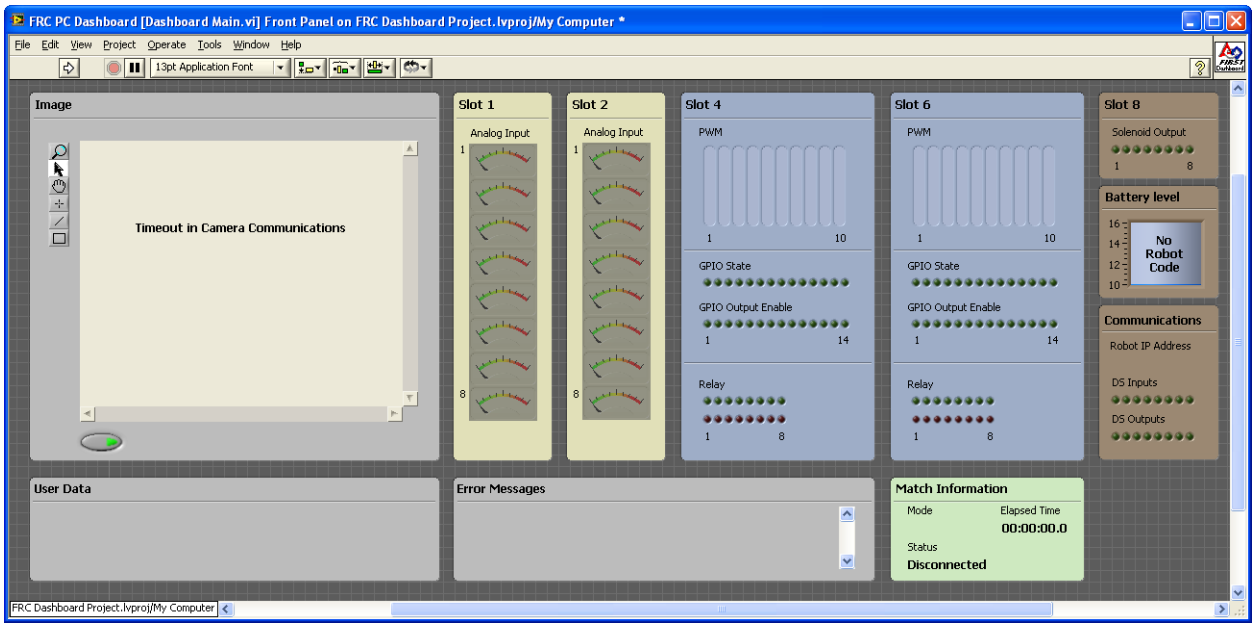
FRC Dashboard Tutorial

Contents

Introduction

In this tutorial, you will explore the FRC dashboard, and learn how to effectively use the dashboard to troubleshoot your FRC robot project.

The FRC dashboard displays information from the camera and cRIO-FRC modules, as well as communications information.



The dashboard is a great tool for testing – it will show you at a glance the state of any I/O channel and also display your own custom-defined data. We will cover how to display data from both the driver station and the cRIO on your dashboard.

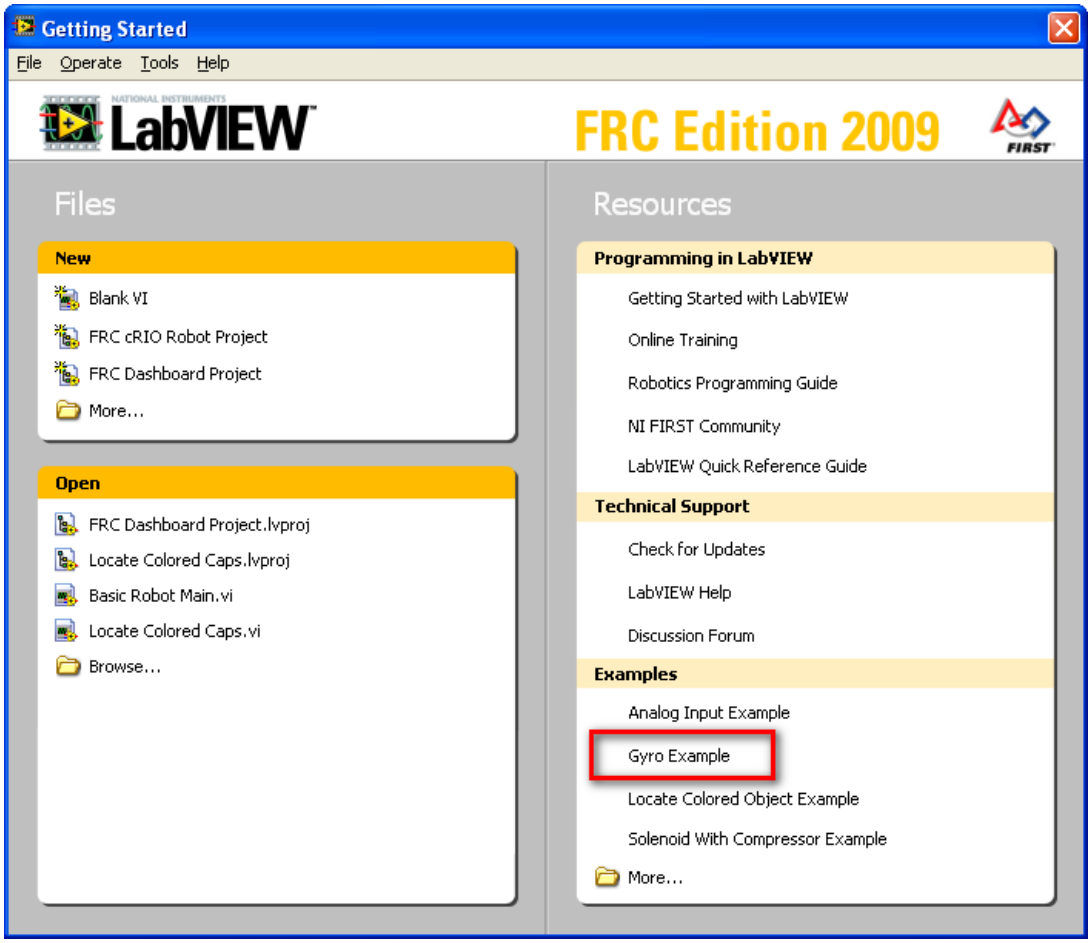
Throughout this tutorial, we will assume that you have all the necessary software for FRC installed on your computer, your cRIO is imaged, and the firmware on your Driver Station is up to date. If you don't

have these steps completed, you can refer to the “Getting Started” tutorials for information and instruction.

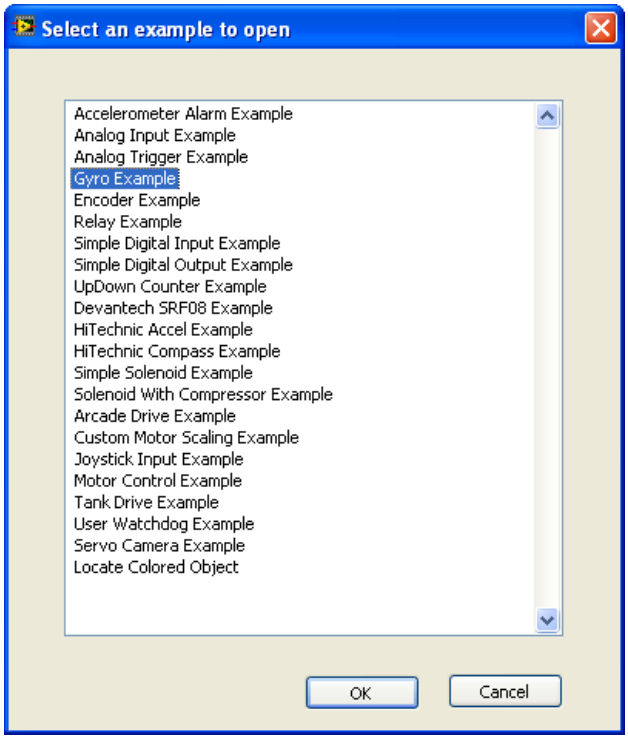
Editing the Gyro Example

Let’s open up an example program that will generate some data from the cRIO. We will be looking at the gyro example to demonstrate how to send directional data from the gyro sensor on the cRIO to the Dashboard VI running on your laptop. This is a great way to monitor the gyro’s functionality in different test conditions.

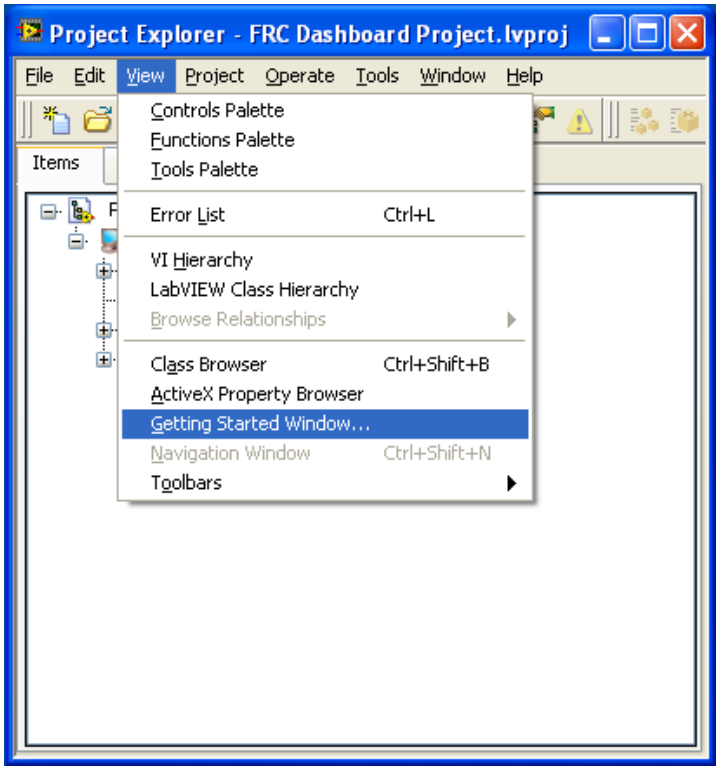
From the LabVIEW Getting Started Windows, select **Gyro Example** from the **Examples** section in the lower right.



If the gyro example doesn’t show up, click on the **More** button to bring up the complete list of examples.

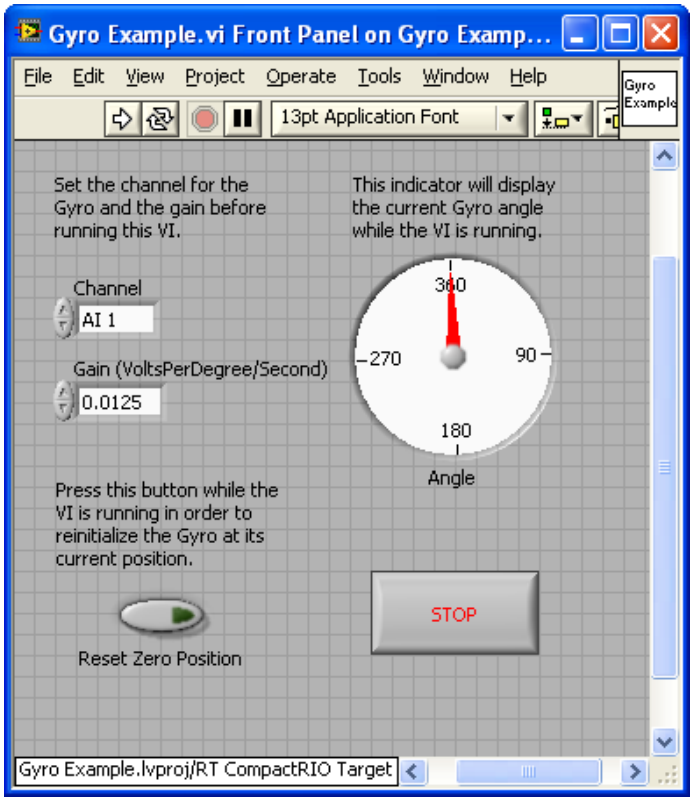


If you are already in another project or VI, you can access the Getting Started Window at any time by going to **View»Getting Started Window…**



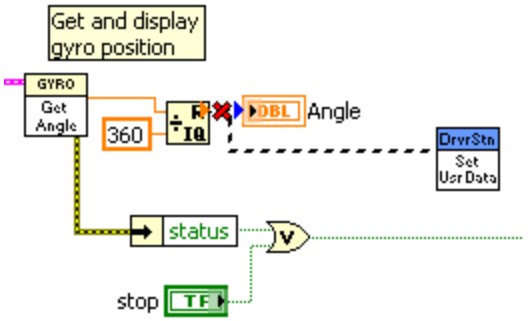
Once the example project loads, Go to **File»Save As** and save a copy of this project to a new folder. Name it *Dashboard Example*.

Double-click on **Gryo Example.vi** in the project to open it. This VI runs on the FRC cRIO and allows you to view the angle of the gyro. We are going to modify it to send information to the Dashboard VI running on the laptop.



Switch to the block diagram. Add a **Set User Data** VI from the WPI **Robotics Library»Driver Station** palette to the right of the *Angle* indicator. This VI will allow us to add user data to the communication packet being constructed and sent to the laptop. Wire the **Angle** output from the Get Speed VI to the **User Data** input of the Set User Data VI to have the Driver Station send this information to the dashboard.

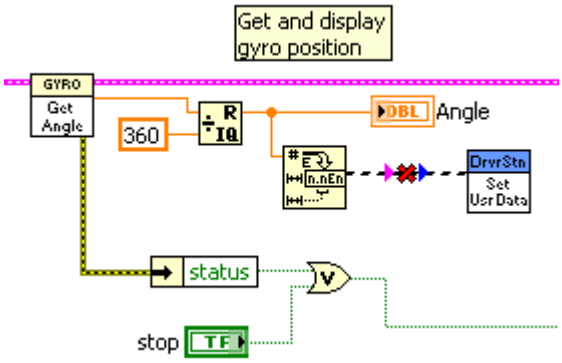
The communications configuration is already set up by the Start Communication VI. You'll notice that there is a type mismatch between the Angle output (a double) and User Data inputs (a U8 array). You can see the details of this mismatch by hovering the mouse of the red X on the wire with context help enabled (keyboard shortcut ctrl+H).



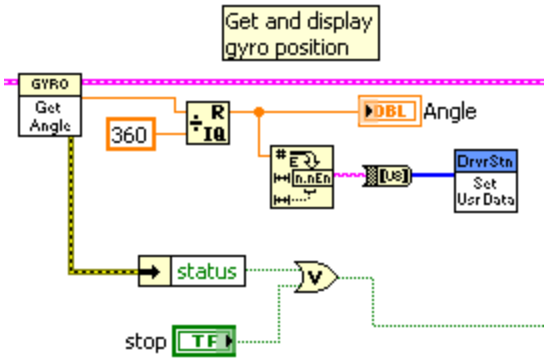
To remedy this, we will need to put our data in a form that can be interpreted by the communication VIs. There are two ways of doing this: a simple method for a single data type, and a more advanced method which enables you to send multiple types of data.

Packaging a Single Data Type

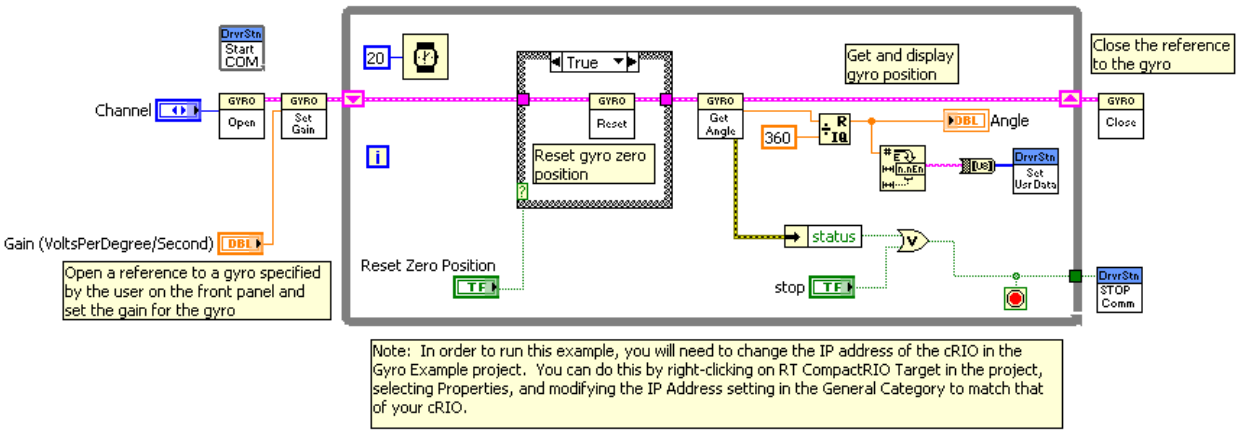
Add a **Number to Exponential String** function from the **Programming»String»String/Number Conversion** palette. Make some more room on the block diagram if you need it.



Next, add a **String to Byte Array** VI from the **Programming»String»String/Array/Path Conversion palette.**



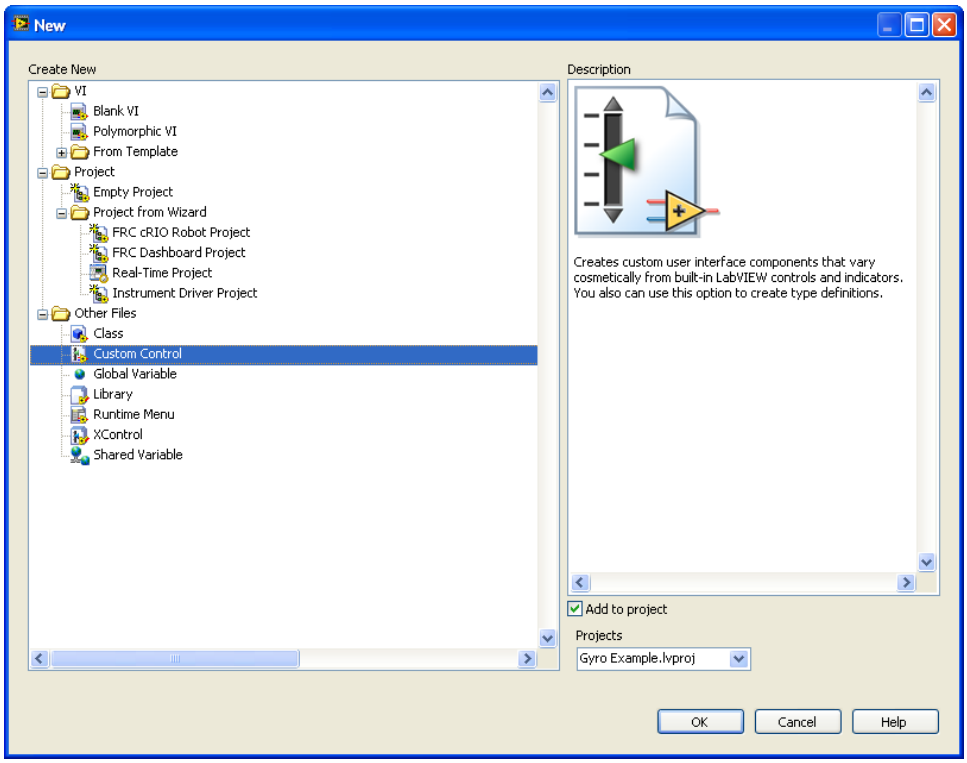
After wiring these VIs together as shown, we now have a VI that will send information to the *User Data* section of the dashboard.



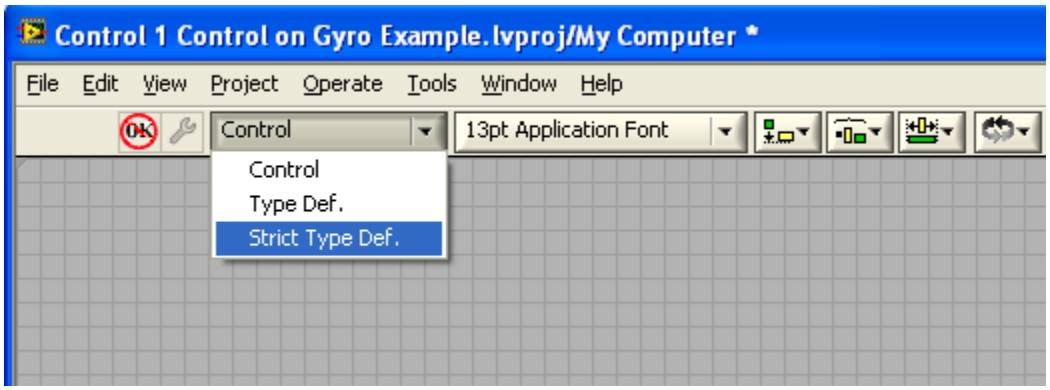
We had to convert the numeric first to a string then to a byte array to change the data returned from the gyro into a type that the dashboard communication VI accepts. Save the VI.

Packaging Multiple Data Types

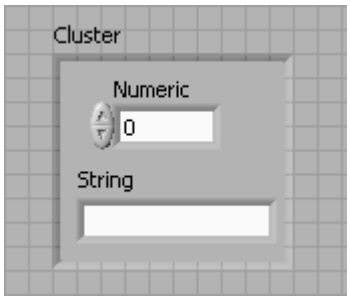
First we will need to specify what types and how much data we are going to send by creating a type definition. To create this definition, go to **File»New…** and then select **Custom Control** and click **OK**.



Click on **Control** and change the custom control to a **Strict Type Def**.



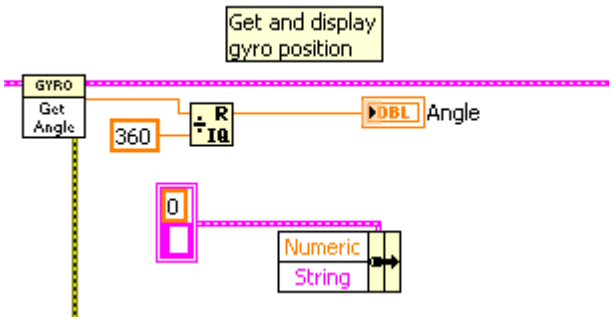
Navigate to the **Modern»Array, Matrix, & Cluster** palette and add a cluster to the Front Panel. We are going to send the gyro angle, a numeric, and a string, so add Controls of both of these data types to the Cluster. In your application, you can change the types you want to send here.



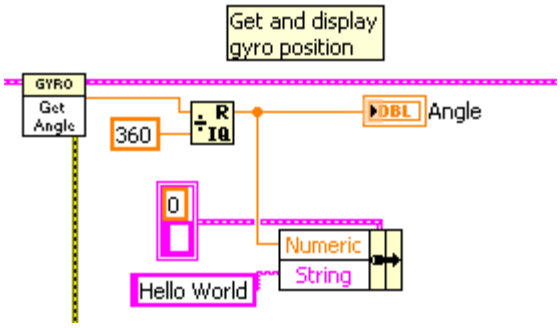
Save this control as *myTypeDef*.

Now we need to open the gyro example again. Instead of adding a Number to Exponential String VI, we are going to use the Flatten to String VI. Before we do that though, we are going to bundle the Angle numeric and string data into a Cluster. Add a **Bundle by Name** VI from the **Programming»Cluster, Class, & Variant** palette. Expand the **Bundle By Name** VI down to have two inputs visible.

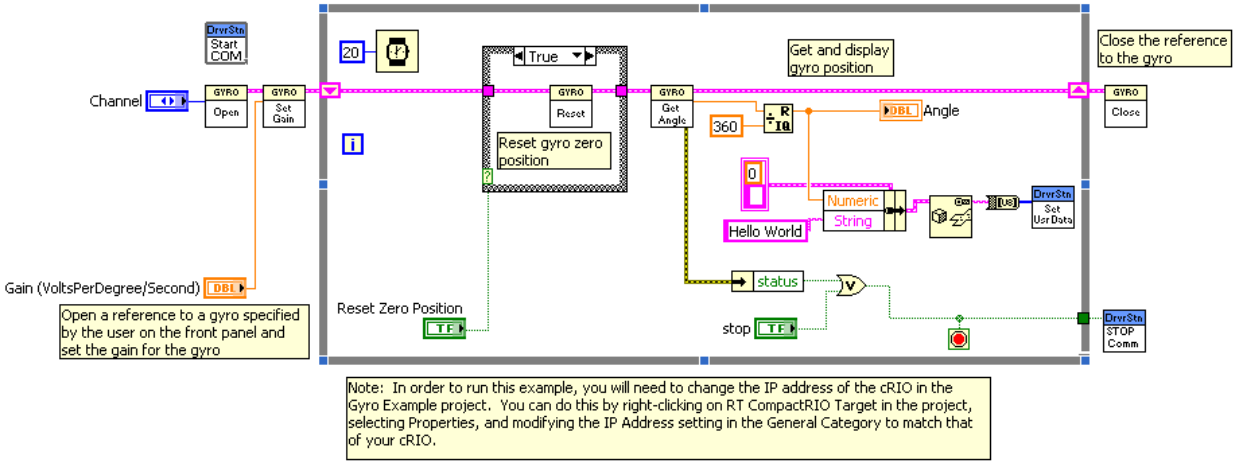
Next, right-click on the Block Diagram and select **Select A VI…** This will allow us to add the type definition we built to this block diagram. Find your *myTypeDef* control and click **OK**. Wire myTypeDef to the input cluster terminal. Click on the lower numeric in the Bundle by Name VI and change its type to a String. Your block diagram should now look like this.



Wire the Angle numeric to the numeric terminal of the **Bundle by Name** VI and a string constant to the string terminal. I used “Hello World” as my constant.



Add a **Flatten to String** VI from the **Mathematics»Numeric»Data Manipulation** palette as well as the **String to Byte Array** VI and **Set User Data** VI that we used in the single data type example.

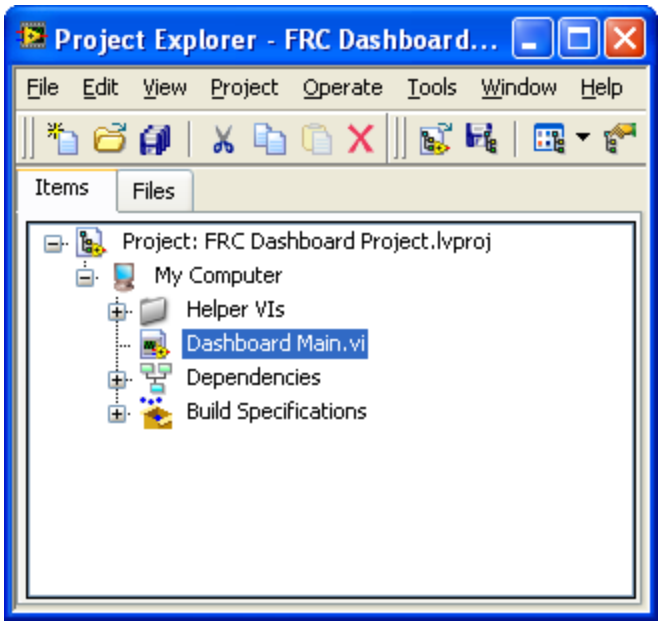


Save the completed VI.

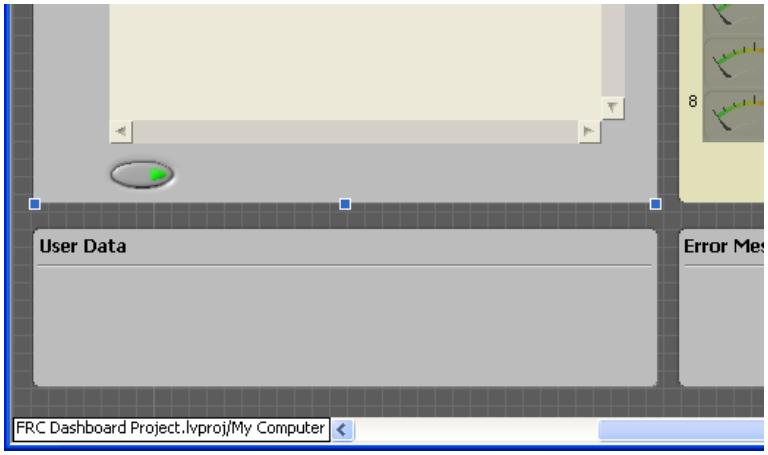
Receiving Data in the Dashboard for a Single Data Type

Since we have now modified the gyro example to send angle data to the dashboard, let’s look into how we can set up the Dashboard VI to receive and display the data.

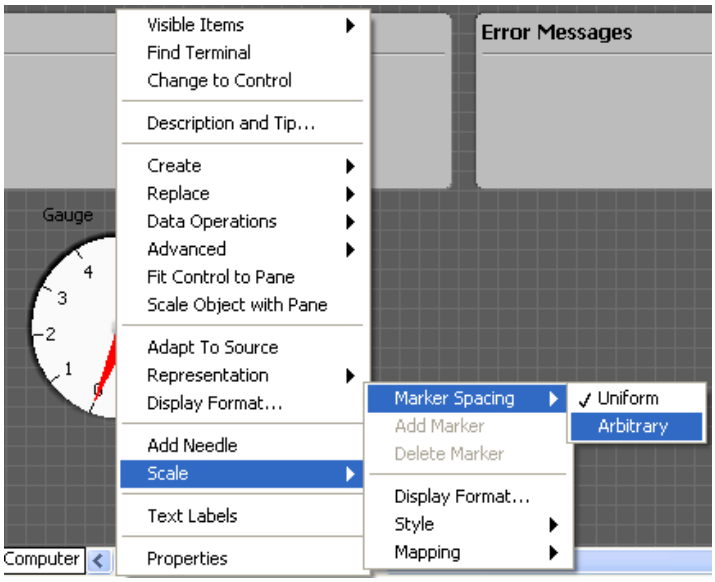
Open a new FRC Dashboard Project. To stay organized, it may be a good idea to save this project in the same directory as your modified gyro example project. For more general information about the basic and advanced framework projects, you can reference the Frameworks training module. The project will already have a Dashboard Main VI. Open that VI.



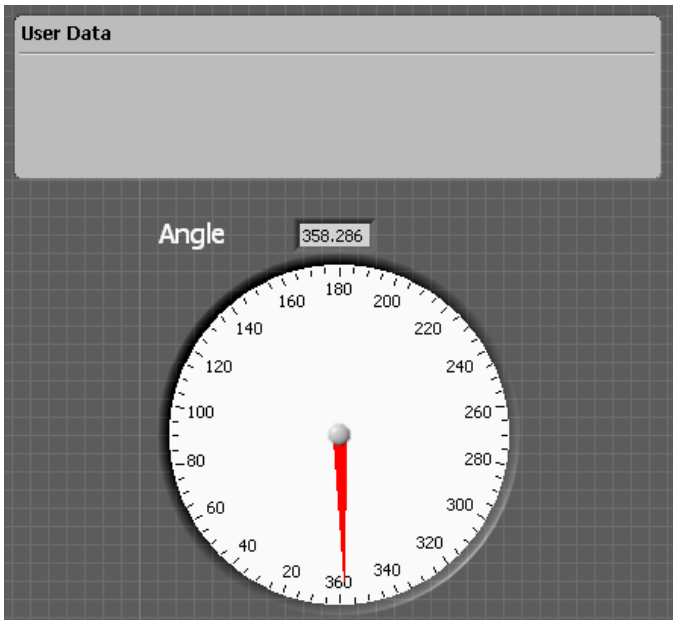
On the Front Panel, note the User Data display in the bottom left corner of the Dashboard. This is the where data we sent from the Gyro Example VI will be displayed.



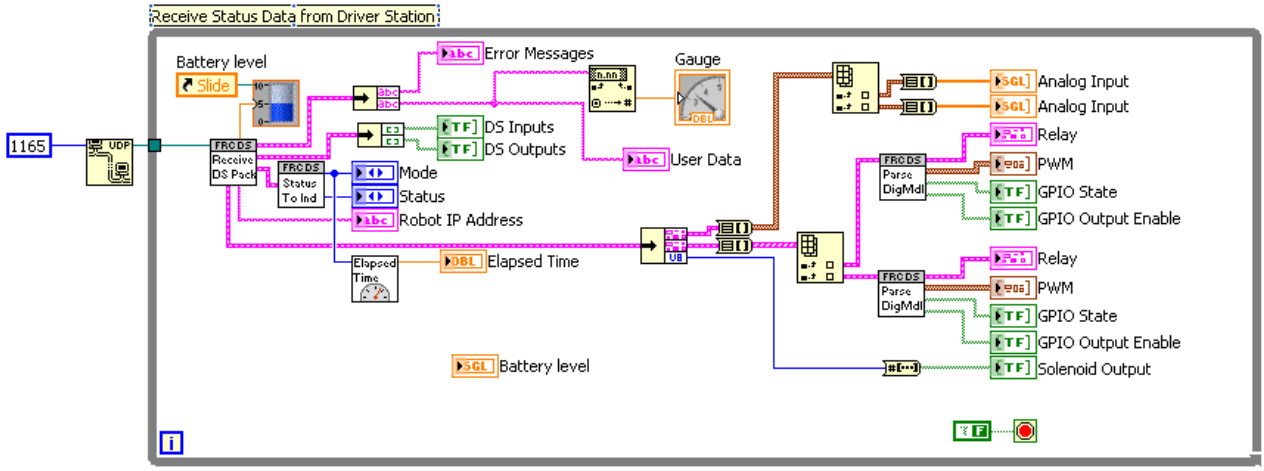
We will also add a gauge display to make reading the angle information easier. Make some room by expanding the Dashboard Front Panel. Then, add a Gauge indicator from the **Modern»Numeric** palette.



Adjust the maximum of the scale to **360** degrees. Next, align 360 with the 0 point. Adjust the size of the gauge and the appearance as you see fit. You can add a digital display and other features by right-clicking on the gauge and selecting **Visible Items**.

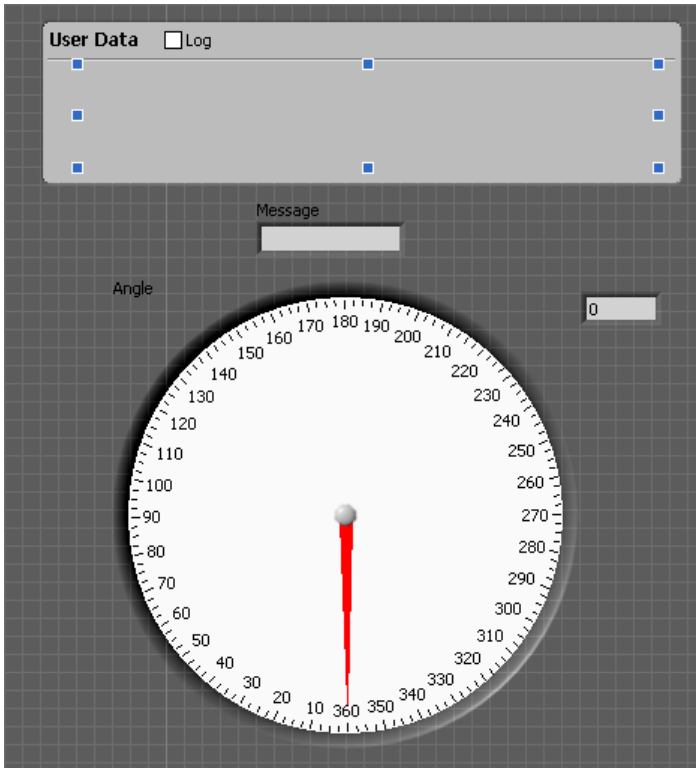


Go to the Block Diagram and move the Gauge indicator inside the lower while loop. Move the *User Data* indicator out of the way. Then add a **Fract/Exp String to Number** VI from the **Programming»String»String/Number** Conversion Palette. Wire these objects together as shown below.

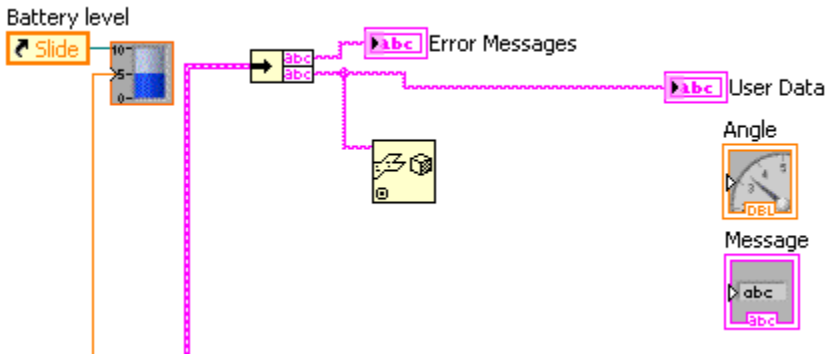


Receiving Data in the Dashboard for a Multiple Data Types

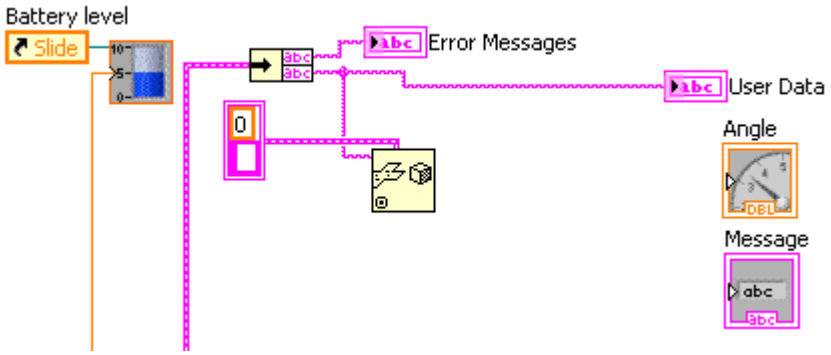
The process for receiving multiple data types is largely the same as for a single data type; the difference lies in the unpacking. As you did in the Single Data Type example, add a gauge indictor to display the angle, but here also add a string indicator to display the message you are sending from the cRIO FRC.



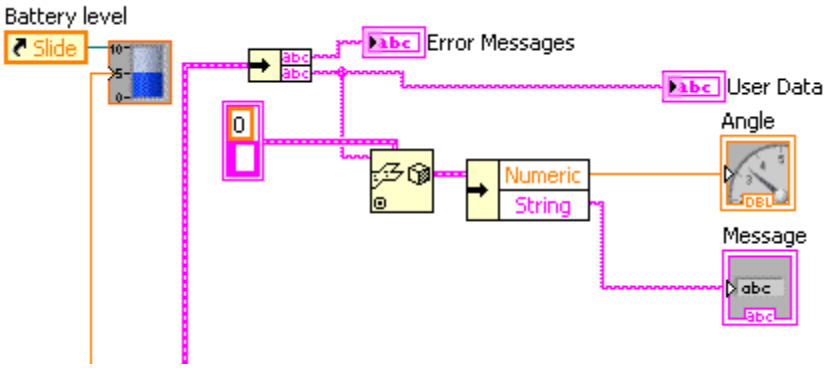
Go to the block diagram and add an **Unflatten from String** VI from the **Mathematics»Numeric»Data Manipulation** palette. Wire the binary string terminal to the *User Data* string. If necessary, make some more room inside the loop for your additional code.



Notice that the Unflatten from String VI has a *type* terminal. This is where we specify the type of data we have sent using the type definition we created earlier. Right-click on the block diagram, select **Select a VI…** and then choose your *myTypeDef* from earlier. Place it on the block diagram and wire it to the type terminal.



Now add an **Unbundle by Name** VI to split up the cluster you created previously. This VI is found in the **Programming»Class, Cluster, & Variant** palette. Expand the Unbundle by Name VI for two data types and select string as the second type. Wire the value terminal of the Unflatten from String VI to this Unbundle by Name VI. Wire up the Angle and Message indicators as well.



Running the Dash Board

Now that we have programmed the Dashboard and the Gyro Example to supply information to the Dashboard, let's explore how this works.

Open up the gyro example you modified earlier. Click the **Run** button which will deploy the VI to the cRIO. Next, start the Dashboard Main VI. Notice that as you change the angle on the gyro, the User Data information as well as the Gauge control on the Dashboard Main VI running on the laptop reflects these changes. If you are using multiple data types, those are updated and displayed as well.

Conclusion

By now you should have a good idea on setting up a dashboard test station for the I/O modules on the cRIO and transferring customized data using the *User String* output. The dashboard has many uses when it comes to project research and development. It's a fantastic tool for the following use cases:

1. Monitoring systems in different operation conditions
2. Logging data points for statistical analysis while in the research and development stage
3. Gaining a better understanding on how components interacts with each other

Congratulations! You've now learned how to return and add user data to the dashboard! The dashboard can also be used to get information from the joystick, sensors, actuators, etc… This information can be invaluable when testing out a new aspect of your robot.