e-ale-scale17x

Embedded Apprentice Linux Engineer SCaLE17x

Version 1.0

e-ale

The E-ALE (Embedded Apprentice Linux Engineer) is a series of seminars held at existing conferences covering topics which are fundamental to a Linux professional in the field of Embedded Linux.

This seminar will spend equal time on lecture and hands on labs at the end of each seminar which allow you to practice the material you've learned.

> This material makes the assumption that you have minimal experience with using Linux in general, and a basic understanding of general industry terms. The assumption is also made that you have access to your own computers upon which to practice this material.

More information can be found at **https://e-ale.org/**

Contents

Chapter 1

I2C

e-ale

1.1 I2C

I2C from Userspace

- Speaker: Michael Welling <mwelling@ieee.org>
- SCaLE17x (2019.03.08)

Based on I2C slides presented by Grant Likely at LCA2019.

I2C

- **I2C** (Inter-Integrated Circuit), pronounced I-squared-C is a bus designed to communicate between chips on a board
- It is also known by the name **IIC**, **TWI** and **smbus** (although smbus is strictly a subset of the I2C specification)
- I2C was initially developed by Philips, however today owned by NXP Semicondictors
- Other companies, who have used I2C-like protocols have used the other listed names (for various reasons)

Two Wire Interface

- The common alternate name of **TWI** (Two Wire Interface) literally describes the physical layout of the bus
- I2C is made up of 2 wires:
  - A data line called SDA
  - A clock line called SCL

Synchronous bus

- I2C is a synchronous, multi-master/slave, packet based bus
- The 2 wires (SDA/SCL) are bidirectional **open collector** or **open drain** lines
- I2C tyipcally runs at 5V or 3.3V, though other voltages are permitted

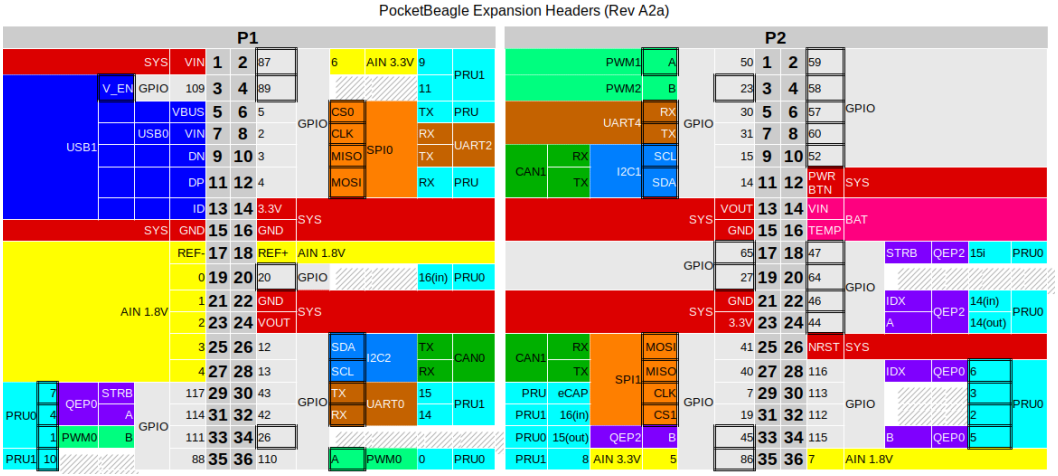
I2C speeds

- It was initially imagined for low speed inter-chip communications
- Although implemented in HW, the lowspeed version of I2C can still be implemented via GPIO using **bit-banging**, although the HW version is vastly preferred.
- HW based versions can run at speeds up to 5 Mbps

I2C addressing

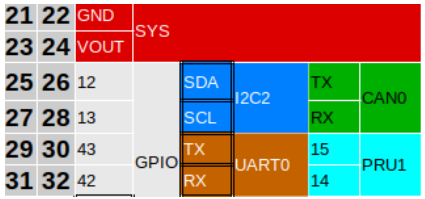
- I2C uses a 7-bit address space (though there is a rarely used 10-bit extension)
- I2C based devices usually have a range of selectable addresses which allow you to have more than one I2C device on the same bus, or more than one of the same kind of I2C device (with a different address)

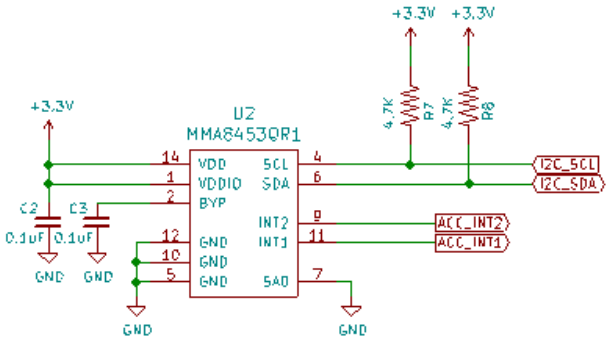
PocketBeagle Pins



- Pins are shared amongst multiple peripherals
- A pin multiplexer is used to choose the configuration of the pins in use.

Pocketbeagle/Baconbits I2C





- We can access I2C devices via the I2C pins
- We will use the I2C bus to talk to the **MMA8453QR1 3-axis, 10-bit accelerometer** on the Baconbits cape.

I2C devices on Linux

- From the Linux userspace, you can access the I2C bus from the `/dev/i2c-*` device files

```
debian@beaglebone:~$ ls -l /dev/i2c-*
crw-rw---- 1 root i2c 89, 0 Oct  7 16:40 /dev/i2c-0
crw-rw---- 1 root i2c 89, 1 Oct  7 16:40 /dev/i2c-1
crw-rw---- 1 root i2c 89, 2 Oct  7 16:40 /dev/i2c-2
```

List I2C devices on the bus

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

- Notice we can see a single device at 0x1c

Dump the register contents of MMA8453

```
debian@beaglebone:~$ i2cdump -y -r 0x00-0x31 2 0x1c
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: ff fe 00 01 80 41 80 00 00 00 00 01 00 3a 00 00    .?.??A?....?.:..
10: 00 80 00 44 84 00 00 00 00 00 00 00 00 00 00 00    .?.D?...........
20: 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00    ..........?.....
30: 00 00
```

- Notice we can see 0x3a indentifying the MMA8453 device.

Read and write single registers of the MMA8453

```
debian@beaglebone:~$ i2cget -y 2 0x1c 0x0d
0x3a
debian@beaglebone:~$ i2cget -y 2 0x1c 0x2a
0x00
debian@beaglebone:~$ i2cset -y 2 0x1c 0x2a 0x01
debian@beaglebone:~$ i2cget -y 2 0x1c 0x2a
0x01
```

- Setting the register 0x2a starts the accelerometer conversion

Labs

- LAB 1 - Investigate i2ctools
- LAB 2 - Write python module to read sensor
- LAB 3 - Emulate mouse with accelerometer