

Les moyens de communication de la carte smoothieboard avec l'extérieur

I) Communication avec UART, SPI ou I2c

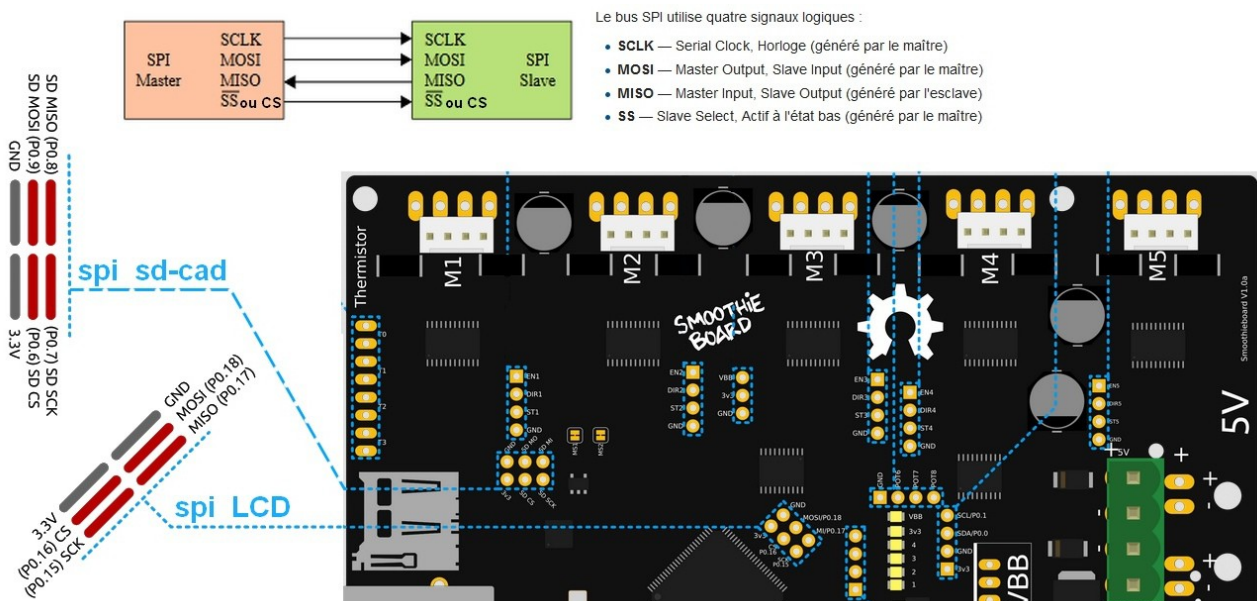
Certains pin GPIO de la carte smoothieboard peuvent être configuré dans le fichier config pour qu'ils permettent des liaison SPI I2c ou UART

Les pin P0.16 P0.15 P0.18 et P0.17 sont utilisés comme liaison SPI par exemple par le Panel GLCD pour gérer l'écran LCD

Dans le fichier config il y a :

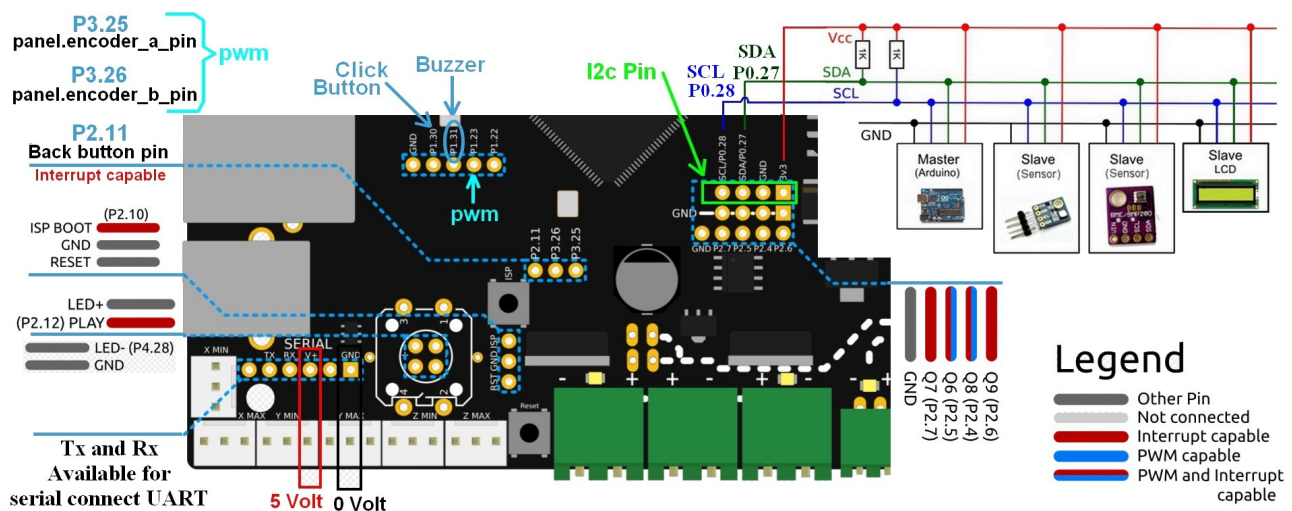
```
# config settings
panel.enable          true          # set to true to enable the panel code
panel.lcd             reprap_discount_glcd # set type of panel
panel.spi_channel     0              # spi channel to use ;
panel.spi_cs_pin      0.16          # spi chip select ; GLCD EXP1 Pin 4
```

Seul le Pin 0.16 (CS) est indiqué dans le fichier config, je suppose que le reste Sériel clock MOSI et MISO c'est codé dans le firmware ?



La liaison spi-card P0.6 est celle de la carte SD qui est reprise par le panel GLCD mais la config pour un lecteur carte sd externe est indiqué dans le fichier config à partir d'une liaison I2c en bas de la carte

```
# setup for external sd card on the GLCD which uses the onboard sdcad SPI port
panel.external_sd true # set to true if there is an extrernal sdcad on the panel
panel.external_sd.spi_channel 1 # set spi channel the sdcad is on
panel.external_sd.spi_cs_pin 0.28 # set spi chip select for the sdcad (or any spare pin)
panel.external_sd.sdcd_pin 0.27!^ # sd detect signal (set to nc if no sdcad detect) (or any spare pin)
```



La carte smoothieboard peut communiquer avec les liaisons SPI, I2c et UART

Page du wiki smoothieboard dédiée aux protocoles logiciels de ces liaisons <http://smoothieware.org/smoothie-accessory-protocol>

II) Communication avec n'importe quel pin GPIO avec les modules

Les cartes à microcontrôleurs (Raspberry Pi, Arduino, Beaglebone, Smoothieboard ...) ont des entrées sorties prévues pour les usages classiques, port USB, port réseau, port écran HDMI ...

Les fonctions de ces cartes à microcontrôleurs sont extensibles en y connectant des appareils, d'autres cartes électroniques.

Les éléments externes qui seront des options utilisés par la suite se connectent par l'intermédiaire des connecteurs **GPIO**

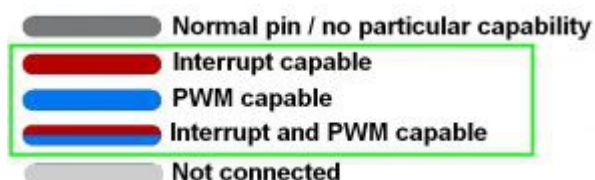
GPIO signifie **General Purpose Input/Output**, littéralement (*Entrée/Sortie pour un Usage Général*)

Au lieu d'être regroupé à un seul endroit comme pour les cartes Raspberry et Arduino.

Les ports GPIO de la carte smoothieboard sont répartis un peu partout.

<http://smoothieware.org/pinout>

Il y a énormément de broches de connection GPIO de disponible dont certaines peuvent avoir des capacités particulières



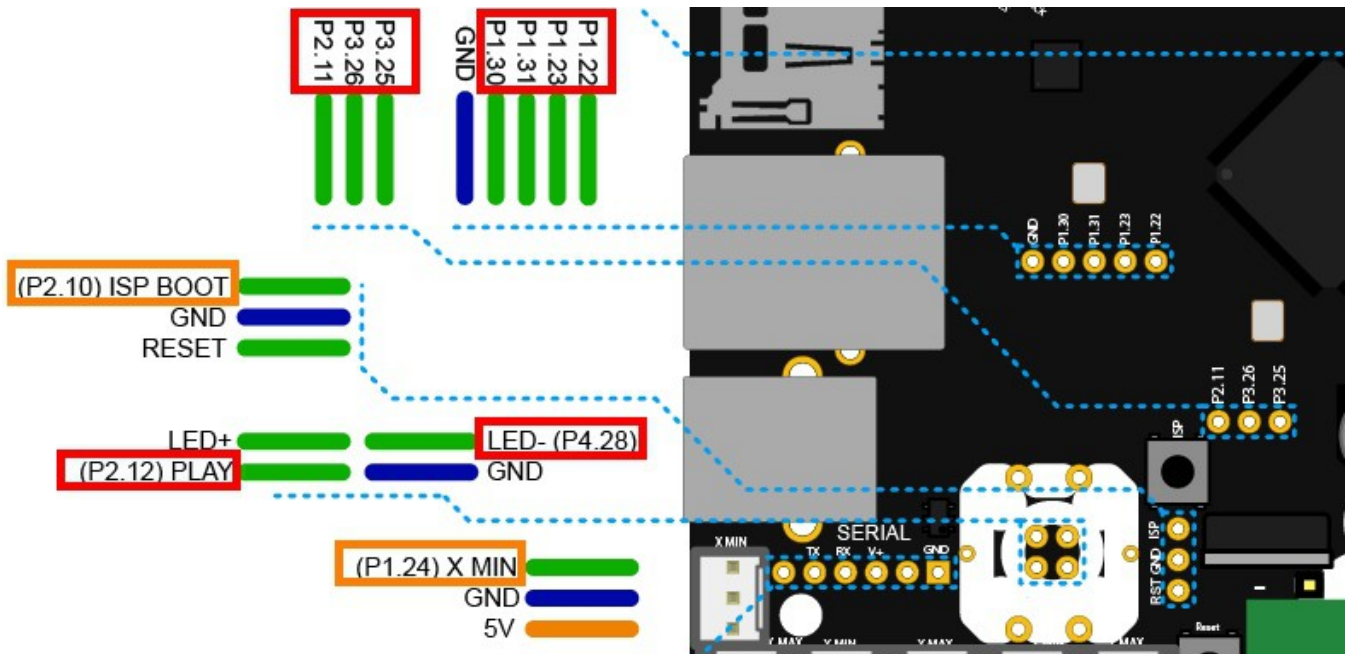
Les broches GPIO avec possibilité d'interruption peuvent être utilisées pour des boutons pour modifier immédiatement le déroulement habituel de l'usage même si celui-ci est en cours.

PWM signifie **Pulse Width Modulation** (modulation de largeur d'impulsions) permet d'avoir une sortie analogique pour par exemple moduler la vitesse de la broche à partir de la carte.

Chaque port GPIO est identifié par **P**(un nombre).(un nombre)

Beaucoup de port GPIO sont libres d'utilisation comme ceux ci-dessous encadrés en rouge.
Mais si certaines fonctions de la carte ne sont pas utilisées il est possible d'affecter les broches correspondante à un usage général GPIO

Ce serait le cas pour les broches ici entourées en orange. Par exemple le capteur endstop X min s'il n'est pas utilisé, sa broche **P1.24** pourrait être affectée à autre chose.



Exemples de configuration de port GPIO grâce aux modules du fichiers config

A) GPIO en sortie

a1) Commander un relais statique pour démarrer et éteindre la broche à partir du Gcode

Switch module for spindle control

```
switch.spindle.enable          true          #
switch.spindle.input_on_command M3          #
switch.spindle.input_off_command M5          #
switch.spindle.output_pin      1.22         #
switch.spindle.output_type     digital      # just an on or off pin
```

B) GPIO en entrée

b1) Créer un bouton pause qui suspend l'usinage lors d'un premier appui et le reprend en appuyant une seconde fois

Switch module for pause button

```
switch.pause.enable           true          # Enable this module
switch.pause.input_pin        2.11^        # Pin where pause button is connected
```

```
switch.pause.output_on_command  suspend          # Suspend command
switch.pause.output_off_command resume          # Resume command
switch.pause.input_pin_behavior toggle          # This pin toggles between it's on and off
states each time it is pressed and released
```

```
after_suspend_gcode            M5            # Gcode to run after suspend
before_resume_gcode            M3            # Gcode to run before resume
```

b2) Créer un bouton reset pour redémarrer la carte smoothieboard

```
# Switch module for reset button
```

```
switch.reset.enable            true          # Enable this module
switch.reset.input_pin         3.26^        # Pin where reset button is connected
switch.reset.output_on_command reset        # Command to reset the board
```

b3) Avoir un bouton kill qui arrête et annule immédiatement l'usinage

```
# Kill button (used to be called pause) maybe assigned to a different pin, set to the onboard pin by
default
```

```
kill_button_enable             true          # set to true to enable a kill button
kill_button_pin                2.12         # kill button pin. default is same as pause button 2.12
(2.11 is another good choice)
```

Plus d'information sur les modules

<http://smoothieware.org/moduleexample>

<http://smoothieware.org/switch>

et la configuration des pin <http://smoothieware.org/pin-configuration>