

Paquetage HWSUPP - Supporte du matériel spécifique Version 3.10.4

L'équipe fli4l
courriel: team@fli4l.de

25 octobre 2015

Table des matières

1. Documentation du paquetage HWSUPP	3
1.1. HWSUPP - Supporte du matériel spécifique	3
1.1.1. Description	3
1.1.2. Configuration du paquetage HWSUPP	3
1.1.3. Paramètre pour expert	7
1.1.4. Prise en charge des cartes VPN	8
A. Annexe du paquetage HWSUPP	9
A.1. HWSUPP - Paramètres dépendant du périphérique	9
A.1.1. Périphérique disponible pour la LED	9
A.1.2. Bouton disponible du périphérique	10
A.1.3. Note sur le matériel spécifique	10
A.2. HWSUPP - Exemple de configuration	11
A.2.1. PC générique	11
A.2.2. PC engines APU	11
A.2.3. PC engines APU avec GPIO	11
A.3. HWSUPP - Séquence de clignotement de la LED	12
A.4. HWSUPP - Conseil pour les développeurs de paquetage	13
A.4.1. Extension pour LED	13
A.4.2. Extension pour le bouton	14
A.4.3. Action du bouton	14
Index	16

1. Documentation du paquetage HWSUPP

1.1. HWSUPP - Supporte du matériel spécifique

1.1.1. Description

Ce paquetage fournit un support pour l'utilisation de composants et de matériels spécifiques. Matériels et composants prisent en charges :

- Capteurs de température
- LEDs
- Capteurs de tension
- Vitesse des ventilateurs
- Bouton poussoir
- Watchdog (ou chien de garde)
- Carte VPN

Il prend en charge aussi les systèmes/cartes mère/cartes VPN :

- Matériel pour PC standard
 - LEDs et bouton d'ordinateur
- Matériel ACPI
- Système embarqué
 - AEWIN SCB6971
 - Fujitsu Siemens Futro S200
 - PC Engines ALIX
 - PC Engines APU
 - PC Engines WRAP
 - Soekris net4801
 - Soekris net5501
- Carte mère
 - Commell LE-575
 - GigaByte GA-M521-S3
 - LEX CV860A
 - SuperMicro PDSME
 - SuperMicro X7SLA
 - Tyan S5112
 - WinNet PC640
 - WinNet PC680
- Carte VPN (PCI, miniPCI et miniPCIE)
 - vpn1401 vpn1411

1.1.2. Configuration du paquetage HWSUPP

La configuration se fait comme les autres paquetages fli4l, en paramétrant le fichier Pfad/fli4l-3.10.4/<config>/hwsupp.txt selon votre propre configuration.

OPT_HWSUPP La valeur 'no' dans cette variable désactive complètement le paquetage OPT_HWSUPP. Il n'y aura aucun changement sur le support de boot de l'archive fli4l rootfs.img n'y dans l'archive opt.img. Pour finir OPT_HWSUPP n'écrase aucune partie de l'installation fli4l.

Pour activer la variable OPT_HWSUPP du paquetage OPT_HWSUPP vous devez placer la valeur sur 'yes'.

HWSUPP_TYPE Dans cette variable vous indiquez le matériel à configurer. Les valeurs suivantes sont disponibles :

- sim
- generic-pc
- generic-acpi
- aewin-scb6971
- commell-le575
- fsc-futro-s200
- gigabyte-ga-m52l-s3
- lex-cv860a
- pcengines-alix
- pcengines-apu
- pcengines-wrap
- soekris-net4801
- soekris-net5501
- supermicro-pdsme
- supermicro-x7sla
- tyan-s5112
- winnet-pc640
- winnet-pc680

HWSUPP_WATCHDOG La valeur 'yes' dans cette variable active le démon Watchdog, si le matériel sélectionné est équipé du Watchdog. le Watchdog redémarre automatiquement le système qui a été momentanément arrêté pour une raison quelconque.

HWSUPP_CPUFREQ Si vous avez indiquez 'yes' dans cette variable, vous pouvez gérer la fréquence du processeur en fonction de la demande en ressources du système et des applications.

HWSUPP_CPUFREQ_GOVERNOR Dans cette variable vous sélectionnez le gouverneur pour la fréquence du processeur. Le gouverneur sélectionné contrôle le comportement et le réglage de la fréquence. Vous pouvez sélectionner l'un d'entre eux :

- performance
Le CPU fonctionne toujours avec la fréquence la plus élevée.
- ondemand
La fréquence du CPU sera ajustée en fonction de l'utilisation du CPU. La fréquence peut changer très rapidement.
- conservative
La fréquence du CPU sera ajustée en fonction de l'utilisation du CPU. La fréquence est modifiée étape par étape.
- powersave
Le CPU fonctionne toujours avec la fréquence la plus basse.
- userspace

La fréquence du CPU peut être réglée manuellement ou par un script utilisateur via la variable `sysfs` dans `/devices/system/cpu/cpu<n>/cpufreq/scaling_setspeed`.

HWSUPP_LED_N Vous indiquez dans cette variable le nombre de LEDs, les appareils sont différents et peuvent avoir un nombre de LED différente.

HWSUPP_LED_x Vous indiquez ici la valeur qui sera affichée par la LED, les valeurs suivantes sont disponibles :

- `ready` - Le routeur `fli4l` est prêt
- `online` - Le routeur `fli4l` est connecté à l'Internet
- `trigger` - La LED est commandée par un déclencheur du kernel
- `user` - L'affichage est commandé par un script utilisateur

La liste des valeurs peut être prolongée en utilisant d'autres paquetages. Si vous chargez le paquetage `WLAN` l'affichage pour le sans fil sera possible.

- `wlan` - Le `WLAN` est activé

Pour créer une telle extension vous avez dans l'annexe [A.4](#) des conseils pour les développeurs de paquetages.

HWSUPP_LED_x_DEVICE Dans cette variable vous indiquez un périphérique pour la LED du matériel utilisé. Soit vous paramétrez un périphérique pour la LED qui se trouve dans le répertoire `/sys/class/leds/` du routeur, soit vous paramétrez un nombre `GPIO` ¹

Une liste de noms des périphériques pour la LED peut être trouvées dans la documentation [annexe](#), selon le matériel spécifique dans `HSUPP_TYPE`.

Si vous indiquez un nombre `GPIO` il doit être indiqué dans le format `gpio::x`. Si vous avez indiqué un `GPIO`, le périphérique correspondant à la LED sera créé automatiquement.

Si vous placez le caractère `/` devant `GPIO`, son fonctionnement sera inversé.

Exemple :

```
HWSUPP_LED_1_DEVICE='apu::1'      # LED 1 sur PC engines APU
HWSUPP_LED_2_DEVICE='gpio::237'    # GPIO 237
HWSUPP_LED_3_DEVICE='/gpio::245'   # GPIO 245 inversé
```

HWSUPP_LED_x_PARAM Dans cette variable vous indiquez les paramètres pour l'affichage de la LED.

Selon la valeur de la variable `HWSUPP_LED_x`, la variable `HWSUPP_LED_x_PARAM` aura une signification différente.

Si vous avez paramétré la variable `HWSUPP_LED_x='trigger'`, avec la valeur `trigger` (ou déclencheur), le contrôle d'activation de la LED doit être paramétré dans la variable `HWSUPP_LED_x_PARAM`.

Dans cette variable vous définissez le `trigger` (ou déclencheur) qui commande la LED. Les déclencheurs disponibles peuvent être affichés avec la commande `cat /sys/class/leds/*/trigger`.

Parmi les déclencheurs créés, par exemple `netfilter` ou les pilotes de matériels comme `ath9k`, d'autres modules de déclencheurs peuvent être chargés via la variable `HWSUPP_DRIVER_x`.

Exemple :

0. Si vous avez paramétré la variable `HWSUPP_LED_x='ready'` avec cette valeur `ready`, la progression du boot sera indiquée par une séquence de clignotement de la LED (voir [appendix A.3](#)).

1. `GPIO` (General Purpose Input/Output, c'est-à-dire entrée/sortie pour un usage général) le nombre correspond à la position physique de la broche sur le circuit intégré, le comportement de la broche peut être programmé au moment de son exécution, notamment s'il s'agit d'une entrée ou d'une sortie.

1. Documentation du paquetage HWSUPP

```
HWSUPP_LED_1='trigger'
HWSUPP_LED_1_TRIGGER='heartbeat'
HWSUPP_LED_2='trigger'
HWSUPP_LED_2_TRIGGER='netfilter-ssh'
```

Si dans la variable 'HWSUPP_LED_x' vous avez indiqué la valeur 'user', vous devez indiquer dans la variable HWSUPP_LED_PARAM le nom du script pour la LED ainsi que le chemin de celui-ci.

Exemple :

```
HWSUPP_LED_1='user'
HWSUPP_LED_1_PARAM='/usr/local/bin/myledscript'
```

Si vous avez indiqué dans la variable la valeur HWSUPP_LED_x='wlan', vous devez indiquer dans la variable HWSUPP_LED_x_PARAM le nom du périphérique.

Vous pouvez indiquer dans cette variable un ou plusieurs périphériques WLAN (ou sans fil). Si vous avez indiqué plusieurs périphériques WLAN ils seront séparés par un espace. Si vous avez indiqué plusieurs WLAN et si vous avez activé la LED, elle aura la signification suivante :

- Éteinte - Tous les périphériques sans fil sont inactifs
- Clignotante - Une partie des périphériques sans fil est actif
- Allumée - Tous les périphériques sans fil sont actifs

Exemple :

```
HWSUPP_LED_1='wlan'
HWSUPP_LED_1_WLAN='wlan0 wlan1'
```

HWSUPP_BOOT_LED Dans cette variable vous pouvez indiquer la séquence de clignotement de la LED par rapport à la progression du boot.

Si vous avez indiqué dans la variable la valeur HWSUPP_LED_x='ready', la variable HWSUPP_BOOT_LED sera ignorée.

HWSUPP_BUTTON_N Dans cette variable vous indiquez le nombre de boutons. Le nombre de boutons dépend du matériel utilisé.

HWSUPP_BUTTON_x Dans cette variable vous définissez l'action qui doit être exécutée lorsque vous pressez le bouton. Les actions suivantes sont possibles :

- reset - Réinitialise le routeur fli4l
- online - Active ou désactive la connexion Internet
- user - Le script utilisateur sera exécutée

La liste des valeurs peut être prolongée en utilisant d'autres paquetages. Si vous chargez le paquetage WLAN une action sur le sans fil sera possible.

- wlan - Active ou désactive le sans fil (ou WLAN)

HWSUPP_BUTTON_x_DEVICE Dans cette variable vous indiquez le périphérique pour le bouton.

Si vous indiquez un nombre GPIO, il doit être dans le format gpio::x. Si vous placez le caractère / devant GPIO, son fonctionnement sera inversé.

Une liste GPIO prédéfinie peut être trouvée dans la documentation annexe [A.1.2](#), selon le matériel spécifique dans `HSUPP_TYPE`.

Exemples :

```
HWSUPP_BUTTON_1_DEVICE='gpio::252'  
HWSUPP_BUTTON_2_DEVICE='/gpio::237'
```

HWSUPP_BUTTON_x_PARAM Dans cette variable vous indiquez les paramètres pour l'action du bouton.

Selon la valeur de la variable `HWSUPP_BUTTON_x`, la variable `HWSUPP_BUTTON_x_PARAM` aura une signification différente.

Si vous avez indiqué dans la variable la valeur `HWSUPP_BUTTON_x='user'`,

Vous devez placer dans `HWSUPP_BUTTON_x_PARAM` le nom du fichier script à exécuter lorsque le bouton sera enfoncé.

Exemple :

```
HWSUPP_BUTTON_1='user'  
HWSUPP_BUTTON_2_WLAN='/usr/local/bin/myscript'
```

Si vous avez indiqué dans la variable la valeur `HWSUPP_BUTTON_x_ACTION='wlan'`, vous devez indiquer dans la variable `HWSUPP_BUTTON_x_PARAM` le nom du périphérique sans fil.

Si dans cette variable vous indiquez un ou plusieurs périphériques WLAN, ils seront surveillés par le système. Si vous indiquez plusieurs périphériques sans fil, vous devez les séparer par un espace. Grâce à la pression sur le bouton le périphérique sera activé ou désactivé.

Exemple :

```
HWSUPP_BUTTON_2='wlan'  
HWSUPP_BUTTON_2_WLAN='wlan0 wlan1'
```

1.1.3. Paramètre pour expert

Les paramètres suivants doivent être utilisés uniquement lorsque vous savez exactement

- quel matériel que vous avez et quel pilote supplémentaire est nécessaires.
- l'adresse et le type de périphérique I²C².

En activant les paramètres experts, vous aurez un message d'avertissement lors de la construction de `fli4l` avec la commande `mkfli4l`.

HWSUPP_DRIVER_N Dans cette variable vous indiquez le nombre de pilotes supplémentaires. Les pilotes dans la variable `HWSUPP_DRIVER_x` seront chargés dans l'ordre enregistré.

HWSUPP_DRIVER_x Dans cette variable vous indiquez le nom du pilote (sans l'extension du fichier `.ko`).

Exemple :

2. Un bus I²C ou un SMBus est un bus série sur PC, il est utilisé par exemple pour la lecture de la température avec un capteur. Dans de nombreux cas, le bus I²C ou le SMBus est disponible par l'intermédiaire d'un connecteur à broche il peut être utilisé pour une extension de matériel personnel.

```
HWSUPP_DRIVER_N='2'  
HWSUPP_DRIVER_1='i2c-piix4'      # pilote du bus I2C  
HWSUPP_DRIVER_2='gpio-pcf857x'  # I2C extension GPIO
```

HWSUPP_I2C_N Dans cette variable vous indiquez le nombre de périphérique I²C à charger. I²C ne supporte pas du tout le mécanisme PnP. Par conséquent, vous devez avoir un numéro de bus pour chaque périphérique I²C, l'adresse du périphérique et le type de périphérique pour la configuration.

HWSUPP_I2C_x_BUS Dans cette variable vous indiquez le numéro de bus I²C donc le périphérique sera associé.

Le numéro de bus doit être saisi dans le format suivant `i2c-x`.

HWSUPP_I2C_x_ADDRESS Dans cette variable vous indiquez l'adresse du périphérique I²C.

L'adresse doit être indiquée comme un nombre hexadécimal et dans la plage de 0x03 à 0x77.

HWSUPP_I2C_x_DEVICE Dans cette variable vous indiquez le type de périphérique I²C qui sera supporté par le pilote précédemment chargé.

Exemple :

```
HWSUPP_I2C_N='1'  
HWSUPP_I2C_1_BUS='i2c-1'  
HWSUPP_I2C_1_ADDRESS='0x38'  
HWSUPP_I2C_1_DEVICE='pcf8574a' # supporté par le pilote gpio-pcf857x
```

1.1.4. Prise en charge des cartes VPN

OPT_VPN_CARD La valeur 'no' dans cette variable désactive complètement le paquetage OPT_VPN_CARD. Il n'y aura aucun changement sur le support de boot de l'archive `fli4l rootfs.img` n'y dans l'archive `opt.img`. Pour finir OPT_VPN_CARD n'écrase aucune partie de l'installation `fli4l`.

Pour activer la variable OPT_VPN_CARD du paquetage OPT_VPN_CARD vous devez placer la valeur sur 'yes'.

VPN_CARD_TYPE Dans cette variable vous pouvez indiquer l'accélérateur VPN. La carte VPN décharge le CPU des tâches informatiques intensives de cryptage et de la fonction de hachage. Les valeurs suivantes sont disponibles :

- `hifn7751` - Pour carte Soekris `vpn1401` et `vpn1411`
- `hifnhipp`

A. Annexe du paquetage HWSUPP

A.1. HWSUPP - Paramètres dépendant du périphérique

A.1.1. Périphérique disponible pour la LED

Selon le matériel voir la variable `HWSUPP_TYPE` différents périphériques pour la LED seront disponibles. Pour le matériel ne figurant pas ici, les LEDs du clavier du PC peuvent être utilisées avec le [pc-générique](#).

Des périphériques supplémentaires pour la LED peuvent être disponibles par exemple pour le périphérique WLAN. Les noms valides des périphériques pour la LED peuvent d'être trouvés avec la commande `ls /sys/class/leds/`. En utilisant par exemple le ssh ou la console du routeur.

sim

Pour gérer une simulation de la LED, qui sera enregistré dans `syslog` :

- `simu::1`
- ...
- `simu::8`

PC générique

Les LEDs du clavier du PC :

- `keyboard::scroll`
- `keyboard::caps`
- `keyboard::num`

Générique acpi

Les LEDs du clavier du PC, comme dans [PC générique](#)

PC engines alix

- `alix::1`
- `alix::2`
- `alix::3`

PC engines apu

- `apu::1`
- `apu::2`
- `apu::3`

PC engines wrap

- wrap::1
- wrap::2
- wrap::3

Soekris net4801

- net48xx::error

Soekris net5501

- net5501::error

A.1.2. Bouton disponible du périphérique

Selon le matériel de la variable `HWSUPP_TYPE`, les périphériques GPIO suivants sont prédéfinis pour les boutons.

PC engines alix

- gpio::24

PC engines apu

- gpio::252

PC engines wrap

- gpio::40

soekris net5501

- gpio::25

Le bouton est nommé 'Reset' pour le matériel Soekris.
Attention : le bouton doit être activé dans le BIOS.

A.1.3. Note sur le matériel spécifique

PC engines alix

Un pilote défectueux du capteur de température LM90, entraîne une perte de contrôle de la température.

Pour contourner ce problème le pilote LM90 sera automatiquement déchargé et rechargé en utilisant une tâche de cron. Il est nécessaire que le paquetage easycron soit installé avec la variable (`OPT_EASYCRON='yes'`).

A.2. HWSUPP - Exemple de configuration

A.2.1. PC générique

```
OPT_HWSUPP='yes'
HWSUPP_TYPE='generic-pc'

HWSUPP_WATCHDOG='no'
HWSUPP_CPUFREQ='no'

HWSUPP_LED_N='3'
HWSUPP_LED_1='ready'
HWSUPP_LED_1_DEVICE='keyboard::num'
HWSUPP_LED_2='online'
HWSUPP_LED_2_DEVICE='keyboard::caps'
HWSUPP_LED_3='wlan'
HWSUPP_LED_3_DEVICE='keyboard::scroll'
HWSUPP_LED_3_WLAN='wlan0'

HWSUPP_BUTTON_N='0'
```

A.2.2. PC engines APU

```
OPT_HWSUPP='yes'
HWSUPP_TYPE='pcengines-apu'

HWSUPP_WATCHDOG='yes'
HWSUPP_CPUFREQ='yes'
HWSUPP_CPUFREQ_GOVERNOR='ondemand'

HWSUPP_LED_N='3'
HWSUPP_LED_1='ready'
HWSUPP_LED_1_DEVICE='apu::1'
HWSUPP_LED_2='wlan'
HWSUPP_LED_2_DEVICE='apu::2'
HWSUPP_LED_2_WLAN='wlan0'
HWSUPP_LED_3='online'
HWSUPP_LED_3_DEVICE='apu::3'

HWSUPP_BUTTON_N='1'
HWSUPP_BUTTON_1='wlan'
HWSUPP_BUTTON_1_DEVICE='gpio::252'
HWSUPP_BUTTON_1_PARAM='wlan0'
```

A.2.3. PC engines APU avec GPIO

```
OPT_HWSUPP='yes'
HWSUPP_TYPE='pcengines-apu'
```

```

HWSUPP_WATCHDOG='yes'
HWSUPP_CPUFREQ='yes'
HWSUPP_CPUFREQ_GOVERNOR='ondemand'

HWSUPP_LED_N='5'
HWSUPP_LED_1='ready'
HWSUPP_LED_1_DEVICE='apu::1'
HWSUPP_LED_2='wlan'
HWSUPP_LED_2_DEVICE='apu::2'
HWSUPP_LED_2_WLAN='wlan0'
HWSUPP_LED_3='online'
HWSUPP_LED_3_DEVICE='apu::3'
HWSUPP_LED_4='trigger'
HWSUPP_LED_4_PARAM='phy0rx'
HWSUPP_LED_4_DEVICE='gpio::237'
HWSUPP_LED_5='trigger'
HWSUPP_LED_5_PARAM='phy0tx'
HWSUPP_LED_5_DEVICE='gpio::245'

HWSUPP_BUTTON_N='2'
HWSUPP_BUTTON_1='wlan'
HWSUPP_BUTTON_1_DEVICE='gpio::252'
HWSUPP_BUTTON_1_PARAM='wlan0'
HWSUPP_BUTTON_2='online'
HWSUPP_BUTTON_2_DEVICE='gpio::236'

```

A.3. HWSUPP - Séquence de clignotement de la LED

Les séquences de clignotants suivantes seront affichés pendant le processus de boot :

1.	⊗				⊗				...
2.	⊗	⊗			⊗	⊗			...
3.	⊗	⊗	⊗		⊗	⊗	⊗		...
4.	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	...

Lors de l'exécution de rc002.* à rc250.* La première séquence s'affiche,
 (1 * flash - pause),
 de rc250.* à rc500.* la seconde séquence (2 * flash - pause),
 de rc500.* à rc750.* la troisième séquence et
 de rc750.* jusqu'à la fin du processus de boot la quatrième séquence (clignotement continu).

A.4. HWSUPP - Conseil pour les développeurs de packaging

Dans ce chapitre nous vous proposons une description pour ajouter une LED ou un bouton pour les développeurs qui veulent créer un packaging¹.

A.4.1. Extension pour LED

Type de LED

Dans le fichier `check/myopt.exp` vous configurez une liste de plusieurs types de LEDs que vous pouvez indiquer dans la variable `HWSUPP_LED_x`.

Exemple :

```
+HWSUPP_LED_TYPE(OPT_MYOPT) = 'myopt'
                               : ', myopt'
```

Contrôle de paramètre

Dans le fichier `check/myopt.ext` vous configurez les paramètres qui seront vérifiés, vous pouvez les indiquer dans la variable `HWSUPP_LED_x_PARAM`.

Exemple :

```
if (opt_hwsupp)
then
    depends on hwsupp version 4.0

    foreach i in hwsupp_led_n
    do
        set action=hwsupp_led_%[i]
        set param=hwsupp_led_%_param[i]
        if (action == "myopt")
        then
            if (!(param =~ "(RE:MYOPT_LED_PARAM)"))
            then
                error "When HWSUPP_LED_\${i}='myopt', ...
                        must be entered in HWSUPP_LED_\${i}_PARAM"
            fi
        fi
    done
fi
```

affichage de la LED

Quand vous définissez une LED dans le script pour un packaging (par ex. `/usr/bin/<opt>_setled`) la requête `/usr/bin/hwsupp_setled <LED> <status>/` sera exécutée.

Le nombre de LED peut être lu dans le fichier `/var/run/hwsupp.conf`.

L'état de la LED peut être `off`, `on` ou `blink`.

Exemple :

1. Vous recherchez un endroit approprié dans `##HWSUPP##` pour paramétrer le packaging WLAN.

```
if [ -f /var/run/hwsupp.conf ]
then
    . /var/run/hwsupp.conf
    [ 0$hwsupp_led_n -eq 0 ] || for i in `seq 1 $hwsupp_led_n`
    do
        eval action=\$hwsupp_led_${i}
        eval param=\$hwsupp_led_${i}_param
        if [ "$action" = "<opt>" ]
        then
            if [ <myexpression> ]
            then
                /usr/bin/hwsupp_setled $i on
            else
                /usr/bin/hwsupp_setled $i off
            fi
        fi
    done
fi
```

L'état actuel d'une LED peut être demandé avec la requête `/usr/bin/hwsupp_getled <LED>/.`
Le résultat sera `off`, `on` ou `blink`.

A.4.2. Extension pour le bouton

A.4.3. Action du bouton

Dans le fichier `check/myopt.exp` vous configurez une liste de plusieurs types de boutons que vous pouvez indiquer dans la variable `HWSUPP_BUTTON_x`.

Exemple :

```
+HWSUPP_BUTTON_TYPE(OPT_MYOPT) = 'myopt'
                                : ', myopt'
```

Contrôle de paramètre

Dans le fichier `check/myopt.ext` vous configurez les paramètres qui seront vérifiés, vous pouvez les indiquer dans la variable `HWSUPP_BUTTON_x_PARAM`.

Exemple :

```
if (opt_hwsupp)
then
    depends on hwsupp version 4.0

    foreach i in hwsupp_button_n
    do
        set action=hwsupp_buttonn_%[i]
        set param=hwsupp_button_%_param[i]
        if (action == "myopt")
        then
```

A. Annexe du packaging HWSUPP

```
add_to_opt "files/usr/bin/myopt_keyprog" "mode=555 flags=sh"
if (!(param =~ "(RE:MYOPT_BUTTON_PARAM)"))
then
    error "When HWSUPP_BUTTON_\${i}='myopt', ...
          must be entered in HWSUPP_BUTTON_\${i}_PARAM"
fi
fi
done
fi
```

Fonction du bouton

Quand un bouton est pressé, le fichier de script /usr/bin/myopt_keyprog sera exécuté.

La valeur de la variable HWSUPP_BUTTON_x_PARAM est transmis.

Exemple :

```
##TODO## exemple
```

Index

HWSUPP_BOOT_LED, [6](#)
HWSUPP_BUTTON_N, [6](#)
HWSUPP_BUTTON_x, [6](#)
HWSUPP_BUTTON_x_DEVICE, [6](#)
HWSUPP_BUTTON_x_PARAM, [7](#)
HWSUPP_CPUFREQ, [4](#)
HWSUPP_CPUFREQ_GOVERNOR, [4](#)
HWSUPP_DRIVER_N, [7](#)
HWSUPP_DRIVER_x, [7](#)
HWSUPP_I2C_N, [8](#)
HWSUPP_I2C_x_ADDRESS, [8](#)
HWSUPP_I2C_x_BUS, [8](#)
HWSUPP_I2C_x_DEVICE, [8](#)
HWSUPP_LED_N, [5](#)
HWSUPP_LED_x, [5](#)
HWSUPP_LED_x_DEVICE, [5](#)
HWSUPP_LED_x_PARAM, [5](#)
HWSUPP_TYPE, [4](#)
HWSUPP_WATCHDOG, [4](#)

OPT_HWSUPP, [3](#)
OPT_VPN_CARD, [8](#)

VPN_CARD_TYPE, [8](#)