

# CANifier User's Guide

Revision 1.0



Cross The Road Electronics

[www.ctr-electronics.com](http://www.ctr-electronics.com)

## Table of Contents

1. What is CANifier? .....	4
1.1. Features .....	5
1.2. Pin Descriptions .....	6
1.3. Electrical Specifications.....	7
1.3.1. Voltage Rail Dropout Plots. ....	7
1.4. Electrical Schematic .....	8
1.5. LEDs.....	9
1.5.1. LED Channel LEDs (3).....	9
1.5.2. Status LED States (2).....	9
2. Installation .....	11
2.1. Installing CAN Bus and Power leads .....	11
2.2. Wiring Common-Anode LED Strip .....	12
2.3. Wiring remaining signals. ....	12
2.3.1. Sensor examples .....	13
2.4. Test CANifier assembly and wiring.....	15
2.5. Enclose/Insulate CANifier assembly.....	16
3. Phoenix Framework (Software) .....	17
3.1. Software Examples.....	17
3.2. Status Frame Rates .....	17
3.3. Custom Platforms.....	17
3.4. HERO LifeBoat / roboRIO Web-based configuration.....	18
4. Functional Limitations.....	19
4.1. Future features to be implemented.....	19
4.2. When hot-swapping certain sensors, CANifier may reset.....	19
4.3. roboRIO Self-test will be added in future.....	19
4.4. Reverse Limit signal is missing ESD diode. ....	19
4.5. Silkscreen suggests max input voltage is 33V. Vdd-max is 28V.....	19
5. Frequently Asked Questions (FAQ) .....	20
5.1. Certain features are not working, but this document indicates they should work.....	20
5.2. Signals updates seem slow. There is stair-stepping in my signal plots.....	20
6. Firmware (CRF) Version History.....	21
7. Revision History .....	21

#### TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your CTRE products. To this end, we will continue to improve our publications, examples, and support to better suit your needs.

If you have any questions or comments regarding this document, or any CTRE product, please contact [support@crosstheroadelectronics.com](mailto:support@crosstheroadelectronics.com)

To obtain the most recent version of this document, please visit [www.ctr-electronics.com](http://www.ctr-electronics.com).

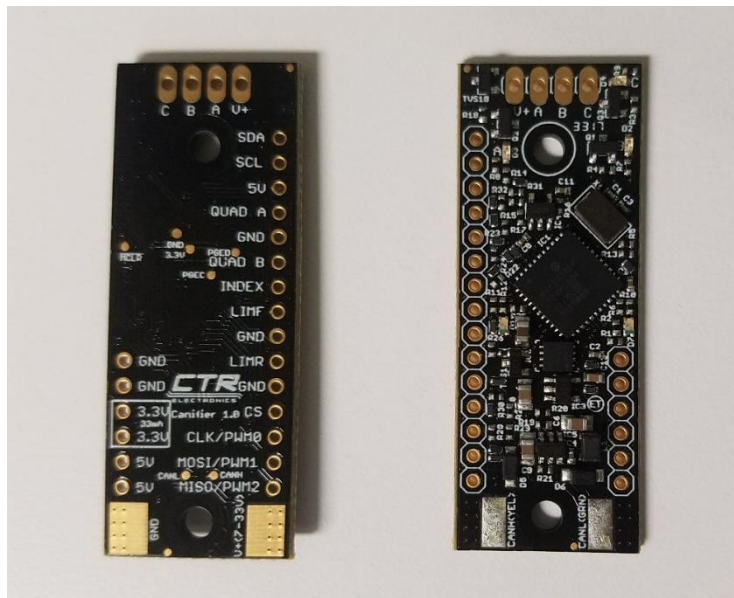
# 1. What is CANifier?

The CTR Electronics CANifier is a CAN-controlled multipurpose LED and General Purpose Input/Output (GPIO) controller. CANifier also supports a variety of sensors including Quadrature, Limit Switch, SPI, I2C, and PWM input/output.

CANifier is designed to “CAN-ify” robot control system components that traditionally do not take advantage of CAN bus. This allows robot developers already using CAN bus to further expand their capabilities.

Example **users** are...

- Commercial Robots developed using the **CTRE HERO Control System**.
- Robots developed for **FRC Competition** (or similar).
- **Academic** robot courses/**design projects**.
- **Rapid Prototyping** (need to quickly add another sensor)



Example **use-cases** solved by CANifier...

- Integrates **PWM** sensors like **LIDAR/Sonar** into a CTRE-HERO/FRC/CAN Bus control system.
- Add more sensors into the FRC CAN bus **without involving the roboRIO** (no hardware SPI/I2C drivers to struggle with). Native support coming in the CTRE Phoenix Framework.
- Allows **direct communication** between external **sensors** and various **CTRE CAN products** for future out-of-box features that “just work”.
- Allows **control of PWM motor controllers** <sup>(1)</sup> like Victor SP via CAN bus commands through CANifier.
- **LED Strip** control via CAN bus.
- Integrate **RC Radio control (PWM)** into CTRE-HERO/FRC/CAN Bus control system.
- Sensors located far from central robot controller can instead tap into the CAN bus.

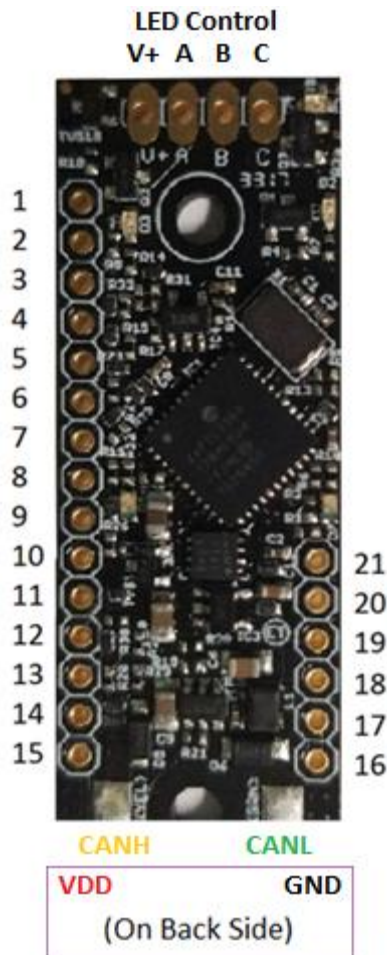
**Note 1:** FRC Teams must confirm what is deemed **legal** when using PWM motor controllers in **competition**. Typically, FRC rules require PWM Motor controllers to be wired directly to roboRIO.

## 1.1. Features

- CAN bus controlled and field-upgradeable
- 3x LED channel drivers
  - Individual RGB control with 10bit resolution.
- Support for many sensor types, including:
  - SRX Mag Encoder / quadrature encoders
  - Limit switches
  - Pixy/CMUcam5
  - Pulse width ultrasonic (e.g. MaxBotix MB1013)
  - LIDAR-Lite v3
  - Talon Tach
  - Digital Sharp distance sensors
- Outputs for common devices, including:
  - PWM motor controllers
  - Small hobby servos
  - PWM LED controllers
  - Any PWM-controlled device
- Electro-static Discharge (ESD) protection on exposed pins
- Future features...
  - Several sensor types supported directly in firmware.
  - SPI/I2C-over-CANBus software driver.
  - Direct communication with CTRE CAN Devices for a variety of features.

## 1.2. Pin Descriptions

The CANifier has a multitude of I/O pins. All pins are counted as GPIO (General Purpose Input/Output) except for power supply pins (5V, GND, 3.3V, VDD).



Pin #	Name	Function
1	SDA	I2C Data pin
2	SCL	I2C Clock pin
3	5V	5V output
4	QUAD A	Quadrature channel A
5	GND	Ground
6	QUAD B	Quadrature channel B
7	INDEX	Quadrature Index
8	LIMF	Limit Switch Forward
9	GND	Ground
10	LIMR	Limit Switch Reverse
11	GND	Ground
12	CS/PWM3	SPI Chip Select / PWM input 3
13	CLK/PWM0	SPI Clock / PWM input 0
14	MOSI/PWM1	SPI Data Out / PWM input 1
15	MISO/PWM2	SPI Data In / PWM input 2
16	5V	5V output
17	5V	5V output
18	3.3V	3.3V output
19	3.3V	3.3V output
20	GND	Ground
21	GND	Ground
	CANH	CAN High
	CANL	CAN Low
	V+	LED Source Voltage (Common anode)
	A	LED Channel A (cathode)
	B	LED Channel B (cathode)
	C	LED Channel C (cathode)
Backside	VDD	Supply Voltage
Backside	GND	Ground

## 1.3. Electrical Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Unit
$T_{amb}$	Ambient temperature		-40		+125	°C
$V_{dd}$	Supply voltage		7		28 <sup>(1)</sup>	V
$I_{supp}$	Supply Current	$7.0V < V_{dd} < 28.0V$	12		24	mA
$V_{drop\_5V}$	5V rail dropout voltage	At $V_{dd}$		6.5 <sup>(2)</sup>		V
$V_{drop\_3.3V}$	3.3V rail dropout voltage	At $V_{dd}$		3.9 <sup>(3)</sup>		V
$V_{drop\_CAN}$	CAN dropout voltage	At $V_{dd}$		5.2		V
$I_{max\_5V}$	Max continuous 5V current draw			500		mA
$I_{max\_3.3V}$	Max continuous 3.3V current draw			100		mA
$I_{max\_ABC}$	Max sink current per LED channel			3.5		A
<b>GPIO Specs</b>						
$V_{OL}$	Low level output voltage				GND+0.4	V
$V_{OH}$	High level output voltage		2.8	3.3		V
$I_O$	Source/Sink current per output pin				3	mA
$R_{GPIO}$	GPIO series resistance <sup>(4)</sup>			22		$\Omega$
$V_{IL}$	Low level input voltage		-0.3		0.2	V
$V_{IH}$	High level input voltage		2.7		5.5	V
$f_{PWM\_ABC}$	A, B, and C PWM switching frequency			1		kHz
$f_{PWM\_OUT\_0123}$	PWM 0-3 Period <sup>(5)</sup>	As output		4.2		ms
$f_{PWM\_IN\_0123}$	Pulse Width Measurement Accuracy	As inputs		250		ns
	Default internal pull mode	As inputs		Pullup		
<b>ESD Rating</b>						
	ESD Protection Contact Discharge <sup>(6)</sup>				$\pm 8$	kV
	ESD Protection Air-Gap Discharge <sup>(6)</sup>				$\pm 8$	kV

**Note 1:** A silkscreen error implies a maximum VDD of 33V. The real maximum VDD is 28V. VDD limits are for guaranteeing full functionality.

**Note 2:** See [Section 1.3.1](#) (Figure 1.1. 5V rail dropout)

**Note 3:** See [Section 1.3.1](#) (Figure 1.2. 3.3V rail dropout)

**Note 4:** See schematic for information on series resistors

**Note 5:** This will be configurable in future updates.

**Note 6:** All pins have TVS diodes for ESD protection **except** for LIMR.

### 1.3.1. Voltage Rail Dropout Plots.

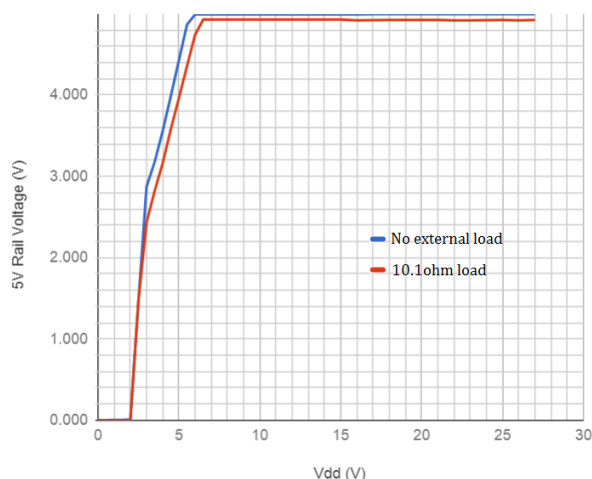


Figure 1.1. **5V** rail dropout voltage

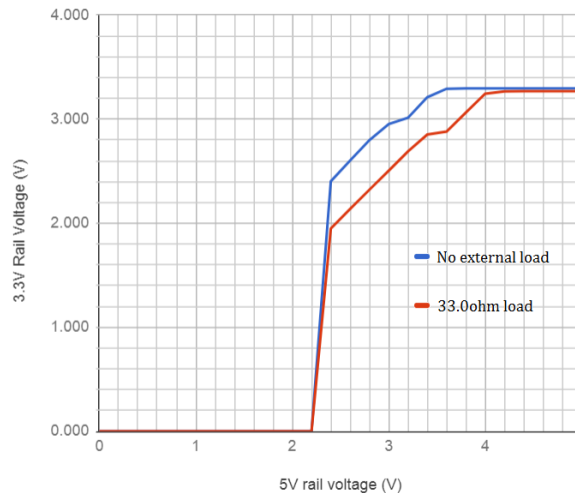
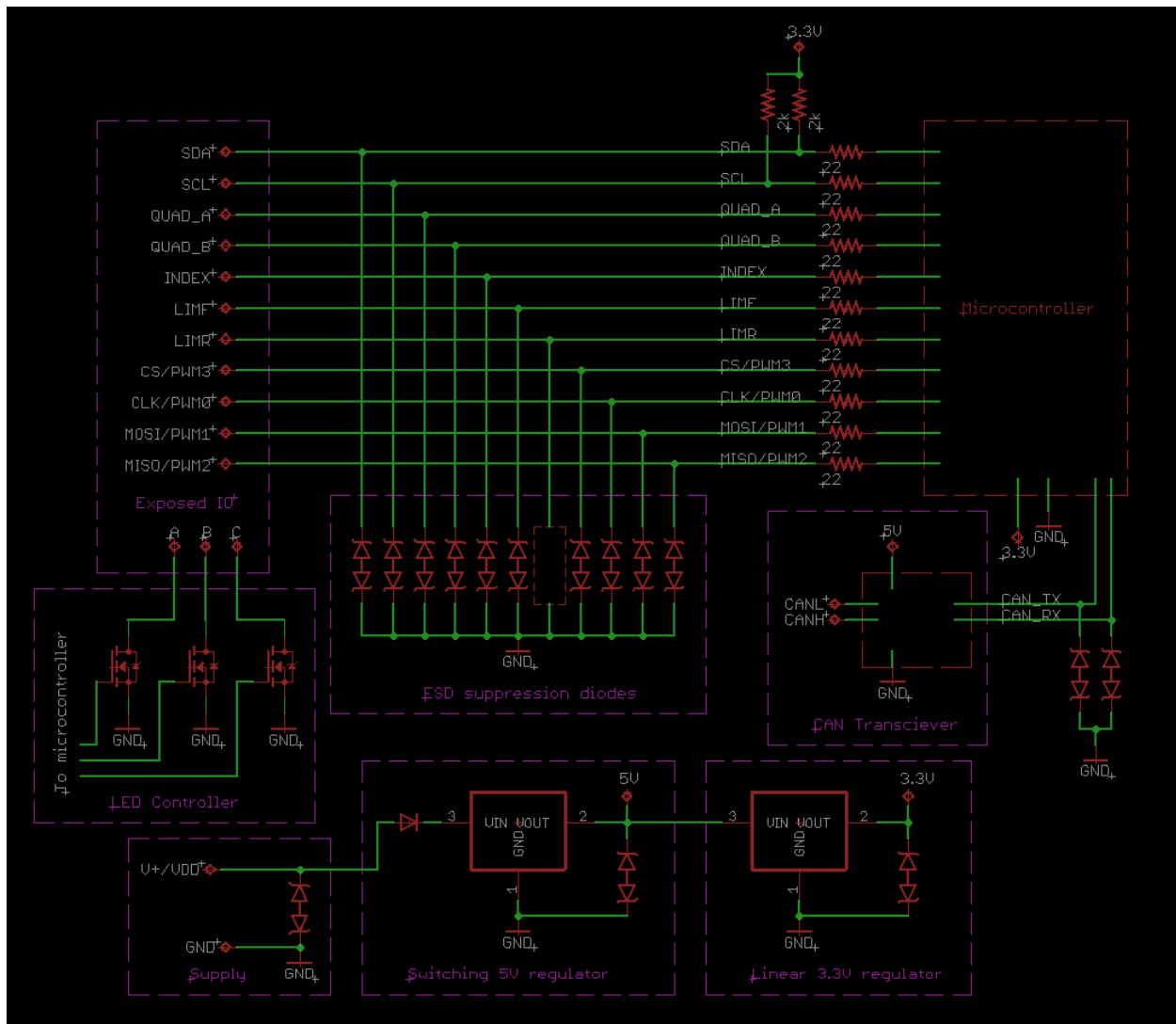


Figure 1.2. **3.3V** rail dropout voltage

## 1.4. Electrical Schematic



Note that V+, 5V, 3.3V, GND, CANH, and CANL have exposed pads on the PCB



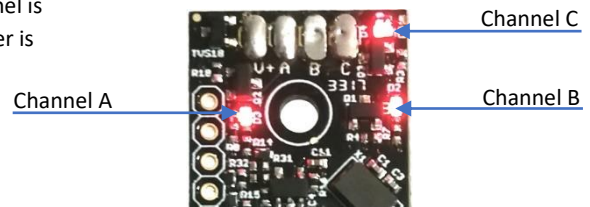
## 1.5. LEDs

CANifier has multiple LEDs...

- Two bicolor LEDs for CAN and control status.
- Three red LEDs to instrument the individual LED channel outputs.

### 1.5.1. LED Channel LEDs (3)

Each LED channel has a dedicated red LED that will illuminate if the channel is modulated (partially or full on). This can be used to confirm if the CANifier is commanded to turn on LED Channels.



### 1.5.2. Status LED States (2)

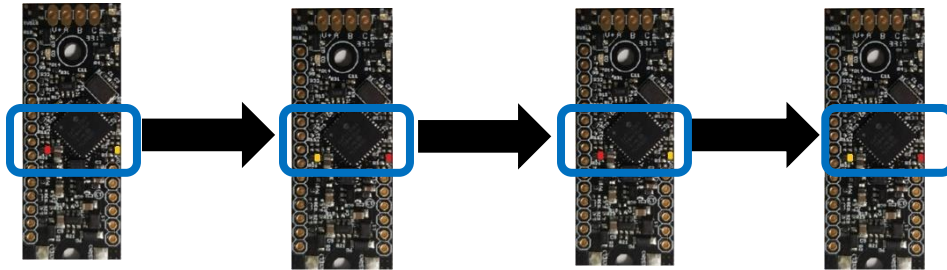
The CANifier features two bicolor LEDs that indicate the status of the CANifier. This feature can be used to confirm proper operation. The table below shows the possible color states and their respective operation.

Blink Codes		
LEDs	Colors	CANifier State
None	None	No Power is being applied to CANifier
LEDs Alternate <sup>(1)</sup>	Off/Red	CAN bus is not detected Check CAN cabling to the CANifier, check other devices on CAN bus to confirm that the bus is operational
LEDs Alternate <sup>(1)</sup>	Off/Orange	CAN bus detected but no commands are being received. Most likely no CANifier object was created in software.
LEDs Alternate <sup>(1)</sup>	Switching between Red/Orange and Orange/Red	Damaged Hardware
LED1 Only "closest" to M+/V+	Green/Orange	In Boot-loader
LEDs Alternate <sup>(1)</sup>	Blinking Green	CAN is connected, the CANifier is being controlled by robot controller.

**Note 1:** LEDs Alternate – Status LEDs are always different colors. The (2X) LEDs will swap colors when blinking.

### 1.5.2.1. Status LEDs – Damaged Hardware

The CANifier can detect certain types of hardware damage. When damage is detected the CANifier will blink in a periodic fashion as mentioned in the table above. This is also illustrated below...



Note that the LEDs **never turn off** when indicating this failure mode. Each LED is always **red** or **orange** at any moment. The **orange/red** LED pattern will continue to blink this pattern with **no intermittent change** in LED activity.

If this is being witnessed, ensure that the LED pattern isn't simply an intermittent **red** LED blink superimposed on to the typical **orange** LED pattern. This is typically observed when there is an intermittent wiring issue or when two or more CANifiers with common device IDs are present on the CAN bus. Both can be checked by connecting the CANifier directly to the robot controller. Do not connect any other CAN devices and avoid complex wire harness strategies.

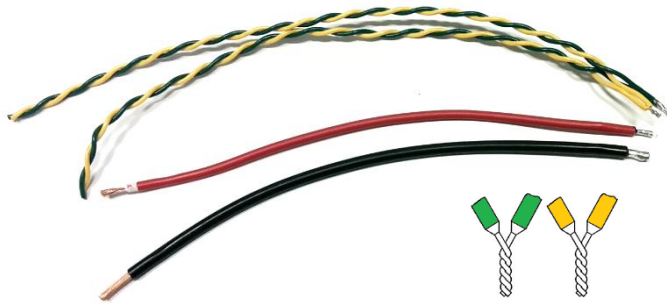
Erratic **red** and **orange** blink patterns are typically caused by intermittent cable issues, or not properly configuring the device IDs of the CANifiers. Such LED patterns should not be misconstrued as the damaged LED code mentioned above.

## 2. Installation

Below is the typical procedure for installing a CANifier.

### 2.1. Installing CAN Bus and Power leads

Prepare wires ahead of time to solder to CANifier.



Generally, every use case will require power and ground wire leads, and CAN bus.

Twist/rat-tail splice two CAN pigtail wire pairs so that it can be easily soldered. This allows for daisy chaining of CAN bus devices.



Pre-tin the stripped CAN bus and power/ground wire ends and respective solder **pads** the CANifier.



Hold pre-tinned solder lead over the pad and apply heat via the soldering iron. Solder joint should appear shiny after cooling with no gaps or discontinuities on the joint surface.



## 2.2. Wiring Common-Anode LED Strip

CANifier is designed to be mated directly or near-proximity to the connection point of a common-anode LED strip. Depending on what is available on the LED strip end, an additional pin header can be solder to facilitate wiring to the CANifier, or the LED strip can be directly soldered to the CANifier pads.

Note the signal orientation (from top to bottom) of LED Strip (Blue, Red, Green, V+) to CANifier (C, B, A, V+). Ensure that the common + rail of the LED Strip connects to the V+ hole/pad on the CANifier.



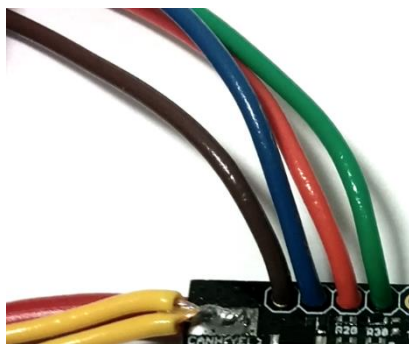
**Warning:** Be sure to select the proper wire gauge for the LED strip. LED strips can draw several Amperes, and selection of a thin gauge may increase the risk of fire. PDP breakers/fuses do not improve this problem since they do not open until 10-40A of continuous draw (depending on which fuse/breaker type). If in doubt, feel the wire harness after several minutes of continuous use to confirm the presence of excessive heat. If excessive heat is detected, consider using a larger gauge.

## 2.3. Wiring remaining signals.

Depending on use-case, additional signals may need to be wired.

The remaining signals that can be wired via the solder holes are...

- Quadrature Encoder.
- I2C.
- SPI.
- PWM (input and output).
- Limit Switches.
- 3.3V/5V/GND for peripherals powered by CANifier.



Strip ~20AWG wire and pass thru the 100mil through-holes. Apply sufficient solder on opposite side and flush-cut the excess stripped wire.

Alternatively, a 100-mil spaced header could be soldered on for convenient socket/breadboard reuse.



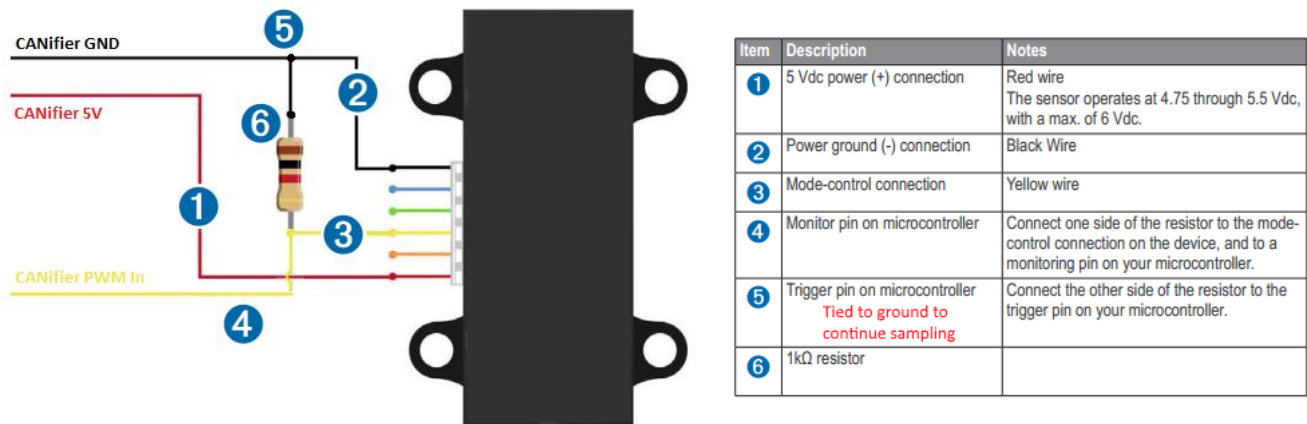
However, in a critical application (mobile robotics for example) a soldered connection is typically **far** more robust.

2.3.1. Sensor examples

Various sensor example configurations are documented below.

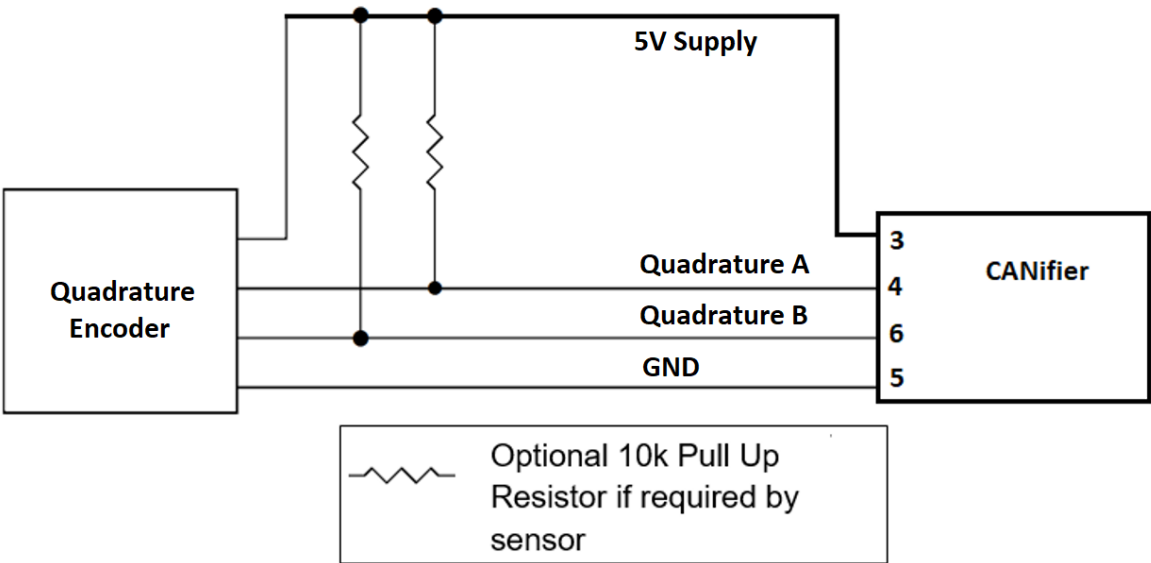
2.3.1.1. LIDAR Lite V3 Distance Sensor

More information can be found in the LIDAR Lite V3 Operation Manual.  
Be sure to tie pull down resistor to ensure LIDAR continuously samples distance.

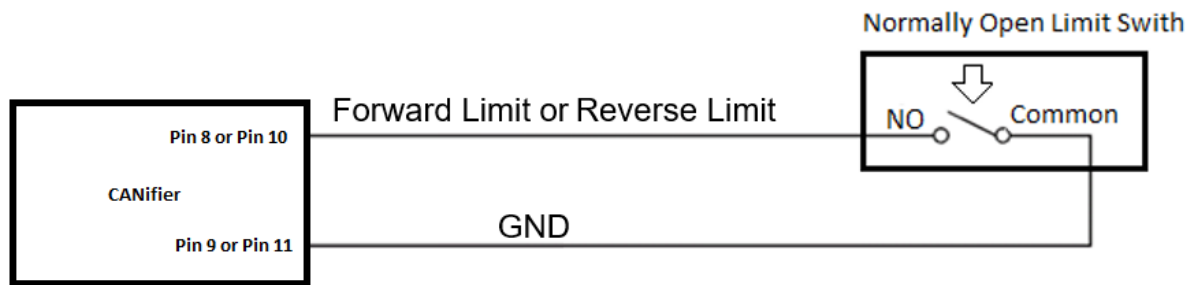
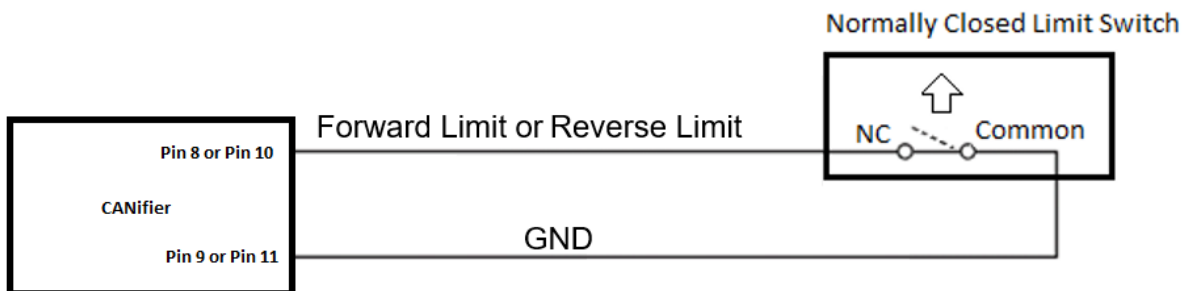


2.3.1.2. Quadrature Encoder

5V Quadrature encoder example below. Quadrature INDEX can be wired as well if application requires it.



## 2.3.1.3. Limit Switch Inputs

**Limit Switch - Normally Open (default)****Limit Switch - Normally Closed**

## 2.4. Test CANifier assembly and wiring

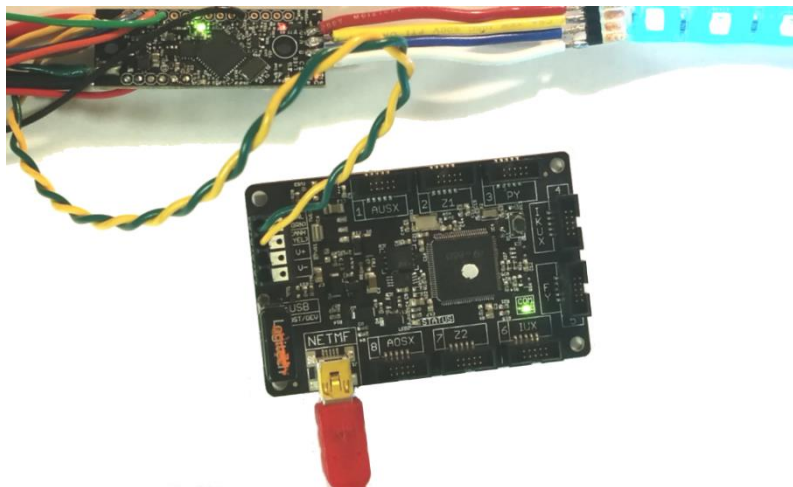
Before installing CANifier into the final enclosure or heat shrink tubing, test CANifier in isolation. This is the best opportunity to field-upgrade CANifier to latest firmware, and test peripherals to ensure wiring is correct before burying the CANifier assembly into the final robot/platform.

**It is recommended to perform basic functional testing on all control system components as soon as they are received. Many of these components have manufacture warranties that can expire if a boxed component is left unused/untested for an excessive period.** Spare/backup components meant for competition should be checked in advance as well. This will ensure the **best chance of success during competition events** where **repair time** is **limited** between matches.

If the 5V or 3.3V rail is used to power a sensor, confirm the sensor is functional and the rails meet the expected voltages with a voltmeter.

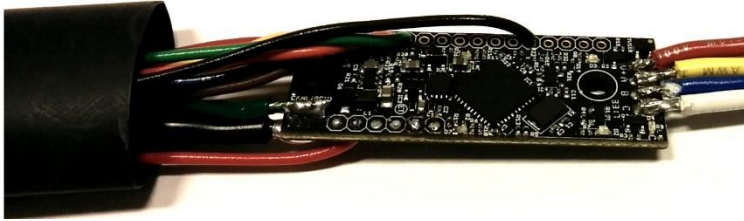
Below, CANifier is directly inserted into a HERO with a simple C# app that streams thru LED strip colors.

Example source is available on [www.ctr-electronics.com](http://www.ctr-electronics.com) and [www.github.com/CrossTheRoadElec](https://www.github.com/CrossTheRoadElec)



## 2.5. Enclose/Insulate CANifier assembly

The recommended approach to insulate CANifier is to shroud entire wire assembly in 1" heat shrink tubing.

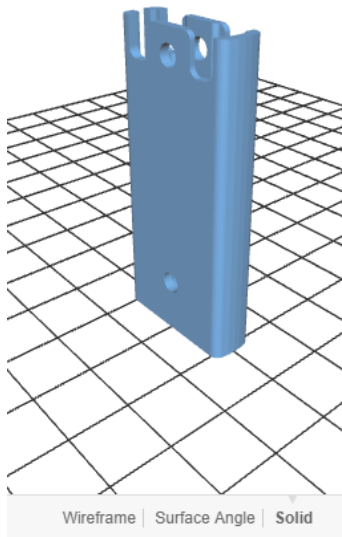


Insert assembly into one side of a precut heat shrink tubing.

Heat shrink should cover all exposed electrical areas of CANifier and LED strip.



After applying heat to heat shrink tubing, neatly wrap and separate the harnesses into their logical sub groups. Illustrated below are two pigtailed for CAN bus, power leads (to PDP/power source), and wire harness for PWMs/sensors/etc.



Alternatively, the CANifier can be mounted to a fixed location using a 3D printed enclosure.

Fixed mounting is accomplished via the two mounting holes in the CANifier. These holes are 0.125" in diameter nominally, and act as clearance for a 4-40 or M3 screw. A case for the CANifier in STEP and STL formats is available on [ctr-electronics.com](http://ctr-electronics.com) to facilitate fixed mounting while covering the exposed pads.

Use any length 4-40 or M3 screw to attach the CANifier to your mounting surface. Ensure that there is no contact between any exposed pads and the mounting surface. Plastic spacers can be used to isolate the CANifier from the surface. The CANifier can also be mounted with double-side tape, but ensure that the tape is nonconductive and that wires are soldered onto the CANifier before mounting.



## 3. Phoenix Framework (Software)

Phoenix Framework is the CTRE Software API for CTRE CAN devices, including CANifier. The current version is in Alpha development and can be found at the CTRE site.

[http://www.ctr-electronics.com/control-system/hro.html#product\\_tabs\\_technical\\_resources](http://www.ctr-electronics.com/control-system/hro.html#product_tabs_technical_resources)

Support will include the following platforms...

- HERO Development board – C#
- Java FRC roboRIO
- C++ FRC roboRIO
- LabVIEW FRC roboRIO

### 3.1. Software Examples

CANifier examples can be found in the following GitHub repositories (or find on the CTRE CANifier product's page).

HERO and FRC programming languages...

<https://github.com/CrossTheRoadElec/Phoenix-Examples-Languages>

FRC LabVIEW...

<https://github.com/CrossTheRoadElec/Phoenix-Examples-LabVIEW>

### 3.2. Status Frame Rates

After CANifier boots application firmware, it transmits the following signals at their respective rates...

Default Signal CAN Bus Frame Rates		
Status Frame	Default Period (ms)	Signals
Status 1	100	LED Duty Cycles, Battery Voltage, Faults
Status 2	10	Quadrature, Digital Inputs
Status 3	100	PWM Channel 0 measurement
Status 4	100	PWM Channel 1 measurement
Status 5	100	PWM Channel 2 measurement
Status 6	100	PWM Channel 3 measurement

These frame rates can be adjusted with the software API in the Phoenix Framework.

Example code snippet below...

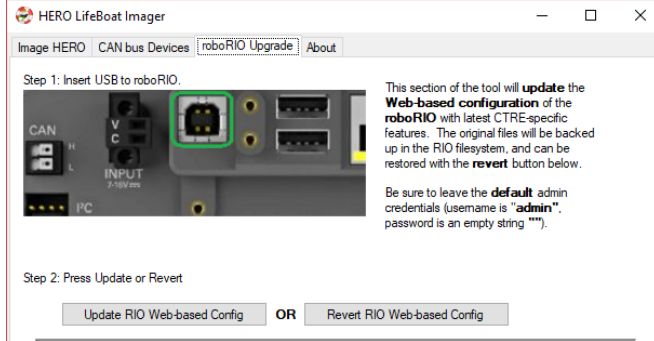
```
public void OnStart()
{
    Hardware.canifier.SetStatusFrameRate(CANifier.StatusFrameRate_1, 10);
}
```



### 3.3. Custom Platforms

Portions of the Phoenix Framework are open source, to allow developers to port the CAN communication to a custom protocol. Check the Phoenix repositories on CTRE's GitHub for more information.

### 3.4. HERO LifeBoat / roboRIO Web-based configuration



For CANifier(s) to appear in the roboRIO Web-based configuration page, run the latest HERO LifeBoat.

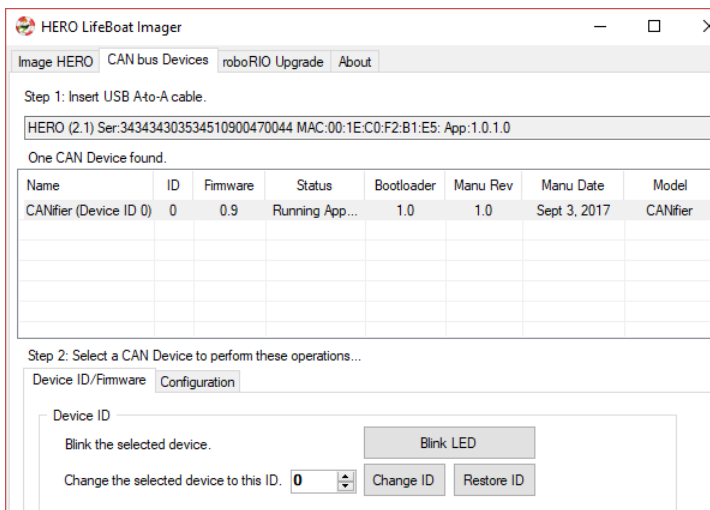
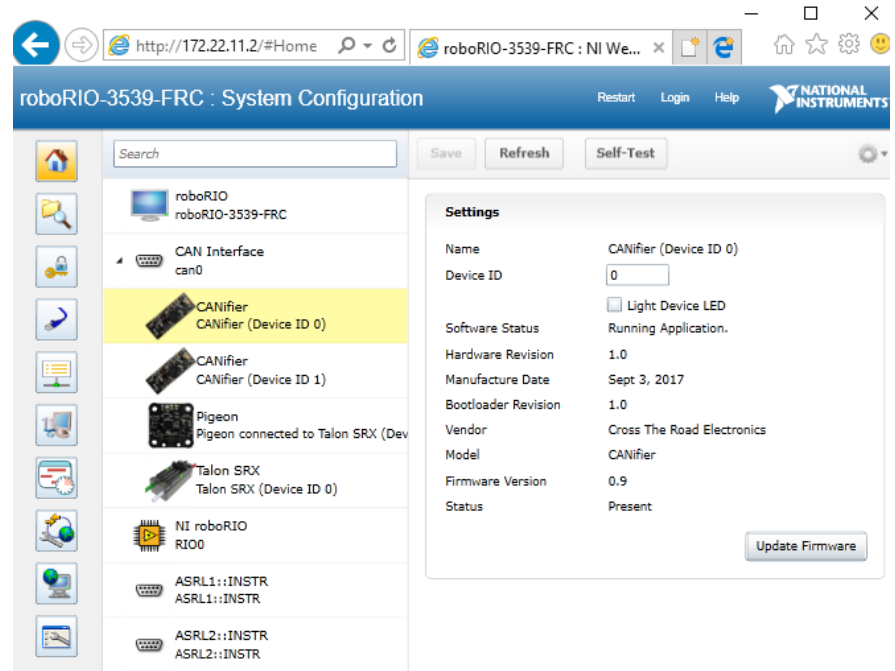
Connect roboRIO with USB cable, and press the “Update RIO Web-based Config” button.

Afterwards, CANifier(s) will appear in the device list of the roboRIO.

“Light Device LED” can be used to confirm which CANifier has which device ID.

No two CANifiers can have the same device ID.

CANifier firmware can be updated via roboRIO in this view as well.



LifeBoat can also be used to update LifeBoat or change Device ID.

## 4. Functional Limitations

### 4.1. Future features to be implemented.

Current version of firmware (CANifier-Application-0.9.crf) supports the following features...

- Pulse width decoding on PWM 0, 1, 2, 3. (Tested with LIDAR v3, various hobby RC Radios, CTRE Mag Sensor, etc.).
- PWM output on PWM 0, 1, 2. (Tested with Talon SRX in PWM mode).
- General purpose digital inputs on all pins.
- General purpose digital outputs on all pins.
- Common anode LED strip control.
- PWM outputs have a fixed period of 4.2ms.

Future versions of firmware will include...

- General I2C/SPI support over CAN bus.
- NeoPixel compliant protocol for addressable LED strips.
- PWM output on PWM 3 (CS)
- Custom control of PWM output period.

... and future CAN bus features that communicate directly with Talon SRXs/Pigeon-IMUs/etc.

### 4.2. When hot-swapping certain sensors, CANifier may reset.

Certain sensors may have an in-rush (on-boot) current draw that exceeds the sensor's typical continuous current draw.

An example of this is the LIDAR-Lite V3 sensor, which has an initial current draw far greater than 500 mA.

Thus, disconnecting/reconnecting the LIDAR Lite sensor while CANifier is powered may cause CANifier to reboot as the supply rail may reset.

Since hot-swapping sensors is typically not done during robotic movement, no workaround is needed beyond ensuring robust connectivity with sensors.

### 4.3. roboRIO Self-test will be added in future.

When Self-Test button is pressed, Web-based configuration will report "not supported in firmware".

This will be added in a future update. The other features in the roboRIO Web-Based Configuration currently function after using HERO LifeBoat to update the roboRIO.

### 4.4. Reverse Limit signal is missing ESD diode.

The current hardware version of CANifier does not have the ESD diode populated for the reverse limit signal.

Though there is protection (series 22Ω resistor), this means LIMR does not meet the full 8KV ESD rating.

This is only a concern during direct ESD strikes occurring on this specific trace.

Regardless, users should take precautions to prevent ESD strikes to electronics under all circumstances.

When approaching robot chassis, be sure to discharge to robot by touching robot chassis before handling electronics.

When handling electronics, take measures to ensure handler is grounded via a wrist strap or grounded work surface.

### 4.5. Silkscreen suggests max input voltage is 33V. Vdd-max is 28V.

Silkscreen is incorrect. Max input voltage is 28V.

## 5. Frequently Asked Questions (FAQ)

### 5.1. Certain features are not working, but this document indicates they should work.

Update to the latest CANifier firmware. CANifier ship firmware is 0.4 which is missing most features.

### 5.2. Signals updates seem slow. There is stair-stepping in my signal plots.

Speed up the CAN frame rates by following [Section 3.2](#).

## 6. Firmware (CRF) Version History

CRF Version	Date	Description
0.9	7-Sept-2017	First public release.
0.4	4-Sept-2017	Ship Firmware

## 7. Revision History

Revision	Date	Description
1.0	8-Oct-2017	First public release.
0.1	22-Sept-2017	Initial Creation.