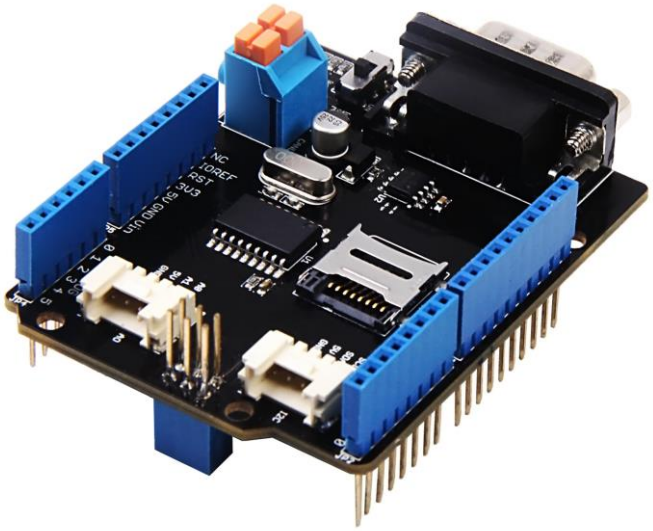
CAN-BUS Shield V2.0



CAN-BUS is a common industrial bus because of its long travel distance, medium communication speed and high reliability. It is commonly found on modern machine tools, such as an automotive diagnostic bus.

This CAN-BUS Shield adopts **MCP2515** CAN Bus controller with SPI interface and **MCP2551** CAN transceiver to give your Arduino/Seeeduino CAN-BUS capability. With an **OBD-II** converter cable added on and the OBD-II library imported, you are ready to build an onboard diagnostic device or data logger.

Previously we have made two versions of CAN-BUS Shield, the V1.0 and V1.2. They are all awesome shields that widely liked by our users. In order to make it better, several months ago we conducted a survey about CAN-BUS Shield V1.2 and received many valuable advices (Thanks to all the users who replied to us), so we decided to make an update and here it is - CAN-BUS Shield V2.

What's new in CAN BUS Shield V2.0

Hardware

- OBD-II or CAN standard pinout can be selected by switching jumpers on DB9 interface, the default pinout is OBD-II.
- Add a TF card slot for data storage and the CS pin can be either set to D4 or D5.
- The INT pin can be set to D2 or D3 by cutting and soldering pad on the back of the shield.
- Moved the P1 pad from front to the back of the shield to make it easier to cut and solder.
- Consider that the D0/D1 pin are usually used for downloading code, we changed the serial Grove connector to pin A0/A1.
- The I2C grove connector is also changed to more reasonable standard SDA/SCL pin instead of previous A4/A5.

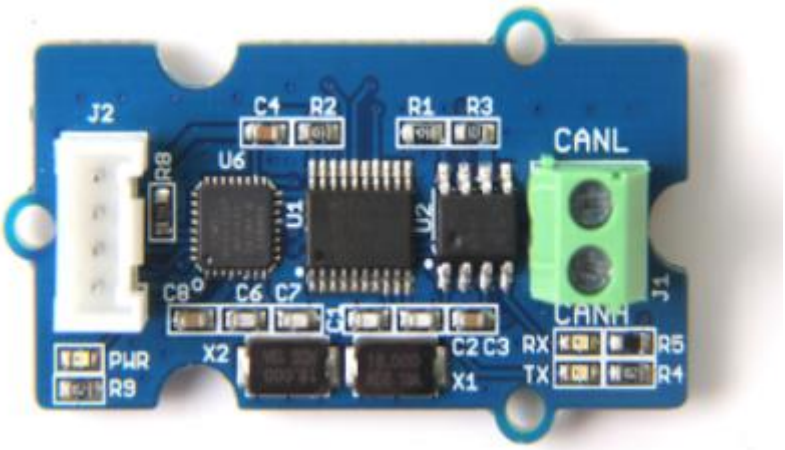
- The two grove connectors are both changed to horizontal rather than vertical to the shield so that it would be more convenient when connecting to other grove modules.

Software

- Add the function and example to access the data of your car.
- Add the function to read the SD card.
- Add the example to store the data of your car into the SD card.
- Fix some bugs and optimize some program.

D-Sub CANbus PinOut

1	Reserved	Upgrade Path
2	CAN_L	Dominant Low
3	CAN_GND	Ground
4	Reserved	Upgrade Path
5	CAN_SHLD	Shield, Optional
6	GND	Ground, Optional
7	CAN_H	Dominant High
8	Reserved	Upgrade Path
9	CAN_V+	Power, Optional



What if I want to connect this shield to my car

If you want to read data or control your car, there's an OBD>DB9 cable available for you, this cable make easier to connect to OBD-connector and DB9-connector. This cable will also work with anything that has a OBD-connector. Add a power switch makes such a satisfying click.

USB-CAN Analyser

If you want a CAN Bus Analyser to debug your CAN Bus, this USB-CAN Analyser is recommended.

Features

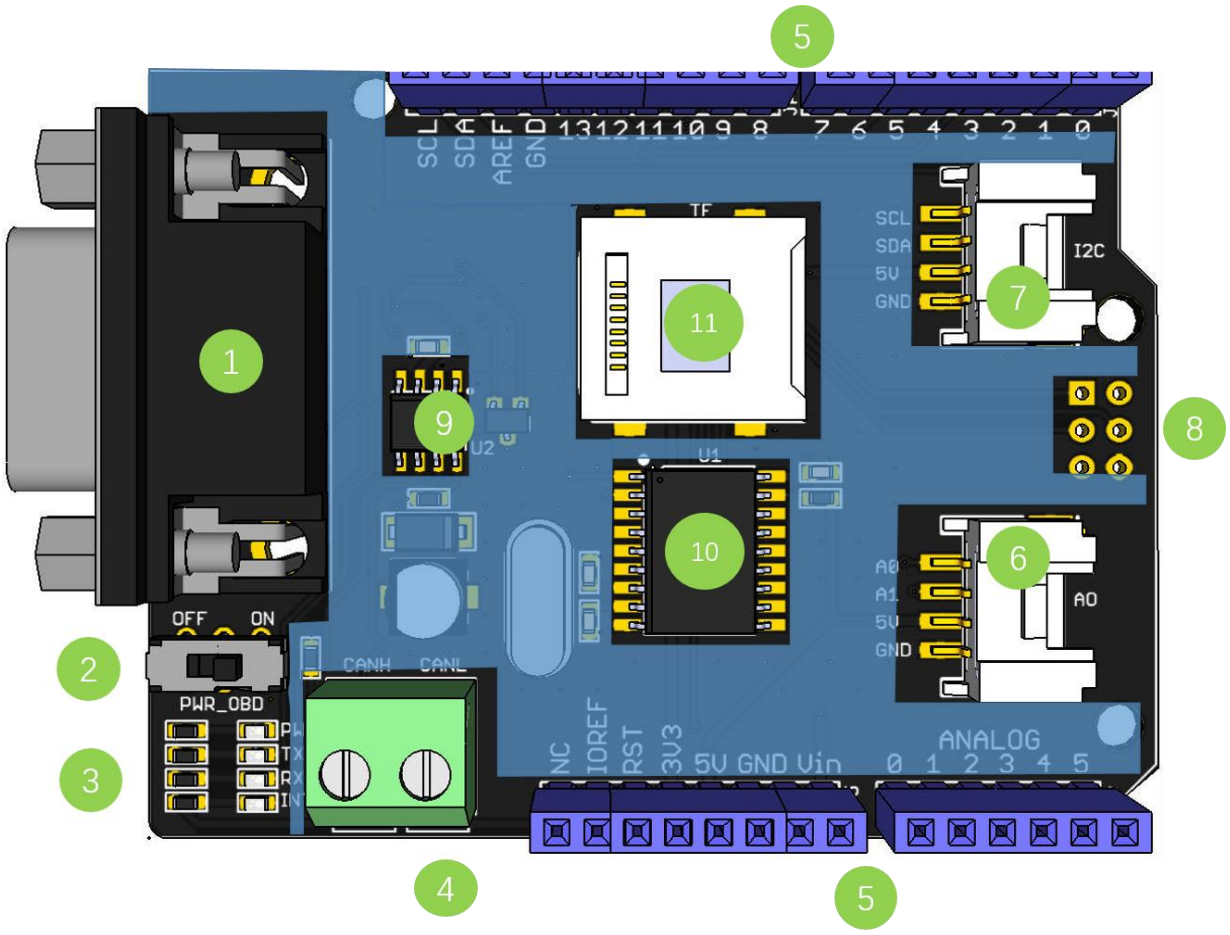
- Implements CAN V2.0B at up to 1 Mb/s
- Industrial standard 9 pin sub-D connector
- OBD-II and CAN standard pinout selectable.
- Changeable chip select pin
- Changeable CS pin for TF card slot

- Changeable INT pin
- Screw terminal that easily to connect CAN_H and CAN_L
- Arduino Uno pin headers
- 2 Grove connectors (I2C and UART)
- SPI Interface up to 10 MHz
- Standard (11 bit) and extended (29 bit) data and remote frames
- Two receive buffers with prioritized message storage

Note

CAN BUS Shield Work well with Arduino UNO (ATmega328), Arduino Mega (ATmega1280/2560) as well as Arduino Leonardo (ATmega32U4).

Hardware Overview¶



1. **DB9 Interface** - to connect to OBDII Interface via a DBG-OBD Cable.
2. **V_OBD** - It gets power from OBDII Interface (from DB9)
3. **Led Indicator**:
    - **PWR**: power

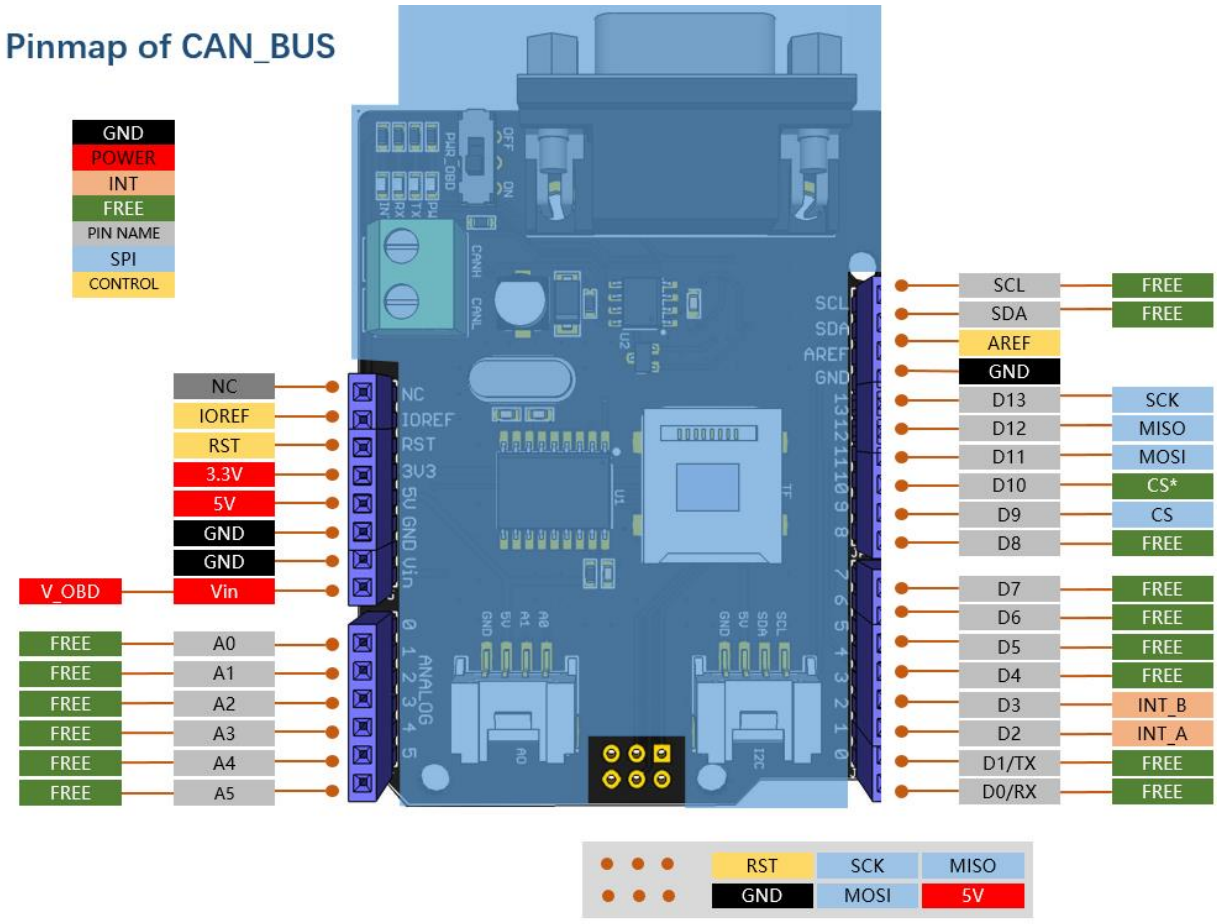
- **TX**: blink when the data is sending
- **RX**: blink when there's data receiving
- **INT**: data interrupt

4. **Terminal** - CAN_H and CAN_L
5. **Arduino UNO pin out**
6. **Serial Grove connector**
7. **I2C Grove connector**
8. **ICSP pins**
9. **IC** - MCP2551, a high-speed CAN transceiver (datasheet)
10. **IC** - MCP2515, stand-alone CAN controller with SPI interface (datasheet)
11. **SD card slot**

Warning

When you use more than two CAN Bus Shield in one net, you should take the impedance into consideration. You should either cut P1 in the PCB with a knife, or just remove R3 on the PCB.

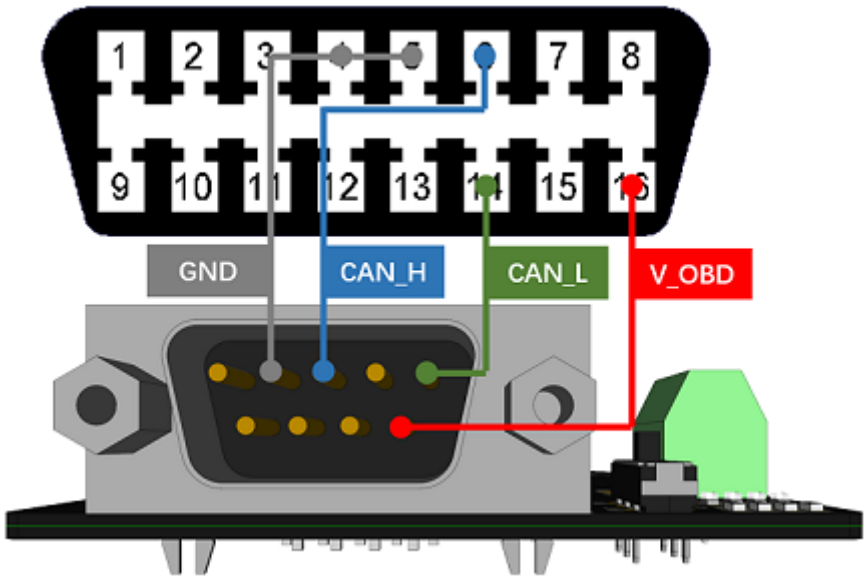
Pin map



Note

- The FREE pin is available for the other usages.

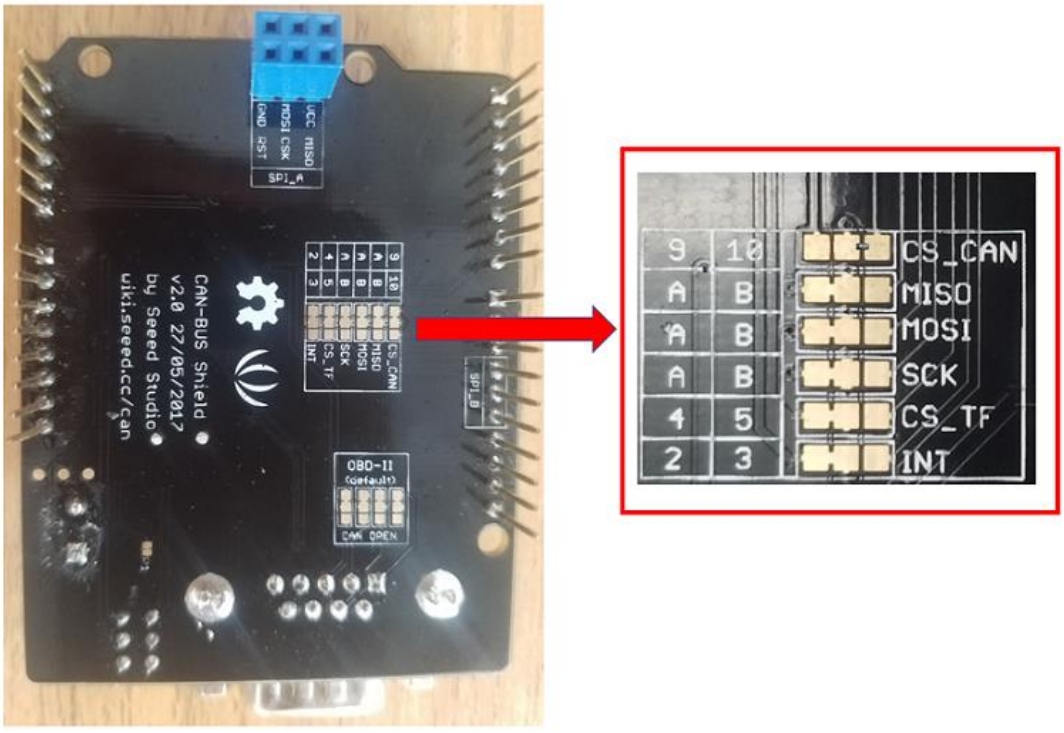
DB9&OBDii Interface



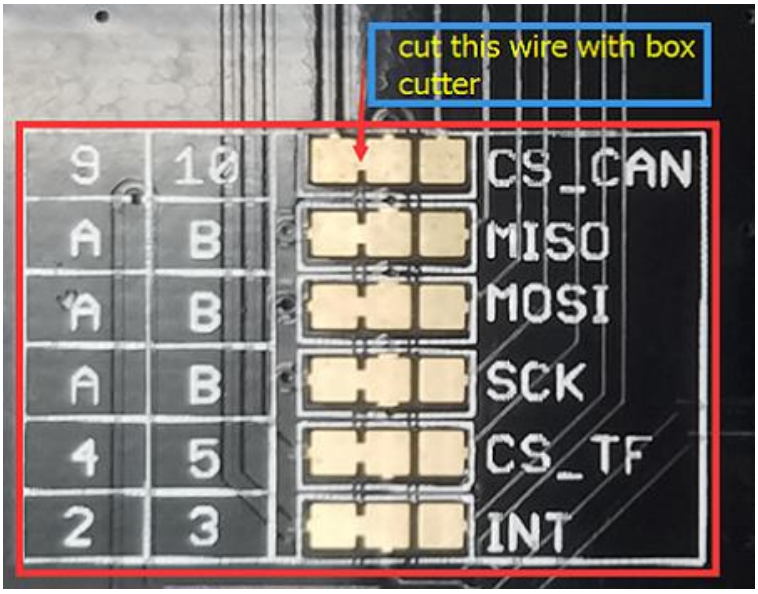
CS_CAN pin

SPI_CS pin of V2.0 is connected to **D9** by default. If you want to change to **D10**, please follow below instructions.

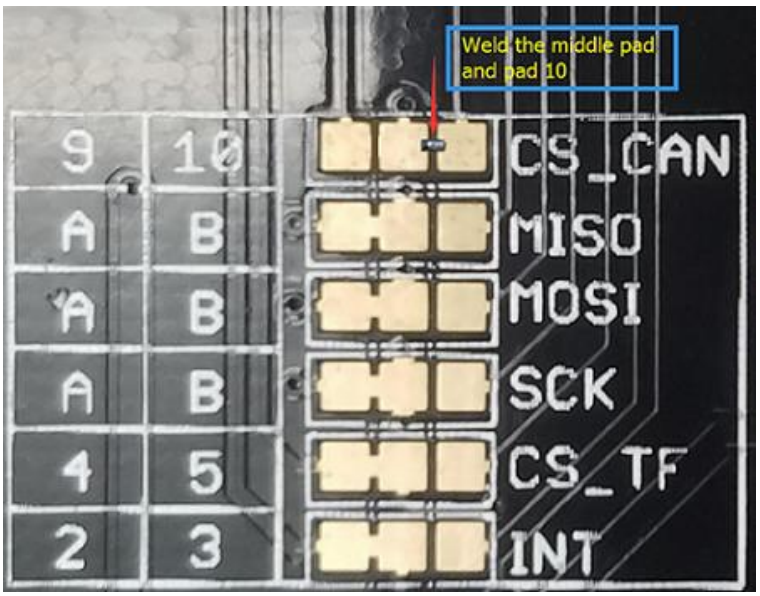
- Step1: Take a look at the backside of the PCBA, you will find a pad named CS_CAN.



- Step2: Cut the wire between pad9 and the middle pad.



- Step3: Weld the middle pad and pad 10.

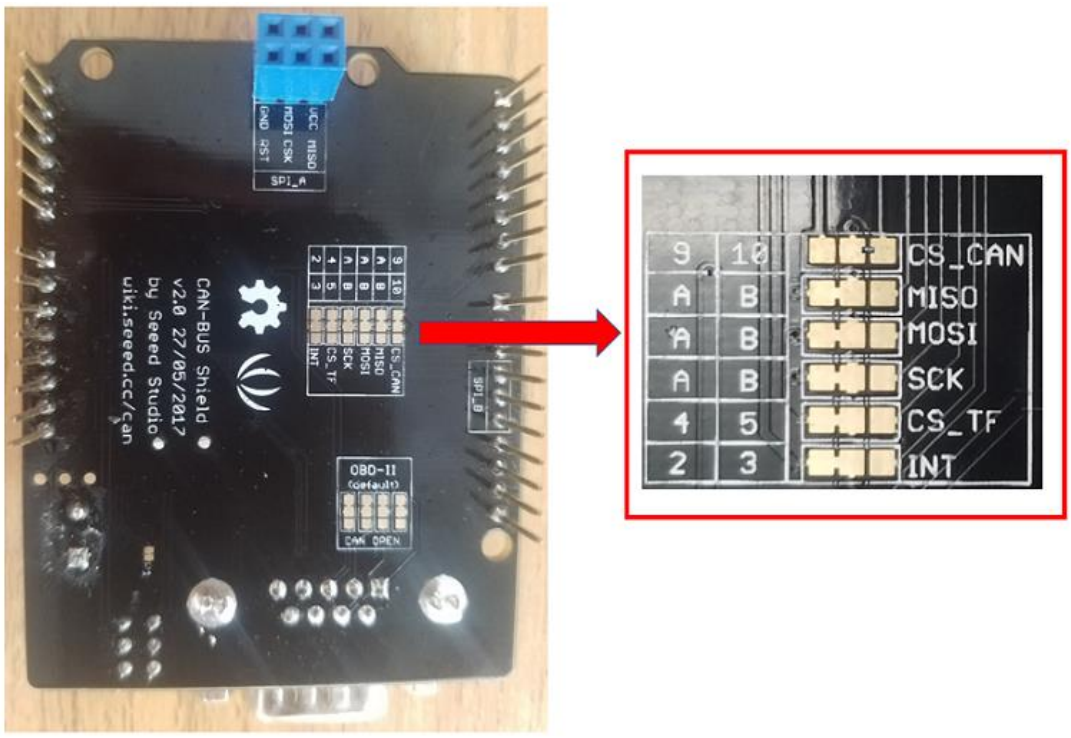


Warning
Be careful with the box cutter, it's easy to hurt yourself or the PCBA.

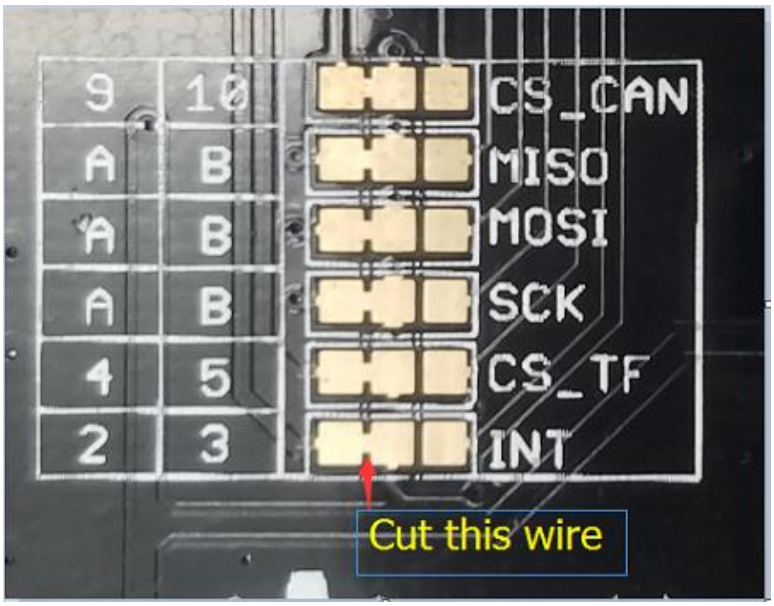
INT pin

INT pin of V2.0 is connected to **D2** by default. If you want to change to **D3**, please follow below instructions.

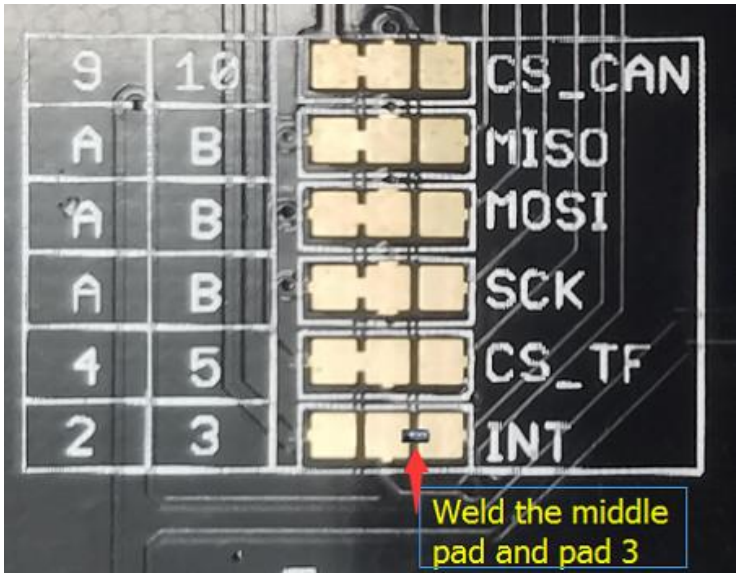
- Step1: Take a look at the backside of the PCBA, you will find a pad named INT.



- Step2: Cut the wire between pad 2 and the middle pad.



- Step3: Weld the middle pad and pad 3.



SPI pins

The SPI pins (SCK, MISO, MOSI) are routed to the ICSP pins by default. But for some boards, the SPI pins are located at D11~D13. if this happens, you need make some change to the PCBA. Take a look at the backside of the PCBA, there're three pads, MOSI, MISO and SCK, they are connected to A by default. You can change them to B if needed.

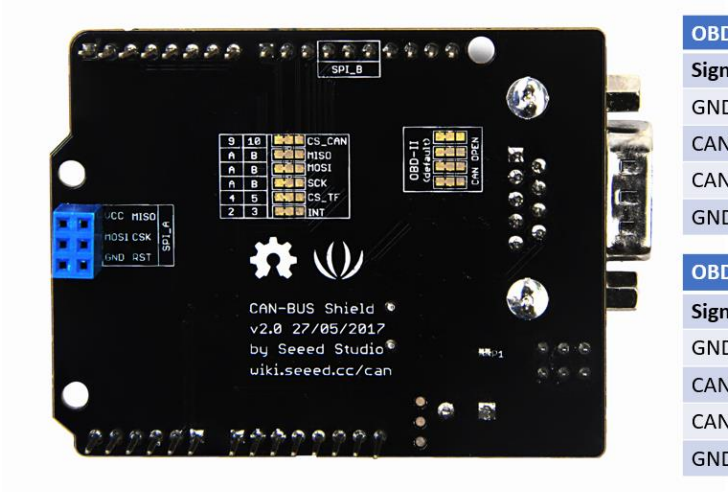
Note

For Arduino UNO, Arduino Mega, Arduino Leonardo and any others AVR based Arduino boards, it works well by default setting.

Warning

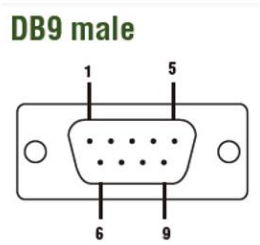
Be careful when you are going to change SPI pins, it's easy to hurt yourself or the PCBA.

ODB pins



OBD–II by Default	
Signal	**DB9**
GND	PIN2
CAN_H	PIN3
CAN_L	PIN5
GND	PIN1

OBD–II Right PAD	
Signal	**DB9**
GND	PIN3
CAN_H	PIN7
CAN_L	PIN2
GND	PIN1



Warning

Please do not cut the forth left PAD connection. Because there is no signal connected with forth right PAD.

Getting Started

Here's a simple example to show you how CAN-BUS Shield works. In this example we need 2 pieces of CAN-BUS Shields as well as Arduino or Seeeduino.

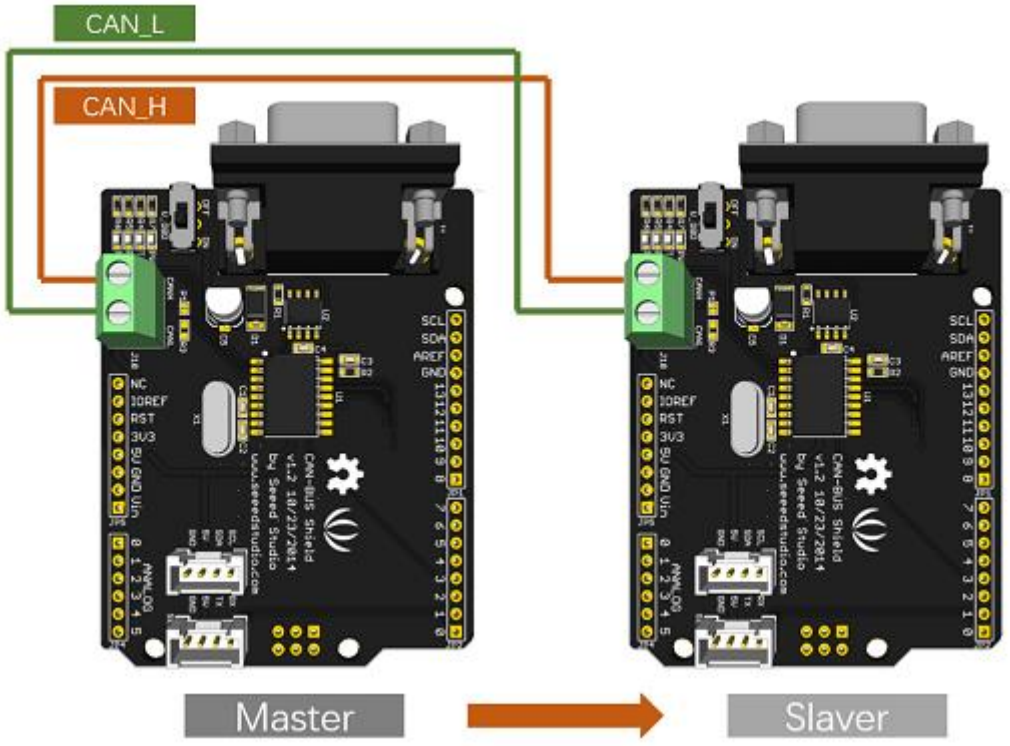
Note
This example is built under Arduino IDE version 1.6.9.

STEP1: What do we need

CAN-BUS Shield	CAN Bus communication	2	link
Seeeduino V4.2	Controller	2	link
Jumper Wire	connection	2	link

STEP2: Hardware Connection

Insert each CAN-BUS Shield into Seeeduino V4.2, and connect the 2 CAN-BUS Shield together via 2 jumper wires. Shown as below images.



Note
CAN_H to CAN_H, CAN_L to CAN_L

STEP3: Software

Please follow how to install an Arduino library procedures to install CAN BUS shield library.

Click on below button to download the library.

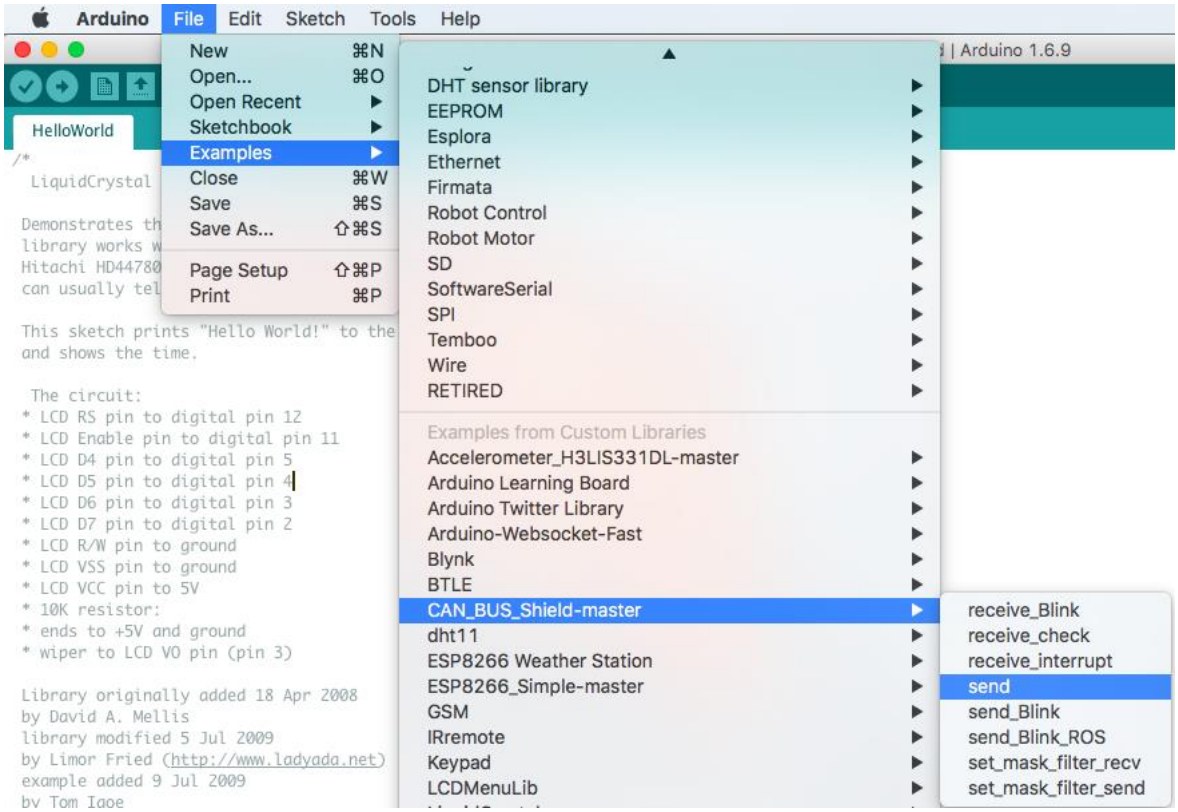
Download CAN BUS Shield Library

Install the library to your Arduino IDE when it is downloaded.

One of the node (a node means Seeeduino + CAN_BUS Shield) acts as master, the other acts as slaver. The master will send data to slaver constantly.

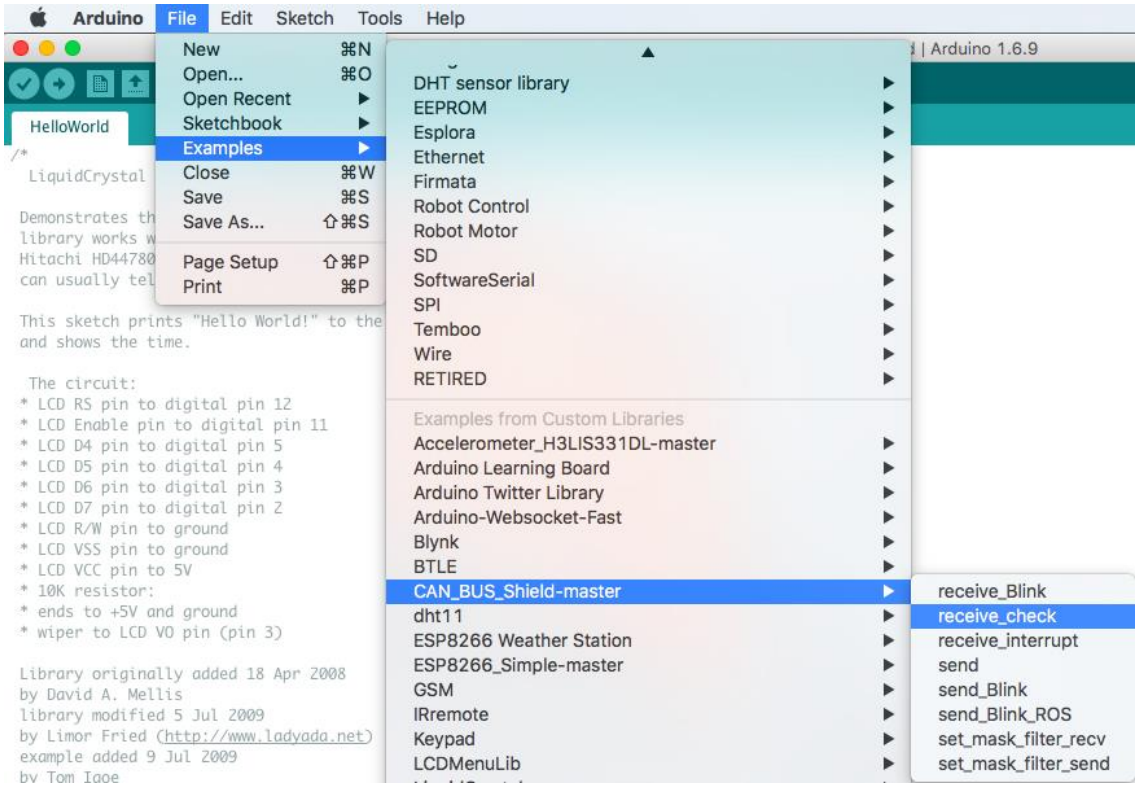
Note
Each node can act as master before the code being uploaded.

Open the **send** example (**File > Examples > CAN_BUS_Shield-master > send**) and upload to the **master**.



Open the **receive_check** example (**File > Examples > CAN_BUS_Shield-master > receive_check**) and upload to

the **slaver**.



STEP4: View Result

Open the Serial Monitor of Arduino IDE(**slaver**), you will get the data sent from the master.