**環境センサー、サーバーの構築**

**作者：劉　浥、崔　徳文、李　剣章**

**監査：下川原　英理**

**完了日：2021/07/09**

**1.PIとAIYの組み立て**

[1.1. RASPBERRY PIの組み立て](#_heading=h.gjdgxs)

[2.1. VISION KITの組み立てとテスト](#_heading=h.gjdgxs)

**2.センーサの組み立て**

2.[1. PI CAMERA V2.1の接続とテスト](#_heading=h.gjdgxs)

2.[2. HC-SR04の接続とテスト](#_heading=h.gjdgxs)

2.[3. AE-BME280の接続とテスト](#_heading=h.gjdgxs)

**3.P IとAIYのネットの設定**

3.[1. PI のネットの設定とテスト](#_heading=h.gjdgxs)

3.[2. AIYのネットの設定とテスト](#_heading=h.gjdgxs)

**4.SLACKに投稿できるようにする**

4.[1. アカウントの取得](#_heading=h.gjdgxs)

4.[2. 通知に使うチャット·ルームを用意する](#_heading=h.gjdgxs)

4.3[. MYサーバーからAPIを呼び出せるようにする](#_heading=h.gjdgxs)

4.4[.PYTHONを使ってAPI動作確認する](#_heading=h.gjdgxs)

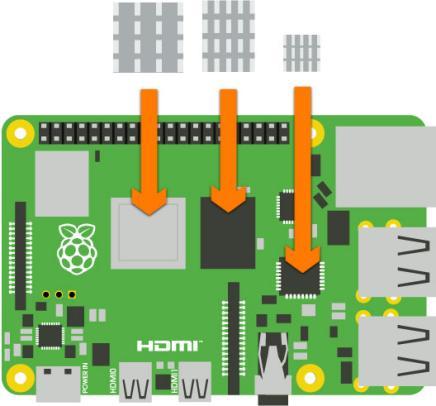
**1.PIとAIYの組み立て**

* 1. **RASPBERRY PIの組み立て**

## 【LINK】

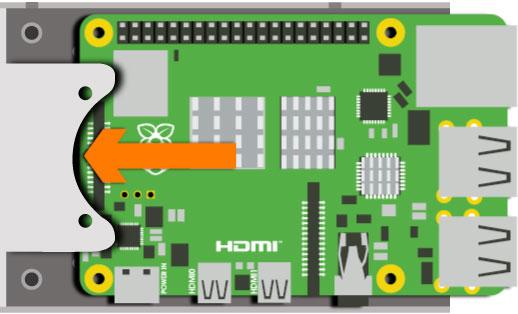
## [HTTPS://WWW.OKDO.COM/GETSTARTED/](https://www.okdo.com/getstarted/)

## 【STEP 1】 ヒートシンクを取り付ける



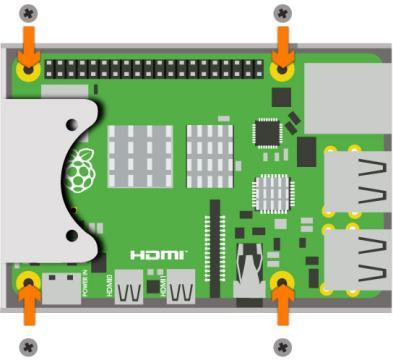
* ヒートシンクから保護シートをはがし、上の図に示されているコンポーネントに貼り付けます。

## 【STEP 2】 RASPBERRY PI をケースに入れる



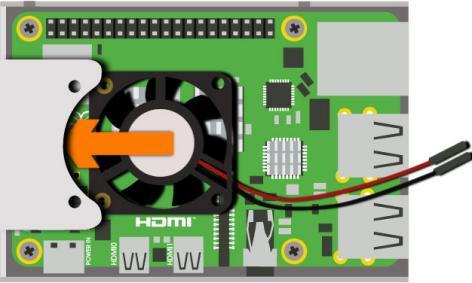
* Raspberry Pi をケースに入れる

## 【STEP 3】 ネジを固定する



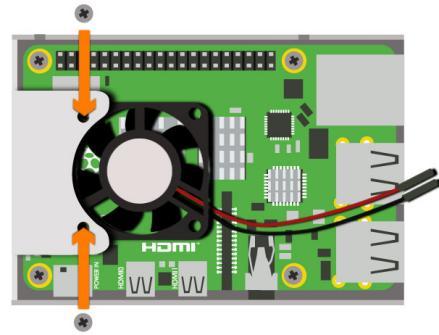
* Raspberry Pi をケースに固定する 4 本のネジを固定します。

## 【STEP 4】 ファンを配置する



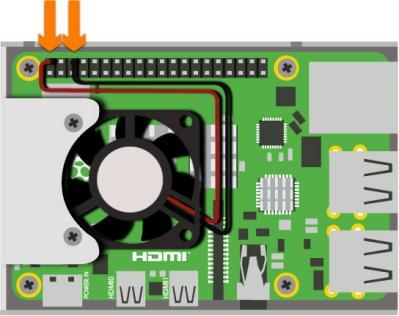
* ファンを所定の位置にスライドさせます

## 【STEP 5】ファンのネジを固定する



* ファンを所定の位置に固定する 2 本のネジを固定します。

## 【STEP 6】 ファンを接続する

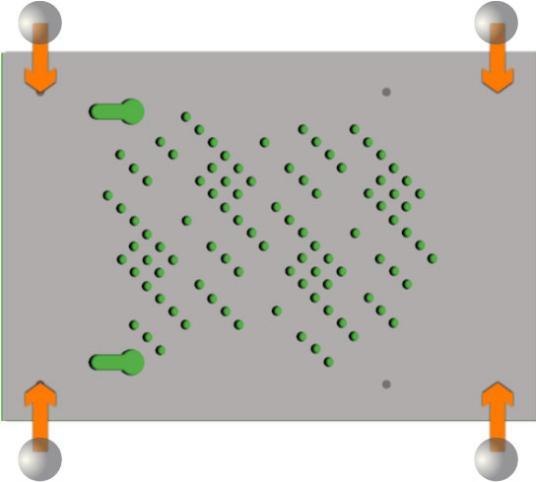


* ファンのプラス (赤) ケーブルを上記のピンに接続します。
* ファンのマイナス (黒) ケーブルを上記のピンに接続します。

警告  
**ファンのリードが正しいピンに接続されていることを確認してください。**

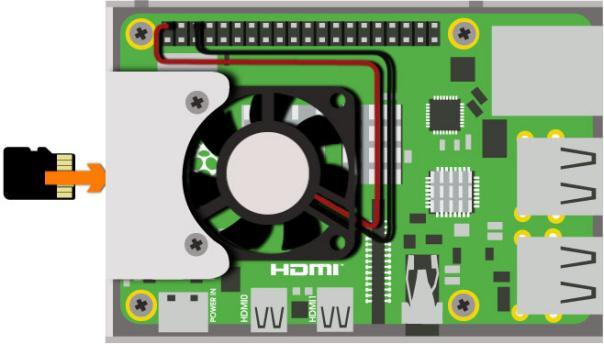
## 

## 【STEP 7】足をスティック



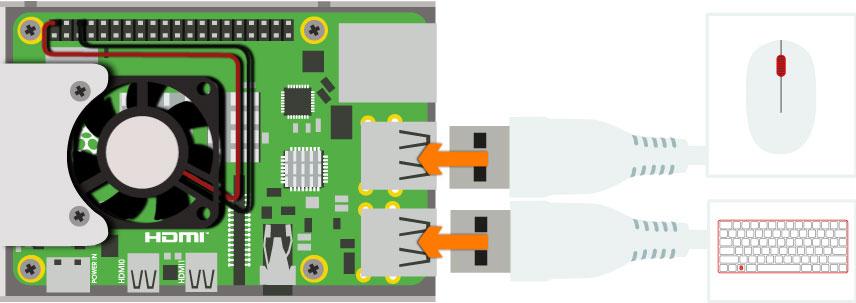
* Raspberry Pi とケースを裏返し、粘着性の脚をケースに貼り付けます。

## 【STEP 8】マイクロSDカードを挿入



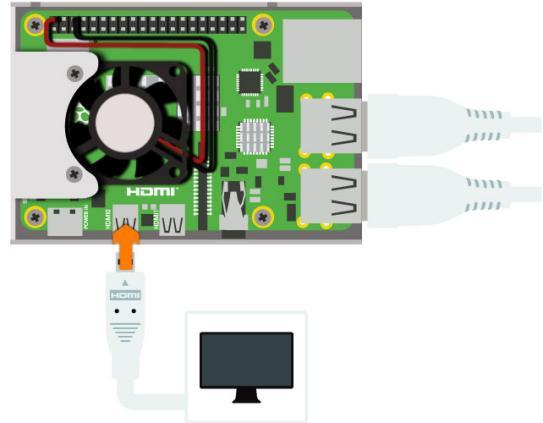
* マイクロ SD カードを Raspberry Pi のスロ​​ットに挿入します。

## 【STEP 9】マウスとキーボードを接続する



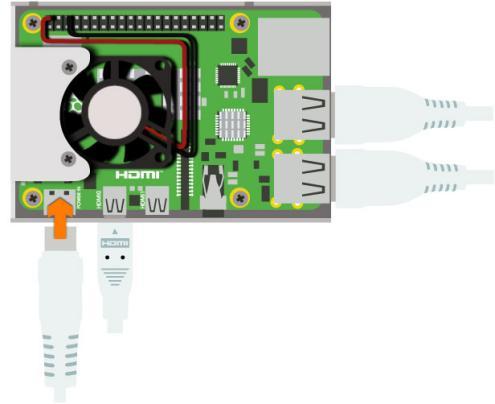
* マウスとキーボードを USB ポートのいずれか 2 つに接続します。

## 【STEP 10】接続画面



* 画面を**「HDMI0」**というラベルの付いた HDMI ポートに接続します**。**

## 【STEP 11】電源を接続する



* 電源ケーブルを**「POWER」**というラベルの付いたポートに接続します**。**
* 電源ケーブルのスイッチがオンになっていることを確認します。

## 【STEP 12】 画面の指示に従ってください



* Raspberry Pi を初めて起動すると、Welcome to Raspberry Pi アプリケーションがポップアップし、初期設定を案内します。
* RASPBIANというシステムをインストールしてください。
* デスクトップが表示されます。

**次の手順はオプションであり、プロジェクトで GPIO ピンを使用する予定がある場合は不要です。このプロジェクトはGPIOピンを使用するので、次の手順をPASSしでください。**

## 【STEP 13】ケースを閉じる

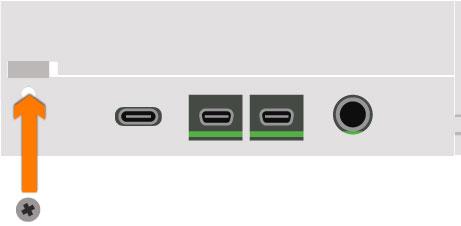


* ケースの上部を慎重に所定の位置にスライドさせ、すべての小さなタブが所定の位置にカチッとはまり、ネジ穴が合うようにします。

警告  
**ファンのリード線が邪魔にならないようにしてください。**

## 

## 【STEP 14】 ケースのネジを固定する



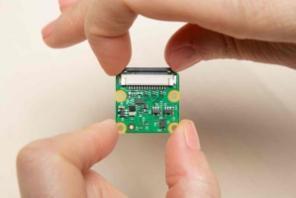
* 残りのネジを使用して、蓋をケースの底に固定します
  1. **VISION KITの組み立てとテスト**



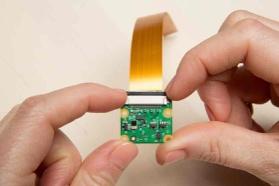
## 【LINK】

## HTTPS://AIYPROJECTS.WITHGOOGLE.COM/VISION

## 【SESSION 1】 内部フレームの折り畳み

コネクタのラッチを開く。

ラッチにロングフレックスを挿入する。

コネクタのラッチを閉じる。

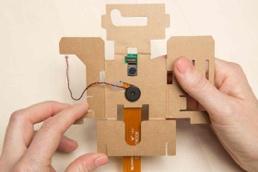
段ボールの向きを変える。

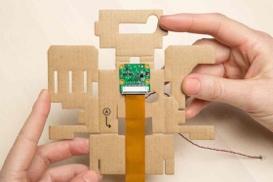
粘着ライナーを取り外す。

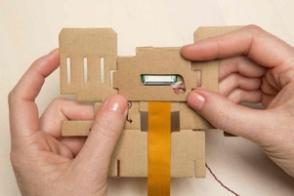
フラップAを折りたたみ。

裏返す。

ピエゾ式ブザーを挿入する。

カメラを追加する。

反対側に反転。

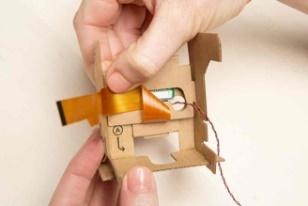
トップフラップを折りたたむ。

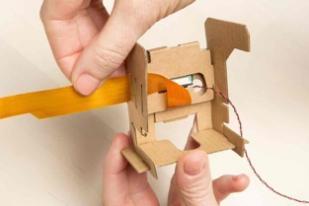
左と右のフラップを折る。

ボトムフラップを折る。

ブザーの線を通す。

長いフレックスを上に折り上げる。

長いフレックスを左に折る。

長いフレックスを通す

続ける。

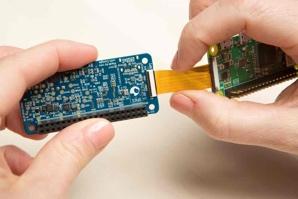
## 【SESSION 2】 基板の接続

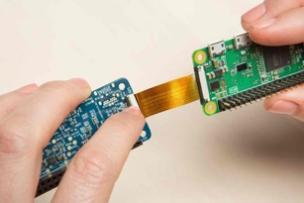
Raspberry Piの上部ケーブルコネクタを開ける。

ショートフレックスを挿入する。

ケーブルコネクターのラッチを閉じる。

ビジョンボンネットのケーブルコネクターのラッチを開く。

短いフレックスを挿入する。

ケーブルコネクターのラッチを閉じる。

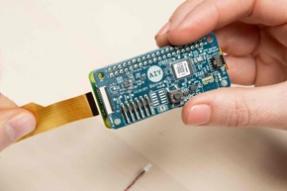
スタンドオフを挿入する。

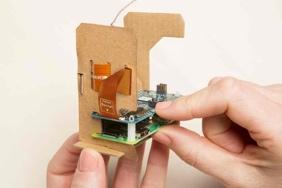
ボードを揃える。

短いフレックスを内側に押し込む。

基板の接続と接続の確認。

## 【SESSION 3】 枠に板を合わせる

基板とフレームを接続する。

ボードをフレームにスライドさせる。

ボタンハーネスを接続する。

## 【SESSION 4】 箱を作る

厚紙のAを折りたたむ。

Bを折りたたむ。

Cを折りたたむ。

接着剤を剥がす。

Dを折りたたむ。

三脚ナットを挿入する。

Eを折りたたむ。

Fを折りたたむ。

Eの接着剤を取り除く。

Gを折りたたむ。

反対側を折って固定する。

ボトムフラップを折る。

## 【SESSION 5】 一致させる

内部フレームへのスライド。

## 【SESSION 6】 LEDを取り付ける

LEDバッフルを取り付ける。

プライバシーLEDを取り付ける。

## 【SESSION 7】 ハードウェアを取り付ける

ナットにワイヤーを通す。

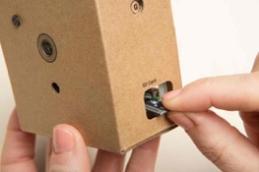
ケーブルをボックスに通す。

ボタンにワイヤーを差し込む。

ボタンを挿入する。

ボタンナットを固定する。

ボックスを閉じる。

SDカードを挿入する。

## 【SESSION 8】 完成



## 【SESSION9】 JOY　DETECTOR

2.1A以上電源に接続して、およそ２分後（初回起動の時間は長い）、音が出ってくるとVISION　KITが起動したと分かる。

VISION　KITのカメラで人を映ると、BUTTONが人の表情によって色が変わる。

**２．センーサの組み立て**

* 1. **PI CAMERA V2.1の接続とテスト**

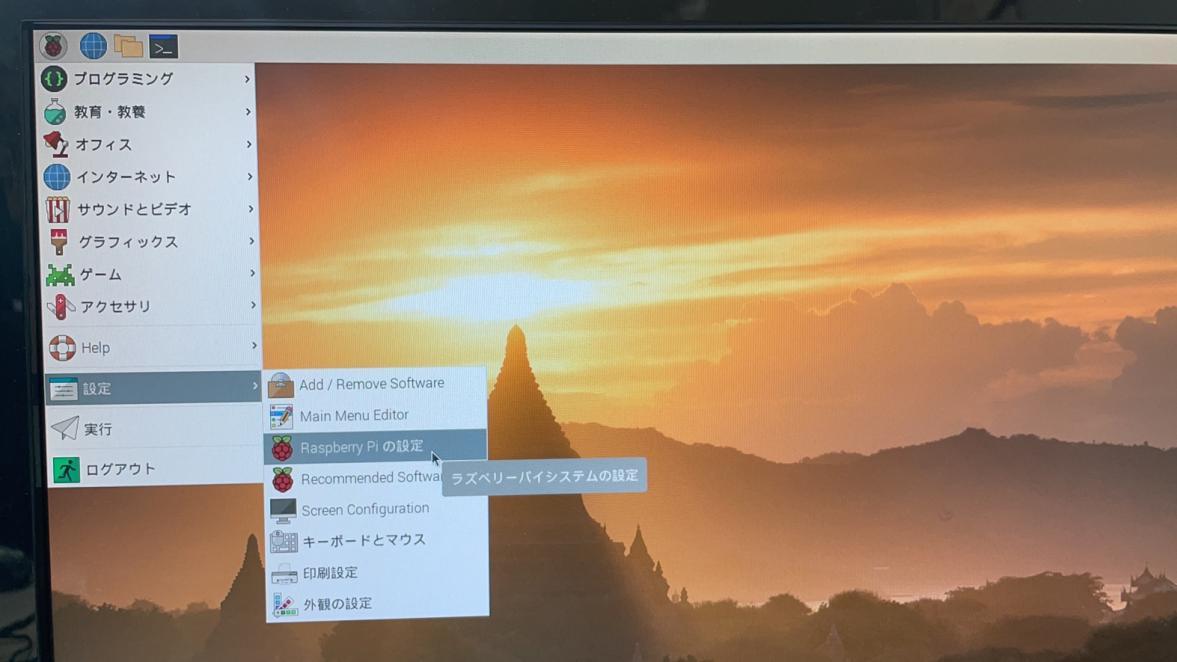
**【SESSION1】 接続**



このように、ケーブルの端をPiのCAMERAの接続箇所に差し込む

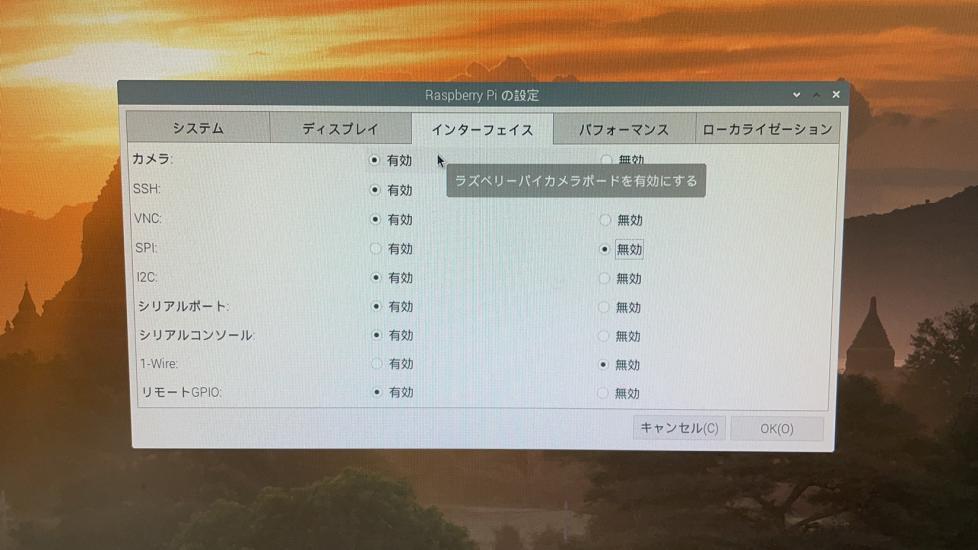
**【SESSION2】 設定**

カメラモジュールの接続ができたら、Raspberry Piの設定からカメラモジュールを読み込ませるための設定を確認しておこう。



左上のRaspberry Piのメニューボタンから「設定」＞「Raspberry Piの設定」をクリックし、システム設定を開く。

下図「インターフェイス」タブをクリックし、「カメラ」が「有効」になっていることを確認します。もし「無効」になっていた場合は「有効」を選択し、「OK」ボタンをクリックしてください。

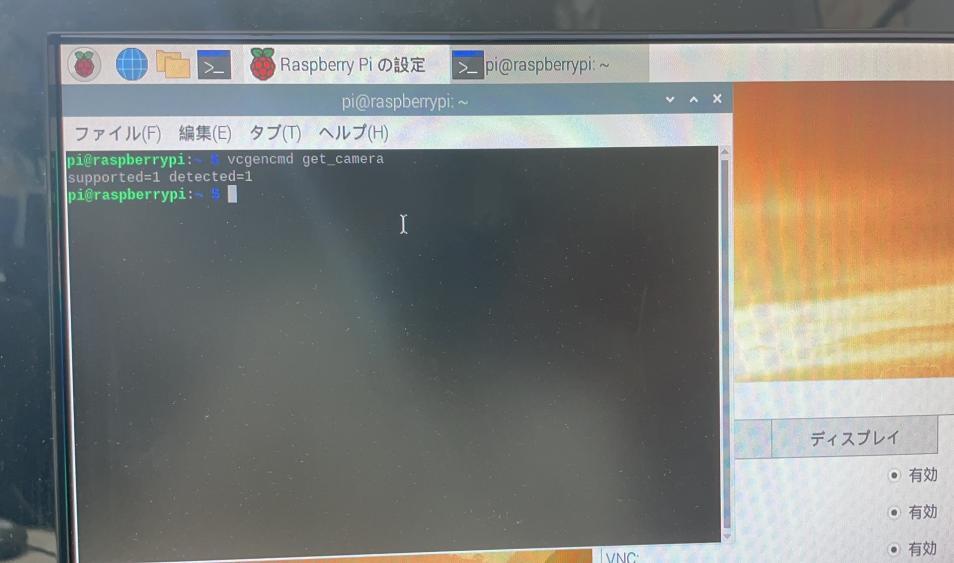


**【SESSION3】 テスト**

PIのLXTerminal（CMD）を起動する。

LXTerminal上で「vcgencmd get\_camera」コマンドをキーボードで入力する。

＞＞vcgencmd get\_camera



コマンドの次の行に「supported=1 detected=1」と表示されたら、カメラが認識されていることになる。

「supported=1 detected=0」など違う表示が出た場合は、正しくカメラモジュールが認識されていないということになる。

**正しくカメラモジュールが認識されていないと思われる場合は**

＞＞sudo apt-get update

**「sudo apt-get update」コマンドを実行してみてください**

次はテストコマンドです。

**（1）写真を撮る：**

LXTerminalで「sudo raspistill -o image.jpg」と入力して実行します。

＞＞sudo raspistill -o image.jpg

コマンドを実行すると、撮影画像が画面上に5秒ほど表示される。

「/home/pi」の中に、先ほどコマンドで入力していたものと同じ「image.jpg」という名前の画像ファイルがあります

**（2）3秒後に静止画を撮影：**

このコマンドは「3秒後に静止画を撮影」するコマンドになります。

＞＞sudo raspistill -t 3000 -o image.jpg

**（3）カメラをオープンして、ストリームで出力する：**

＞＞raspistill -t 0

**（４）撮影：**

＞＞raspivid -o test-vedio.h264

* 1. **HC-SR04の接続とテスト**

**使用する部品**

**ジャンパ線**

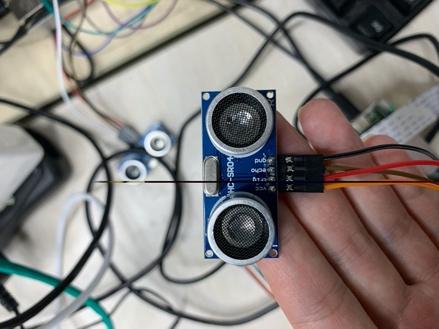
**地上的冲浪板

低可信度描述已自动生成**

ブレッドボートに差し込み、電子部品の間を電気的に接続する。

↑の写真では両側が「ピン」になっており、ブレッドボートの穴に差し込んで使用する。

**超音波センサ（HC-SR04）**



HC-SR04は、超音波が反射する時間を利用して非接触で距離を測定するモジュールだ。

「トリガー」ピンの状態をHighにすると超音波パルスが送信され、物体に反射して戻ってきた信号を受信する。超音波パルスを『送信した時間』と『戻ってきた信号を受信した時間』の差分を求めることで距離を測定する。

ラズパイのGPIOにて「トリガー」ピンを出力、「エコー」ピンを入力に割付けて使用する。

HC-SR04の主要なスペックは以下のようになる。

| 測定範囲 | ：2～400cm |
| --- | --- |
| 電源電圧 | ：DC5.0V |
| 動作電流 | ：15mA |
| 動作周波数 | ：40kHz |
| トリガー信号 | ：10μs |

**接続**

超音波センサHC-SR04には以下の4ヶのピンがある。

VCC：DC5.0V電源（一番左側）

Trig：トリガー信号

Echo：エコー信号

GND：グランド（一番右側）

ラズベリーパイとHC-SR04を以下のように接続する。

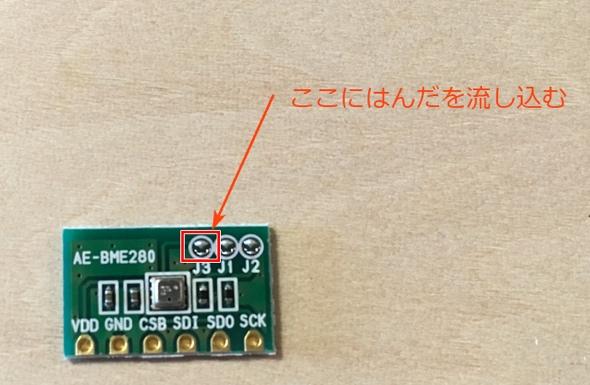
| **HC-SR04** | **ラズベリーパイ** | **ジャンパ線** |
| --- | --- | --- |
| VCC | ：5V | ：赤色 |
| Trig | ：GPIO 27番ポート（13番ピン） | ：青色 |
| Echo | ：GPIO 18番ポート（12番ピン）※抵抗経由 | ：黄色 |
| GND | ：GND | ：黒色 |

ジャンパ線は上記の色でなければいけない訳ではない。

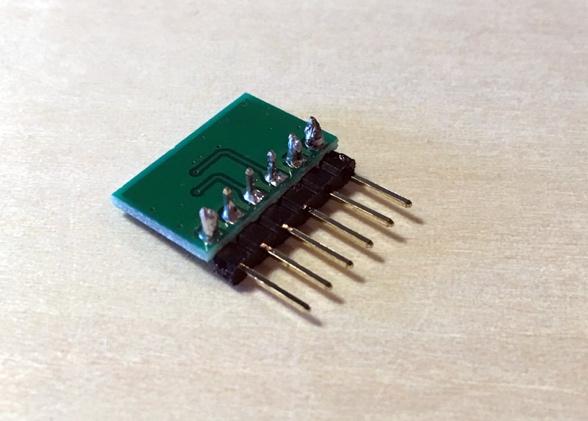
Trigに接続するGPIO 27番ポートはラズベリーパイからトリガーを出力するため出力モードとして使用する。

対して、Echoに接続するGPIO 18番ポートはHC-SR04から”受信した”信号をラズベリーパイが入力するため入力モードとして使用する。

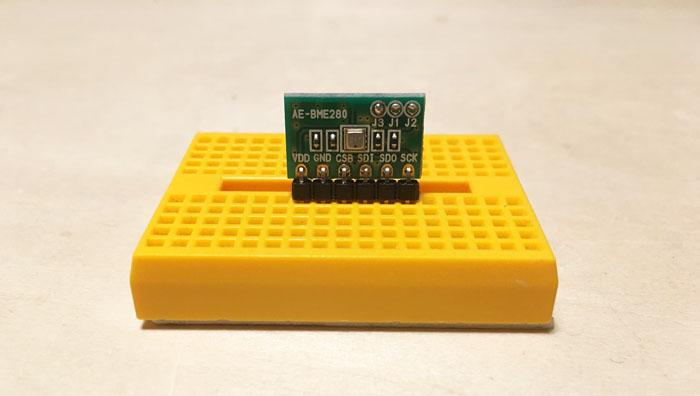
* 1. **AE-BME280の接続とテスト**



まずは、表面のはんだジャンパた。一番左の「J3」と書かれた部分に、はんだを流し込む。となりのピンとの間隔が狭いので、繋がってしまわないように注意しよう。

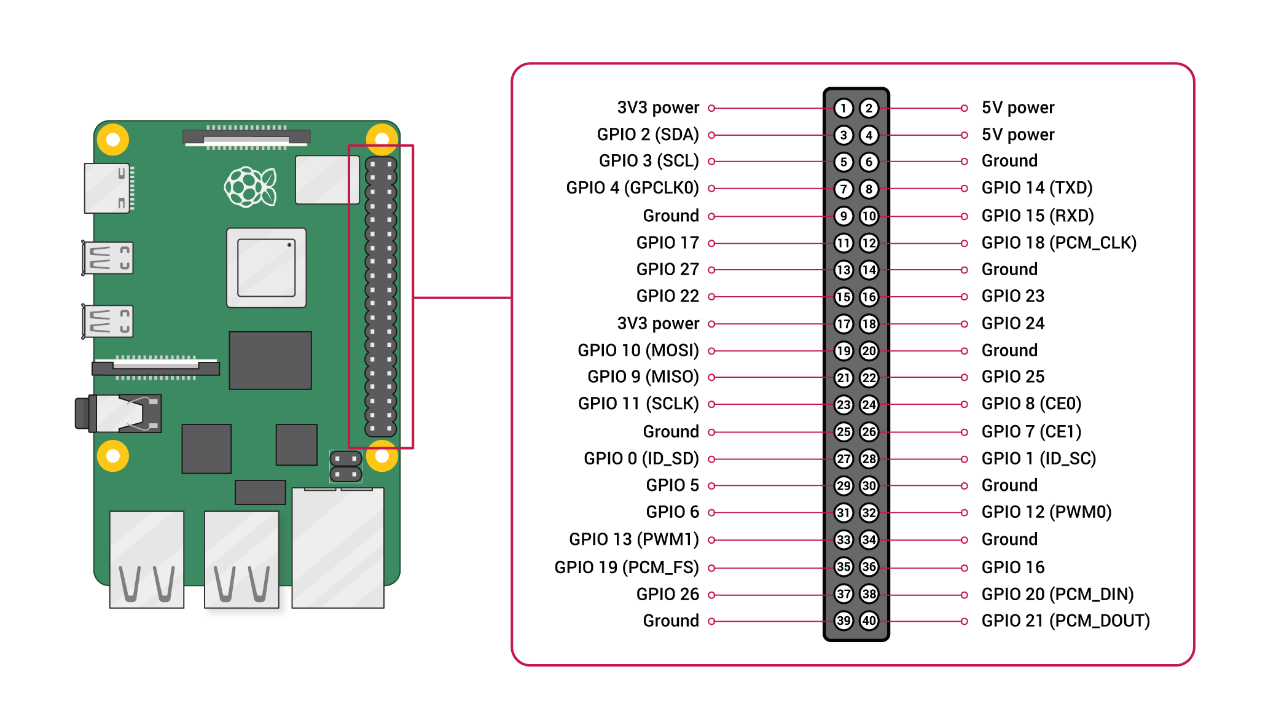


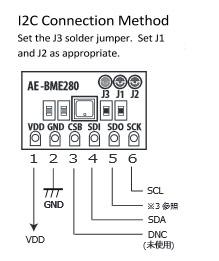
次はピンヘッダの接続です。こちらは裏側からはんだ付けした。  
ピンにこて先を当てて、少しあたためてから、はんだを軽く押し当てるようにするのがコツた。こて先が熱くなりすぎると、はんだが焦げてダマになってしまうので、落ち着いてゆっくり作業したいときは、はんだごてのコンセントを抜いておくことをおすすめする。



ピンヘッダが付いて、ブレッドボードに縦に差し込めるようになった。  
はんだ付けの最中、ピンヘッダにこて先が当たってしまって、黒い部分が少し溶けてしまいましたが、値の取得には問題なかった。

**配線**

GPIOポートを下図のように接続する。



VDDと3V3power

GNDとどのGroundでもいい

CSB接続不要

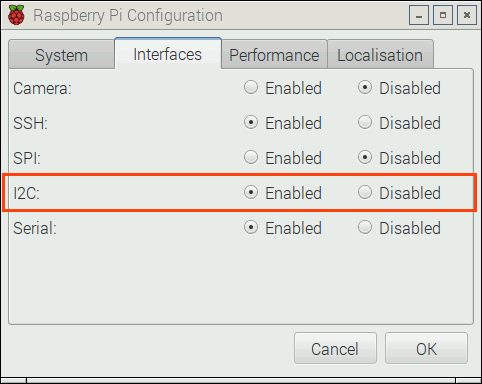
SDIとGPIO2（SDA）

SDOとGND，VDDとでもいけます。（要注意、そうすると、データの収取位置が変わります）。

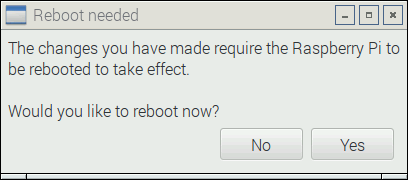
SCKとGPIO3（SCL）

**ラズベリーパイの設定**

メニューから[Preferences]-[Raspberry Pi Configuration]と選択し、設定画面を開きます。



「Interfaces」タブをクリックし、「I2C」を「Enable」に設定します。



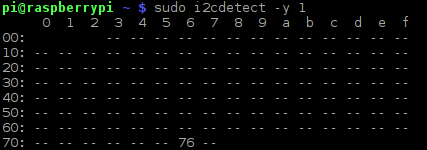
設定を有効にするため、再起動をするかのメッセージが表示されるので、Yesボタンをクリックしましょう。再起動後、I2C通信が有効になる。  
（OS：Raspbian Jessie/2015年11月21日リリース版にて確認済み）

ここで、I2Cを使うために必要なパッケージをインストールしておこう。コマンドラインから操作するには「i2c-tools」パッケージ、Pythonから操作するには「python-smbus」パッケージが必要になる。

インストールコマンド  
＞＞sudo apt-get install i2c-tools  
＞＞sudo apt-get python-smbus

i2cdetectコマンドを使用すると、I2Cで接続されているすべてのデバイスを検出できる。

＞＞sudo i2cdetect -y 1



76と表示された。16進数の76なので、正確には0x76たね。

この数字については、付属の説明書にも書かれている。

可能な問題：updateは正常に実行できない、404not foundを示して、smbuを正常にダウンロードできない。

解決方法；ソフトソースを替える。

インストールコマンド

＞＞Sudo nano /etc/apt/sources.list

次のように、ミラーサイトを使うように変更をした。

>>deb http://ftp.jaist.ac.jp/raspbian/ buster main contrib non-free rpi

>>deb http://ftp.tsukuba.wide.ad.jp/Linux/raspbian/raspbian/ buster main contrib non-free rpi

これでこのような問題は発生しなくなった。

**3.P IとAIYのネットの設定**

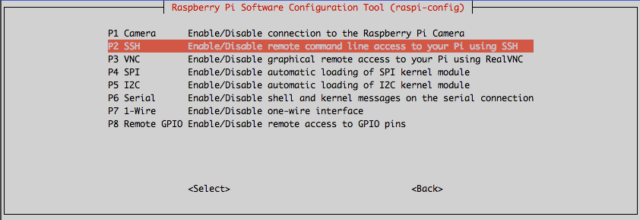
**3.1. PIのネットの設定とテスト**

Raspberry Piを使用する場合、ディスプレイ・キーボード・マウスを接続すれば、何ら問題はないのですが、出先などでそれらが使用できず、PCからリモート接続して操作する環境しかない場合には、RaspberryPiのWi-FiやIPアドレスの設定が必須になる。

**「SESSION1」 SSH機能を有効にさせる**

Raspberry Pi は、デフォルトでSSHはが無効になっています。 有効にするにはraspi-configで設定する必要がある。

＞＞**raspi-config**



ENTERで、SSHを有効にしてください。

**「SESSION2」WIFIに接続**

Wi-Fiの設定は、/etc/wpa\_supplicant/wpa\_supplicant.confに記述する。

＞＞sudo vi /etc/wpa\_supplicant/wpa\_supplicant.conf

以下の通り、WIFIのssidとpskを修正してください。

ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev

update\_config=1

country=JP

network={

ssid="Free001"

psk="password"

}

**「SESSION3」STABLE　IPアドレスを設定**

dhcpcd.confでwlan0 interfaceのstable ipを設定してください。

＞＞sudo nano /etc/dhcpcd.conf

# Example static IP configuration:

interface wlan0

static ip\_address=192.168.11.11/24

#static ip6\_address=fd51:42f8:caae:d92e::ff/64

static routers=192.168.11.1

#static domain\_name\_servers=192.168.11.1 8.8.8.8 fd51:42f8:caae:d92e::1

MYサーバが3台構成の場合に、僕らはPIのip\_addressを192.168.11.11、192.168.11.10、192.168.11.6にした。

>>sudo nano /etc/resolv.conf

namserver 8.8.8.8

のようにDNSを8.8.8.8に設定してください。

**「SESSION4」ネットをテスト**

ネットの詳細をみたいとCMDでifconfigを使ってください。

＞＞ifconfig

＞＞ping 192.168.11.11

＞＞ping 192.168.11.10

＞＞ping 192.168.11.6

＞＞Ping raspberrypi.local

PING raspberrypi.local (192.168.11.11): 56(84) data bytes

64 bytes from 192.168.11.11: icmp\_seq=0 ttl=64 time=2.037 ms

64 bytes from 192.168.11.11: icmp\_seq=1 ttl=64 time=0.559 ms

でネットをテストしてください。

**3.2. AIYのネットの設定とテスト**

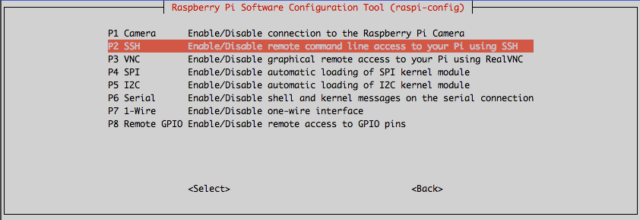
Raspberry Piを使用する場合、ディスプレイ・キーボード・マウスを接続すれば、何ら問題はないのですが、出先などでそれらが使用できず、PCからリモート接続して操作する環境しかない場合には、RaspberryPiのWi-FiやIPアドレスの設定が必須になる。

AIYはRaspberry Pi　ZEROを使用するので、最新バーションのシステムをインストールしてください。

**「SESSION1」 SSH機能を有効にする**

Raspberry Pi は、デフォルトでSSHはが無効になっています。 有効にするにはraspi-configで設定する必要がある。

＞＞raspi-config



ENTERで、SSHを有効にしてください。

**「SESSION2」WIFIに接続**

Wi-Fiの設定は、/etc/wpa\_supplicant/wpa\_supplicant.confに記述する。

＞＞sudo vi /etc/wpa\_supplicant/wpa\_supplicant.conf

以下の通り、WIFIのssidとpskを修正してください。

ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev

update\_config=1

country=JP

network={

ssid="Free001"

psk="password"

}

**「SESSION3」STABLE　IPアドレスを設定**

dhcpcd.confでwlan0 interfaceのstable ipを設定してください。

＞＞sudo nano /etc/dhcpcd.conf

# Example static IP configuration:

interface wlan0

static ip\_address=192.168.11.11/24

#static ip6\_address=fd51:42f8:caae:d92e::ff/64

static routers=192.168.11.1

#static domain\_name\_servers=192.168.11.1 8.8.8.8 fd51:42f8:caae:d92e::1

MYサーバが3台構成の場合に、僕らはPIのip\_addressを192.168.11.11、192.168.11.10、192.168.11.6にした。

>>sudo nano /etc/resolv.conf

namserver 8.8.8.8

のようにDNSを8.8.8.8に設定してください。

**「SESSION4」ネットをテスト**

ネットの詳細をみたいとCMDでifconfigを使ってください。

＞＞ifconfig

＞＞ping 192.168.11.11

＞＞ping 192.168.11.10

＞＞ping 192.168.11.6

＞＞Ping raspberrypi.local

PING raspberrypi.local (192.168.11.11): 56(84) data bytes

64 bytes from 192.168.11.11: icmp\_seq=0 ttl=64 time=2.037 ms

64 bytes from 192.168.11.11: icmp\_seq=1 ttl=64 time=0.559 ms

でネットをテストしてください。

**3.2. AIYのネットの設定とテスト**

AIYはRaspberry Pi　ZEROを使用する。

実際に設定の時、WIFIに接続してもネット使えない、JOY　DETECTOR無効にしてもカメラが使えないなどの場合があるので、https://aiyprojects.withgoogle.com/vision/#more-infoのガイドを参考して、SDカードに最新バーションのシステムをインストールしてください。

**「SESSION1」最新のAIYのRAPBIANシステムをインストール**

[https://aiyprojects.withgoogle.com/vision/#more-infoのガイドを参考してください](https://aiyprojects.withgoogle.com/vision/#more-info%E3%81%AE%E3%82%AC%E3%82%A4%E3%83%89%E3%82%92%E5%8F%82%E8%80%83%E3%81%97%E3%81%A6%E3%81%8F%E3%81%A0%E3%81%95%E3%81%84)。

**「SESSION2」SSH機能を有効にする**

AIYのssh機能は事前に有効にしたけど、確認した方がいい。

**「SESSION3」WIFIに接続とネットのテスト**

大体PIと同じの流れで設定する。

DESKTOPの右の上に、WIFI接続のICONもある。ここで設定してもいい（同じのWLANに接続してください）。

こう設定すれば、AIYは外部のサイトとの通信ができ、内部のIPとの通信ができない場合がある。それはROUTEの設定が衝突したからだ。

ROUTE　TABLEをクリアして、もう一度WIFIと接続してください。

＞＞sudo ip route flush table main

Cmdでroute命令で、ROUTEの詳細が見られる

＞＞route

**4.SLACKに投稿できるようにする**

**4.1. アカウントの取得**

Slackを利用するにはアカウントの登録が必要たから、以下の手順でアカウントの登録とメッセージの送信先となるチャンネルを作成する。

ウェブ·ブラウザで以下のURLにアクセスする。

https://slack.com/get-started#/email

メール·アドレスを入力し，「続行する」をクリックすると，登録用コード入力のページに切り替わります.メールで送られてくる登録用コードを入力する。



**4.2. 通知に使うチャット·ルームを用意する**

アカウントの取得に続き，画面の指示に従いワークスペースとチャンネルを作成する。

それぞれ好きな名称を設定できる。

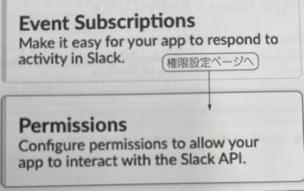
既にSlackを利用している場合は、今回使用するチャンネルの追加だけ行う。チャンネルの準備ができたらMyサーバからSlackにメッセージを書き込めるように、APIの設定を行う。

**4.3. MYサーバーからAPIを呼び出せるようにする**

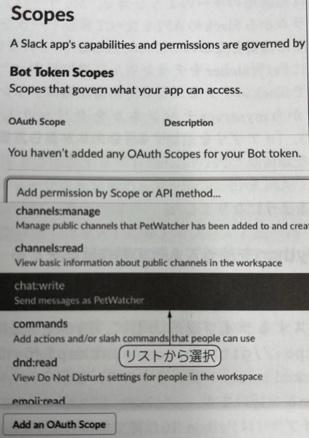
ブラウザで<https://api.slack.com/apps>にアクセスして、「Create New App」ボタンを押してSlack　Appの作成を行う。ここでは、アプリ名とワークスペース名を指定する。



BASIC　Information画面になるので、「Permissions」をクリックする。

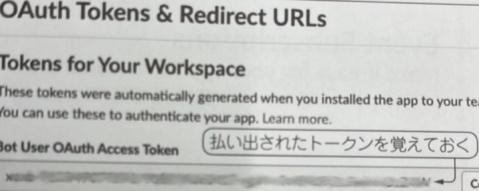


OAuth&Permissions画面に遷移するので、そこからScopesの設定領域のBot Token Scopesで権限を追加する。Permissionとして、「chat:write」を選ぶ。



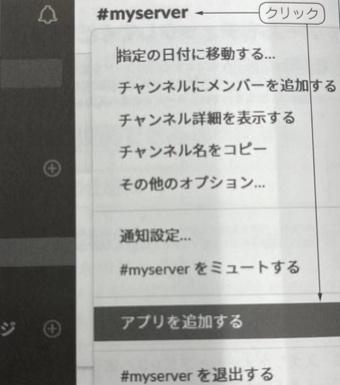
これでMyサーバからSlackのチャンネルにメッセージを書き込めるようになった。

遷移した画面で「Install App to workspace」ボタンをクリックし，登録を実行する。登録が完了すると，OAuth Access Tokenが払い出される。



これは認証用のキーのような物で，Myサーバ上のプログラムからSlackのAPIを使ってメッセージを投稿する際に必要となるので控えておく。

次にPetWatcherをチャンネルに追加する。ブラウザでSlackにアクセスして，画面左側のチャンネルの中からmyserverチャンネルをクリックする。



「＋アプリを追加する」をクリックし，表示される登録画面でPetWatcherアプリを追加する。

これでAPIを使ってMyサーバからメッセージを投稿できるようになった。

**4.4. PYTHONを使ってAPIの動作確認をする**

Pythonを使ってチャンネルヘメッセージを書き込みできるかどうか確かめる。PythonからSlackへアクセスするライブラリとして，Python slackclient

（https://github.com/slackapi/python-slackclient)を使うので，まずそれらのイン

ストールを行う。

ライブラリはPython 3.6以降で動作します（Raspbianには通常はインストール済み）。

**手順1:pipコマンドのインストール**

コマンドラインで次のコマンドを入力する。

＄ wget https://bootstrap.pypa.io/get-pip.pyd

＄ sudo python3 get-pip.py

Python3に対応したpipコマンドがインストールされる。

**手順2:Slack用APIのインストール**

pipコマンドでslackclientパッケージをインストールする。その際に，使用するPythonのバージョンに合わせる必要がある。バージョンを確認して使用するpipコマンドを決める。検証した環境ではPython 3.7だったためpip 3.7を使用した．

＄ python3 -V

＄ pip3.7 install slackclient

**手順3:メッセージを送信してみる**

準備が整ったので，以下の手順でメッセージをAPI経由で送信する。

Python3を起動して以下のように命令を打ち込みます。＇トークン＇の部分は，登録したアプリに対して払い出されたトークンを指定する.

＄ python3

import os

import slack

slack\_token='トークン＇

client = slack.webClient（token=slack token)

client.chat postMessage（channel="myserver",text="NMLGBD")

処理がエラーにならなければメッセージが届いているはず。ブラウザやスマートフォン·アプリでチャンネルmyserverを確認した。



PythonからSlack APIを使用する方法の詳細は，

https://slack.dev/python-slackclient/index.htmlを参照することができる。配布プログラムを使う限りは気にしなくてもよいが，機能拡張する際などにはここを参照して改造してください。