controller is connected to a physical process (i.e. operational and controlling something!) Never experiment with an operational controller – for development it is recommended to only use an offline controller, on an isolated RS-485 serial bus.**

*Integrated Pro is not affiliated with Love Controls, a Division of Dwyer Instruments, Inc.

http://www.integratedpro.com

2. Installation

Install the VI library by running the installation program, setup.exe. Choose the installation directory, and review the Software License Agreement. The VI library installs under the most current version of LabVIEW. Be sure to review the Readme.txt file information at the end of the installation.

Restart LabVIEW after installing so that the library VIs will show up on the Tools palette (right click the block diagram). You can find them on the C Series palette under User Libraries. You can access help and example VIs under the LabVIEW Help menu.

3. Communications Setup

Before using the VI library, you need to know three things:

1. Which communications port your RS-485 serial interface is using to communicate with the Controller. You can then use this information to set the VISA Resource which LabVIEW will use to communicate with the controller.
2. The address of the Controller on the RS-485 serial bus. This will be a number between 1 and 255.
3. The serial baud rate. By default, this is set to 9600 baud. Other serial settings should be set to 8 bit data, No parity, and 1 Stop bit.

Finding the Communications Port   To find out this information consult the documentation for your RS-485 serial interface. Most RS-485 serial interfaces install under Windows as a standard serial port, such as COM5, COM6, etc. In Windows, you can view the installed serial ports under the device manager. MAC OS and Linux have similar utilities.

Finding the Controller Address and Serial Baud Rate   On the C Series Controllers, the Controller address and serial settings are found using the Initial Setting Menu (press and hold the ENTER key for at least 3 seconds while at the Home Display). Consult the User Manual for further details.

RS-485 Serial Bus   All devices on the RS-485 serial bus must use the <u>same</u> serial baud rate, but have <u>different</u> addresses. LabVIEW acts as a serial Master device and the Controller acts as a Slave device. The VI library only supports RTU mode, not ASCII, communications. (note that some versions of the C series User Manual state that ASCII mode is used - this is incorrect)

http://www.integratedpro.com

4. Using Example VIs

Two example VIs are included with the VI library: C series Example Query Settings.vi and C series Example PV and SV panel and log.vi contained in the folder: [LabVIEW]\examples\C Series Controller, where [LabVIEW] represents the path to your LabVIEW installation.

Both of these VIs demonstrate how to use the subVIs contained in the VI library to communicate with the Controller and are meant to be starting points for verifying communication and for use in creating your own custom VIs.

Example VIs can be accessed either under C series Controller Library on the LabVIEW Help menu or from the Tools Palette (right-click on the block diagram). On the Tools Palette, choose the C Series Controller palette under User Libraries as shown in Figure 1. When you drop on of the example VIs on a block diagram, it will automatically open to show the front panel.
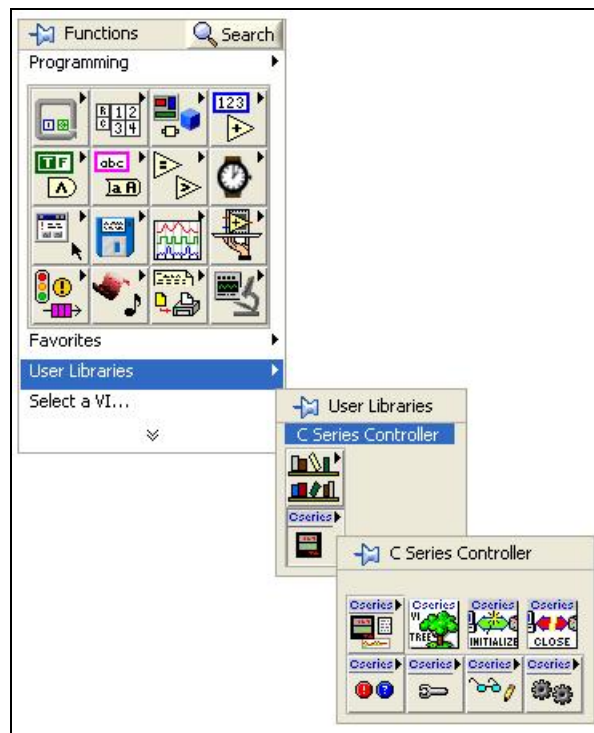

Figure 1 - C Series Controller VIs under User Libraries on the Tools Palette

http://www.integratedpro.com

C series Example Query Settings.vi   The purpose of this VI is to query the settings and error status of the Controller at the address you provide and report this in text form, as well as indicators. The front panel of this example VI is shown in Figure 2.

To run this example VI, follow the instructions on the front panel. The VI will execute a single time and the Controller Status Information should report all of the user configurable settings of the Controller. For example: whether the Controller is Running or Stopped, what the Temperature Units, Alarm Limits, and other values are. This information is also shown on the indicators on the right hand side of the front panel.
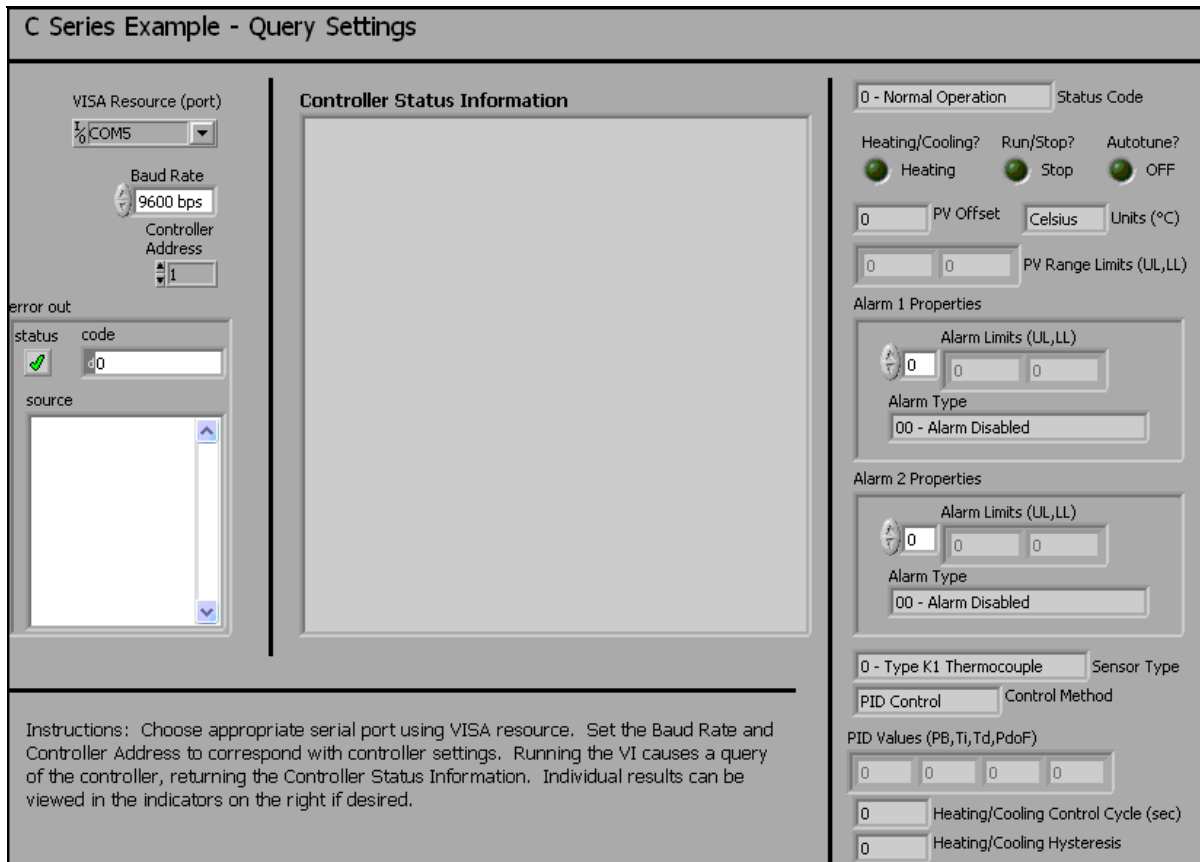

Figure 2 – C Series Example – Query Settings front panel

http://www.integratedpro.com

C series Example PV and SV Panel and Log.vi The purpose of this VI is to simulate the front panel of the Controller, showing the PV and SV values, temperature units, alarm LEDs, output LEDs and autotune status LED. The trend values of PV and SV are also logged to a chart on the front panel at an update rate of approximately 1 second. The front panel of this example VI is shown in Figure 3.

To run this example VI, follow the instructions on the front panel. The VI will execute in a loop, with an event structure controlling the update rate and responding to user input from the front panel controls.

By default, the VI will only retrieve the values of PV, SV and the LEDs. By turning the Write Enable switch ON, the VI can be used to set the value of SV either by clicking on the UP or DOWN arrow buttons on the simulated Controller panel, or by entering a new value for the Update SV numeric control. The Up/Down increment controls how much the SV changes when clicking on the UP or DOWN arrow buttons.
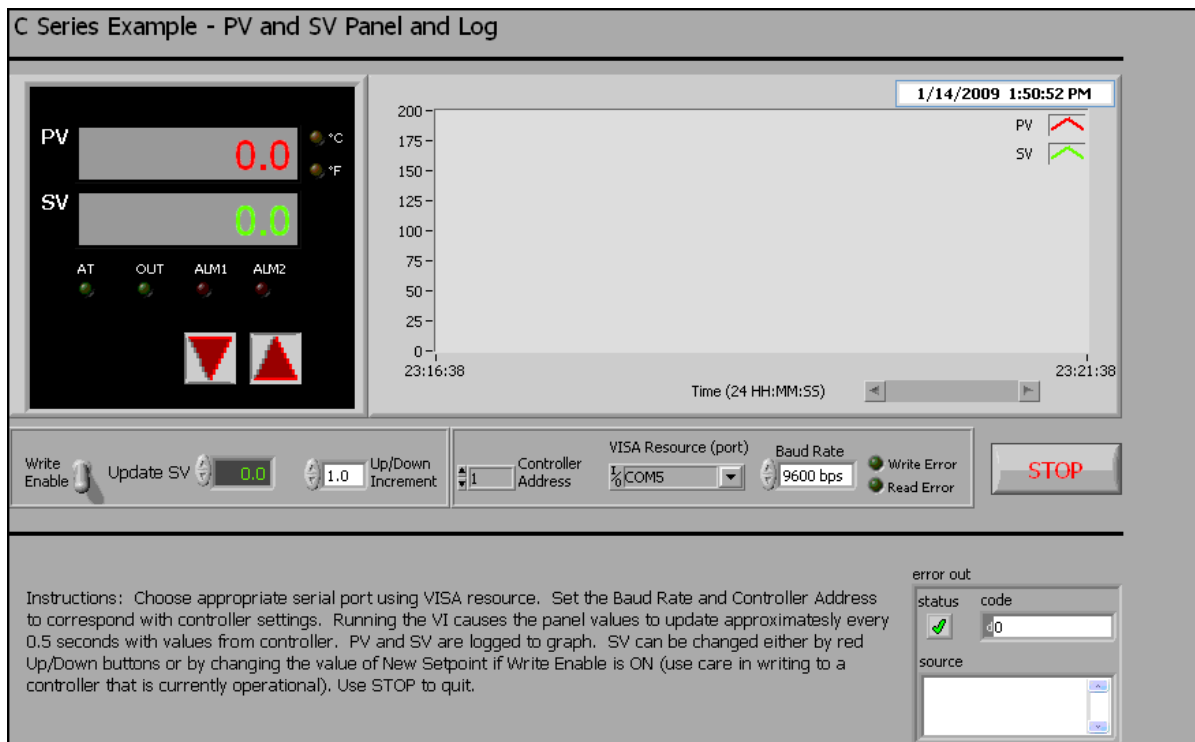

Figure 3 - C Series Example - PV and SV Panel and Log.vi front panel

5. Creating Custom VIs

After using the example VIs to verify communications between LabVIEW and the Controller, you can use the VI library to create your own custom VIs or modify the examples. Here is a simple example showing how to use a few of the library VIs to develop a custom VI.

CAUTION!!!! The VI library can change the configuration, setpoint or other operating parameters of your Controller(s) or other devices connected to the RS-485 bus, resulting in changes to Controller behavior and especially if the Controller is connected to a physical process (i.e. operational and controlling something!) Never experiment with an operational controller – for development it is recommended to only use an offline controller, on an isolated RS-485 serial bus.

A Basic VI Application Start with a blank VI and access by right-clicking on the block diagram of a new VI, choosing User Libraries from the Tools Palette and opening the C Series Controller palette as shown in Figure 1. Make it stay open "Pinning" the palette (click the push-pin icon in the upper left hand corner) of the C Series Controller palette.

Drag the following subVIs onto the block diagram and arrange them as shown in Figure 5: Initialize C series.vi, Close C series.vi, and Read Alarm LEDs C series.vi (from the Data subpalette). You can view the connector pane for each of the subVIs by turning on the LabVIEW context help (the **?** button in the upper right of the block diagram window) and hovering with the mouse over each of the icons. Notice that there are required inputs on some of the subVIs, so connect these up to a control or a constant.

When correctly wired, the front panel and block diagram should look similar to Figure 4 and Figure 5. Set the VISA resource name and baud rate, save the VI and run it. The state of the Alarm LEDs should now correspond to those on the Controller.

Adding More Functionality The basic pattern for the subVIs on the block diagram is always similar to that shown in the simple example, except that the Initialize C series.vi and Close C series.vi subVIs are only called at the beginning and end of the application and only once for any particular RS-485 serial bus, even if there are multiple Controllers. To further illustrate things, let's add a timed loop to read the values of PV.
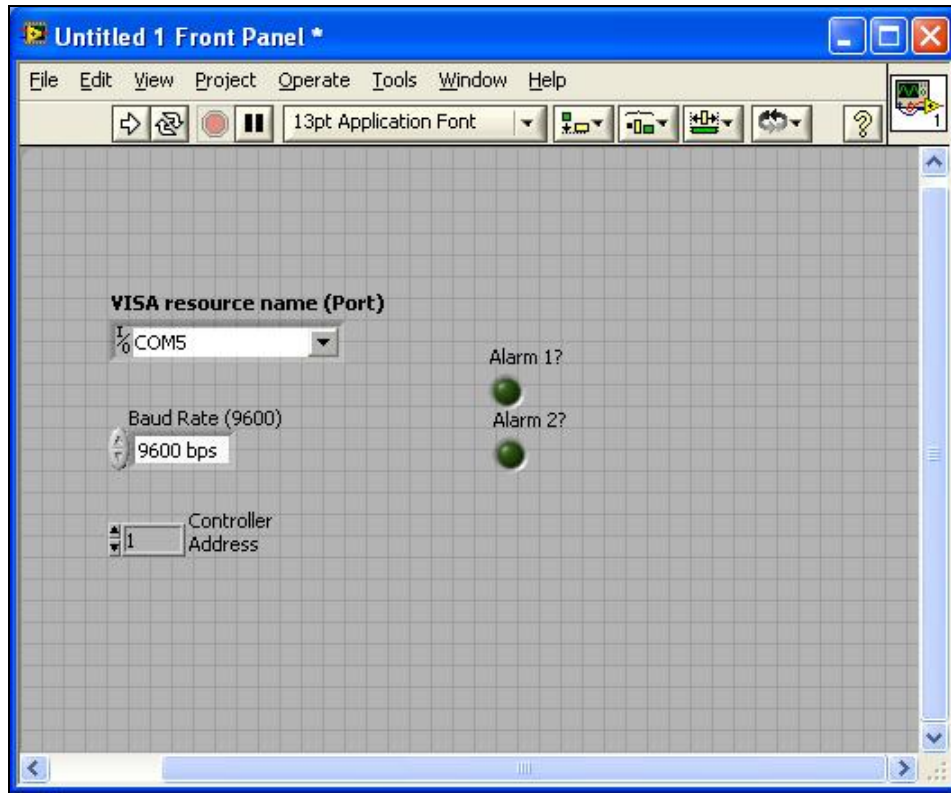
http://www.integratedpro.com

Figure 4 – Basic Application VI front panel
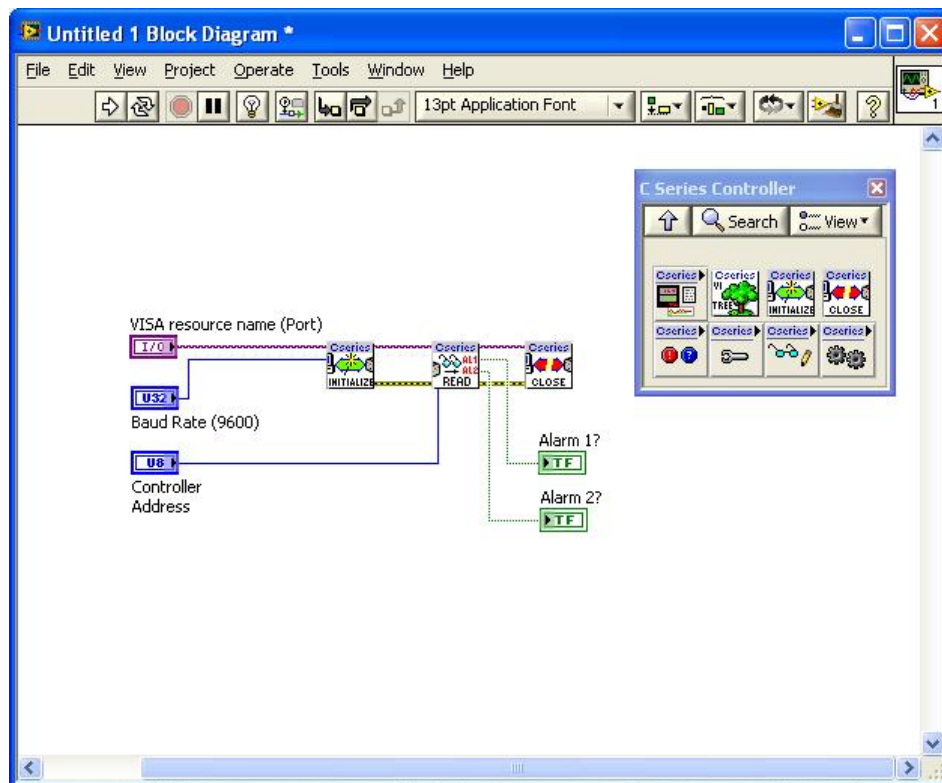
Figure 5 – Basic Application VI block diagram

http://www.integratedpro.com

Using the Basic Application VI as a starting point (save it as a new VI if you would like), add a timed loop (found on the Timed Structures subpalette of the Structures palette) to control how often the Alarm LEDs are updated. Leave the default timing values (1000ms update) and don't forget to add a STOP button control. Also add the subVI Read PV and SV C series.vi from the Data subpalette and wire it in as shown in Figure 6.

Your front panel should look similar to Figure 7.   Notice that Initialize C series.vi and Close C series.vi are called only at the beginning and end Also notice that the VISA resource and error wires are connected through all the subVIs from left to right and that each individual library VI requires the controller address. If you have multiple controllers on your RS-485 serial bus, you can call them individually by simply changing the controller address input on the given subVI.

Running the VI should cause the PV and Alarm LEDs to update approximately once per second to match the values on the front panel of the Controller.
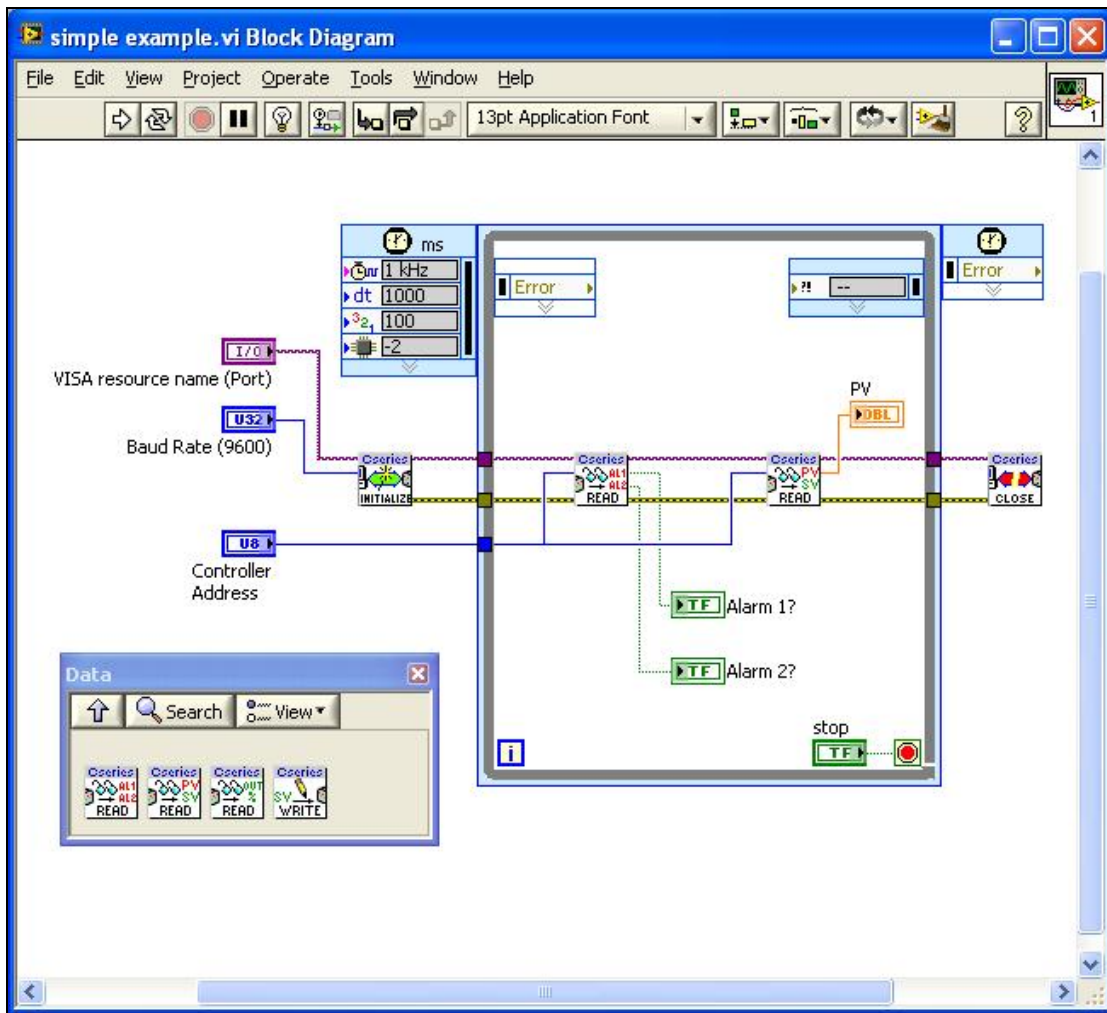

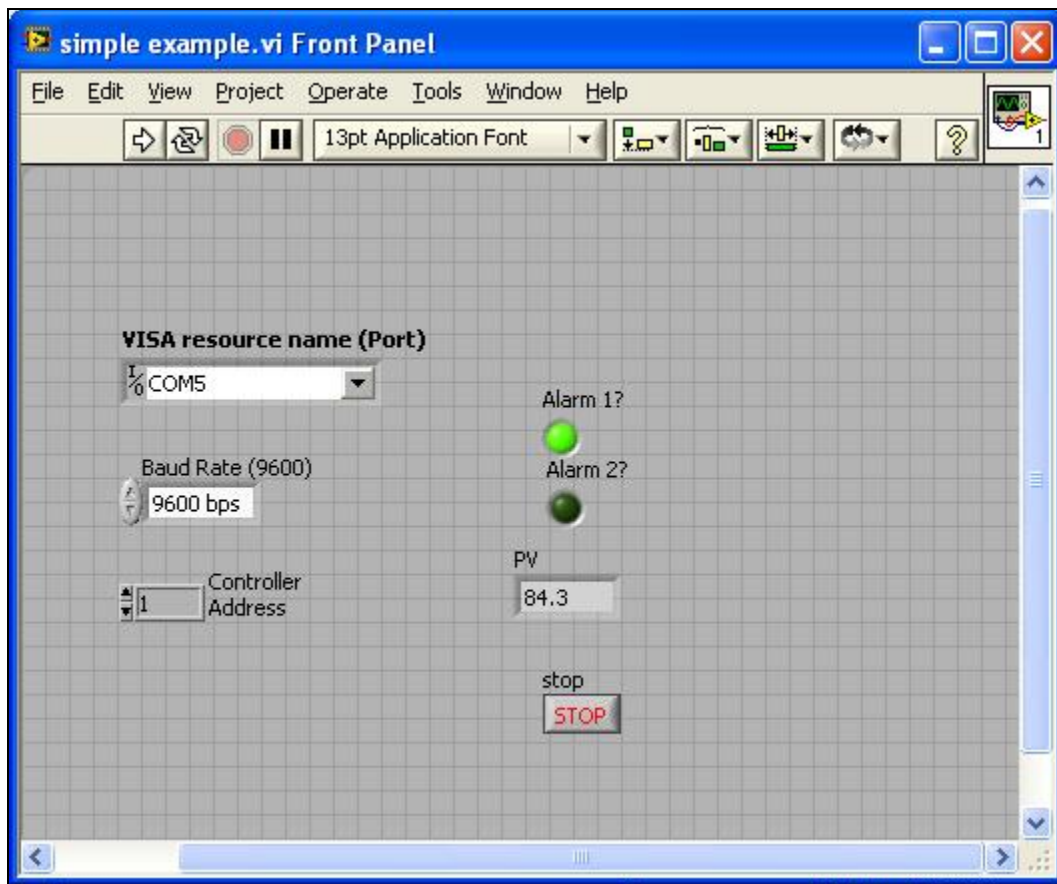**Figure 6 - Simple Example VI block diagram**

Figure 7 - Simple Example front panel

Changing Configuration Values If you want to add the ability to change a configuration value on the Controller, this can be done by inserting a subVI and using the Set/Get property. For example, the block diagram of a simple VI to change the Alarm properties would look similar to Figure 8.

The default value of the Set/Get property is to Get (read) the current configuration value. If it is desired to change the value, then the desired values must be wired to the appropriate input of the subVI and the Set/Get property must be changed to Set (write). Note that the subVI will always Get (read) the value of the property after setting it and return the current configuration value.
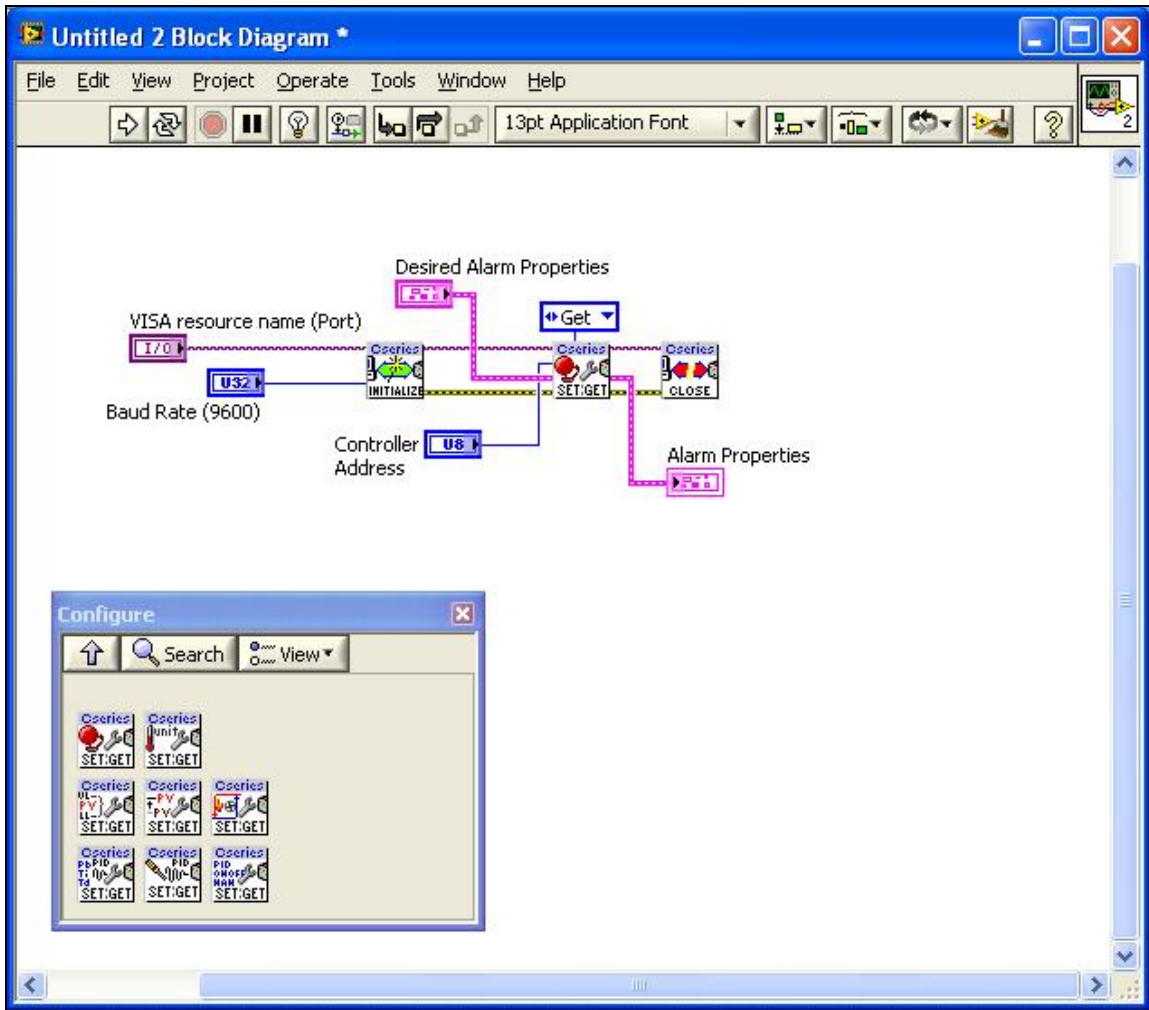
http://www.integratedpro.com

Figure 8 - Using Set/Get property to read or write configuration values

6. Library Reference

Each of the subVIs in the VI library is shown on the VI Tree.vi block diagram in Figure 9. The VIs in the library and the VI Tree.vi can all be accessed by right-clicking on the block diagram of a new VI, choosing User Libraries and opening the C Series Controller palette as shown before in Figure 1.

As shown in Figure 9, the subVIs in the VI library are divided into different categories, approximately indicating their usage in communicating with the Controller. The VIs in the Configuration and Action/Status categories use the Set/Get property to read or write configuration values. The VIs in the Data category are used to read or write data values, such as PV and SV or to read the current value of the Alarms or Output.

A brief listing of the functions of each of the subVIs in the library follows. A more detailed description of the input and output parameters for each subVI is contained in the VI documentation for each individual subVI. This can be easily viewed by turning on the LabVIEW context help (the ? button in the upper right of the block diagram window) and hovering with the mouse over each of the subVI icons.

A mapping of the correspondence between library VIs and communication register addresses on the C Series Controller is given in Table 1.
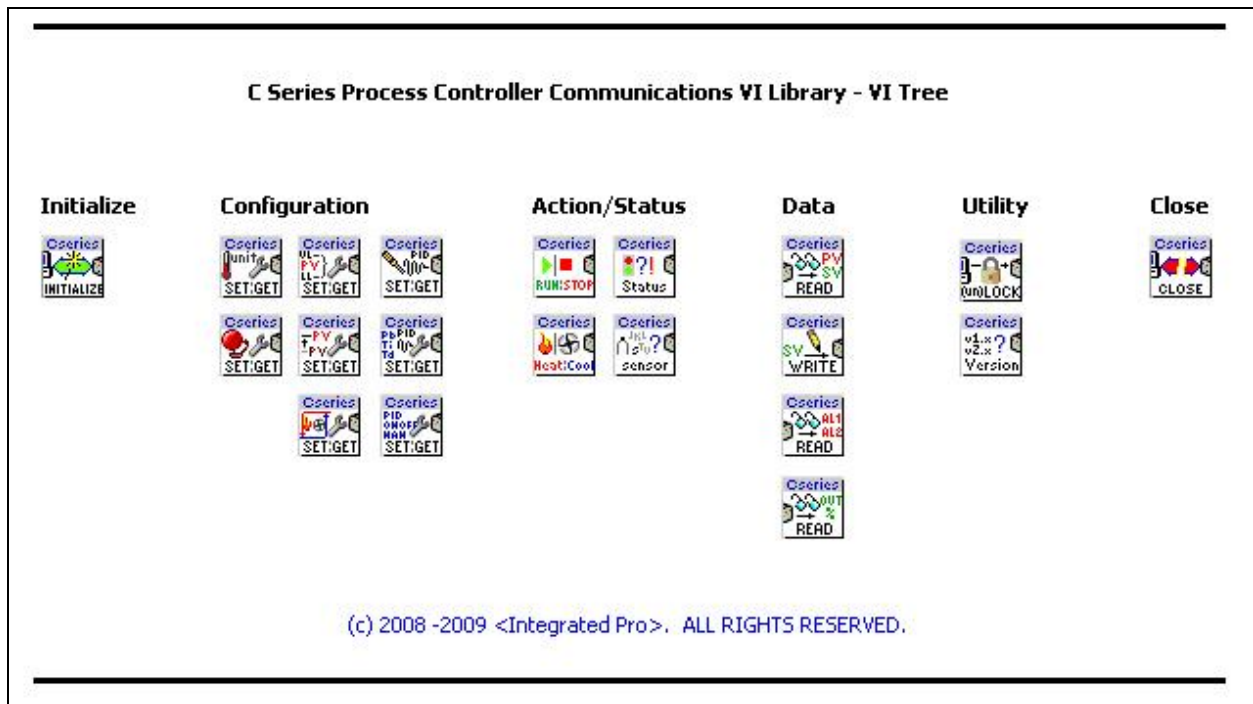

Figure 9 - VI Tree.vi block diagram showing library subVIs

Initialize C series.vi

Call this VI before calling other process controller communications functions, but generally you need to call the Initialize VI only once at the beginning of an application. Initializes Serial Communications on given port at chosen baud rate. Returns error if unsuccessful.

Assumes RTU mode communications and uses the following serial port settings: 8bit data, No Parity, 1 Stop bit with Termination Character disabled, with port timeout of 10seconds. Other settings are at default for the platform. Also stores global Optional Timing Parameters to be used in RS-485 communications.

Set or Get Heating Cooling Mode C series.vi

This VI gets or sets the Heating/Cooling Control mode register value (0x4718H). See Controller documentation. This VI always reads the current contents of the register in addition to writing a new value if desired. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Set or Get Run Stop C series.vi

This VI gets or sets the Control Run/Stop setting register value (0x4719H). If this value is set to Run, the output of the process controller is enabled. This VI always reads the current contents of the register in addition to writing a new value if desired. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Get Error Status C series.vi

This VI gets the status code register value (0x472BH). See Controller documentation. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Get Sensor Type C series.vi

This VI gets the Sensor Type Register value (0x4710H). See Controller documentation for sensor type table and limits. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Set or Get Alarm Configuration C series.vi

Use this VI to get or set the Alarm Upper and Lower Limits and Type (0x4702H - 0x4705H and 0x4715H - 0x4716H) See Controller documentation. This VI always reads the current property value in addition to setting the property value if desired. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Set or Get Autotune ON OFF C series.vi

This VI gets or sets the Autotune setting register value (0x4729H). See Controller documentation. This VI always reads the current contents of the register in addition to writing a new value if desired. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Set or Get Control Method C series.vi

Use this VI to get or set the Control Method property (0x4711). See Controller documentation. This VI always reads the current property value in addition to setting the property value if desired. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Set or Get Heating Cooling Values C series.vi

Use this VI to get or set the Heating or Cooling Parameters (0x470BH and 0x4712H). See Controller documentation. This VI always reads the current property value in addition to setting the property value if desired. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Set or Get PID Values C series.vi

Use this VI to get or set the PID Control parameters (0x4708H - 0x470AH and 0x4713H). See Controller documentation. Note that the ioF value is always written/read by this VI even if the value of Ti = 0 and the PdoF value is always written/read even if Ti is non-zero. This VI always reads the current property value in addition to setting the property value if desired. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Set or Get PV Offset C series.vi

Use this VI to get or set the Process Value (PV) Offset (0x4714H). This value offsets the PV value to match an external reference. See Controller documentation. This VI always reads the current property value in addition to setting the property value if desired. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Set or Get PV Range Limits C series.vi

Use this VI to get or set the Process Value (PV) Range Upper and Lower Limits (0x4706-0x4707H). See Controller documentation. This VI always reads the current property value in addition to setting the property value if desired. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Set or Get Temperature Units C series.vi

Use this VI to get or set the Temperature Unit Display Selection property (0x4717). This VI always reads the current property value in addition to setting the property value if desired. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Read PV and SV C series.vi

Reads Process Value (PV) and Setpoint Value (SV) from Process Controller. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

http://www.integratedpro.com

Write SV C series.vi

Writes Setpoint Value (SV) to Process Controller. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Read Alarm LEDs C series.vi

This reads whether the Alarm LEDs are on or off on the controller from the LED Status Register value (0x471CH). Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Read Output LED and Amount C series.vi

This reads whether the Output LED is on or off on the controller from the LED Status Register value (0x471CH). It also reads the Output amount in % full scale (0x471EH). Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Get Controller Firmware Version C series.vi

This VI gets the software version register value (0x471BH). See Controller documentation. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Lock or Unlock Communication Write In C series.vi

This VI gets or sets the Communication Write In lock register value (0x471AH). If this value is set, remote communications cannot write in register values to the Controller (except this register). Note that this is not the same thing as Panel Lockout function on the actual device. This VI always reads the current contents of the register in addition to writing a new value if desired. Error checking is used - If error input is wired, this VI checks for previous errors and does not execute if error.

Close C series.vi

This terminates the software connection to the RS-485 serial bus. Call this VI to make sure the serial port is closed at the end of your application or if your application stops on an error.

http://www.integratedpro.com

Table 1 – C Series Controller Communication Register to Library VI Mapping

To Read/Write Address(es)	Use Library VI (s)
4700H	Read PV and SV
4701H	Read PV and SV, Write SV
4702H – 4705H	Set or Get Alarm Configuration
4706H – 4707H	Set or Get PV Range Limits
4708H – 470AH	Set or Get PID Values
470BH	Set or Get Heating Cooling Values
470EH	Set or Get PID Values
470FH	n/a (not supported by Library VIs)
4710H	Get Sensor Type
4711H	Set or Get Control Method
4712H	Set or Get Heating Cooling Values
4713H	Set or Get PID Values
4714H	Set or Get PV Offset
4715H – 4716H	Set or Get Alarm Configuration
4717H	Set or Get Temperature Units
4718H	Set or Get Heating Cooling Mode
4719H	Set or Get Run Stop
471AH	Lock or Unlock Communication Write In
471BH	Get Controller Firmware Version
471CH	Read Output LED and Amount, Read Alarm LEDs
471EH	Read Output LED and Amount
4729H	Set or Get Autotune ON OFF
472BH	Get Error Status

http://www.integratedpro.com

7. Optional Parameters

Optional Timing Parameters   Initialize C series.vi is used to open communications between LabVIEW and the RS-485 bus. It also sets global values for the Optional Timing Parameters to be used in RS-485 communications. These are values specific to RS-485 communications that are used throughout the program. They can be modified through an input cluster on the Initialize C series.vi, but unless dictated by such factors as electrical interference on the RS-485 bus, for instance, it is generally not necessary to change these values:

>   Max Retries (5): The number of times the program will retry automatically after receiving a serial communication error.

>   Optional write delay (20ms): The modbus specification requires that the RS-485 serial bus be quiet between transmitted messages. These delays are based on the baud rate. It may be desirable to insert additional delay to account for noise, slow settling, slow device response or other factors. This optional delay is inserted before the required minimum delay, before a message is written.

>   Optional read delay (20ms): (see Optional write delay, above) This optional delay is inserted before the required minimum delay, before a reply is read.

>   Device read timeout (200ms): Specifies how long to wait for a reply from the device before retrying. This delay is a programmatic delay and not the same thing as the VISA serial port timeout, which will return an error in the case of hardware or other failure, and cause the subVI not to retry, but rather to return the error.

Optional Scaling Value   SubVIs that read or write the PV or SV, or that adjust configuration parameters relating to PV (such as Set or Get PV Offset C series.vi) have optional scaling value inputs. The default scaling value is 0.1 to apply a single decimal place to the value returned from (or written to) the Controller so that numeric controls in LabVIEW will correspond to those shown on the Controller front panel. For the C series Controllers, the temperature sensor input is also scaled this way.

Unless you have a good reason for doing so, it is not recommended to change these values; however, if you do, you should make sure that all VIs having this input terminal receive the same value for PV and SV scaling.