

# Playing with Fusion

SEN-36005 Time of Flight Sensor – FRC User’s Manual



## Introduction

The SEN-36005 is a highly integrated time-of-flight ranging/distance sensor with a 1M bit CAN

CAN communication is abstracted through the PlayingWithFusionDriver library. This library provides FRC compatible C++, Java, and LabView APIs, as well as a C API which is intended for teams using unofficially supported languages such as Python.

## Glossary

CAN	Controller Area Network
PWF	Playing With Fusion
TOF	Time of flight

## Electrical connections

The TOF Sensor has four electrical connections. Power and ground (Red and Black) should be connected to the battery through the FRC power distribution board and protected by a 10 Amp fuse.

Warning! Do not reverse the polarity of the power connections. This will cause permanent damage to the sensor

The green and yellow wires are used for CAN communication. Connect the green wire to CAN low on the roboRIO and yellow to CAN high. In general, CAN devices should be connected in a ‘daisy chain’. The CAN bus should route directly between two devices, then between the next two, and so on. It is acceptable to ‘T’ into a CAN bus as long as the length of the ‘T’ (also known as the drop) is less than 11 inches. This length is based on the 1Mbit bitrate used by FRC.

Example of a CAN Bus using short drops to tie left and right motor pairs onto the main CAN trunk:

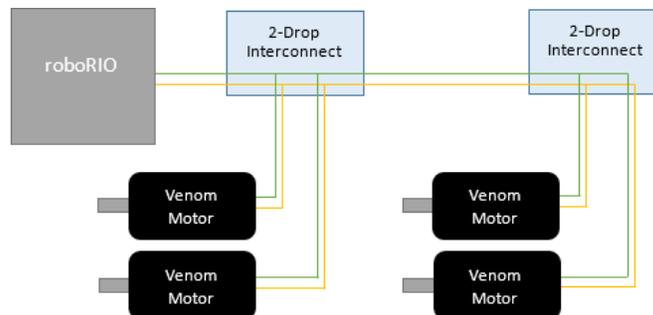


Figure 1 - CAN bus with multiple short drops

The bus length between the roboRIO and the first 2-drop interconnect and the length between the two 2-drop interconnects can be relatively long. If there are additional CAN devices on the bus they should

be connected to the right of the second 2-drop interconnect. The length of that bus can also be relatively long. These sections of the bus follow the recommended daisy chain design.

The length of the drops between the 2-Drop interconnects and the Venom motors must be less than 11 inches to avoid CAN communication issues.

## Importing the FRC compatible software libraries

### Online C++ and Java driver installation

The Venom control library (PlayingWithFusionDriver) can be imported into a robot project directly from the Visual Studio Code application. The following steps will download the latest version of the driver from the Playing With Fusion website.

- Click on the WPILib Command Pallet icon on the top right corner of the VS Code window.
- Select WPILib: Manage Vendor Libraries
- Then select Install new library (online) and enter the following URL:
  - <https://www.playingwithfusion.com/frc/playingwithfusion2020.json>
  - Or for the 2019 rio image:  
<https://www.playingwithfusion.com/frc/playingwithfusion2019.json>
- Press enter

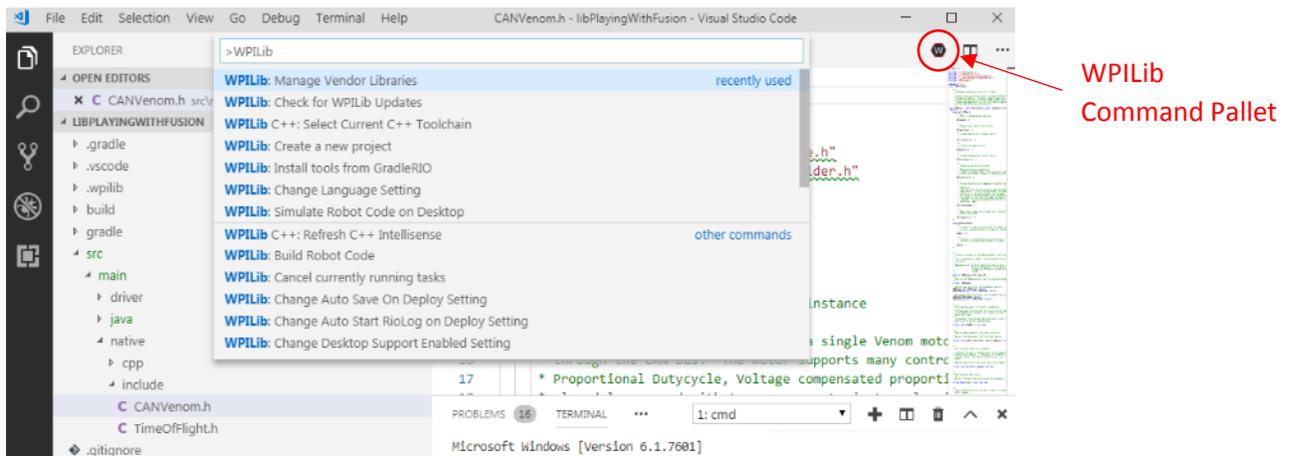


Figure 2 - WPILib Command Pallet in VS Code

If everything works, a folder named vendordeps will be created in the root directory of the project and will contain a file named playingwithfusion2020.json.

### *Offline C++ and Java driver installation*

The Venom library may also be installed without an internet connection. First, download `PlayingWithFusionLibrary2020.zip` from the Venom product page on the Playing With Fusion website. <https://www.playingwithfusion.com/productview.php?pdid=99&catid=1014>

Once the driver files are available, preform the following steps:

- Unzip the `PlayingWithFusionDriver` zip file to `C:\Users\Public\wpilib\2020\` under Windows or `~/wpilib/2020/` under Mac/Linux.
  - (Unzip to `C:\Users\Public\fr2019\` of `~/fr2019/` if using the 2019 toolchain)
  - If asked to merge, select yes.
  - This zip file includes the driver libraries (so files), C++ header files, Java source as well as the JSON file which is used by the FRC build system to identify the Playing With Fusion driver.
- Create a robot project using Visual Studio Code (if one does not already exist)
- Click on the WPILib Command Pallet icon on the top right corner of the VS Code window.
- Select WPILib: Manage Vendor Libraries
- Then select Install new library (offline)
- Select `PlayingWithFusion`

If everything works a folder named `vendordeps` will be created in the root directory of the project and will contain a file named `playingwithfusion.json`.

### *LabVIEW Driver Installation*

The Venom LabVIEW library may be installed without an internet connection. First, download `libPlayingWithFusionDriver.so` and `playing_with_fusion_lib_for_frc-*.vip` from the Venom product page on the Playing With Fusion website.

<https://www.playingwithfusion.com/productview.php?pdid=99&catid=1014>

Once the driver files are available, preform the following steps:

- Copy `libPlayingWithFusionDriver.so` into the `/usr/local/fr2019/third-party/lib` folder on the roboRIO
- Double click on the `*.vip` file. This will open the LabVIEW Package Manager
- Click the Install button

## Driver usage

### *Ranging Mode*

The PWF TOF sensor supports three ranging modes. Short mode (default) works the best in bright lighting conditions and allows the fastest sample period (24ms), but can only measure 1.3 meters. Long mode can measure up to 4 meters in the dark, but may only be able to measure shorter distances depending on the lighting conditions and is limited to slower sample periods. See the VI53L1x datasheet for more information

The ranging mode and sensor time may be specified using the `SetRangingMode()` API function.

### Measured Distance

The TOF sensor returns distance between the back (mounting surface) of the sensor enclosure and the closest object in millimeters. Ranging error may be estimated by using the `GetRangeSigma()` API function. This function returns the standard deviation of the measured distance. If the standard deviation is excessive, evaluate other ranging modes or reduce the sample rate using the `SetRangingMode()` function.

The measured distance in millimeters may be read using the `GetRange()` function.

### Measured Ambient Light Intensity

The TOF sensor may also be used to evaluate the ambient light intensity through the `GetAmbientLightLevel()` API. The output of this function increases in proportion to measured brightness.

## Sensor Configuration

The `PlayingWithFusionDriver` library includes a web interface to manage Playing with Fusion devices

Each device must be assigned an ID that is unique for all other devices of the same type. For example, only one motor controller may use ID 1. It is perfectly acceptable to also configure a time of flight sensor as ID 1 because it is a different device type than the Venom motor.

The configuration interfaces may be accessed by typing in the IP address of the roboRIO into a web browser followed by `:5812`.

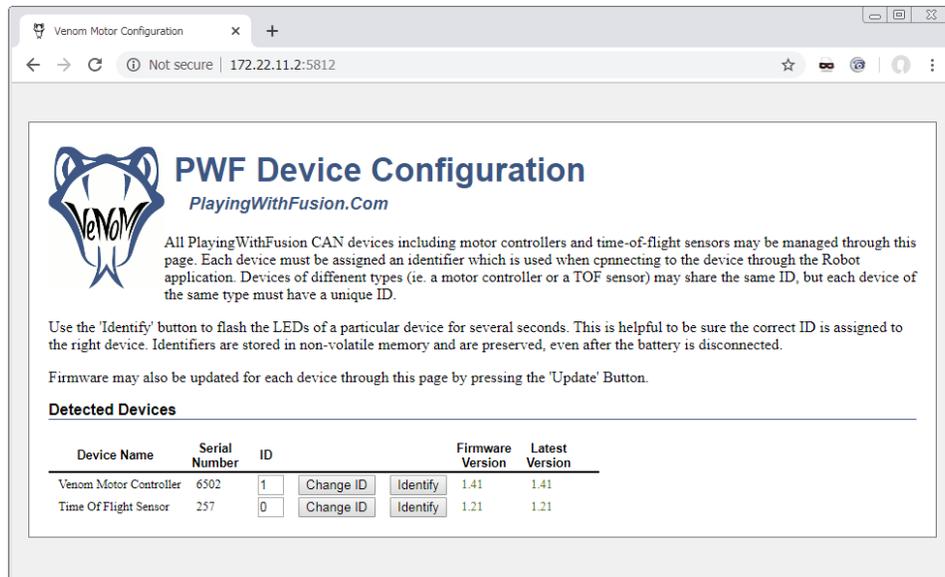


Figure 3 - PWF Device Configuration Page

Each detected Playing with Fusion device will be listed on the configuration page. If a device is missing, check power to the device and verify the CAN bus connections.

Device IDs may be changed by entering the desired device ID into text box on the corresponding row then pressing the Change ID button. The change will take effect immediately and will be stored across power cycles.

The Identify button is a useful tool to determine which physical device is associated with each ID. The amber LED on a TOF sensor blink for a short period when Identify is pressed.

If new firmware is available for any device, an Update button will appear in the right most column for that device. Firmware is included as part of the PlayingWithFusionDriver and cannot be downloaded separately from the Playing with Fusion website.

To update device firmware:

- Disable the robot using the driver station
- Ensure the robot battery is fully charged
- Press the Update button.
- Do not initiate another firmware update or remove power from the roboRIO until the update process is complete.
- A message will appear above the device list which will display whether or not the reprogramming attempt was successful.
- If you have difficulties reprogramming contact Playing With Fusion technical support at [TechnicalSupport@PlayingWithFusion.com](mailto:TechnicalSupport@PlayingWithFusion.com)