

We use some essential cookies to make our website work.

We use optional cookies, as detailed in our [cookie policy](#), to remember your settings and understand how you use our website.

[Accept optional cookies](#)

[Reject optional cookies](#)



Raspberry Pi

- [Forum FAQ](#)
- [About](#)
 - [About us](#)
 - [Contact us](#)
 - [Documentation](#)
 - [Trademark rules](#)

- [Login](#)
- [Register](#)
- -
 - [Unanswered topics](#)
 - [Active topics](#)
 -
 - [Forum FAQ](#)
 - [About](#)
 - [About us](#)
 - [Contact us](#)
 - [Documentation](#)
 - [Trademark rules](#)
- [Board index Hardware and peripherals Interfacing \(DSI, CSI, I2C, etc.\)](#)

[quick-guide] CAN bus on raspberry pi with MCP2515 and Arduino

Locked

- [Print view](#)

Sunday

1 post • Page 1 of 1

Posts: 5

Joined: Mon Dec 07, 2020 5:00 pm

Location: san jose

[quick-guide] CAN bus on raspberry pi with MCP2515 and Arduino

Fri Dec 25, 2020 4:08 am

Hi,

I am seeing many people who want to establish CAN communication between Raspberry Pi and Arduino and that made me post the solution here .

VERY IMPORTANT NOTE:

1. Most of my steps are from [viewtopic.php?t=141052](https://www.raspberrypi.org/forums/viewtopic.php?t=141052) (<https://www.raspberrypi.org/forums/viewtopic.php?t=141052>) , Please check that out before coming here

2. You should not give 5V to MCP2515 as it may damage your Raspberry pi GPIOs , So there is a step for giving 3.3V to MCP2515(CAN controller) and 5V for TJA1050(CAN transceiver) , That soldering step is available in [viewtopic.php?t=141052](https://www.raspberrypi.org/forums/viewtopic.php?t=141052) (<https://www.raspberrypi.org/forums/viewtopic.php?t=141052>) , Please do that first and come here or else your raspberry pi will be damaged.

[OR]

You can use a logic level shifter similar to this <https://learn.sparkfun.com/tutorials/bi-directional-logic-level-converter-hookup-guide/all> (<https://learn.sparkfun.com/tutorials/bi-directional-logic-level-converter-hookup-guide/all>) , Please share if you do so!!

Connections:

Code:

```
RPi to CAN module
3.3V (Physical Pin 1) <==> Vcc CAN module
5V (Physical Pin 2) <==> soldered capacitor behind CAN module
GND (Physical Pin 6) <==> GND CAN module
CE0 (Physical Pin 24) <==> CS CAN module
MISO (Physical Pin 21) <==> SO CAN module
MOSI (Physical Pin 19) <==> SI CAN module
SCLK (Physical Pin 23) <==> SCK CAN module
GPIO-BCM-12 (Physical Pin 32) <==> INT CAN module
```

Code:

```
UNO to CAN module
5V <==> Vcc
GND <==> GND
Pin 13 <==> SCK
Pin 12 <==> SO
Pin 11 <==> SI
Pin 10 <==> CS
```

Image at bottom of post for connections



Image

Do the following :

1.Open the configurations file

Code:

```
sudo nano /boot/config.txt
```

2.Add these lines

Code:

```
dtparam=spi=on
dtoverlay=mcp2515-can0,oscillator=8000000,interrupt=12
dtoverlay=spi-bcm2835-overlay
```

(if dtparam=spi=on is uncommented and existing , please leave it)

3.

Code:

```
sudo apt-get install can-utils
```

4.

Code:

```
sudo reboot
```

5.The following should throw similar output

Code:

```
ls /sys/bus/spi/devices/spi0.0
driver modalias net of_node power statistics subsystem uevent
ls /sys/bus/spi/devices/spi0.0/net
can0
ls /sys/bus/spi/devices/spi0.0/net/can0/
addr_assign_type dev_id link_mode proto_down
address dev_port mtu queues
addr_len dormant name_assign_type speed
broadcast duplex netdev_group statistics
carrier flags operstate subsystem
carrier_changes gro_flush_timeout phys_port_id tx_queue_len
carrier_down_count ifalias phys_port_name type
carrier_up_count ifindex phys_switch_id uevent
device iflink power
```

6.Setup the CAN interface

Code:

```
sudo ip link set can0 up type can bitrate 500000
```

(NOTE: to down the bus
sudo ip link set can0 down)

7.

Code:

```
sudo ifconfig
```

Code:

```
can0: flags=193<UP,RUNNING,NOARP> mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10 (UNSPEC)
    RX packets 1651057 bytes 717672 (700.8 KiB)
    RX errors 1707 dropped 39 overruns 0 frame 1707
    TX packets 23 bytes 141 (141.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

8. To send and receive data we can use Python for RPi and C for Arduino (In RPi we can use can-utils for testing)

LIBRARIES:

Arduino C CAN : https://github.com/Seeed-Studio/Seeed_Arduino_CAN/tree/old (https://github.com/Seeed-Studio/Seeed_Arduino_CAN/tree/old)

Code:

Install the above library in your arduino IDE

Raspberry Pi Python CAN : <https://python-can.readthedocs.io/en/master/> (<https://python-can.readthedocs.io/en/master/>)

Code:

```
pip3 install python-can
```

1.Run python3 can_basic_send.py in RPi and CAN_RX.ino in Uno

2.Run python3 can_basic_recv.py in RPi and CAN_TX.ino in Uno

VERY IMPORTANT NOTE:

Bitrate of RPI & UNO is 500000(set in program)

Clock frequency of RPI & UNO is 8Mhz(set in config file and program)

Remove unwanted LED stuff from arduino code if you want!!

can_basic_recv.py OUTPUT:

Code:

```
Timestamp: 1607220936.314801      ID: 0043      S      DLC: 8      01 02 03 04 05 06 07 08      Channel: can0
Timestamp: 1607220937.314749      ID: 0043      S      DLC: 8      01 02 03 04 05 06 07 08      Channel: can0
Timestamp: 1607220937.315004      ID: 0043      S      DLC: 8      01 02 03 04 05 06 07 08      Channel: can0
Timestamp: 1607220937.315288      ID: 0043      S      DLC: 8      01 02 03 04 05 06 07 08      Channel: can0
Timestamp: 1607220938.315610      ID: 0043      S      DLC: 8      01 02 03 04 05 06 07 08      Channel: can0
```

CAN_RX.ino OUTPUT:

Code:

```
Data from ID: 0x7EE
0      1      3      1      4      1
-----
Data from ID: 0x7EE
0      1      3      1      4      1
-----
Data from ID: 0x7EE
0      1      3      1      4      1
-----
Data from ID: 0x7EE
0      1      3      1      4      1
-----
Data from ID: 0x7EE
0      1      3      1      4      1
```

In Arduino Run

CAN_RX.ino

Code:

```
#include <SPI.h>
#include "mcp_can.h"

const int spiCSPin = 10;
const int ledPin = 2;
boolean ledON = 1;

MCP_CAN CAN(spiCSPin);

void setup()
{
    Serial.begin(115200);
    pinMode(ledPin,OUTPUT);

    while (CAN_OK != CAN.begin(CAN_500KBPS,MCP_8MHz))
    {
        Serial.println("CAN BUS Init Failed");
    }
}
```

```

        delay(100);
    }
    Serial.println("CAN BUS  Init OK!");
}

void loop()
{
    unsigned char len = 0;
    unsigned char buf[8];

    if(CAN_MSGAVAIL == CAN.checkReceive())
    {
        CAN.readMsgBuf(&len, buf);

        unsigned long canId = CAN.getCanId();

        Serial.println("-----");
        Serial.print("Data from ID: 0x");
        Serial.println(canId, HEX);

        for(int i = 0; i<len; i++)
        {
            Serial.print(buf[i]);
            Serial.print("\t");
            if(ledON && i==0)
            {
                digitalWrite(ledPin, buf[i]);
                ledON = 0;
                delay(500);
            }
            else if(!(ledON) && i==4)
            {
                digitalWrite(ledPin, buf[i]);
                ledON = 1;
            }
        }
        Serial.println();
    }
}

```

In Raspberry Pi Run

can-basic-send.py

Code:

```

import time
import can

bustype = 'socketcan'
channel = 'can0'
bus = can.interface.Bus(channel=channel, bustype=bustype, bitrate=500000)

msg = can.Message(arbitration_id=0xc0ffee, data=[0, 1, 3, 1, 4, 1], is_extended_id=False)
while True:
    bus.send(msg)
    time.sleep(1)

```

In Arduino Run

CAN_TX.ino

Code:

```
#include <SPI.h>
#include <mcp_can.h>

const int spiCSPin = 10;
int ledHIGH    = 1;
int ledLOW     = 0;

MCP_CAN CAN(spiCSPin);

void setup()
{
    Serial.begin(115200);

    while (CAN_OK != CAN.begin(CAN_500KBPS,MCP_8MHz))
    {
        Serial.println("CAN BUS init Failed");
        delay(100);
    }
    Serial.println("CAN BUS Shield Init OK!");
}

unsigned char stmp[8] = {1, 2, 3, 4, 5, 6, 7, 8};

void loop()
{
    Serial.println("In loop");
    CAN.sendMsgBuf(0x43, 0, 8, stmp);
    delay(1000);
}
```

In Raspberry Pi Run
can-basic-recv.py

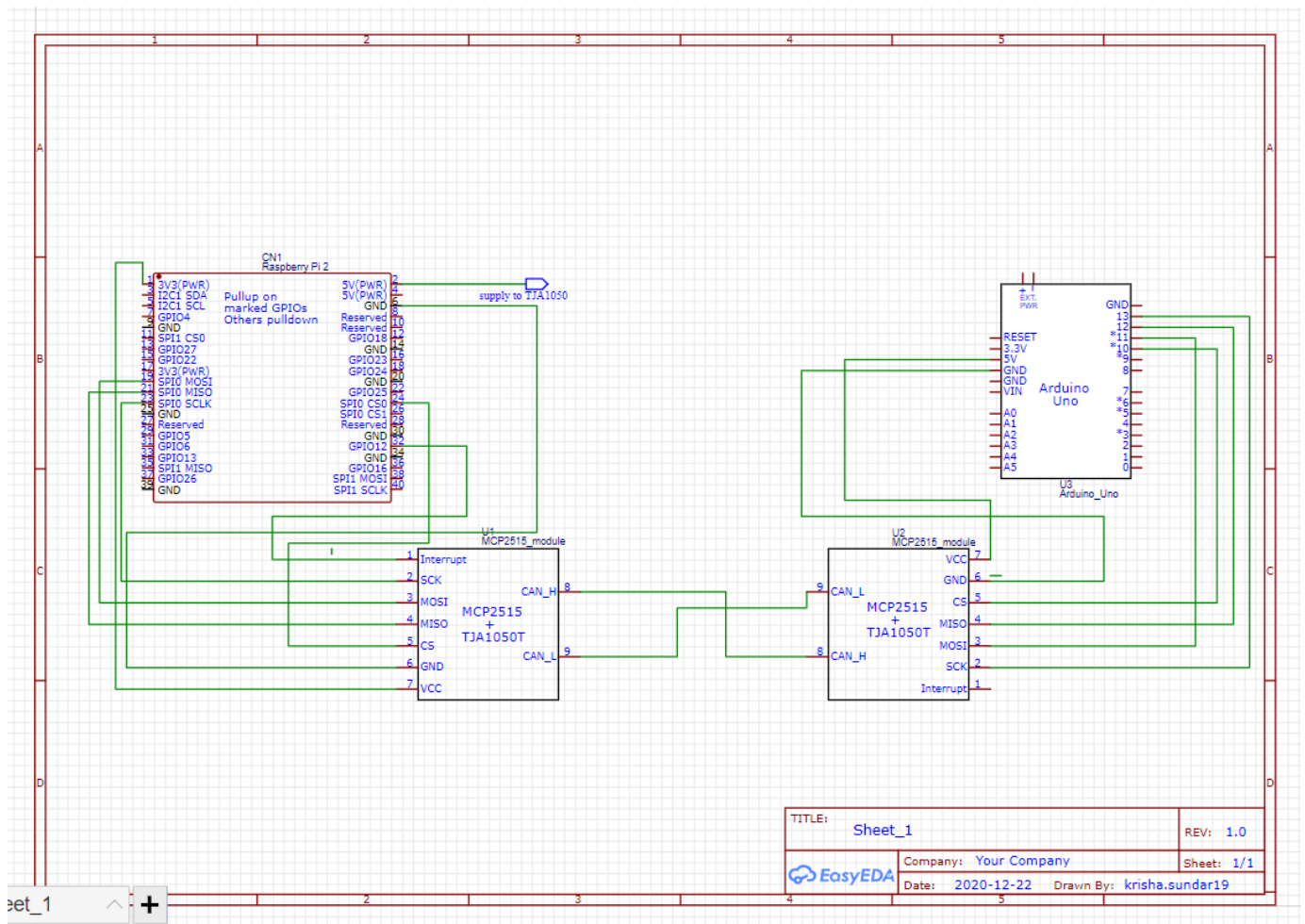
Code:

```
import can
import time

can_interface = 'can0'
bus = can.interface.Bus(can_interface, bustype='socketcan',bitrate=500000)

while True:
    message = bus.recv()
    print(message)
#for msg in bus:
    #print(msg.data)
```

Attachments



pi_uno_can.png (132.05 KiB) Viewed 25474 times

Locked

- [Print view](#)

1 post • Page 1 of 1

[Return to "Interfacing \(DSI, CSI, I2C, etc.\)"](#)

Jump to


- [Community](#)
- [General discussion](#)
- [Announcements](#)
- [Other languages](#)
- [Deutsch](#)
- [Español](#)
- [Français](#)
- [Italiano](#)
- [Nederlands](#)
- [日本語](#)
- [Polski](#)
- [Português](#)
- [Русский](#)
- [Türkçe](#)
- [User groups and events](#)
- [The MagPi](#)
- [Using the Raspberry Pi](#)
- [Beginners](#)
- [Troubleshooting](#)
- [Advanced users](#)
- [Assistive technology and accessibility](#)

- Education
 - Picademy
 - Teaching and learning resources
 - Staffroom, classroom and projects
 - Astro Pi
 - Mathematica
 - High Altitude Balloon
 - Weather station
- Programming
 - C/C++
 - Java
 - Python
 - Scratch
 - Other programming languages
 - Windows 10 for IoT
 - Wolfram Language
 - Bare metal, Assembly language
 - Graphics programming
 - OpenGL
 - OpenVG
 - OpenMAX
 - General programming discussion
- Projects
 - Networking and servers
 - Automation, sensing and robotics
 - Graphics, sound and multimedia
 - Other projects
 - Media centres
 - Gaming
 - AIY Projects
- Hardware and peripherals
 - Camera board
 - Compute Module
 - Official Display
 - HATs and other add-ons
 - Device Tree
 - Interfacing (DSI, CSI, I2C, etc.)
 - Raspberry Pi 400 and 500
 - Raspberry Pi Pico
 - General
 - SDK
 - MicroPython
 - Other RP2040 boards
 - AI Accelerator
 - AI Camera - IMX500
 - Hailo
- Software
 - Raspberry Pi OS
 - Raspberry Pi Connect
 - Raspberry Pi Desktop for PC and Mac
 - Other
 - Android
 - Debian
 - FreeBSD
 - Gentoo
 - Linux Kernel
 - NetBSD
 - openSUSE
 - Plan 9
 - Puppy
 - Arch
 - Pidora / Fedora
 - RISCOS
 - Ubuntu

- [Ye Olde Pi Shoppe](#)
- [For sale](#)
- [Wanted](#)
- [Off topic](#)
- [Off topic discussion](#)

- [Board index](#)
- All times are UTC
- [Delete cookies](#)


Follow us

 Sign up to newsletter

 X

 Threads

 TikTok

 YouTube

 Instagram

 LinkedIn

 Facebook

About Raspberry Pi

[News](#)

[Investor relations](#)

[Contact us](#)

[Trademark](#)

[About us](#)

[Our Approved Resellers](#)

[Jobs](#)

[Accessibility](#)

[Modern slavery statement](#)

[Site use terms and conditions](#)

[Acceptable use](#)

[Cookies](#)

[Licensing](#)

[Terms and conditions of sale](#)

[Privacy](#)

[Security](#)

[Verify our bank details](#)

For home

[Raspberry Pi for home](#)

[Tutorials](#)

For industry

[Raspberry Pi for industry](#)

[Thin clients](#)

[Raspberry Pi in space](#)

[Powered by Raspberry Pi](#)

[Design partners](#)

[Success stories](#)

Hardware

[Computers and microcontrollers](#)

[Cameras and displays](#)

Software

[Raspberry Pi Connect](#)

[Raspberry Pi Desktop](#)

[Raspberry Pi Imager](#)

[Raspberry Pi OS](#)

Documentation

[All categories](#)

[Product information portal](#)

[Datasheets](#)

Community

[Forums](#)

[Events](#)

Add-on boards

Power supplies and cables

Cases

Peripherals

Raspberry Pi Store

Store information

Raspberry Pi Press

About Raspberry Pi Press

The MagPi

HackSpace