fortune datko

Cookies along the way.

BeagleBone Black I2C References

NOVEMBER 3, 2013MAY 22, 2014 / JOSH

I'm planning on heavily using I2C for my CryptoCape (http://www.cryptotronix.com/products/cryptocape/). I'm still working through issues, but I wanted to capture some of my research in this post. As I'm currently experimenting, please treat this information as notes and not as definitive information. Feel free to post corrections in the comments and I'll update the content.

The BeagleBone Black (BBB) has three I2C buses (thanks to Smith Winston providing most of this information at his BeagleBoard.org discussion post (https://groups.google.com/forum/#!msg/beagleboard/v9r8UkN7klk/h3rFKFJDLnUJ)):

- i2c0: Not exposed in the expansion headers
- i2c1: pins P9 17,18 (and 24,26)
- i2c2: pins P9 19,20 (and 21,22)

But, not all three buses are exported by default and more confusingly, under linux, the buses are named in the order they are enumerated which may have nothing to do with the physical names.

So, it’s best to check the memory addresses:

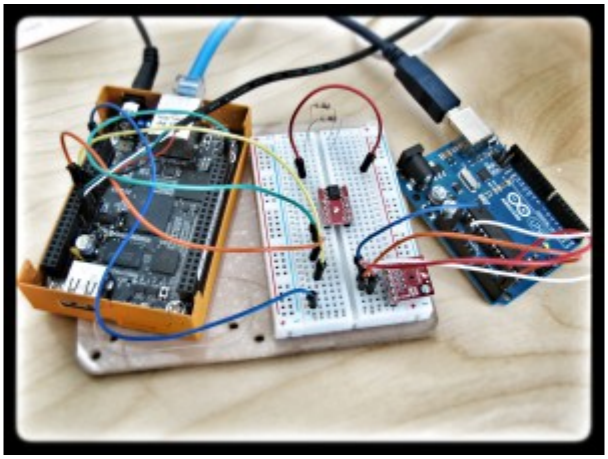
- i2c0: 0x44E0_B000
- i2c1: 0x4802_A000
- i2c2: 0x4819_C000

Then check the mapping in `/sys/bus/i2c/devices`:

```
root@beaglebone:~# ls -l /sys/bus/i2c/devices/i
lrwxrwxrwx    1 root     root                 0 Jan
lrwxrwxrwx    1 root     root                 0 Jan
```

In this example, i2c0 is mapped to `/dev/i2c-0` but i2c2 is mapped to `/dev/i2c-1`.

What the BBB System Reference Manual refers to I2C2-2, which is defaulted to pins P9-19 and P920, are probably `/dev/i2c-1` under linux.



(http://jbdatko.files.wordpress.com/2013/11/i2c_hacking.jpg)
My I2C hacking rig. BBB, Arduino, and Sparkfun TMP102 and an Atmel ECC chip soldered onto a Sparkfun SOIC 8 Lead breakout board.

Enabling the Third I2C Bus

I received an answer on this post (https://groups.google.com/forum/#!msg/beagleboard/7reYt7Tmdjs/5LLRvCTPNuoJ) about how to see all three I2C buses. The following incantation seems to do the trick, but I'm not sure if interferes with any capes (the number after the bone_capemgr can change, so check your system):

```
root@arm:~# echo BB-I2C1 > /sys/devices/bone_cap
root@arm:~# ls -l /sys/bus/i2c/devices/i2c-*
lrwxrwxrwx 1 root root 0 Nov  3 21:49 /sys/bus/i
lrwxrwxrwx 1 root root 0 Nov  3 21:49 /sys/bus/i
lrwxrwxrwx 1 root root 0 Nov  3 21:50 /sys/bus/i
```

Programming I2C

While Adafruit has a Python I2C library (http://learn.adafruit.com/setting-up-io-python-library-on-beaglebone-black/i2c) one can make straightforward C file calls as in the following example (snippet lifted from eLinux's Interfacing with I2C Devices (http://elinux.org/Interfacing_with_I2C_Devices)):

Opening the Bus:

```
#include <errno.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <linux/i2c-dev.h>
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
int file;
char *filename = "/dev/i2c-1";
if ((file = open(filename, O_RDWR)) < 0) {
    /* ERROR HANDLING: you can check errno to s
    perror("Failed to open the i2c bus");
    exit(1);
}
```

Initiating Comms:

```
int addr = 0x48;     // The I2C address of the
if (ioctl(file, I2C_SLAVE, addr) < 0) {
    printf("Failed to acquire bus access and/or
    /* ERROR HANDLING; you can check errno to s
    exit(1);
}
```

Reading from the device:

```
unsigned char buf[10] = {0};

for (int i = 0; i<4; i++) {
    // Using I2C Read
    if (read(file,buf,2) != 2) {
        /* ERROR HANDLING: i2c transaction fail
        printf("Failed to read from the i2c bus
            printf("\n\n");
    } else {
        /* Device specific stuff here */
    }
}
```

Kernel Documentation

The linux kernel has documentation on instantiating I2C devices (https://www.kernel.org/doc/Documentation/i2c/instantiating-devices).
See the last section for user space details.

Other links and projects

- Great tutorial (http://derekmolloy.ie/beaglebone/beaglebone-an-i2c-tutorial-interfacing-to-a-bma180-accelerometer/) on I2C for the BeagleBone White. Some details have changed, but otherwise, very helpful.
- Nice post on the effects of varying I2C Pull-up Resistors (http://www.dsscircuits.com/articles/effects-of-varying-i2c-pull-up-resistors.html)
- A very simple Digital Temperature I2C sensor via Sparkfun (https://www.sparkfun.com/products/11931) and associated tutorial (http://bildr.org/2011/01/tmp102-arduino/) for Arduino.

BeagleBone
ADAFRUIT INDUSTRIES **ARDUINO** **BEAGLEBONE BLACK** **I2C** **I^{2}C** **LINUX** **SPARKFUN ELECTRONICS**

One thought on “BeagleBone Black I2C References”

1. Pingback: External hardware random number generator for the BeagleBone Black | Cryptotronix, LLC

Comments are closed.