Connect Colibri T30 & Adafruit PWM Driver

Michal Zajacik

2015-06-22

Contents

1 Pinout of connection

Iris X16	Adafruit
12	VCC
5	SDA
6	SCL
7	GND

2 Powersource

2.1 Capacitor

There is a free capacitor spot on Adafruit, recommended to use n*100uF, where n is # of servos. Handy in case of lot of drain from unstable source. We do not need this at the moment.

2.2 Power

V+ connector on Adafruit is for powering servos. Either from shielded connector, or from V+ pin next to VCC on control header. Temporarily I have connected V+ and VCC (servos and Adafruit is powered from 5V pin of Iris). This, hovever, is not the recommended way. But it works.

3 System

Default device on Toradex T30 is /dev/i2c-0, open it for read, write with plain C open() function. It is possible to determine which device and with what address is connected with a command

```
i2cdetect -y -r i2c_device_number
```

Output:

```
root@colibri-t30:~# i2cdetect -y -r 0
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: 20 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- UU -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Which means that for device /dev/i2c-0 there is on address 0x20 (row + column) connected some i2c device. UU means, that this is system address and thus probably not the device you want.

4 Driver

Driver consists of set of constants with addresses of registers and command values. First, device is open with plain C function open(), then the device is bind as a slave, so the kernel i2c driver knows how to handle reading and writing. Only 1 BYTE reads and writes are requied.

To set a PWM, just create the PWM object (default constructor takes the bus /dev/i2c-0 and device on address 0x40), run function open device,

set PWM frequency and then set PWM for the servo. Don't forget to close the device after the work is done :)

Example of code:

```
PWMDriver pwm;

pwm.openDevice();
pwm.setPWMFrequency(60);

pwm.setPWM(0, 3000, 0);
usleep(1000000);

pwm.setPWM(0, 0, 0);

pwm.closeDevice();
```