

Team 1912 Combustion Ignition Team Manual Microsoft Kinect



Written by Rachel Holladay and Edited by
Sam Holladay

Kinect Software

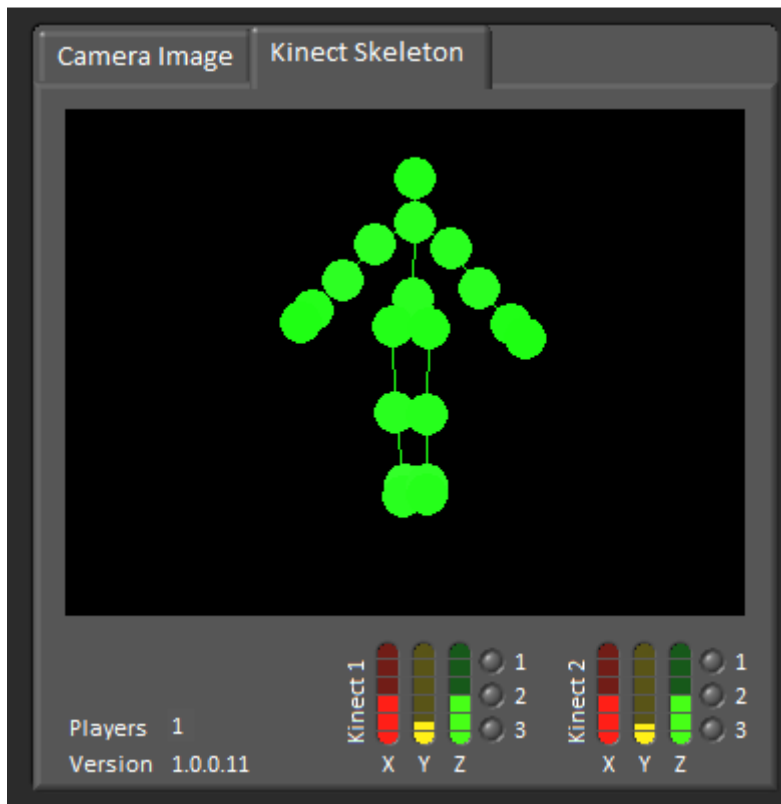
First and foremost, download the Kinect SDK server from Microsoft (<http://www.microsoft.com/en-us/kinectforwindows/download/>). Depending on your computer type, you must either chose the 32bit or 64bit. You also need the Visual C# 2010 Express software found at <http://www.microsoft.com/visualstudio/en-us/products/2010-editions/express>. With all FIRST software, there are updates, the first of which is (http://firstforge.wpi.edu/sf/frs/do/listReleases/projects.wpilib/frs.2012_frc_kinect_server). We anticipate that FIRST or Microsoft may put out more updates as the season progresses. Just like the cRIO image, the Kinect version will show up on the diagnostics tab of the Driver Station.

Kinect Hardware

To install the Kinect hardware simply plug one end to a regular power outlet and the other USB end to whatever computer you are planning to use. Woo hoo for the easy things!

Kinect Skeleton

The green Kinect skeleton is visible on the “Kinect Skeleton” tab on the dashboard. (Appropriately named, eh?) . What the Kinect does is convert a human body into a series of dots connected by lines (see image below). The dots represent the different joints in a human body. The dots show up as solid normally, but when you cross hands over, they become empty. Note that there will be no real difference in Kinect control if the human player is short, tall or average height because the distances are relative. Also, as seen below, there are little level indicators that show the XYZ value of the movements. There is a slight delay in the screen, which should be expected. This does however mean that the Kinect has trouble picking up rapid movement (so dancing to control your robot would probably NOT work). We would really suggest standing in front and playing around with the Kinect to get use to the movements. (Don't you wish Dean's Homework was something like this?)

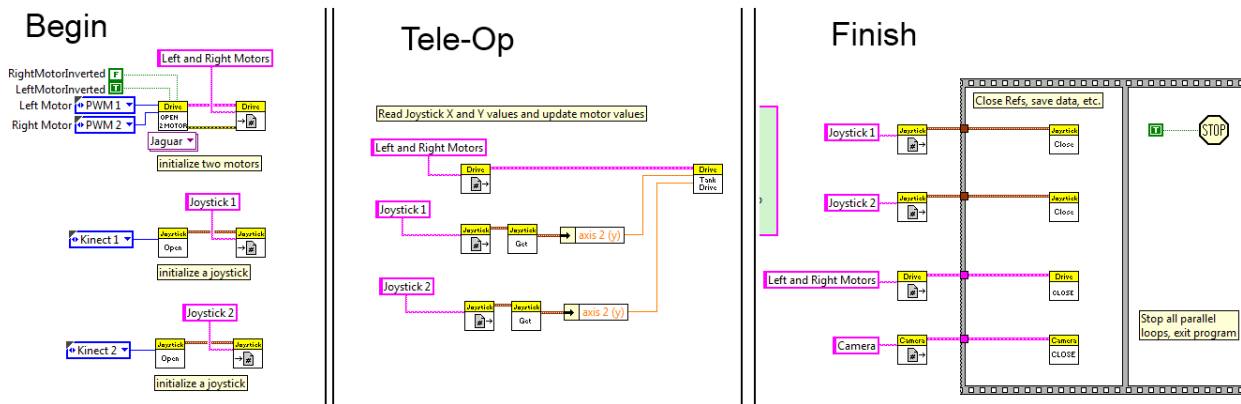


Kinect Environment

There is a reason why FIRST set up special Kinect zones on the field: they need their space. Only one human player can be in range at once. Otherwise the dots go nuts and become a jumbled mess. Basically, no other person (or robot for that matter) can be within about a six foot radius of the human player. You can tilt the Kinect up and down to put the human player in view. To set up, have the human player hold their hands in the air. Then move the Kinect up and down (as well as having the player take a step back and forward) in till the entire skeleton is in view.

Basic Drive with Kinect

The Kinect and code is set up to work easiest with Tank Drive. The code snippet is below.



As you can see the only difference between this and normal tank drive is the use of “Kinect 1” and “Kinect 2” for the joystick input instead of “USB 1” and “USB 2”. In a sense there is no Kinect 1 and 2 because it is only one physical device. (Don’t worry, you don’t need to buy two!) However we believe FIRST set it up this way so it would directly match the USBs for tank drive. Once you have this code built (and a tank drive robot with one motor per side) you can pretty much recreate the Youtube demonstration as seen at <http://www.youtube.com/watch?v=IYhxqsj70VY> (although perhaps you will look a little less dorky). Each arm corresponds to an axis. On joystick tank drive only the y-axis matters and similarly, with your arms, only the y-axis is relevant.

Kinect Buttons

On the Kinect, as explained above, your arms are preprogrammed to work as the axis. Likewise, there are certain preprogrammed movements that correspond to joystick buttons. The list (as it was in beta testing, we have not confirmed this during the build season so this is subject to change) is below. For example, instead of pressing button 2, you would tilt your head to the left to actuate the button. The programming for it would be exactly the same as if it were a regular joystick button.

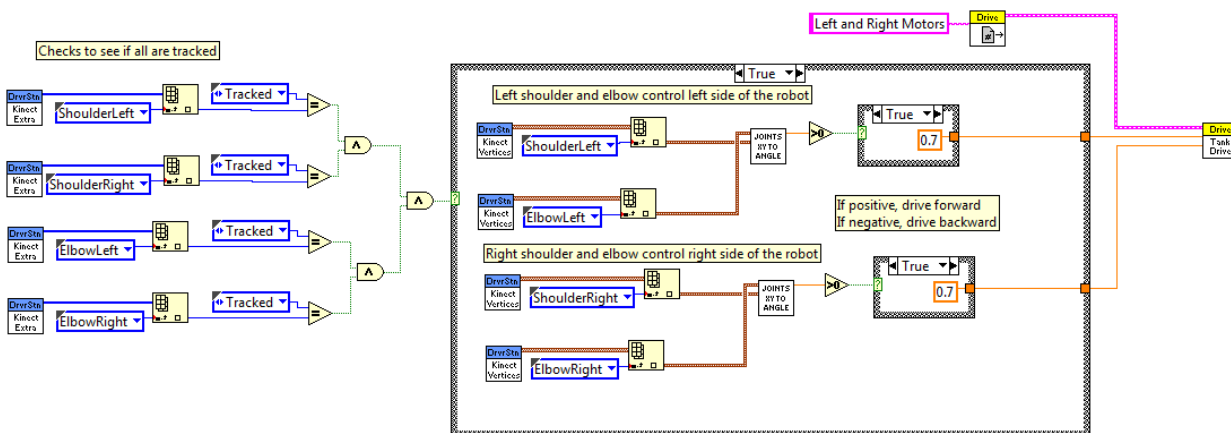
- Button 1 – Head to the right
- Button 2 – Head to the left
- Button 3 – Right Leg out to the right
- Button 4 – Left Leg out to the left
- Button 5 – Right Leg forward
- Button 6 – Right Leg back
- Button 7 – Left leg forward
- Button 8 – Left leg back

More Advanced Kinect Control

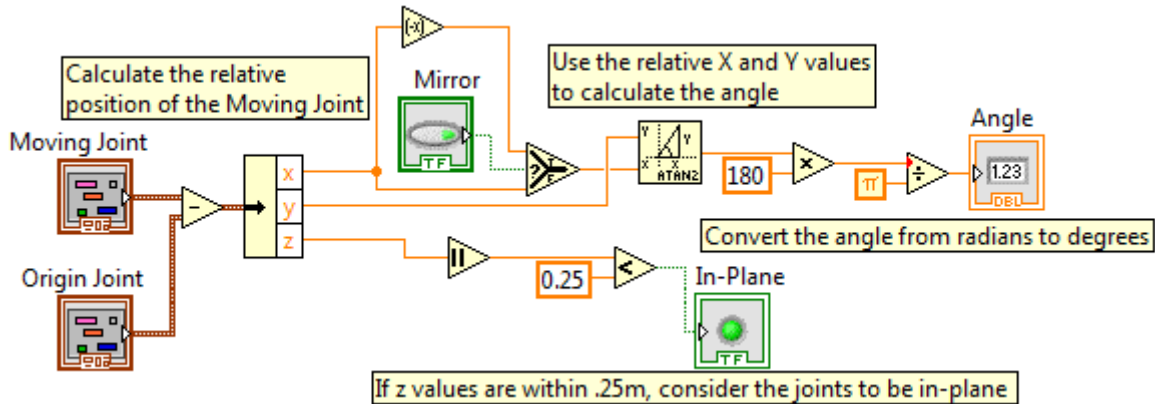
You are not limited to preprogrammed gesture. FIRST provides 20 joints, listed below, that are ripe for manipulation. We have not tested all of these and you may have to play around with each to figure out how well they work.

- HipCenter
- Spine
- ShoulderCenter
- ✓ Head
- ShoulderLeft
- ElbowLeft
- WristLeft
- HandLeft
- ShoulderRight
- ElbowRight
- WristRight
- HandRight
- HipLeft
- KneeLeft
- AnkleLeft
- FootLeft
- HipRight
- KneeRight
- AnkleRight
- FootRight

We created our own custom code that is shown in two screen shots below. Essentially it is still tank drive so the robot moves the same way it did in the Youtube video. Each arm still controls a side of the robot. However instead of holding your arm outstretched to drive a motor, you make a 90 degree angle with your elbow. Pointing your arm up drive the robot forward and pointing down moves it backward. True, within competition this is no great advantage. We worked on this more as an experiment to prove that you could do custom Kinect movements, not that you specifically should.



The real work behind this code was the “Joints XY to Angle” subVI created by the great (in our personal opinion) FRC Engineer Kevin O’Conner. We thank him for his ability to solve almost all our problems during Beta Testing.



Switching between Kinect in Autonomous and Joysticks in Tele-Op

Should you decide to use the Kinect in autonomous, you will need to move back to joysticks for tele-op. This is actually surprisingly simple. Open two Joystick references, one for Kinect 1 and one for Kinect 2. Then open separate Joystick references for USB 1, 2 etc. The two don’t actually conflict even though the Kinect plugs into a USB port. As far as programming, just make sure you are opening the correct reference. And remember that if it’s in autonomous, it must be in a loop.

General Notes

For now, this concludes our FRC introduction to the Kinect. It is entirely unofficial and made only on our experiences. I’m sure FIRST will not endorse or official-ize this, so do not use it as a definite source. Please pose any questions to Team1912Combustion@gmail.com as this will go straight to our Controls Captain and Beta Test Lead. Should it be required this document might be updated with more information or clarification.