A debugger is used to control program flow and monitor variables in order to assist in debugging a program. This section will describe how to set up a debug session for an FRC robot program.

Set a Breakpoint

Double-click in the left margin of the source code window to set a breakpoint in your user program: A small red circle indicates the breakpoint has been set on the corresponding line.

Start Debugging

Press Ctrl+Shift+P and type WPILib or click on the WPILib Menu item to open the Command palette with WPILib pre-populated. Type Debug and select the Debug Robot Code menu item to start debugging. The code will download to the roboRIO and begin debugging.
The Debug tab

The Debug tab is accessed by clicking on the debug icon on the far left pane.
The Variables Pane

The Variables view shows the current values of variables. To see a variable that is not displayed, select the “Watch” pane and enter the variable name. This will show the variable’s value if it’s in scope. You may also want to click on the arrows next to a variable to expand the tree and show it's members. For example, expanding our Robot Demo variable ("this") shows our "myRobot" and "stick" variables.
Debugging a robot program

Watch Pane

The Watch Pane can be used to monitor specific variables or expressions while debugging. To add an expression, right-click and select Add Expression.

Call Stack

The Call Stack pane is used to display the current Call Stack of the running program. This can be used to monitor the current call hierarchy of your program while debugging.
Debugging a robot program

Breakpoint Pane

The Breakpoint Pane displays all of the current breakpoints. To temporarily disable a breakpoint without permanently removing it, click the checkbox.

Your Breakpoint

The program will start running and then pause at the breakpoint. From here you can use the panes on the left of the screen and the items in the Debug dropdown menu to monitor and control the execution of your program.

Debugging with Console

Another way to debug your program is to use System.out.println statements in your code and receive them using the RioLog in VSCode.