Longan-Labs CANbed_Module

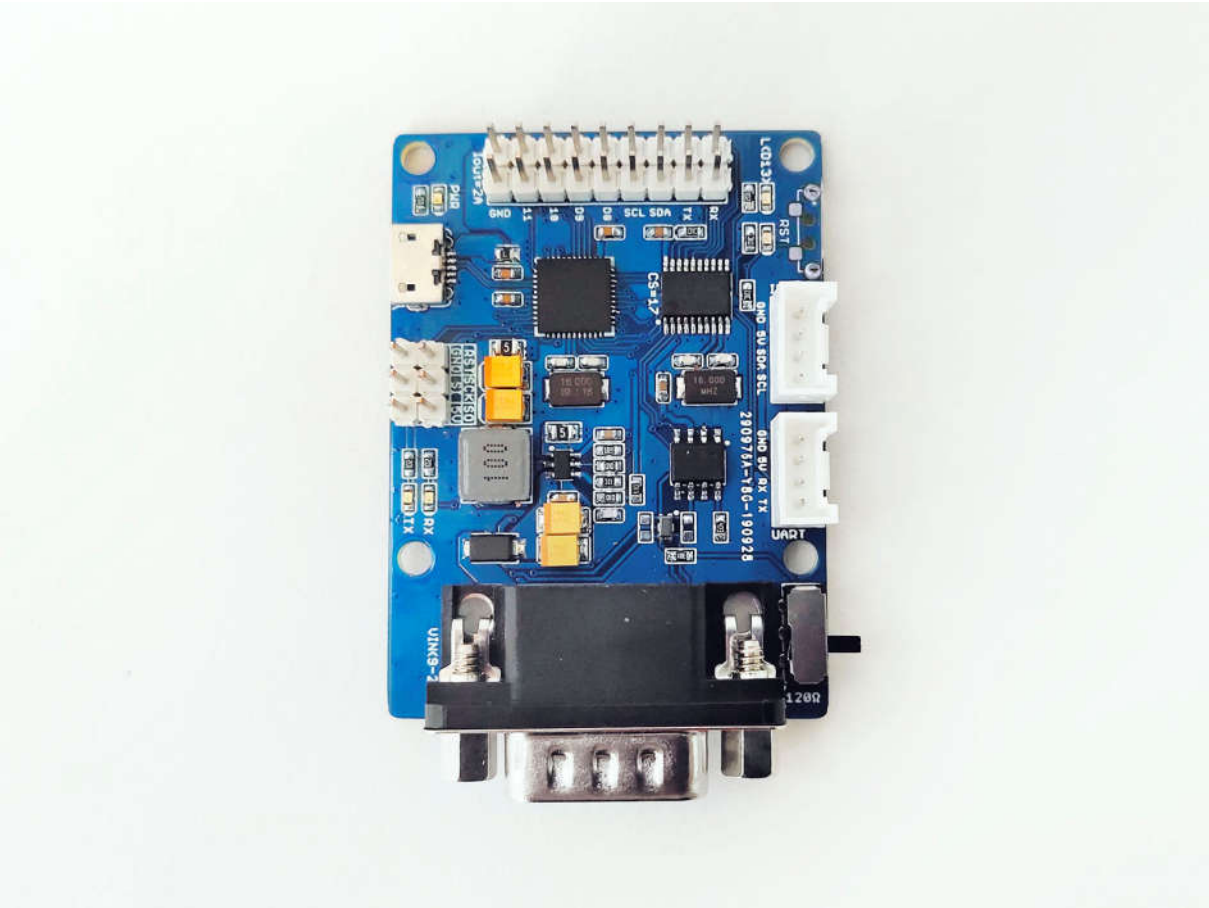
A link to the CANbed module documentation and libraries -

https://docs.longan-labs.cc/1030008/

Table of contents

CANBed V1

Introduction



CAN Bus is a common industrial bus because of its ability to work with an unsophisticated twisted pair over a reasonable distance, medium communication speed and high reliability. It is commonly found on modern machine tools, and modern production vehicles.

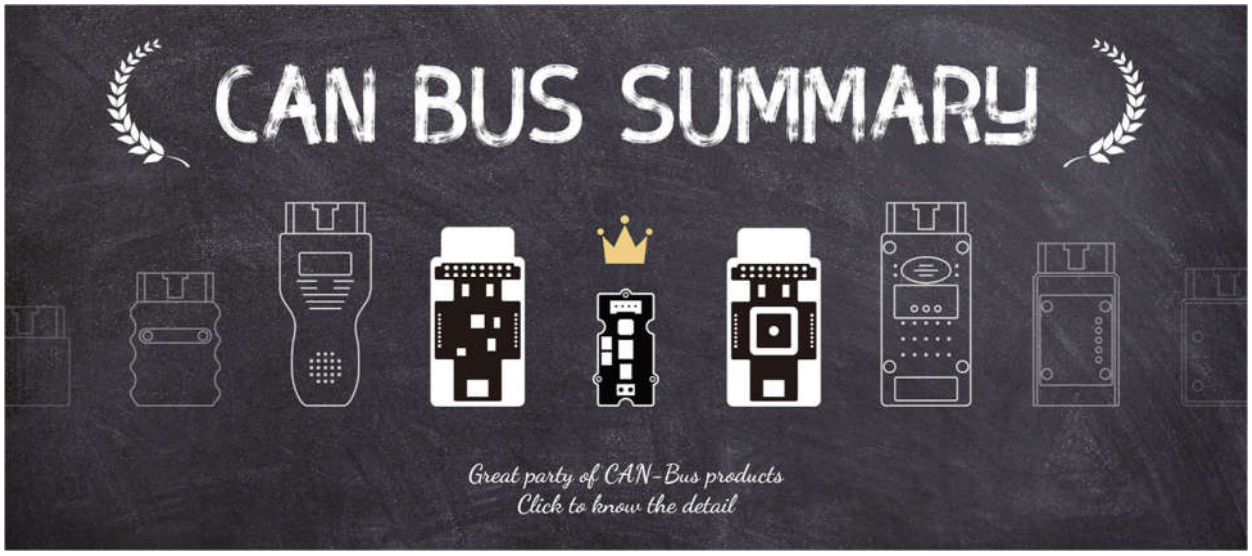
This CANBed adopts a **MCP2515** CAN Bus controller with SPI interface and **MCP2551** CAN transceiver to achieve the CAN-BUS capability. With an **OBD-II** converter cable, and the OBD-II imported library, you are ready to build an on-board diagnostic device.

What's more, a Atmega32U4 CPU with **Arduino Leonardo** bootloader is fitted to the board, and you can use Arduino IDE to easily program the device.

CANBed Family

VERSION	CANBed V1	CANBed FD	CANBed M0	CANBed M4
PICTURES				
MCU	Atmega32U4	Atmega32U4	ATSAMD21G18	ATSAME51G19A
CORE	AVR 8 bit	AVR 8 bit	ARM Cortex M0+ 32bit	ARM Cortex M4 32bit
PROTOCOL	CAN2.0	CANFD & CAN2.0	CAN2.0	CANFD & CAN2.0
CLOCK	16MHz	16MHz	48MHz	120MHz
FLASH	32KB	32KB	256KB	512KB
RAM	2.5KB	2.5KB	32KB	192KB
PRICE	$14.9	$17.9	$16.9	$19.9
LINK	GET ONE NOW	GET ONE NOW	GET ONE NOW	GET ONE NOW

LONGAN LABS CAN BUS PRODUCTS LIST



Logan-Labs have manufactured many different Can-Bus products. You can get more information through the following list, and choose a suitable product.

PRODUCT NAME	LINK	PRICE	MCU	CHIP	PROTOCOL
Serial CAN Bus Module	LINK	$14.9	ATMEGA168PA	MCP2515	CAN2.0
I2C CAN Bus Module	LINK	$14.9	ATMEGA168PA	MCP2515	CAN2.0
OBD-II CAN Bus Dev Kit	LINK	$17.9	ATMEGA168PA	MCP2515	CAN2.0
OBD-II CAN Bus GPS Dev Kit	LINK	$29.9	ATMEGA32U4	MCP2515	CAN2.0
OBD-II CAN Bus Basic Dev Kit	LINK	$19.9	ATMEGA32U4	MCP2515	CAN2.0
CAN-FD Shield	LINK	$8.9	NO MCU	MCP2517FD	CAN-FD
CAN Bus Shield	LINK	$5.9	NO MCU	MCP2515	CAN2.0
CANBed	LINK	$19.9	ATMEGA32U4	MCP2515	CAN2.0
CANBed-FD	LINK	$19.9	ATMEGA32U4	MCP2517FD	CAN-FD
CANBed M0	LINK	$19.9	ATSAMD21	MCP2515	CAN2.0
CANBed M4	LINK	$24.9	ATSAME51	-	CAN-FD

CANBed Features

- Implements CAN V2.0B at up to 1 Mb/s
- Industrial standard 9 pin sub-D connector or 4PIN Terminal.
- OBD-II and CAN standard pinout selectable at sub-D connector
- 2x4Pin Connector compatable with Grove system from Seeedstudio
- SPI Interface up to 10 MHz
- Standard (11 bit) and extended (29 bit) data and remote frames
- Power input from 9-28V

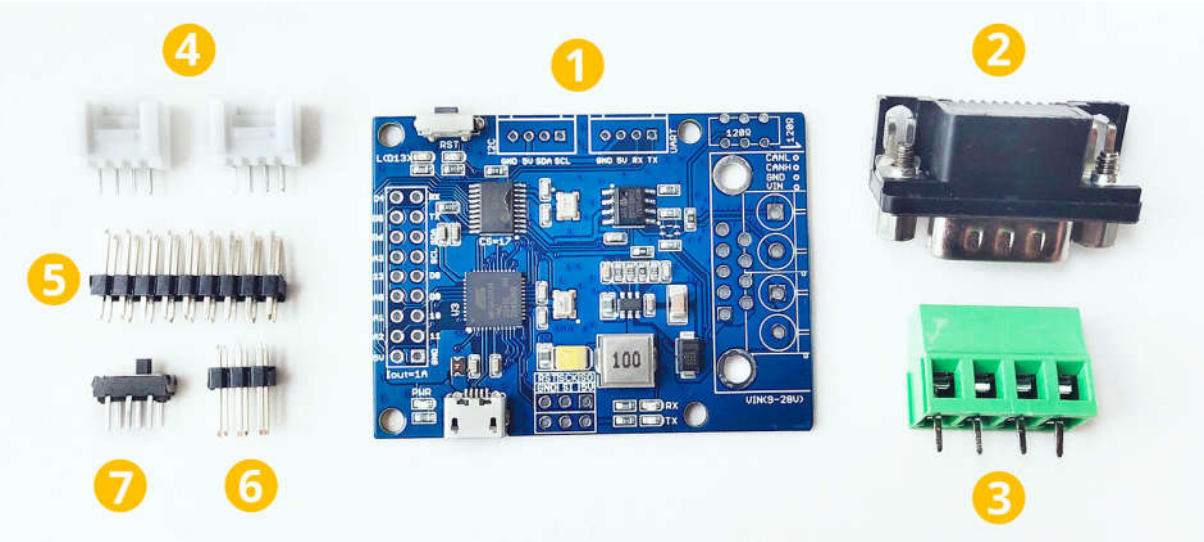
Specifications

Parameter	Value
MCU	Atmega32U4 (with Arduino Leonardo bootloader)
Clock Speed	16MHz
Flash Memory	32KB
SRAM	2.5KB
EERROM	1KB
Operating Voltage	9-28V
* Output Current @ 5V	1A
Input Interface	sub-D

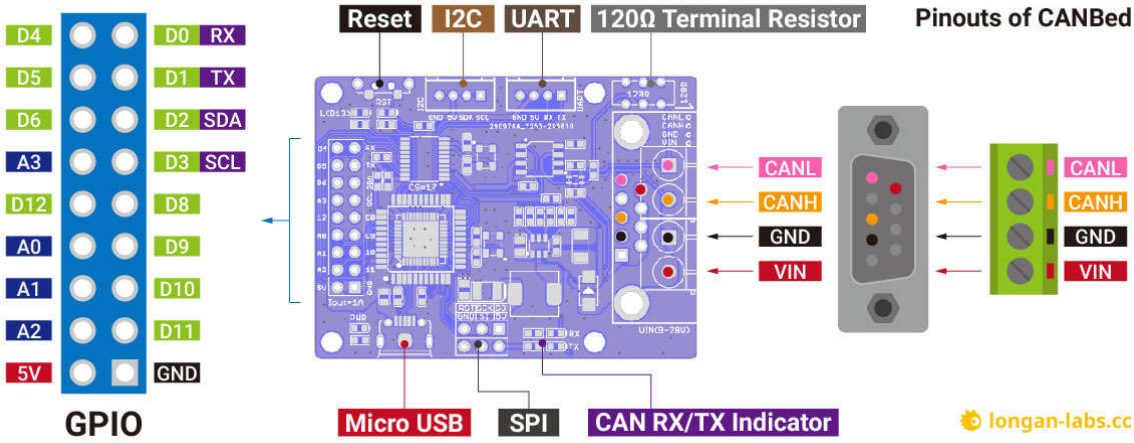
Hardware Overview

Part List

- CANBed PCBA
- sub-D connector
- 4PIN Terminal
- 4PIN HY2.0 Connector x 2
- 9x2 2.54 Header x 1
- 3x3 2.54 Header x 1



Pin out



1. 9x2 IO Pin OUT:

The Atmega32U4 IO is listed here.

2. Atmega32U4:

The master of the entire module, mainly used to store data on the TF card or transfer data to the computer through the type C cable. In addition, since it's Arduino compatible, you can use it to implement some simple controls, such as triggering a buzzer alarm when the speed exceeds a certain value.

2. Reset Button:

Resets the on-board Atmega chip.

3. Micro USB connector for programming

4. ICSP Header for uploading bootloader

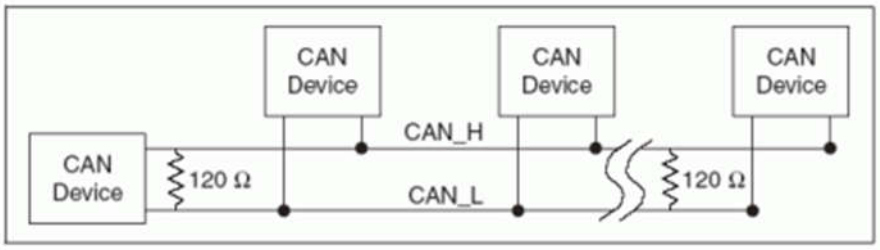
5. CAN RX/TX Indicator

6. sub-D connector or Terminal for CAN Bus

D-Sub CANbus PinOut

pin#	Signal names	Signal Description
1	Reserved	Upgrade Path
2	CAN_L	Dominant Low
3	CAN_GND	Ground
4	Reserved	Upgrade Path
5	CAN_SHLD	Shiled, Optional
6	GND	Ground, Optional
7	CAN_H	Dominant High
8	Reserved	Upgrade Path
9	CAN_V+	Power, Optional

7. Switch for the 120Ω terminal resistor for CAN Bus



If you use this board on the end of the CAN bus, please set this switch to 120Ω. For more detail about the CAN bus protocol, please refer to the NI CAN Physical Layer and Termination Guide

8. Grove connector for UART

Note

Use Serial1 in the code

9. Grove connector for I2C

Usage

IDE and Driver

There's Arduino leonardo bootloader in the board.

Click this link to get the latest Arduino IDE.

There's Arduino Leonardo driver in the folder: Arduino/drivers.

After the IDE is installed, please select **Arduino Leonardo** in at Tools > Board

Arduino Code

We provide an Arduino library for CANBed.

Note

Please change the SPI_CS_PIN to 17 for CANBed V1

There're many examples in the library, which is consists of,

- ***OBDII-PIDs*** - Get data from OBD-II interface
- ***send*** - Send a frame to CAN Bus
- ***recv*** - Receive a frame from CAN Bus
- ***set_mask_filter_recv*** - Receive a frame from CAN Bus with mask and filter setting

APIs

1. Set the Baud rate

This function is used to initialize the baud rate of the CAN Bus system.

The available baud rates are listed as follows:

```
#define CAN_5KBPS    1
#define CAN_10KBPS   2
#define CAN_20KBPS   3
#define CAN_25KBPS   4
#define CAN_31K25BPS 5
#define CAN_33KBPS   6
#define CAN_40KBPS   7
#define CAN_50KBPS   8
#define CAN_80KBPS   9
#define CAN_83K3BPS  10
#define CAN_95KBPS   11
#define CAN_100KBPS  12
#define CAN_125KBPS  13
#define CAN_200KBPS  14
#define CAN_250KBPS  15
#define CAN_500KBPS  16
#define CAN_666kbps  17
#define CAN_1000KBPS 18
```

2. Set Receive Mask and Filter

There are **2** receive mask registers and **5** filter registers on the controller chip that guarantee you getting data from the target device. They are useful especially in a large network consisting of numerous nodes.

We provide two functions for you to utilize these mask and filter registers. They are:

Mask:

```
init_Mask(unsigned char num, unsigned char ext, unsigned char
ulData);
```

Filter:

```
init_Filt(unsigned char num, unsigned char ext, unsigned char
ulData);
```

- **num** represents which register to use. You can fill 0 or 1 for mask and 0 to 5 for filter.
- **ext** represents the status of the frame. 0 means it's a mask or filter for a standard frame. 1 means it's for a extended frame.
- **ulData** represents the content of the mask of filter.

3. Check Receive

The MCP2515 can operate in either a polled mode, where the software checks for a received frame, or using additional pins to signal that a frame has been received or transmit completed.

Use the following function to poll for received frames.

```
INT8U MCP_CAN::checkReceive(void);
```

The function will return 1 if a frame arrives, and 0 if nothing arrives.

4. Get the CAN ID

When some data arrives, you can use the following function to get the CAN ID of the "send" node.

```
INT32U MCP_CAN::getCanId(void)
```

5. Send a frame

```
CAN.sendMsgBuf(INT8U id, INT8U ext, INT8U len, data_buf);
```

It is a function to send data onto the bus. In which:

- **id** represents where location of the data source.

- **ext** represents the status of the frame. '0' means standard frame. '1' means extended frame.
- **len** represents the length of this frame.
- **data_buf** is the content of this message.

For example, in the 'send' example, we have:

```
unsigned char stmp[8] = {0, 1, 2, 3, 4, 5, 6, 7};
CAN.sendMsgBuf(0x00, 0, 8, stmp); //send out the message 'stmp' to the bus
and tell other devices this is a standard frame from 0x00.
```

6. Receive a frame

The following function is used to receive data in the 'receive' node:

```
CAN.readMsgBuf(unsigned char len, unsigned char buf);
```

In conditions where masks and filters have been set, Ths function can only get frames that meet the requirements of masks and filters.

- **len** represents the data length.
- **buf** is where you store the data.

FAQ

Q, I can't upload code to CANBed FD

A. If your PC recognizes the COM port, please try pressing the reset button, then click on the Upload button in Arduino IDE, when the IDE shows compile done, release the reset button immediately.

Q. The RX/TX led light up and never turns off

A. Check the following:

- Check if the baudrate of CAN Bus setting is correct
- Try turning the switch for the terminal resistor on/off
- Check if CANH and CANL is connected correctly

How to find the technical support

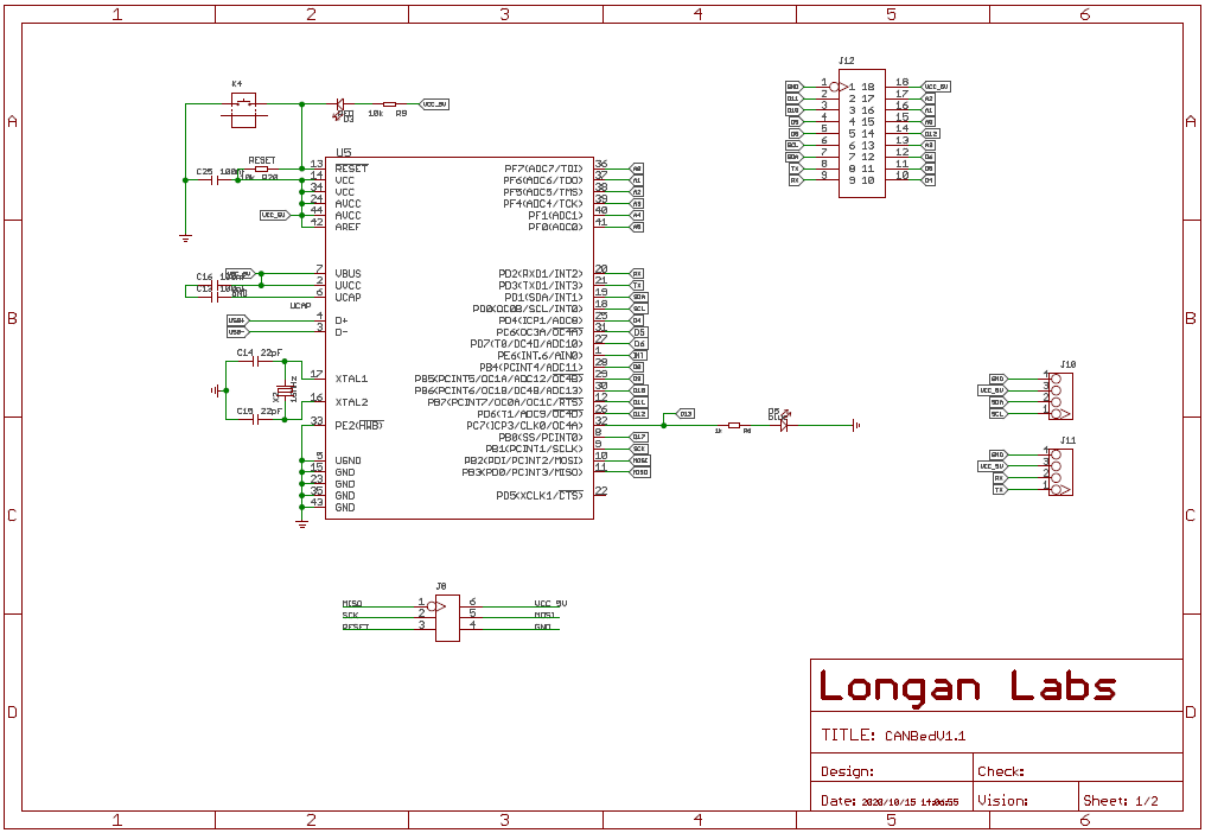
Please feel free to contact joney.sui@longan-labs.cc if you need more help.

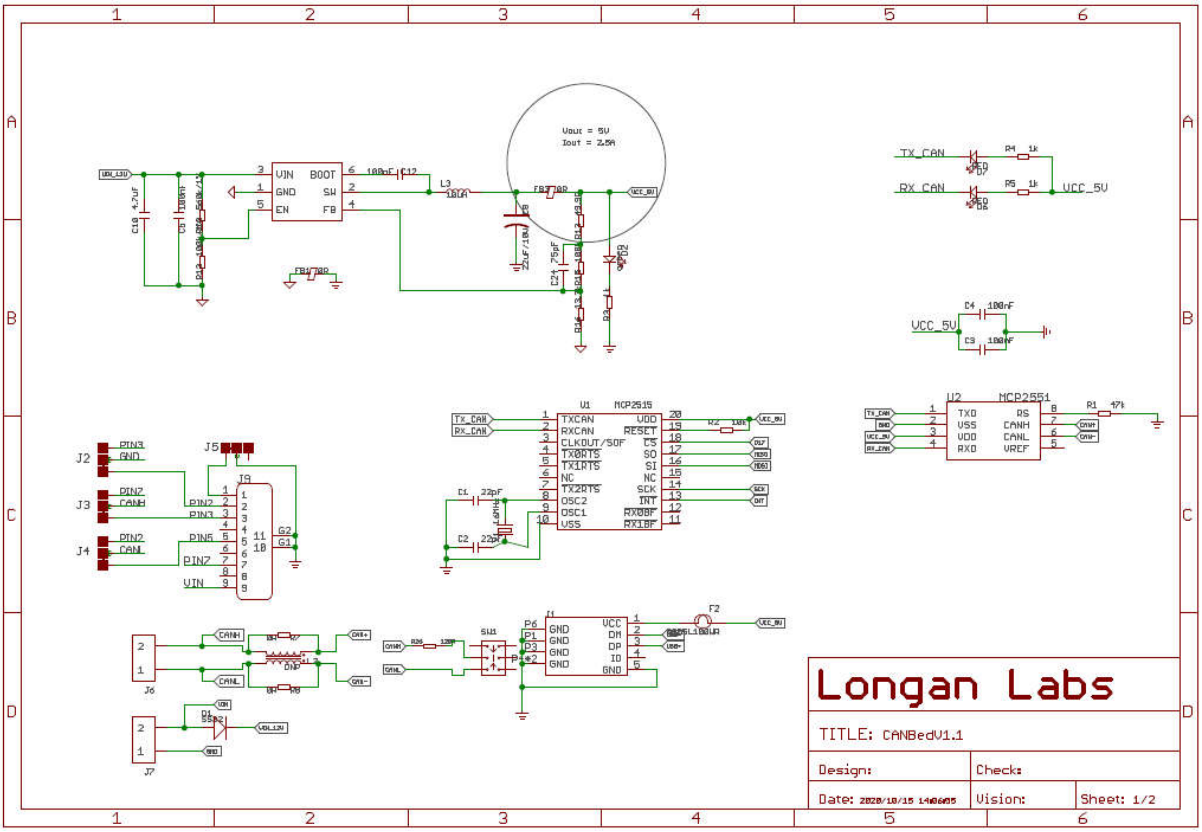
Reference

- Arduino Library
- Schematics in Eagle
- Schematics in PDF

- MCP2515 Datasheet
- MCP2551 Datasheet

Schematics




Vout = 5V
Iout = 2.5A
VIN
BOOT
GND
SW
EN
FB
TX_CAN
RX_CAN
VCC_5V
MCP2515
MCP2551
TXCAN
RXCAN
CLKOUT/SOF
TX0RTS
TX1RTS
OSC2
OSC1
VSS
VDD
RESET
CS
SO
SI
NC
SCK
INT
RX0BF
RX1BF
TXD
RXD
RS
CANH
CANL
VREF
J2
J3
J4
J5
J6
J7
PIN3
PIN2
PIN5
GND
CANH
CAN
VIN
G2
G1
DNP
Longan Labs
TITLE: CANBedV1.1
Design:
Check:
Date: 2020/10/15 140605
Vision:
Sheet: 1/2