Basics of Labview

Micheal Shumaker
shumakermikey@gmail.com
(269) 598-9526

Basic Programming in Labview

- Data Types
- Controls/Indicators
- Vi’s
- Numerical Operators
- Boolean Operators
- Comparisons
- Commenting

Data Types

In Labview, there are 4 different data types:

- Doubles(decimals)
- Integers(whole numbers)
- Strings(words, letters)
- Booleans(true/false)

Each type is represented by a different color wire:

Data Type	Scalar	1D Array	2D Array	Color
Numeric - Floating Point				Orange
Numeric - Integer				Blue
Boolean				Green
String				Pink

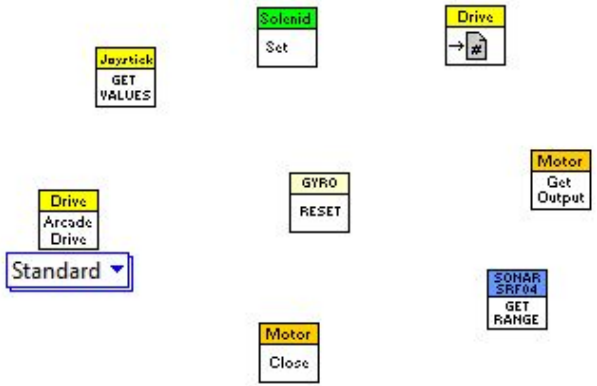
Controls vs. Indicators

Controls are usually inputs by the user. They can be constants, variables, or values inputted by a joystick. Indicators are outputs that come from joysticks, encoders, or motors.

Vi’s

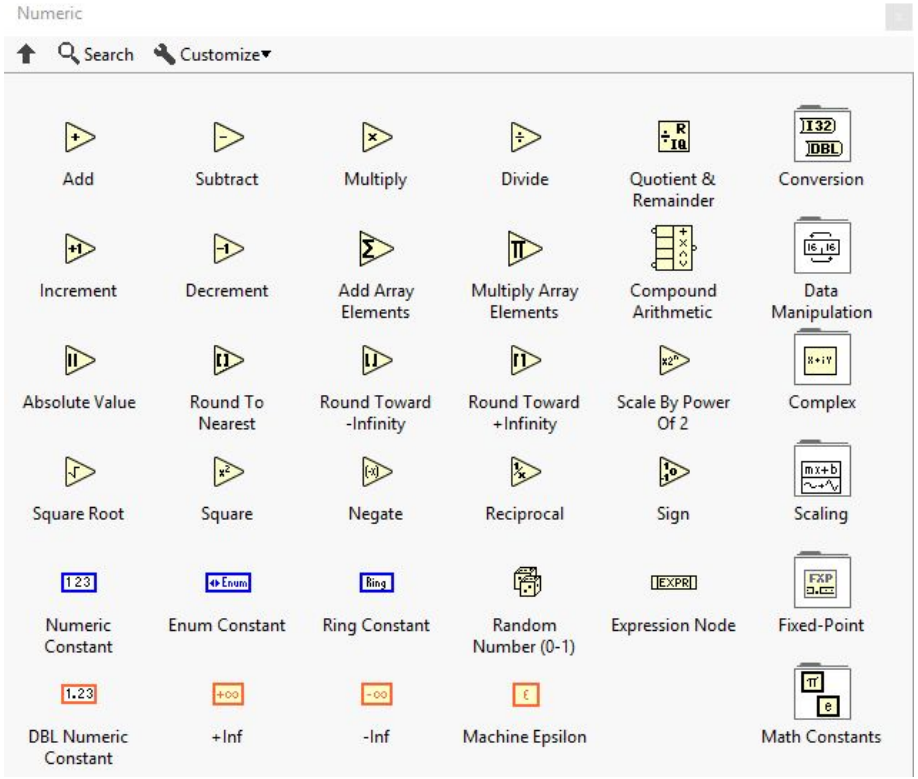
.vi is the filename for part of a labview project. It is also what any block is referred to as in the block diagram

Examples:



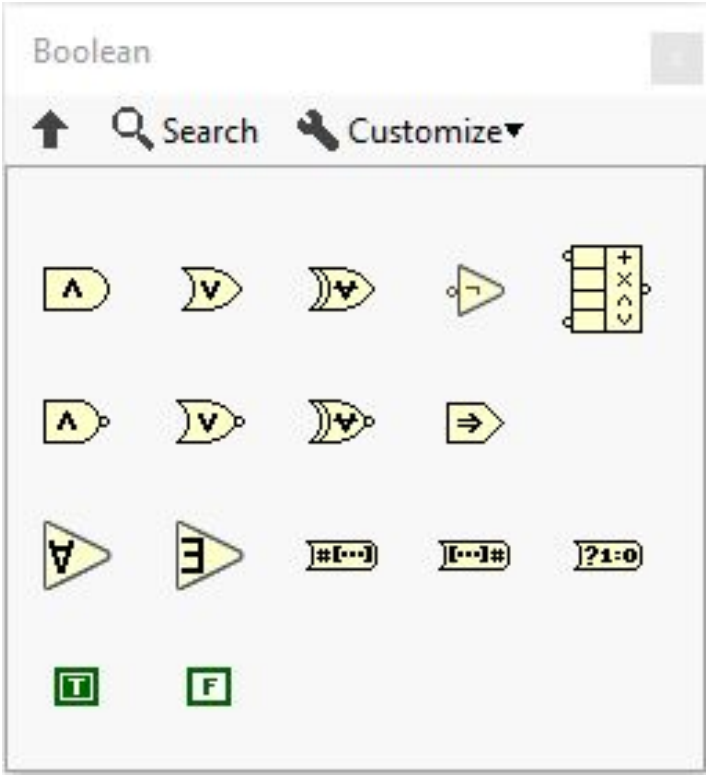
Numerical Operators

- Standard mathematical equations
- Can be used with joystick values



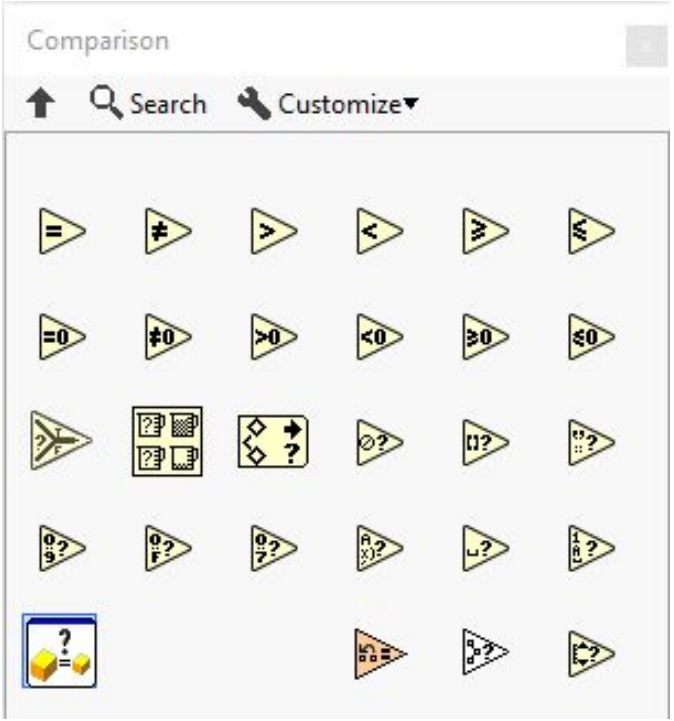
Boolean Operators

- Deals with boolean values
- Main operators
  - And- returns true if both values are true
  - Or- returns true if at least one value is true
  - Exclusive or- will return false if both values are true or both values are false
  - Not- returns false if the value is true and returns true if the value is false



Comparisons

- Operations that allow two numerical values to be evaluated, which leads to a certain action
- Takes a numerical input and a boolean output
- Comparisons include:
  - Greater than
  - Less than
  - Equal to
  - Greater than or equal to
  - Less than or equal to
  - Not equal



Comments

Commenting your code is very important to help you and other people reading your code understand what is happening. To create a comment in Labview, simply double click and type what you would like.

Questions?

Robot Code in Labview

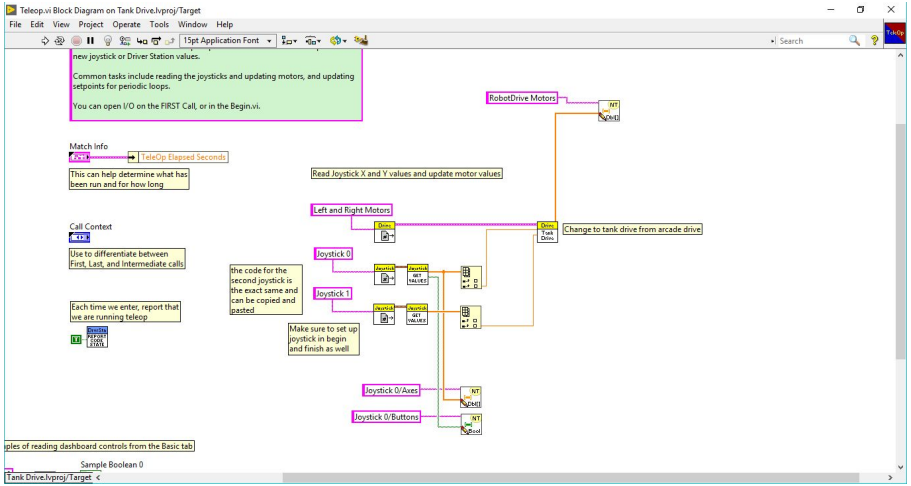
- Creating a New Robot Project
- Front Panel/Block Diagram
- Refnums
- Begin, Finish, Autonomous Independent, Teleop

Creating a New Robot Project

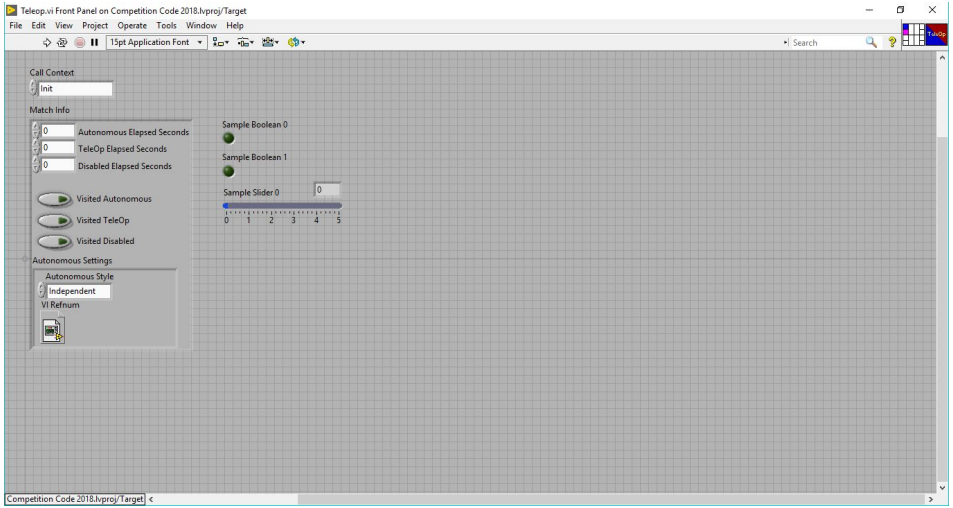
- When Labview is opened, click “create new RoboRio Project”
- Give your project a descriptive name
- Enter your team number
- Hit finish and wait for the project to load

Front Panel/Block Diagram

- Front Panel
  - Grey, grid-like window that opens up first
  - Not used very often
- Block Diagram
  - White
  - Used for most code in Labview



Front Panel



Block Diagram

Refnums

Refnums are a type of vi that allow data from one area of a project to be transferred to another. Things like the type of motor and PWM channel that are set in begin will need to be communicated to teleop in order for the code to know which motor it is controlling. The biggest thing with refnums is to make sure that wherever you use a refnum, the names are all the same.

Begin

- This is where any motors or sensors or actuators are created
- Use an open vi of whatever is being created
- In begin, always use set refnum
- Make sure that PWM channels or any other specifications are correct

Teleop

- This is where the main code is placed
- Teleop will loop every 20ms
- Make sure to use get refnum

Finish

- Finish will end any refnums that were opened in begin
- Each refnum needs to be in finish

Here we can take a break if anyone has any questions on how to do something specific in teleop code

Autonomous Independent

- This is where the code for the autonomous period of a match will be written
- You still need to make sure you use the refnums to get all of the things you will be using in autonomous
- Something new this year: the randomized colors on the scale
- There is a very easy way in the code to discern which side is a specific color
- https://wpilib.screenstepslive.com/s/currentCS/m/getting_started/l/826278-2018-game-data-details

Creating an Autonomous Program

- There are many different ways to write code that will successfully run on the robot
- I will go through a few ways and also show how the color selector works

Build Your Project

- In project window, click the build specifications drop down
- Right click FRC Robot Boot-up Deployment
- Select build
- After that is complete, connect to the robot radio
- Right click on the same thing and select run as startup