Programming Bible with TurtleShell

Sweeney Salzmann

2016

Contents

Chapter 1

Introduction

Hello.

1.1 Introduction

Hello!

Chapter 2

Physical Hardware

This section covers the physical devices associated with the robot that must be understood in order to properly program the robot. A basic familiarity with all parts of the robot is required, but certain parts, namely those associated with the control system must be understood in greater depth. This section will dedicate time to each class of components in rough proportion to their importance to programming.

2.1 A Note On The Separation of Programming and Electrical

The boundary between programming and electrical, especially with sensors, can be unclear. The generally accepted split is that electrical is in charge (no pun intended) of the hardware and wiring, while programming is responsible for the control logic. However, programming is often given control over some aspects of hardware for the placement of sensors, although it is still electrical's responsibility to wire the sensors.

2.2 Motor Controllers

Motor controllers are what regulate the motors. They have a connection to the Power Distribution Panel (PDP), the motors themselves, and the roboRio through a PWM (or Pulse Width Modulation) control. They take the PWM signal and use it to control the voltage going to the motor, and thus control the motor.

2.2.1 Victor 88x

Victor 884 and 888s are the simplest type of motor controller, and overall they are effective. The differences between the two are minor, with some changes

to the PWM curve. They are the second-largest motor controller, with the required fan taking up much of the space. They have been made obsolete by the introduction of the Victor SP, which is strictly better.

2.2.2 Victor SP

The Victor SP is VEX's new motor controller, and features several improvements over the previous models, including a much smaller form factor and an integrated heatsink which leaves a fan optional. Use-wise, it functions similar to the previous models with a PWM input.

2.2.3 Talon SR

The Talon SR is very similar to the Victor SP, although it has a larger footprint more comparable to the Victor 88x models. It has been deprecated in favour of the Talon SRX, but the Victor SP is a more direct successor in function.

2.2.4 Talon SRX

The Talon SRX is a much more advanced motor controller, comparable to a Jaguar in power. It can connect over CAN or PWM, and features functions that make it easier to set up PID control. However, this does lead to a more expensive motor controller and reduced programmer control over functionality.

2.2.5 Jaguar

Jaguars are the most advanced motor controller, but they suffer heavily for their features, which include connectiblity over CAN, PWM, and Ethernet. Jaguars are far larger than other motor controllers, in footprint, height, and cost. They suffer an even larger lack of programmer control, with them refusing to work if they are at risk of overheating.

2.2.6 Note on Programmer Control

Best practices developed at Team 1458 have been to do the majority of the coding manually. There have been issues with documentation and errors in WPILib, leading to the restriction of its use to the component interactions. Not only does coding it ourselves allow us to avoid these pitfalls, but allows programmers to get additional experience and learn more. Thus, WPILib is only used in TurtleShell for motor & sensor interfaces, no higher-order logic is taken from it.

2.3 Chassis/Drive Train

The chassis/drivetrain refers to the physical structure of the robot. This is the domain of mechanical, and programmers simply need to understand how the

motor movements correspond to actions in the physical world. More information on drive systems can be found in Part 3.

2.4 Electrical Devices

This section covers the Power Distribution Panel (PDP), Pneumatic Control Module (PCM), and the Voltage Regulator Module (VRM). These devices are usually not used directly by programming, but understanding their function is necessary.

2.4.1 Power Distribution Panel (PDP)

The PDP is what provides power to most of the robot. It has a connection to the battery through the circuit breaker, which serves as the on/off switch on the robot. It then connects through circuit breakers to all devices on the robot that require power. It is equipped with CAN in order to monitor the connections and their current draw.

2.4.2 Pneumatic Control Module (PCM)

The PCM is connected over CAN to the roboRIO. All controls for pneumatic devices go through here. It provides power for the compressor and regulates it, along with controlling any solenoids.

2.4.3 Voltage Regulator Module (VRM)

The VRM is connected to the PDP, and provides lower-voltage power. It provides power to the router on board, as well as other devices that dont require high amperages (so not most motors), such as LED lights.

2.5 Sensors

This section discusses a variety of types of different sensors available on the robot, and their possible uses and drawbacks. It doesnt cover their coding or integration into software. This section is divided based on utility, however some sensors may fit under multiple categories.

2.5.1 Distance

Distance sensors determine how far away something is from the sensor. It can be useful in aligning the robot or in preventing collisions.

Infrared

Infrared sensors detect how far away something is from the sensor based on the timing of pulses of infrared light. They are able to detect small objects and work in enclosed spaces, but are more expensive than ultrasonic sensors.

Ultrasonic

Ultrasonic sensors emit sounds above the range of human hearing and count the time it takes the sound to return. They are cheap, however they are bad at detecting small or curved objects, and are subject to interference from objects in the way.

Laser

Laser distance sensors are extremely accurate, quite often on the millimetre level, and do not suffer from as much interference as other sensors. However, they are the most expensive and fragile, qualities often unsuitable for the robot.

2.5.2 Rotation

Gyroscope

Gyroscopes are the best known rotation sensors. They are actually unable to measure the way the robot is facing, instead it measures the change and uses that to determine the direction relative to the start. They are widely available, but they have a problem in accuracy called yaw drift, where the angle will shift by 2-3 degrees per minute, even while stationary for the best gyros, and larger drift for lower quality.

Magnetometer

Magnetometers measure the strength of magnetic fields on certain axes. In the context of the FRC, they are used to measure the Earths magnetic field and thus act as a compass, providing a reference point for rotation. While they dont have the same problem with drift, they are much more difficult to calibrate, with the calibration changing based on where the robot is located and what ambient magnetic fields exist. This can prove a problem, as the robots motors generate magnetic fields which can interfere with proper function. They are also much more difficult to program.

2.5.3 Vision

These sensors work with vision, which can be used for a variety of tasks depending on the year.

Camera

Cameras are really the only device that can be used for vision, the lights are only there to assist the camera. The camera can see the retroreflective targets that are often visible during the autonomous period, and so the robot can take action based on that. Coding the camera is a difficult task, several libraries such as GRIP and NIVision are able to work with it.

Light

The lights are to illuminate the retroreflective target so the camera can identify it. They are usually green, as green stands out the most from other colours present.

2.5.4 Contact

Pushbutton

Pushbutton switches consist of a button that makes (or sometimes breaks) an electrical connection. They can be useful as bumpers to know if the robot has made contact.

Limit Switch

Limit switches have some sort of object that when it is pressed down, it triggers. They are electronically identical to pushbuttons, but they have a switch rather than a button. They are useful in safety for moving devices, to ensure they dont go too far and damage something.

2.5.5 Position

Accelerometer

Accelerometers dont actually measure position, they measure acceleration. It's technically possible to derive (well, integrate) position from acceleration, but it is a difficult process. Accelerometers can still be used in autonomous for positioning, but arent very useful beyond that. In years where the robot is tilting, it can be used to determine whether or not the robot is flat.

Rotary Encoder

Rotary encoders (usually just called encoders) are one of the most useful sensors. They record rotations of a shaft, which can be used to do everything from measuring the distance the robot has moved forward to the rotation of an appendage. They work by causing an electrical pulse whenever the shaft moves a certain amount (Such as 1°). The roboRio counts these pulses and uses that to determine how far the shaft has rotated. In order so that to roboRio can determine which direction the shaft is rotating, there are two channels that are

slightly offset, so the roboRio can tell the direction by which triggers first. This means that it requires two digital inputs.

2.6 roboRio

The roboRio is the brain of the robot. The code runs on the roboRio, which runs a modified version of Linux. The roboRio is one of the few "smart" devices on the robot, with most of the remaining devices serving to interface between it and the physical world. It has digital, analog, I^2C, SPI, USB, CAN and Ethernet I/O (Input/Output), along with PWM output and the MXP. (Dont worry, all of those acronyms will be explained shortly) By the FRC rules, all control of motors has be done through the roboRio.

2.6.1 MXP (MyRio eXpansion Port)

The MXP is the extra set of pins in the middle of the roboRio. It is meant for custom devices, although a few commercial ones are available. It has a massive amount of possible inputs, about doubling the possible amounts of each input and output. Its use is largely unnecessary unless a large amount of inputs or outputs are needed.

2.6.2 Digital I/O

The digital input and output ports use the same cables as the PWM inputs. The red and black are to power the sensors (with red power and black ground), while the white carries the signal in a digital fashion. Digital is where there are only two values, on and off. Sensors that use this include limit switches and rotary encoders.

2.6.3 Analog Input

Analog Inputs are similar to the digital inputs, with the same cables and colours for power, ground, and signal. Analog is where the voltage varies, so there is a continuous range of values. Analog inputs power different kinds of sensors, with some gyroscopes and accelerometers, and most infrared and ultrasonic sensors relying on an analog input. The voltage from these must be interpreted to get meaningful results.

2.6.4 PWM

PWM (Pulse Width Modulation) is how the roboRio communicates with the motor controllers. PWM is a system that changes the width of pulses, usually in the range of fractions of milliseconds, to alter motor power or servo position. It uses the same cables as analog and digital inputs. The main use of PWM outputs is to control motors.

2.6.5 I^2C

I^2C(Pronounced Eye-two-see) is another communication protocol. It is very advanced, and is capable of exchanging large amounts of information and connecting multiple devices, but is very complicated (An interviewed company contracted it out rather than work with it). Luckily, some of the very low-level communication is done with the libraries, so the programmer only has to work with the bytes that are being transferred, rather than the process of transferring them.

2.6.6 SPI

SPI is another complicated communications protocol, one which is not commonly used in FRC. Wikipedia states:

> The Serial Peripheral Interface (SPI) bus is a synchronous serial communication interface specification used for short distance communication, primarily in embedded systems. The interface was developed by Motorola and has become a de facto standard. Typical applications include sensors, Secure Digital cards, and liquid crystal displays. SPI devices communicate in full duplex mode using a master-slave architecture with a single master. The master device originates the frame for reading and writing. Multiple slave devices are supported through selection with individual slave select (SS) lines. Sometimes SPI is called a four-wire serial bus, contrasting with three-, two-, and one-wire serial buses. The SPI may be accurately described as a synchronous serial interface, but it is different from the Synchronous Serial Interface (SSI) protocol, which is also a four-wire synchronous serial communication protocol, but employs differential signaling and provides only a single simplex communication channel.

2.6.7 CANbus

The CAN (controller area network) bus is another way of interfacing with devices. It is commonly used in industry, and it is found in modern cars. It involves the high, yellow wire and the low, green wire. They can be used for Jaguars, and other advanced electronics. One key aspect is their use in of pneumatics and the PDP, CAN is the only way to connect to those. All workings with the CAN system are hidden in WPILib, a reasonable approach given its complexity. The PDP and respective pneumatic classes can be used to interact with those systems.

2.6.8 Ethernet

There is an Ethernet port on the roboRIO. Ethernet is a standard for communication, often used as a wired way to obtain internet access. For the robot, the

Ethernet port is used to connect to the router that provides wireless connectivity. Cat 5e or above is recommended, as well remember to make sure there are not significant kinks in the cable, it degrades performance.

2.6.9 USB

There are three USB ports on the roboRIO. Two are USB Type-A, the standard USB port. Devices such as cameras should be connected here, as well as any sensors that work over USB.
The USB Type-B (The cable with the pentagonal other end, often used for printers) port is useful for connecting to the computer, use it instead of Ethernet for direct connections to the roboRIO.

2.7 The Human Body Analogy

For many people, an analogy comparing the parts of the robot to parts of the human body is useful for understanding the way the robot works and how things interact. Henceforth, let the analogy commence:

Skeletal System

The frame and mechanical portions of the robot.

Muscular System

Motors and solenoids.

Circulatory System

Components carrying electricity for the purpose of powering components (not signals).

Heart

The PDP.

Arteries

Red cables, where power flows out.

Veins

Black cables, where power flows back.

Lungs

The battery, the source of all electrical power on the robot.

Nervous System

"Smart" components as well as signal-carrying wires.

Axons

Signal-carrying wires, most colours but red and black, especially white, yellow, and green.

Nerve Endings

Sensors.

Brain

The roboRIO, other computing devices would be secondary brains.

Chapter 3

Programming on the roboRIO

This chapter will cover the roboRIO, languages running on it, the Java language, and some basic Java code on the robot. This chapter will not cover TurtleShell, and some things introduced in 3.5 will be overridden in 4.

3.1 roboRIO Basics

The roboRIO is a proper computer. It is produced by National Instruments, based on the Xilinix© Zynq^{TM}-7020 platform. It has as its processor an ARM Cortex^{TM}-A9, with 256MB of RAM and 512MB of flash storage. It runs a real-time variant of Linux, and supports LabVIEW, C++, and Java. This may be a restatement of the fact sheet.
It is possible to connect to the roboRIO

3.2 roboRIO Intermediates

3.3 Languages

java

3.4 Basic Java

period

3.5 Simple Java on the roboRIO

easy

Chapter 4

TurtleShell

best

4.1 Design Philosophy

Turtwig

4.2 Basic Functionality

sensors, motors, etc

4.3 Advanced Features

pid