*How to set up I/O on an ABB Robot with an IRC5 Controller Box*

Phillip Dupree
Mechatronics Lord

So you have an IRC5 robot controller and you want to set up digital inputs/outputs, do you? You poor bastard. But whether you're turning on and off a relay or a robotic end effector, or trying to read inputs from a similar device, this is necessary. Unfortunately it's terribly documented and can take forever to figure out. This guide should hopefully streamline the process, saving you a least some of the pain and torment I had to go through. I'm setting up a digital output in this guide, but inputs should be fairly straightforward after working through this process.

First, watch this youtube video (https://www.youtube.com/watch?v=JSdcVxgR6hg) entitled "How to configure I/O for a robot controller system in RobotStudio" by Henrik Berlin. Then watch it again. I wish I were joking, but after watching it about six times it should start to make some sense. Fun Fact! No, your headphones aren't broken. All the things that would make it infinitely more understandable - audio walking you through the steps, text overlay describing where he's clicking, or a textual explanation in the description - are all missing. Here's what's happening.

1. Henrik is opening RobotStudio, going to "Solution With Station and Robot Controller" and selecting his robot type (not in the video).
2. He's clicking the "Controller" tab at the top.
3. On the left hand toolbar, he's looking under IRB120_3kg_.5 (his robot, yours may be different. For the duration of the guide I'll refer to this as IRBX) and going to Configuration > I/O. After clicking I/O we see a menu to the right called Type with a bunch of stuff under it, including Signal, Unit, and Unit Type.
4. He's clicking on Unit and adding a new Unit (we'll get back to this).
5. Next he goes to Signal. He adds a new DigitalOutput or DigitalInput to the new Unit he just created.
6. He then right clicks on IRBX and "Warm Restarts" his controller.
7. Then he goes to IRBX > I/O > Virtual > MyUnit. MyUnit is the new Unit he created in Step 4. Upon clicking MyUnit, he sees a window to the right with the DigitalInput and DigitalOutput he created in Step 5.
8. He tries to change the value of the DigitalOutput and gets an error. This is to demonstrate that when creating the Signal, he left the Access Level at Default. He goes back to change it to All. Now he can toggle the value of his DigitalOutput between 0 and 1.

Now what Henrik did in that video is very similar to what we're going to do, the only real difference being that he configured a Virtual device (good for programming offline) and I'm going to tell you have to configure an honest-to-god DSQC 652 I/O board found inside your IRC5 (maybe). But, once you parse just what on Earth was going on, he has shown you the main steps.

- Create an I/O Unit in RobotStudio.

- Create Signals (whether they are DigitalOutput or DigitalInput) and attach them to your newly created I/O Unit.
- Toggle your digital outputs to your heart's content by going to IRXX > IO System > BUS (in our case, DeviceNet) > MyUnit (or whatever it is you named your unit).

So after days of trawling various internet forums, old comments from years ago, youtube videos, and product manuals, this is what I learned I have to do. This is my robotics lab's setup:

- An ABB 6700 205/2.8M Robot (not important)
- An IRC5 Controller (not the compact version) (very important)
- A DSQC 652 I/O Module installed on the inside cover of the IRC5 (very important)

The DSQC 652 is the I/O Module. Unlike your typical microcomputer its digital high and low are 0V and 24V, not 5V or 3.3V. Know this before you fry something. I have attached a picture of it. It does not come standard, so make sure your IRC5 has it. If not, you may need to order and install it.

Now let’s get started.

## The Software Side

Here is what we have to do in software. I am using RobotStudio 6.01.01.

0. Before you do anything, turn on your IRC5 and connect it to your computer with an ethernet cable.
1. Open RobotStudio. Go to "Solution with Station and Robot Controller". Select your robot type.
2. Once everything's loaded, click Request Write Access. Go over to your FlexPendant and select Yes.
3. Then go to Add Controller. You may have to type in its IP address under "ADD Controller > Add Controller.." if the "One Click Connect" doesn't work (it never does for me).
4. Go to the "Controller" tab at the very top.
5. Go to IRBX > Configuration > I/O > Unit Type. You should see a Unit Type "d652" on Bus "DeviceNet". Yes, this is your DSQC 652 I/O module which is connected to DeviceNet, a Bus used for I/O communication. See it there? Good.
6. Now we're going to create our DSQC 652 in software. Go to Unit. Right click and select "New Unit".
    a. Name: Call it whatever you like. I'll call mine Unit_IO.
    b. Type of Unit: D652.
    c. Connected to Bus: DeviceNet (at first I thought it was on Local, ‘cause it's in the box right? It's not. The Bus in DeviceNet).
    d. Address: 10. (Note, its address may not be 10. This is apparently the Bus address they are usually configured at. You can figure this out by looking at NA0 through NA5 on X5 on the DSQC652 (pictured in Appendix). Apparently the bus address is the binary value of NA0 through NA5. You may see a grey plastic thing with some metal tabs on it

grounding some pins of NA0-NA5 and powering others. When I tried setting it to 10, I stopped getting "Incorrect BUS Address/Hardware not found" errors from the IRC5.)

7. Now go to Signal.
8. Right click and select New Signal.
    a. Name: Whatever. I'm calling mine Output1.
    b. Type of Signal: Digital Output
    c. Assigned to Unit: Unit_IO (the one we just created)
    d. Unit Mapping: 0
    e. Access Level: All
9. Click OK. You should get a message saying changes won't take place until after restart. Right click on IRBX and select “Warm Restart”.
9. Now you should be able to go to IRBX > I/O System > DeviceNet > Unit_IO and see your newly created signal Output1. Double click it to toggle its value.

Okay, sweet. We're done with the software side of things. Rest up because the fight is far from over.

## The Hardware Side

Sorry, it's a pain in the ass too. But not quite as big a pain in the ass.

1. Open up your IRC5. You should see your DSQC 652 on the inside of the door (pictured in the Appendix). To the right of it you should see a grey terminal thingy labeled XT31.
2. Fun fact! The internal logic of the DSQC 652 is powered, but the digital outputs are NOT. You have to power them externally by connecting 24V and 0V (GND) to the rightmost pins on the X1 or X2 rail of the DSQC 652 (also pictured in Appendix). Can this come from a car battery? Technically, yes, but you're better off using the XT31, that funny WAGO connector block designed for powering I/O boards. Pins 1 and 5 on the XT31 are 24V and GND/0V respectively. In some way or another, hook these up to 24V and 0V on the X1 or X2.
3. Another fun fact! By now you're probably realizing that every time I say "fun fact" I actually mean "days of fury, frustration, wailing and gnashing of teeth". Remember when we were creating our Signal and we set "Unit Mapping" to 0? A zero in Unit Mapping corresponds to the pin 1 on the X1. A one in Unit Mapping corresponds to pin 2, etc. Real intuitive, I know.
4. Is your XT31 hooked up to your DSQC 652? Is your circuit, whatever it may be, hooked up to X1 pin 1 and 0V/GND? Good. In software, whether on RobotStudio or on your FlexPendant, toggle the value of your output. Your circuit should behave accordingly as the Voltage goes from 0V to 24V.

With this handy guide, I hope this process will be less frustrating for you, dear reader, than it was for me. Thanks for everyone on the ABB Forum for helping me out.

## Appendix

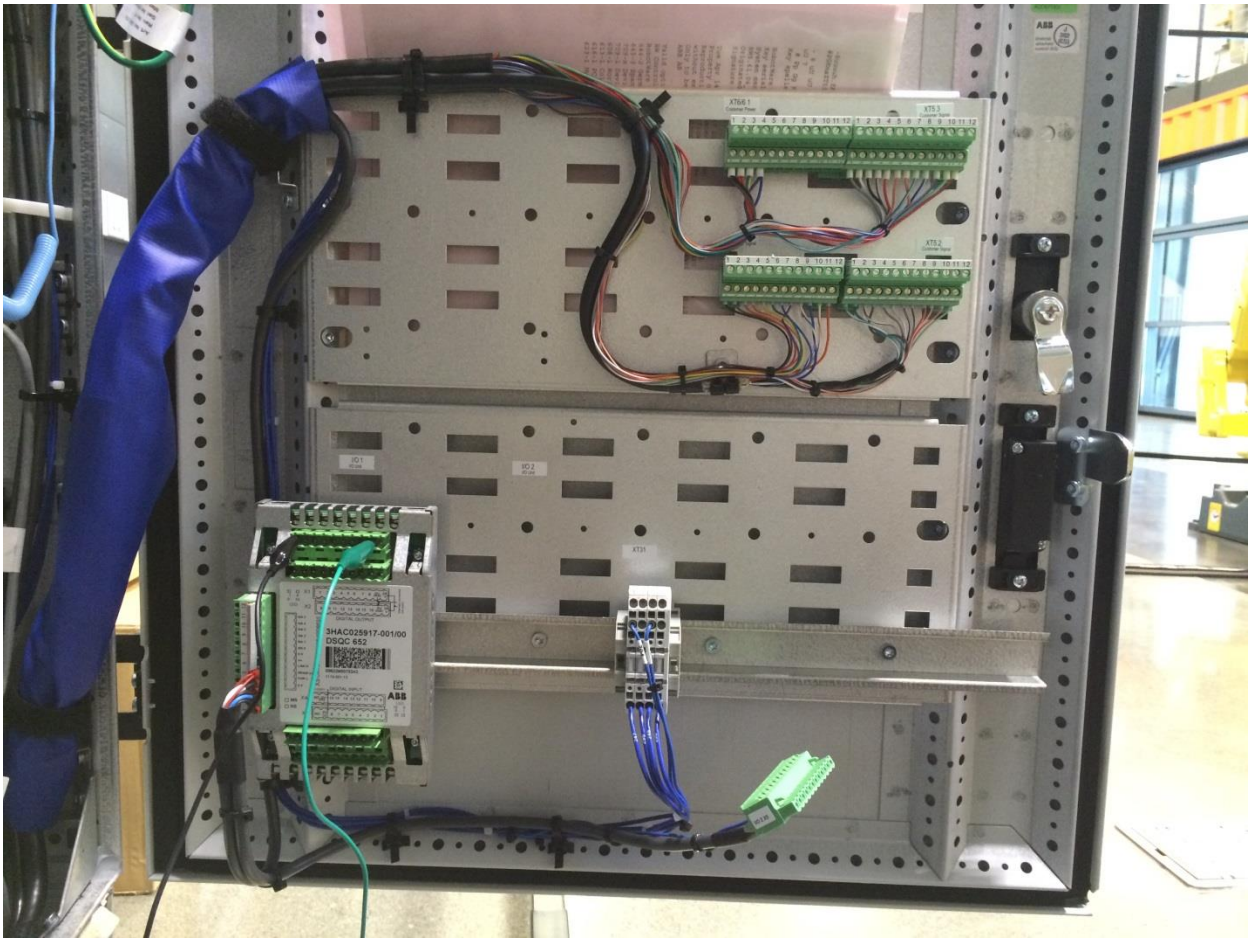


**Figure 1: The DSQC 652 (bottom left) and the XT31 (bottom right)**

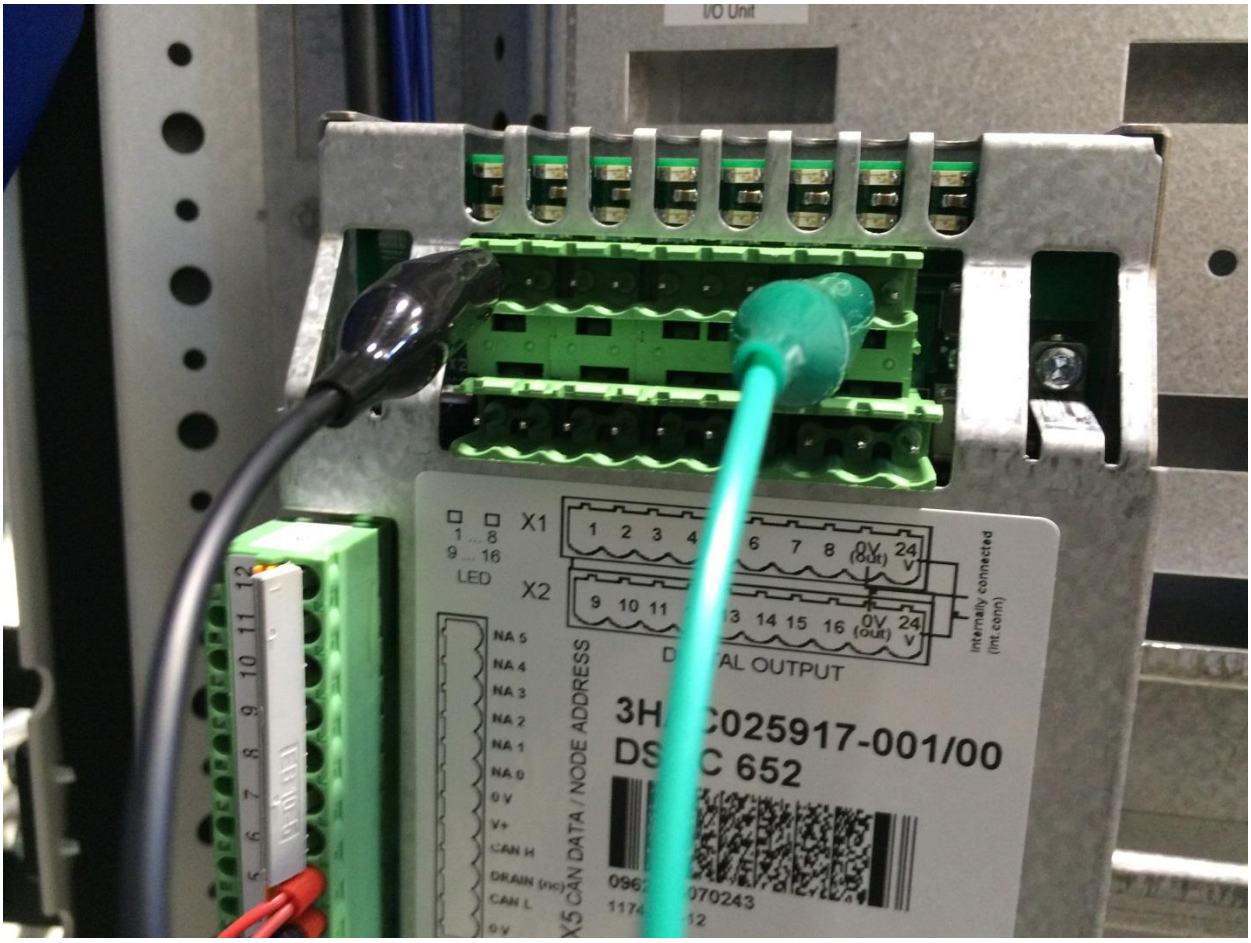


**Figure 2: Close up of the DSQC 652. You can see NA0 - NA5, which ostensibly set the value of the Bus Address. You can also see pins 1 - 8 and 0V, 24V on the X1 rail and pins 9-16 and 0V, 24V on the X2 rail.**