

Logic

Table of Contents:

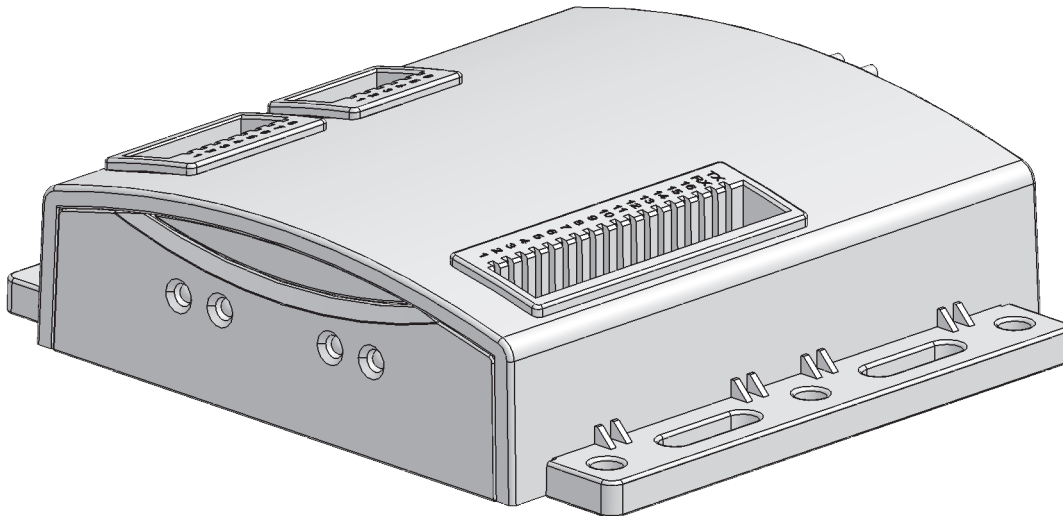
Introduction to the Logic Subsystem	7.2
Concepts to Understand	7.10
Subsystem Interactions	7.12

Introduction to the Logic Subsystem

The VEX Microcontroller coordinates the flow of information and power on the robot. All other electronic system components must interface to the Microcontroller and function as a cohesive unit.

It helps to assemble the Logic Subsystem last when building your robot, because all the plugging-in of components will take place in this section. It would make the most sense if those parts were already attached to the robot before plugging them in.

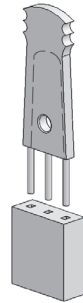
Microcontroller Module



Logic Subsystem

Jumpers

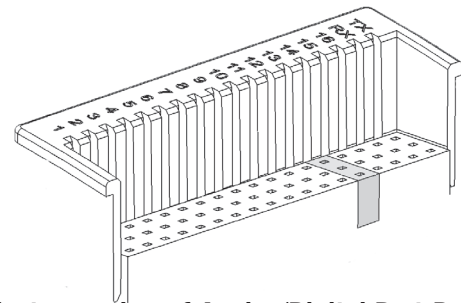
Jumpers are small plastic pieces with a metal connector inside them. When inserted into a port, a jumper completes an electrical connection between two of the metal contacts inside, like closing a switch in a circuit. In fact, software running on the Microcontroller cannot distinguish between a permanently pressed switch and an inserted jumper, nor can it distinguish between a released switch and a removed jumper. The VEX jumper clips are 3 holes wide because they join the two outer contacts in the port and skip the middle one (they don't join all three, that would cause a short circuit).



Jumpers are most commonly used to set an option on the Robot Controller by providing a Ground or Low signal on the port selected. Placing a jumper on a given Analog/Digital port will typically activate a certain behavior in the robot's programming (see Port Mappings later in this section).

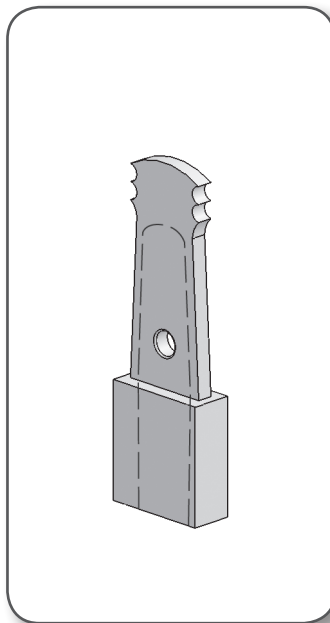
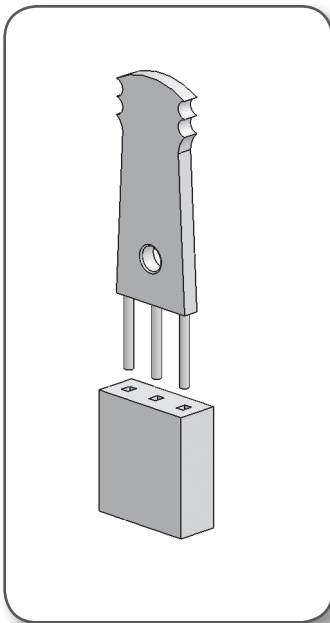
How to install a jumper

First, find the port you would like to place the jumper on. Remember that a port is a set of three holes running horizontally next to the port number. A port bank (like the Analog/Digital Port Bank) consists of a collection of ports.



(cutaway view of Analog/Digital Port Bank)

Once you have found the port, all you need to do is insert the jumper so that the metal pins go into the three holes, and the jumper itself sits securely in position over the port. It does not matter which way you install the jumper.



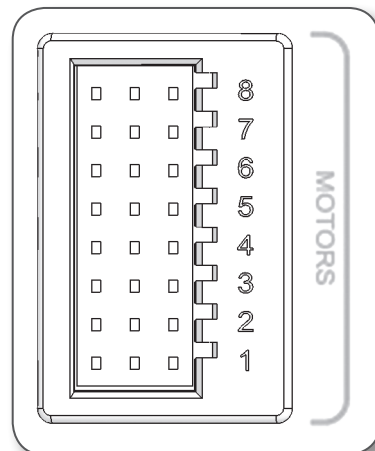
Logic Subsystem, continued

Port Mappings - Ports on top of the controller

MOTORS

These ports are for motors or servos only. By default, your robot's left drive motor goes on Port 3, and your right drive motor goes on Port 2. However, there are a number of alternate control configurations available: see "Basic Controller Operation" in the Control Subsystem chapter for basic configurations, and "Advanced Operator Control Configurations" at the end of this chapter for more advanced options (including the 4WD mode mentioned in the chart below).

Shown here are the motor port allocations for the two "Basic" control layouts. For more details consult the "Control" Appendices at the end of this manual.



	DEFAULT "23 mode" Control Tank Style	"12 mode" Control Arcade Style
Motor Port 1	*	LEFT MOTOR
Motor Port 2	RIGHT MOTOR	RIGHT MOTOR
Motor Port 3	LEFT MOTOR	*
Motor Port 4	*	*
Motor Port 5	*	*
Motor Port 6	*	*
Motor Port 7	Right Rear Motor in 4WD mode **	Right Rear Motor in 4WD mode ***
Motor Port 8	Left Rear Motor in 4WD mode **	Left Rear Motor in 4WD mode ***

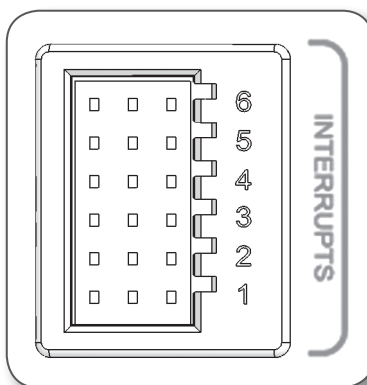
* The inventor may use these motor ports to control inventor designed accessories.

** Only with Jumper 15 set on the Microcontroller.

*** Only with Jumper 16 set on the Microcontroller.

INTERRUPTS:

Interrupts are digital inputs designed for high priority signals that need immediate attention from the Microcontroller. These are used with some of the advanced sensors of the VEX Robotics Design System, such as the Ultrasonic Range Finder or Quadrature Shaft Encoder.



Logic Subsystem, continued

Digital/Analog Input Ports:

This is a bank of 16 multipurpose input/output ports. In the pre-programmed configuration, most of these ports (1-12) are used as sensor inputs, and have behaviors associated with them that are activated when a sensor is plugged in and activated. The remainder are used to configure advanced settings on the controller by setting or removing jumpers.

Jumper Ports

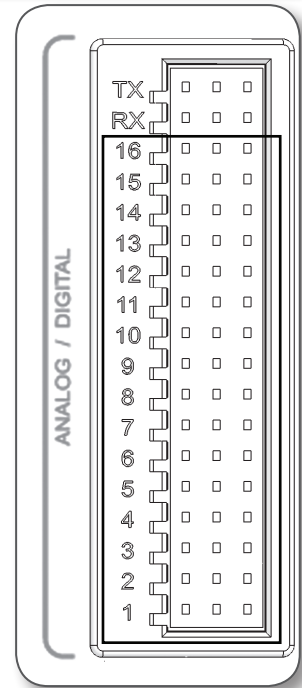
Port	Function	For More Information
Jumper Port 16	"12 Mode" 4-Wheel Drive Enable/Disable	Advanced Operator Control Configurations section in this chapter
Jumper Port 15	"23 Mode" 4-Wheel Drive Enable/Disable	Advanced Operator Control Configurations section in this chapter
Jumper Port 14	"Software 12 mix Mode" Enable/Disable	Advanced Operator Control Configurations section in this chapter
Jumper Port 13	Autonomous Mode Enable/Disable	Autonomous Mode section in this chapter

Sensor Ports

Port	Function	For More Information
Sensor Port 12	Autonomous Mode Collision Detection Sensor Ports	Autonomous Mode section in this chapter
Sensor Port 11		
Sensor Port 10	Collision Emergency Stop Ports	Sensor Subsystem Chapter
Sensor Port 9		
Sensor Port 8	Limit Switch Ports	Sensor Subsystem Chapter
Sensor Port 7		
Sensor Port 6		
Sensor Port 5		
Sensor Port 4		
Sensor Port 3		
Sensor Port 2		
Sensor Port 1		

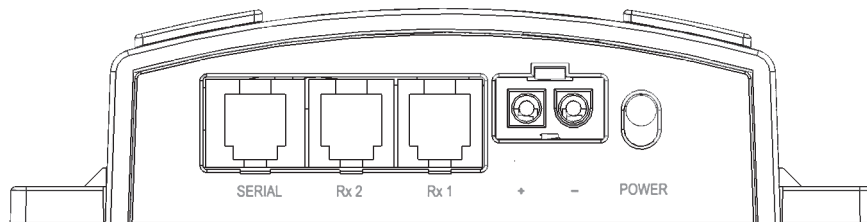
TX/RX Ports

At the end of the Digital/Analog Port Bank are two ports marked TX and RX. These ports allow the user to communicate directly with the user processor inside the Microcontroller via a TTL serial port. This is a feature for advanced users only and is not needed for normal operation.



Logic Subsystem, continued

Port Mappings - Ports on the back panel of the controller



Serial (Programming) Port

This port is for use with the VEX Programming Kit. This port allows the user to communicate with the processor inside the Microcontroller. One can hook a computer up to the VEX Microcontroller through this port to either download code into the Microcontroller, or to receive feedback from the Microcontroller. Both the User and master processors can be programmed from this port.

Rx1 and Rx2 (Radio Receiver) Ports

These ports are for radio receivers to communicate with the Microcontroller. The VEX Microcontroller supports up to two receivers simultaneously, allowing two operators to control different parts of the same robot at the same time. One operator will usually focus on maneuvering the robot, while the other operator runs the other onboard attachments and equipment. This two operator mode requires 2 different crystal sets, a TX/RX pair for each operator.

Rx1: Primary operator (driver) port

Rx2: Secondary operator port

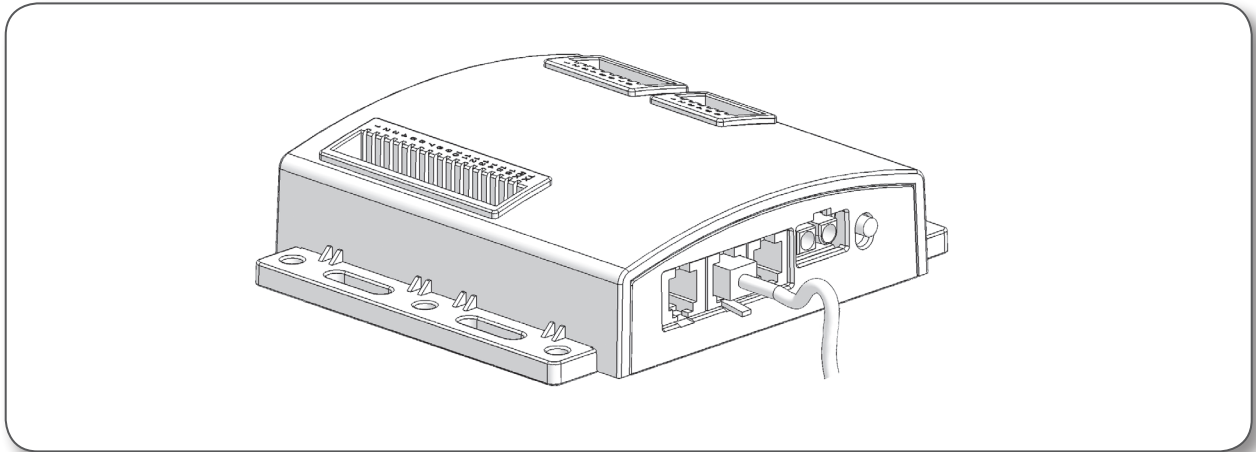
(+ -) Battery Port

This is a standard connector for 7.2V power sources, such as the Battery Holder or the VEX 7.2V Robot Battery. It is "keyed" with an asymmetric plug shape to prevent accidental insertion in the wrong direction. The Microcontroller draws power for both its own operation and for all other attached devices through this port.

Logic Subsystem, continued

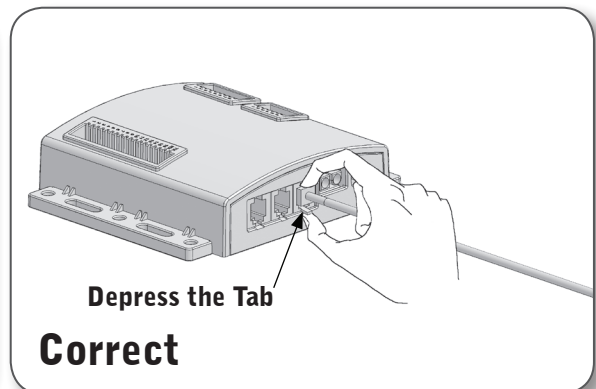
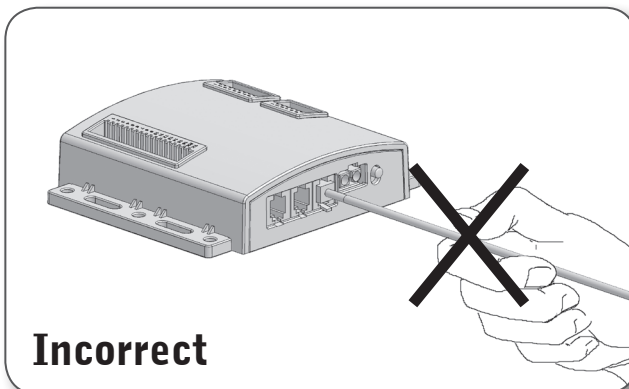
Wires and Connectors

Be careful with wires. Not only are they crucial for your robot to work, but they are fragile and often messy to manage, and worse, they have a tendency to become exposed to environmental hazards (like other robots that can snag on them). Here are a few tips:



Tie your wires. The VEX Robotics Design System comes with lots of wire ties, so there is no excuse for not tying your wires down to keep them safe! Tie wires to parts of your robot that are far away from moving parts if possible, and try to keep them out of the way of areas where you will need to do maintenance work. Be extremely careful when removing wire ties. It is easy to nick or cut wires running through the ties while trying to cut the tie.

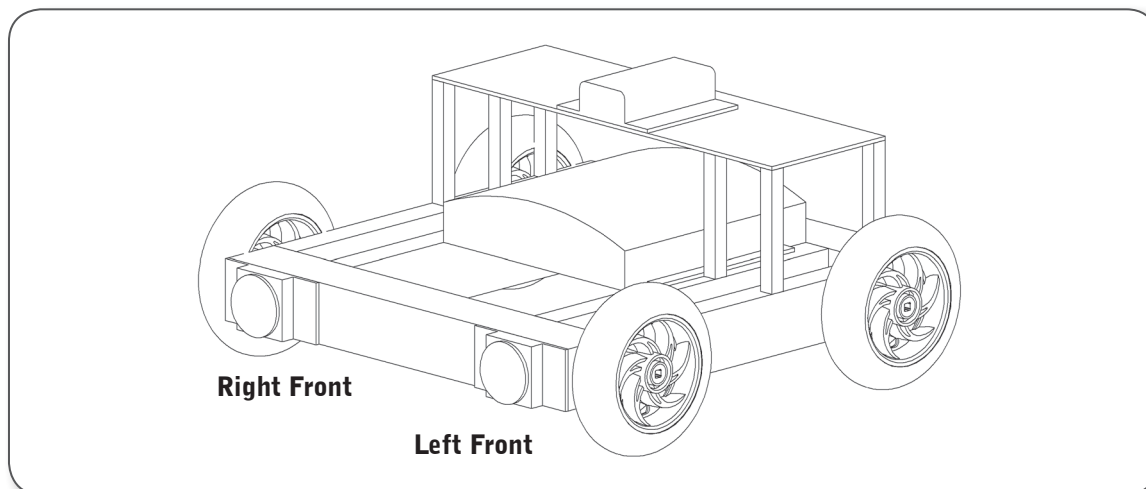
Don't pull wires. When removing a (serial or RX) wire that is already plugged in, never pull on the wire itself. Always grip the connector head depressing the tab and pull. Never apply force to the wire, or it may become detached from the head.



Don't force connectors. All the connector ports in the VEX Robotics Design System are "keyed", meaning that the plastic connectors are constructed so that they only fit in one direction. Do not force connectors in backwards! Not only will this physically damage the port and head, but you also risk electrical damage to the Microcontroller and the component you were trying to plug in.

Logic Subsystem, continued

Autonomous Mode



Port	What to plug in	Function
Analog/Digital 13	Jumper	Activates autonomous mode
Analog/Digital 11	Left-front Bumper Switch Sensor	Left side collision detector
Analog/Digital 12	Right-front Bumper Switch Sensor	Right side collision detector
Motor Port 3	Left Motor	Move robot according to commands issued by autonomous mode program
Motor Port 2	Right Motor	

Autonomous mode allows the robot to operate one very simple pre-programmed behavior without any human input. When enabled, the robot will wander around, using a pair of bumper or limit switch sensors to detect obstacles.

- You must enable Autonomous Mode in order to use the autonomous behavior. Place a Jumper Clip on Digital Input 13 to enable Autonomous Mode operation.
- Plug the Bumper Switch or Limit Switch Sensors into ports 11 and 12 to use this behavior. The “left-front” sensor should go into port 11, and the “right-front” sensor should go into port 12.
- The robot will begin running autonomously 2-3 seconds after being turned on. The delay is intended to give the person who turns it on some time to get out of its way.
- The robot will drive forward without human control when in Autonomous Mode. When the bumper switch on input port 11 is pushed in, the robot will interpret this as an indication that it has run into an immovable obstacle on the left side, and will turn to the right until the switch is no longer pushed in. The bumper switch on input port 12 does the same thing, but monitors the right side and makes the robot turn left when it is pushed.
- Human control is disabled! Be sure to watch the robot to ensure that it does not venture into any dangerous areas. In particular, the robot has no way of detecting “negative obstacles”, like ditches or cliffs. Pick up the robot and stop it if it is in danger of damaging itself!

Logic Subsystem, continued

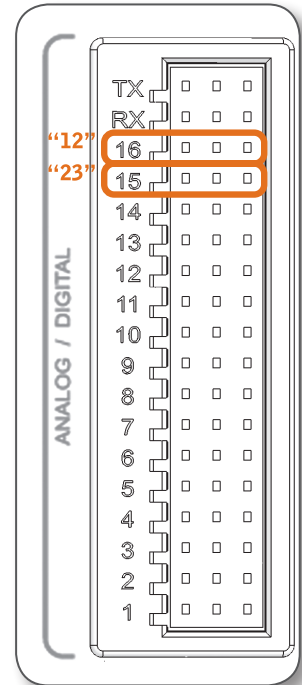
Advanced Operator Control Configurations

The descriptions below are summaries. You will find a complete listing of different control layouts in Appendix D.

4WD (Four Wheel Drive) Jumpers 15 and 16

Jumpers 15 and 16 are used to set the robot into four-wheel drive mode, which uses four motors to run a four-wheeled robot (one motor on each wheel). The two left wheels will always turn together, as will the two right wheels.

The rear-left motor should go on Motor Port 8, and the rear-right motor should go on Motor Port 7. If the Transmitter is in "23 mode", you should then attach a jumper to Digital/Analog input port 15. If the Transmitter is in "12 mode", put the jumper on port 16 instead. If you wish to use both 4WD and "Software 12 mix" mode (see below), you do not need either jumper 15 or 16, because the "Software 12 mix" automatically enables 4WD.

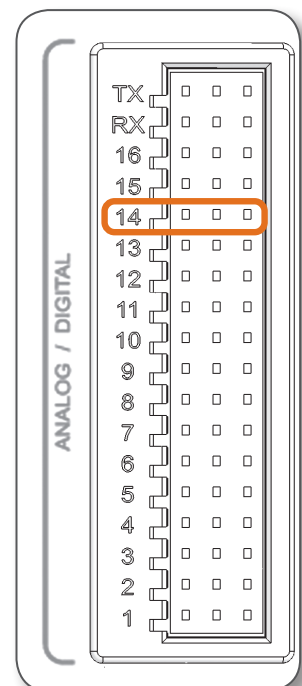


"Software 12 mix" Jumper 14

Placing a jumper on this port will activate programming on the Microcontroller that will allow it to behave as if you had activated the "12 mode" on the Transmitter (see Basic Controller Operation in the Sensor Subsystem chapter for more information on "12 mode" on the Transmitter). You must set the Transmitter itself to "23 mode" in order to use this feature otherwise your controls will not behave correctly (your input will be "mixed" twice, producing the wrong motor commands).

This version of the "12 mode" is slightly different from the Transmitter's "12 mode", however, because it will allow you to go at full speed straight forward, whereas the Transmitter's "12 mode" will only go 60% of full speed forward. The result is that the "Software 12 mix" mode goes faster, but feels like it slows down during turns; Transmitter "12 mix" mode will go slower, but feels more responsive in turns.

4WD is enabled automatically in "Software 12 mix" mode, so ports 7 and 8 will be treated as right-rear and left-rear wheel motors respectively; you can hook up additional motors to use 4WD, but you can't use these ports to independently control other attachments.



Concepts to Understand

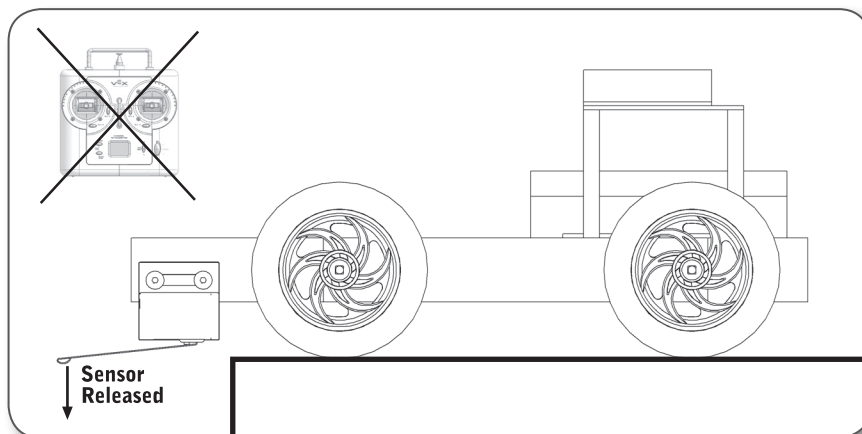
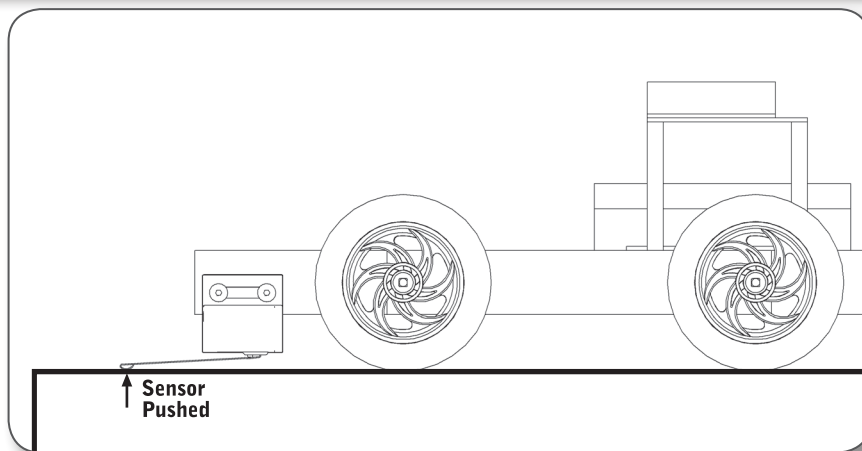
Role of Electronic Control

A robot is a very complex system of parts that must work together in order to achieve a desired goal. Electronic control provided by a programmable controller like the VEX Microcontroller allows the robot to coordinate the operation of the different components and achieve its goals.

Sensing, Planning, Acting: Cliffbot

Robots characteristically have two major capabilities that other mechanical systems don't have: they can sense important things about the world around them through the use of sensors, and they can process sensor information according to a pre-programmed routine. They then act on that plan, by controlling motors and other output devices.

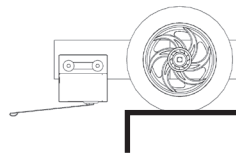
For instance, consider the robot to the right, which has been programmed to respond to radio control commands from a human operator so it can be driven just like a normal radio controlled car, except that the robot is also equipped with a "cliff detector" sensor attachment. The robot will drive around as directed, unless the sensor detects that the robot is about to run off a cliff. Upon detecting a cliff hazard, the robot would temporarily disable human control, give a visible signal to warn the driver of the danger, and then return control to the human operator, who can then maneuver safely away from the cliff (or keep going, if the robot was mistaken).



Concepts to Understand, continued

Sensing, Planning, Acting: Cliffbot, continued

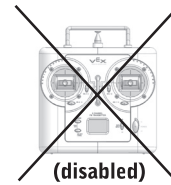
In order to accomplish this, the robot needs to be able to sense the presence of a cliff hazard through the “cliff detector” sensor. If the robot doesn’t detect a cliff, then it obeys the human controller’s input, but if the robot detects a cliff, it will instead automatically stop and alert the driver.



Sense

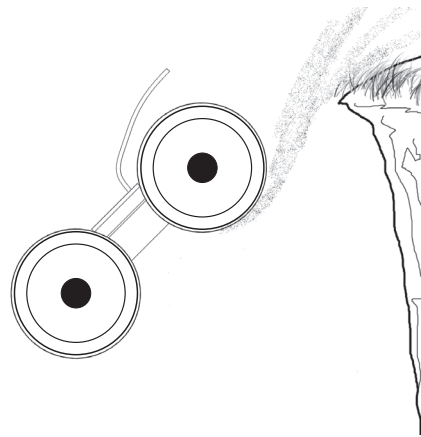


Process



Act

In contrast, a simple radio controlled car would not have this capability. Without the ability to sense, or to react based on sensor feedback, the car would have no way to do anything but blindly follow the commands of a human controller who might not be able to see very well from where he or she is standing...



Subsystem Interactions

How does the Logic Subsystem interact with...

...the Structure Subsystem?

- The Structure Subsystem holds the Microcontroller in place. Since the Microcontroller is a very delicate and important part of the robot, the Structure Subsystem will also need to provide physical protection for the Microcontroller by keeping it in a secure spot.
- The Structure Subsystem does not plug into the Microcontroller in any way; however, the Structure Subsystem does need to provide accommodation and protection for the wires that run between the Microcontroller and other pieces.

...the Motion Subsystem?

- The Logic Subsystem provides power and issues output commands to the Motion Subsystem components. Though these commands are usually based on user input (relayed through the Control Subsystem), the final decision on what command is issued, as well as the actual flow of electricity (from the Power Subsystem) is all controlled by the Logic Subsystem.
- Motors from the Motion Subsystem plug into the Motors port bank on the top of the Microcontroller.

...the Sensor Subsystem?

- The Logic Subsystem relies on feedback from the Sensor Subsystem to provide information about the robot's environment. It uses this data to make informed decisions about how the robot should behave.
- The actual behavior that is activated when a sensor is triggered depends on which port the sensor is plugged into on the Microcontroller and the Code being executed.

...the Control Subsystem?

- The Microcontroller module controls the flow of commands from the human operator to the robot. The Microcontroller ultimately decides whether to pass joystick commands on to the motors, to modify them, or to override them, based on its programmed behavior and other information available to it (from sensors, for instance).
- The RF Receiver Module is connected to the VEX Microcontroller through the yellow control cable that runs from the back of the Receiver into the rear of the Microcontroller. Up to two Receivers can be supported simultaneously.
- If desired, the Transmitter can be hooked directly into the VEX Microcontroller using a tether cable, bypassing the RF Receiver Module (see the Control Subsystem chapter for details).
- The specific way that the robot will respond to joystick movement is determined by a combination of Transmitter and Microcontroller settings. See Appendix on Control for a full listing of control layouts.

... the Power Subsystem?

- Since the Microcontroller is a device that contains a large number of electronic components, including two computer processors, it needs power, which is provided by the Power Subsystem.
- The Microcontroller redistributes electrical energy from the Power Subsystem battery to the other subsystems' components (like motors).
- The Power Subsystem battery holder (or battery pack) connects to the Microcontroller module using a two-pin connector inside a plastic housing, which is mechanically keyed to prevent it from being plugged in backward. The port is on the rear of the Microcontroller next to the power switch.