

---

# **LinuxCanBus Documentation**

*Release 0.1.0*

**Geoffrey Hunter**

**May 20, 2019**



---

Contents:

---

<b>1</b>	<b>A C++ user-space CAN bus driver for Linux (using socketCAN).</b>	<b>1</b>
<b>2</b>	<b>Features</b>	<b>3</b>
2.1	Installation . . . . .	3
2.2	Basic Example . . . . .	4
2.3	Indices and tables . . . . .	4



# CHAPTER 1

---

A C++ user-space CAN bus driver for Linux (using socketCAN).

---

See <http://linuxcanbus.readthedocs.io> for installation instructions, examples and other documentation!



- Easy to use API for controller a CAN device in Linux
- CMake-based build system
- Conan (package manager) support

## 2.1 Installation

### 2.1.1 Automatic Build/Test/Package

Requires Conan to be installed.

```
~/LinuxCanBus$ conan create . testuser/testing
```

This will provide a static library called `libLinuxCanBus.a` and header files under a folder called `include/LinuxCanBus`, available to other Conan packages.

### 2.1.2 Manual Build

```
~/LinuxCanBus$ mkdir build
~/LinuxCanBus$ cd build/
~/LinuxCanBus/build$ conan install ..
~/LinuxCanBus/build$ conan build ..
```

This will build a static library called `libLinuxCanBus.a` and header files under a folder called `include/LinuxCanBus`.

Once you have installed LinuxCanBus, go to the [Basic Example page](#) to see how to use it!

## 2.1.3 Documentation

This documentation was created with `sphinx`. To build the documentation, first make sure `sphinx` is installed, and then navigate to the `docs/` folder and run:

```
make html
```

## 2.2 Basic Example

NOTE: LinuxCanBus does not configure and bring “up” the CAN interface itself. This has to be done with command-line calls to `ip link` and similar before using the library.

```
#include "LinuxCanBus/LinuxCanBus.hpp"

int main() {

    LinuxCanBus canBus;

    // Setup CAN bus using CAN interface can0 and the standard frame format (i.e. not_
↳extended)
    canBus.Init("can0", LinuxCanBus::FrameFormat::STANDARD);

    // Write to CAN bus
    CanMsg writeMsg;
    writeMsg.SetAddress(0x01);
    writeMsg.GetDataMutable().push_back(0x12);
    writeMsg.GetDataMutable().push_back(0x34);
    canBus.Write(writeMsg);

    // Read from CAN bus, if available
    // (0 means no blocking)
    CanMsg readMsg;
    canBus.Read(readMsg, 0);

    return 0;
}
```

## 2.3 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)