



FRC IOT Manual

FIRST FRC IOT Manual

Charger Robotics - Digital Solutions

Public

Version: 1.0
Date: 3/27/2020



Table of Contents

<i>Acknowledgements</i>	3
1. <i>Overview</i>	3
2. <i>Bill of Materials</i>	3
3. <i>Implementation Guide</i>	4
3.1. <i>CANBUS Edge Device</i>	4
3.2. <i>Broker API</i>	5
3.3. <i>Gateway Dashboard</i>	6
3.4. <i>Thingworx Platform Configuration</i>	7
3.5. <i>Using the Solution</i>	10
4. <i>Appendix A – Misellaneous Items</i>	12



Acknowledgements

Charger Robotics would like to recognize, and thank all the organizations that came together to make this initial FRC IOT release possible. To Rockwell Automation for the engineering and architecture support they provided. PTC for their technical support, and providing the team with a Thingworx license. Also, CTR Electronics for providing a coding solution for the CANBUS Edge Device sampling. FIRST, without them none of this would be possible. To all the student participant of FIRST FRC. They are the future of STEM.

1. Overview

An FRC IOT solution allows a FIRST Robotics team to capture IOT data from an FRC robot and send to a Thingworx Foundation instance in the cloud, or on-prem.

FRC Robot(CANBUS Edge Device) → Driver's Console → Thingworx

Charger Robotics, Team 537, provides an end-to-end, preconfigure solution for up to eight (8) Phoenix based motor controllers. The solution can be easily expanded to handle any IOT type sensors an FRC team may want to include in their solution.

Constraints

The implementation described in this document has these constraints.

1. Only CTR Electronics motor controllers leveraging the Phoenix SDK are supported by this solution.
2. A PTC Thingworx Foundation instance.
3. The robot network needs be configured to use static IP addresses. See the Appendix at the end of this paper for more information

Currently PTC does not have an educational license for an on-prem (FRC Driver's Console) deployment. The current solution implementation leverages a PaaS cloud offering by PTC. This limits the real-time capabilities of the Thingworx platform. The solution design will change in the future when Thingworx educational licenses become available for on-prem.

2. Bill of Materials

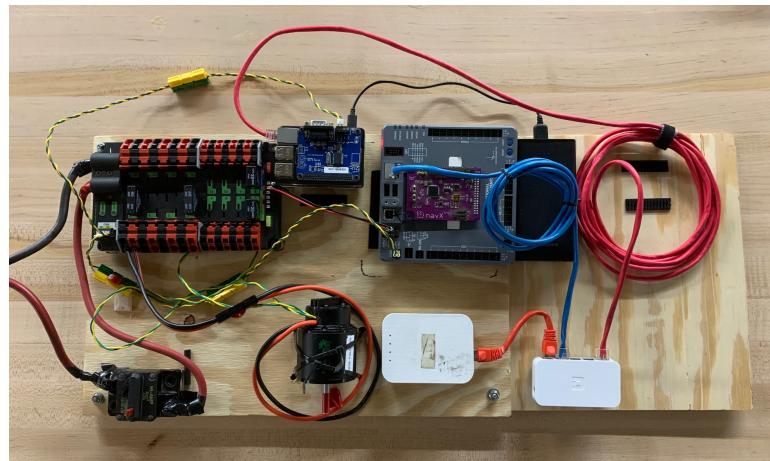
The following materials are required to build an FRC IOT solution.

- Raspberry Pi 3 B+ System With Single CAN Bus Interface.
<https://copperhilltech.com/raspberry-pi-3-b-system-with-single-can-bus-interface/>
- PTC Thingworx Educational License:
- Network Hub/Switch.
- Talon SRX Motor Controller (any Phoenix code based controller)
- RoboRIO
- CAN Bus Controller

- FRC Radio
- Windows 10 PC as Driver's Console
- USB Lithium Battery

3. Implementation Guide

This section provides instructions for implementing an IOT solution for a FIRST FRC Robot. The implementation pulls from a few external sources. The instructions will reference those sources where applicable. Below is a picture of the testboard configuration.



3.1. CANBUS Edge Device

A CANBUS Edge Device is required for the sampling of motor controller data. The PI will have its own power source supplied by a USB lithium battery (check FRC rules for allowed specs). While the sampling could be done on the RoboRIO, it is preferred to move the IOT processing to a separate device. This will prevent the sampling code from interfering with robot operations. The Edge Device is a Raspberry Pi with a Control Area Network (CAN) Interface. The Raspberry PI with single can-bus interface from Copperhill Technologies is recommended:

<https://copperhilltech.com/raspberry-pi-3-b-system-with-single-can-bus-interface/>



Follow the initial configuration instructions provided on the Copperhill website.

<https://copperhilltech.com/pican2-controller-area-network-can-interface-for-raspberry-pi/> For this implementation you will want to use the screw terminal to connect to the CANBUS network. The



FRC IOT Manual

CANBUS interface on the PI terminates the CANBUS network, so the Pi will need to be at the end of the network and the PDP on the robot will need to be configured not to terminate. The BAUD rate needs to be set to 1000000 to communicate with the Phoenix SDK. You will also want to configure the PI to have a static IP of 10.xx.xx.222.

Code Deployment

There are three (3) files that need to be copied into the root directory of the Raspberry PI. The files are located here:

<https://github.com/beckologist/FRCIOTRelease/tree/master/RaspberryPiCode>

3.2. Broker API

The broker API is the listener on the Operators Station (Windows 10 PC) that will receive data from the Java program of the CANBUS Sampler.

Download all the GatewayAPI folder and its contents to your desktop.

<https://github.com/beckologist/FRCIOTRelease>

Open the GatewayAPI folder and right-click on the GatewayAPI.exe file, and choose “Run As Administrator”.

Name	Date modified	Type	Size
GatewayAPI.exe	3/6/20 11:17 PM	Application	9 KB
GatewayAPI.exe.config	2/19/20 8:16 PM	XML Configuration	1 KB
GatewayAPI.pdb	3/6/20 11:17 PM	Program Debug Data	24 KB
Newtonsoft.Json.dll	8/3/14 9:33 PM	DLL File	491 KB
Newtonsoft.Json.xml	8/3/14 9:33 PM	XML Document	468 KB
System.Net.Http.Formatting.dll	11/28/18 2:00 PM	DLL File	175 KB
System.Net.Http.Formatting.xml	11/28/18 2:00 PM	XML Document	189 KB
System.Web.Http.dll	11/28/18 2:01 PM	DLL File	445 KB
System.Web.Http.SelfHost.dll	11/28/18 2:02 PM	DLL File	88 KB
System.Web.Http.SelfHost.xml	11/28/18 2:02 PM	XML Document	15 KB
System.Web.Http.xml	11/28/18 2:01 PM	XML Document	528 KB

You should see the following results:



The Operator's Station is now listening for incoming messages from the CANBUS Edge Device.

Note: The Windows 10 firewall rules will need to be configured to allow external HTTP calls to port 5801. Depending on your teams security posture, you may want to only allow the IP address of the CANBUS Sample to access this Port.

3.3. Gateway Dashboard

The Gateway Dashboard is the UI for controlling the FRC IOT eco-system. It provides the following capabilities:

- Start/Stop sampling and sending of the CANBUS motor controller data
- Visibility into the queueing activity
- Start/Stop of data forwarding to the Thingworx Cloud Platform
- Visibility of Notification messages for the RoboRIO
- Configuration of system parameters

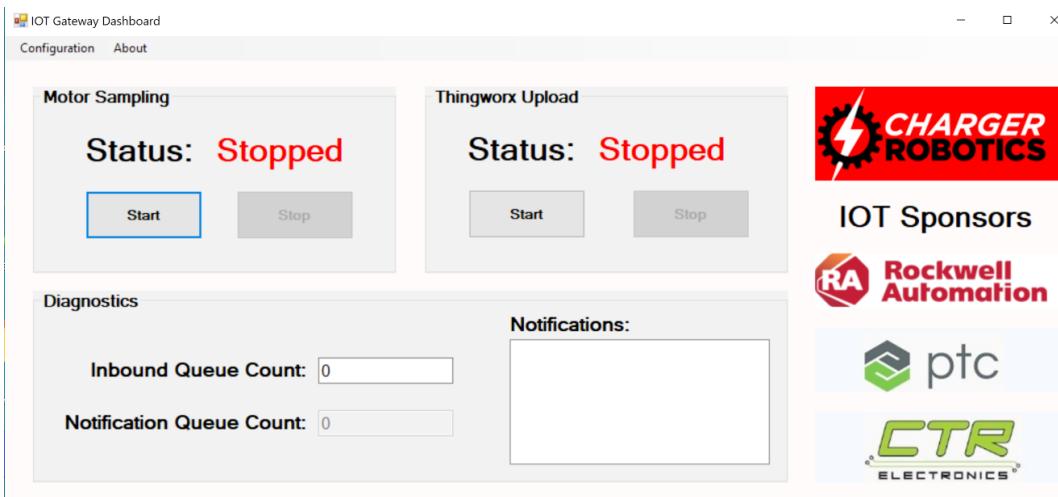
Download the GatewayBashboard folder and its contents to your desktop.

<https://github.com/beckologist/FRCIOTRelease>

Open the GatewayDashboard folder and right-click on the SamplingStartStop.exe file, and choose “Run As Administrator”.

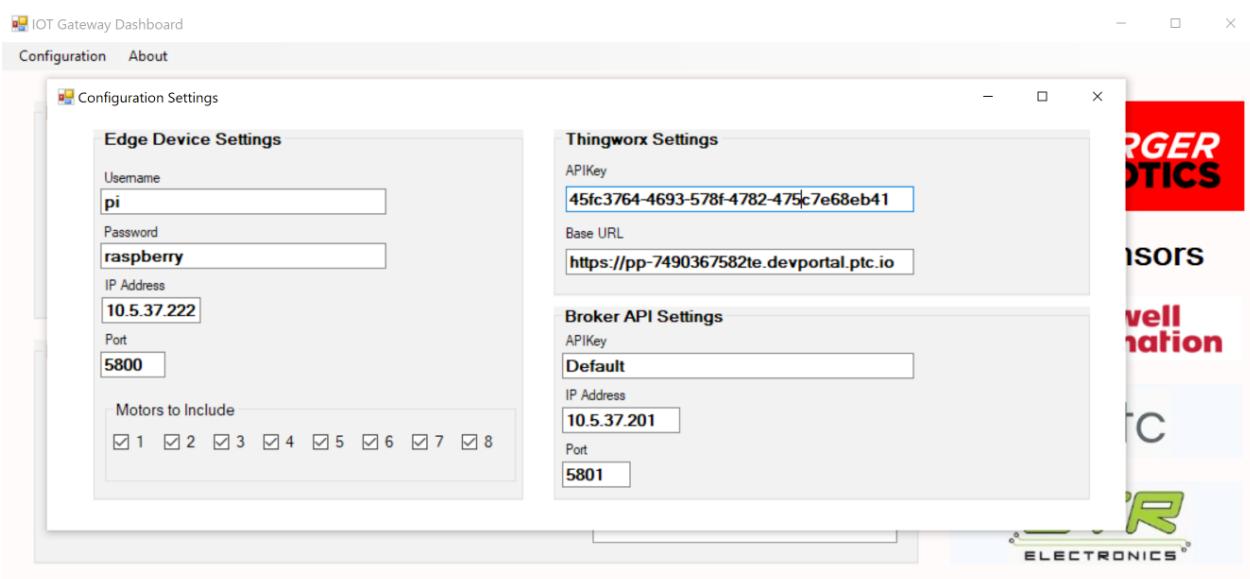
Name	Date modified	Type	Size
SamplingStartStop.pdb	3/13/20 9:56 PM	Program Debug D...	84 KB
SamplingStartStop.exe.config	3/13/20 2:06 PM	XML Configuratio...	3 KB
SamplingStartStop.exe	3/13/20 9:56 PM	Application	87 KB
RestSharp.xml	1/12/20 11:02 AM	XML Document	181 KB
RestSharp.dll	1/12/20 11:02 AM	DLL File	182 KB
Renci.SshNet.xml	10/16/17 9:53 PM	XML Document	1,022 KB
Renci.SshNet.dll	10/16/17 9:53 PM	DLL File	413 KB
Newtonsoft.Json.xml	8/3/14 9:33 PM	XML Document	468 KB
Newtonsoft.Json.dll	8/3/14 9:33 PM	DLL File	491 KB

The Gateway Dashboard interface will start.



3.3.1. System Configuration Parameters

The parameters for configuring the IOT system can be found under the Configuration -> Setting menu.



3.4. Thingworx Platform Configuration

The final component to configure is the Thingworx platform. There are several options available for Thingworx deployment. At the time of this writing, an educational license did not exist for anything but a Cloud deployment.



Platform Installation (Windows)

<https://developer.thingworx.com/en/resources/guides/foundation-windows-install-guide>

Platform Configuration

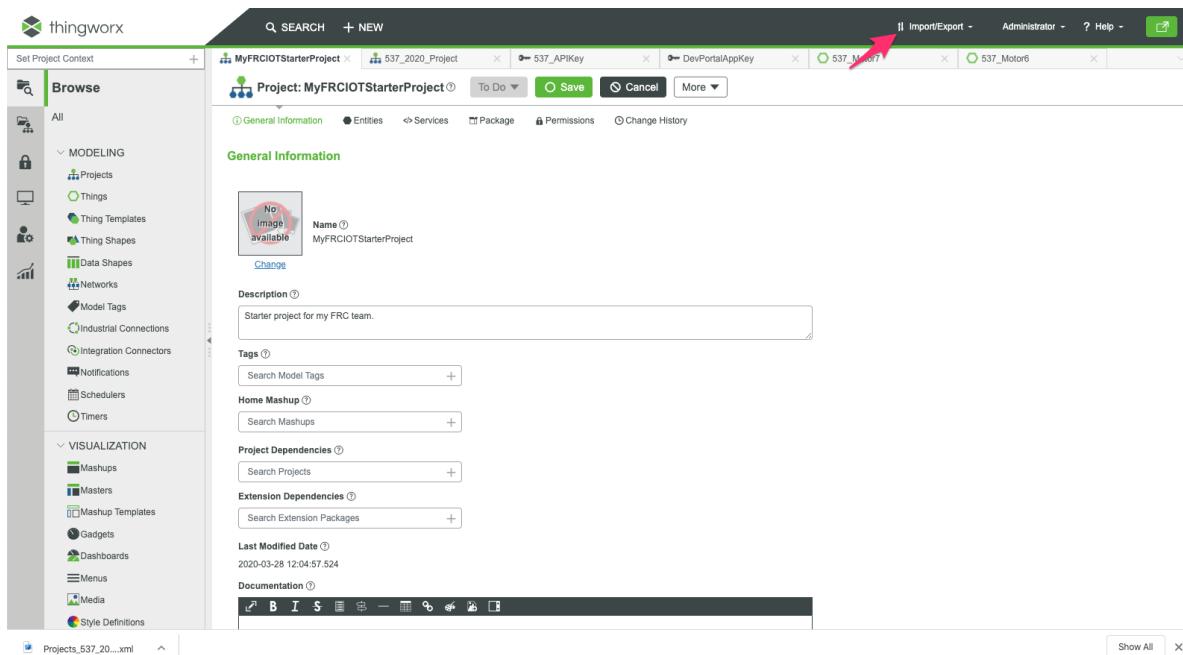
<https://developer.thingworx.com/resources/guides/thingworx-foundation-quickstart>

Once the Thingworx instance is up and running, a “Starter” project configuration file can be imported into Thingworx. This is an XML file that contains a prebuilt project for working with up to 8 motors. It provides all the required Thing Shapes, Thing Templates, and Things for up to 8 motors. A set of basic Mashups are also provided as a starting point.

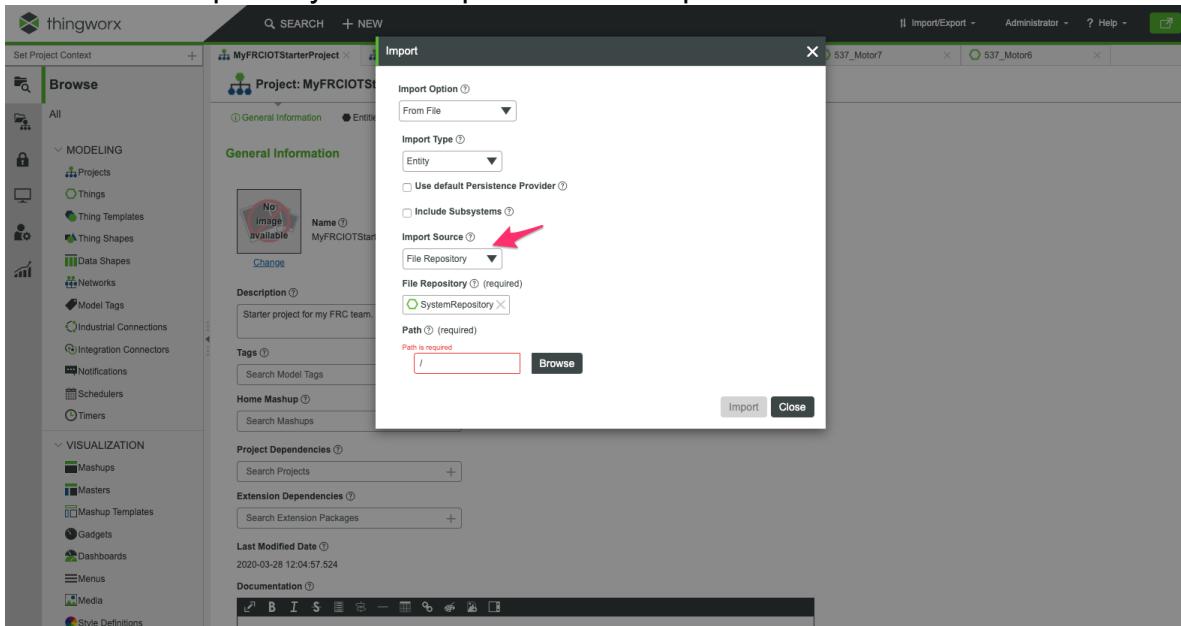
The name of the file is FRC_IOT_Starter_Project.xml. The file is located here:

<https://github.com/beckologist/FRCIOT>. Once you are logged into Thingworx:

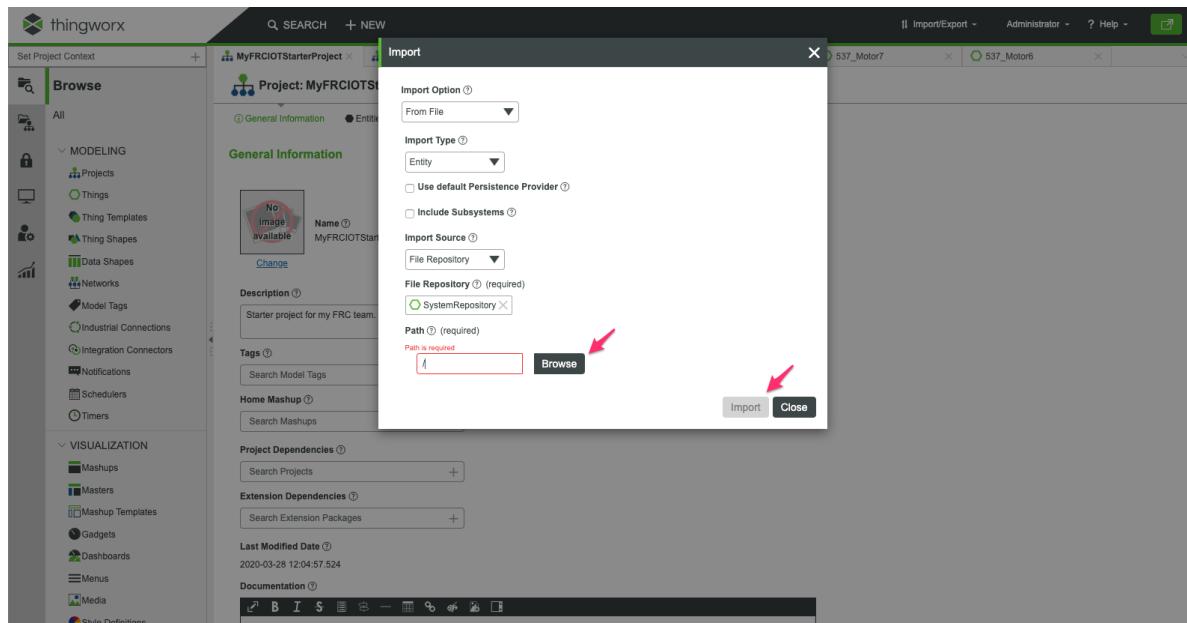
1. Select Import/Export at the top of the screen.



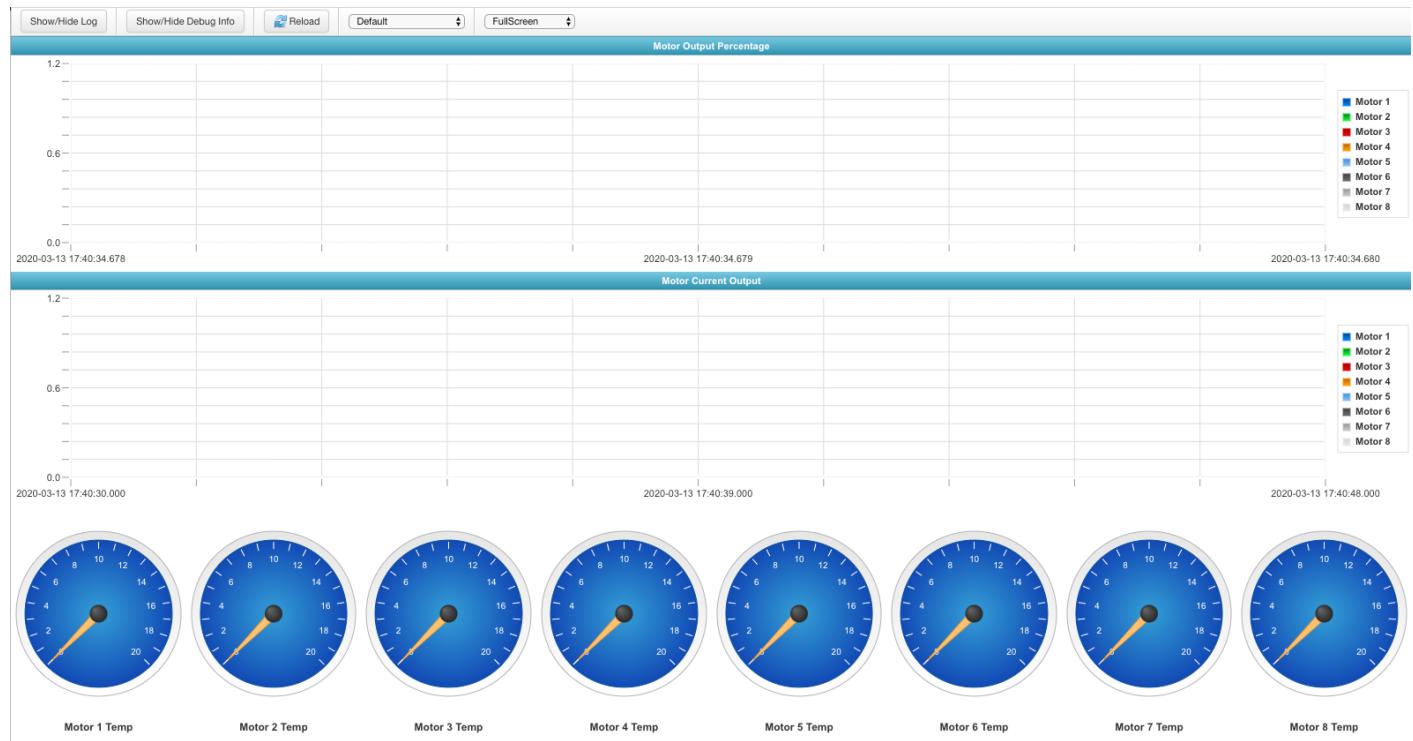
2. Select “File Repository” from Import Source drop down.



3. Browse to the FRC_IOT_Starter_Project.xml file that was downloaded from the GITHUB repository(<https://github.com/beckologist/FRCIOT>). Then click on the Import button.



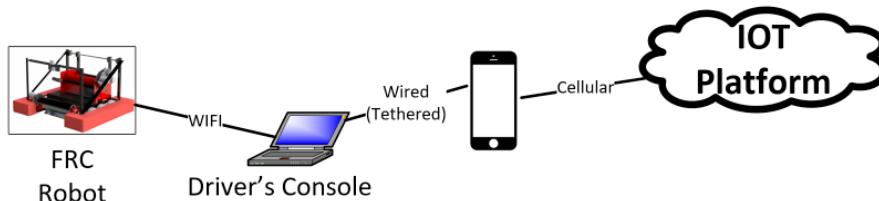
Standard Mashups are provided in the FRC_IOT_Starter_Project.xml file.



3.5. Using the Solution

There are two different scenarios that an FRC IOT solution will be used; Competition, and Non-Competition. Per FRC rules state that during competition the Driver's Console can only be connected to field network. No other netrok connections are permitted. If your Thingworx instance is not running locally on the Driver's Console (i.e. a cloud-hosted instance), the IOT data must be stored locally on the Driver's Console, and forwarded to a Thingworx Cloud instance after the match has completed. Fortunately, the Gateway Dashboard supports all possible Thingworx deployment options.

The Driver's Console running the Broker API and Gateway Dashboard will need to be able to connect to the FRC robot radio, and the internet. This will require the standard FRC Driver's Console connection to the FRC Radio on the robot, and a tethered cellular device for internet access.



Checklist:

- The CANBUS Edge Device is configured a ready to start sending data (Section 3.2).
- The Broker API is deployed and running with the appropriate firewall settings (Section 3.3).
- The Gateway Dashboard is installed,configured, and ready to use (Section 3.4).
- A Thingworx instance is configured and ready to use (Section 3.5).

3.5.1. Non-Competition (Cloud-hosted Thingworx)

1. Connect the Driver's Console the the FRC Radio on the robot via WIFI.
2. Teather the Driver's Station to a cellphone or tablet with cellular data connection.
3. Click on the "Start" button in the Motor Sampling section of the Gateway Dashboard. This will ssh to the CANBUS Edge Device and start the sample.sh script. The Inbound Queue Count numbers should begin to go up as messages arrive. Messages are stored in the MSMQ queues for forwarding when requested.
4. Click on the "Start" button in the Thingworx Upload section of the Gateway Dashboard. This will start the sending of messages to the Thingworx platform.

This will result in the data being sent to Thingworx in “realtime”.

3.5.2. Competition (Cloud-hosted Thingworx)

1. Connect the Driver's Console the the FRC Radio on the robot via WIFI.
2. Click on the "Start" button in the Motor Sampling section of the Gateway Dashboard. This will ssh to the CANBUS Edge Device and start the sample.sh script. The Inbound Queue Count numbers should begin to go up as messages arrive. Messages are stored in the MSMQ queues for forwarding when requested.

***** Compete in the competition round – drive it like you stole it! ***
Once your Alliance has won, and you are back in the Pit Area...**

3. Teather the Driver's Station to a cellphone or tablet with cellular data connection.
4. Click on the "Start" button in the Thingworx Upload section of the Gateway Dashboard. This will start the sending of messages to the Thingworx platform.

This will result in the data being sent to Thingworx.

3.5.3. Competition (Locally-hosted Thingworx)

This is similar to Non-Competition(Cloud-hosted Thingworx), with the exception of not needing a Cellular connection.

1. Connect the Driver's Console the the FRC Radio on the robot via WIFI.
2. Click on the "Start" button in the Motor Sampling section of the Gateway Dashboard. This will ssh to the CANBUS Edge Device and start the sample.sh script. The Inbound Queue Count

numbers should begin to go up as messages arrive. Messages are stored in the MSMQ queues for forwarding when requested.

3. Click on the “Start” button in the Thingworx Upload section of the Gateway Dashboard. This will start the sending of messages to the Thingworx platform.

This will result in the data being sent to Thingworx in “realtime”.

4. Appendix A – Misellaneous Items

- CAN Interface Baudrate = 1 000 000
- Static IPs
- Set PI SSH port to 5800
- Broker API listens on port 5801
- Notification API listens on 5802
- PI Static IP is 10.5.3.222
- Gateway inbound queue named: IOTData
- Gateway notification queue named: IOTNotification