

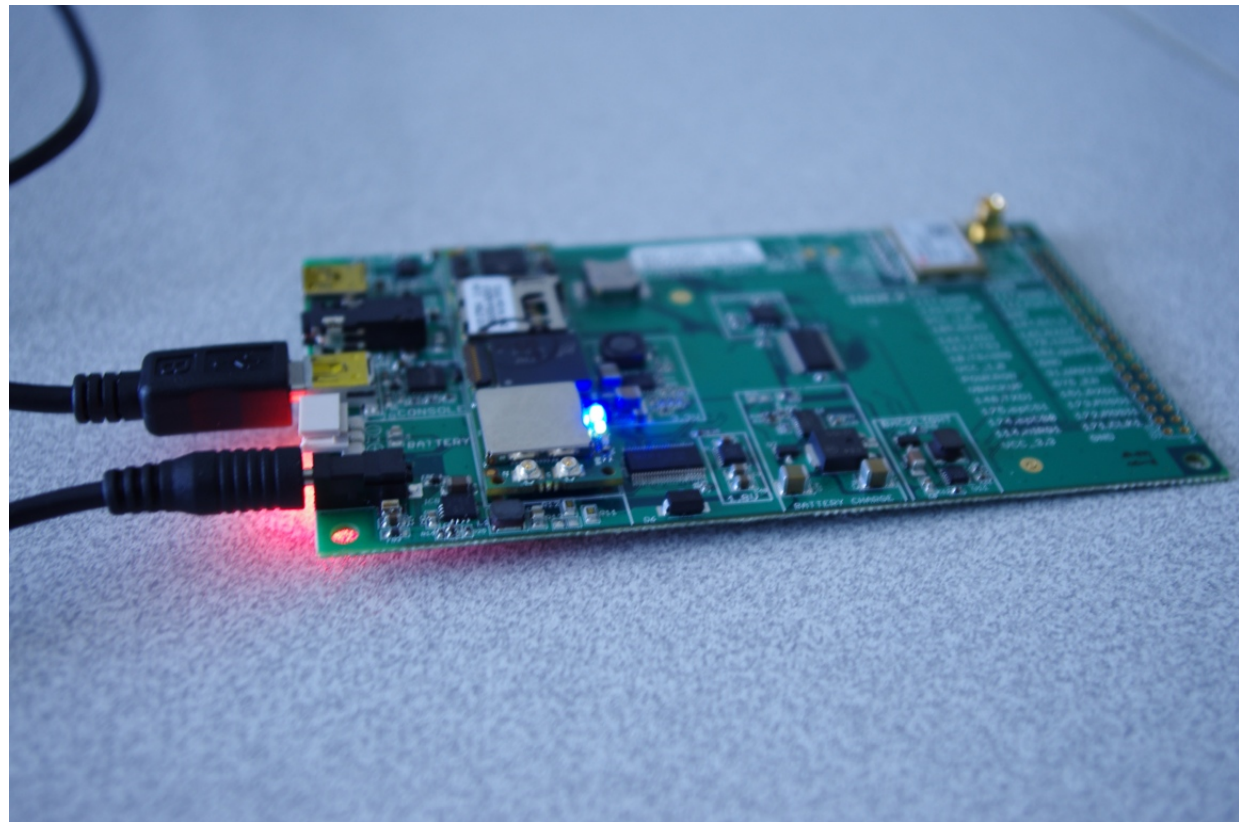
Développement sur Gumstix

Master ISIM-SIC

2011

Gumstix

Principe d'un pc embarqué de dernière génération



Overo AIR

Carte d'extension : Gallop43

Linux

quelques centimètres carrés

Gumstix

Processor: ARM Cortex-A8 CPU 600 MHz

Memory: 512MB RAM 512MB Flash

Features:

OMAP3503 Application Processor **802.11b/g** wireless communications

Bluetooth communications

microSD card slot

TPS65950 Power Management

Expandability:

via [one 140-pin expansion board of Gumstix Overo series](#) or custom,

140-pin expansion board

via 27-pin camera board

Connections:

- [\(2\) 70-pin connectors](#) with 140 signals for:

- I2C, PWM lines (6), A/D (6), 1-wire

- UART, SPI, Extra MMC lines

- Headset, Microphone

- Backup battery

- High Speed USB Host and USB OTG

[\(1\) 27-pin connector](#) with signals for camera board

(2) x u.fl antenna connectors

Power:

Powered via expansion board (Overo series or custom) connected to dual 70-pin connector

Layout:

For OEM design, use layout version R2606 or later, [shown here](#).

Size:

17mm x 58mm x 4.2mm

(0.67 in. x 2.28 in. 0.16 in.)

Weight:

GUM3503A @ 42.6g (incl. shipping case & 2 x antenna)

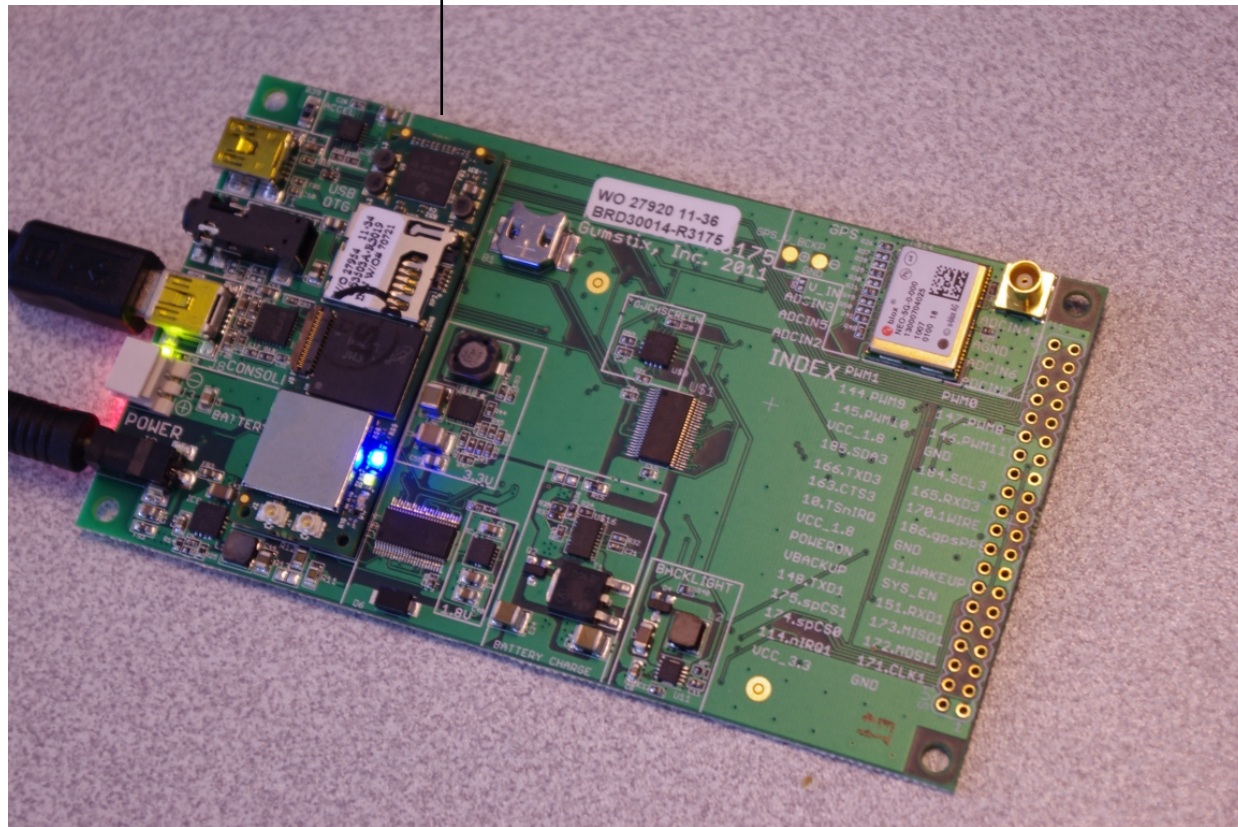
GS3503A @ 5.6g

Temperatures:

Built with components rated $0C < T < 75C$

Mounting:

Four (4) x #0 mounting holes for securing to Overo-series, or custom, expansion board



Gallop43

Features:

GPS via the u-blox NEO-5G module
3-axis accelerometer

[LCD-ready for 4.3" touch-screen](#)
[LCD display @ 24 bits per pixel](#)
Touchscreen controller

USB mini-AB with OTG signals
USB mini-B console port

Coin cell backup battery holder
3.5-mm stereo headset jack

MCX antenna connector for GPS
Two (2) user-configurable LEDs
Two (2) user-configurable push buttons
Forty (40) signals available on 0.100" through-holes at 1.8V logic
Two (2) two-wire serial ports
One 1-wire port
6 PWM lines
I2C port
SPI bus
6 A/D input lines
processor control signals

Power:

Connect a 5v wall adapter to power this expansion board, the connected Overo COM and the attached LCD touch screen
Alternate power jack provided for 2-cell NiMH battery pack

Connecting both the 5v wall adapter and a NiMH battery pack will charge the NiMH batteries

Connections:

2 x 70-pin connectors for connection of any Overo COM

Computing:

Requires [any Gumstix Overo COM](#)

Temperatures:

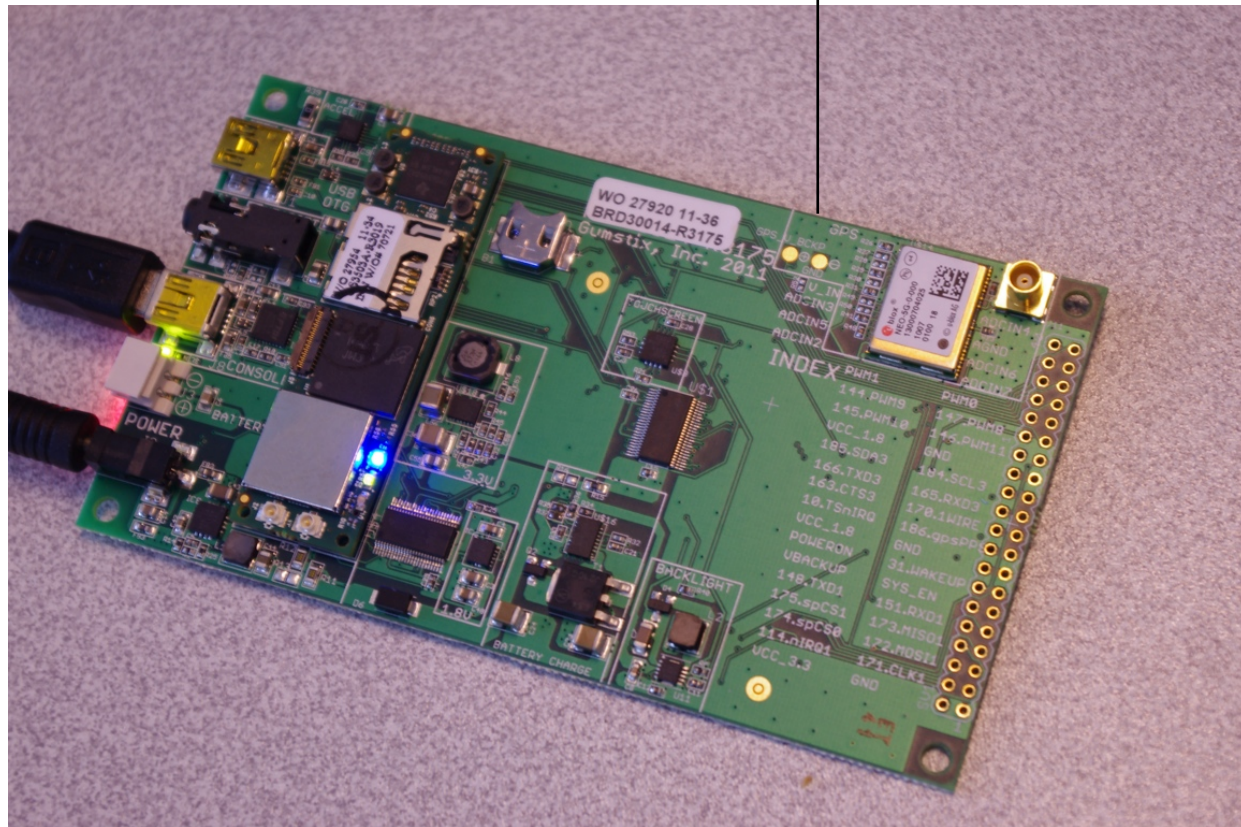
Built with components rated $-20 < T < 70C$

Size:

118.2mm x 67.2mm

Weight:

BRD30014 @ 36.2g



Software

Linux Angström

- Projet basé sur open embedded
- Linux pouvant tourner sur 4MB de mémoire flash
- embarqué sur beagle board
- le gumstix autorise la compilation native
- la compilation croisée passe par la chaine [bitbake](#)
- gestion de packages [opkg](#), proche de ipkg (debian)

- `opkg install nomdupackage.opkg`

- `opkg remove nomdupackage.opkg`

- <http://gumstix.org/add-software-packages.html>

Connexion

Connecting with Minicom

1. First, configure Minicom

1. As root, run `minicom -s`
2. Under "Serial Port Setup" choose the following
 - Serial Device: `/dev/ttyUSB0`
 - Bps/Par/Bits: **115200 8N1**
 - Hardware Flow Control: **No** (this is important)
 - Software Flow Control: **No**
3. Hit [ESC] until you return to the main configuration menu
4. Select "Save Setup as dfl" to save the default setup
5. Select "Exit from Minicom" to exit the program

Ne pas brancher ni connecter la carte

2. Connect the serial cable to the serial port on your computer and the serial port on the Gumstix

3. Run Minicom again with `minicom`
4. **Plug the power adapter into the power jack of the Gumstix.** When connected and powered, you should see a message from U-boot followed by the normal Gumstix boot sequence in the minicom window.
5. Log in for the first time with username *root* and password *gumstix*
6. When finished, you can exit Minicom by typing [CTRL-A] then pressing **Q**. This exits without running the normal modem reset sequence, which will only send garbage to the Gumstix.

connecter la carte

brancher la carte

Cross-compilation (test)

l'environnement de développement et de cross compilation est installé

bitbake

open embedded

`/home/ucp/overo-oe/bitbake`

`/home/ucp/overo-oe`

toutes les sources sont disponibles à la cross compilation dans le répertoire de travail.

- vérifiez que vous avez les variables suivantes définies :
 - BBPATH et PATH positionnée correctement (sur `/cheminbb/bitabke/bin`)
- testez la cross compilation de nano : `bitbake nano`
- le package doit se trouver dans : `/home/ucp/overo-oe/tmp/deploy/glibc/ipk/armv7a`

Transfert de fichier

communication gumstix <-> pc hote (1) : par lien WIFI ad-hoc

GUMSTIX

ESSID GMSTX

- channel 2
- no encryption
- mode ad-hoc
- 192.168.10.10
- utiliser ifconfig, iwconfig, ping

PC HOTE

ESSID GMSTX

- channel 2
- no encryption
- mode ad-hoc
- 192.168.10.2
- utiliser ifconfig, iwconfig, ping

Une fois la connection établie, utiliser scp pour le transfert de fichier

- La machine 01 possède une carte wifi

Transfert de fichier

communication gumstix <-> pc hote (1) : par lien WIFI ad-hoc

n'oubliez pas d'enlever network manager :

```
opkg remove networkmanager
```

pour vous aider :

```
iwconfig wlan0 power on
```

```
iwlist wlan0 scan
```

Transfert de fichier

communication gumstix -> pc hote (2) : par lien série RSD32 (protocole zmodem)

Cross-compilez pour Gumstix : lrzsz (utilitaire d'envoi et de réception des fichier via le port série) et installer lrzsz sur le Gumstix

GUMSTIX

- lancer rz

PC HOTE

- lancer minicom
- se logger sur le gm
- [Ctrl A] - S
- transferer le fichier en zmodem



Développement

HelloCergy.c

-----> HelloCergy.opkg

Editez votre Helloworld

Creez votre branche de développement dans :

~ucp/overo-oe/org.openembedded/recipes/

placez vos sources dans la branche dans le répertoire *file*

creez le fichier de cross compilation bitbake :

HelloCergy_0.0.bb

(standard : *packagename_versionnumber.bb*)

```
DESCRIPTION = "hello world sample program"
PR = "r0"
SRC_URI = "file://hello.c"

S = "${WORKDIR}"

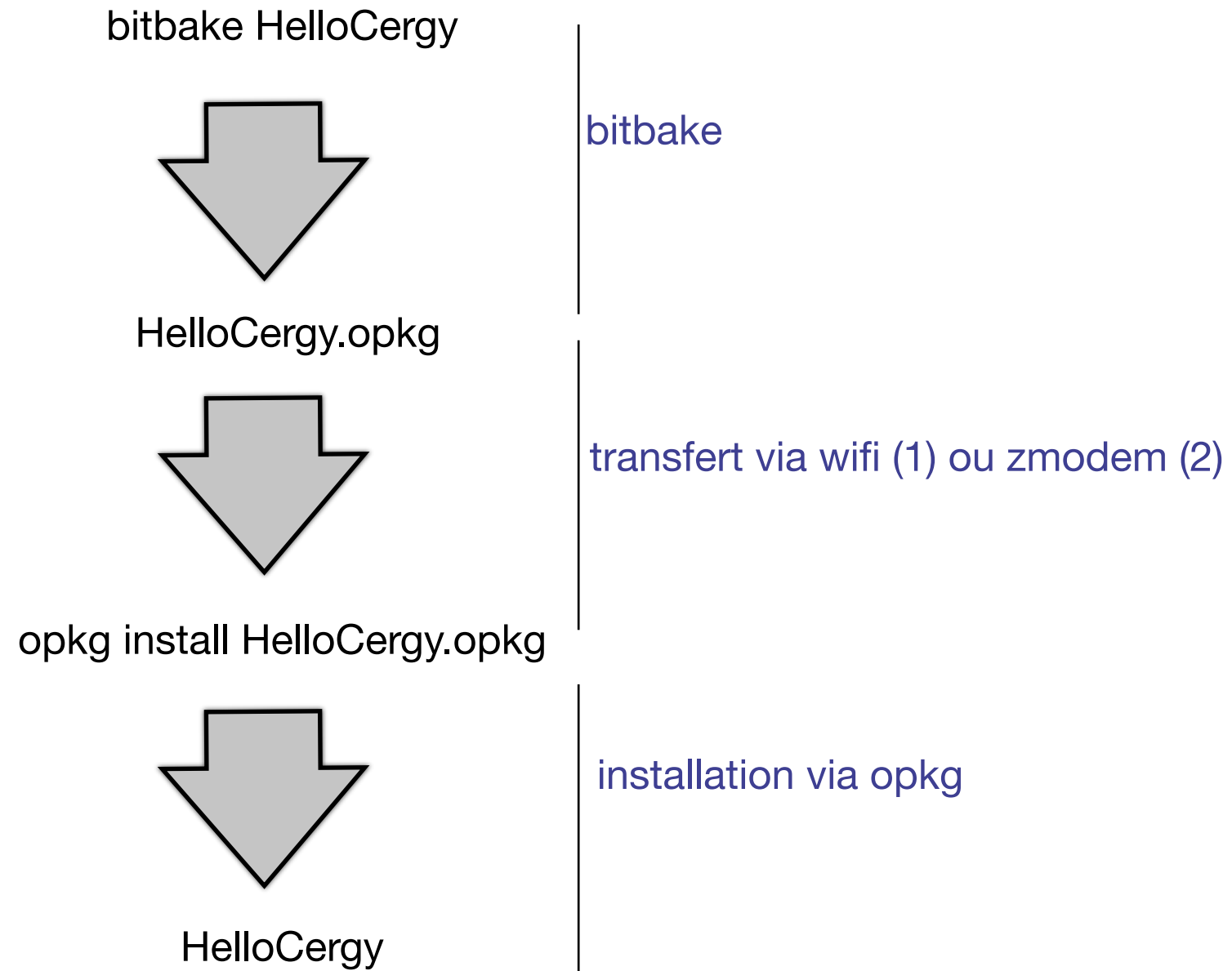
do_compile () {
    ${CC} ${CFLAGS} ${LDFLAGS} -o hello hello.c
}

do_install () {
    install -d ${D}${bindir}/
    install -m 0755 ${S}/hello ${D}${bindir}/
}
FILES_${PN} = "${bindir}/hello"
```

Bitbake builds a recipe by executing a series of steps: fetching code, unpacking, compiling, linking, staging, and packaging.

<http://gumstix.org/software-development/open-embedded/160-bitbake-tutorial.html>

Développement



n'oubliez pas de
'bumper' le Pr

plus loin.....

Gestion des accéléromètres

Accéléromètre 3 axes.

L'accéléromètre est un capteur LIS3DE dont le datasheet est dans le répertoire ci-joint. [lis3de.pdf](#).

-> Page 22 se trouve le registre mapping.

on note que les infos x, y et z se trouvent aux registres respectifs : 29, 2B, et 2D

-> Page 23 contient des informations sur le contrôle des données, qui peut être écrit pour modifier la fréquence de mise à jour des données. notamment le champ DR, avec 0 correspondant à 100HZ.

Par défaut, on a 0000111 avec les 3 derniers bits à 1 pour lire les infos sur les infos en x, y et z.

plus loin.....

Gestion des accéléromètres

il faut d'abord activer l'autorisation en lecture de l'accéléromètre :

```
i2cset -y 3 0x1d 0x20 0x47 b
```

-
- The diagram consists of a horizontal line under the argument '0x1d' in the command above. A vertical line descends from the center of this horizontal line, then a horizontal line extends to the right, meeting a vertical line that runs down the length of the list. This vertical line has four short horizontal dashes pointing to the left, each aligned with one of the list items.
- Adresse du bus
 - Adresse du capteur
 - Adresse du registre
 - Valeur

plus loin.....

Gestion des accéléromètres

Puis interroger les registres

en x :

```
i2cget -f -y 3 0x1d 0x29
```

en y :

```
i2cget -f -y 3 0x1d 0x2B
```

en z :

```
i2cget -f -y 3 0x1d 0x2D
```



En C !

Gestion des accéléromètres

- Reprendre la configuration ecriture/lecture en C
- Autoriser la lecture des accelromètres
- récupérer les donées des accelromètres
- récupérér les fichier sur machine “classique”
- afficher les valeurs des accelromètres

- i2cset.c
- i2c-dev.h

- i2cget.c
- i2c-dev.h
- busses.h
- utils.h

- connection TCP ou UDP
reseau en C

<http://lm-sensors.org/svn/i2c-tools/tags/V3-0-1/tools/>