I2C not working in mraa #394

New issue

Closed **fryguy1013** opened this issue on Jan 3 · 6 comments

fryguy1013 commented on Jan 3

I'm using the Sparkfun DEV-13042 (PWM block) attached to the Sparkfun DEV-13097 (Edison Arduino breakout), and have version 0.9.0 of the mraa library.

I have this c++ code that uses mraa to read registers on the PWM block over i2c:

```
i2c = new mraa::I2c(1);
retval = i2c->address(0x40);
cout << "return (address): " << retval << endl;

int mode = i2c->readReg(MODE1);
cout << "Mode (reg 0x00): " << mode << endl;
```

Running this program shows:

```
return (address): 0
Mode (reg 0x00): 0
```

I would expect this to be 0x11, and journalctl -f shows the following when I run this program:

```
Jan 04 00:48:03 edison libmraa[656]: libmraa version v0.9.0 initialised by user 'root' with
Jan 04 00:48:03 edison libmraa[656]: libmraa initialised for platform 'Intel Edison' of typ
Jan 04 00:48:15 edison libmraa[656]: i2c: Selected bus 1
Jan 04 00:48:15 edison libmraa[656]: Invalid i2c bus, moving to default i2c bus
Jan 04 00:48:15 edison libmraa[656]: i2c: Failed to write
Jan 04 00:48:15 edison libmraa[656]: i2c: Failed to write
```

Yet, if I use the i2cdump, it's able to read the registers fine:

```
root@edison:~# i2cdump -y 1 0x40
No size specified (using byte-data access)
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f    0123456789abcdef
00: 11 04 e2 e4 e8 e0 00 00 00 10 00 00 00 10 00 00    ??????...?...?..
10: 00 10 00 00 00 10 00 00 00 10 00 00 00 10 00 00    .?...?...?...?..
20: 00 10 00 00 00 10 00 00 00 10 00 00 00 10 00 00    .?...?...?...?..
30: 00 10 00 00 00 10 00 00 00 10 00 00 00 10 00 00    .?...?...?...?..
40: 00 10 00 00 00 10 XX XX XX XX XX XX XX XX XX XX    .?...?XXXXXXXXXX
50: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX    XXXXXXXXXXXXXXXX
60: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX    XXXXXXXXXXXXXXXX
70: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX    XXXXXXXXXXXXXXXX
80: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX    XXXXXXXXXXXXXXXX
90: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX    XXXXXXXXXXXXXXXX
a0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX    XXXXXXXXXXXXXXXX
b0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX    XXXXXXXXXXXXXXXX
c0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX    XXXXXXXXXXXXXXXX
d0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX    XXXXXXXXXXXXXXXX
e0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX    XXXXXXXXXXXXXXXX
f0: XX XX XX XX XX XX XX XX XX XX 00 00 00 00 1e 00    XXXXXXXXXX....?.
```

Labels

documentation

question

Milestone

No milestone

Assignees

arfoll

3 participants

This is likely the same issue as #187, but it was requested I open a new issue instead.

fryguy1013 commented on Jan 3

I have no way of testing this right now since I'm on Windows and don't have the stuff installed to compile cmake projects, but I suspect that this is the problem: fryguy1013@ `c64999b` . I would open a PR if I were more confident this fixed my problem, but perhaps it was intentionally not in the list.

Also, I noticed that my board didn't have a /sys/class/i2c-dev/i2c-0, but the code in mraa.c::mraa_find_i2c_bus(...) assumes it exists, even if `startfrom` isn't 0. I don't think this is my problem though, as that function seems to only be used in other boards. I'm just mentioning it as the other boards might behave as mine do and not have the i2c-0 path.

fryguy1013 commented on Jan 3

I think the pull request works, because I made this function and called it, and it seemed to make the i2c work for me:

```
void HACK_fix_it()
{
    plat->i2c_bus[1].bus_id = 1;
    plat->i2c_bus[1].sda = 7;
    plat->i2c_bus[1].scl = 19;
}
```

fryguy1013 referenced this issue on Jan 3

Add I2C bus 1 to intel edison with arduino #395 Closed

arfoll commented on Jan 4 Intel iot-devkit libraries member

Ah yes I understand the issue now, we don't support people stacking stuff on the arduino breakout though there is no good reason for not supporting it since as you've found the pins are there. Your PR won't work because we have checks to make sure that only bus 6 can be used on !miniboard (see intel_edison_i2c_init_pre()). I'll try get this working today.

alext-mkrs added the `bug` label on Jan 4

arfoll commented on Jan 4 Intel iot-devkit libraries member

Actually I'm going to go back on my word since adding i2c bus 1 on arduino is going to cause confusion, in this case please use the raw mode of mraa, I tried it using the sparkfun i2c breakout and it works well, just do something like

```
import mraa as m
x = m.I2c(1, True)
x.address(myaddr)
x.writeReg(0, 0)
```

Is that ok for you?

arfoll added `enhancement` `question` labels on Jan 4

arfoll self-assigned this on Jan 4

fryguy1013 commented on Jan 4

I made it work yesterday by including the mraa source into my project and changing the intel_edison_i2c_init_pre() function() and the other change in my PR. I didn't update my PR yet, but I'll put finish it off even though you won't put it in master, so if someone else like me sees this bug report they can just take my branch and fix their problem as well, even though it's not officially supported.

I don't really understand why it would cause confusion to allow i2c 1. Of course, I don't know what i2c 6 is used for.

alext-mkrs removed the bug label on Jan 4

arfoll commented on Jan 5 — Intel iot-devkit libraries member

It's confusing because it's not a normally exposed bus on the arduino board and the arduino board is not meant to be 'stacked' on. I don't really want it showing during a scan and then people getting confused why they can't plug their jumper to it etc... i2c-6 is the I2c exposed on the SCL/SDA pins of the edison arduino breakout.

Using the raw mode to override the safety is the right approach in this case and there's no real downside of doing this that way. I'll add something in the docs to mention it because as you say it may be useful.

arfoll added a commit that closed this issue on Jan 5

`edison.md: Add notes on using i2c-1 w/Arduino breakout …` `7f85f2b`

arfoll closed this in `7f85f2b` on Jan 5

alext-mkrs added documentation and removed enhancement labels on Jan 5

fryguy1013 referenced this issue on May 31

SparkFun 9DOF Block is detected by mraa as Edison kit for Arduino, causing I2C errors #504 — Closed