Operator Interface Control Board User's Guide

Created by: Ryan Nazaretian

Version 1.1

Changelog

Version	Date	Changes
1.0	2017-04-15	Initial Version
1.1	2017-10-19	Added information on robot integration

Contents

1 About

The Operator Interface Control Board (OICB) is both a hardware and software solution used to give you custom I/O at your robot's operator interface without any additional breakout boards, complex wiring, or setup.

The OICB v1.1 has 16 LED outputs @ 20mA a piece, 16 Switch Inputs, 16 Analog Inputs, and 11 PWM Outputs. The pinout for the headers are like what is used on your FRC robot using what is typically called PWM cables and runs off a single USB A-B cable.

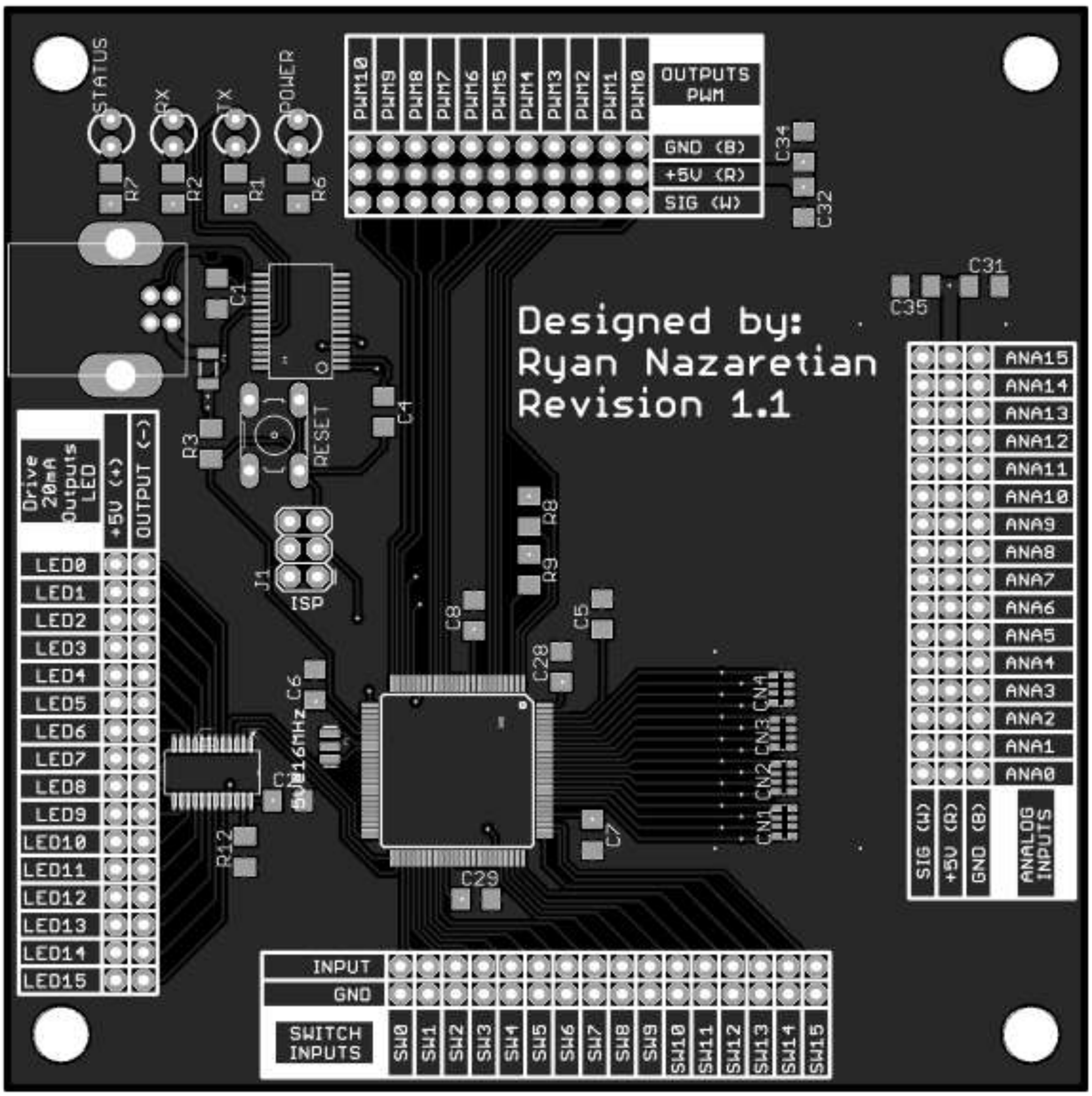


Figure 1. OICB v1.1

The goal is to give a team this board and the software that is familiar with them. The OICB takes the I/O from the interface board and updates the robot's Network Table.

2 Installation

1. Download the latest version from the [Garnet Squadron Github](#).
2. Run the installer on your PC.
3. You will see the OICB icon on your desktop.

Figure 2. Desktop Icon

3 First Run

Double click on the desktop icon. On the first run, the program will start up in a with a simulator connected and try to connect to a simulated robot on *localhost*.

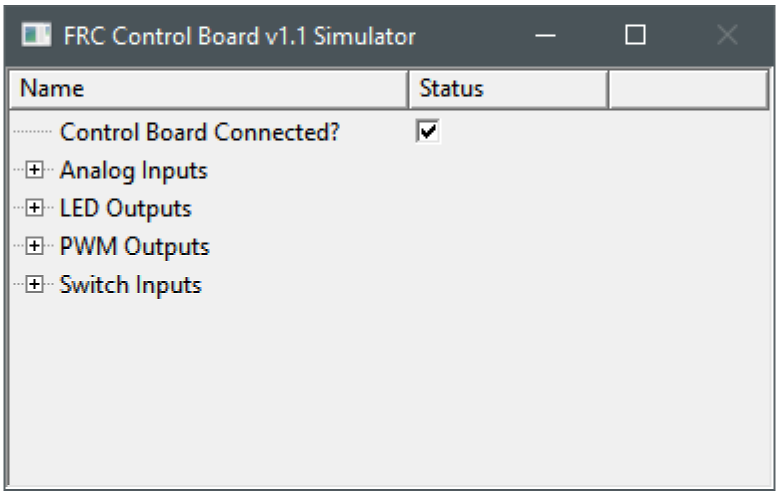


Figure 3. FRC Control Board v1.1 Simulator

You will also notice a new icon in your notification area shown in Figure 4. You can double click this to bring up the user interface, or you can right click to show a context menu, then click "Show Data" as shown in Figure 5.

Figure 4. OICB in Notification Area

Figure 5. OICB notification area context menu

After showing the window, you should see the main interface.

4 GUI Overview

4.1 Main Window

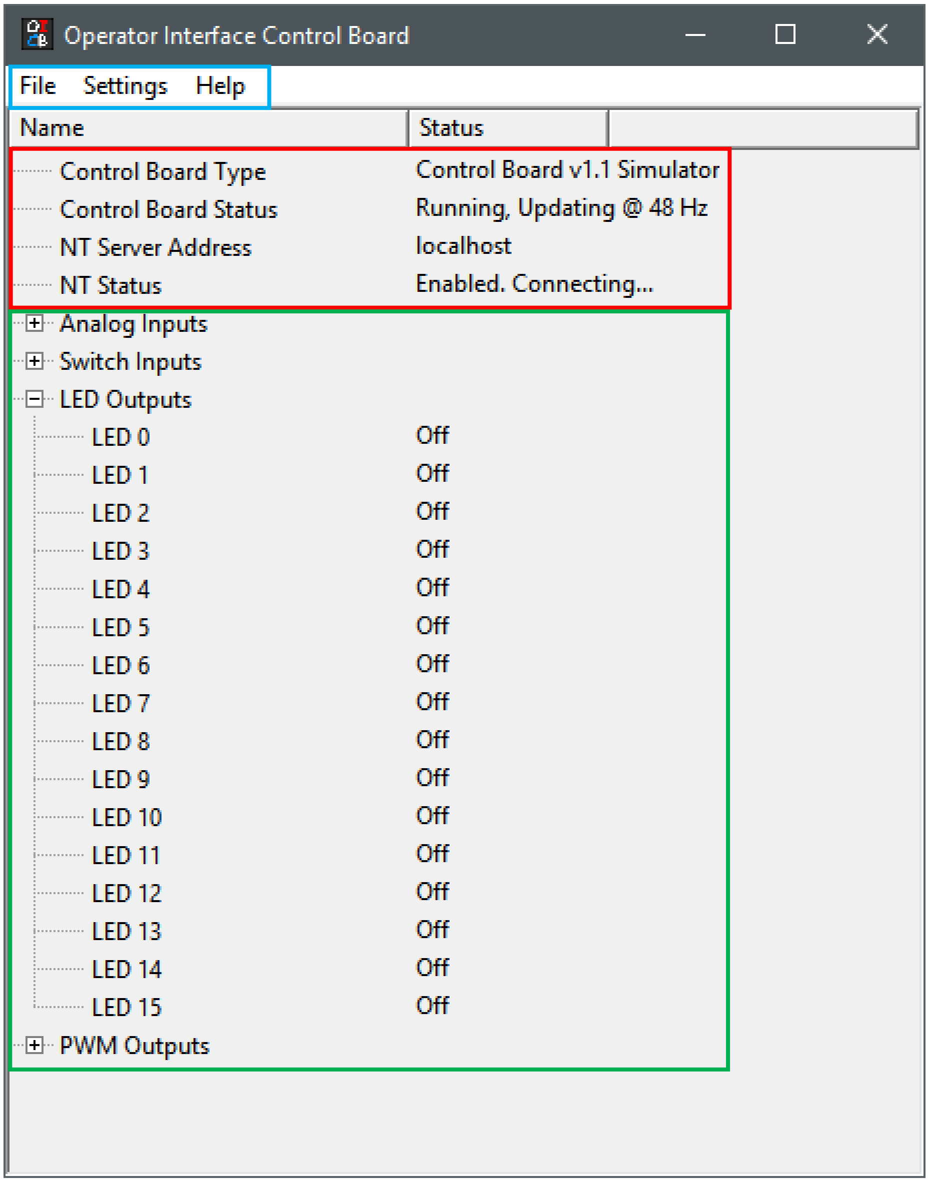


Figure 6. OICB main window

The main window shown above in Figure 6 is your way of interacting with the board from the computer. It has three main parts, from top to bottom.

1. Highlighted in blue is the menu bar. Its uses will be covered in section 4.2.
2. Highlighted in red is the OICB status.
3. Highlighted in green is the current status of the control board's inputs and outputs.

4.1.1 OICB Statuses

a. "Control Board Type" shows the currently set control board type.

b. "Control Board Status" shows the status of the interface. If it is running, it will also show the rate of update in hertz. The different statuses include:

Status	Description	If successful, go to	If failure, go to
Initializing	This is the initial state of the Finite State Machine. It immediately transitions to the next state.	Control board disconnected	N/A
Control board disconnected	The software checks the available COM ports and opens the COM port that matches the USB PID and VID described for the selected control board. If successful, it will open the COM port and go to the next state.	Control board connected	Control board disconnected
Control board connected	Logs that the control board was connected (Logging Level >= INFO).	Checking connection	N/A
Checking connection	Checks that the COM port is valid and opened.	Resetting control board	Control board disconnected
Resetting control board	The software pulses the reset signal (DTR) for 50ms and waits for the control board's firmware to send the welcome message.	Running	Control board disconnected
Running	The software has established a valid connection	Running	Control board disconnected
Stopped	The software has requested the interface to stop. It must be reinitialized. This state is encountered when the software is switching interfaces or the software is shutting down.	Stopped	Stopped

Figure 7. Control board status table

c. "NT Server Address" shows the current Network Table server address. Typically, this will be set to the address of your RoboRIO. You have full flexibility to change this in the settings.

d. "NT Status" shows the Network Table Abstraction Layer (NTAL) status. The different statuses include:

Status	Description	If successful, go to	If failure, go to
Initializing	This is the initial state.	Enabling…	N/A
Enabling…	The software is setting the NT Client parameters.	Enabled. Connecting...	Error
Enabled. Connecting...	The IP address of the NT Server is set and the NT Client is attempting to connect to the server.	Connected	Enabled. Connecting...
Connected	The software is currently connected to the NT Server.	Connected	Enabled. Connecting...
Disabling…	The software has requested that the network table be disabled. This can has three causes: 1. Shutting down 2. Test mode enabled 3. The NT Server address is changing.	Disabled	N/A
Disabled	The NTAL will stay in this state until reenabled. The only control the user has over this is to disable Test mode.	Disabled	N/A
Error	The software encountered an error and will return to the previous state once the error is cleared.	Previous State	Error

Figure 8. NT status table

4.2 Menu

4.2.1 File

The file menu has two options:

1. Hide – Hides the main window. Does not quit. This has the same effect as clicking the X in the top right.
2. Quit – Quits the application completely.

4.2.2 Settings

4.2.2.1 Test Mode

Test mode gives you the ability to test your control board without having the robot present. You can toggle LEDs or set PWM outputs. You will still have the ability to view the analog and switch inputs. Test mode will disable the NT Server connection.

4.2.2.2 Control Board Types

Clicking on "Set Control Board Type" will present you with a selector window shown in Figure 9 to select your control board. If you wish to cancel, click "Close".

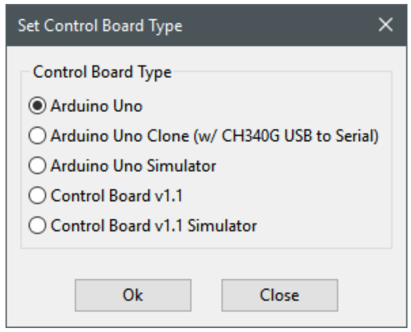


Figure 9. Set Control Board Type window

To accept your new selection, click “Ok”. At this time, there are five different control board types:

4.2.2.2.1 Arduino Uno

You can use an Arduino Uno for your control board interface. This is primarily done to provide you with the ability to fully evaluate the software before building your own board. Here’s the pin map list:

Pin	Function
A0	ANA 0
A1	ANA 1
A2	ANA 2
A3	ANA 3
A4	ANA 4
A5	ANA 5
0	Serial RX – Do not use!
1	Serial TX – Do not use!
2	SW 0
3	SW 1
4	SW 2
5	SW 3
6	SW 4
7	SW 5
8	LED 0
9	LED 1
10	LED 2
11	PWM 0
12	LED 3
13	Status LED – Do not use!

Figure 10. Arduino Uno pinout

4.2.2.2.2 Arduino Uno Clone (w/ CH340G USB to Serial)

This has the same pinout, but the USB chip’s PID and VID were updated to allow the software to connect to it.

4.2.2.2.3 Arduino Uno Simulator

This gives you the ability to simulate the use of an Arduino Uno and give you the same number of I/O available in the Arduino Uno firmware. For more simulator information, please refer to section 5.

4.2.2.2.4 Control Board v1.1

This provides the OICB I/O mentioned in section 1. The OICB v1.1 board uses an ATMega2560, same as an Arduino Mega 2560. All pins are 5V tolerant. The switch inputs need to be shorted between GND and the input to register “Closed”. The analog input values range from 0 to 255. The PWM outputs range from 0 to 255. Finally, the LED outputs are sourced by an LED driver chip set to drive a constant 20mA per pin.

4.2.2.2.5 Control Board v1.1 Simulator

This simulates the OICB v1.1 board with all the I/O available. For more simulator information, please refer to section 5.

4.2.2.3 Network Table Address

The OICB software gives you the flexibility and ease to set the NT server address any way you want. It has the ability to create IPv4 addresses, mDNS addresses, connect to a simulator such as the LabVIEW simulator or Python simulator, or you can set the address to something else entirely different. Click “Ok” to accept the address at the bottom or “Close” to discard any changes.

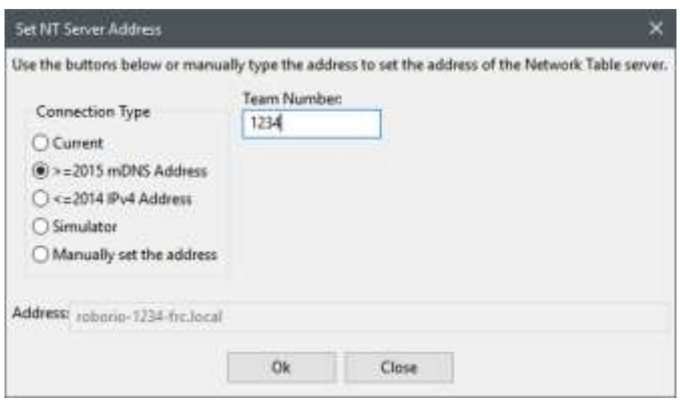


Figure 11. Set NT Server Address window

4.2.2.4 Logging Level

This is Version 1.0 software. You are likely to find a bug. If that is the case, I’ll probably need some debug information from you. The logging level has five different levels:

1. Fatal (Logging disabled) – Nothing is logged Fatal. If you never want to write log data, set to Fatal.
2. Error (Includes Fatal) – This will log exceptions and errors. Normally you will not see anything in your log. If the software encounters some error, either user or software related, it will be logged. Common errors would be invalid data size, no connection, or connection lost.

3. Warning (Includes Error and Fatal) – Warnings from the software will also be logged.
4. Info (Includes Warning, Error, and Fatal) – This log level logs state transitions and version information.
5. Debug (Includes Info, Warning, Error, and Fatal) – This will log almost everything the software is doing. If you leave this set, your log file will become big. This would be most useful for someone learning how the software works.

4.2.3 Help

4.2.3.1 Show Log

This will open the log in your default editor associated with *.log files. This file is located in the root of your user directory as *OperatorInterfaceControlBoard.log*. If you do not know where your root is, use command prompt and type *cd %USERPROFILE%*

4.2.3.2 Help

The Help button will open this PDF in your default PDF application.

4.2.3.3 About

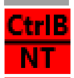
The About button will open a dialog with application information like shown in Figure 12.

Figure 12. About window

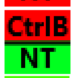
5 Taskbar Icon

The taskbar icon shows a quick glance of the current state of the software. There are seven different icons:

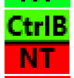
Initializing software



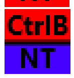
The control board and NT server are disconnected.



The control board is disconnected. The NT server is connected.



The control board is connected. The NT server is disconnected.



The control board is disconnected. The NT server is disabled (Test mode enabled).

The control board is connected. The NT server is disabled (Test mode enabled).

The control board and NT server are connected.

6 Firmware

The firmware used in the OICB does not contain any user adjustable settings. Any modifications are done at your own risk. If you mess it up, you can always redownload the firmware from the repository and load it with the Arduino IDE.

There are 4 status LEDs on the OICB:

LED	Description
Power	Indicates that 5V is present from USB
Tx (Transmit)	Data is currently being transmitted from OICB to your computer.
Rx (Receive)	Data is currently being transmitted from the computer to our OICB.
Status	Blinking – Standby; Solid - Running

Figure 13. Control board LED description

When the status is solid / running, the Tx and Rx LEDs will rapidly flash.

7 Simulator

The simulator is used in lieu of having a physical control board. This can allow you to test your software or robot when the control board is still being made or if you want to build the control board after your robot is bagged.

The user interface for the simulator is the same across all the different variants. The only difference is the number of I/O available.

Figure 14. Simulator UI

The simulator cannot be closed, only minimized. If you want to simulate a board disconnect (always good to think of for any device your robot uses for input), you can uncheck "Control Board Connected?". You can adjust the values for Analog inputs and Switch inputs. Information on the current PWM outputs and LED outputs will also be shown.

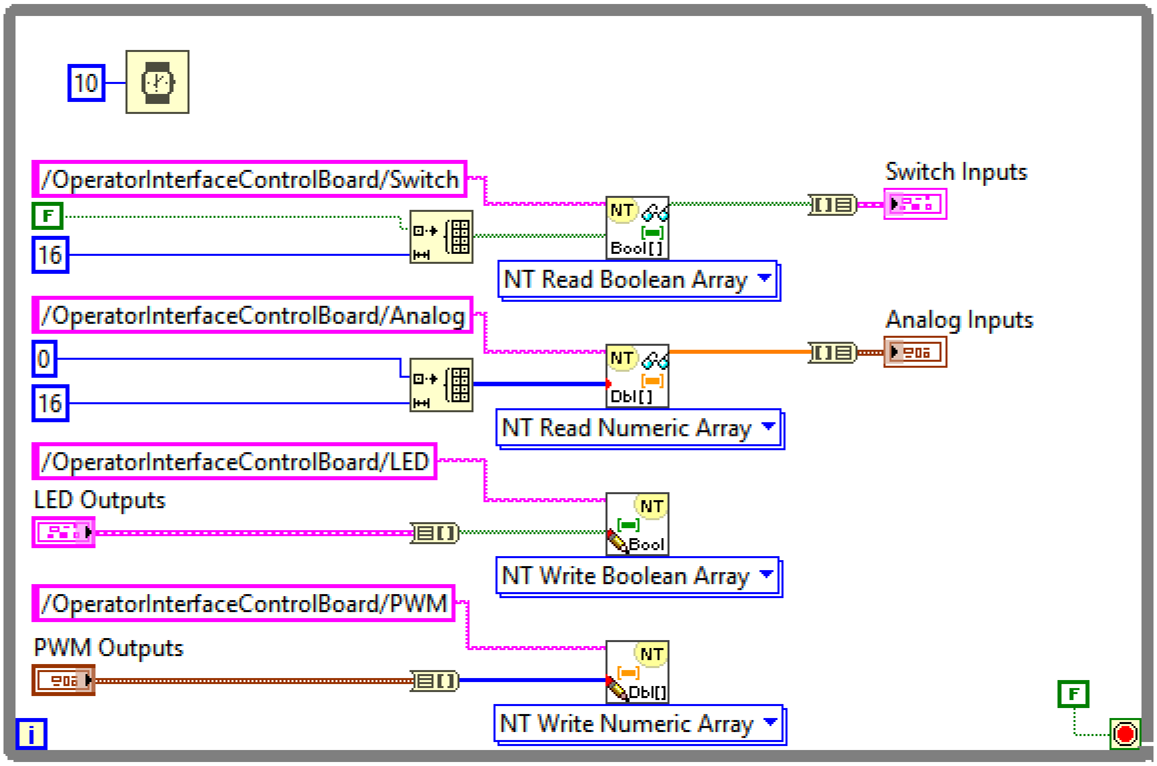
8 Robot Code Integration

So you've carefully read this document up to this point, but I still haven't covered how to use the board with your robot. Not to worry, the time has come.

It's actually easy to do, like everything up to this point. The path for the four groups of signals are:

1) "/OperatorInterfaceControlBoard/Switch" – 16-element Boolean Array (NT Read)
2) "/OperatorInterfaceControlBoard/Analog" – 16-element Numeric Array (NT Read)
3) "/OperatorInterfaceControlBoard/LED" – 16-element Boolean Array (NT Write)
4) "/OperatorInterfaceControlBoard/PWM" – 11-element Numeric Array (NT Write)

Here's an example of LabVIEW Block Diagram code:



Here's an example of the Front Panel for that Block Diagram:

9 Acknowledgements

This project would not have been possible if it were not for some key teams and people.

1) Team Fusion #364 – This team provided me with the opportunity to try the OICB v1.0 back in 2013. While the hardware design has changed slightly (really the new board just has less parts and better analog inputs), their board still works. We used it for the 2013 season to test the reliability of the hardware. We used LabVIEW code built into the dashboard to communicate back to the robot.
2) Garnet Squadron #4901 – This team allowed me to spin OICB v1.1 for the 2016 season. That year, we found that LabVIEW had some weird issue taking in a video stream from our Raspberry Pi that caused it to lag to the point of the OICB being unusable. This affected the control board's operation during some matches at the Orlando regional. Later in 2016, I started developing this Python based application, which brings me to my third acknowledgement.
3) Peter Johsnson (PeterJohnson) and Dustin Spicuzza (virtuald) provided the pynetworktables library for Python 3. Without this library, I would not be able to communicate back to the robot as effortlessly as I do.
4) My wife. Seriously, I have spent a ton of time on this project, and she still loves me. I knew she was special when I proposed to her at the Bayou Regional back in 2011.

10 Contact Information

Need to contact me? Did you read this manual first? If so, here is my email:
ryannazaretian@gmail.com