Rotom Toaster Robot Manual

Release v1.1-3-g6bed5f2

Tidal Force, FRC Team 1721

Mar 24, 2023

CONTENTS:

CHAPTER

ONE

INTRODUCTION

OUTREACH

Warning: Make sure this manual's revision hash and the robot hash match!

FRC TEAM 1721 TIDAL FORCE
OUR TEAM HAS PROUDLY WORKED
88.0
DAYS
WITHOUT AN ACCIDENT.
SUPPORT OUR TEAM'S MISSIO
SUPPORT SAFTEY
Record 24
Time 78
FIRST

Fig. 1: Accident Counter

This is the official Robot Manual for Tidal Force FRC Team 1721's 2023 robot, Rotom Toaster. For the FRC Game Charged Up!

This manual contains technical diagrams, prototyping processes and in depth technical descriptions of all robot functions, from software to hardware and the messy bits in-between.

If you're really that interested to know all the details, you'll find them here!

CHAPTER

TWO

CRITICAL DESIGN REVIEW

2.1 Introduction

OUTREACH

Design review process

2.2 Bumpers

OUTREACH

CHAPTER

THREE

THE DRIVETRAIN

3.1 Introduction

OUTREACH

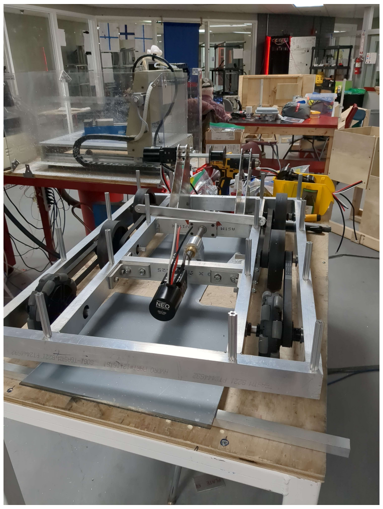


Fig. 1: The drivetrain during construction

The tank drivetrain is tried and tested, even by us! During Prototyping we experimented with our kitbot drivetrain and used it as a solid base to train new members with.

The drivetrain is a 3x3 tank drive base, with all wheels powered, and the first two being omnidirectional.

The drivetrain's top speed is estimated to be 45kts. Although it has not yet been in wide enough area to confirm.

3.2 Prototyping

OUTREACH

In the pre season we prototyped using our kitbot and counterweights.

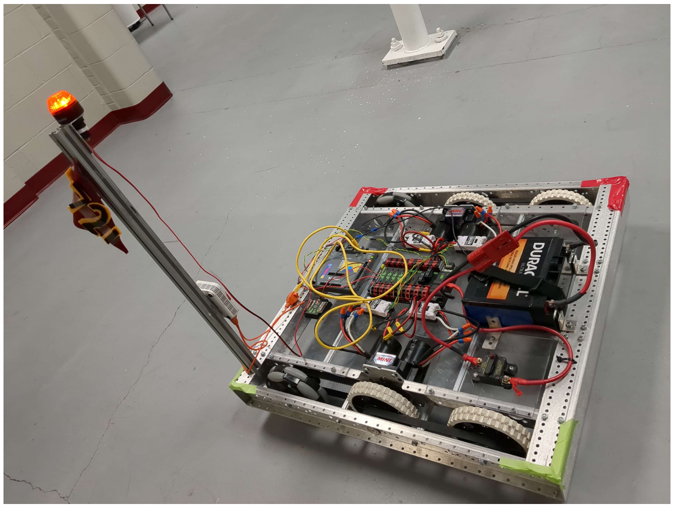


Fig. 2: The kitbot/prototype drivetrain.

3.3 Software

SOFTWARE

The drive base uses simple two wheel steering kinematics and an odometry with encoders, gyro and accelerometers to detect robot position.

3.3.1 Config Files

```
drivetrain:
  max_velocity: 1.5                # Updated Never by Nobody
  encoderConversionFactor: 21.43  # Updated 1/27 by Joe

  # All pose coordinates are in meters.
  leftMotor:
    Motor1Port: 2                 # Updated 1/12 by Keegan
    Motor2Port: 1                 # Updated 1/12 by Keegan
    EncoderPorts: [2, 1]          # Updated 2/12 by Keegan
    EncoderReversed: False        # Updated Never by Nobody
    Inverted: False               # Updated Never by Nobody

  rightMotor:
    Motor1Port: 3                 # Updated 1/8 by Keegan
    Motor2Port: 4                 # Updated 1/8 by Keegan
    EncoderPorts: [3, 4]          # Updated Never by Nobody
    EncoderReversed: True         # Updated Never by Nobody
    Inverted: True                # Updated Never by Nobody

  navX: # NavX ids
    can_id: 0                     # Updated 1/13 by Keegan
    yaw: 0                        # Updated 1/13 by Keegan
    pitch: 0                      # Updated 1/13 by Keegan
```

(continues on next page)

(continued from previous page)

```
    roll: 0                      # Updated 1/13 by Keegan
```

```
drive:
  # Diff lock
  kStabilizationP: 0.011
  kStabilizationI: 0.0001
  kStabilizationD: 0

  # TurnToAngle
  kTurnP: 0.007
  kTurnI: 0.006
  kTurnD: 0.001

  kMaxTurnRateDegPerS: 100
  kMaxTurnAccelerationDegPerSSquared: 300

  kTurnToleranceDeg: 5
  kTurnRateToleranceDegPerS: 10 # degrees per second
```

CHAPTER

FOUR

THE ARM

4.1 Introduction

OUTREACH

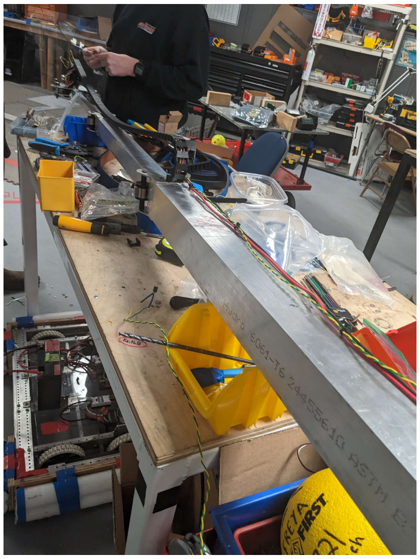
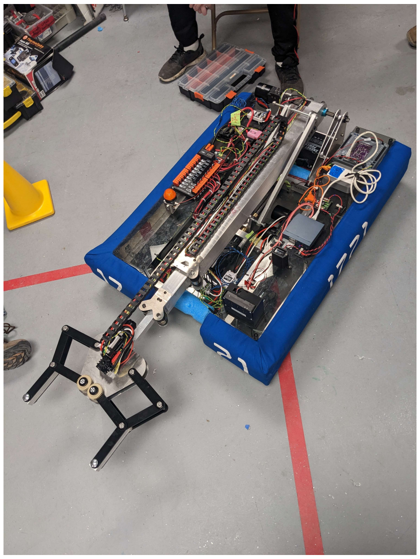


Fig. 1: The arm during construction

The arm is a 3 stage nested tube stock assembly powered by springs. Its a design we've become familiar with. It is powered by a pair of Rev NEO motors. One for its angle, and another for its extension.

The angle is determined by a lead screw and lifting arm assembly so as to avoid excessive motor load and precise location tuning.



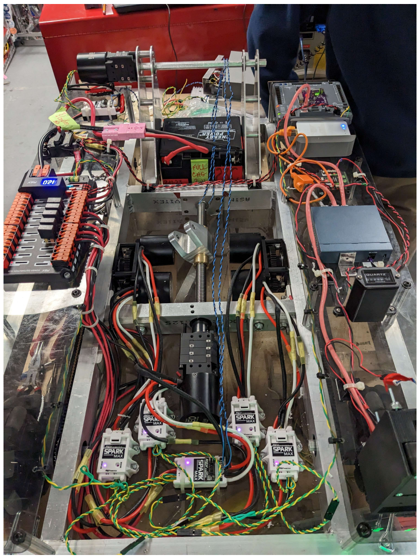


Fig. 2: A close up of the lead screw

4.2 Software

SOFTWARE

The arm is driven using the embedded PID loops on both its motors. Limit switches limit its maximum extents to protect the robot from damage.

4.2.1 Config Files

```
arm:
  elevatorMotor: # Drives the spool to move the lift in and out
    MotorPort: 5                 # Updated 1/22 by Keegan
    Inverted: False              # Updated 1/22 by Keegan
    LimitSwitch: 0               # Updated 2/15 by Keegan
    ConversionFactor: 1          # Updated 1/18 by Joe
    Start: 0                     # Updated 3/18 by Keegan

  ladderMotor: # Drives the lead screw to run the lift up and down
    MotorPort: 6             # Updated 1/22 by Keegan
    Inverted: False          # Updated 1/22 by Keegan
```

```
arm:
  elevator:
    kp: 0.075
    ki: 0
    kd: 0.1
    ff: 0

  ladder:
    kp: 0.1
    ki: 0
    kd: 0.1
    ff: 0
```

CHAPTER

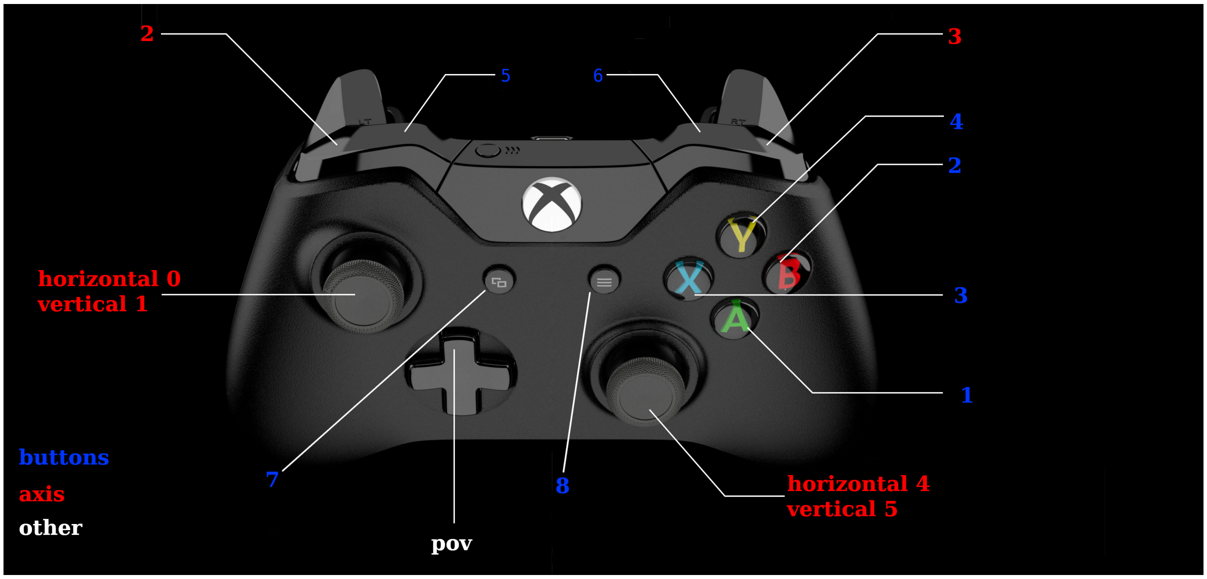
FIVE

OPERATING AND CONTROLS

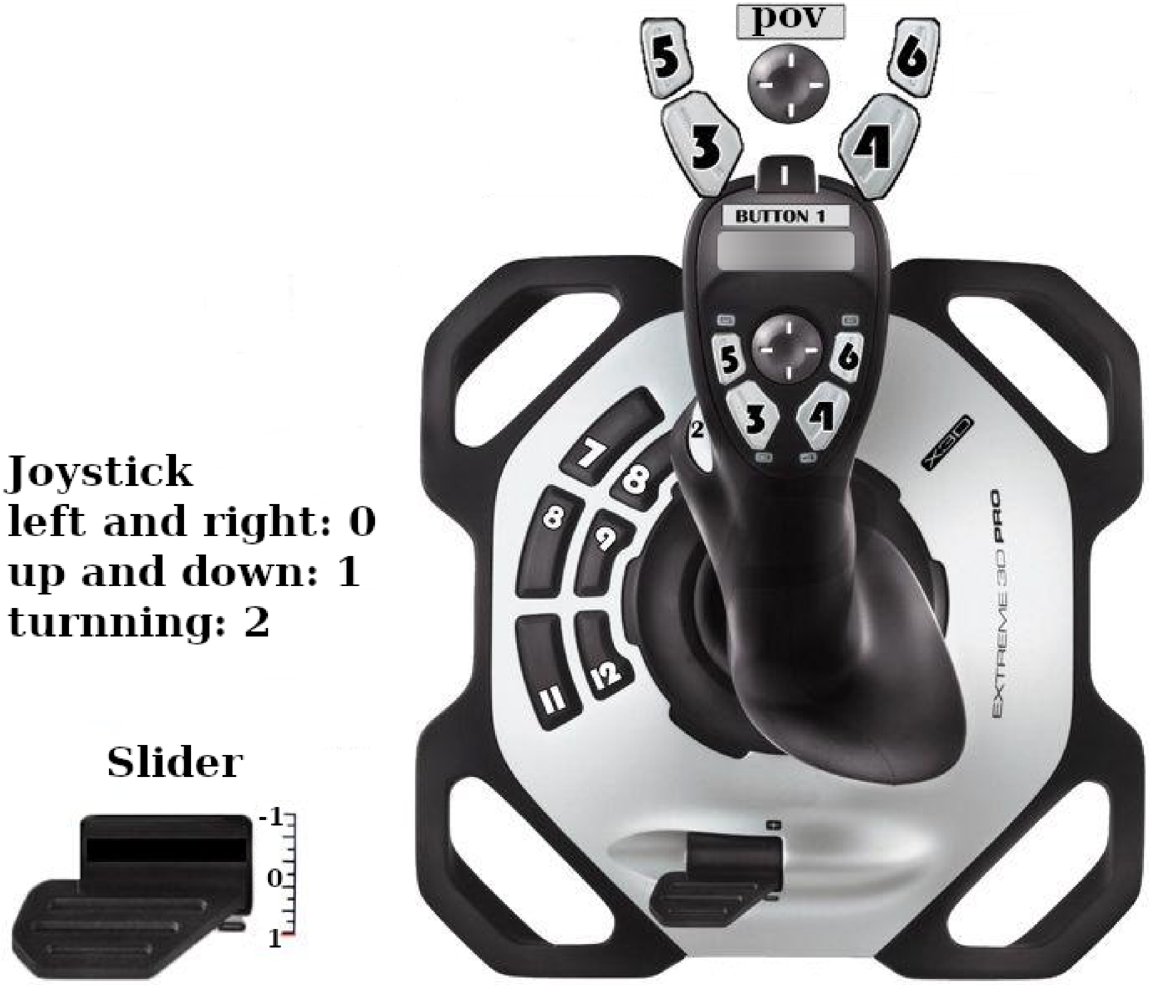
5.1 Controller Map

DRIVETEAM

```
# This file defines the input
# and control scheme of the robot.
# this is orginized into control modes

main mode:
  driver: # using a stick controller
    controller_port: 0 # What usb port the driver controller is on

    # Axis
    ForwardAxis: 1
    SteerAxis: 2

    # Buttons
    HalfSpeedButton: 1
    DiffLock: 2
    Turn90: 3
    TurnAnti90: 4
    StartConfig: 5
    Slow: 10
    Crawl: 3

  operator: # using an "x-box" controller
    controller_port: 1

    # Hold Buttons
    Unclamp: 6
    Clamp: 5
    FindZero: 8
    hold: 3

    # Presets/Modes
    LowGoal: 1
    HighGoal: 4
    ManualMode: 7
```



This needs to be filled out!



This needs to be filled out!

CHAPTER

SIX

ROBOT WIRING

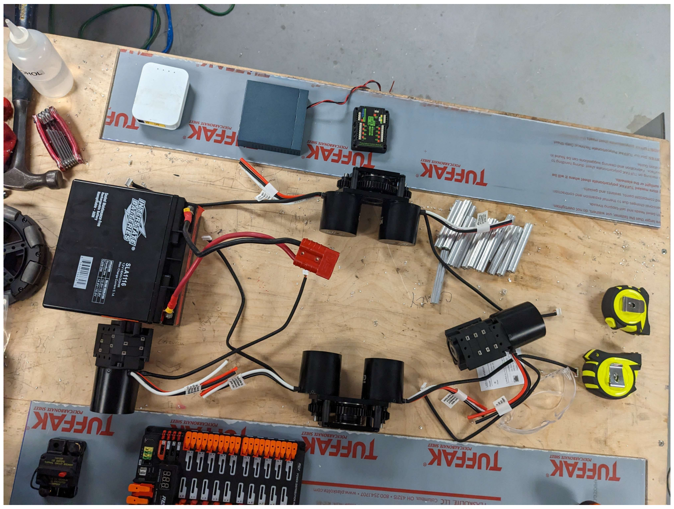


Fig. 1: Wiring Prototype

6.1 Full Electrical Schematic

OUTREACH PITCREW

Schematic here!

CHAPTER

SEVEN

ROBOT NETWORK ARCHITECTURE

The robot contains several networked computers, all talking over high speed tcp/ip. The layout of the network may be of importance during service and debugging.

7.1 Network layout

SOFTWARE PITCREW

Table 1: Robot Network Hosts

Hostname	mDNS	IP
roborio	I forget	10.17.21.2
nuc		10.17.21.11
limelight	TBD	TBD
aivision	TBD	TBD
driverstation		DHCP

CHAPTER

EIGHT

DEVELOPING

8.1 Configuration Files

PITCREW SOFTWARE

This is literally the robot hardware configuration written in yaml.

These are provided as reference for use by pit crew.

Use this to confirm/reconfigure motor controller addresses or examine if pose x/y cords match actual.

```yaml
# This file defines the robot's
# physical dimensions. Things like
# Motor placement, max and min extensions
# of arms, and similar should go here.

drivetrain:
  max_velocity: 1.5                 # Updated Never by Nobody
  encoderConversionFactor: 21.43  # Updated 1/27 by Joe

  # All pose coordinates are in meters.
  leftMotor:
    Motor1Port: 2                 # Updated 1/12 by Keegan
    Motor2Port: 1                 # Updated 1/12 by Keegan
    EncoderPorts: [2, 1]          # Updated 2/12 by Keegan
    EncoderReversed: False        # Updated Never by Nobody
    Inverted: False               # Updated Never by Nobody

  rightMotor:
    Motor1Port: 3                 # Updated 1/8 by Keegan
    Motor2Port: 4                 # Updated 1/8 by Keegan
    EncoderPorts: [3, 4]          # Updated Never by Nobody
    EncoderReversed: True         # Updated Never by Nobody
    Inverted: True                # Updated Never by Nobody

  navX: # NavX ids
    can_id: 0                     # Updated 1/13 by Keegan
    yaw: 0                        # Updated 1/13 by Keegan
    pitch: 0                      # Updated 1/13 by Keegan
    roll: 0                       # Updated 1/13 by Keegan
```

(continues on next page)

(continued from previous page)

```
arm:
  elevatorMotor: # Drives the spool to move the lift in and out
    MotorPort: 5                # Updated 1/22 by Keegan
    Inverted: False             # Updated 1/22 by Keegan
    LimitSwitch: 0              # Updated 2/15 by Keegan
    ConversionFactor: 1         # Updated 1/18 by Joe
    Start: 0                    # Updated 3/18 by Keegan

  ladderMotor: # Drives the lead screw to run the lift up and down
    MotorPort: 6            # Updated 1/22 by Keegan
    Inverted: False         # Updated 1/22 by Keegan
    ConversionFactor: 1     # Updated 1/18 by Joe
    Start: 55               # Updated 3/18 by Keegan



claw:
  MotorPort: 7            # Updated 1/22 by Keegan
  Inverted: False         # Updated 1/22 by Keegan
  EncoderPorts: 7         # Updated 2/11 by Keegan
  EncoderReversed: False  # Updated 1/22 by Keegan
```

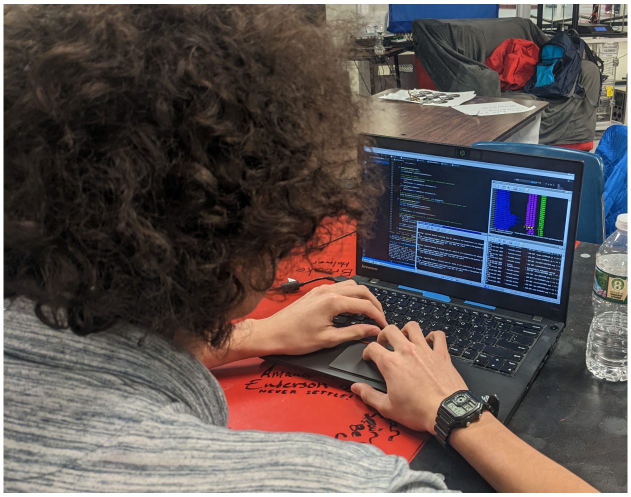


Fig. 1: A programmer in his natural habitat

8.2 Modifying these docs

SOFTWARE

8.2.1 Requirements for building the docs

You'll need to install several packages to build these docs

```
# Ubuntu/Debian users
sudo apt install texlive-latex-recommended texlive-latex-extra texlive-pictures pandoc␣
↪rename latexmk

# Arch/Manjaro users
pacman -S texlive-pictures pandoc latexmk
```

8.2.2 Making the docs

Under the `Docs` directory in `1721-ChargedUp`, setup a pipenv using the provided files and invoke the `make latexpdf` command to build the docs.

```
pipenv install # May take a while
pipenv shell
make latexpdf # Builds the manual
```

Find the generated `.pdf` under `docs/_build/latex/` it is recommended that you also have a restructured text interpreter such as reStructuredText. Documentation for sphinx can be found here

8.3 Using Github

SOFTWARE

We use Github as our main version control and automation system, paired with github actions and our own jenkins build server in the build space.

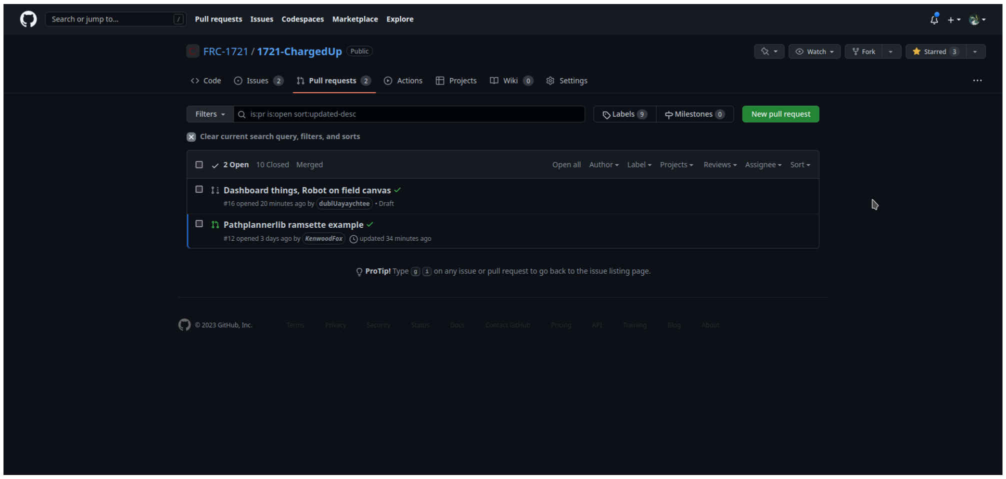
8.3.1 Making Pull requests

While you will mostly interact with github through git you should know how to make pull requests.

First go to the current repo 1721 ChargedUp.

You first under the pull requests tab create a new pull request, select the branch you want to make the pull request for.

Once approved by someone with proper clearance (Keegan or Joe) you may merge it, please **squash and merge** to not have billions of commits in main.



Dashboard things, Robot on field canvas ✓
#16 opened 20 minutes ago by dublUayaychtee • Draft

Pathplannerlib ramsette example ✓
#12 opened 3 days ago by KenwoodFox updated 34 minutes ago

ProTip! Type g i on any issue or pull request to go back to the issue listing page.

8.3.2 Creating Issues

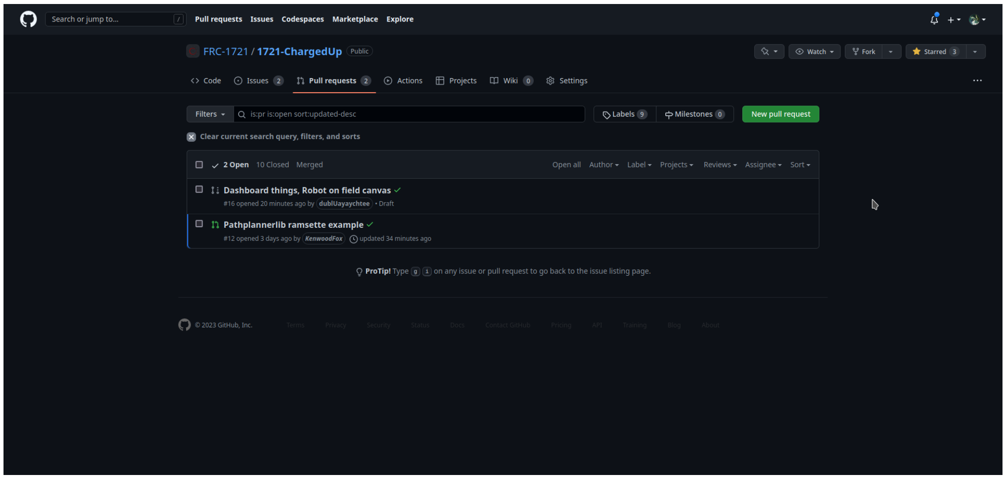
This will go over creating github issues, github issues are one way we communicate what needs to be done.

On the issues tab there will be a button saying *New issue*, click that and enter the information about the issue.

8.3.3 Making and Adding Labels and Milestones

Labels and Milestones are a nice way to categorize issues and pull requests, they both can created and added the same way.

To create a Label or Milestone on the main page of issues or pull requests there are buttons, saying *Labels* and *Milestones*.



Dashboard things, Robot on field canvas ✓
#16 opened 20 minutes ago by dublUayaychtee • Draft

Pathplannerlib ramsette example ✓
#12 opened 3 days ago by KenwoodFox updated 34 minutes ago

ProTip! Type g i on any issue or pull request to go back to the issue listing page.

After pressing input a name and a description, github uses a hex system for colors to help select a color use this.

8.4 Robot Simulator

SOFTWARE

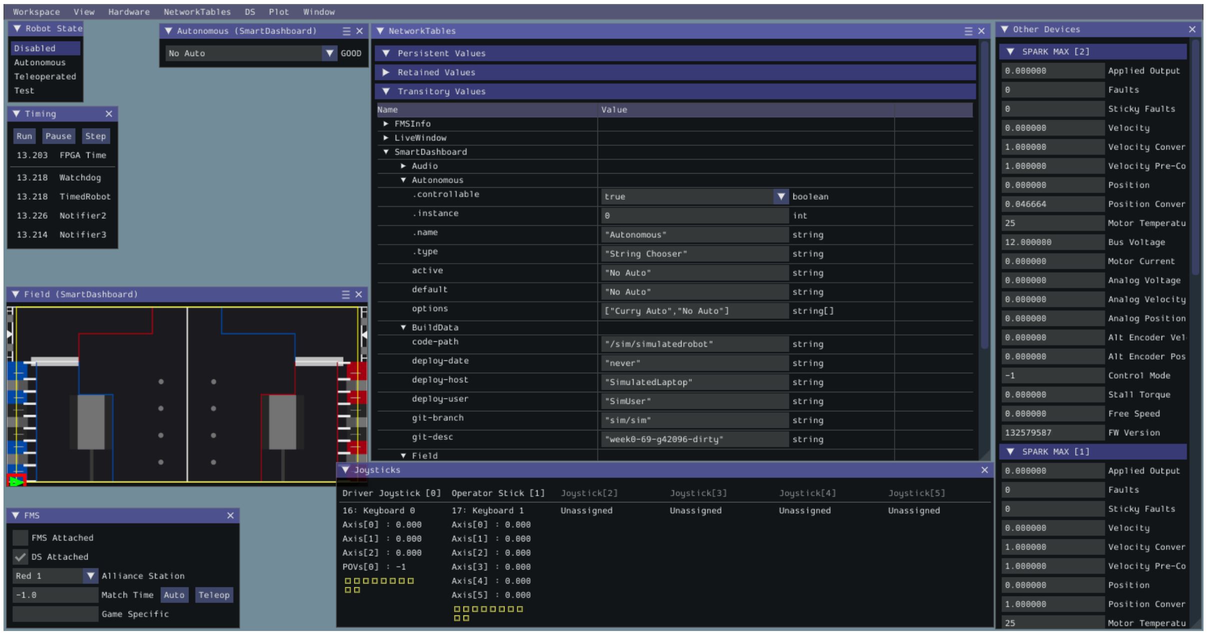
8.4.1 Requirements

Under the `rio` directory setup a pipenv using the provided files and invoke the `make sim` command to run the simulator.

```
pipenv install # May take a while
pipenv shell
make sim # Alternatively run: python3 robot.py sim
```

This will start the robot simulator. See *Using the Simulator* for more info.

8.4.2 Using the Simulator



No Auto GOOD

13.203	FPGA Time
13.218	Watchdog
13.218	TimedRobot
13.226	Notifier2
13.214	Notifier3

Name	Value	
▶ FMSInfo		
▶ LiveWindow		
▼ SmartDashboard		
▶ Audio		
▼ Autonomous		
.controllable	true	boolean
.instance	0	int
.name	"Autonomous"	string
.type	"String Chooser"	string
active	"No Auto"	string
default	"No Auto"	string
options	["Curry Auto","No Auto"]	string[]
▼ BuildData		
code-path	"/sim/simulatedrobot"	string
deploy-date	"never"	string
deploy-host	"SimulatedLaptop"	string
deploy-user	"SimUser"	string
git-branch	"sim/sim"	string
git-desc	"week0-69-g42096-dirty"	string
▼ Field		

SPARK MAX [2]

0.000000	Applied Output
0	Faults
0	Sticky Faults
0.000000	Velocity
1.000000	Velocity Conver
1.000000	Velocity Pre-Co
0.000000	Position
0.046664	Position Conver
25	Motor Temperatu
12.000000	Bus Voltage
0.000000	Motor Current
0.000000	Analog Voltage
0.000000	Analog Velocity
0.000000	Analog Position
0.000000	Alt Encoder Vel
0.000000	Alt Encoder Pos
-1	Control Mode
0.000000	Stall Torque
0.000000	Free Speed
132579587	FW Version

SPARK MAX [1]

0.000000	Applied Output
0	Faults
0	Sticky Faults
0.000000	Velocity
1.000000	Velocity Conver
1.000000	Velocity Pre-Co
0.000000	Position
1.000000	Position Conver
25	Motor Temperatu

```
Driver Joystick [0]  Operator Stick [1]  Joystick[2]  Joystick[3]  Joystick[4]  Joystick[5]
16: Keyboard 0       17: Keyboard 1      Unassigned   Unassigned   Unassigned   Unassigned
Axis[0] : 0.000      Axis[0] : 0.000
Axis[1] : 0.000      Axis[1] : 0.000
Axis[2] : 0.000      Axis[2] : 0.000
POVs[0] : -1         Axis[3] : 0.000
□□□□□□□□             Axis[4] : 0.000
□□                   Axis[5] : 0.000
                     □□□□□□□□
                     □□
```

□ FMS Attached
☑ DS Attached
Red 1 Alliance Station
-1.0 Match Time
Game Specific

To be able to “operate” the robot while any mode will work it is recommended that under robot state, Teleoperated is enabled.

Under the Joysticks there will be Joysticks 1-5 all grayed out, to select one go to System Joysticks and drag on over.

When starting up the program there will be no input method selected under System Joysticks there may be Joysticks, Xbox controllers, or Keyboards 0-3.

Joysticks and Xbox controllers work as if piloting an actual robot, to find layouts for controls see *Controller Map*.

Keyboard 0 is the WASD keys to move and E and R to rotate the controller buttons 1 through 4 can be access by Z,X,C,V in that order.

Keyboards 1-3 are differing controller schemes that are useless for the purposes of testing due to lack of movement.

Keyboard inputs require you to have the active window be the simulator while, Joysticks and xbox controllers don’t.

8.5 Dashboard

SOFTWARE DRIVETEAM

8.5.1 Dashboard Requirements

Under the `dashboard` directory setup a pipenv using the provided files and invoke the `make run` command to start the Dashboard's local host.

```
pipenv install # May take a while
pipenv shell
make run # makes a locally hosted website dashboard
```

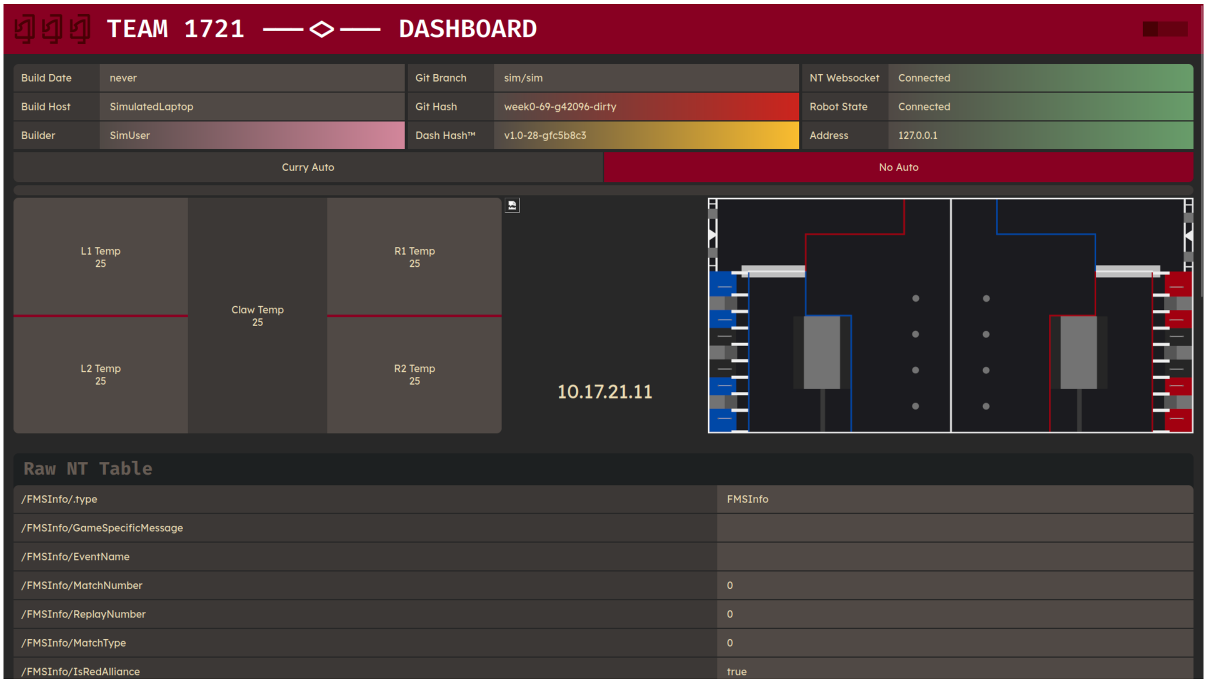
This will start the website dashboard. See *Using the Dashboard* for more info.

8.5.2 Using the Dashboard

Activating the dashboard doesn't automatically start. To start the Dashboard within your terminal it will print

```
cd www && python -m pynetworktables2js
16:32:28:683 INFO    : dashboard            : Connecting to NetworkTables at Ip address
16:32:28:683 INFO    : nt                   : NetworkTables initialized in client mode
16:32:28:683 INFO    : dashboard            : NetworkTables Initialized
16:32:28:684 INFO    : dashboard            : Listening on http://localhost:5800/
16:32:28:978 INFO    : tornado.access       : 101 GET /networktables/ws (Ip address) 0.
↪89ms
16:32:28:978 INFO    : net2js               : NetworkTables websocket opened
```

Within this you will see one called local host, you will need to copy that link and put it in a web browser. It is recommend that you use this in incognito due to network cache causing issues with not updating.



Build Date	never	Git Branch	sim/sim	NT Websocket	Connected
Build Host	SimulatedLaptop	Git Hash	week0-69-g42096-dirty	Robot State	Connected
Builder	SimUser	Dash Hash™	v1.0-28-gfc5b8c3	Address	127.0.0.1

L1 Temp 25

R1 Temp 25

Claw Temp 25

L2 Temp 25

R2 Temp 25

10.17.21.11

Raw NT Table

/FMSInfo/.type	FMSInfo
/FMSInfo/GameSpecificMessage	
/FMSInfo/EventName	
/FMSInfo/MatchNumber	0
/FMSInfo/ReplayNumber	0
/FMSInfo/MatchType	0
/FMSInfo/IsRedAlliance	true

This will auto connect to the simulation if one is running, if you start one after remember to refresh the page.

For this station to active things such as the swerve drive wheels, the simulation must be set to Teleoperated see *Using the Simulator*.

8.6 Robot code

SOFTWARE

8.6.1 Deploying code

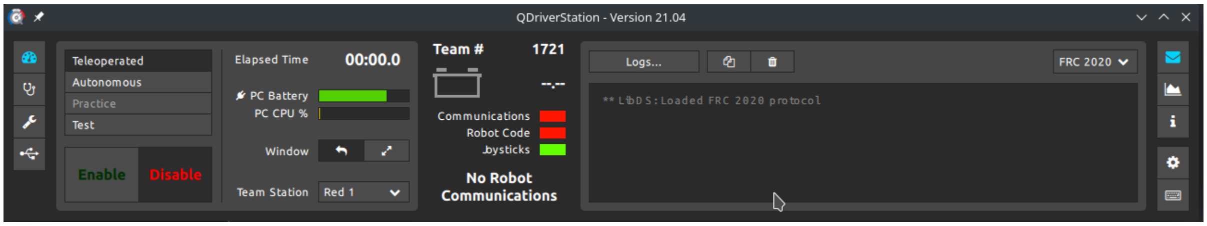
First remember to have the *FRC Driver Station* open first otherwise this won't work properly.

To deploy robot code first go to the `rio` setup a pipenv and invoke the `make deploy`, always run the *Robot Simulator* on code before deploying!

```
pipenv install # this may take some time
pipenv shell
make deploy # this run python3 robot.py deploy
```

First time running it will ask for a team number (1721). After the first time it will know the team number deploy the code.

8.6.2 FRC Driver Station



Elapsed Time 00:00.0

Team # 1721

Team Station Red 1

```
** LibDS: Loaded FRC 2020 protocol
```

Fig. 2: Q driver Station

The FRC Driver Station is made for windows, but there is a linux version Q Driver Station.

This will assume your using the windows version but they both mostly the same.

Most of this won't be in your control during competitions, but this will go over it all.

The base mode is Teleoperated but you can also set it to Autonomous, practice, and test.

Autonomous will simulate code that would be run in Autonomous. Practice will simulate competition timings, for testing use Teleoperated or in sim, see *Robot Simulator*. Test, tests the code a better way to test is *Robot Simulator*.

You can enable the robot after a joystick is detected, make sure to plug one in, it has robot code, and has communications with the robot.

Also see *Dashboard* for how to use the dashboard.

CHAPTER

NINE

THE EVENT CHECKLIST

9.1 Introduction

OUTREACH PITCREW

This is the checklist for events, this should be gone through with every event attended.

It is best to run through the night before to give for things to be found and secured in the trailer.

preferable this list should be checked at least twice in order to nto have any missing pieces.

9.2 Checklist

PITCREW

Table 1: Checklist

Item	Use	Checkbox
Drive Laptop	To deploy and run the robots code.	
Controllers	To control the robot.	
The Robot	It is our only way of participating.	
Tool Chest	To store the tools used in basic repairs.	
Chargers	Specifically the Laptop and Battery Chargers.	
Batteries	To power the robot.	
Battery Beak	Needed for checking batteries	

CHAPTER

TEN

BILL OF MATERIALS

OUTREACH

Table 1: Bill Of Materials

	Qty	Unit Cost	Total Cost
Electronics			
RoboRio 2	1	$485.00	$485.00
navX2	1	$107.00	$107.00
Drivetrain			
Arm			
			$592.00