

Using NetworkTables with RoboRealm

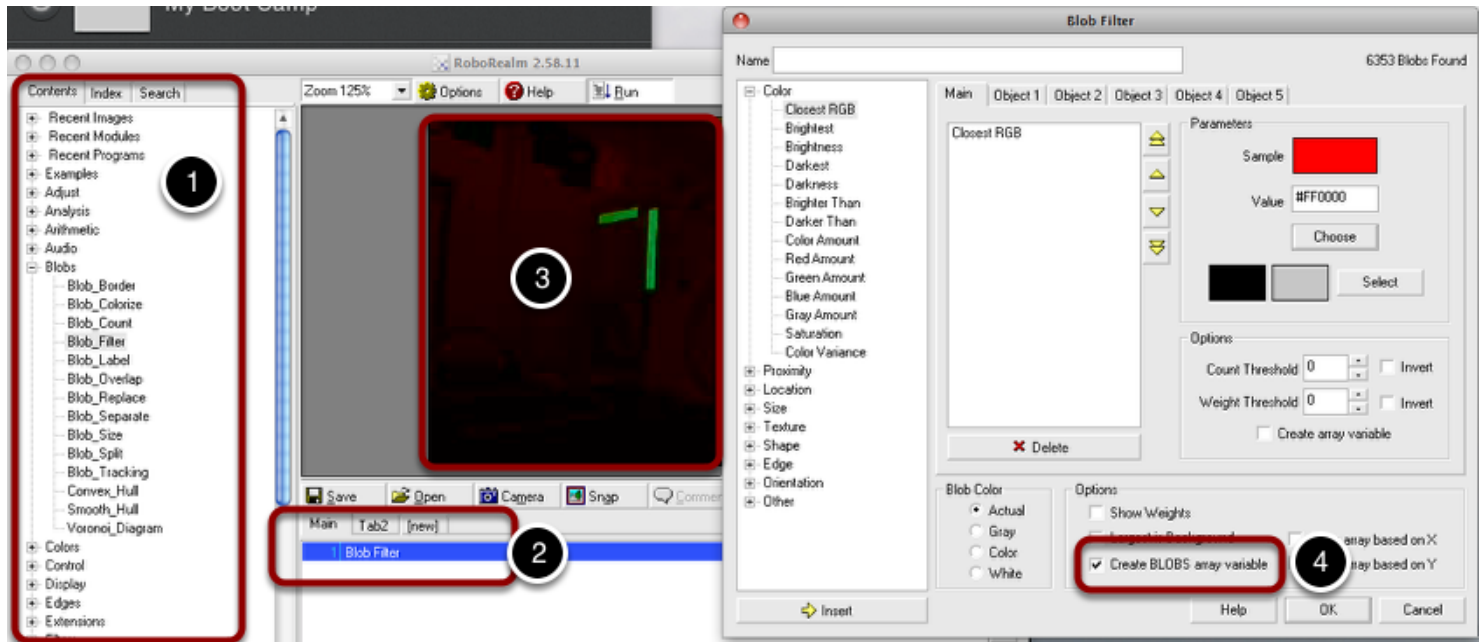
RoboRealm is a desktop vision application that you run on your driver station and can connect to a camera on your robot, do a set of vision processing steps that you define, then send the results back to the robot using NetworkTables. Using RoboRealm is easy since you don't need a robot to try it. In fact, you can write programs with just images that were taken such as those that come with any of the three language distributions. For Java and C++, installing the 2014 Sample Vision program will include a bunch of pictures taken with an Axis camera of the actual field that you can use to make sure your vision algorithm works.

There is a card included with your kit of parts that contains instructions for getting RoboRealm.

The idea is that you create a sequence of image processing steps with RoboRealm that create the results in variables. Then send those variables to the robot using NetworkTables. The robot gets the results and uses them to control the robot behavior such as aiming, driving to a target, setting shooter speed, etc.

Using NetworkTables with RoboRealm

Creating the RoboRealm program

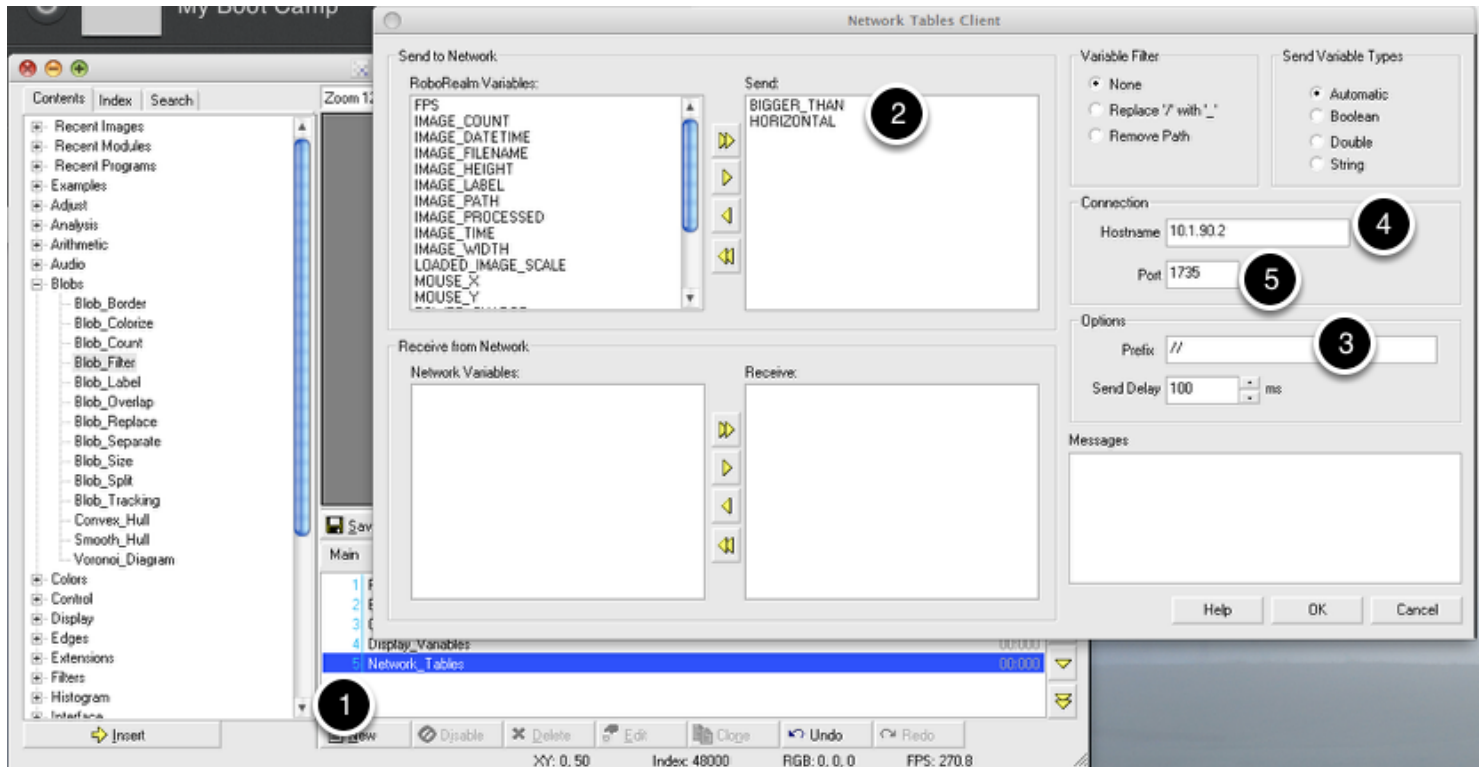


Create the RoboRealm program using the image processing elements shown in the "Contents" tab (1) on the left side of the interface. The program will appear in the bottom window (2). You can drag images from Windows Explorer into the Image window (3) to make sure that your algorithm works well and is repeatable for all the positions that the robot might be in.

By clicking on any of the steps in the program window, you can see the result of that processing step to make sure it's doing what you expect. Many of the steps can create variables with the results of that step. For example, the Blob Filter step shown has a checkbox to create an array of information for the detected blobs (4).

Using NetworkTables with RoboRealm

Interfacing with NetworkTables



Variables that are created in RoboRealm can be sent back to the robot using NetworkTables. To do that add a "Network_Tables" step (1) to your RoboRealm program and add the variables you defined to be sent to the robot (2). Set the "Prefix" to "/" (3) and the "Hostname" to the value of your robot network address (4) and the port to 1735 (5).

In this case it will send an array of values back to the robot for each of the detected blobs.

Using NetworkTables with RoboRealm

Retrieving the values on the robot

```
5  import edu.wpi.first.wpilibj.Victor;
6  import edu.wpi.first.wpilibj.networktables.NetworkTable;
7  import edu.wpi.first.wpilibj.networktables2.type.NumberArray;
8
9  public class RobotTemplate extends SimpleRobot {
10
11     NetworkTable table; 1
12     Victor claw;
13
14     public void robotInit() {
15         table = NetworkTable.getTable(""); 2
16         claw = new Victor(3);
17     }
18
19     public void setClaw(double value) {
20         claw.set(value);
21         Timer.delay(1);
22         claw.set(0);
23     }
24
25     public void operatorControl() {
26         boolean isHot = false;
27
28         while (isOperatorControl() && isEnabled()) {
29             NumberArray values = new NumberArray(); 3
30             try {
31                 table.retrieveValue("HORIZONTAL", values);
32                 4
33                 if (values.size() > 0 != isHot) {
34                     isHot = values.size() > 0;
35                     setClaw(isHot ? -1 : 1);
36                 }
37             } catch (Exception e) {
38                 e.printStackTrace();
39             }
40             Timer.delay(0.1);
41         }
42     }
43 }
```

Here's a Java program that retrieves those values on the robot and opens or closes a claw (just as a test) depending on whether there is at least one element in the array of values sent back. The steps to make this work are:

1. Declare the NetworkTable that will contain the values
2. Get an instance of the table and make sure everything is initialized
3. Create a new NumberArray object that will hold the results
4. Retrieve the array elements (values) for the NetworkTable variable called "HORIZONTAL" that was exported by RoboRealm

Now whenever the size of the array changes and contains at least one element the method setClaw(-1) will be called. Whenever the size changes and there are no elements in the array, setClaw(1) is called.