

[Home \(/\)](#) Circuits (/circuits/) Workshop (/workshop/) Craft (/craft/) Cooking (/cooking/) Living (/living/)

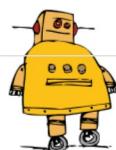
Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/)

[Log In \(/account/login/\)](#) | [Sign Up](#)[\(/account/register/\)](#)

Outside (/outside/) Teachers (/teachers/)

[Download](#)

Favorite I Made It

**instructables** (/circuits/)
circuits[Projects \(/circuits/projects/\)](#)[Contests \(/contest/\)](#)[PUBLISH \(/CREATE/\)](#)[Let's Make...](#)

Yes We CAN BUS With Arduino in 30 Seconds!

By omarCartera (/member/omarCartera/) in Circuits (/circuits/) > Arduino (/circuits/arduino/projects/)

106,094

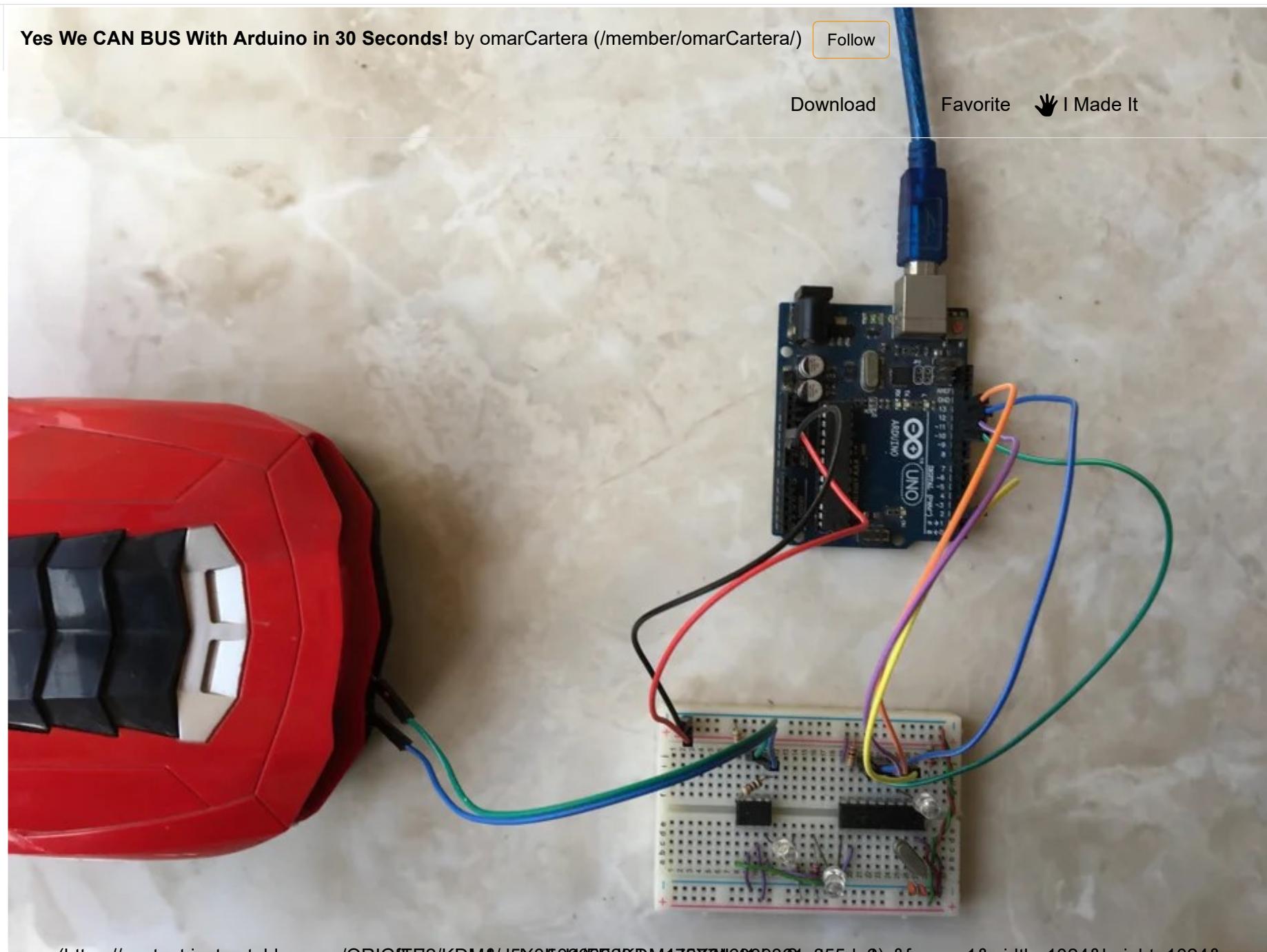
110

25

Featured

[Download](#)

Favorite

Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/)[Follow](#)[Download](#)[Favorite](#) https://content.instructables.com/0D10/SEED0KHM9LGMV0PNE8RDM17VTHICM0D91H95E/foto_9frames-1.9.width-1024.height-1024

Yes We CAN BUS With Arduino in 30 Seconds!



By **omarCartera** (/member/[omarCartera](#))

[Follow](#)

[Follow](#)

About: We make because why not! More About [omarCartera](#) » (/member/[omarCartera](#))
[Download](#) [Favorite](#)

Hello Arduinos!

This Instructable is trying to summarize what I ended up with after a long time of search, tutorials, trials and datasheets to build a functional **CAN BUS** node. I will try to keep it as easy and concise as possible to get you straight to a properly working setup, saving your time for further developments afterwards.

The first 3 steps are the basic ones that will initiate a **CAN BUS** at your home, the rest of the steps are a little bit advanced and real-life **CAN** situations.

CAN BUS is a two-wire, half-duplex communication protocol that is widely used in Automotive industry. One of its greatest advantages is that it connects any number of ECUs (or microcontrollers) in your car through the two-wire bus, **CAN High** and **CAN Low**, reducing the weight of wires that could be gained by using point-to-point communication between ECUs.

Enough talking and let's grease our hands!

You can continue reading about CAN from [wiki \(https://en.wikipedia.org/wiki/CAN_bus\)](https://en.wikipedia.org/wiki/CAN_bus) as it really gives a very good and sufficient introduction to the topic.

Yes We CAN BUS With Arduino in 30 Seconds!

Add Tip

Ask Question

Comment

Follow

Download

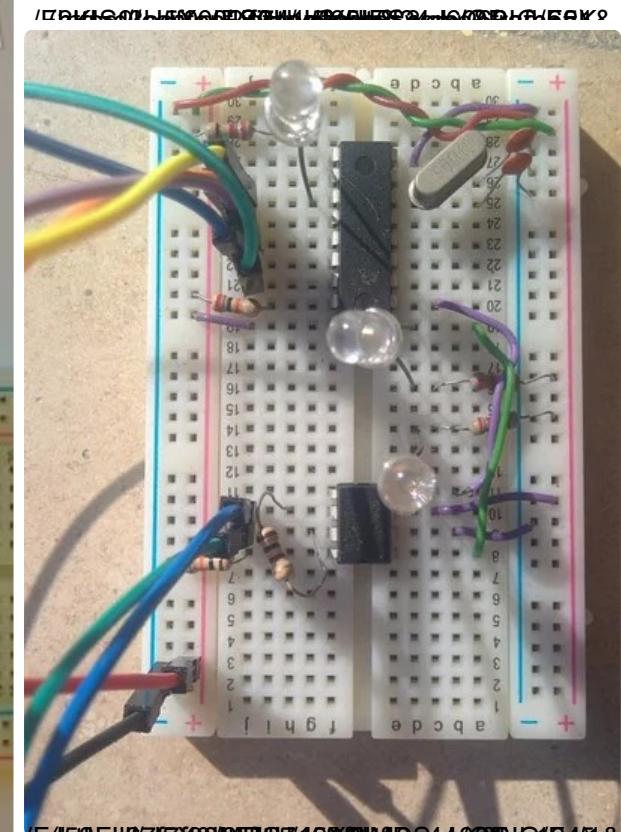
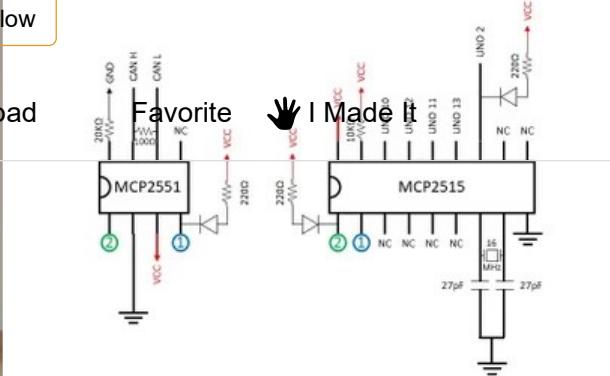
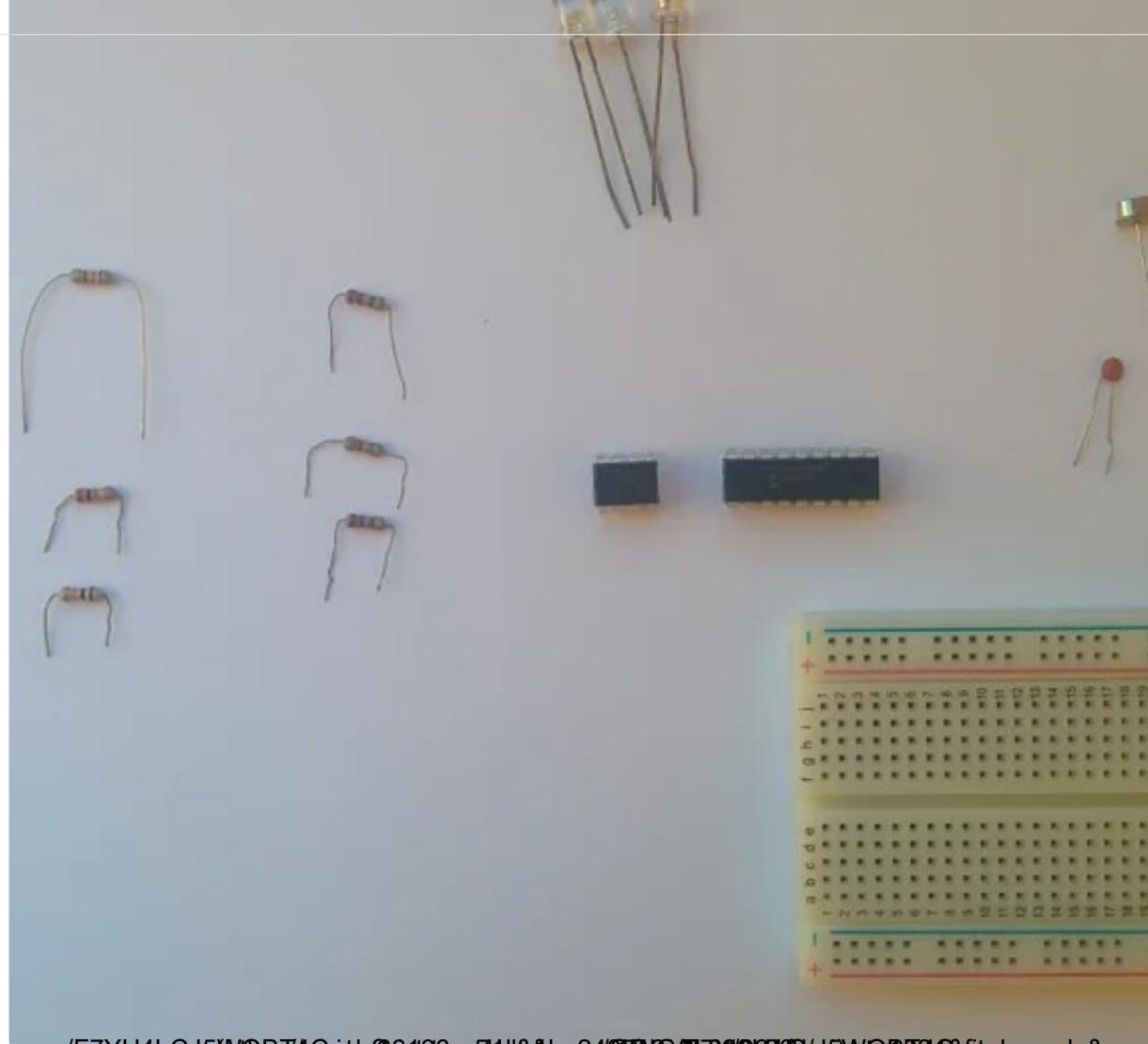
Download

Favorite

 I Made It

Step 1: Build the Hardware

Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/)

[Follow](#)[Download](#)

We could buy any of the plug-and-play Arduino **CAN** shields, but building the hardware ourselves is easy, more fun and cost reduction, bro.

What you need to build one node:

Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/)

[Follow](#)

- x1 Breadboard.
- Download
- Favorite
-  I Made It
- x1 [MCP2515](http://ww1.microchip.com/downloads/en/DeviceDoc/21667f.pdf) (<http://ww1.microchip.com/downloads/en/DeviceDoc/21667f.pdf>) Microchip CAN Controller.
- x1 [MCP2551](http://ww1.microchip.com/downloads/en/DeviceDoc/21801e.pdf) (<http://ww1.microchip.com/downloads/en/DeviceDoc/21801e.pdf>) Microchip CAN Transceiver.
- x1 20KΩ Resistor.
- x1 10KΩ Resistor.
- x1 100Ω Resistor.
- x1 16 MHz Crystal Oscillator.
- x2 27 pF Capacitors.
- x3 LEDs.
- x3 220Ω Resistors.

Schematic key:

CAN H: CAN High and is connected to the CAN High wire of the bus.

CAN L: CAN Low and is connected to the CAN Low wire of the bus.

VCC: 5V power source from the Arduino.

GND: connected to Arduino's ground pin.

UNO X: as X is an integer means connect this pin to Arduino's digital pin X.
Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera ([member/omarCartera/](#)) [Follow](#)

Here, our setup is using 3 means of communication protocols

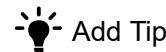
[Download](#)[Favorite](#)  [I Made It](#)

1) UART: to talk to your computer's Serial monitor.

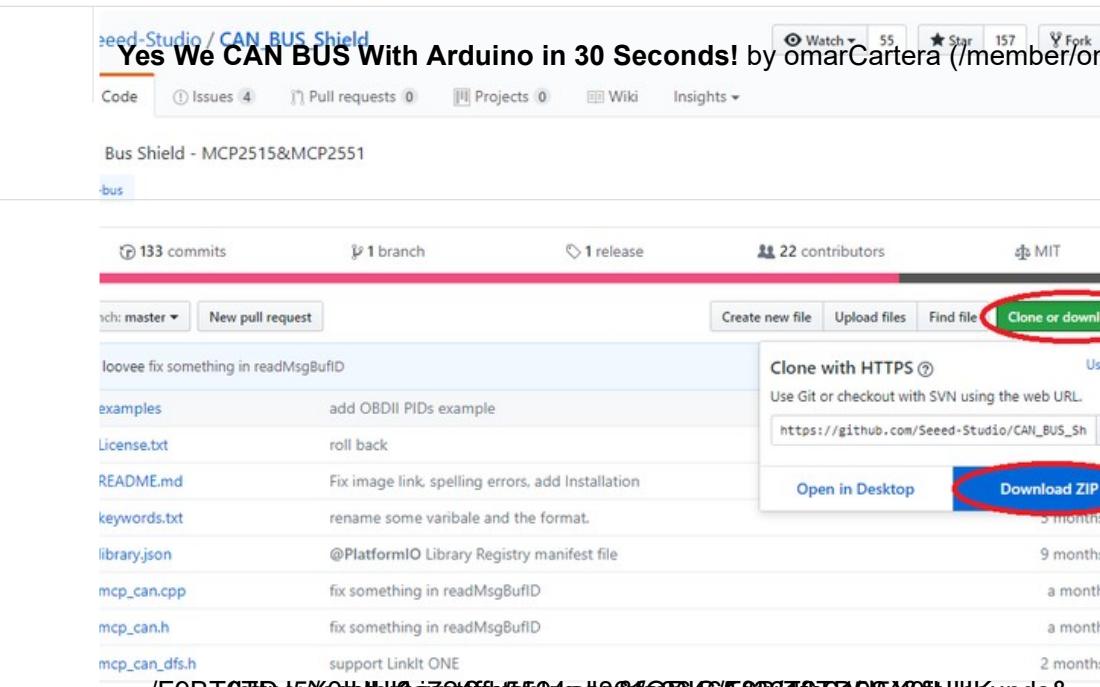
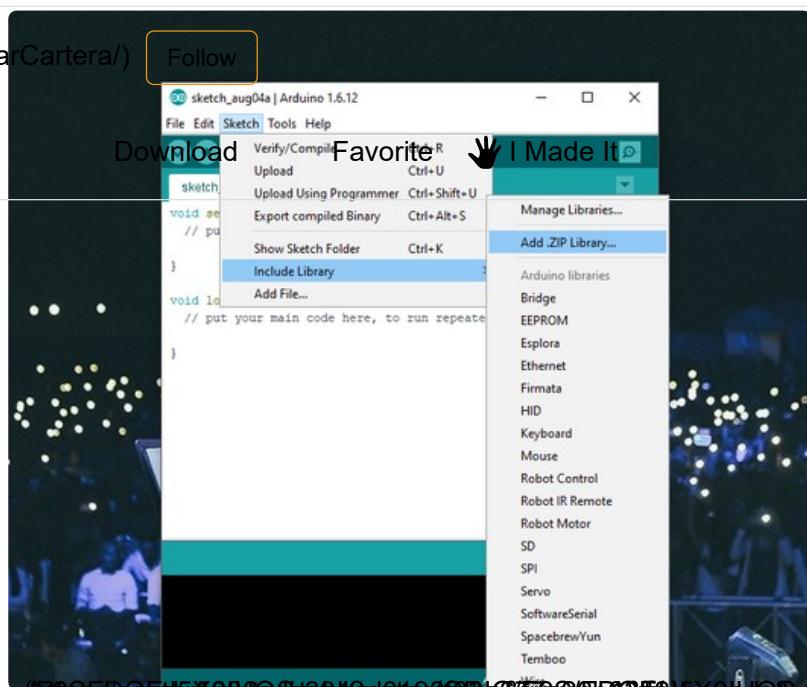
2) SPI: to talk to the **CAN** controller.

3) CAN: to talk to other neighbours in the bus.

Wire up your components to the board according to the schematics attached above and let's move to the next step.

[Add Tip](#)[Ask Question](#)[Comment](#)[Download](#)

Step 2: Download and Install the CAN Library

The image shows two screenshots. The left screenshot is a GitHub repository page for 'Seeed Studio / CAN BUS Shield'. It displays basic repository statistics: 133 commits, 1 branch, 1 release, and 22 contributors. Below these stats are several file listing, including 'examples', 'LICENSE.txt', 'README.md', and various C/C++ source files like 'library.json', 'mcp_can.cpp', 'mcp_can.h', and 'mcp_can_dfs.h'. At the top of the page, there's a prominent green 'Clone or download' button, which is circled in red. Below this, there's another button labeled 'Download ZIP', also circled in red. The right screenshot is a close-up of the Arduino IDE's 'Sketch' menu. The 'Add ZIP Library...' option is highlighted with a red circle. A hand icon is placed over the 'I Made It' button in the top right corner of the IDE window.

In this Instructable, I'm using Seeed Studio open-source [CAN Library \(\[https://github.com/Seeed-Studio/CAN_BUS_Shield\]\(https://github.com/Seeed-Studio/CAN_BUS_Shield\)\)](https://github.com/Seeed-Studio/CAN_BUS_Shield) which you can download from their github as shown above.

Keep the downloaded file as zipped as it is (because Arduino likes this) and add the library to Arduino as shown above, as well.

By this point you are ready to move to the next step to combine both software and hardware and try your Hello CAN examples :D

I'm currently using the latest Arduino (1.8.3) but it works with the old versions as well.



Add Tip



Ask Question



Comment

Download

Step 3: Upload Your First Code

Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/) [Follow](#)

[Download](#)[Favorite](#)[I Made It](#)

The image shows two side-by-side Arduino IDE windows. The left window is titled 'CAN_Send' and the right window is titled 'CAN_Receive'. Both windows have a dark theme. The code in both windows is identical, demonstrating basic CAN bus communication setup.

```
#include <SPI.h>           //SPI is used to talk to the CAN Controller
#include <mcp_can.h>

MCP_CAN CAN(10);           //set SPI Chip Select to pin 10

void setup()
{
    Serial.begin(115200);   //to communicate with Serial monitor

    //tries to initialize, if failed --> it will loop here for ever
    START_INIT:

    if(CAN_OK == CAN.begin(CAN_500Kbps))      //setting CAN baud rate
    {
        Serial.println("CAN BUS Shield init ok!");
    }
    else
    {
        Serial.println("CAN BUS Shield init failed!");
    }
}
```

The bottom status bar of both windows indicates 'Arduino/Genuino Uno on COM3'.

The codes attached are basically the examples of the library itself but with a gentle touch of simplicity. I think that both files are over-commented, but feel free to ask about any bit of code you find unclear, because when you solid understand this basic example, you can follow up with the next few steps and even dig deeper and tweak the codes as you like.

The file called **Send** packs 8 characters (8 bytes) into a message of the ID 0xF1, because I'm an F1 fan, and puts it on the bus.

[Download](#)[Favorite](#)

The file called **Receive** keeps polling the **CAN** receive buffers until any message comes in. It then breaks the incoming data into an ID, data length and the data itself.

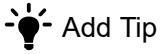
Yes, You CAN BUS now!



CAN_Send.ino (<http://instructables.com/GD6/DR/G/KNY/VEKOM/BA0NJVMK/NYJ3XOMJ5XMM/M.ino>)



CAN_Receive.ino (<http://instructables.com/GR1/G7F4Z/0ZJ5X05X0M/JFW74ZFQZ5X08X0M/JN/ino>)

[Add Tip](#)[Ask Question](#)[Comment](#)[Download](#)

Step 4: CAN Is a Message-Based Protocol!

CAN is a message-based protocol, which means that the messages and their content are more important than the sender ECU itself. So, the ECUs aren't given IDs, but each message has a unique ID in a specific bus. These IDs are responsible for setting the priority of messages in case of two or more ECUs are trying to put their messages on the bus. The LOWER decimal value ID has HIGHER priority.

Given that, the message with ID = 0x05 has more priority than our beloved message of the previous example with ID = 0xF1.

A Big Example:

Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/)

[Follow](#)

If we consider ourselves in a real car, we might assume that the message that informs about a fatal problem in the engine will be given the highest ever priority (Logical, no?). So, whatever ECU tries to send this message will win the bus and continue sending its message while everyone else is just listening until it finishes.

At any time, every **CAN BUS** node sees the message being sent through the bus. But not all of them read it and send it to their ECUs. That's because in our example the rear wing ECU or the front-right headlights ECU don't care at all about a problem in the engine, so they see the message that contains engine failure and ignore it. On the other hand, they are the only ECUs who read the messages like: retract the wing or shut the headlights.

And here it comes the idea of **Message Filtering** that lets an ECU reads only the messages useful to it and ignore everything else. And since our code allows us to know the ID of the message, we can easily apply filtering.

The example file attached here adds only one *if statement* to the basic **Receive** file to read only the messages with ID = **0xF2**. Let this new code receive from the basic **Send** code and it will print nothing.



CAN_Filter.Dro (<https://content.instructables.com/SOIG/05TQ7VSPRBMNISORPSXORMJN/0N.ino>)

-💡- Add Tip

❓ Ask Question

💬 Comment

Download

Step 5: Extract Useful Signals From a CAN Message

Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/)

[Follow](#)

Message ID 0xF1

[Download](#)Favorite  I Made It

7	6	5	4	3	2	1	0	<i>bits bytes</i>
1	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1	1
1	1	1	0	1	1	1	0	2
								3
								4
								5
								6
								7

	Speed
	RPM
	Check Engine Light
	Oil Indicator
	Not used

https://content.instructables.com/0D1C/6KHWKA0RMAQ5QNTINFACT0D0H20N0YD1M0N7_instructable_w.htm?fr=1&fit=border

By reaching this line now we:

Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/)

[Follow](#)

1. Wired up the hardware circuit.

[Download](#)[Favorite](#) I Made It

2. Sent and received CAN messages.

3. Filtered the stream of messages and read only those that interest us.

But all we were exchanging was a message containing useless 8 characters forming my nickname, ahem.

What if we wanted to exchange informative signals and simulate what might be happening in a real **CAN BUS** from the dashboard ECU's point of view!

Because it would be waste of time, and bits actually, if you use the 8 data bytes to represent only one piece of information, cars might pack many signals in one message, *as shown in the picture above*, telling you that for the message of the ID = 0xF1:

- The first whole byte represents the speed of the car with a maximum value of 255 mph.
- The next 14 bits represent the engine revs up to 16383 RPM. Just to be able to work with an F1 car.
- The next bit tells the dashboard to turn On/Off the check engine indicator.
- The next bit tells whether the oil level is under a threshold or not to turn ON/OFF its indicator.
- The rest of the bits are not used in our made-up message.

Check the attached files for the code, and come back here for the explanation.

Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/)[Follow](#)

For the first piece of information, as it fits in one byte and our ~~code~~ let us deal with ~~every~~ byte of the buffer separately, all you need to do here is to extract the car speed directly from the first byte only, easy!

- i.e. `car_speed = buf[0];`

For the rest cases where the signal bits take more or less than a byte, you will need *bit manipulation* to put those bits in the right setup before reading them.

To read the RPM which lies in two different bytes (1 and 2) you need to do as follows:

- First, read the higher byte (byte 1) into a two-byte integer, it will lay in the lower byte in `engine_rpm`.
 - `unsigned int engine_rpm = buf[1];`
- Then, shift this value 8 bits to the left to put the higher byte you read in its right position.
 - `engine_rpm = engine_rpm << 8;`

Yes We CAN BUS With Arduino in 30 Seconds!

by [SmartCartera](#) (member for 1 year)

[Follow](#)

- `char temp = buf[2] & 0x3F;`

[Download](#)[Favorite](#)

- Here, we just need to add the higher byte to the lower one and get our final value.
 - `engine_rpm = engine_rpm + temp;`

To read the value of **Check Engine** and **Oil** bits, you will need to mask off all the bits but the one you want, piece of cake!

Tataaaaa! We correctly extracted all the information embedded in the message!



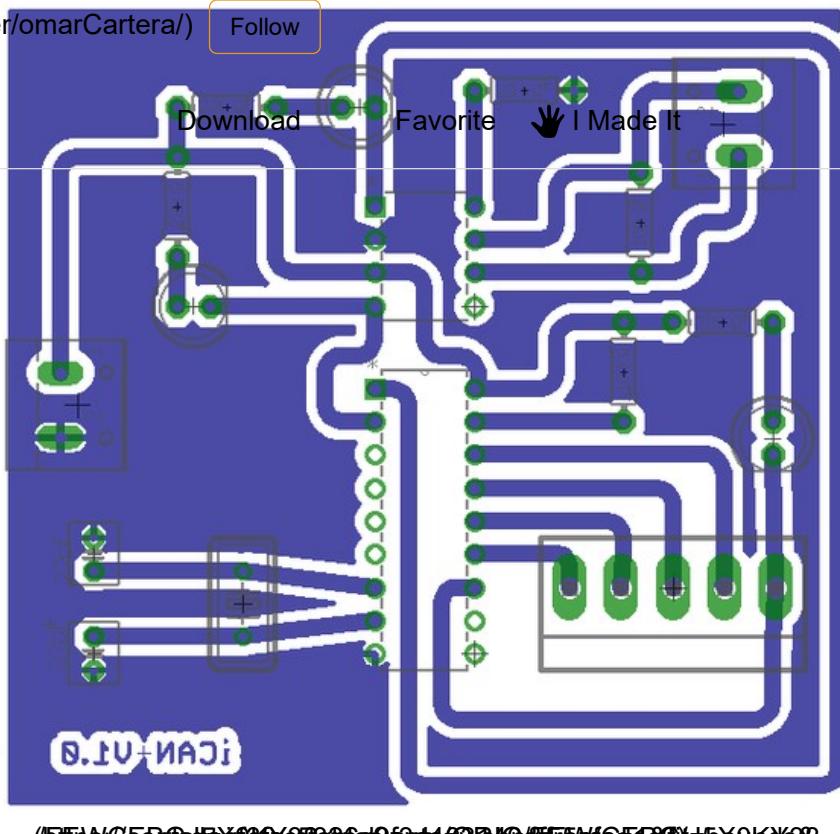
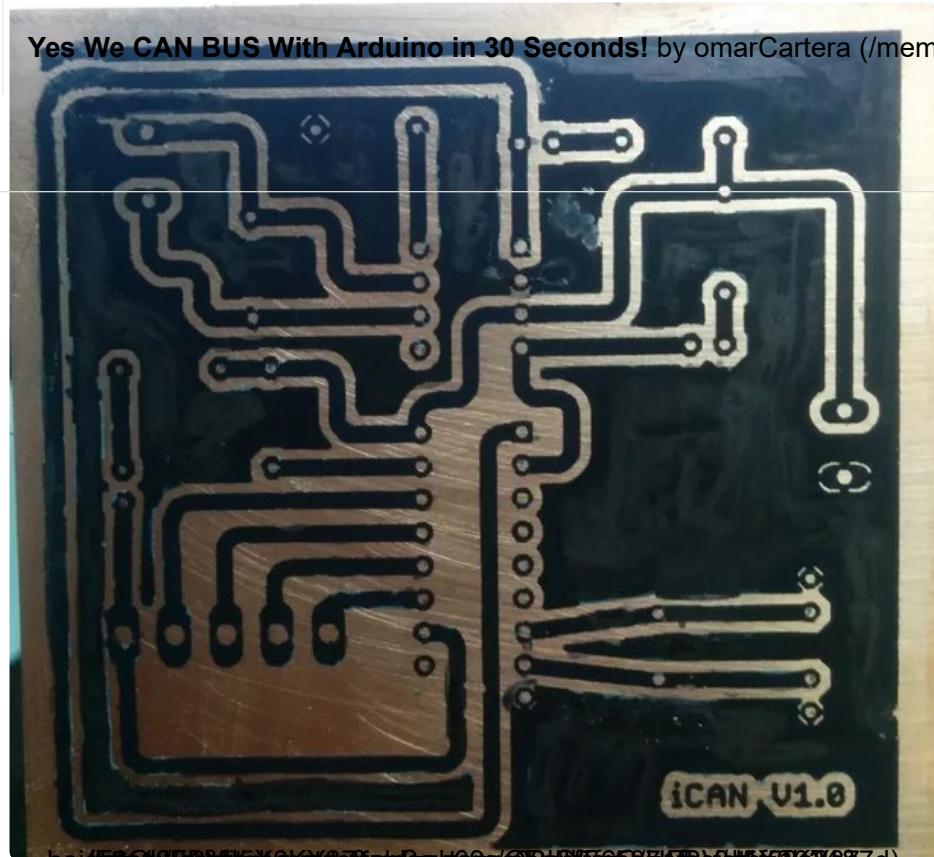
Signal_Reading.ino (<https://content.instructables.com/GR1/G/R13/M1QJ7d5X015XGKX4/W64P/55P015XGKXG0X.ino>)



Signal_Writing.ino (<https://content.instructables.com/GR1/G/R13/M1QJ7d5X015XGKX4/W64P/55P015XGKXG0X.ino>)

[Add Tip](#)[Ask Question](#)[Comment](#)[Download](#)

Step 6: The Evolution of Breadboard to PCB



After testing and making sure that everything is working properly, I decided to start learning PCBs making by converting this very project to a PCB.

Attached here are the **Eagle** files for the layout. The layout was designed, printed on a copper board but not yet milled or tested, you can revise the design and give your comments below, you can also print it and make your own PCB and then tell us what you ended up with. And you can even suggest enhancements to the layout design!



iCAN V1.0.rar (<https://www.instructables.com/6B1C6FBX0RXGKF5XOKZ3V5BXOKZKV5x0KZ3V.rar>)

Yes We CAN BUS With Arduino in 30 Seconds!

Add Tip

Ask Question

Comment

Follow

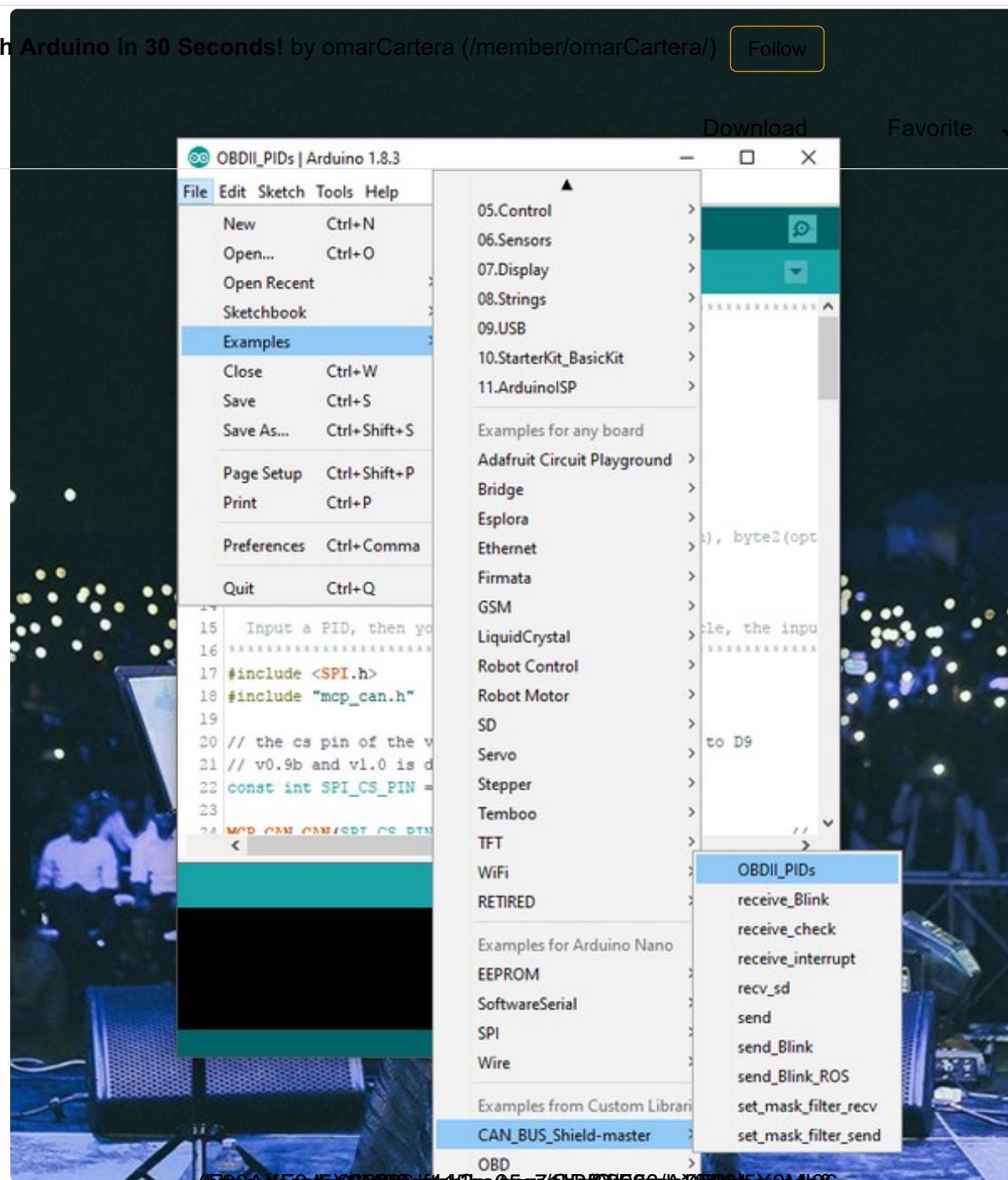
Download

Download

Favorite

 I Made It

Step 7: Have a Little Chat With Your Car

Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/)[Follow](#)[Favorite](#) [I Made It](#)

Have you ever been to your mechanic complaining from a mysterious light in your dashboard

Yes We CAN BUS With Arduino in 30 Seconds! by omar Cartera (Member of omar Cartera) Follow

and you see him plugging some device to your car ~magic happens~ that shows you a fault code describing in details what the non-mysterious light means?!

[Download](#)[Favorite](#) I Made It

Some say, with our **CAN** setup, we can talk to our cars through their OBD-II (https://en.wikipedia.org/wiki/On-board_diagnostics#OBD-II) port, just like the mechanic in the previous paragraph, by sending the so called PIDs (https://en.wikipedia.org/wiki/OBD-II_PIDs) to the OBD requesting some parameter from the car communication buses and wait for the response message carrying the values you asked for.

An example covering this PID part is given in the examples section of the library, and the list of available PIDs is available on wiki.

I didn't try it, but I will do this very soon. **Take care of your connections and read about OBD before you plug anything to your car.**

Feel free to discuss anything with us below, we are all here to learn from each other, Peace!

 Add Tip Ask Question Comment[Download](#)

**Yes We CAN BUS With Arduino in 30 Seconds!** by omarCartera (/member/omarCartera/)

Participated in the

[Follow](#)First Time Author Contest (/contest/fta2017/)[Download](#)[Favorite](#)[View Contest](#)
 I Made It

Be the First to Share

Did you make this project? Share it with us!

[I Made It!](#)

Recommendations

[\(/DIY-Solar-Bottle-Lamp/\)](#)**DIY Solar Bottle Lamp** (/DIY-Solar-Bottle-Lamp/) by opengreenenergy (/member/opengreenenergy/) in LEDs (/circuits)[\(/Automated-Chessboard/\)](#)**Automated Chessboard** (/Automated-Chessboard/) by Greg06 (/member/Greg06/) in Arduino (/circuits/arduino/projects/)

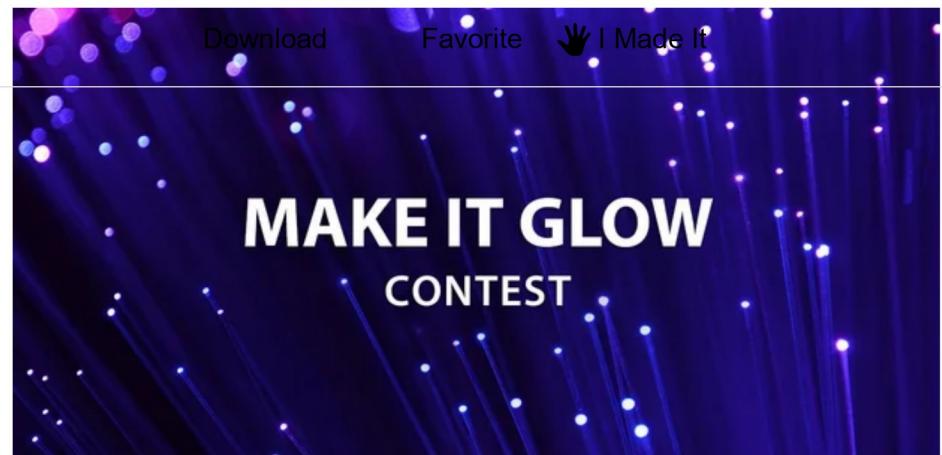
Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/)

304 31K

Follow



(/contest/homeimprovement/)



(/contest/makeitglow2022/)



Add Tip



Ask Question



Post Comment

We have a **be nice** policy.
Please be positive and constructive.

Add Images

Post

25 Comments



(/member/ilkierro/) ilkierro (/member/ilkierro/) 7 months ago

Reply

▲ Upvote

Hello,
Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/)
I did not try the project yet, but I will very soon. I have one question, how did you pick the capacitor
values, when I check the MCP2515 Datasheet, 16Mhz Crystal Oscillator is not listed, there are 4, 8, 20
Mhz with 27, 22, 15 pF capacitors.
Download Favorite 

Thanks



(/member/william.siffer/) william.siffer (/member/william.siffer/) Question 1 year ago

[Answer](#)

 Upvote

Hi- I made this but I cannot seem to get any data from my OBD2 port. I am connected to 6/14 and 2 of the led's are blinking. I am also connected to CAN ground. The problem is it is not getting any data with the car on, instead it only shows symbols like arrows and boxes and question marks. Any ideas?

6 answers ▾



(/member/DrewYu/) DrewYu (/member/DrewYu/) 1 year ago

[Reply](#)

 Upvote

Hi there, excuse my ignorance but I was just wondering what you have the Arduino plugged into in your first picture? It is some sort of small RC car with a CAN Bus? I'm looking for some sort of HW Simulator for CAN Bus and that would be awesome!

1 reply ▾



(/member/steelhanzero/) steelhanzero (/member/steelhanzero/) Question 1 year ago on Step 7

[Answer](#)

 Upvote

hello~

I have got a question.

in your source code use a 1 ID of buffer..
Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/)
If i want to use a 3 id of buffer data..

[Follow](#)

what can i change the code?

[Download](#)[Favorite](#)  [I Made It](#)

like

```
if(canID=0x01) {.a=buf[1]....}  
if(canID=0x02) { b=buf[1]...}  
if(canID=0x03) {c=buf[1]...}
```

is it ok?

i tried but it may had a buffer problems..

2 answers ▾



(/member/Abd2490/) Abd2490 (/member/Abd2490/) Question 2 years ago on Step 3

[Answer](#)[▲ Upvote](#)

hi,

can someone explain what is sending CAN messages from arduino, we actually need to extract the messages from the CAN bus if I have understood correctly, i am a beginner kindly assist ?

1 answer ▾



(/member/JSobell/) JSobell (/member/JSobell/) 3 years ago

[Reply](#)[▲ Upvote](#)

I think it either has to be

`engine_rpm = ((buf[2] & 0x3F) << 8) | buf[1]`

or

`engine_rpm = (buf[1] << 6) | (buf[2] & 0x3F)`
Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/) [Follow](#)

The example given is adding 0-63 to a number that is a multiple of 256, so assuming buf[1] is the low bits, the first applies. If the lowest bits are in buf[2] then the second applies.

[Download](#)[Favorite](#)[I Made It](#)

Regardless of that minor issue, it's a nice example! Good job!

4 replies ▾

[More Comments](#)[Post Comment](#)

Yes We CAN BUS With Arduino in 30 Seconds! by omarCartera (/member/omarCartera/)**Categories**[Follow](#)**About Us****Resources**
 Circuits
(/circuits/)
 Workshop
(/workshop/)
 Craft
(/craft/)
 Cooking
(/cooking/)
 Living
(/living/)
 Outside
(/outside/)
 Teachers
(/teachers/)
[Download](#)
We Are Favorite
(/about/)
[Why Publish?](#)
(/create/)
 I Made It
Sitemap
(/sitemap/)
[Help](#) (/how-to-write-a-great-instructable/)
[Contact](#) (/contact/)**Find Us**
[\(https://www.instagram.com/instructables/\)](https://www.instagram.com/instructables/)
[\(https://www.pinterest.com/instructables/\)](https://www.pinterest.com/instructables/)
[\(https://www.facebook.com/instructables/\)](https://www.facebook.com/instructables/)
[\(https://www.twitter.com/instructables/\)](https://www.twitter.com/instructables/)

© 2022 Autodesk, Inc.

[Terms of Service](#)
 (<http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=21959721>)
 |
 |

[Privacy Statement](#)
 (<http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=21292079>)
 |
 |

[Privacy settings](#) |
[Do not sell my personal information](#)
 (<https://www.autodesk.com/company/legal-notices-trademarks/ccpa-do-not-sell>)
 |

[Legal Notices & Trademarks](#)
 (<http://usa.autodesk.com/legal-notices-trademarks/>)