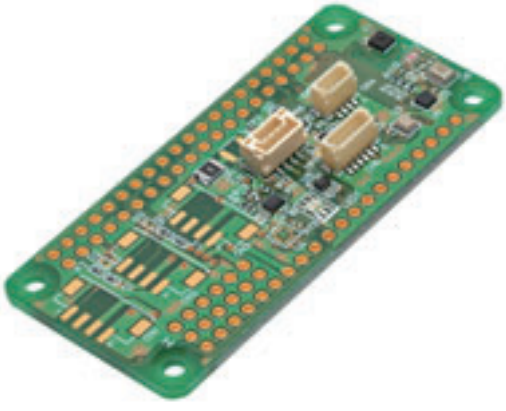
OMRON

Sensor Evaluation Board

2JCIE-EV01-RP1

User's Manual

Sensor Evaluation Board



A296-E1-02

Safety Precautions

Be sure to read the data sheet, and use only if you agree to the contents.

Contents

1. Overview

This manual explains how to use the Raspberry Pi (*1) compatible OMRON Sensor Evaluation Board (2JCIE-EV01-RP1), including special notes and other information. This Sensor Evaluation Board provides 6 sensing functions (temperature, humidity, air pressure, illumination, sound, acceleration). This manual explains how to acquire these sensing data.

To connect a sensor other than the 6 types of sensors provided on the Sensor Evaluation Board, check GitHub(*2). For details on the provided sensor types, connector layout, and other specifications, see the data sheet for this product.

2. Items Required

- ☐ Sensor Evaluation Board (2JCIE-EV01-RP1) x1
- ☐ Raspberry Pi with OS setup completed x1

(*1) Supported models
Raspberry Pi 1 Model B+
Raspberry Pi 2 Model B
Raspberry Pi 3 Model A
Raspberry Pi 3 Model B
Raspberry Pi 3 Model B+
Raspberry Pi Zero
Raspberry Pi Zero W
Raspberry Pi Zero WH

(*2)GitHub URL

MEMS Thermal Sensor	https://github.com/omron-devhub/d6t-2jcieev01-raspberrypi
MEMS Flow Sensor	https://github.com/omron-devhub/d6f-2jcieev01-raspberrypi

The explanations in this manual assume that the Raspberry Pi 3 Model B (with pin header) is being used.

3. Sensor Board Setup

3-1 Checking Sensor Evaluation Board Components

Make sure the following components are included with the Sensor Evaluation Board.

- ☐ Sensor Evaluation Board x1
- ☐ Pin socket x1

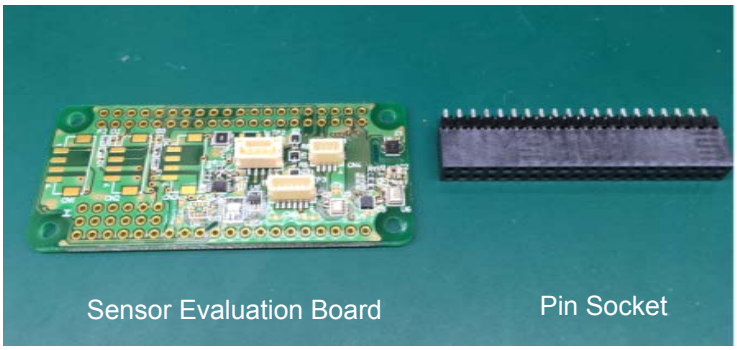
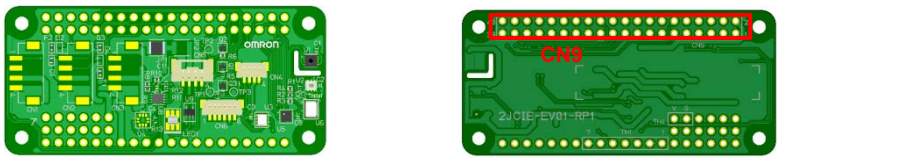


Fig. 1 Sensor Evaluation Board Components

3-2 Soldering

1) From the back of the Sensor Evaluation Board, insert the provided pin socket into the through-hole (CN9).
   Insert the pin socket firmly into the through-hole in the Sensor Evaluation Board so that the pin socket does not shift out of position.



Sensor Evaluation Board / front Sensor Evaluation Board / back

Fig. 2 Appearance of Sensor Evaluation Board

Note: This side with the printed OMRON logo is the front side. If soldering fails and you need to purchase a Pin Socket, purchase a pin socket with two rows of 20 pins at a pitch of 2.54 mm. Recommended pin socket: SAMTEC / ESQ-120-12-L-D

2) Solder all pins from the front side of the Sensor Evaluation Board to secure them. (40 pins)

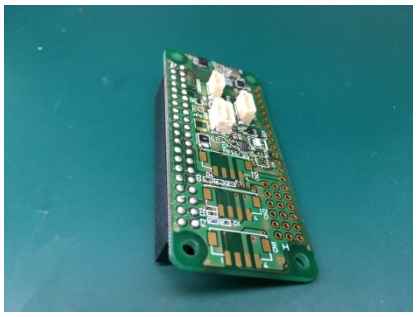


Fig. 3 Pin Socket Soldering

Note: Take care not to burn yourself when soldering. Take care not to inhale smoke when soldering.

3-3 Raspberry Pi Connection

Insert the pin header on the Raspberry Pi into the pin socket on the Sensor Evaluation Board. The Sensor Evaluation Board is on top and Raspberry Pi is on the bottom.

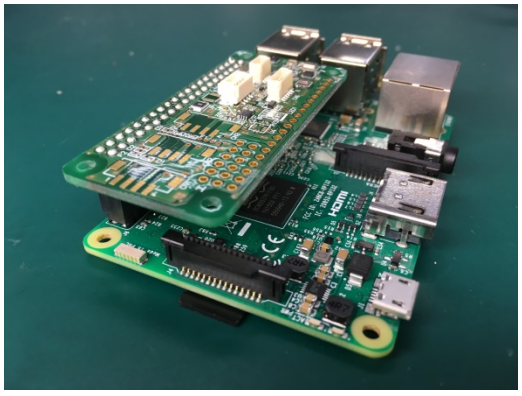


Fig. 4 Raspberry Pi connection

Note: The temperature sensor on the Sensor Evaluation Board may output a high value due to heat from the Raspberry Pi unit. To eliminate this effect, either connect with a cable to separate the two, or use an externally connected temperature/humidity sensor.

4. Enabling Raspberry Pi SPI and I2C Settings

Start Raspberry Pi, open "Settings" > "Raspberry Pi Settings" from the desktop start menu, "Enable" the SPI and I2C items in "Interface", and restart.

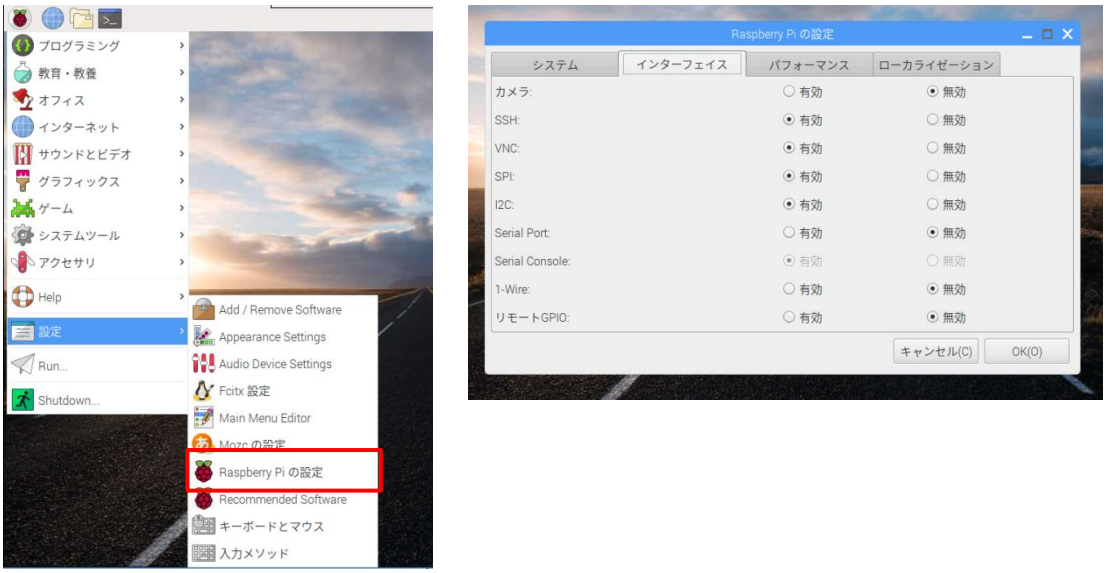


Fig. 5 Enabling SPI and I2C

5. Downloading the Sample Program

Access GitHub at the URL below and download the zip file. Extract the contents and save in any folder.

GitHub URL https://github.com/omron-devhub/2jcieev01-raspberrypi

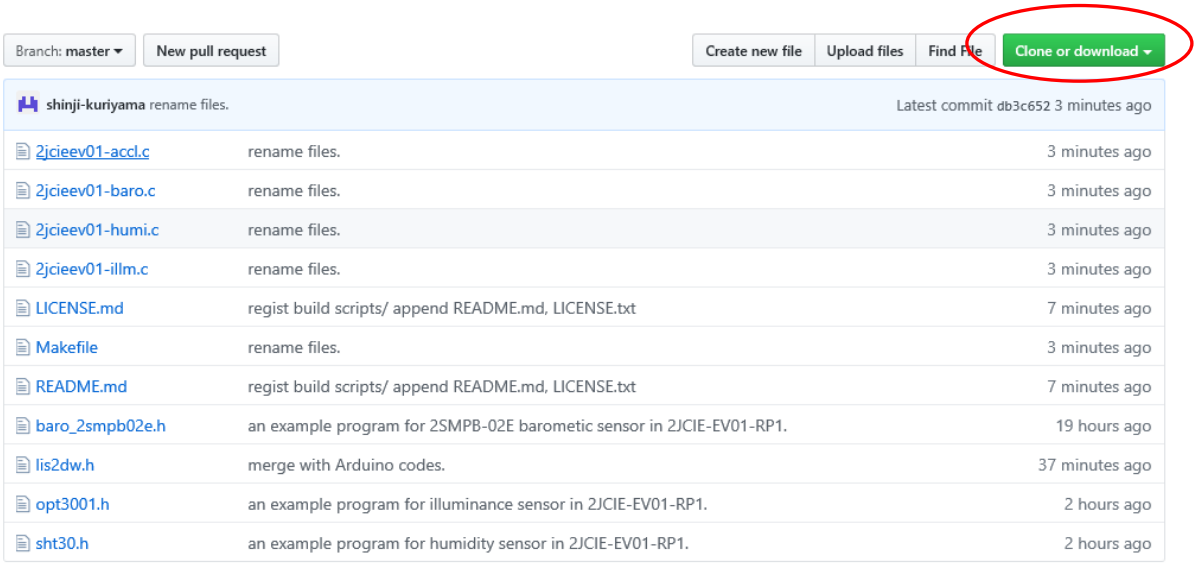


Fig. 6 Downloading the Zip File

Or, open a terminal and execute the following command to save it in the 2jcieev01-raspbertypi folder:

```
~$ git clone https://github.com/omron-devhub/2jcieev01-raspberrypi
```

Note: In some cases it may take time to download the file. The sample source code is only for testing purposes. OMRON does not guarantee its operation. OMRON does not assume responsibility to make changes to the sample source code or correct mistakes or defects in the sample source code for any reason. We cannot accept any inquiries regarding the sample source code.

6. Running the Sample Program (Excluding MEMS Microphone)

Open terminal, move to the folder (directory) where you saved the sample program, and execute the following command.

```
~$ make all
```

MEMS digital barometric pressure sensor

```
~$ ./2jcieev01-baro
```

For repeated execution:

```
~$ while ./2jcieev01-baro; do ./2jcieev01-baro; done
```

Ambient light sensor

```
~$ ./2jcieev01-illm
```

For repeated execution:

```
~$ while ./2jcieev01-illm; do ./2jcieev01-illm; done
```

Temperature / humidity sensor

```
~$ ./2jcieev01-humi
```

For repeated execution:

```
~$ while ./2jcieev01-humi; do ./2jcieev01-humi; done
```

MEMS digital motion sensor

```
~$ ./2jcieev01-accl
```

For repeated execution:

```
~$ while ./2jcieev01-accl; do ./2jcieev01-accl; done
```

7. Reference Running the Sample Program (MEMS Microphone)

After completing the tasks in this section, you will not be able to use the Web Browser.
It may not be available on the package-updated (sudo apt-get upgrade and sudo apt-get dist-upgrade) Raspberry pi.

7-1 I2S Settings

To enable the I2S interface, you must start Terminal and enter commands. The commands to be entered in Terminal are shown below. *~$* represents the prompt that indicates the current directory. Do not include it in a command.

Install asoundlib.h for the microphone.

```
~$ sudo apt-get install libasound2-dev
```

Edit */boot/config*.txt and enable I2S support.

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

Open *config*.txt in *nano* text editor, delete "#" from the beginning of the "#dtparam=i2s=on" line, and save.

This manual uses the *nano* text editor to edit the file. If you are using a different editor, change "nano" to the text editor you are using. For information on how to use *nano*, refer to the website below and other explanations on the Internet.

GNU nano website
https://www.nano-editor.org/

Next, check if sound support is enabled in the kernal.

```
~$ sudo nano /etc/modules
```

Add the "snd-bcm2835" line at the end of the modules file, save, and restart.

```
~$ sudo reboot
```

After restarting, open terminal and enter as follows to check that the module has been loaded.

```
~$ lsmod | grep snd
```

Make sure that a line starting with "snd-bcm2835" appears in the displayed information.
Update the current Raspbian to the latest version of the kernal.

```
~$ sudo apt-get update
~$ sudo apt-get install rpi-update
```

The following procedure has only been checked in kernal version 4.14.98.
Downgrade kernal version 4.14.98 with the following command.

```
~$ sudo rpi-update a08ece3d48c3c40bf1b501772af9933249c11c5b
```

When finished, restart.

```
~$ sudo reboot
```

Install compile-dependent data.

```
~$ sudo apt-get install git bc libncurses5-dev
```

Install Bison and Flex.

```
~$ sudo apt-get install bison flex
```

Download and compile the kernel source.

```
~$ wget https://raw.githubusercontent.com/notro/rpi-source/master/rpi-source
~$ chmod +x rpi-source
~$ ./rpi-source -q --tag-update
```

```
~$ ./rpi-source --skip-gcc
```

7-2 Preparations for Compiling the I2S Module

Check if preparations for compiling I2S support are completed. Execute the following command.

```
~$ sudo mount -t debugfs debugs /sys/kernel/debug
```

In some environments, "mount: debugs is already mounted or /sys/kernel/debug busy" may appear. Continue anyway.

Execute the following command.

```
~$ sudo cat /sys/kernel/debug/asoc/platforms
```

Make sure that the following line appears in the execution results.

```
3f203000.i2s
snd-soc-dummy
```

If you are using Pi 3 or Pi 2, make sure that the module name is "3f203000.i2s".
If you are using Pi Zero or Pi Zero W, the module name will be "20203000.i2s".

Download the sound module.

```
~$ git clone https://github.com/PaulCreaser/rpi-i2s-audio
~$ cd rpi-i2s-audio
```

In the case of Pi Zero or Pi Zero W, changes must be made to "my_loader.c". For any other types, skip this procedure and proceed directly to compiling the module.

```
~/rpi-i2s-audio$ nano my_loader.c
```

Make the following changes with *nano*, and save.

```
.platform = "3f203000.i2s" → .platform = "20203000.i2s"
.name = "3f203000.i2s" → .name = "20203000.i2s"
```

Compile the module and install.

```
~/rpi-i2s-audio$ make -C /lib/modules/$(uname -r)/build M=$(pwd) modules
~/rpi-i2s-audio$ sudo insmod my_loader.ko
```

Make sure that the module has been loaded.

```
~/rpi-i2s-audio$ lsmod | grep my_loader
```

Make sure that the line starting with "my_loader" appears.

```
~/rpi-i2s-audio$ dmesg | tail
```

Make sure that a line including the following appears at the end of the kernal message log.

```
asoc-simple-card asoc-simple-card.0: snd-doc-dummy-dai <-> 3f203000.i2s mapping ok
```

If you are using Pi Zero or Pi Zero W, this will be "20203000.i2s".

7-3 Auto Module Loading at Startup

Set the module to load each time Raspberry Pi is started.

```
~/rpi-i2s-audio$ sudo cp my_loader.ko /lib/modules/$(uname -r)
~/rpi-i2s-audio$ echo 'my_loader' | sudo tee --append /etc/modules > /dev/null
~/rpi-i2s-audio$ sudo depmod –a
~/rpi-i2s-audio$ sudo modprobe my_loader
```

When finished, restart.

```
~/rpi-i2s-audio$ sudo reboot
```

7-4 Checking Devices

A list of usable devices will appear. Execute the following command.

```
~$ arecord –l
```

Make sure that "snd_rpi_simple_card" appears.

Change */home/pi/.asoundrc* as follows. If *.asoundrc* does not exist, create it.
*Note that in some versions of Raspberry-pi, *.asoundrc* may revert to the original when you restart.

```
pcm.!default {
        type plug
        slave {
                pcm     i2s
        }
        ttable.0.0 1.0
        ttable.0.1 1.0
}
pcm.i2s {
        type hw
        card 1
}
ctl.!default {
        type hw
        card 0
}
```

You can record a wav file in monaural with the following command.

```
~$ arecord -D plughw:1 -c1 -r 48000 -f S32_LE -t wav -V mono -v file.wav
```

If everything is functioning properly, you will see the VU meter respond at the bottom of the terminal window.
Copy *file.wave*, the wav file that was output, to your PC, play it, and make sure that audio was recorded.
*Reference sites
https://learn.adafruit.com/adafruit-i2s-mems-microphone-breakout/raspberry-pi-wiring-and-test

8. Removing the Sensor Evaluation Board

To remove the Sensor Evaluation Board from Raspberry Pi, turn off the power to the Sensor Evaluation Board, grasp the sides of the board without touching the USB port, connector, or other mounted components, and slowly remove the board.

Please check each region's Terms & Conditions by region website.

OMRON Corporation

Electronic and Mechanical Components Company

Regional Contact

Americas
https://www.components.omron.com/

Europe
http://components.omron.eu/

Asia-Pacific
https://ecb.omron.com.sg/

China
https://www.ecb.omron.com.cn/

Korea
https://www.omron-ecb.co.kr/

Japan
https://www.omron.co.jp/ecb/

Cat. No. A296-E1-02
1219(0619)(O)