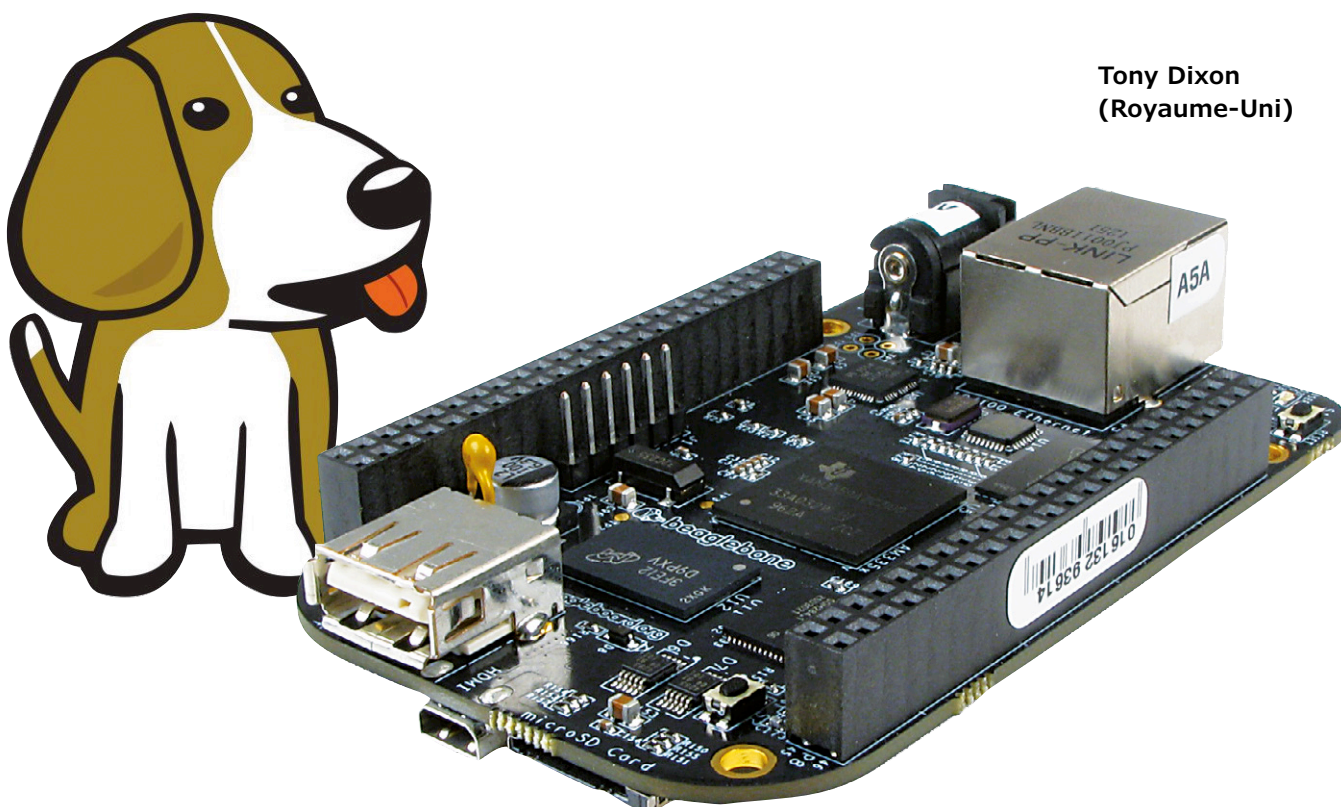


# BeagleBone Black (3)

## entrées analogiques

Tony Dixon  
(Royaume-Uni)



Passez la laisse USB à votre Beagle, après avoir joué avec ses E/S numériques (cf. le premier volet) nous allons lui faire faire le beau avec ses pattes analogiques.

Contrairement à ce qu'aimeraient nous faire croire certains zéloteurs de la Silicon Valley, ou ce genre de type toujours placé près d'une prise Ethernet, le monde n'est pas *que* numérique.

### Les E/S analogiques du BBB

Commençons par jeter un œil sur les caractéristiques du CAN du BBB [1] :

- résolution de 12 bits (0 à 4095) ;
- temps d'échantillonnage de 125 ns ;
- intervalle de tension de 0 V à 1,8 V (!!)

Les connecteurs d'extension du BBB sont équipés de 7 entrées analogiques. Le **tableau 1**

liste les broches analogiques, le **tableau 2** montre le brochage complet du BBB.

Outre les sept signaux analogiques, le port 9 possède deux broches d'alimentation : AVCC (*Analogue VCC*) et AGND (*Analog Ground*).

**Attention**, le niveau de tension maximal des ports GPIO est de 3,3 V, mais celui des broches analogiques n'est que de 1,8 V. Soyez donc attentif aux signaux que vous y connectez si vous ne voulez pas griller votre beagle jusqu'à l'os. Pour mesurer des tensions supérieures à 1,8 V, utilisez un pont diviseur avec pour la résistance « du bas » une valeur de 1 kΩ.

**Tableau 1. Brochage du port analogique**

Signaux (P9)	Broche
AIN0	39
AIN1	40
AIN2	37
AIN3	38
AIN4	35
AIN5	36
AIN6	33
AGND	34
AVCC	32

### Profiter de sysfs

Nous l'avons vu lors du premier article, le système de fichiers virtuel *sysfs* de Linux permet d'accéder aux broches du BBB sans avoir à écrire la moindre ligne de code.

Ouvrons donc un terminal et activons le pilote du CAN avec la commande suivante :

```
echo cape-bone-iiio > /sys/devices/
bone_capemgr.*/slots
```

Nous pouvons lire (mesurer) le potentiel (exprimé en millivolts) qui règne sur la broche AIN0 à l'aide de la commande Linux *cat* :

```
cat /sys/bus/iio/devices/iio\:device0/
in_voltage0_raw
```

Nous pouvons aussi lire directement la valeur du CAN contenue dans le fichier AIN0 :

```
cat /sys/devices/ocp.2/helper.14/AIN0
```

### Un peu de pâtée code

Le système *sysfs* est parfait pour effectuer un test rapide, mais pour des lectures répétées mieux vaut regrouper toutes les opérations nécessaires dans un programme C/C++.

Pour notre essai, nous utilisons un poten-

**Tableau 2. Brochage du BBB**

Signal	P8		Signal	P9		Signal
GND	1	2	GND	1	2	GND
GPIO1_6	3	4	GPIO1_7	3	4	3.3V
GPIO1_2	5	6	GPIO1_3	5	6	5V
TIMER4	7	8	TIMER7	7	8	5V_SYS
TIMER5	9	10	TIMER6	9	10	PWR_BUTTON
GPIO1_13	11	12	GPIO1_12	11	12	UART4_RXD
EHRPWM2B	13	14	GPIO2_26	13	14	GPIO4_TXD
GPIO1_15	15	16	GPIO1_14	15	16	GPIO1_16
GPIO0_27	17	18	GPIO2_1	17	18	I2C1_SCL
EHRPWM2A	19	20	GPIO1_31	19	20	I2C2_SCL
GPIO1_30	21	22	GPIO1_5	21	22	UART2_TXD
GPIO1_4	23	24	GPIO1_1	23	24	GPIO1_17
GPIO1_0	25	26	GPIO1_29	25	26	GPIO3_21
GPIO2_22	27	28	GPIO2_24	27	28	GPIO3_19
GPIO2_23	29	30	GPIO2_25	29	30	SPI1_D0
UART5_CTS	31	32	UART5_RTS	31	32	SPI1_SCLK
UART4_RTS	33	34	UART3_RTS	33	34	AIN4
UART4_CTS	35	36	UART3_CTS	35	36	AIN6
UART5_TXD	37	38	UART5_RXD	37	38	AIN2
GPIO2_12	39	40	GPIO2_13	39	40	AIN0
GPIO2_10	41	42	GPIO2_11	41	42	GPIO_20
GPIO2_08	43	44	GPIO2_09	43	44	GND
GPIO2_6	45	46	GPIO2_07	45	46	GND

tiomètre de 5 kΩ relié à AVCC (broche 32) et AGND (broche 34). Le curseur est relié à AIN0 (broche 39). Lancez l'éditeur *nano* avec :

```
nano analogue.cpp
```

Recopiez le code du **listage 1**, enregistrez le fichier avec Ctrl+X, puis Y et Entrée pour confirmer la sauvegarde. Le code du programme *analogue.cpp* est également dans l'archive 130492-11.zip que vous pouvez télécharger depuis [2].

Pour compiler ce programme C/C++ depuis le terminal, entrez :

```
g++ analogue.cpp -o analogue
```

Si la compilation s'est déroulée sans erreur,

exécutez le programme avec :

```
./analogue
```

La broche analogique est scrutée une fois par seconde. Tournez le curseur et observez les valeurs affichées.

Ce code pourrait servir à mesurer une température à l'aide d'un TMP36 dont la tension de sortie est, ou(a)f, comprise entre 0 V et 1,8 V.

(130492 - version française : Hervé Moreau)

### Liens

[1] Le site de Beagle : <http://beagleboard.org>

[2] [www.elektor-magazine.com/130492](http://www.elektor-magazine.com/130492)

#### Listage 1

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

int main()
{
    int fd, fdstat;
    char buffer[1024];

    const char AIN0 [] = "/sys/bus/iio/devices/iio\:device0/in_voltage0_raw";

    /* Open sysfs to Analogue input */
    fd = open (AIN0, O_RDONLY);

    while (1)
    {
        /* Read Analogue input */
        fdstat = read(fd, buffer, sizeof(buffer));

        /* Print result */
        if (fdstat != -1)
        {
            buffer[fdstat] = '\0';

            /* Print string and value*/
            printf("AIN0 value = %s \n", buffer);
        }
    }
}
```

```
    lseek(fd, 0, 0);  
}  
  
/* Small delay */  
sleep(1);  
}  
  
/* Close sysfs & exit */  
close(fd);  
return 0;  
}
```