

WPILib Documentation Template

Overview

Two parallel paths:

- **Getting Started Guide:** Step-by-step help for beginners to build and deploy quickly.
- **User Manual:** Comprehensive reference material for more advanced or returning users.

Landing Page Call-to-Actions

I'm new here → Getting Started Guide

I want deep technical info → WPILib Manual

New Season Let's Get Started → Quick Start for Returning Teams

I want to learn by doing → Tutorials

*In 2027 there will be actions for both FTC and FRC

Getting Started Guide

Introduction

- What is WPILib? - Existing
- Capabilities and use cases (maybe talk about XRP here) -New Doc
- Software and hardware requirements - new doc pull from installation guide

Installing WPILib- Exists in Installation Guide

- Downloading the WPILib installer
- Installing VS Code and toolchains
- Setting up GradleRIO
- Creating a New Project

Building Your Robot

- Introduction to Hardware Components -Exists
- Introduction to Wiring -Exists
- Sensors Overview - Exists, but needs updating
- ~~Vision Options - New~~
- CAN configuration - New

Creating Your First Robot Program Exists in Create your first robot project

- Creating a test program
- Command-based template overview
- Recommended Folder and file structure

Deploying Code -Exists in Building and Deploying Robot Code

- Connecting to the roboRIO
- Using VS Code to deploy
- Troubleshooting

Basic Teleop and Autonomous -Exists

- Writing your first drive command
- Mapping joystick input
- Simple autonomous routine

Next Steps

- Integrating sensors (gyro, encoders)

- Introduction to command scheduling
- Basic PID tuning
- Basic FeedForward
- Simulation environment overview
- Vision Options - New

WPILib User Manual

Architecture Overview- Exists

- TimedRobot vs CommandRobot
- Command-based framework
- Scheduler and lifecycle

Robot Code Structure- Exists

- Subsystems
- Commands
- Triggers and bindings
- Button-mapping best practices

Subsystems & Commands- Exists

- Creating reusable components
- Grouping commands
- Default commands and interrupting behavior

Hardware Interfaces- Exists

- Motor controllers (PWM, CAN, brushless)
- Sensors (analog, digital, SPI, I2C)
- Pneumatics (PCM, PH)
- Vision cameras and processing

Mechanism Control -Exists

- Motion profiling and trajectories
- Feedforward and PID tuning
- Path planning libraries

Simulation Tools - Exists

- WPILib simulator
- Hardware-in-loop simulation

- Logging simulated data

Debugging & Logging -Exists

- SmartDashboard
- Logging strategies (DataLog, Epilogue, AdvantageKit,)
- NetworkTables viewer

Integration -Exists

- NetworkTables API usage
- Incorporating third-party libraries (REV, CTRE, PhotonVision)
- Link to Vendor Documentation and Libraries

Tutorials -Exists, may need updates for 2027

- CAN bus wiring and device configuration
- Autonomous path planning with PathPlanner
- Creating and testing custom commands
- Subsystem isolation and testing strategies
- Unit testing in WPILib (JUnit, mocks)
- Vision (Aiming/ Targeting/Full Field localization)

Tools

- Driver Station basics - May need updating for 2027
- Quick Start for Returning Teams - Existing
- PathPlanner UI overview
- SysId system identification
- AdvantageScope for data analysis

Resources -All Exist

- FAQs
- Common troubleshooting scenarios
- Migration guide between WPILib versions
- Glossary of WPILib and FRC terms

- External links (Vendor docs, vendor APIs, GitHub examples)
- Contribute to WPILib