近接センサー(APDS-9930)

更新履歴

版数	更新内容	更新日
0.2	Refsを追加した。	2023/01/11
0.1	初版を0.1版とする。	2022/09/09

目次

近接センサー（APDS-9930）概要

APDS-9930

Digital Proximity and Ambient Light Sensor

AVAGO TECHNOLOGIES

Data Sheet

Description

The APDS-9930 provides digital ambient light sensing (ALS), IR LED and a complete proximity detection system in a single 8 pin package. The proximity function offers plug and play detection to 100 mm (without front glass) thus eliminating the need for factory calibration of the end equipment or sub-assembly. The proximity detection feature operates well from bright sunlight to dark rooms. The wide dynamic range also allows for operation in short distance detection behind dark glass such as a cell phone. In addition, an internal state machine provides the ability to put the device into a low power mode in between ALS and proximity measurements providing very low average power consumption. The ALS provides a *photopic* response to light intensity in very low light condition or behind a dark faceplate.

The APDS-9930 is particularly useful for display management with the purpose of extending battery life and providing optimum viewing in diverse lighting conditions. Display panel and keyboard backlighting can account for up to 30 to 40 percent of total platform power. The ALS features are ideal for use in notebook PCs, LCD monitors, flat-panel televisions, and cell phones.

The proximity function is targeted specifically towards near field proximity applications. In cell phones, the proximity detection can detect when the user positions the phone close to their ear. The device is fast enough to provide proximity information at a high repetition rate needed when answering a phone call. This provides both improved "green" power saving capability and the added security to lock the computer when the user is not present. The addition of the micro-optics lenses within the module, provide highly efficient transmission and reception of infrared energy which lowers overall power dissipation.

Ordering Information

Part Number	Packaging	Quantity
APDS-9930	Tape & Reel	5000 per reel
APDS-9930-140	Tape & Reel	1000 per reel
APDS-9930-200	Tape & Reel	1000 per reel

Features

ALS, IR LED and Proximity Detector in an Optical Module

- Ambient Light Sensing (ALS)
  - Approximates Human Eye Response
  - Programmable Interrupt Function with Upper and Lower Threshold
  - Up to 16-Bit Resolution
  - High Sensitivity Operates Behind Darkened Glass
  - Low Lux Performance at 0.01 lux
- Proximity Detection
  - Fully Calibrated to 100 mm Detection
  - Integrated IR LED and Synchronous LED Driver
  - Eliminates "Factory Calibration" of Prox
- Programmable Wait Timer
  - Wait State Power – 90 µA Typical
  - Programmable from 2.7 ms to > 8 sec
- I²C Interface Compatible
  - Up to 400 kHz (I²C Fast-Mode)
  - Dedicated Interrupt Pin
- Sleep Mode Power - 2.2 µA Typical
- Small Package L3.94 x W2.36 x H1.35 mm

Applications

- Cell Phone Backlight Dimming
- Cell Phone Touch-screen Disable
- Notebook/Monitor Security
- Automatic Speakerphone Enable
- Automatic Menu Pop-up
- Digital Camera Eye Sensor

Package Diagram

- Description
  - APDS-9930 は、デジタル I2C 互換インタフェース照度センサ (ALS) と近接センサを IR LED と共に 1 つの 8 ピン・パッケージに組み込んでいます。近接センサは、100mm 以内の物体を検出するように完全校正 (キャリブレーション) され、したがって最終装置またはサブアセンブリの工場出荷時の校正 (キャリブレーション) が不要です。
- Features
  - 光学モジュール内の ALS、IR LED、および近接検出器
    - 環境光センシング (ALS)
    - 近接検出100 mm 検出用に完全に校正済み
    - I2Cインターフェース対応最大 400 kHz (I2C 高速モード)
- Applications
  - 携帯電話のバックライトの調光
  - 携帯電話のタッチスクリーンを無効にする
  - ノートブック/モニターのセキュリティ
  - スピーカーフォンの自動有効化
  - 自動メニューポップアップ
  - デジタルカメラのアイセンサー

[Refs]

https://www.mouser.jp/datasheet/2/678/av02-3190en_ds_apds-9930_2015-11-13-1828481.pdf

https://jp.broadcom.com/products/optical-sensors/integrated-ambient-light-proximity-sensors/apds-9930

Spresenseメインボードピンレイアウト図

Spresense メインボードの基準I/O電圧は1.8Vです。1.8V以上の電圧を加えると Spresense のチップセットが破壊される可能性があります。取扱いには十分注意してください。

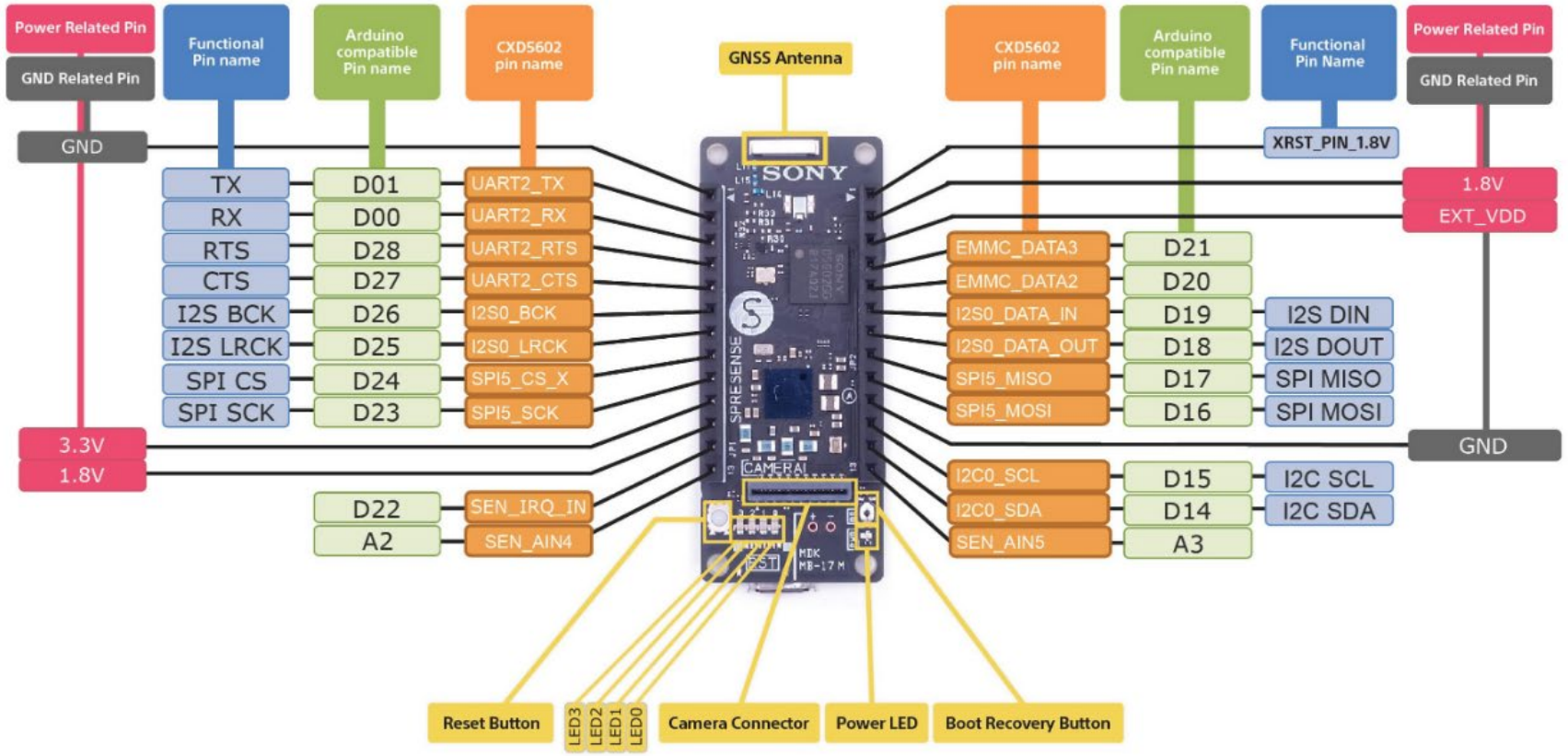
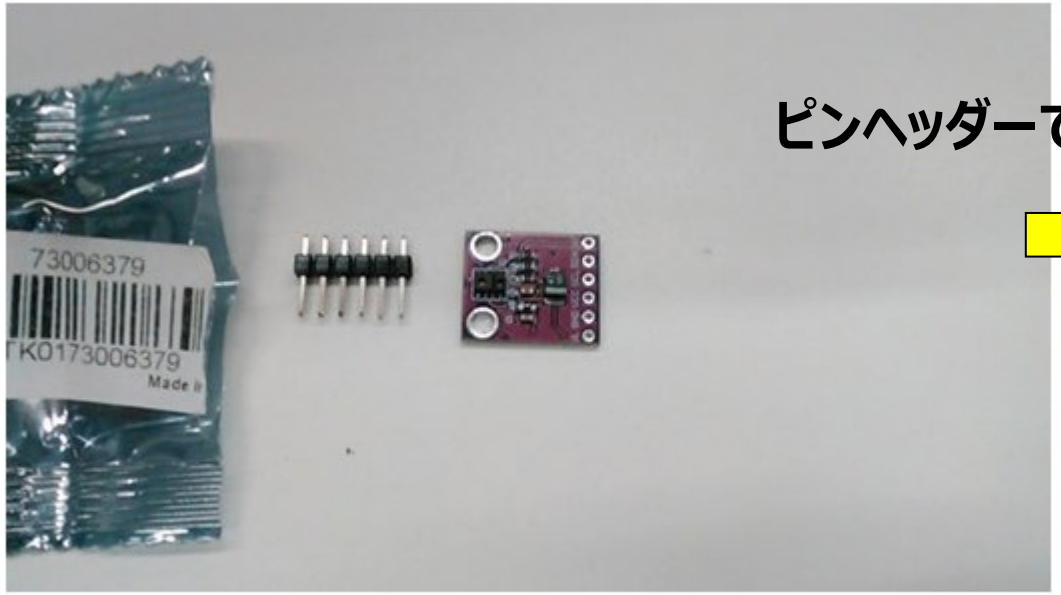


図 3. Spresense メインボード

[Refs]
https://developer.sony.com/develop/spresense/docs/introduction_ja.html#_spresense_%E3%83%A1%E3%82%A4%E3%83%B3%E3%83%9C%E3%83%BC%E3%83%89

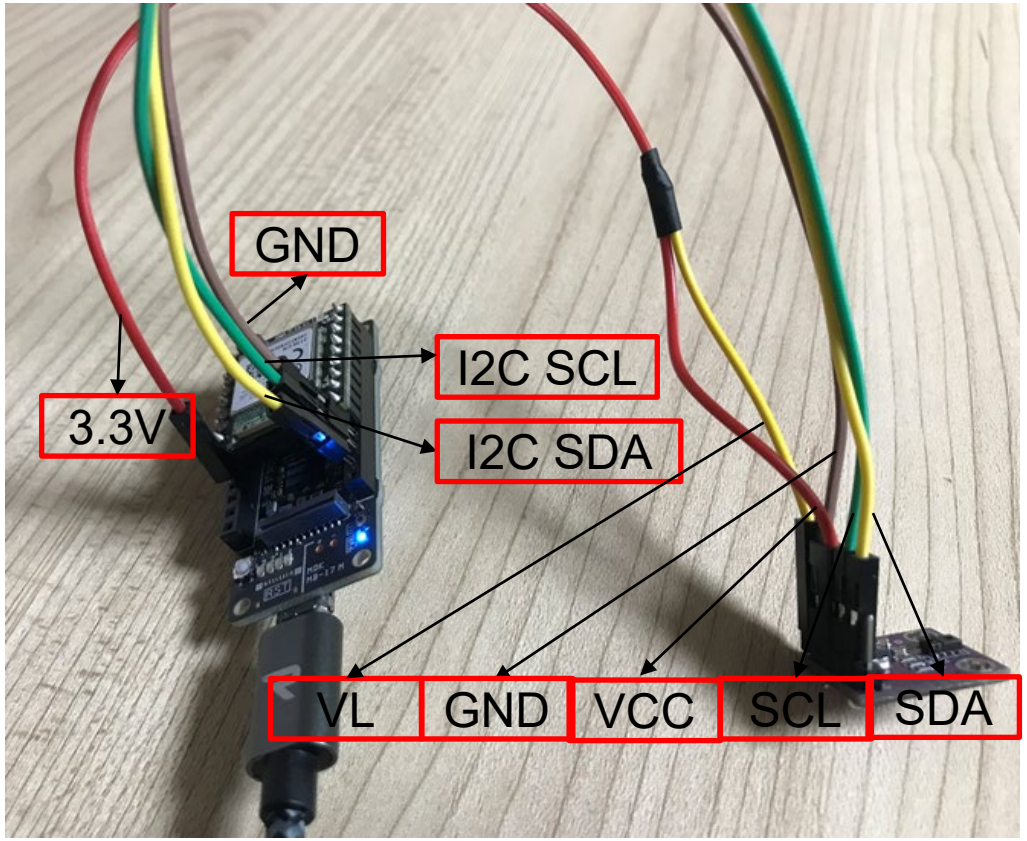
APDS-9930はんだ付け作業



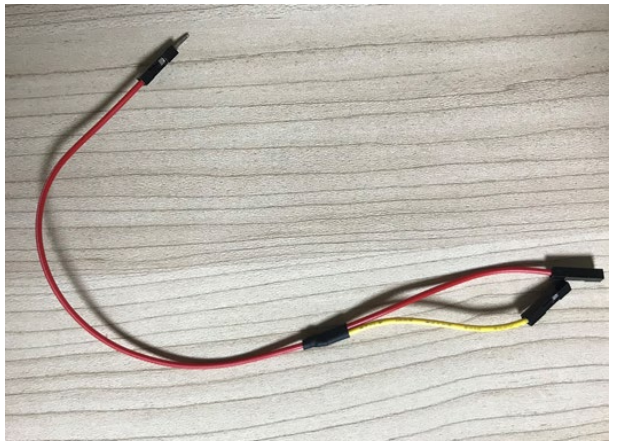
ピンヘッダーではんだ付け

接近センサーADPS-9930をはんだ付け完了しました。

Spresenseメインボード + APDS-9930近接センサー 物理接続図



※Spresenseメインボードピンレイアウト図を参照してください

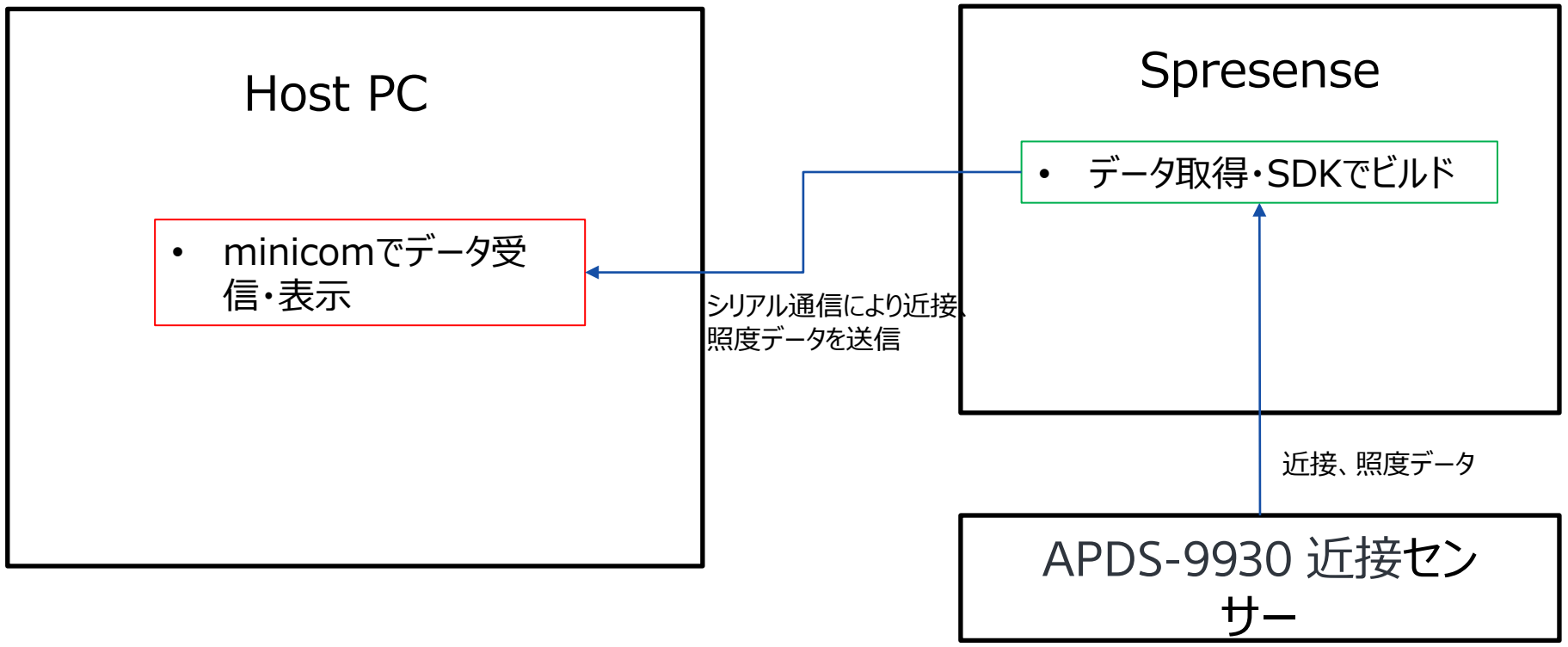


※近接センサーを使う場合はVL(赤外線LED電源)を3.3Vに接続しないと、近接データが取れないため、例としては電源ジャンパーとVLジャンパーをショットして、Spresenseの3.3Vに接続する。

APDS-9930近接センサー	Spresense
GND	GND
VCC	3.3 V
VL	3.3 V
SCL	I2C SCL
SDA	I2C SDA

Spresense + 近接センサー + Host PC システム構成図

1. 概略図



Spresense SDK環境で近接センサー動作確認

ビルド&ロード手順
ここではコマンドラインによるビルド手順を示します。

1.sdk ディレクトリへ移動します。
build-env.sh スクリプトを読み込むことで、config.py ツールの Tab 補完機能が有効になります。

```
cd spresense/sdk
source tools/build-env.sh
```

2.SDK のコンフィグレーションとビルドを行います。
引数に examples/proximity を指定してコンフィグレーションを実行します。
ビルドに成功すると sdk フォルダ直下に nuttx.spk ファイルが生成されます。

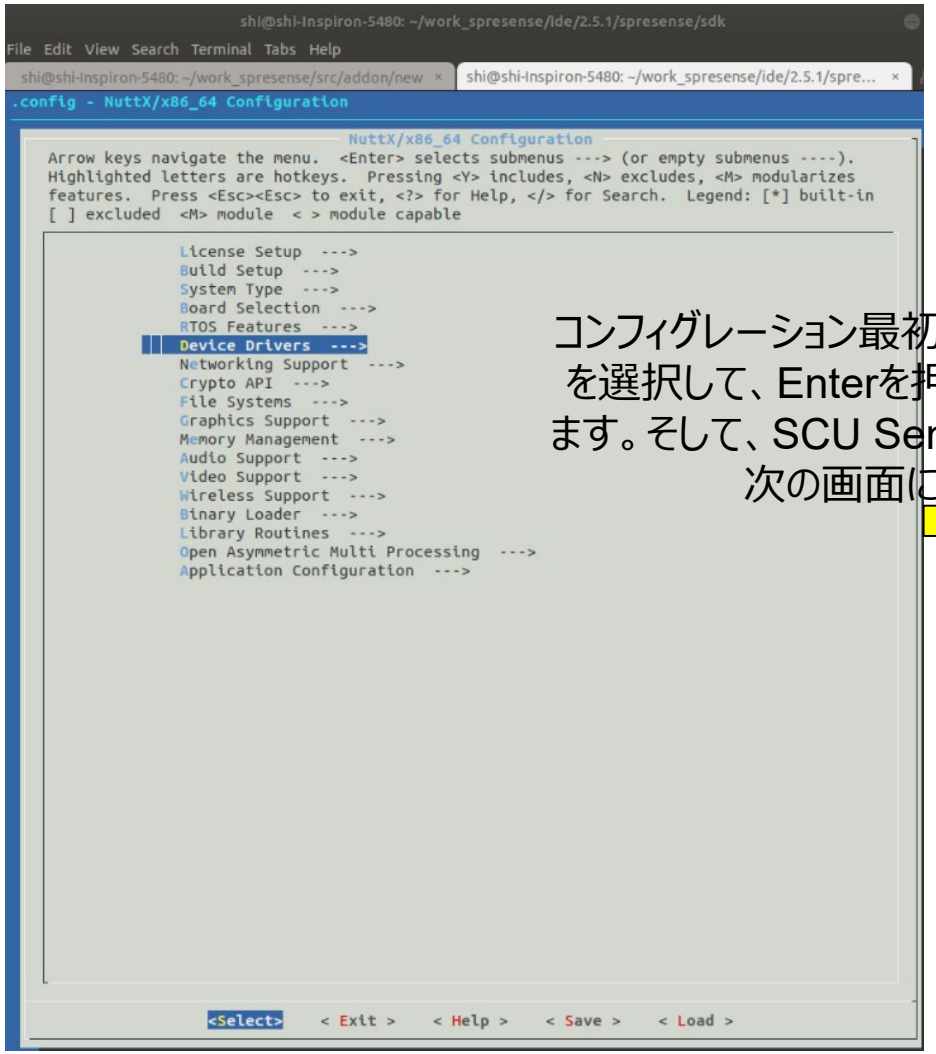
```
tools/config.py examples/proximity -m
```

```
//照度データを確認する場合、下記のコンフィグレーションを実行します
tools/config.py examples/light -m
```

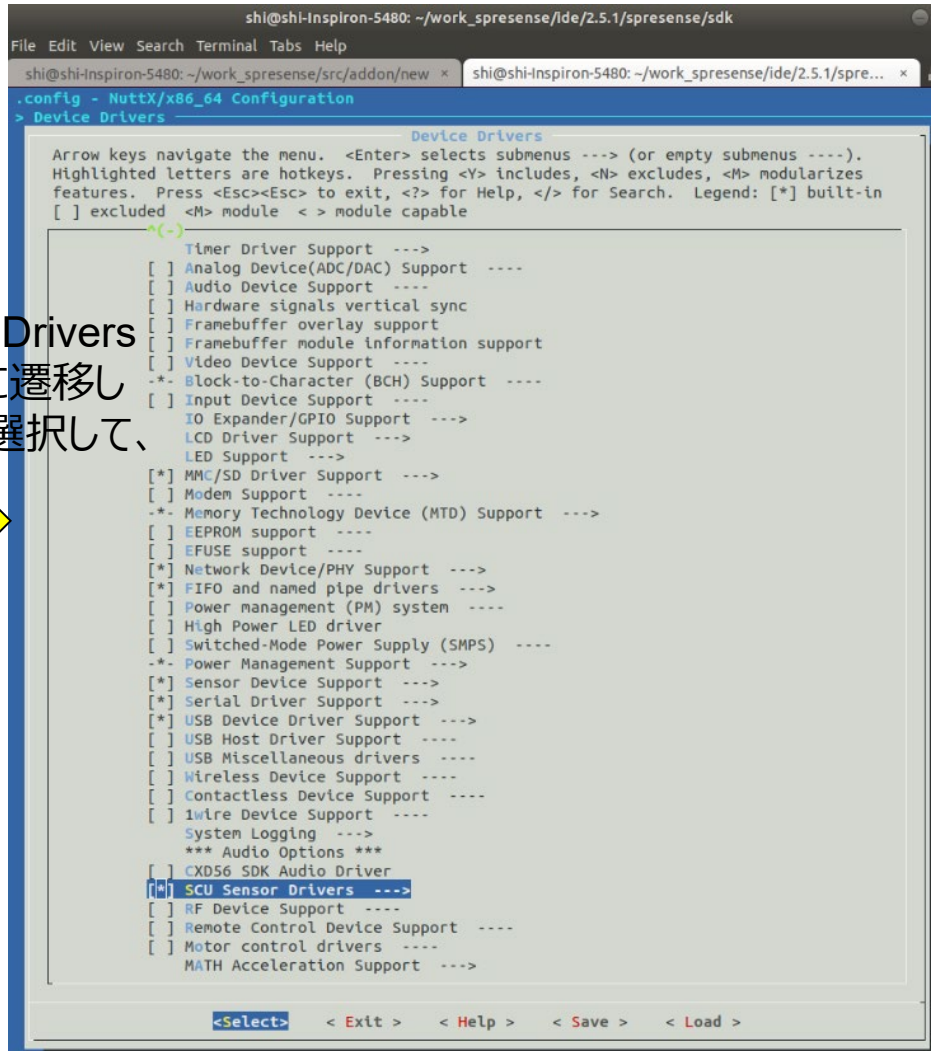
次ページのコンフィグメニュー画面が開きます

[Refs]
https://developer.sony.com/develop/spresense/docs/sdk_tutorials_ja.html

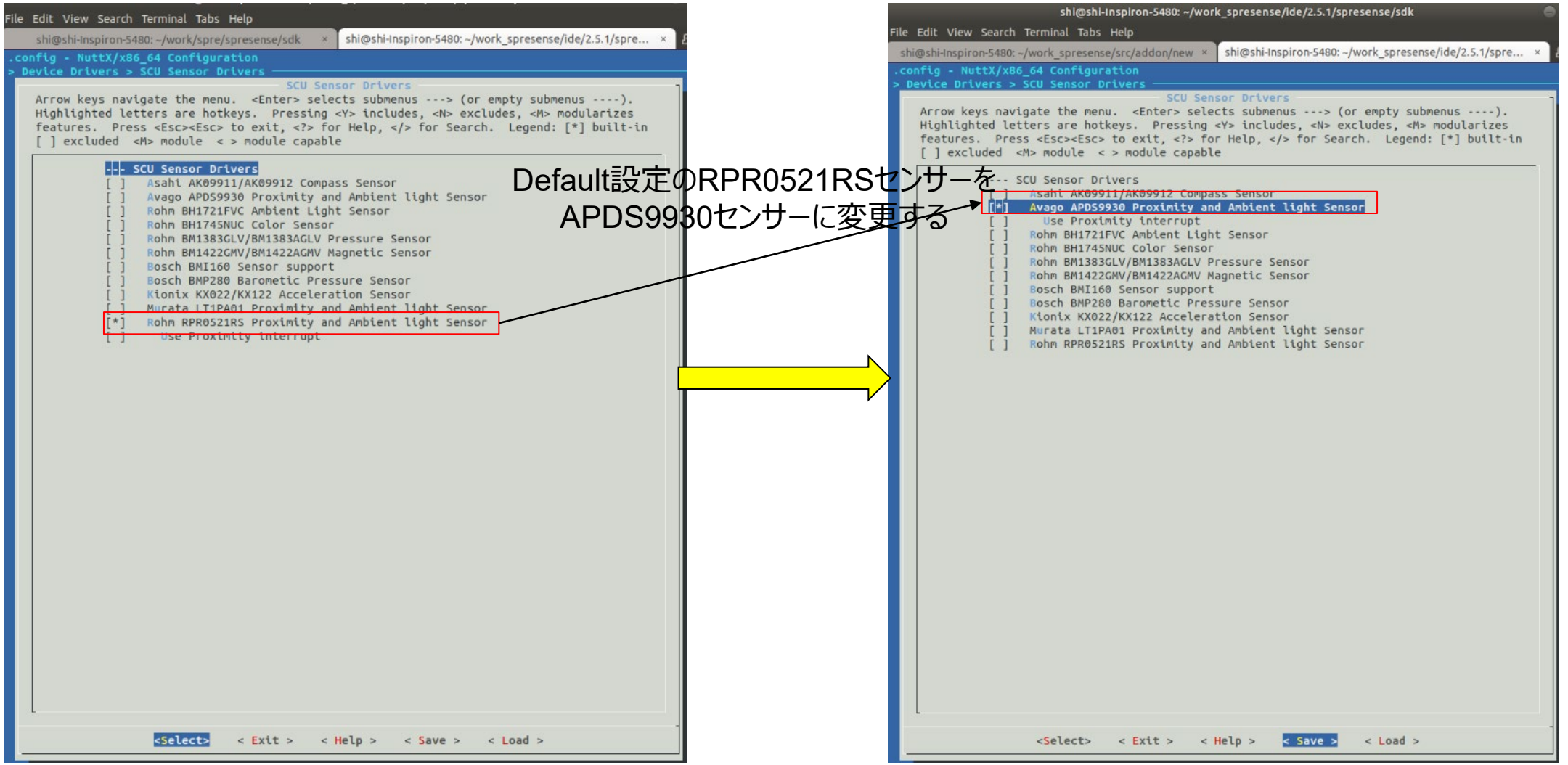
Spresense SDK環境で近接センサー動作確認



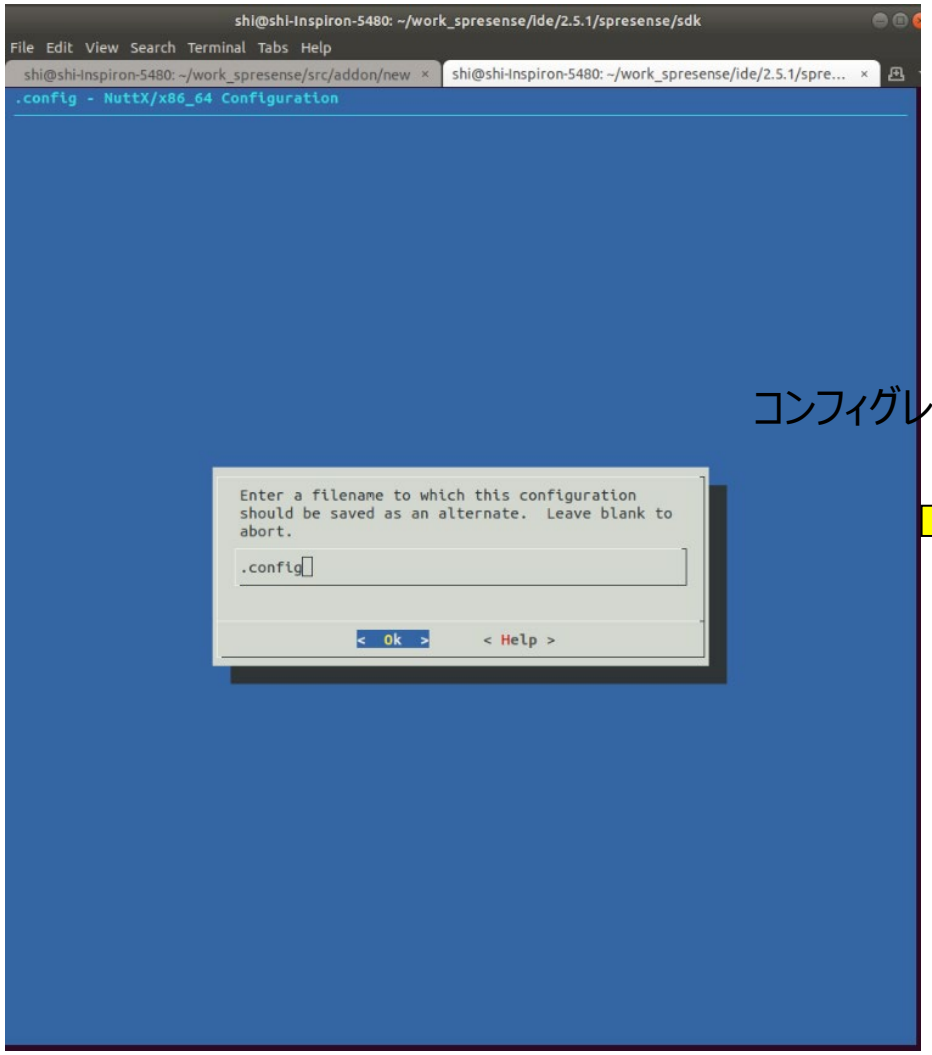
コンフィグレーション最初画面のDevice Drivers を選択して、Enterを押して次の画面に遷移します。そして、SCU Sensor Driversを選択して、次の画面に遷移します。



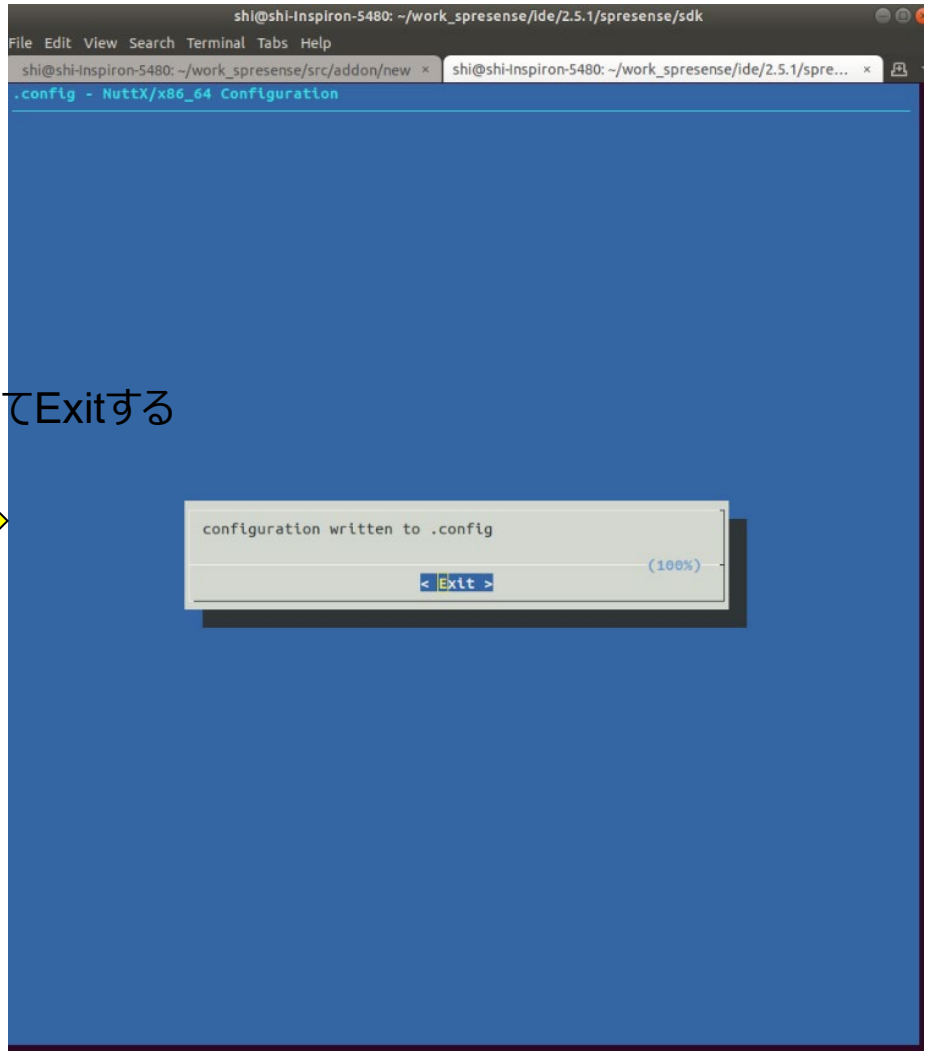
Spresense SDK環境で近接センサー動作確認



Spresense SDK環境で近接センサー動作確認



コンフィグレーションを保存してExitする



Spresense SDK環境で近接センサー動作確認

コンフィグレーションが終わったら、次はmakeでビルドします。

```
make
```

3. nuttx.spk を Spresense ボードへ書き込みます。

この例では シリアルポートとして /dev/ttyUSB0 を、書き込み速度の baudrate に 500000 bps を設定しています。お使いの環境に合わせて変更してください。

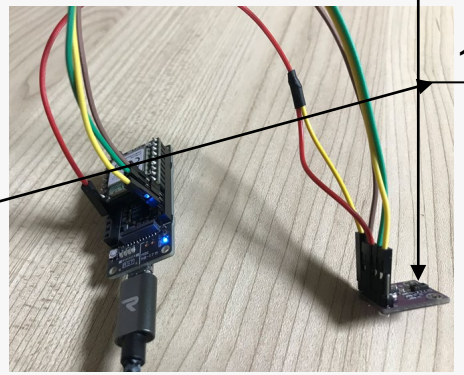
```
tools/flash.sh -c /dev/ttyUSB0 -b 500000 nuttx.spk
```

[Refs]
https://developer.sony.com/develop/spresense/docs/sdk_tutorials_ja.html

Spresense SDK環境で近接センサー動作確認

動作確認ログ：

```
NuttShell (NSH) nsh> proximity
SCU Timestamp: 2272.12182 (1.00001)
[0] PS:0000
[1] PS:0000
[2] PS:0000
[3] PS:000d
[4] PS:0007
[5] PS:001a
[6] PS:0026
[7] PS:0035
SCU Timestamp: 2273.12180 (0.32766)
[0] PS:0036
[1] PS:0033
[2] PS:0040
[3] PS:004b
[4] PS:005d
[5] PS:006c
[6] PS:0098
[7] PS:00e1
SCU Timestamp: 2274.12181 (1.00001)
[0] PS:00f6
[1] PS:0107
[2] PS:0113
[3] PS:0129
[4] PS:0154
[5] PS:016f
[6] PS:01d0
[7] PS:0245
SCU Timestamp: 2275.12181 (1.00000)
[0] PS:0297
[1] PS:02dd
[2] PS:02f0
[3] PS:02f8
[4] PS:037f
[5] PS:03ff
[6] PS:03ff
[7] PS:03ff
```



4cm

3cm

2cm

1cm

対象物は近接センサーの赤外線LEDと
およそ4cmぐらい近づくと、最小値0以上変化する

対象物は近接センサーの赤外線LEDと近づ
けば近いほど値は大きくなる

対象物は近接センサーの赤外線LEDと
およそ1cmぐらい近づくと、最大値0x3ffになる

※サンプルに下記のコメントがあります。
SCU(Sensor Control Unit)は1秒ごとにAPPに信号データを送信します。

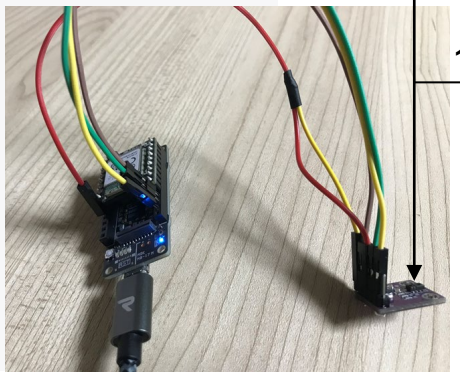
```
//SCU may send signal every 1 second.
```

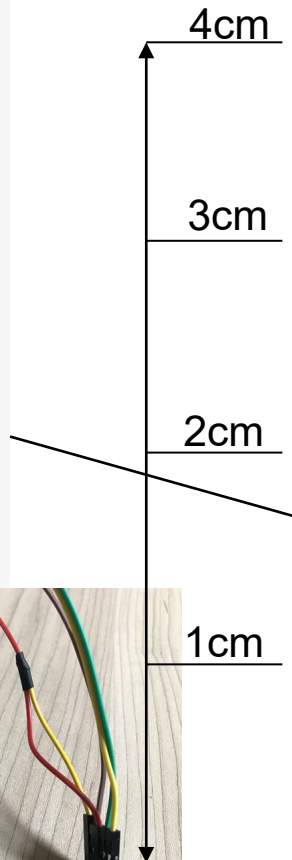
【動作確認サンプル】
https://github.com/sonydevworld/spresense/tree/master/examples/proximity

proximityサンプルのログ確認

動作確認ログ：

```
NuttShell (NSH) nsh> proximity
SCU Timestamp: 2272.12182 (1.00001)
[0] PS:0000
[1] PS:0000
[2] PS:0000
[3] PS:000d
[4] PS:0007
[5] PS:001a
[6] PS:0026
[7] PS:0035
SCU Timestamp: 2273.12180 (0.32766)
[0] PS:0036
[1] PS:0033
[2] PS:0040
[3] PS:004b
[4] PS:005d
[5] PS:006c
[6] PS:0098
[7] PS:00e1
SCU Timestamp: 2274.12181 (1.00001)
[0] PS:00f6
[1] PS:0107
[2] PS:0113
[3] PS:0129
[4] PS:0154
[5] PS:016f
[6] PS:01d0
[7] PS:0245
SCU Timestamp: 2275.12181 (1.00000)
[0] PS:0297
[1] PS:02dd
[2] PS:02f0
[3] PS:02f8
[4] PS:037f
[5] PS:03ff
[6] PS:03ff
[7] PS:03ff
```





[Refs：datasheetのP5]
https://www.mouser.jp/datasheet/2/678/av02-3190en_ds_apds-9930_2015-11-13-1828481.pdf

Proximity Characteristics, V_{DD} = 3 V, T_A = 25° C, PGAIN = 1, PEN = 1 (unless otherwise noted)

Parameter	Min	Typ	Max	Units	Test Conditions
I_{DD} Supply current – LDR Pulse On		3		mA	
ADC Conversion Time Step Size	2.58	2.73	2.9	ms	PTIME = 0xff
ADC Number of Integration Steps		1		steps	PTIME = 0xff
Full Scale ADC Counts			1023	counts	PTIME = 0xff
Proximity IR LED Pulse Count	0		255	pulses	
Proximity Pulse Period		16.0		µs	
Proximity Pulse – LED On Time		7.3		µs	
Proximity LED Drive		100		mA	PDRIVE = 0
		50			PDRIVE = 1
		25			PDRIVE = 2

I_{SINK} Sink current @ 600 mV, LDR Pin

最大値1023=0x3ff

Figure 10. Proximity Detection

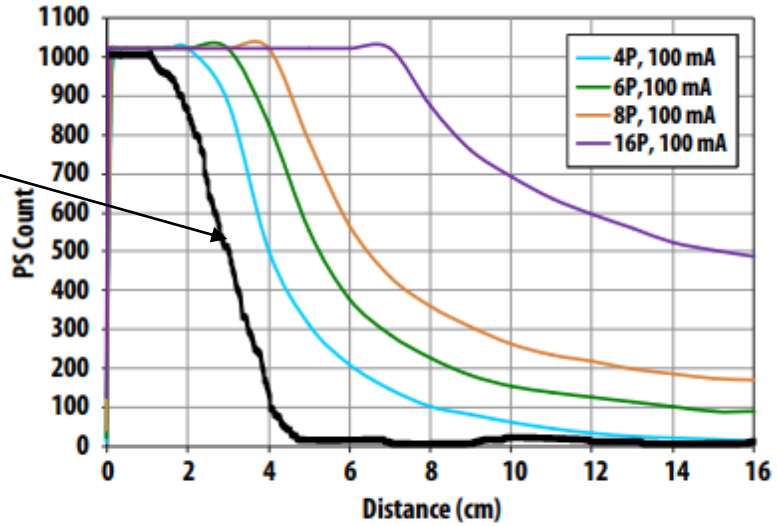


Figure 11a. PS Output vs. Distance at 100 mA, PGAIN = 10, at various Pulse Count. No glass in front of the module, 18% Kodak Grey Card.

[Refs： datasheetのP11]
https://www.mouser.jp/datasheet/2/678/av02-3190en_ds_apds-9930_2015-11-13-1828481.pdf

サンプルのDefault設定は1P,100mAです。黒線(1P,100mA)は実際の測定値を参照して、作成した曲線です。P(パルス)は大きく設定すればするほど、遠距離検知をできるが、近距離検知できないことがわかる。

【動作確認サンプル】

https://github.com/sonydevworld/spresense/tree/master/examples/proximity

なぜPPLUSEが大きければ大きいほど距離が長く測定できるか(ノイズも大きくなる)

①近接検出は、内部 IR LED から対象物に反射された IR エネルギーの量で距離を決定する。

②図 9 は、LEDオンパルスの幅は 7.3 µs に固定され、周期は 16.0 µs です。そこで、近接駆動電流の設定に加えて、1～255 個の近接パルス (PPULSE) をプログラムで設定できます。近接パルスの数を決定するときは、信号はPPULSE に比例して増加しますが、ノイズは PPULSE の平方根で増加します。

Proximity Detection

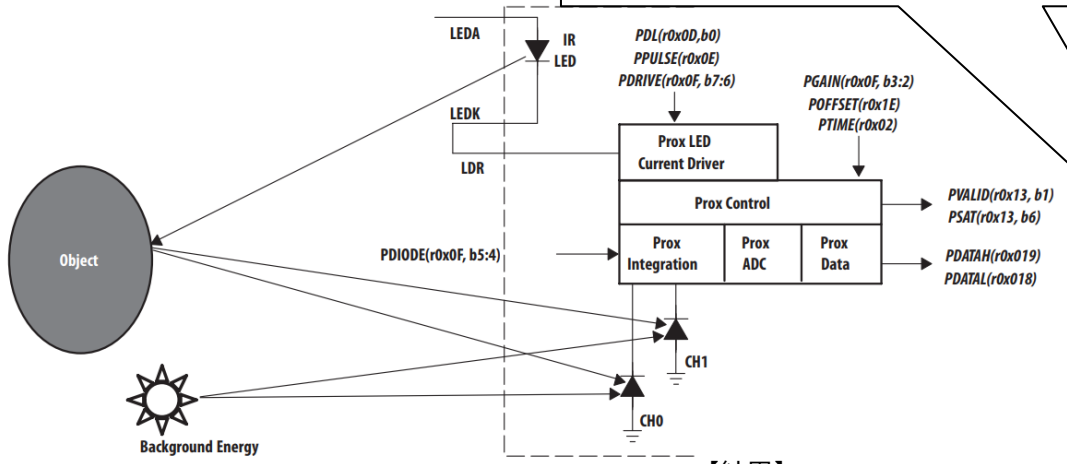


Figure 8. Proximity Detection

【結果】

PPLUSE値が小さく設定する場合

- 受け取り側は受光時間が短くて、遠くから反射光を受け取れないため、近距離測定する場合使います。

PPLUSE値が大きく設定する場合

- 受け取り側は受光時間が長くなって、遠くから反射光を受け取れるようになる。遠距離測定する場合使います。

Proximity detection is accomplished by measuring the amount of IR energy, from the internal IR LED, reflected off an object to determine its distance. The internal proximity IR LED is driven by the integrated proximity LED current driver as shown in Figure 8.

The LED current driver, output on the LDR terminal, provides a regulated current sink that eliminates the need for an external current limiting resistor. The combination of proximity LED drive strength (PDRIVE) and proximity drive level (PDL) determine the drive current. PDRIVE sets the drive current to 100 mA, 50 mA, 25 mA, or 12.5 mA when PDL is not asserted. However, when PDL is asserted, the drive current is reduced by a factor of 9.

Referring to the Detailed State Machine figure, the LED current driver pulses the IR LED as shown in Figure 9 during the Prox Accum state. Figure 9 also illustrates that the LED On pulse has a fixed width of 7.3 µs and period of 16.0 µs. So, in addition to setting the proximity drive current, 1 to 255 proximity pulses (PPULSE) can be programmed. When deciding on the number of proximity pulses, keep in mind that the signal increases proportionally to PPULSE, while noise increases by the square root of PPULSE.

Figure 8 illustrates light rays emitting from the internal IR LED, reflecting off an object, and being absorbed by the CH1 photodiodes. The proximity diode selector (PDIODE) selects Ch1 diode for a given proximity measurement. Note that PDIODE must be set for proximity detection to work.

Referring again to Figure 9, the reflected IR LED and the background energy is integrated during the LED On time, then during the LED Off time, the integrated background energy is subtracted from the LED On time energy, leaving the IR LED energy to accumulate from pulse to pulse. The proximity gain (PGAIN) determines the integration rate, which can be programmed to 1×, 2×, 4×, or 8× gain. At power up, PGAIN defaults to 1× gain, which is recommended for most applications. For reference, PGAIN equal to 4× is comparable to the APDS-9900's 1× gain setting. During LED On time integration, the proximity saturation bit in the Status register (0x13) will be set if the integrator saturates. This condition can occur if the proximity gain is set too high for the lighting conditions, such as in the presence of bright sunlight. Once asserted, PSAT will remain set until a special function proximity interrupt clear command is received from the host (see command register).

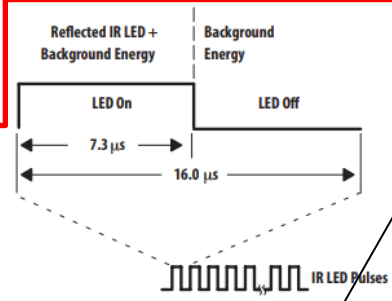


Figure 9. Proximity LED Current Driver Waveform

PGAINについて

はDefaultで1×で設定されています。これは、ほとんどのアプリケーションに推奨されます。PGAINを高く設定すると、PSATがセットされたまま変化しない可能性があります。

[Refs：datasheetのP10]

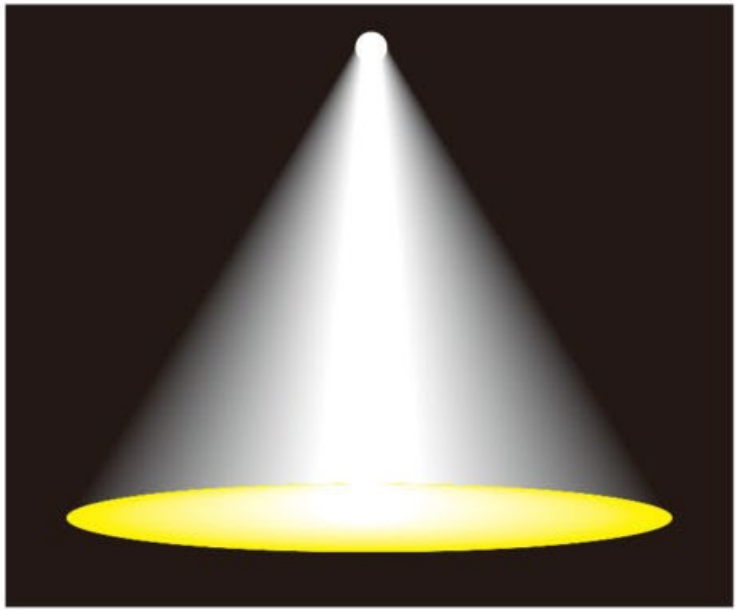
https://www.mouser.jp/datasheet/2/678/av02-3190en_ds_apds-9930_2015-11-13-1828481.pdf

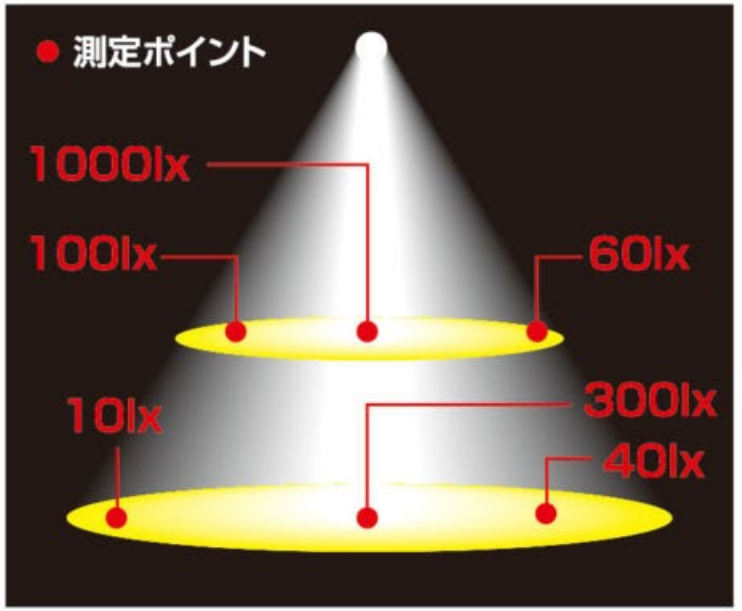
Spresense SDK環境で近接センサー動作確認

[Refs]
https://www.gentos.jp/blog/%E6%98%8E%E3%82%8B%E3%81%95%E3%81%AB%E3%81%A4%E3%81%84%E3%81%A6/

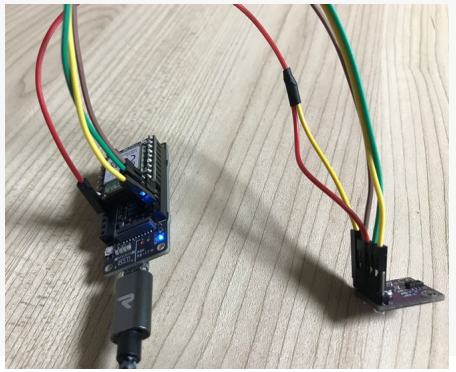
同様にexamples/light動作確認ログ：

```
nsh> light
Sensing start...
[0] 18[lx] Raw[0002 0000]
[1] 18[lx] Raw[0002 0000]
[2] 18[lx] Raw[0002 0000]
[3] 18[lx] Raw[0002 0000]
[4] 18[lx] Raw[0002 0000]
[5] 18[lx] Raw[0002 0000]
[6] 18[lx] Raw[0002 0000]
[7] 18[lx] Raw[0002 0000]
SCU Timestamp: 511.24803 (1.00000)
[0] 18[lx] Raw[0002 0000]
[1] 18[lx] Raw[0002 0000]
[2] 18[lx] Raw[0002 0000]
[3] 27[lx] Raw[0003 0000]
[4] 27[lx] Raw[0003 0000]
[5] 27[lx] Raw[0003 0000]
[6] 27[lx] Raw[0003 0000]
[7] 37[lx] Raw[0004 0000]
SCU Timestamp: 517.24802 (0.32767)
[0] 2599[lx] Raw[0161 0028]
[1] 2875[lx] Raw[0186 002c]
[2] 2943[lx] Raw[0191 002e]
[3] 2736[lx] Raw[0179 002d]
[4] 2773[lx] Raw[017d 002d]
[5] 3370[lx] Raw[01ca 0034]
[6] 2887[lx] Raw[018b 002e]
[7] 2117[lx] Raw[0124 0023]
```





光源は照度センサーに近づけば近いほど値は大きくなる
※照度データを確認する場合、VLピンを3.3Vに接続必要がない



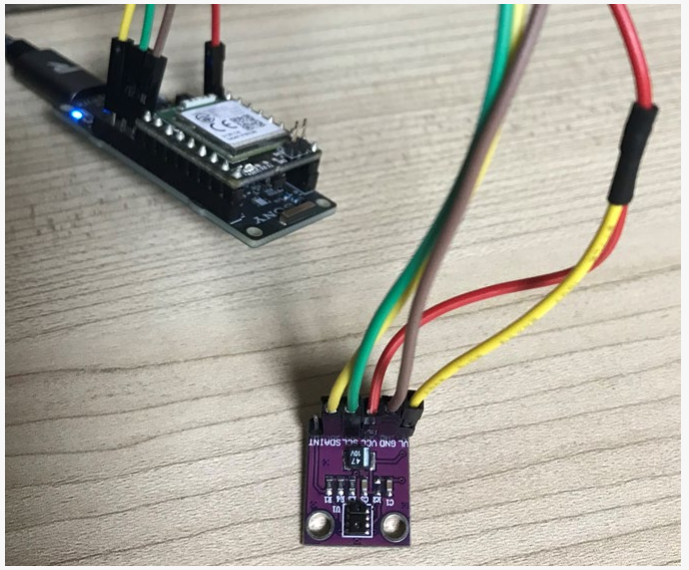
【動作確認サンプル】
https://github.com/sonydevworld/spresense/tree/master/examples/light

lightサンプルのログ確認

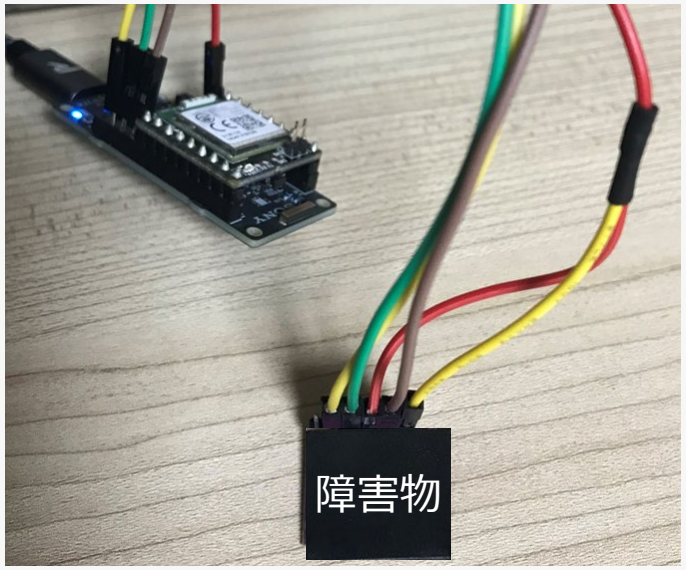
同様にexamples/light動作確認ログ：

```
nsh> light
Sensing start...
SCU Timestamp: 2871.04122 (1.00000)
[0] 95[lx] Raw[000e 0002]
[1] 95[lx] Raw[000e 0002]
[2] 95[lx] Raw[000e 0002]
[3] 95[lx] Raw[000e 0002]
[4] 95[lx] Raw[000e 0002]
[5] 95[lx] Raw[000e 0002]
[6] 95[lx] Raw[000e 0002]
[7] 95[lx] Raw[000e 0002]
SCU Timestamp: 2877.04122 (1.00001)
[0] 0[lx] Raw[0000 0000]
[1] 0[lx] Raw[0000 0000]
[2] 0[lx] Raw[0000 0000]
[3] 0[lx] Raw[0000 0000]
[4] 0[lx] Raw[0000 0000]
[5] 0[lx] Raw[0000 0000]
[6] 0[lx] Raw[0000 0000]
[7] 0[lx] Raw[0000 0000]
SCU Timestamp: 2878.04121 (0.32767)
[0] 0[lx] Raw[0000 0000]
[1] 0[lx] Raw[0000 0000]
[2] 0[lx] Raw[0000 0000]
[3] 9[lx] Raw[0001 0000]
[4] 75[lx] Raw[000a 0001]
[5] 95[lx] Raw[000e 0002]
[6] 95[lx] Raw[000e 0002]
[7] 95[lx] Raw[000e 0002]
SCU Timestamp: 2879.04121 (1.00000)
[0] 95[lx] Raw[000e 0002]
[1] 95[lx] Raw[000e 0002]
[2] 95[lx] Raw[000e 0002]
[3] 95[lx] Raw[000e 0002]
[4] 95[lx] Raw[000e 0002]
[5] 95[lx] Raw[000e 0002]
[6] 95[lx] Raw[000e 0002]
[7] 95[lx] Raw[000e 0002]
```

①環境室内LED灯点灯中：
95ルーメン



②右図通り、照度センサーに黒い障害物を被ると1秒後0ルーメンになる。



③また黒い障害物を外すと1秒後もとの95ルーメンに戻る
※環境は変わらない限り、安定な照度データが取れます。

※サンプルに下記のコメントがあります。
SCU(Sensor Control Unit)は1秒ごとにAPPに信号データを送信します。

```
//SCU may send signal every 1 second.
```

【動作確認サンプル】
https://github.com/sonydevworld/spresense/tree/master/examples/light