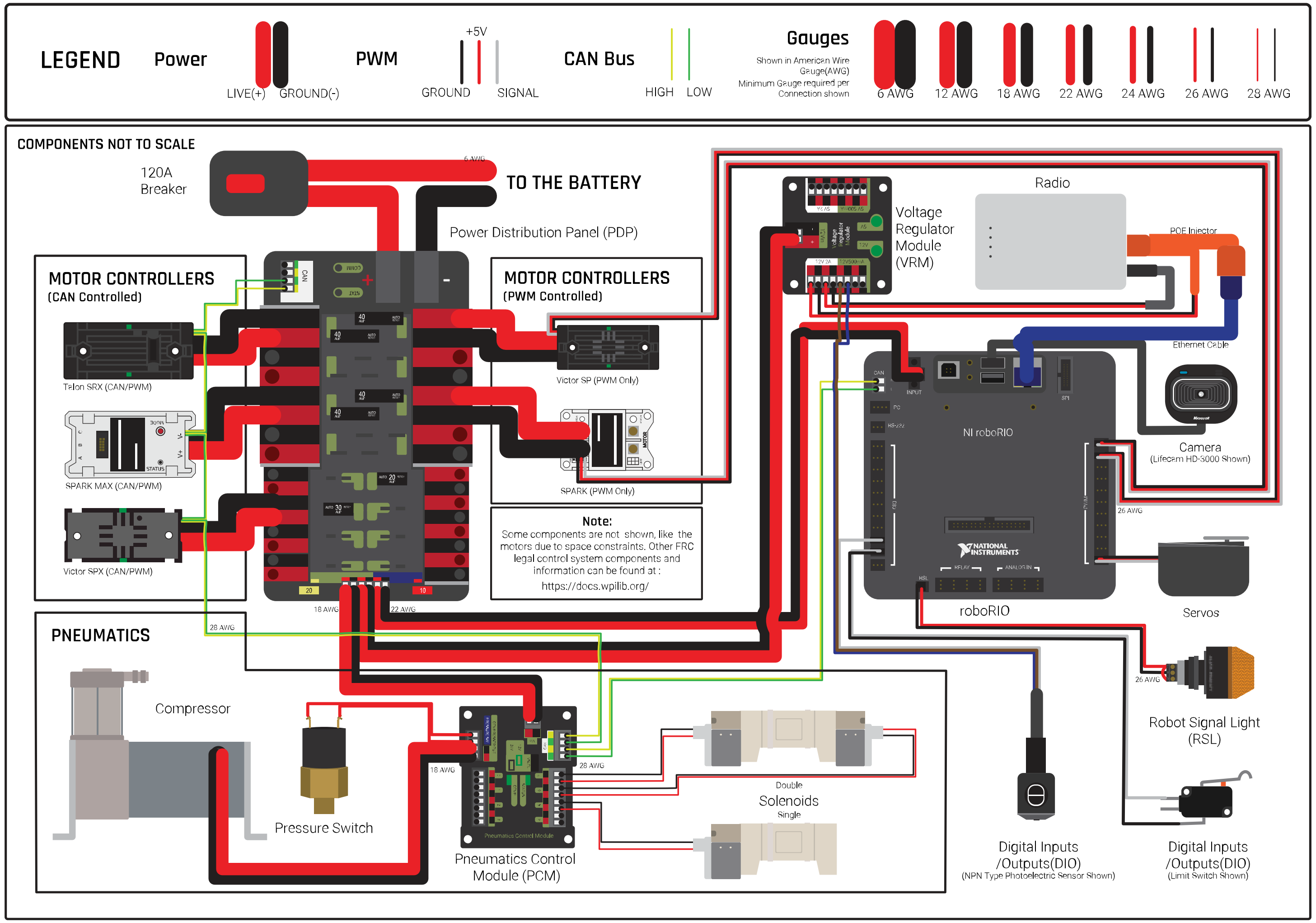
FRC Electrical System

Goals

Hopefully by the end of this PowerPoint this diagram will make sense to you!



Before we get started…

Please

Please

Please

Please

Please

Please

Read the manual – I don't know if this was already required of you, but if not, reading the manual will give you the true idea of what we need to do in this game. (I know its long and boring but I promise once you read it through you will be 1000x better equipped to tackle the game this year)

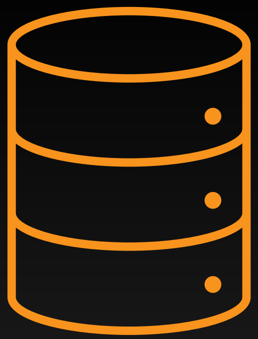
Also

Please join #control_systems on Slack. This will be our “Home base” to stay in touch with our projects and our learning, as well as where I will be posting the presentations from our meetings as well as other resources.

Our control systems team will have two main subgroups: programming and wiring. There are respective Slack channels #programming and #wiring for those interested. Please join which ever one you feel interests you, or feel free to join both!

Having active communication on Slack is going to be how our team can succeed in the difficult climate of remote learning / remote robot-ing, so don’t hesitate to ask questions or share things with each other in these channels.

How do we make a robot do something?



The data /information about what the robot should do

The energy for the robot to do the thing it has been told to do

“Something that can create some physical motion in its environment,” said Hadas Kress-Gazit, a roboticist and mechanical engineering professor at Cornell University. “It has the ability to change something in the world around you.”

How do we achieve this said physical motion?

Motors and Pistons

Voltage and Current and Resistance

"In electronics, we deal with three basic units. Voltage, current, and resistance. These are measured in Volts (V), Amps (A), and ohms (o). These are related by the equation V= IR, where V is voltage and A is amperage.

If we imagine electricity as water traveling through a pipe, electricity always flows from the high + to the low -. Voltage is the pressure applied to pump the water through, amperage is the flow rate of the water, and resistance is the size of the pipe.

This is why touching both terminals of a 12V battery with dry hands won't hurt the resistance of your hands is so high very little current can flow. It's also why licking a 12 V battery is a bad plan – I know from experience. Lots of current can flow and your tongue gets shocked.

A given electronic component (like a wire or a motor that is turning at a certain RPM) has a certain resistance. For more on circuits (such as parallel circuits vs. series circuits), check out the website hyperphysics."

-Taken from a presentation given by Arpan Rau, mentor on FRC 4979 and FRC 5125

Wire Guage

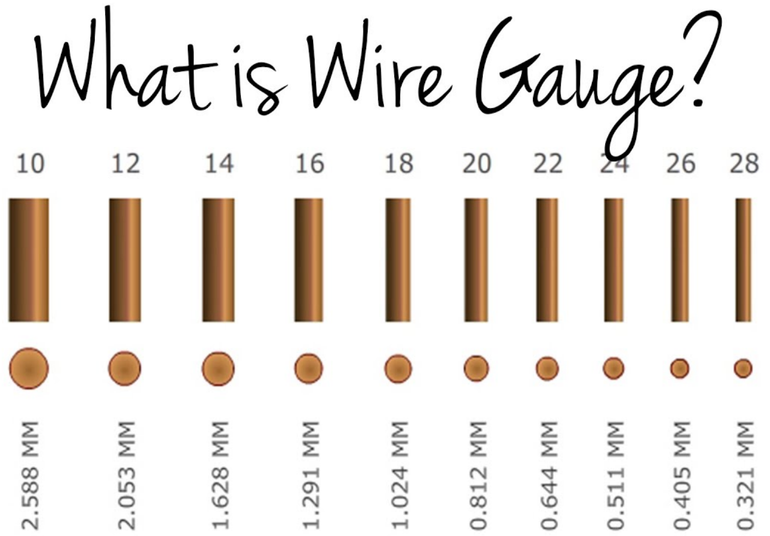
-The thicker a wire is, the more power it can carry without getting too hot.

-Wire thickness is measured in AWG. The smaller the guage number, the thicker the wire.

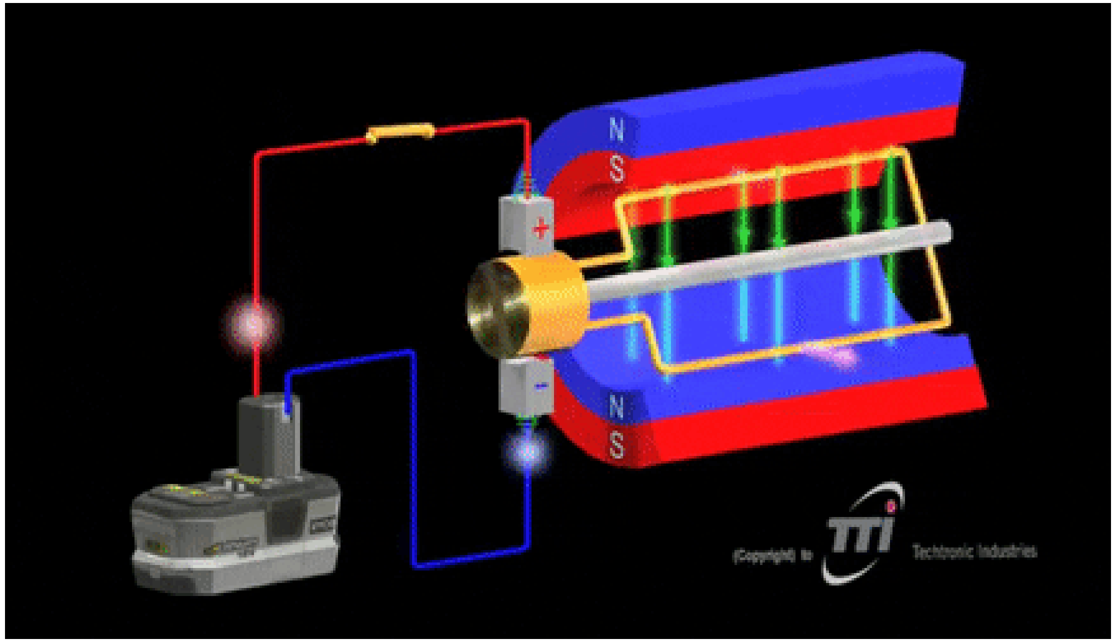
-Each standard guage number corresponds to a specific diameter.

-There is a lot more physics behind wire guage which is beyond the scope of FRC, but feel free to message me on Slack if you want to learn more.

-The rulebook contains a section on what wire guage is needed depending on how much current that circuit is capable of pushing. **READ THIS OVER!**



Direct Current Motors



What you actually need to know:

The output speed of the motor is linearly proportional to the input voltage, provided that the load (Torque) on the motor is constant. *

*Oversimplified but good enough for what we need to do

If you are interested in learning the physics behind how this works, let me know in Slack and I can send you some cool links to check out. (Not super relevant to robots but its neat)

Legal Motors

Complete Motor List

- AndyMark 9015 (am-0912)
- AndyMark NeveRest (am-3104)
- AndyMark PG (am-2161, am-2765, am-2194, am-2766)
- AndyMark RedLine Motor (am-3775, am-3775a)
- AndyMark Snow Blower Motor (am-2235, am-2235a)
- Banebots (am-3830, M7-RS775-18, RS775WC-8514, M5 – RS550-12, RS550VC-7527, RS550)
- CIM (FR801-001, M4-R0062-12, AM802-001A, 217-2000, PM25R-44F-1005, PM25R-45F-1004, PM25R-45F-1003, PMR25R-45F-1003, PMR25R-44F-1005, am-0255)
- CTR Electronics/VEX Robotics Falcon 500 (217-6515, 19-708850, am-6515, am-6515_Short)
- KOP Automotive Motors (Denso AE235100-0160, Denso 5-163800-RC1, Denso 262100-3030, Denso 262100-3040, Bosch 6 004 RA3 194-06, Johnson Electric JE-PLG-149)
- Nidec Dynamo BLDC Motor (am-3740, DM3012-1063)
- Playing With Fusion Venom (BDC-10001)
- REV Robotics NEO Brushless (REV-21-1650)
- REV Robotics NEO 550 (REV-21-1651)
- VEX BAG (217-3351)
- VEX Mini CIM (217-3371)
- West Coast Products RS775 Pro (217-4347)
- Electrical solenoid actuators, no greater than 1 in. (nominal) stroke and rated electrical input power no greater than 10 watts (W) continuous duty at 12 volts (VDC)
- Fans, no greater than 120mm (nominal) size and rated electrical input power no greater than 10 watts (W) continuous duty at 12 volts (VDC)
- Hard drive motors part of a legal COTS computing device
- Factory installed vibration and autofocus motors resident in COTS computing devices (e.g. rumble motor in a smartphone).
- PWM COTS servos with a retail cost < $75.
- Motors integral to a COTS sensor (e.g. LIDAR, scanning sonar, etc.), provided the device is not modified except to facilitate mounting
- One (1) compressor compliant with Rxx and used to compress air for the ROBOT'S pneumatic system

DC Motors that we care about

-775 Style (AndyMark RedLine and Vex Pro 775)

-CIM

-Mini CIM

-BAG

"DC motors, at their core, operate under a straightforward principle: the conversion of electrical energy (an input **voltage** and drawn **current**) into mechanical energy (an output shaft spinning at some **speed**, with some amount of **torque**)."

If you want to learn more about the physics behind choosing the right motor for the right application on a robot, which is more of a mechanical concept, you can read up on this topic at **motors.vex.com/introduction**

CIM Motor

-All of the slides on motors are not going to talk about the details from a torque and speed perspective of the motor, but rather, conceptually why we would select a particular motor and what we need to know electrically about that motor.

-Heaviest duty DC motor available in FRC (337W). Produces lots of torque and has a fairly low speed, and due to its large thermal mass, can run for a while without getting hot. However, once it does get hot, it takes a while to cool off.

-Very heavy, at 2.8lbs per motor. This means that a 6 CIM drivetrain (3 on each side) would use up a whopping 16.8lbs of your overall allowed weight (This is before gearboxes!)

-2x 12 AWG wires, one for + and one for –, come directly out of the motor casing, and can be connected directly to a DC speed controller (More on speed controllers later)

Mini-CIM Motor

-Basically a CIM motor but mini

- 2/3 of the power of a full sized CIM motor (215W), same speed output, same output shaft, same mounting hole pattern, same diameter but less length. Slightly less thermal mass means it heats up faster, but also cools down faster too

-Less heavy, at 2.16 lbs per motor. This means that a 6 CIM drivetrain (3 on each side) would use up a 12.96 lbs of your overall allowed weight (This is before gearboxes!)

-2x 12 AWG wires, one for + and one for –, come directly out of the motor casing, and can be connected directly to a DC speed controller (More on speed controllers later)

BAG Motor

-Basically a CIM motor but ultra mini (Actually not really but it sounds funny)

- Way less power than a CIM or a mini CIM (149W). A higher speed output at 13,180 rpm, compared to the 5,310 rpm of the CIM or mini CIM

-Physically much smaller than the CIM or mini CIM, with a different mounting hole pattern than the CIM or mini CIM due to its much smaller size. Also has a much smaller output shaft.

-Not good compared to the 775 for its size but has one advantage: Can stall for a lot longer before burning out.

-2x 14 AWG wires, one for + and one for –, come directly out of the motor casing, and can be connected directly to a DC speed controller (More on speed controllers later)

775 Style Motor


-Used to be the motor used for everything except for drivetrains before brushless motors became legal, but still is pretty damn awesome (More on this later)

-Only a little bit bigger than the BAG motor but has way more power (347W)

-Highest speed of our DC motor line-up (18,730 rpm)

-Can seat directly into these premade gearboxes sold by VEX that have a very small footprint, and are modular so you can add these stackable stages to change the gear ratio.

-Biggest drawback is that they do not survive for very long when stalled. (Motor dies and smoke comes out, becomes shiny paperweight after that. This is referred to as the "Magic smoke" in FRC)

-Has 2x male spade terminals on the rear, a female spade terminal mates with this connector

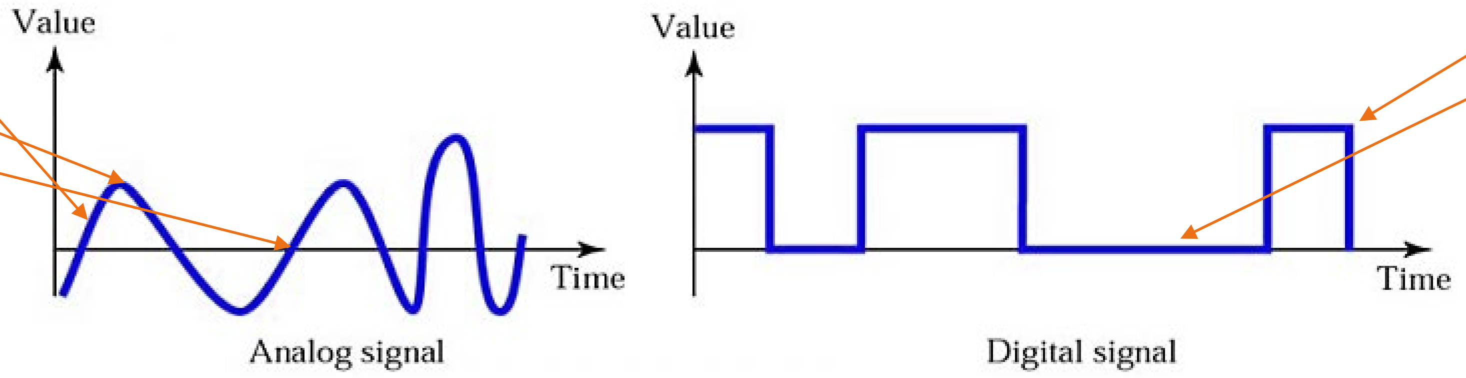
-Spade connectors suck. Highly recommended to use this device sold by VEX to convert female spade terminals w Anderson power pole connectors. (More on connectors later)

So if we want to run a 12v motor at 50% we just program the computer to output 6v right?

Well… of course it can't be that easy☺

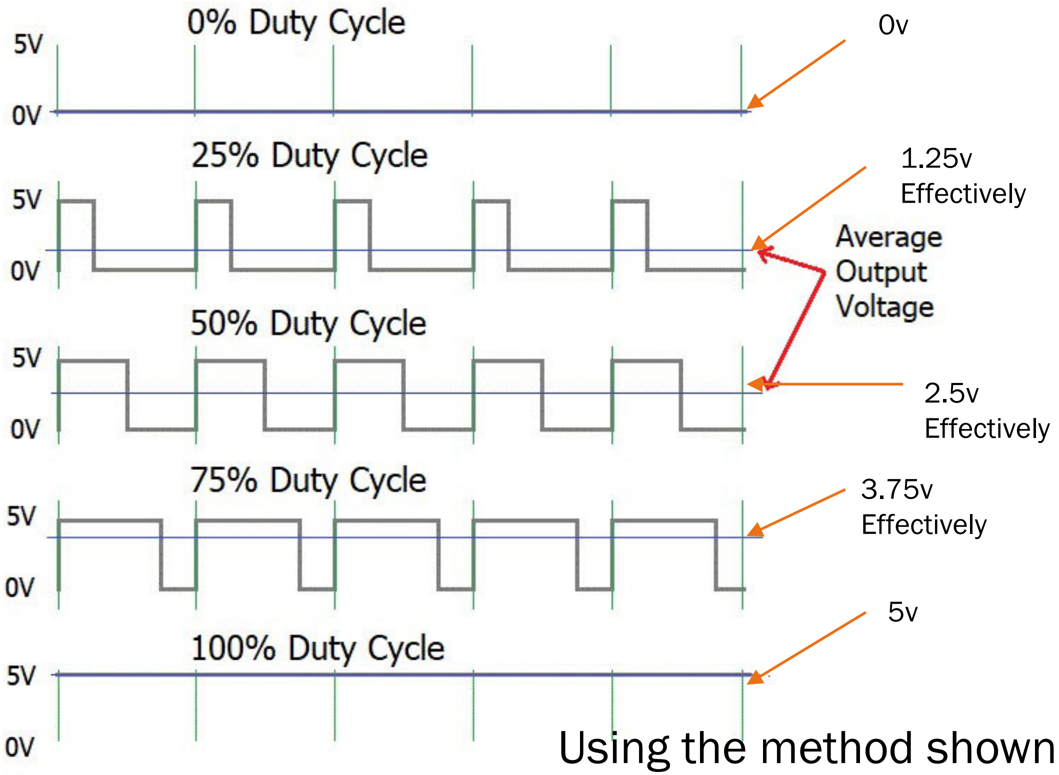
Outputting 6v from a 12v system requires an analog system, but we are dealing with digital electronics!

Output voltage can be any value within an allotted range!

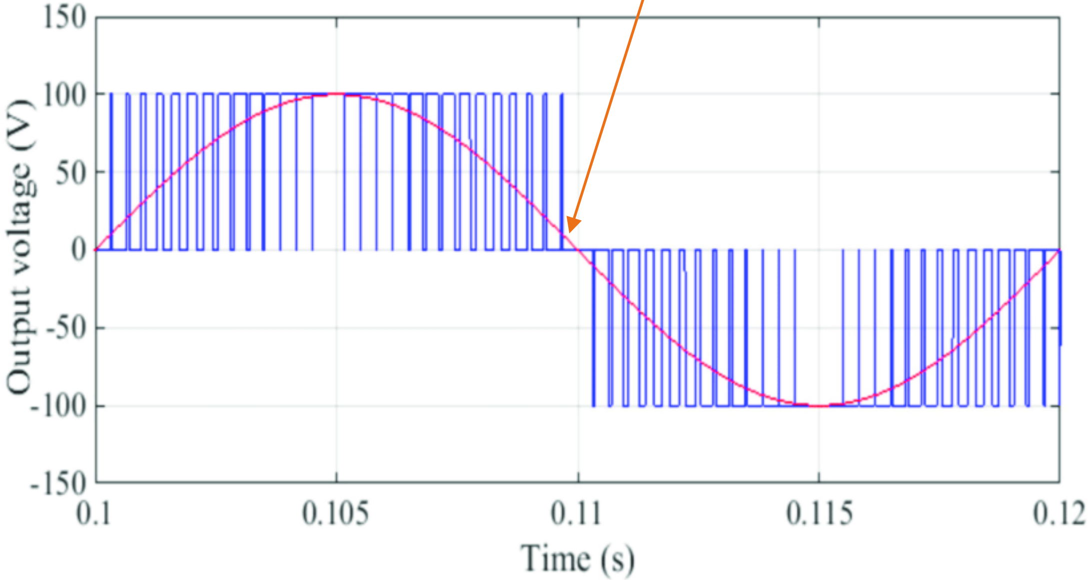


Everything needs to either be Max voltage (1) (On) or No voltage (0) (Off)

Solution: Pulse Width Modulation



Pink line here is the "effective" output voltage



Using the method shown on the left, we can create complex "effective" output voltages like the sine wave on the right

A 3-wire system is used: One white, one black, one red

So can we just plug a DC motor into our computer if we program the output voltage to be in PWM?

Well… once again of course it can’t be that easy ☺

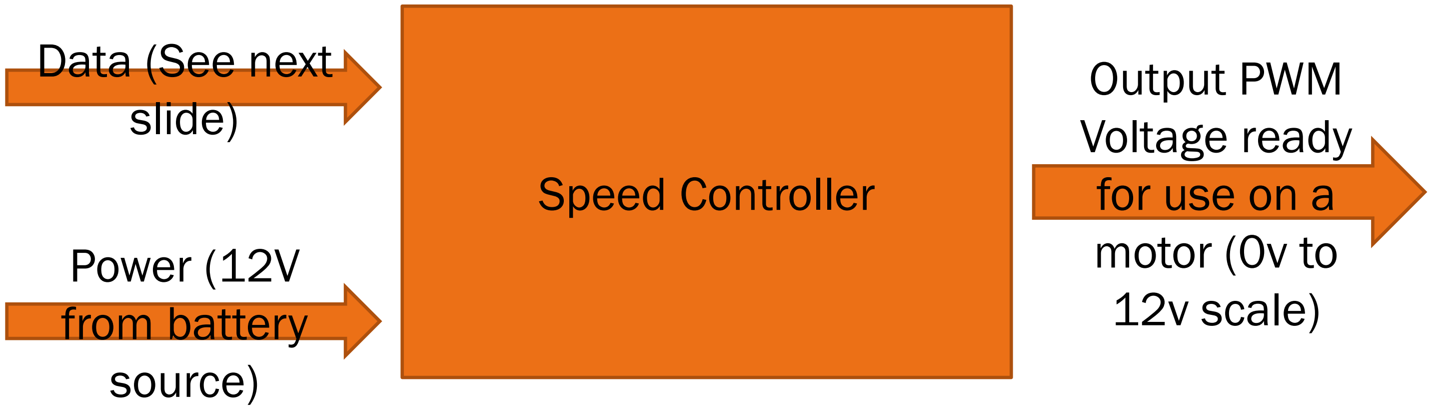
2 Problems:

1) Our computer outputs its digital voltages as either 0v or 5v, but our motors operate on a scale of 0v to 12v. This means that the fastest we would be able to run the motor would be at 41.6% speed!

2) Voltage isn’t the whole story here. Our robot’s computer can only output a veryvery very tiny amount of current (Much less than 1A), but our motors are going to need a whole lot more to run properly. (Sustained could be as high as like 30 or 40A, instantaneous could be as high as 120A!)

Solution: Speed Controller

A dedicated device with 2 inputs and 1 output. One input is for data, the other input is for power. On the output of the device is a PWM voltage that can feed lots of current (Amount depends on controller, but this is usually around the 40A or 60A threshold for continuous draw, and somewhere around 100A for instantaneous draw) and is on a 0v to 12v scale.



Speed Controller Input: Data

The speed controllers in FRC actually use two different communication protocols for the computer to communicate to the speed controller what speed it wants the motor to spin at. Some speed controllers only support one of the two methods, other controllers support both. These two methods are PWM and CAN. The PWM data protocol is exactly what we already talked about (From the robot’s computer it is going to be that 0v or 5v signal), but what about CAN?

CAN

Note from Monday Noah who is making this PowerPoint: I got too excited about CAN and got carried away over the next few slides. I'm going to skip them when presenting, but feel free to download this PowerPoint on your own from Slack and read it over if you are curious.

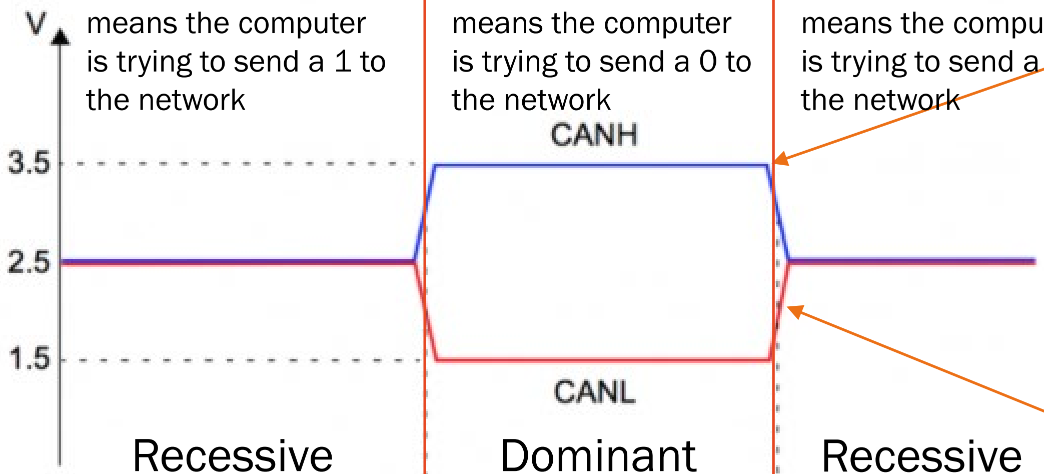
- A two-wire system (One yellow and one green)
- Allows for the device connected to "talk back" to the computer, instead of one-way communication like PWM
- Multiple devices can be connected to the network. (Basically for each speed controller that uses PWM as its data input, it will need its own wire connection directly from the computer to the speed controller. With CAN, the devices can be "daisy-chained" together and the two CAN wires can be run from the computer, to speed controller #1, and then directly to speed controller #2, and so on)

CAN Details (Skip me)

In this little section here, CANH is 0 and CANL is 0, which means the computer is trying to send a 1 to the network

In this little section here, CANH is 1 and CANL is 1, which means the computer is trying to send a 0 to the network

In this little section here, CANH is 0 and CANL is 0, which means the computer is trying to send a 1 to the network



Recessive

1

Dominant

0

Recessive

1

Wire #1's Voltage is shown in blue (Called CANH)

Some internal circuitry in the computer allows it to output a digital signal (typically a 5v for a 1 or a 0v for a 0) through CANH on a different scale (3.5v for a 1 or 2.5v for a 0)

Wire #2's Voltage is shown in red (Called CANL)

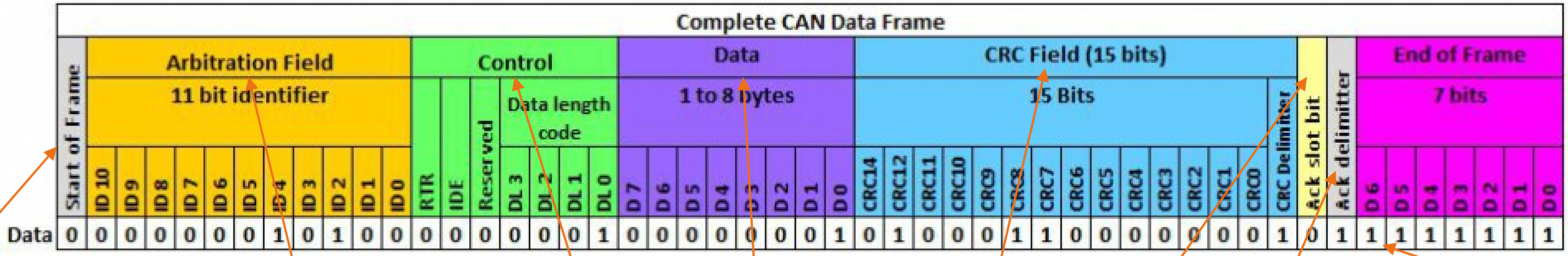
Some internal circuitry in the computer allows it to output a digital signal (typically a 5v for a 1 or a 0v for a 0) through CANL on a different scale (1.5v for a 1 or 2.5v for a 0)

This means that the CAN wires will have a value of 1 by default, and only switch to 0 if a device makes them switch by outputting the correct voltage to CANH and CANL

Why such a complicated method for sending a 1 or a 0? Well, this system basically makes it such that if a bunch of devices are connected to the same two wires, and one device tries to send a 0 at the same time another device tries to send a 1, the 0 will be the value that the wires take on. (This is due to some internal circuitry that exists inside every device plugged into the CAN network)

CAN Details (Skip me)

-So 99.9% of the time all of that complicated 1 / 0 representation stuff doesn’t really matter. This is because whenever a device on the CAN network wants to send a message out to the CAN network, it will wait for the network to be “clear” (Meaning that there are no other devices on the network that are in the middle of sending a message out to the network over the two wires)



As soon as a 0 is sent to the network, this is called the Start of Frame (SOF) bit, and indicates to all the other devices to not send any messages until they receive an EOF

This 11 bit field is where the sender of the message will tell the network who is sending the message. Each device has a unique ID number. The device with the lower ID number will take priority if multiple messages are trying to be sent at the exact same time

Doesn’t matter for our purposes

This is the important spot. These 8 bits contain the speed of the motor. This is represented in decimal from 0 to 255. This means that (for example) 128 (Or 10000000 in binary) would be 50%

Doesn’t matter for our purposes

Doesn’t matter for our purposes

Doesn’t matter for our purposes

End of Frame (EOF) Indicates to the network that the CAN bus is now available for a new message to be sent (Is always 7 1’s)

CAN Details (Skip me)

-But what about that rare occurrence when multiple devices try to send a message at different times?

-This is where the device ID being sent in the 11-bit arbitration field comes into play.

-In addition to sending out the desired bit a device wants onto the two wires, each device is also listening back on the two wires to see whether it has the value of a 1 or a 0.

-Since a 0 is dominant (If a 1 and a 0 are both sent to the wires at the same time, the 0 will be the one that ends up showing up on the wires), if a device tries to send a 1 but the wires instead take the value of a 0, this means another device was trying to write a bit at the same time but the 0 took priority. The device trying to send a 1 will now stop sending its message and will wait for the network to clear (EOF sent) before retrying its message.

-In binary, the lower a number is, the more 0s the number starts with. This means that when multiple devices are trying to send a message at the same time, and part of the message is the device's ID number, that the device ID with the most leading 0s (lowest number) will take priority because as the other devices send a 1 earlier in the 11-bit arbitration field, those 1s will be overwritten by 0s from the lower device number, and will realize this and stop sending their message.

Legal Speed Controllers

A. Motor Controllers

i. DMC 60/DMC 60c Motor Controller (P/N: 410-334-1, 410-334-2)
ii. Jaguar Motor Controller (P/N: MDL-BDC, MDL-BDC24, and 217-3367) connected to PWM only
iii. Nidec Dynamo BLDC Motor with Controller to control integral motor only (P/N 840205-000, am-3740)
iv. SD540 Motor Controller (P/N: SD540x1, SD540x2, SD540x4, SD540Bx1, SD540Bx2, SD540Bx4, SD540C)
v. Spark Motor Controller (P/N: REV-11-1200)
vi. Spark MAX Motor Controller (P/N: REV-11-2158)
vii. Talon FX Motor Controller (P/N: 217-6515, 19-708850, am-6515, am-6515_Short) for controlling integral Falcon 500 only.
viii. Talon Motor Controller (P/N: CTRE_Talon, CTRE_Talon_SR, and am-2195)
ix. Talon SRX Motor Controller (P/N: 217-8080, am-2854, 14-838288)
x. Victor 884 Motor Controller (P/N: VICTOR-884-12/12)
xi. Victor 888 Motor Controller (P/N: 217-2769)
xii. Victor SP Motor Controller (P/N: 217-9090, am-2855, 14-868380)
xiii. Victor SPX Motor Controller (P/N: 217-9191, 17-868388, am-3748)
xiv. Venom Motor with Controller (P/N BDC-10001) for controlling integral motor only

DC Motor Speed Controllers that we care about

-Spark Motor Controller

-Spark MAX Motor Controller

-Talon Motor Controller

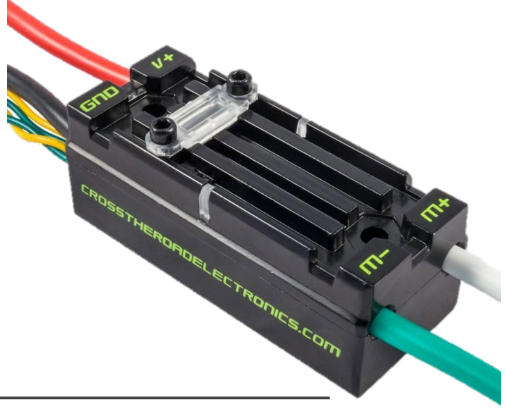
-Victor Motor Controller

Spark Speed Controller

-A little outdated and lacks a lot of the other features that some other speed controllers offer, but is good for the money.
-Does not support CAN input, only PWM
-Has limit switch input support. (Basically you can attach a up to 2 limit switches to the speed controller, and if limit switch 1 is triggered, the speed controller will refuse to run the motor backwards even if the PWM input signal tells the controller to do so. If limit switch 2 is triggered, the speed controller will refuse to run the motor forwards even if the PWM input signal tells the controller to do so. This can be useful because it is a feature that doesn't have to be added into code, and can make systems where you want the motor to "run until X" very simple to accomplish)
-Has a brake and coast mode togglable via button (When the computer says "don't rotate the motor, should the speed controller simply not send any power to the motor, or should it send power to the motor in such a way that "fights" the motor from spinning?")
-Uses "Yellow" sized ring connectors for power input and for output.
-Lightweight
-Has four mounting holes to be held in place

Spark MAX Motor Controller

-Primarily used for the Neo or Neo 550 Brushless Motor, but has a feature that allows it to be used with DC motors
-More expensive than the Spark, but has a LOT of cool features
-Supports both PWM and CAN for inputs
-Can be plugged into a laptop via USB C and can be commanded to output at different levels based on a very easy to use software utility, allowing for motor testing without needing to write robot code
-In addition to limit switch input, it can also be used with an encoder to keep track of how many rotations a motor has completed, and can be set to automatically stop driving the motor in a particular direction if the encoder has rotated too many times in a specific direction
-Has a closed loop mode, allowing the computer to instruct the motor controller to do things such as "Rotate the motor 50 turns clockwise" and the motor controller will look at encoder data and precisely rotate the motor a specific number of rotations or partial rotations based on sensor data
-Has a brake and coast mode controllable through commands sent over CAN in robot code
-Has 2x 12 AWG wires for power, and 3x 12 AWG wires for output, although only 2x are used for DC motor mode



Talon Speed Controller

-Supports both CAN and PWM

-Has limit switch input support, as well as support for encoder input

-Has a brake and coast mode togglable via software commands

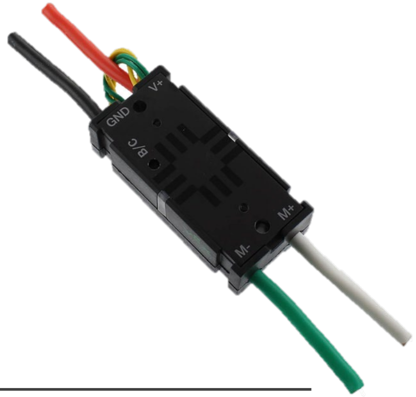
-Has 2x sets of 2x 12 AWG wires for input and for output.

-Has two mounting holes to be held in place

-Has support for closed loop operation

-Can be used as a master and can command any number of Victor speed controllers on the same CAN network to behave identically to it (Basically allows any Victors on the same gearbox as it to operate in closed loop operation or to have limit switch input support or encoder input)

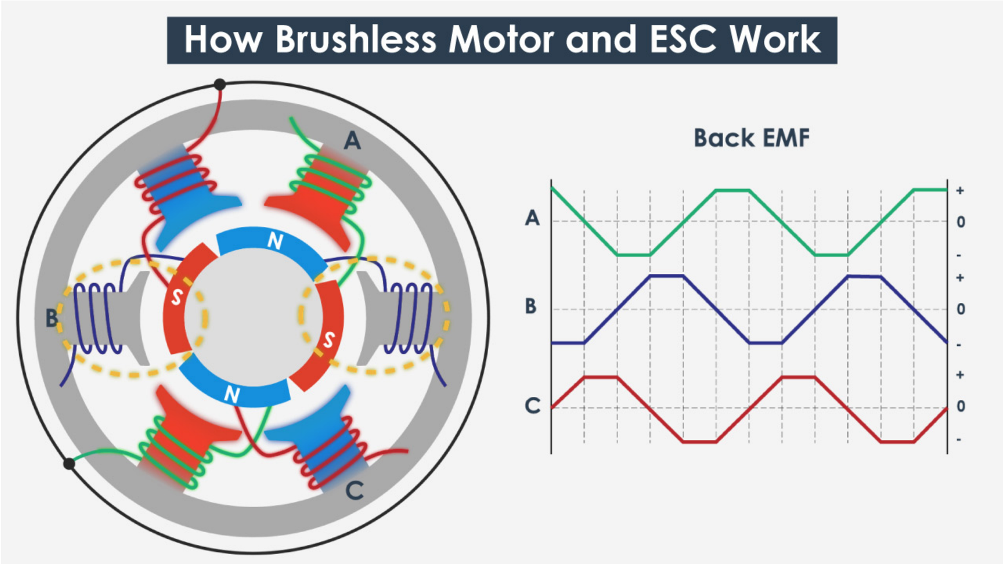
Victor Speed Controller



-Basically a Talon but cheaper and no support for closed loop operation, or for limit switch or encoder input

-Weighs less than a Talon

Brushless Motors



Once again, if you are interested in learning the physics behind how this works, let me know in Slack and I can send you some cool links to check out. (Not super relevant to robots but its neat)

What you need to know:

-Three wires are used instead of two and applying a DC voltage across any of the two wires WILL NOT make the motor spin (Unlike a DC motor you can't just directly connect the wires to a 12V battery and have the motor run).

-A brushless motor requires a motor controller to apply the proper voltage on the proper wire at the proper time. This is achieved through sensors inside the motor that communicate back to the motor controller about the rotation of the output shaft so that the speed controller can figure out what voltage needs to be applied on what wire and when.

* Once again, oversimplified but good enough for what we need to do

Brushless vs Brushed (DC)

Directly from our good friends over at Wikipedia:

"Brushless motors offer several advantages over brushed DC motors, including ***high torque to weight ratio****,* ***more torque per watt*** *(increased efficiency),* ***increased reliability****, reduced noise,* ***longer lifetime*** *(no brush and commutator erosion), elimination of ionizing sparks from the commutator, and an overall reduction of electromagnetic interference (EMI)."*

For FRC purposes – brushless motors are much more powerful than an equivalently weighing DC motor, and come with an additional benefit of being much harder to "smoke" under stall conditions.

Legal Brushless Motors

Complete Motor List

- AndyMark 9015 (am-0912)
- AndyMark NeveRest (am-3104)
- AndyMark PG (am-2161, am-2765, am-2194, am-2766)
- AndyMark RedLine Motor (am-3775, am-3775a)
- AndyMark Snow Blower Motor (am-2235, am-2235a)
- Banebots (am-3830, M7-RS775-18, RS775WC-8514, M5 – RS550-12, RS550VC-7527, RS550)
- CIM (FR801-001, M4-R0062-12, AM802-001A, 217-2000, PM25R-44F-1005, PM25R-45F-1004, PM25R-45F-1003, PMR25R-45F-1003, PMR25R-44F-1005, am-0255)
- CTR Electronics/VEX Robotics Falcon 500 (217-6515, 19-708850, am-6515, am-6515_Short)
- KOP Automotive Motors (Denso AE235100-0160, Denso 5-163800-RC1, Denso 262100-3030, Denso 262100-3040, Bosch 6 004 RA3 194-06, Johnson Electric JE-PLG-149)
- Nidec Dynamo BLDC Motor (am-3740, DM3012-1063)
- Playing With Fusion Venom (BDC-10001)
- REV Robotics NEO Brushless (REV-21-1650)
- REV Robotics NEO 550 (REV-21-1651)
- VEX BAG (217-3351)
- VEX Mini CIM (217-3371)
- West Coast Products RS775 Pro (217-4347)
- Electrical solenoid actuators, no greater than 1 in. (nominal) stroke and rated electrical input power no greater than 10 watts (W) continuous duty at 12 volts (VDC)
- Fans, no greater than 120mm (nominal) size and rated electrical input power no greater than 10 watts (W) continuous duty at 12 volts (VDC)
- Hard drive motors part of a legal COTS computing device
- Factory installed vibration and autofocus motors resident in COTS computing devices (e.g. rumble motor in a smartphone).
- PWM COTS servos with a retail cost < $75.
- Motors integral to a COTS sensor (e.g. LIDAR, scanning sonar, etc.), provided the device is not modified except to facilitate mounting
- One (1) compressor compliant with Rxx and used to compress air for the ROBOT'S pneumatic system

Neo 550 Motor

-Super tiny yet has an output power like the 775. (279W). Its super small size means that it can fit in all sorts of tricky places on a robot. 1.752” length, 1.378” body diameter.

-Insanely light wight. 0.313 lbs! Super useful for intakes and other similar robot mechanisms.

-Very high free speed of 11000 RPM.

-4x metric M3 holes on the face of the motor for mounting

-Built in encoder that is readable from robot code allows for precise rotation measurement as well as closed loop operation.

-3x 16 AWG wires, one red, one white, and one black come out of the motor casing, and can be connected directly to a Spark Max speed controller.

Neo Motor

-Incredibly powerful motor designed to be used in place of a CIM or Mini CIM motor. (380W) Has the same mounting holes as a CIM, so can be used as a “drop-in” replacement. Same body diameter, but shorter at 2.3” length.

-Mostly going to be used for drivetrains, elevators, and shooters.

-Much lighter than a CIM at 0.938 lbs.

-Similar free speed to a CIM or Mini CIM at 5676 rpm.

-Built in encoder that is readable from robot code allows for precise rotation measurement as well as closed loop operation.

-3x 12 AWG wires, one red, one white, and one black come out of the motor casing, and can be connected directly to a Spark Max speed controller.

Falcon 500 Motor

-Incredibly powerful motor. (406W)

-Mostly going to be used for drivetrains, elevators, and shooters.

-Much lighter than a CIM at 1.1 lbs

-Higher free speed than CIM or Mini CIM at 6380 rpm.

-Built in encoder that is readable from robot code allows for precise rotation measurement as well as closed loop operation.

-Has a motor controller built into the motor casing, allowing for simpler wiring and less weight used for a separate motor controller.

-2x 12 AWG wires, one red, and one black come out of the motor casing, and can be connected directly to the ***12V supply on the PDP.***

Legal Brushless Speed Controllers

A. Motor Controllers

i. DMC 60/DMC 60c Motor Controller (P/N: 410-334-1, 410-334-2)
ii. Jaguar Motor Controller (P/N: MDL-BDC, MDL-BDC24, and 217-3367) connected to PWM only
iii. Nidec Dynamo BLDC Motor with Controller to control integral motor only (P/N 840205-000, am-3740)
iv. SD540 Motor Controller (P/N: SD540x1, SD540x2, SD540x4, SD540Bx1, SD540Bx2, SD540Bx4, SD540C)
v. Spark Motor Controller (P/N: REV-11-1200)
vi. Spark MAX Motor Controller (P/N: REV-11-2158)
vii. Talon FX Motor Controller (P/N: 217-6515, 19-708850, am-6515, am-6515_Short) for controlling integral Falcon 500 only.
viii. Talon Motor Controller (P/N: CTRE_Talon, CTRE_Talon_SR, and am-2195)
ix. Talon SRX Motor Controller (P/N: 217-8080, am-2854, 14-838288)
x. Victor 884 Motor Controller (P/N: VICTOR-884-12/12)
xi. Victor 888 Motor Controller (P/N: 217-2769)
xii. Victor SP Motor Controller (P/N: 217-9090, am-2855, 14-868380)
xiii. Victor SPX Motor Controller (P/N: 217-9191, 17-868388, am-3748)
xiv. Venom Motor with Controller (P/N BDC-10001) for controlling integral motor only

Spark MAX Motor Controller (Again)

-Used for the Neo or Neo 550 Brushless Motor

-Can be plugged into a laptop via USB C and can be commanded to output at different levels based on a very easy to use software utility, allowing for motor testing without needing to write robot code

-Allows for use of built-in encoder on Neo or Neo 550 Motor

-Has a closed loop mode, allowing the computer to instruct the motor controller to do things such as “Rotate the motor 50 turns clockwise” and the motor controller will look at encoder data and precisely rotate the motor a specific number of rotations or partial rotations based on sensor data

-Has a brake and coast mode controllable through commands sent over CAN in robot code

-Has 2x 12 AWG wires for power, and 3x 12 AWG wires for output

Integrated Controller on Falcon 500 Motor

-Honestly not much to say here because the speed controller is built right into the motor

-You don't have to worry about wiring the speed controller to the motor, you just apply power to the motor and supply data over CAN

-Simple ☺

Pistons

What you need to know:

-Two ports on either end that you connect to the Solenoid (more on this next) to control it.

-The rod at the end extends out or retracts in depending on the pressure inside (controlled by the Solenoid)

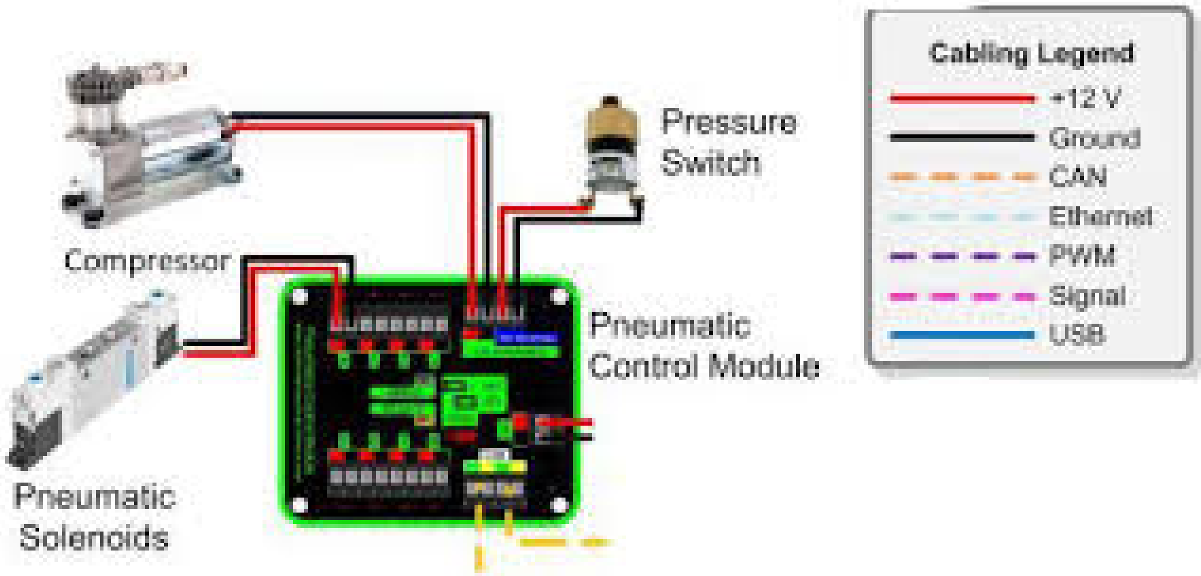
-Great for providing linear motion (When you need something extended or retracted in a straight line) in a very simple way

-Incredibly easy to control from a programming view

Solenoids

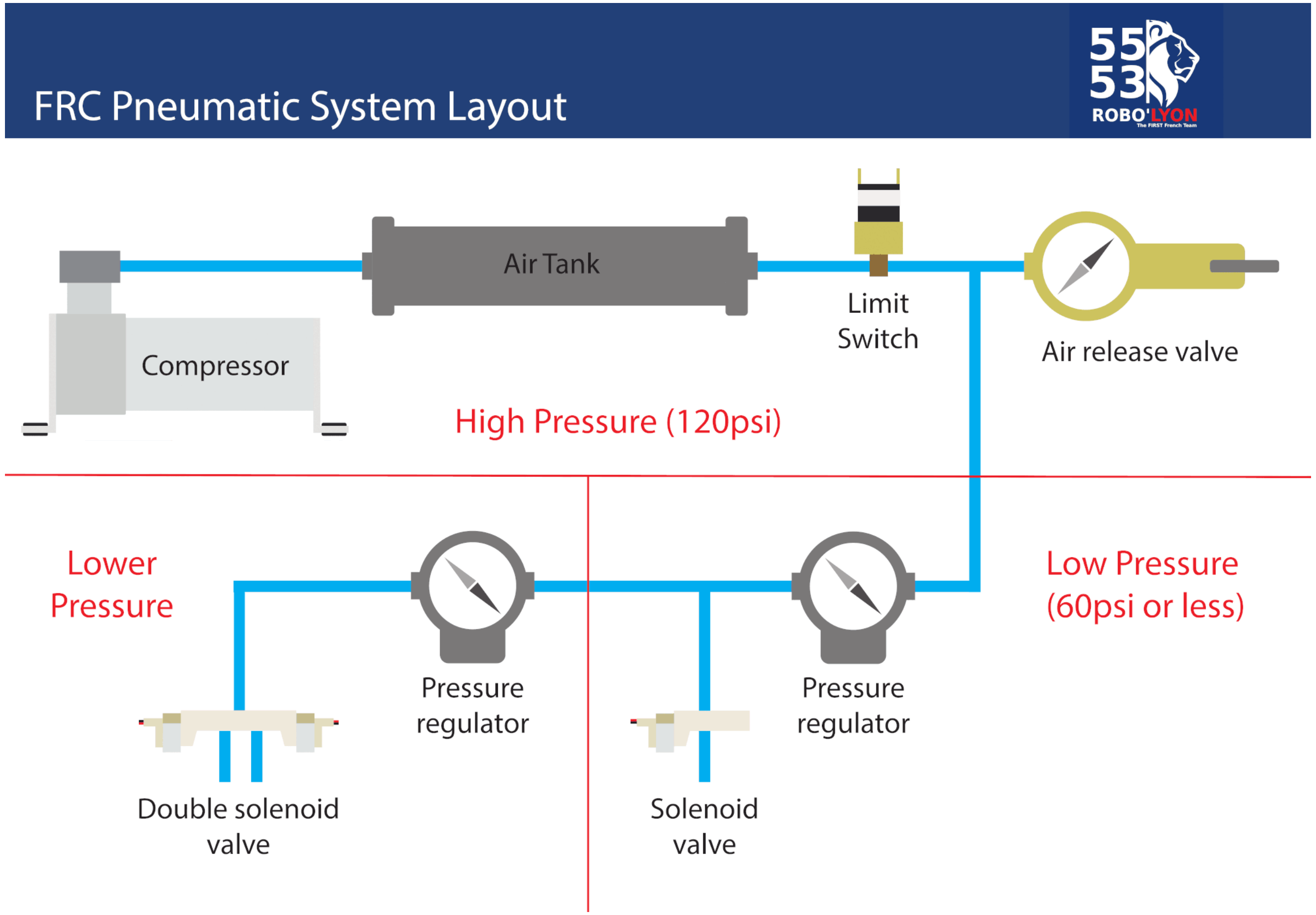
- Two outputs, to be connected up to the two ports on the piston
- A single input where air pressure is applied to (Usually a tank holds air pressure and is connected to this port)
- The wires connect to the PCM (Pneumatics Control Module)
- The PCM is sent data over CAN from the roboRIO and is told when to make a solenoid open or close (causing the corresponding piston to either extend or retract)

PCM (Pneumatics Control Module)



- Allows the roboRIO to control up to 8 Solenoids
- 12V power is applied to the PCM from the PDP
- Solenoids are connected to the 8 ports on the PCM
- An air compressor is connected to the PCM, which will build air pressure inside of the storage tanks for use in extending and retracting the pistons
- A pressure switch is connected to the air hose network, telling the PCM when the pressure in the system has reached a high enough value and that it is good to turn the compressor off
- CAN allows the roboRIO to instruct the PCM when to signal certain Solenoids to open or close

Pneumatic System



PDP (Power Distribution Panel)

- 12V Power from the battery comes in through 6 AWG wire
- Supplies 12V power to all of the other devices on the robot that require 12V power
- Each device that needs 12V power gets connected to the PDP and has its own circuit with breaker to prevent too much current being used on that circuit
- Circuit breaker sizes of 40A, 30A, 20A, and 15A are available to use for each circuit
- Half of the available circuits allow for 12 AWG wire and up to 40A breakers to be used (Typically for high current applications like drivetrain or shooter)
- The other half allow for 16 or 18 AWG wire and up to 30A breakers to be used (For elevators, intakes, or other lower current systems)

PDP (Power Distribution Panel)

- At the very bottom, 18 AWG ports with a 20A breaker supply power to the PCM and to the VRM
- Also at the very bottom, 22 AWG ports with a 10A breaker supply power to the roboRIO
- Connects to the CAN network and allows the roboRIO to read information like real-time current draw per circuit (Super useful for automated intake or hopper systems)

VRM (Voltage Regulator Module)

-Gets its input power from the 18 AWG run to the PDP over the 20A circuit at the bottom of the unit

-Outputs regulated 5V or 12V power to devices

-The radio’s barrel connector gets connected to the 12V 2A circuit. The POE injector also connects to the other 12V 2A circuit

-You might connect LEDs or a Raspberry Pi to this

What is regulated power?

-Under zero load (no current draw) a 12V FRC battery is going to have a voltage somewhere around 14V

-When we first start drawing current from the fully charged battery to power our robot's electronics, the battery voltage will drop down to the 9V to 12V range

-As the battery begins to run out of stored power, our normal current draw will have more of an impact on the battery output voltage (Instead of dropping from our desired 12V down to 9V, it may drop down to 8V, 7V, perhaps even 6V!)

-When the battery voltage under current load leaches 6.3V, the robot begins to "brownout". This means the roboRIO will momentarily stop motor controllers to reduce current draw, until battery voltage is increased back up to 6.3V.

-This is why during a robot brownout, the motors will rapidly stop and start over and over again.

-Power coming out of the VRM is "regulated" to be guaranteed to be 12V or 5V, not a range from 14V to 6.3V!

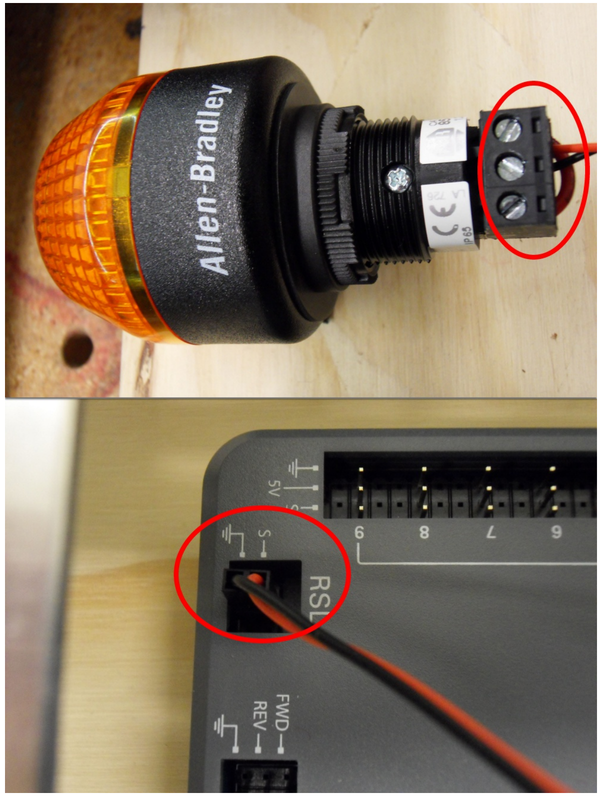
-VRM is used to power sensitive electronics that can't deal with voltage spikes or drops

Radio

- The robot radio is pretty much a WiFi router that you would find in your home or school
- Has 2 different modes: practice and competition
- 90% of the time you will have it in practice mode, as to switch into competition mode you have to plug into a special laptop at an FRC event
- Practice mode allows you to connect your laptop to a WiFi network that allows data to be passed between the laptop and the robot
- Gets its power through a barrel connector
- Can also be powered through POE by using a POE injector
- Both POE injector and barrel connector should be used for redundancy
- Yes, you can choose the name of the WiFi network broadcasted ☺
- Has two ethernet ports on it. One should connect to the POE injector, while the other is either is left for an additional networked device.

POE Injector Cable

The RSL AKA “The Idiot Light”



-A light that per the rule book must be visible on your robot at all times

-Isn’t programmed by the programmer, but is controlled automatically by firmware on the roboRIO

-Connects to the RSL port on the roboRIO

-Off when robot is powered off, on when robot is powered on but is disabled in code, blinks when robot is powered on but is enabled in code

Main Breaker

-One end connects to the positive end of the battery, the other end is connected to the positive end of the PDP
-Acts as the “On / Off” switch of an FRC robot
-Press the red button to turn off
-Slide the black bar towards the top to turn on
-Is a resettable circuit braker that will pop at 12 seconds of 240A, 4 seconds of 360A (May be lower time if hot)
-If this thing ever pops then that means there is usually some serious issues to diagnose

The roboRIO

- The computer on the robot that controls it all
- Connects to the PDP for 12V power through the INPUT port
- Connects to the CAN network through the CAN port
- Connects to the laptop to push code through the USB type B port
- Connects to a USB webcam through the USB ports
- Connects to the POE injector (which connects to the Radio) through the ethernet port
- Connects to the RSL light through the RSL port
- Connects to sensors like limit switches or reed switches through the DIO ports
- Connects to sensors like ultrasonic, potentiometer, encoder, or light sensor through the ANALOG IN ports
- Connects to PWM speed controllers through the PWM ports

The End ☺

PLEASE Reach out to either of the control systems mentors over Slack if you have any questions!