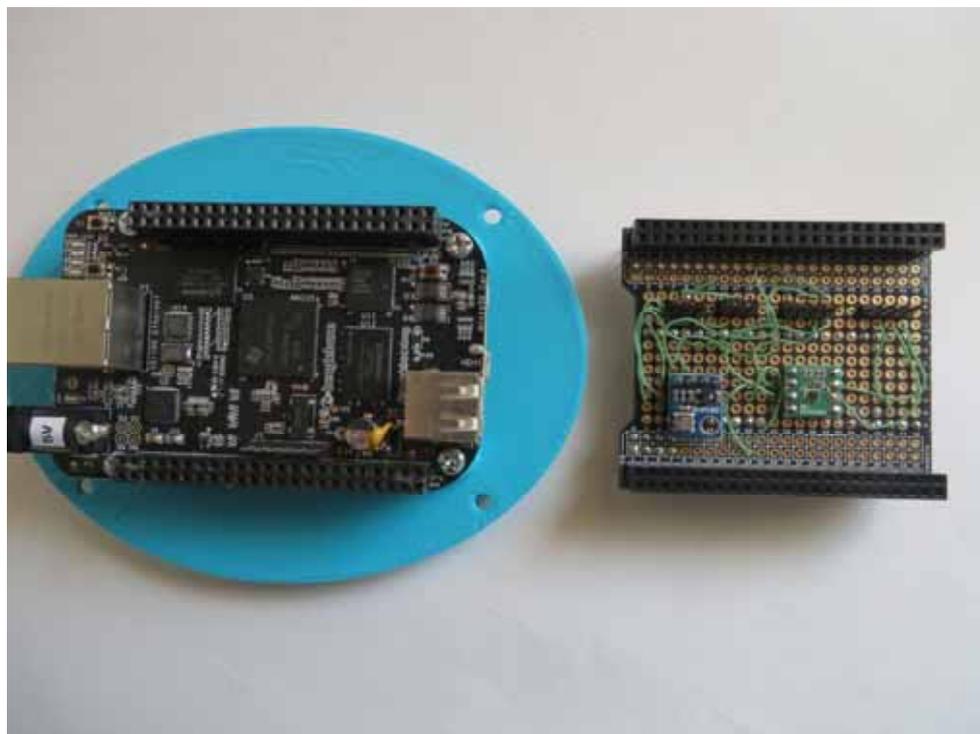


Beaglebone black 対応 Weather station
温度、湿度、照度、気圧ロギング
インターネット端末
CWLogger

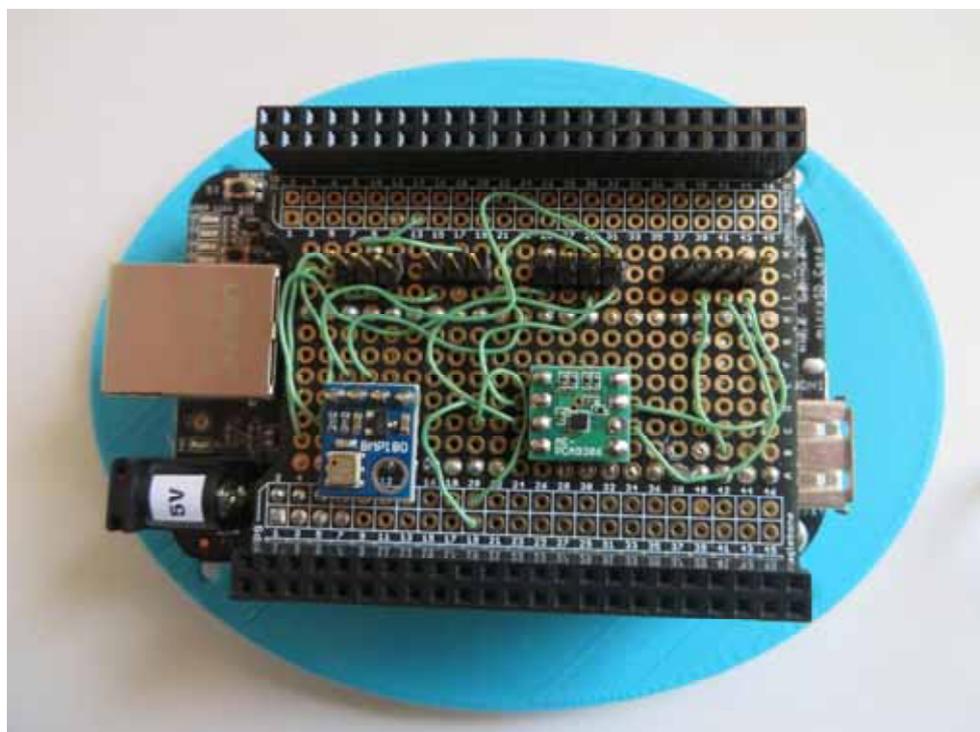


1. 本体組立

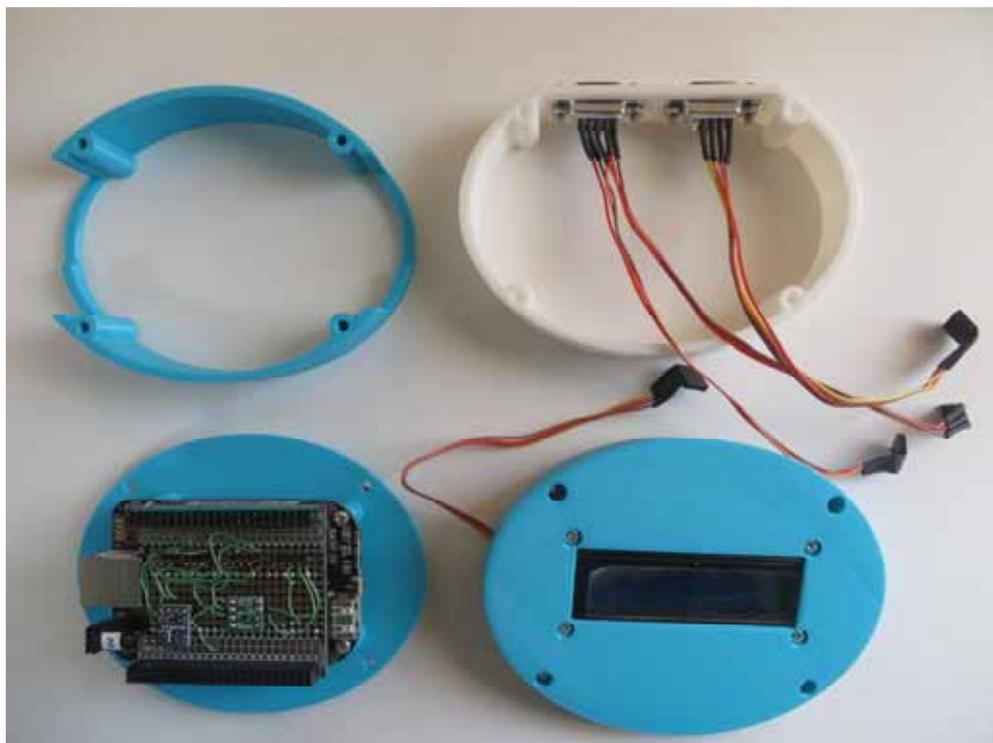
ボトムケースに BeagleBone Black を 4 つのビス、ナットで取り付けます。



次に拡張ボードを取り付けます。これは、2つの両サイドにコネクタに挿入します。



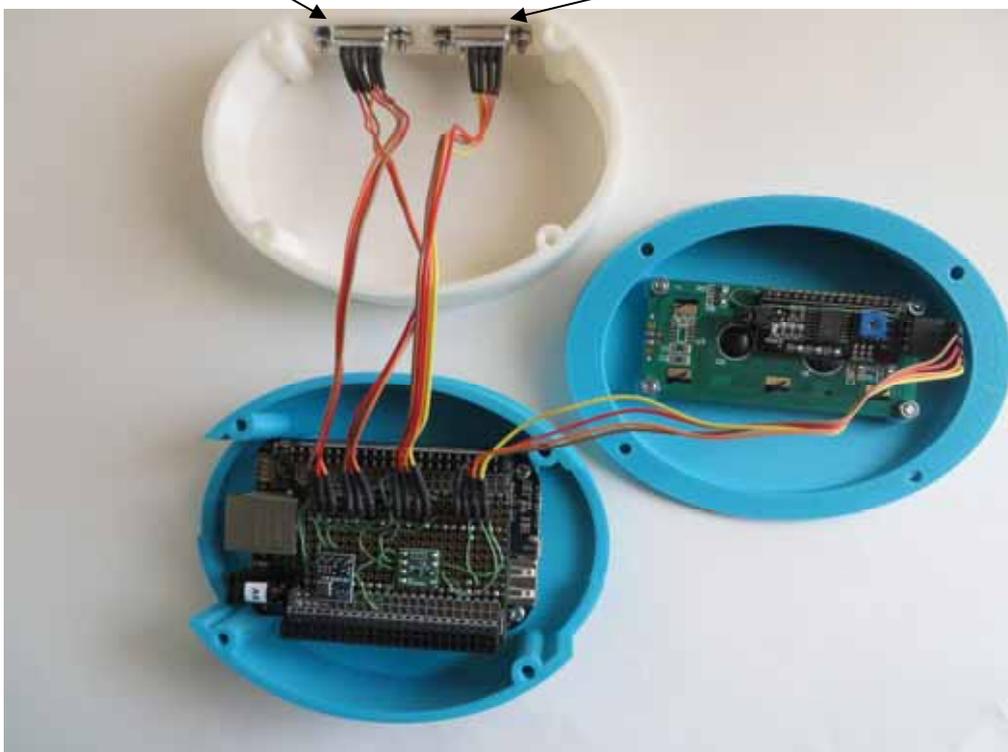
サイドケース2、1、およびトップケースを取り付けます。



各コネクタは左1ピンでケーブル色は茶色となります

2 x 温湿度センサーへ

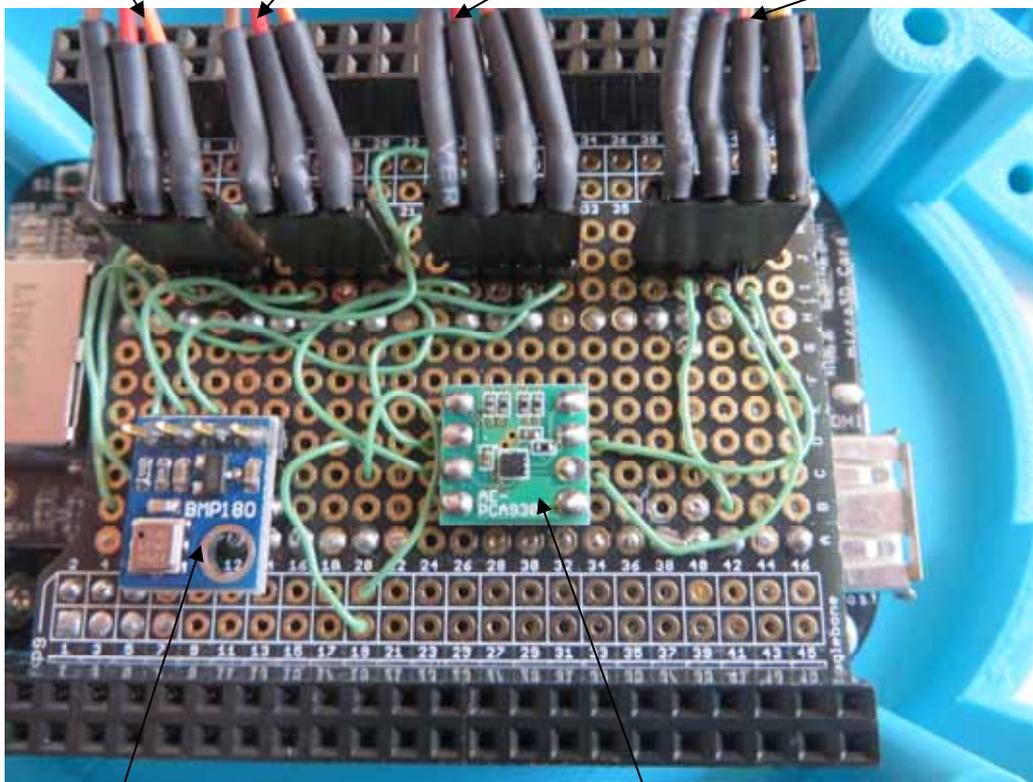
照度センサーへ



対応ケーブル、コネクタ、コネクタ極性に注意してください。間違って電源いれると破損します。

対応コネクタ、極性に注意

CH1 温湿度センサー CH2 温湿度センサー CH4 照度センサー LCD ケーブル

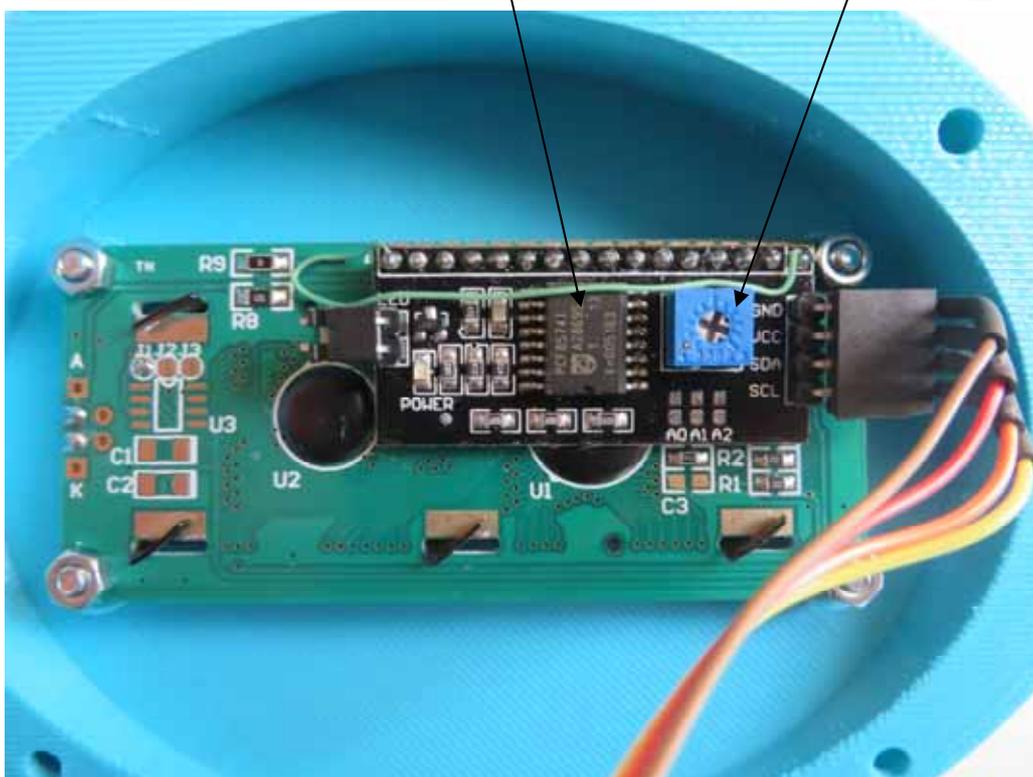


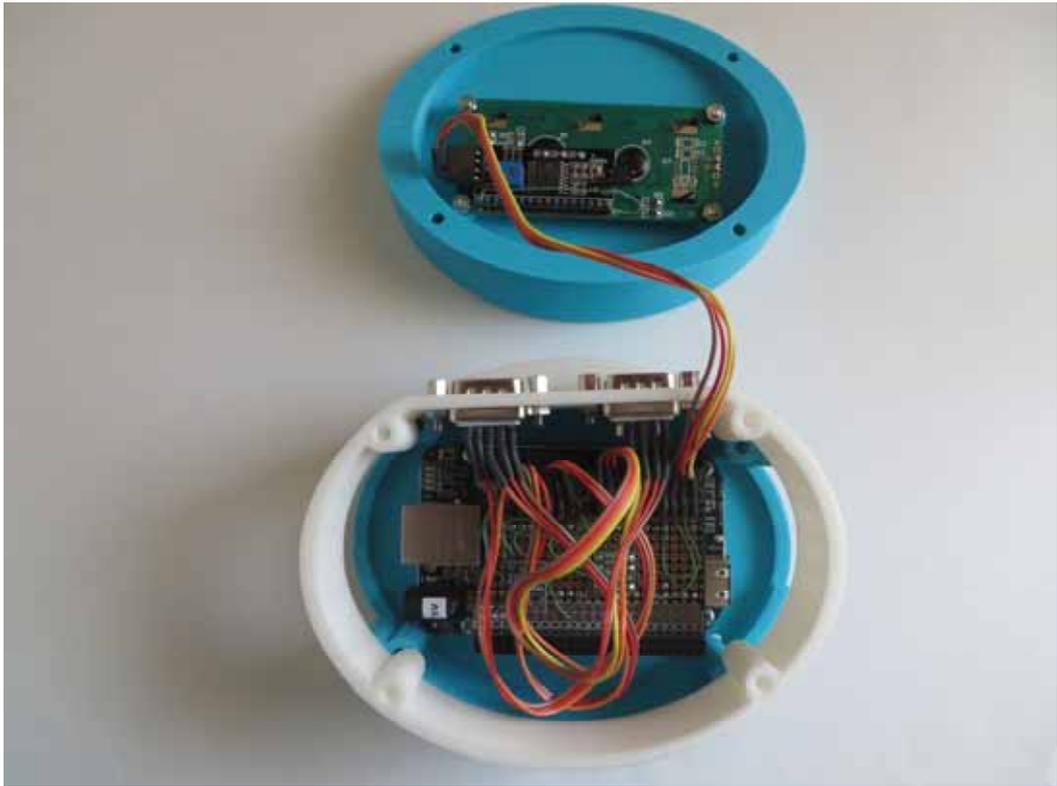
CH3 気圧センサー

I2C レベルコンバーター

LCD 1 16 ピンをショート

バックライト輝度調整

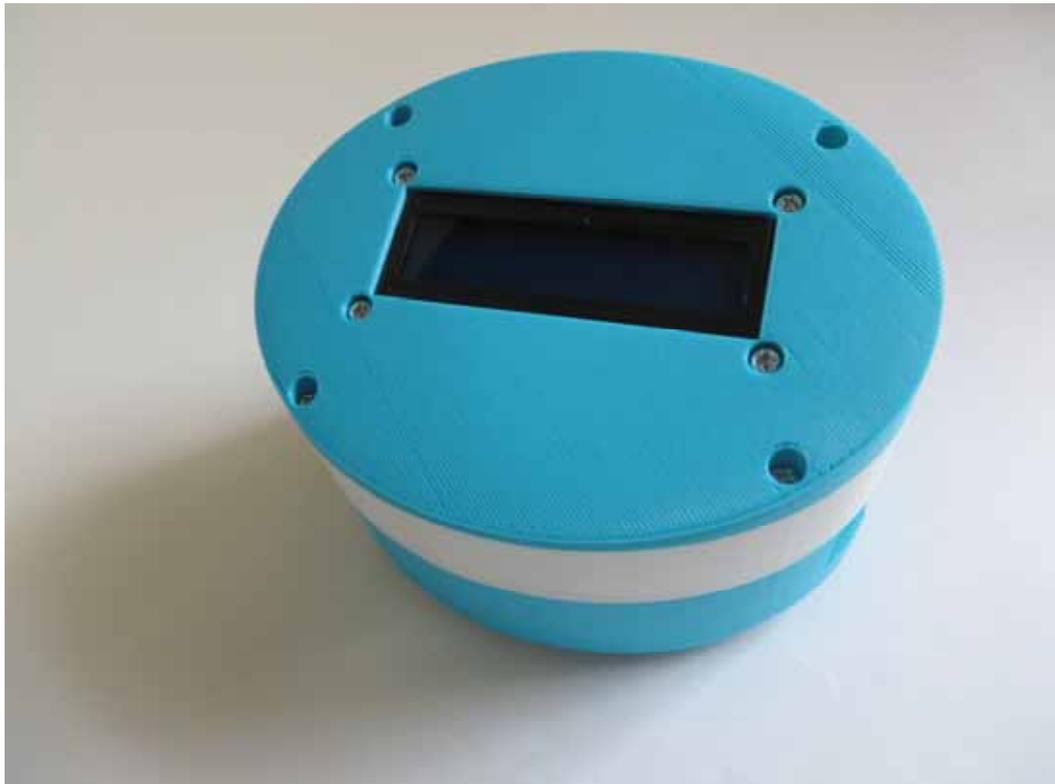




M3x60 のビス 4本 + ナット



4つのネジで全ケース固定、完成



2. センサーベース

CH4 照度センサー

CH1, CH2 温湿度センサー用ケーブル



CH1 温度湿度センサー

CH2 温湿度センサー

CH4 照度センサーケーブル

本体とセンサーベースの Dsub9 ケーブル 2 本を接続、完成、Dsub ケーブルを正しく接続。



3. 計測プログラム起動

以下のサイトからドライバーを PC にインストールします。

<http://beagleboard.org/getting-started>

マイクロ USB で PC と接続します。

マイクロ USB のみから電源も供給できますが、この電源のみでは不安定なので、
かならず、DC 電源も接続します。

パソコンに USB、LAN を接続する



192.168.7.2 をブラウザでアクセスします。
Beaglebone black のサイトが表示されます。

また、PC からターミナルソフトでログオンします。

```
ユーザ名  debian  
パスワード  temppwd
```

```
debian@beaglebone:~$  
~debian/CWLogger のディレクトリに移動します。  
cd CWLogger
```

このフォルダ内の cwlogger_bb2.py のプログラムを実行させます。
sudo python cwlogger_bb2.py

計測プログラム起動後、2 - 3 分間はデータをダミーで計測するため、データは表示されません。また CSV データは同じディレクトリに保存されます。

shutdown しないで DC 電源を抜くと破損する場合があります。

cwlogger_bb2.py のソース内で、メールアドレス、パスワード等、再設定して下さい。
現在、松山アドバンスのメールアドレスになっています。

```
192.168.7.2:22 - debian@beaglebone: ~/CWLogger VT
ファイル(F) 編集(E) 設定(S) コントロール(Q) ウィンドウ(W) ヘルプ(H)
Last login: Thu Apr 24 05:21:06 2014 from yamada-pc.local
debian@beaglebone: ~$ cd CWLogger
debian@beaglebone: ~/CWLogger$ sudo python cwlogger_bb2.py
```

```
192.168.7.2:22 - debian@beaglebone: ~/CWLogger VT
ファイル(F) 編集(E) 設定(S) コントロール(Q) ウィンドウ(W) ヘルプ(H)
01/31 13:42:28
01/31 13:42:29
01/31 13:42:30
01/31 13:42:31
01/31 13:42:32
01/31 13:42:33
01/31 13:42:34
01/31 13:42:35
01/31 13:42:36
01/31 13:42:37
01/31 13:42:38
01/31 13:42:39
01/31 13:42:40
temperature1:23.1C
humidity1:25.9%
temperature2:23.2C
humidity2:22.2%
pressure:1022.0hPa
illumination:9980 lux
01/31 13:42:43
01/31 13:42:44
01/31 13:42:45
01/31 13:42:46
```

4 . ハードウェア技術資料

温度湿度センサー

DHT22 Digital Temperature and Humidity Sensor AM2302 Module

<https://learn.adafruit.com/>

[dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging/software-dated](https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging/software-dated)

VCC と Data 間にプルアップ抵抗が必要、10K-1K 現在 2K を追加。

ない場合ケーブルは 10cm 程度、ある場合は 5m まで、これ以上は不可能。

気圧センサー

I2C buss

BMP180 Digital Barometric Pressure Sensor Board Module compatible with BMP085

<https://learn.adafruit.com/using-the-bmp085-with-raspberry-pi>

ケーブルは 5m 以上可能と思われる。実験必要。

照度センサー

I2C buss

BH1750FVI Digital Light Intensity Sensor Module For AVR Arduino 3V-5V Power

<http://bozontlabs.blogspot.jp/>

[2014/09/beaglebone-black-and-bh1750-light-sensor.html](http://bozontlabs.blogspot.jp/2014/09/beaglebone-black-and-bh1750-light-sensor.html)

ケーブルは 5m 以上可能と思われる。実験必要。

LCD モジュール

I2C buss(3.3 and 5V)

IIC/I2C 1602 Serial Blue Backlight LCD Display For Arduino 2560 UNO AVR A004

<http://arduino-info.wikispaces.com/LCD-Blue-I2C>

LCD 自体

16	15		14	13	12	11	10	9	8	7	6	5	4	3		2	1
K(-)	A(+)		BD7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	E	R/W	RS	Vo		VDD	VSS
GND	電流制限抵抗		<----->				<-- NC ----->		固定抵抗	5V	GND	バックライト					
lcd_rs	=	0															
lcd_en	=	2															
lcd_d4	=	4															
lcd_d5	=	5															
lcd_d6	=	6															
lcd_d7	=	7															
lcd_backlight	=	3															
#lcd_green	=	8															
#lcd_blue	=	3															

同等品

秋月：I2C 接続キャラクタ LCD モジュール 16 x 2 行 白色バックライト付

I2C コンバーター

I 2 C バス用双方向電圧レベル変換モジュール (P C A 9 3 0 6)

3.3V	<-> VREF1	VREF2 <-> 5V
SCL P9-19(I2C2_SCL)	<-> SCL1	SCL2 <-> LCD SCL
SDL P9-20(I2C2_SDA)	<-> SDL1	SDA2 <-> LCD SDA
	NC VPU	GNG <-> GND

5 . ドライバー技術資料

パッケージ管理システム

APT Advanced Package Tool >> apt-get
apt-setup コマンド >>/etc/apt/sources.list

apt-get と aptitude は同じ,
コマンドラインでは apt-get

ログオン

192.168.7.2
ユーザー名:debian
パスワード:temppwd

ログオンユーザー名 : debian@beaglebone

/home/debian
bin
Desktop

1) キーボード変更

まずログオンする

sudo nano で管理者モードでエディタ nano を起動

ファイル

etc/default/keyboard
を開く

```
-----  
# KEYBOARD CONFIGURATION FILE  
  
# Consult the keyboard(5) manual page.  
  
XKBMODEL="pc105"  
XKBLAYOUT="us"  
XKBVARIANT=""  
XKBOPTIONS=""  
  
BACKSPACE="guess"  
-----
```

以下に変更

```
-----  
# KEYBOARD CONFIGURATION FILE  
  
# Consult the keyboard(5) manual page.  
  
XKBMODEL="jp106"  
XKBLAYOUT="jp"  
XKBVARIANT=""
```

```
XKBOPTIONS=""
```

```
BACKSPACE="guess"
```

```
-----
```

2)

ATP ライブラリを最新版へ更新

```
sudo apt-get update
```

これをしないと、ntp のインストールでエラーになる
2 時間程度要した。

パッケージをすべて更新する

```
sudo apt-get upgrade
```

7 時間程度要した。

これは不要か？

3)

NTPD をインストールする

```
sudo apt-get install ntp
```

ntp サーバーを書き換える

```
sudo nano /etc/ntp.conf
```

```
#server 0.debian.pool.ntp.org iburst  
#server 1.debian.pool.ntp.org iburst  
#server 2.debian.pool.ntp.org iburst  
#server 3.debian.pool.ntp.org iburst
```

```
server ntp.nict.jp  
server ntp.nict.jp  
server ntp.nict.jp
```

ntp を再スタート

```
sudo /etc/init.d/ntp restart
```

```
[ ok ] Restarting ntp (via systemctl): ntp.service.  
debian@beaglebone:~$
```

ntpの動作確認

```
debian@beaglebone:~$ sudo ntpq -p
```

remote	refid	st	t	when	poll	reach	delay	offset	jitter
ntp-b3.nict.go.	.NICT.	1	u	50	64	3	16.537	-39.940	2.836
ntp-a3.nict.go.	.NICT.	1	u	49	64	3	16.166	-39.768	2.457
ntp-a2.nict.go.	.NICT.	1	u	51	64	3	16.302	-39.822	2.105

同期がとれると

remote	refid	st	t	when	poll	reach	delay	offset	jitter
+ntp-b3.nict.go.	.NICT.	1	u	9	64	77	16.421	-30.703	10.025
+ntp-a3.nict.go.	.NICT.	1	u	9	64	77	15.985	-30.684	10.030
*ntp-a2.nict.go.	.NICT.	1	u	10	64	77	15.974	-30.858	9.948

4) タイムゾーンの設定

```
sudo dpkg-reconfigure tzdata
```

起動設定画面で Asia/Tokyo を選択する

```
Current default time zone: 'Asia/Tokyo'
```

```
Local time is now:      Wed Jan 28 19:51:47 JST 2015.
```

```
Universal Time is now:  Wed Jan 28 10:51:47 UTC 2015.
```

5) パイソン他基本ツールインストール

```
sudo apt-get install git build-essential python-dev python-setuptools python-pip python-smbus
```

```
>>>
```

```
build-essential is already the newest version.
```

```
python-setuptools is already the newest version.
```

```
git is already the newest version.
```

```
python-pip is already the newest version.
```

```
The following NEW packages will be installed:
```

```
python-dev python-smbus
```

手動インストール

```
sudo apt-get update
sudo apt-get install build-essential python-dev python-pip python-smbus -y
git clone git://github.com/adafruit/adafruit-beaglebone-io-python.git
cd adafruit-beaglebone-io-python
sudo python setup.py install
cd ..
sudo rm -rf adafruit-beaglebone-io-python
```

```
sudo pip install Adafruit_BBIO
```

```
>>>
Requirement already satisfied (use --upgrade to upgrade): Adafruit-BBIO in
/usr/local/lib/python2.7/dist-packages
Cleaning up...
```

I2C ツールインストール

```
sudo apt-get install i2c-tools
```

```
>>>
Reading package lists... Done
Building dependency tree
Reading state information... Done
i2c-tools is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 103 not upgraded.
```

6) 温湿度センサーライブラリインストール

CH1

Vin P9-3.3V

Gnd P9-GND

Data P8-11

CH2

Vin P9-3.3V

Gnd P9-GND

Data P8-13

ライブラリインストール

この前にカレントを home に戻すこと

```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
```

```
>>>
Cloning into 'Adafruit_Python_DHT'...
remote: Counting objects: 97, done.
```

```
remote: Total 97 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (97/97), done.
```

```
cd Adafruit_Python_DHT
sudo python setup.py install
```

サンプル実行

```
cd examples
sudo ./AdafruitDHT.py 2302 P8_11
```

7) 気圧センサーライブラリインストール

配線

```
I2C バス:1(default)
0x77
```

```
Vin P9-3.3V
Gnd P9-GND
SCL P9-19
SDL P9-20
```

ライブラリインストール

この前にカレントを home に戻すこと

```
git clone https://github.com/adafruit/Adafruit_Python_BMP.git
```

```
cd Adafruit_Python_BMP
sudo python setup.py install
```

サンプル実行

```
cd examples
sudo python simpletest.py
```

8) 照度センサーライブラリインストール

```
I2C buss:1(default)
address:0x23
```

```
Vin P9-3.3V
Gnd P9-GND
SCL P9-19(I2C2_SCL)
SDL P9-20(I2C2_SDA)
```

専用ライブラリのインストールは不要。

Buss1 の検出

```
sudo i2cdetect -r 1
```

8) LCD モジュールライブラリ

```
I2C buss:1(default)  
address:0x23
```

```
Vin P9-5V  
Gnd P9-GND  
SCL Convert SCL  
SDL Convert SDL
```

この前にカレントを home に戻すこと

```
git clone https://github.com/adafruit/Adafruit_Python_CharLCD.git
```

```
cd Adafruit_Python_CharLCD  
sudo python setup.py install
```

メモ-----

```
git clone https://github.com/adafruit/Adafruit_Python_CharLCD.git
```

の実行で

カレントフォルダ以下のフォルダ、ファイルが作成される

```
Adafruit_Python_CharLCD  
フォルダ  Adafruit_CharLCD  
          examples
```

```
ファイル LICENSE  
          README.md  
          ez_setup.py  
          setup.py
```

```
sudo python setup.py install
```

の実行で

カレントフォルダ以下のフォルダ、ファイルが作成される

```
フォルダ  build  
          dist  
          Adafruit_CharLCD.egg-info
```

```
ファイル LICENSE  
          README.md  
          ez_setup.pyc
```

I2C の指定アドレスにデバイスがないと、python エラーとなることに注意
remote I/O error

6 . ケース 3D プリンター用 STL ファイル一覧

本体トップケース	BgbTopCaseV2.STL
本体サイドケース 1	BbbScase1V2.STL
本体サイドケース 2	BbbScase2V2.STL
本体ボトムケース	BeagleBoneCaseV2.stl

センサーベース	SensorBase.STL
温度センサーケース	TempCase.STL
温度センサーカバー	TempCover.STL
照度センサーケース	LuxCase.STL
照度センサーカバー	LuxCover.STL