CAN Transmitter User Guide

Table of Contents

[2 Overview 2](#_Toc25326412)

[3 Required Files and BOM 2](#_Toc25326413)

[3.1 BOM 2](#_Toc25326414)

[3.2 Enclosure Models 2](#_Toc25326415)

[3.3 Arduino Sketch 2](#_Toc25326416)

[4 Construction 3](#_Toc25326417)

[4.1 Panel Mount Components 3](#_Toc25326418)

[4.2 Electrical Connections 4](#_Toc25326419)

[4.3 Photos 4](#_Toc25326420)

[5 Environment Setup 5](#_Toc25326421)

[5.1 Install Arduino IDE 5](#_Toc25326422)

[5.2 Install mcp\_can Library 5](#_Toc25326423)

[6 Configuration and Programming 6](#_Toc25326424)

[7 Use 7](#_Toc25326425)

[8 Licensing and Disclaimer 7](#_Toc25326426)

[8.1 CAN Transmitter 7](#_Toc25326427)

[8.2 mcp\_can Library 7](#_Toc25326428)

# Overview

The purpose of this CAN transmitter is to send cyclic CAN messages required by DECUs undergoing testing. This CAN transmitter is intended to be easy to create, easy to program, and easy to configure. The user can configure the following parameters:

* Baud rate
* CAN ID type (11/29 bit)
* Number of CAN messages
  + CAN message IDs
  + CAN message DLCs
  + CAN message data
  + CAN message period

This project is and requires software licensed under the *MIT License*, which permits commercial use, modification, and distribution. Additional details are provided under *Licensing and Disclaimer.*

# Required Files and BOM

## BOM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Description** | **Manufacturer Part #** | **Qty** | **Price (USD)** | **Ext. Price (USD)** | **In-House** |
| ARDUINO UNO SMD R3 ATMEGA328 | A000073 | 1 | 20.90 | 20.90 |  |
| CANBUS SHIELD V2 | 103030215 | 1 | 24.99 | 24.99 |  |
| PNL MNT W/ WIRE 568NM 40MCD GRN | PM5GT | 1 | 0.39 | 0.39 |  |
| PNL MNT W/ WIRE 590NM 40MCD YLW | PM5YT | 1 | 0.39 | 0.39 |  |
| SWITCH ROCKER DPDT 16A 125V | GRS-4022-1600 | 1 | 1.38 | 1.38 |  |
| CONN RCPT HSG 3POS | DT04-3P-L012 | 1 | 4.35 | 4.35 | Yes |
| CONN PIN 14AWG NICKEL CRIMP | 0460-215-16141 | 1 | 0.63 | 1.26 | Yes |
| 18-8 Stainless Steel Socket Cap Screw | 73412 | 4 | - | - | Yes |
| #4 F/W S/S .125x.312 | 71004 | 4 | - | - | Yes |
| 4-40 S/S MS Nut | 1170704 | 4 | - | - | Yes |
| RES 100 OHM 1/4W 5% AXIAL | CF14JT100R | 1 | 0.10 | 0.10 | Yes |
|  |  |  |  |  | Yes |
| AC/DC WALL MOUNT ADAPTER 9V 5W | VEL05US090-US-JA | 1 | 6.50 | 6.50 |  |
| BUMPER SQU 0.812"L X 0.812"W BLK | SJ-5523 (BLACK) | 1 | 1.50 | 1.50 |  |
| USB CABLE TYPE A TO B 30CM BLACK | Seeed Technology Co., Ltd | 1 | 1.99 | 1.99 | Yes |
|  | | | **Total** | **69.36** |  |

The CAN transmitter is based off the *Arduino Uno* development board and the *Seeed Technology CAN BUS shield*. Minor and optional materials, such as solder, heat shrink tubing, and grommets are not included.

## Enclosure Models

The CAN transmitter requires a box and lid. These two *.stl* files are to be 3D printed.

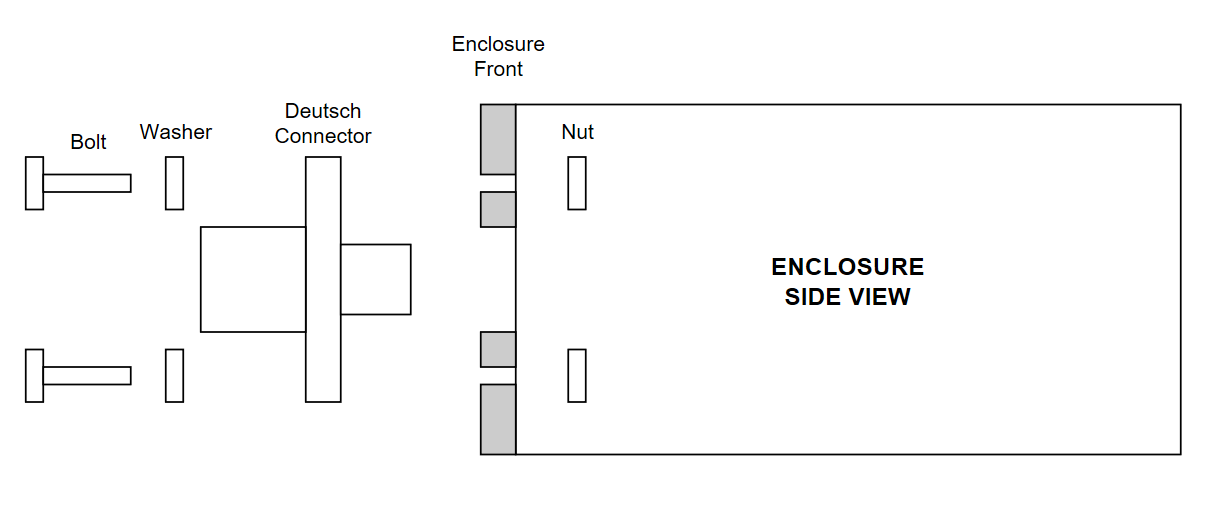
## Arduino Sketch

The CAN transmitter uses the Arduino sketch *CAN\_Transmitter.ino* and the *mcp\_can* library.

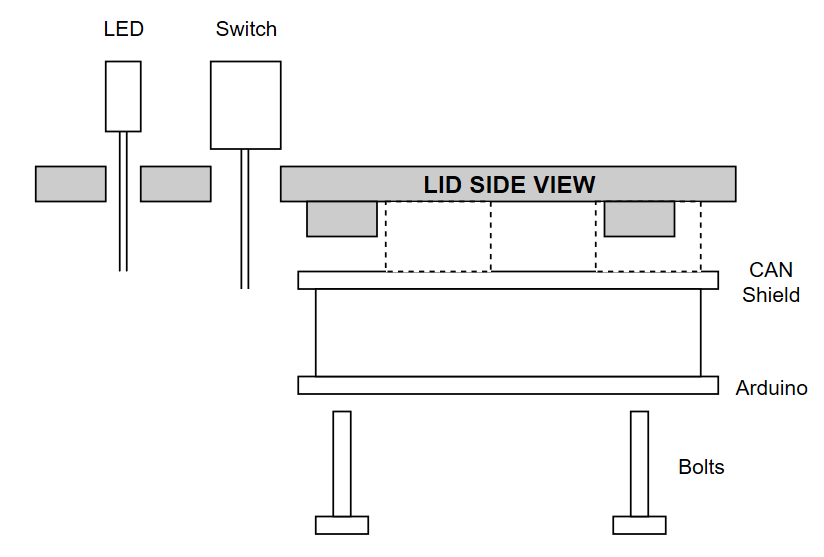
# Construction

## Panel Mount Components

The Deutsch 3 pin connector is mounted on the front face using the bolt, washer, and nut as shown.



The indicator LEDs and switch are snap fitted into the lid cutouts. Pin 1 of the switch is farthest back from the orientation shown below. the LED arrangement does not matter. The CAN shield is attached to the Arduino, and the electronics assembly is screwed into the lid as shown below. It is recommended to label the switch “250kbs” and “500kbs”.



The power supply cable is cut to bare wire and pushed through the small hole at the back of the enclosure. Optionally, a ¼” grommet can be used. It is recommended to tightly knot the wire or use another form of strain relief to protect the solder joints from mechanical stress.

## Electrical Connections

The following electrical connections are made. All connections are soldered except for the Deutsch pins, which are crimped.

|  |  |  |
| --- | --- | --- |
| **Components to be Joined** | | **Notes** |
| Deutsch Pin A | CAN Shield CAN H | Deutsch pin crimped and snap fit into connector |
| Deutsch Pin B | CAN Shield CAN L | Deutsch pin crimped and snap fit into connector |
| Power VIN | Arduino VIN |  |
| Power GND | Arduino GND |  |
| LED Green Cathode | Arduino Pin D5 | Add 100 Ω resistor in series |
| LED Green Anode | Arduino GND |  |
| LED Yellow Cathode | Arduino Pin D6 | Add 100 Ω resistor in series |
| LED Yellow Anode | Arduino GND |  |
| Switch Pin 1 | Arduino Pin D8 |  |
| Switch Pin 2 | Arduino +5V | Optionally add 100 Ω resistor in series |
| Switch Pin 3 | Arduino Pin D9 |  |

## Photos

# Environment Setup

## Install Arduino IDE

Download and run the *Windows Installer* from the link below.

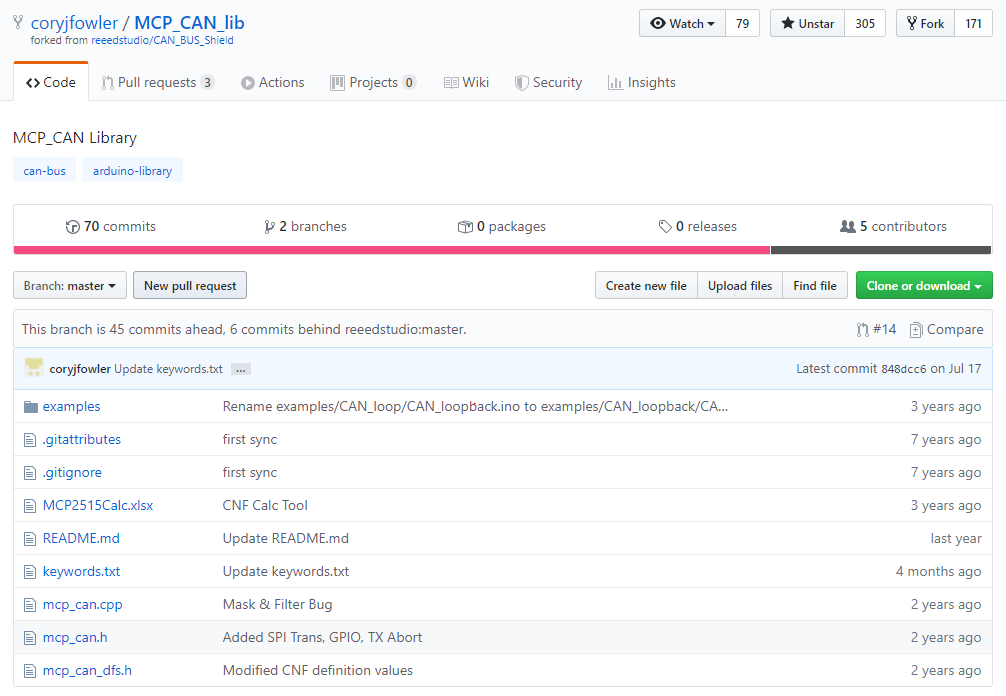
<https://www.arduino.cc/en/Main/Software>



## Install mcp\_can Library

Download the *mcp\_can* library by clicking *Clone or Download*, and *Download ZIP* from the link below.

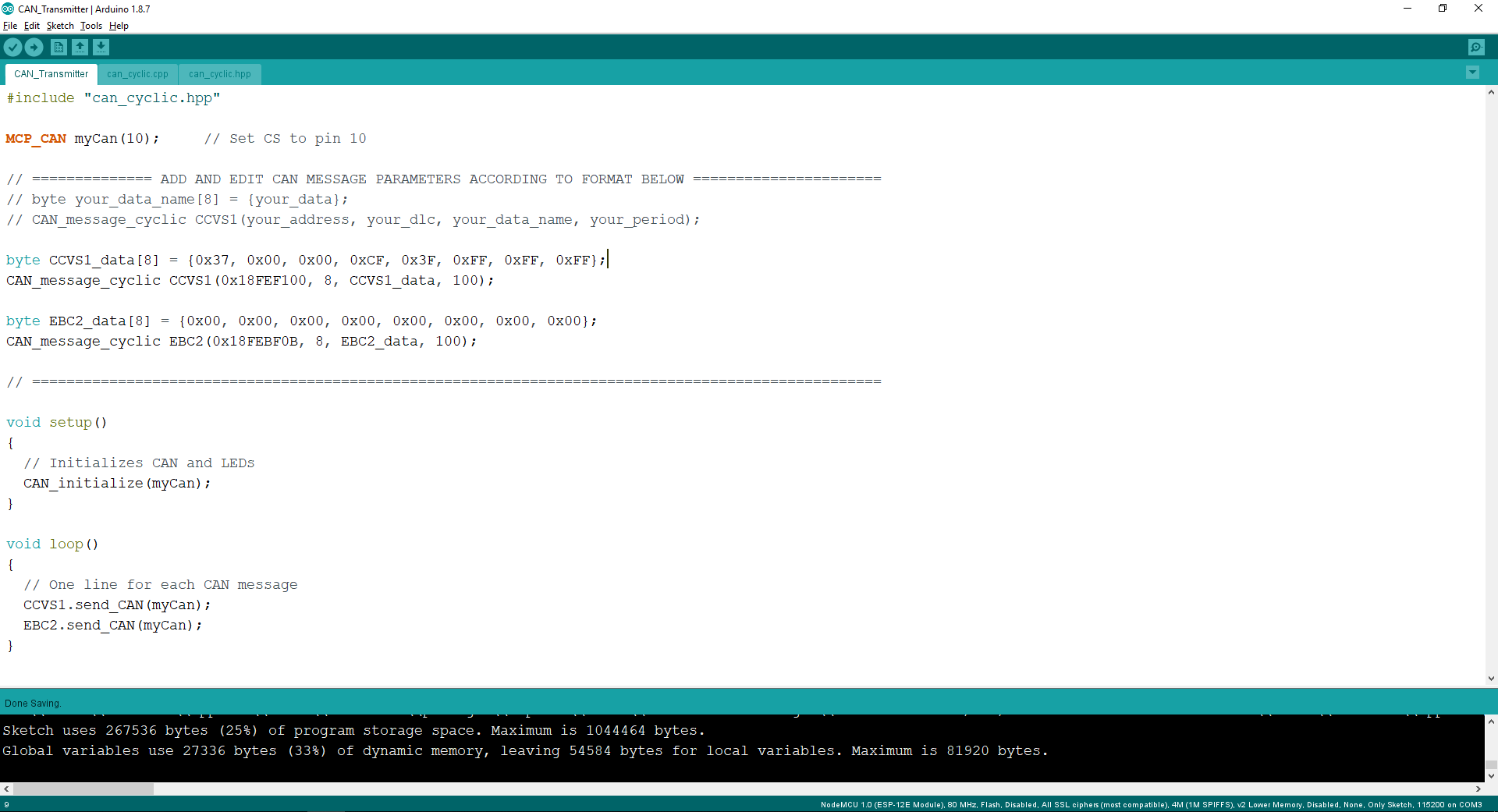
<https://github.com/coryjfowler/MCP_CAN_lib>



To install, open *CAN\_Transmitter.ino*, navigate to *Sketch » Include Library » Add .ZIP Library…*, and add the downloaded *.zip* files.

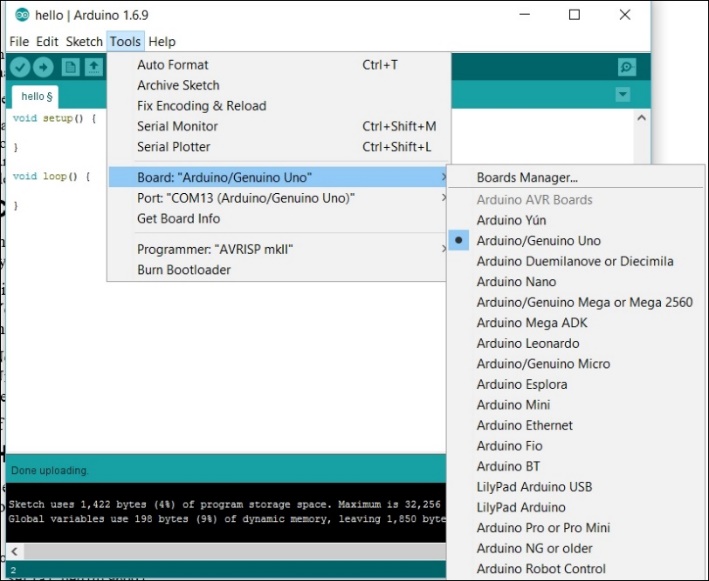
# Configuration and Programming

Open *CAN\_Transmitter.ino*, ensuring it is in the folder with “can\_cyclic.cpp” and “can\_cyclic.hpp”.

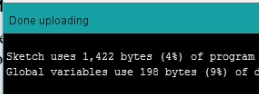
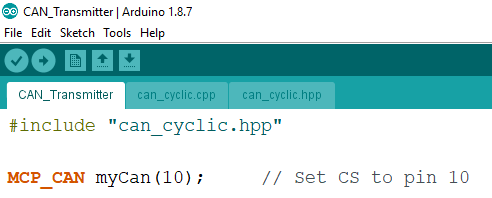


The example above sends the CCVS1 and EBC2 messages with a period of 100ms each. To add a new cyclic CAN message, follow the same format with a new data name.

To program, connect your PC to the Arduino with a USB A Male to USB B Male connector. Under *Tools*, ensure *Board* is selected as *“Arduino/Genuino Uno”*, and *Port* is selected as *“(Arduino/Genuino Uno)”*.



In the top left corner, click the *“Upload”* button, and wait for the *“Done Uploading”* message to appear.



# Use

Plug in the wall adapter, and the *Power* LED will light. Messages will begin transmitting at the baud rate set by the switch. The *Transmit* LED will flash once per message sent.

To switch baud rate, flip the switch. The change will immediately take into effect.

# Licensing and Disclaimer

## CAN Transmitter

The MIT License (MIT)

Copyright (c) 2019 Jonathan Lin

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in

all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

## mcp\_can Library

The MIT License (MIT)

Copyright (c) 2013 Seeed Technology Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in

all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.