<Raspberry Pi 1・2・3の基本設定>

（1）Raspberry Pi財団から最新版のRasbianをダウンロードしてください。

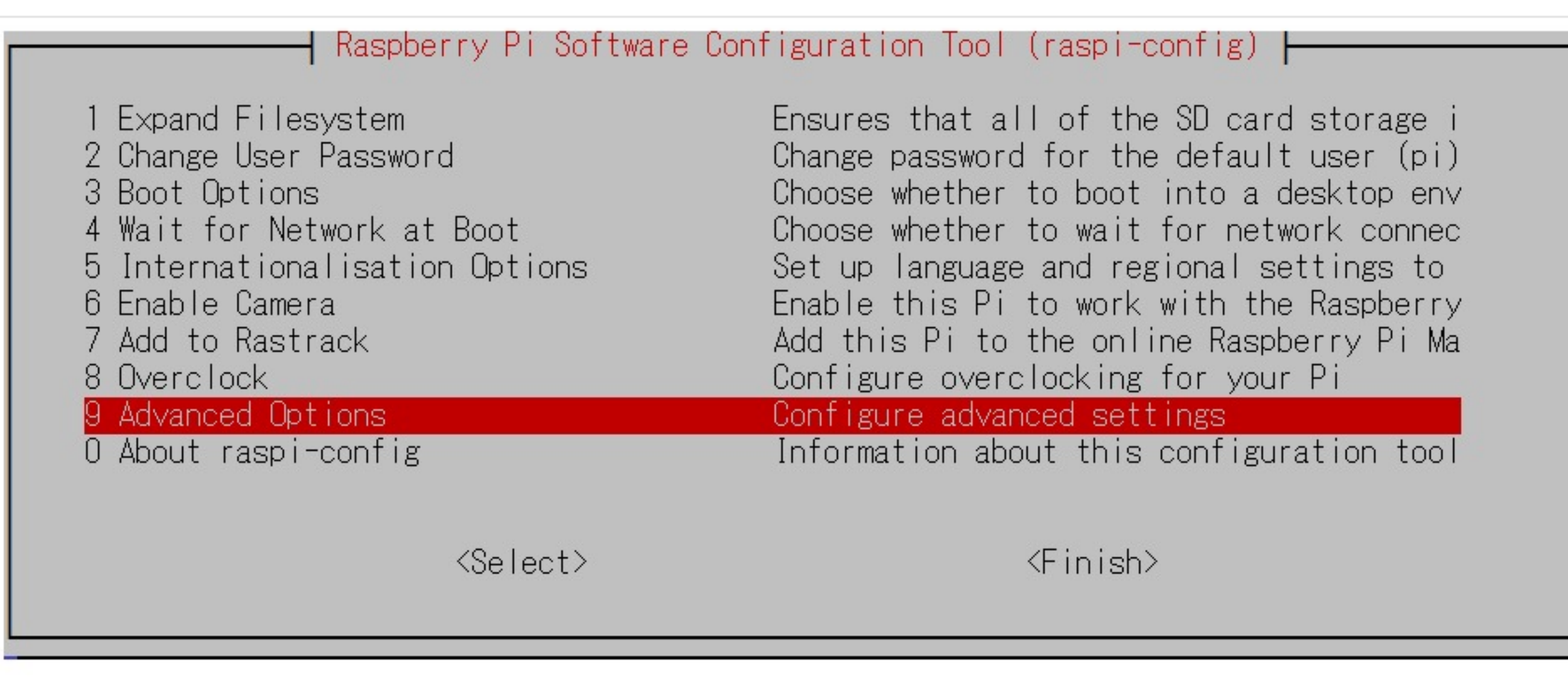
（2）Win32DiskImager等でイメージをSDカードに書き込んでください。

（3）HDMIディスプレイとキーボード、マウスをつないで、以下を設定します。起動したGUIで、ターミナルウィンドウを開き、以下を設定してください。

（4）初期イメージでは残りのディスク容量がとても少ないため、raspi-configコマンドで使用領域を拡張してください。

```
$ sudo raspi-config
```

起動したGUIを以下に示します。Expand Filesystemを選択し、領域を拡張してください。

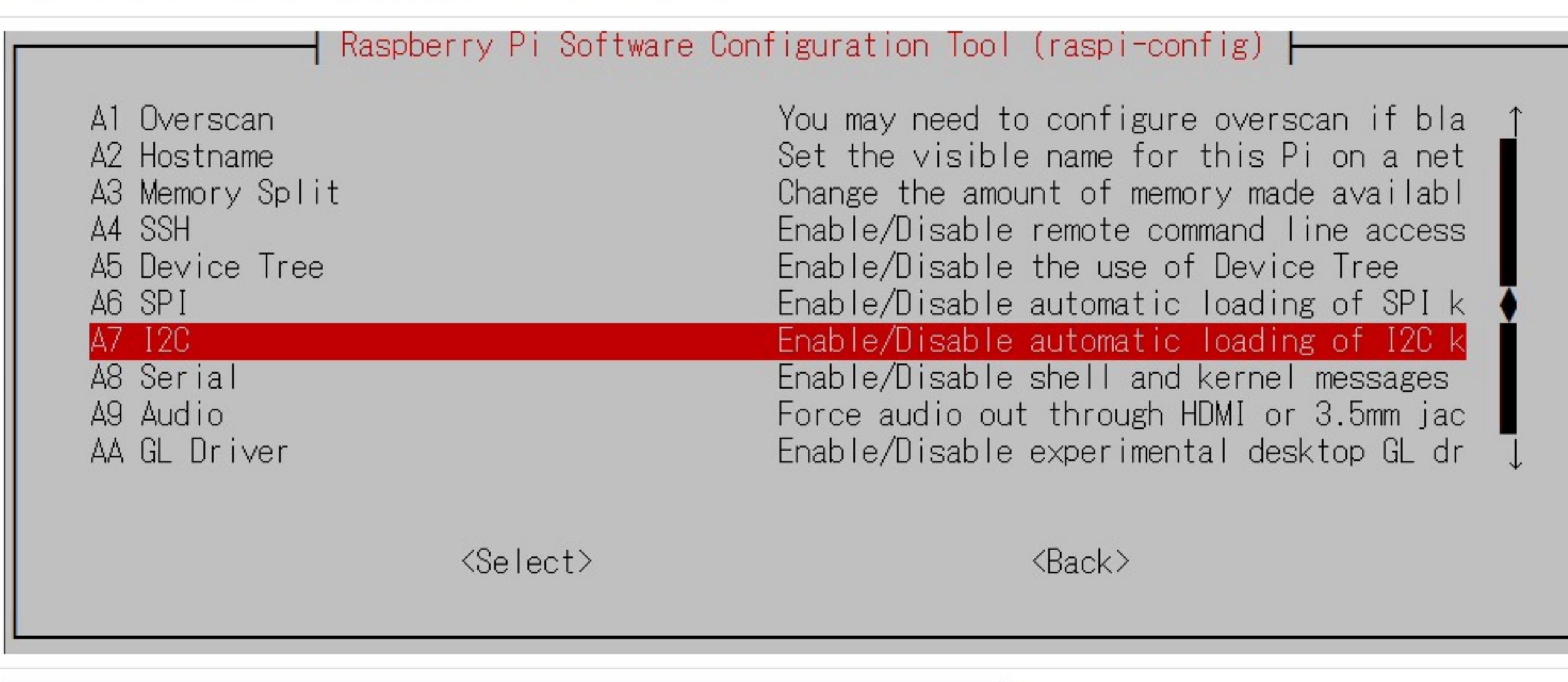


```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Expand Filesystem                 Ensures that all of the SD card storage i
2 Change User Password              Change password for the default user (pi)
3 Boot Options                      Choose whether to boot into a desktop env
4 Wait for Network at Boot          Choose whether to wait for network connec
5 Internationalisation Options      Set up language and regional settings to
6 Enable Camera                     Enable this Pi to work with the Raspberry
7 Add to Rastrack                   Add this Pi to the online Raspberry Pi Ma
8 Overclock                         Configure overclocking for your Pi
9 Advanced Options                  Configure advanced settings
0 About raspi-config                Information about this configuration tool

                <Select>                        <Finish>
```

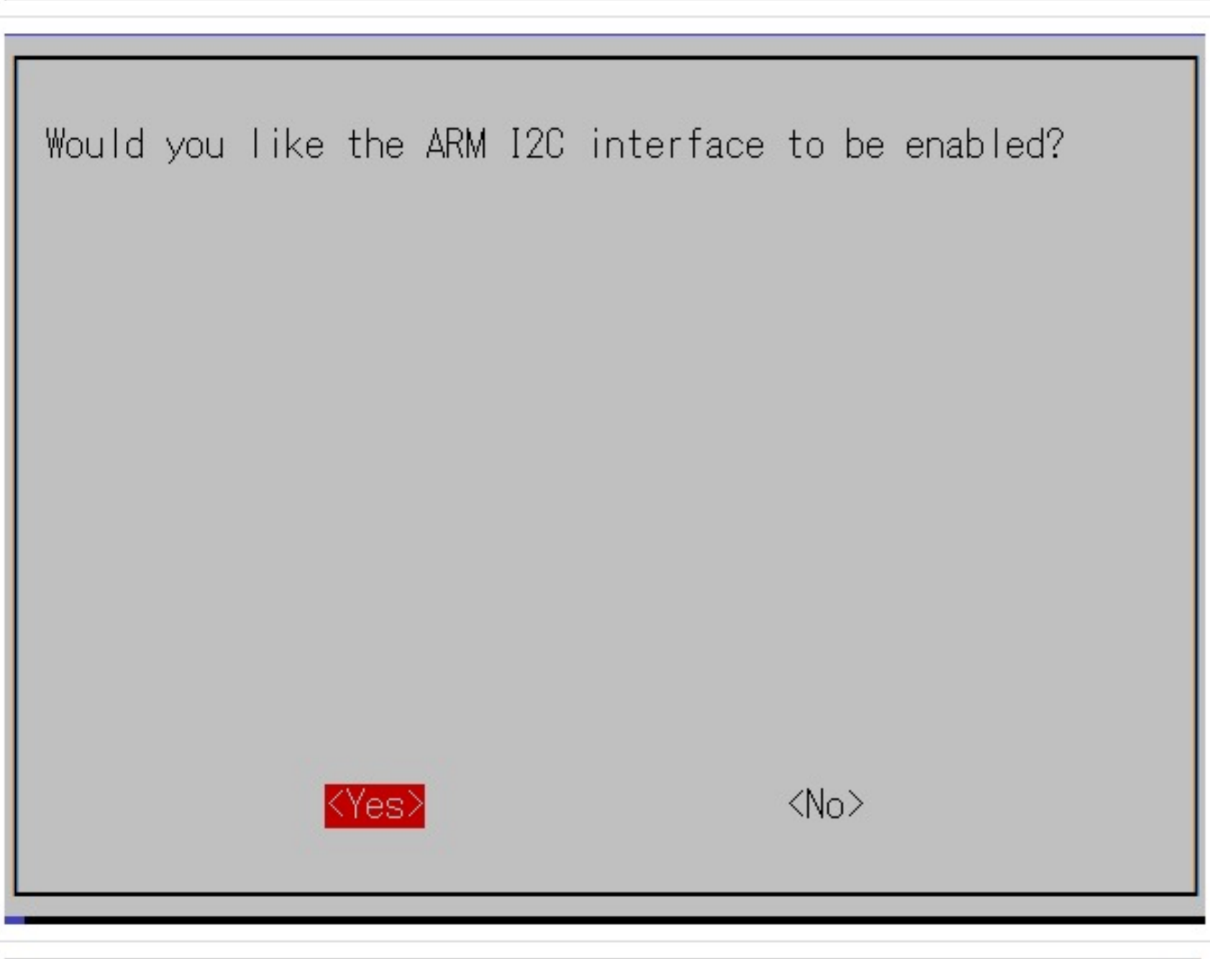
（5）パッケージとファームウェアを以下のコマンドで最新にしてください。

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo rpi-update
$ sudo reboot
```

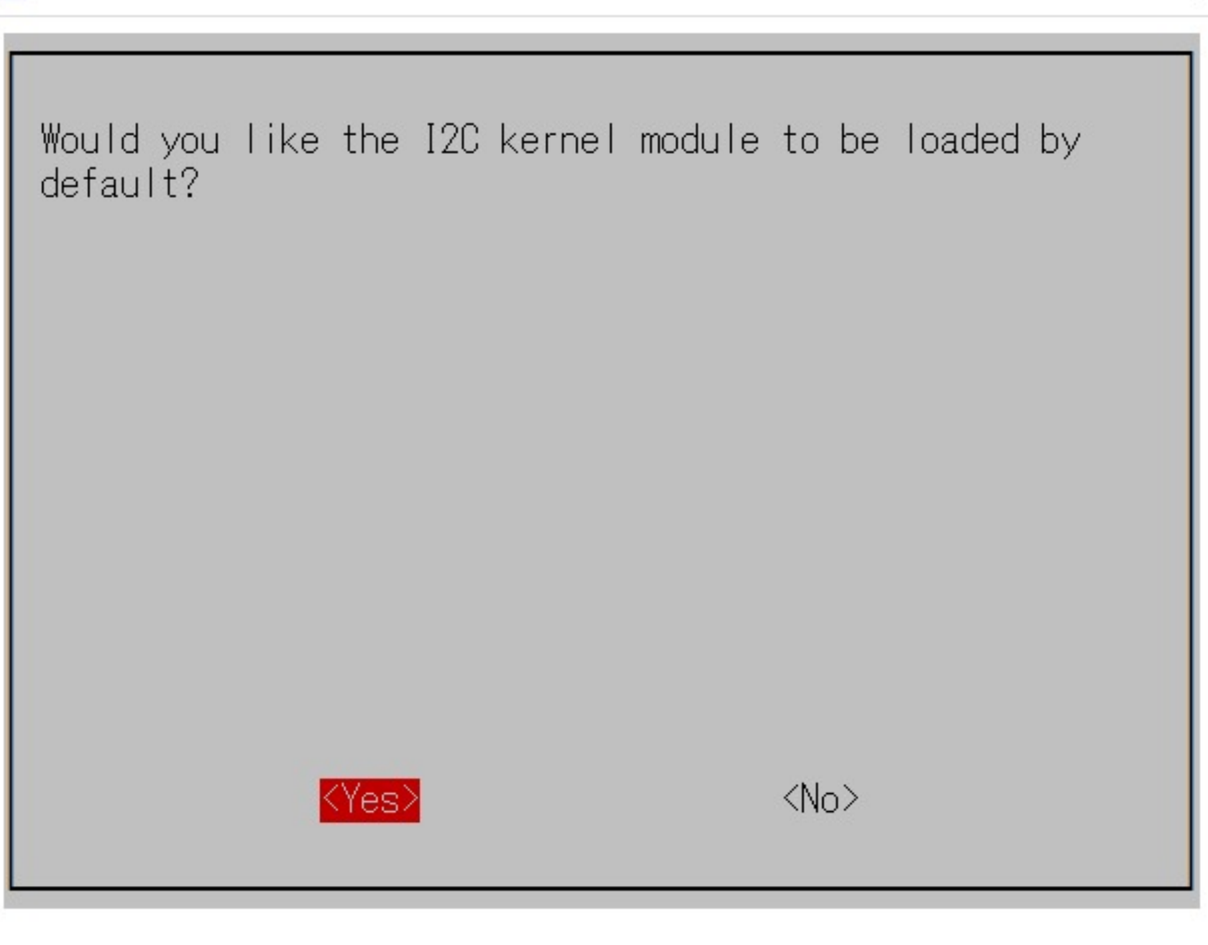
（6）raspi-configでi2cを有効にします。
「Interface options」からI2Cを選び、Enableにします。



```
Raspberry Pi Software Configuration Tool (raspi-config)

A1 Overscan                         You may need to configure overscan if bla
A2 Hostname                         Set the visible name for this Pi on a net
A3 Memory Split                     Change the amount of memory made availabl
A4 SSH                              Enable/Disable remote command line access
A5 Device Tree                      Enable/Disable the use of Device Tree
A6 SPI                              Enable/Disable automatic loading of SPI k
A7 I2C                              Enable/Disable automatic loading of I2C k
A8 Serial                           Enable/Disable shell and kernel messages
A9 Audio                            Force audio out through HDMI or 3.5mm jac
AA GL Driver                        Enable/Disable experimental desktop GL dr

                <Select>                        <Back>
```

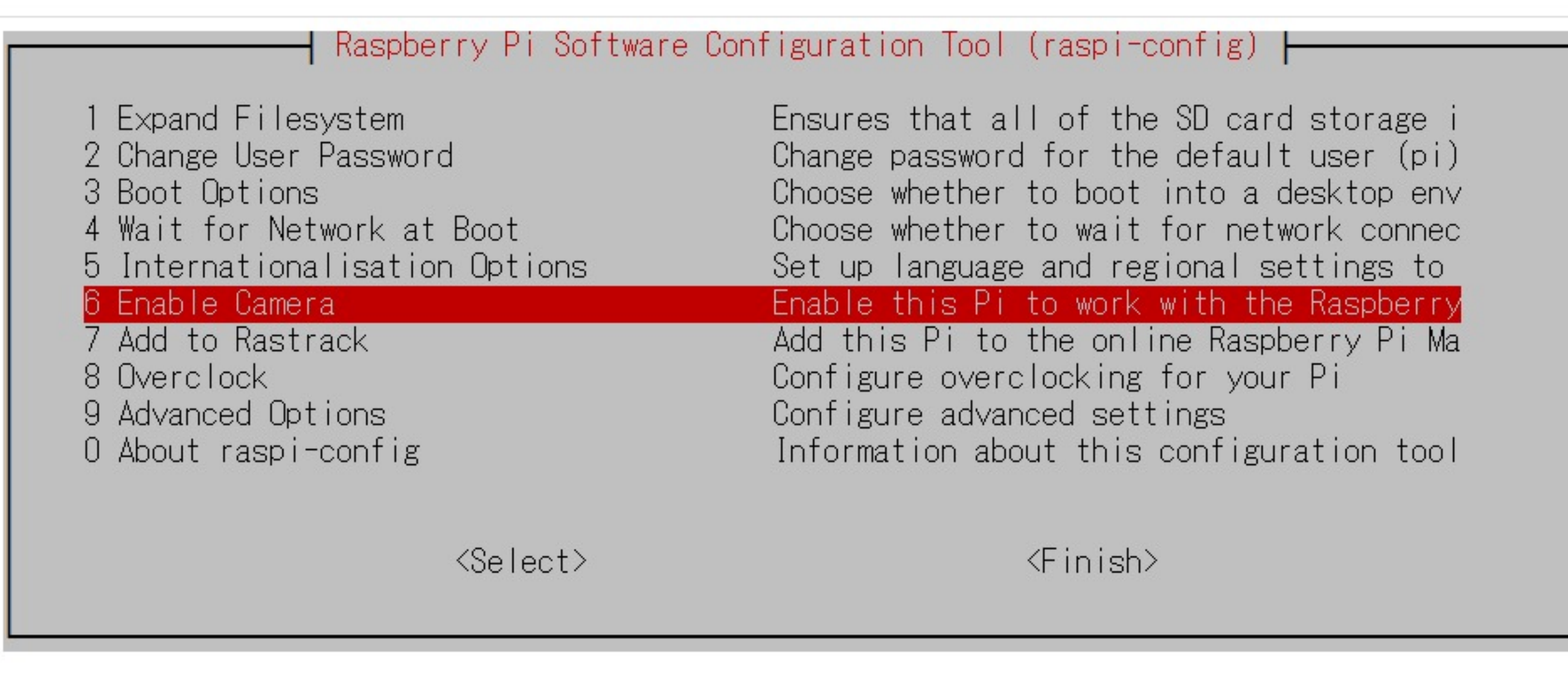


Would you like the ARM I2C interface to be enabled?



Would you like the I2C kernel module to be loaded by default?

（7）raspi-configでCameraを有効にします。
「Interface Options」から「Camera」を選び、カメラ機能をイネーブルします。設定の最後に再起動が促されますので再起動してください。



```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Expand Filesystem                 Ensures that all of the SD card storage i
2 Change User Password              Change password for the default user (pi)
3 Boot Options                      Choose whether to boot into a desktop env
4 Wait for Network at Boot          Choose whether to wait for network connec
5 Internationalisation Options      Set up language and regional settings to
6 Enable Camera                     Enable this Pi to work with the Raspberry
7 Add to Rastrack                   Add this Pi to the online Raspberry Pi Ma
8 Overclock                         Configure overclocking for your Pi
9 Advanced Options                  Configure advanced settings
0 About raspi-config                Information about this configuration tool

                <Select>                        <Finish>
```

<Raspberry Pi 1・2での設定>

これ以降はRaspberry Pi2での設定です。他のプラットフォームの場合は、読み飛ばしてください。

（1）VideoCoreのI2Cを有効にします。CSI-2のインタフェースはVideoCoreと呼ばれているRaspberry PiのGPUに接続されています。CCIとして用いるI2Cインタフェースも同様にVideoCoreのものを用います。上記のraspi-configコマンドでのI2CのアクティベーションはARMチップの側のI2Cを有効にしているだけなので、VideoCoreのI2Cを明示的に有効にします。
以下のコマンドで/boot/config.txtを開いてください。

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

以下の1行をファイルの最終行に追加してください。

```
dtparam=i2c_vc=on
```

設定ができたら再起動してください。

```
$ sudo reboot
```

Raspberry Pi 2に関しては、ここまでの設定で準備ができました。

<Raspberry Pi 3での設定>

以下はRaspberry Pi 3専用の設定です。他のRaspberry Piのバージョンの設定は行わず、ここに示された方法のみを実行してください。
Raspberry Pi 3の場合、以下の1行を/boot/config.txtファイルの最後に追加するだけです。

```
dtoverlay=i2c1-bcm2708,sda1_pin=44,scl1_pin=45,pin_func=6
```

注意）dtparam=i2c_vc=onが記述されている場合はコメントアウトしてください（書いてあっても動くと思いますが、RasPi3では意味ありませんので念のため、書かない方が良いでしょう）。また、この1行を追加したSDカードをRaspberry Pi 2で用いるとUSB機器の動作が不安定になりますので、Raspberry Pi 2で動作させる場合は上記のRaspberry Pi2専用の変更のみを行ってください。

以下のコマンドで/boot/config.txtを開いて、ファイルの最終行に追加してください。

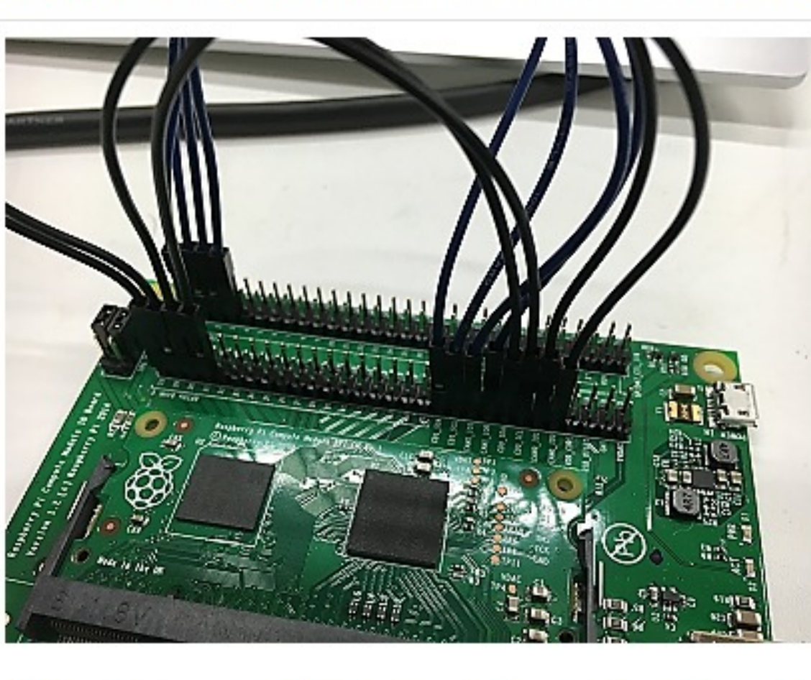
```
$ sudo nano /boot/config.txt
```

<Rasberry Pi Compute Moduleでの設定>

現在のSSPライブラリでは、CAM1ポートでの2レーン、4レーンでのCSI-2インタフェースが利用できます。

（1）Raspberry Pi財団の「FLASHING THE COMPUTE MODULE EMMC」のページを参考に、Flashに最新のRasbianイメージを書き込んでください。

（2）CAM0、CAM1のジャンパ配線をしてください。
Raspberry Pi財団の「ATTACHING A RASPBERRY PI CAMERA TO THE COMPUTE MODULE IO BOARD」のページに詳しく書いてあります。
正しくつながると、
CD1_SDA→GPIO0
CD1_SCL→GPIO1
CAM1_IO1→GPIO2
CAM1_IO0→GPIO3
となるように接続されます。以下の図を参考にしてみてください。



（3）CAM1に22ピンフラットケーブルでイメージセンサーモジュールを接続し、電源を入れてください。

（4）下記コマンドでCompute Module用のピン設定ファイルを/bootディレクトリに書き込んでください。

```
$ sudo wget http://goo.gl/lozvZB -O /boot/dt-blob.bin
```

以降は、上記の<Raspberry Pi 1・2での設定>に同じですので、それに従って設定してください。

<Rasberry Pi Compute Module 3での設定>

Compute Moduleの場合とほぼ同じですが、デバイスツリーの変更などが異なっていますので、以下の手順どおり設定してみてください。

（1）CAM0、CAM1のジャンパ配線をしてください。

CD1_SDA → GPIO0
CD1_SCL → GPIO1
CAM1_IO1 → GPIO2
CAM1_IO0 → GPIO3
CD0_SDA → GPIO28
CD0_SCL → GPIO29
CAM0_IO1 → GPIO30
CAM0_IO0 → GPIO31

注意）SSPは現在のところCAM1をサポートしていますが、将来はCAM0とCAM1が選択できるようになりますので、この配線に従って全てを配線してください。ここで書かれている設定をすると、raspistillもcsオプションを使ってCAM0、CAM1を選択できるようになります。RaspberryPi財団の手順どおり行ってもraspistillで双方のCAMを認識してくれませんのでご注意ください。

（2）wiringPi、ファームウェアをアップデートする。
以下のコマンドでアップデート作業を行ってください。最新のSDカードイメージの場合は必要ありません。

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

（3）２つのカメラをサポートするデバイスツリーをインストールする。
Raspberry Pi財団の「ATTACHING A RASPBERRY PI CAMERA TO THE COMPUTE MODULE IO BOARD」のページの最後にある「Enable both cameras」のデバイスツリーファイルをインストールします。以下のように行ってください。

```
$ sudo apt-get install device-tree-compiler
$ wget https://www.raspberrypi.org/documentation/hardware/computemodule/dt-blob-dualcam.dts
$ sudo dtc -I dts -O dtb -o /boot/dt-blob.bin dt-blob-dualcam.dts
```

（4）/boot/config.txtに設定を追加
/boot/config.txtの一番最後に以下の２行を追加してください。

```
dtoverlay=i2c1-bcm2708,sda1_pin=00,scl1_pin=01,pin_func=6
dtoverlay=i2c0-bcm2708,sda0_pin=28,scl0_pin=29,pin_func=6
```

<全てのRaspberry Piで共通の設定>

最初に、HexaVisionControlライブラリを設定します。
SSPライブラリのバージョン1.10から、HexaVision Monoがサポートされています。SSPライブラリがこれを参照するため、かならず設定しなければなりません。

（1）HexaVisionControlライブラリのパッケージをダウンロードのページからダウンロードし、tarコマンドで以下のように展開してください。

```
$ tar zxvf HexaVisionControl-0.92.tar.gz
```

（2）以下のコマンドで、libssp.confを作成します。

```
$ sudo nano /etc/ld.so.conf.d/libhexavision_ctl.conf
```

このファイルの先頭にHexaVisionControlライブラリのパスを追加します。例えば、「/home/ユーザ名/HexaVision」にHexaVisionControlライブラリを展開した場合は、

/home/ユーザ名/HexaVision/HexaVisionControl/バージョン/lib

の1行を追加します。

次に、SSPライブラリを展開します。

（3）SSPライブラリのパッケージをダウンロードのページからダウンロードし、tarコマンドで以下のように展開してください。

```
$ tar zxvf libssp-1.10.tar.gz
```

（4）SSPライブラリのlibディレクトリのパスをライブラリのサーチパスに追加します。SSPライブラリはダイナミックリンクライブラリで提供されていますので、それを自動的に探せるように、ldconfigで設定します。

まず、以下のコマンドで、libssp.confを作成します。

```
$ sudo nano /etc/ld.so.conf.d/libssp.conf
```

このファイルの先頭にSSPライブラリのパスを追加します。例えば、「/home/ユーザ名/」にSSPライブラリを展開した場合は、

/home/ユーザ名/libssp/バージョン/lib

の1行を追加します。

（5）上記のライブラリパス設定が終ったら、

```
$ sudo ldconfig
```

を実行してください。

<サンプルを実行する>

設定ができたら、SSPライブラリのsampleディレクトリにあるテストプログラムを実行してみましょう。
イメージセンサーのプロファイルXMLファイルを「ダウンロードページ」からダウンロードしてみてください。

（注意）センサープロファイルはSSPライブラリのsampleディレクトリにも保存されています。バージョン1.10以降はHexaVision Monoをサポートするため、そのセンサープロファイルも収録されています。バージョン1.10以降は、SSPEntryProVer0.9以降のもので編集してください。

（1）スチルのテスト：stillsample
このサンプルは画面出力がなくても（XやHDMI出力がなくても）、コマンドラインからRAWイメージを取得できるサンプルです。
コマンドラインから、

```
$ sudo stillsample センサープロファイルXML 保存ファイル
```

を実行します。
保存されたファイルは10ビットのRAWイメージです。このイメージを見るにはRAWViewerを使います。RAWViewerは「ダウンロードページ」からダウンロードしてみてください。
以下のようなRAWイメージが取得できます。この例ではIMX219（1280x720 60Hz）のセンサープロファイルXMLを用いています。

（2）リアルタイム映像表示サンプル：viewersampleGain
このサンプルでは、X Windowの起動が必要です。OpenCVでセンサーからのイメージをリアルタイムに確認できるサンプルです。このサンプル実行には、OpenCVの開発用パッケージが必要なりますので、下記のコマンドで、OpenCVをインストールしてください。

```
$ sudo apt-get install libopencv-dev
```

HDMIで画面出力を出して、ターミナルWindowで以下のコマンドラインを実行してください。

```
$ sudo viewersampleGain センサープロファイルXML センサーモデル
```

センサーモデルは現在のところ、IMX219、IMX408、IU233をサポートしています。ゲインと露光（シャッタースピード）のレジスタを変更できます。カメラ画像の出ているウィンドウで、「g」キーを押すと、ゲイン変更モード、「s」キーを押すと、露光変更モードになります。それぞれのモードで、「+」を押すと現行モードのレジスタ値をインクリメントし、「-」を押すとディクリメントします。また、「0」を押すとガンマ補正のイネーブル/ディスエーブルをトグルします。デフォルトでは、ガンマ補正は用いないモードになっています。

<Compute Module 3 マルチカメラ　サンプルを実行する>

（1）viewersampleGainMultiCamサンプル
このサンプルでは、別々のプロセスで異なるカメラ番号を指定して、別々のOpenCVのWindowを開き、それぞれのイメージセンサからのフレームを描画します。
例えば、以下の2つのコマンドを異なる2つのターミナルで実行してみてください。

```
$ ./viewersampleGainMultiCam IMX219のプロファイル IMX219 0
$ ./viewersampleGainMultiCam IMX219のプロファイル IMX219 1
```

2つのウインドウが開き、アクティブにした方のウインドウに関連するゲインを+、-キーで変更することができます。

（2）viewersampleGainMultiCamThreadサンプル
このサンプルでは、単一プロセスで、異なるカメラ番号に紐付けた複数のスレッドを生成し、単一のOpenCVのWindowを開き、それぞれのイメージセンサからのフレームを描画します。OpenCVではcvNamedWindow関数で開くウィンドウは1つのプロセスに1つだけ、という制限があることから、1つのウィンドウを複数のスレッドで共有します。
フレーム取り出しのコールバックは、2つのセンサで共通で使っていますが、frameに関連付けられたカメラ番号によって、描画をする振フレームを選んで、上述のウインドウに表示しています。カメラ表示の変更はcキーに割り当てられています。また、表示されているイメージセンサーに関連付けられたCCIポートは、ssp_control_focus_on_camera関数でアクセできるようになるので、そのイメージセンサのレジスタだけ、変更しています。
例えば、以下のコマンドをターミナルで実行してみてください。

```
$ ./viewersampleGainMultiCamThread IMX219のプロファイル IMX219 IMX219のプロファイル IMX219
```

この例では、CAM0、CAM1にIMX219を接続した場合です。