Day 02-2

Assignment:

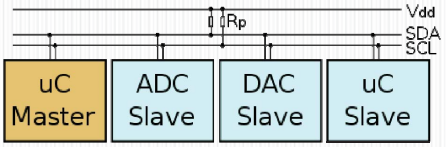
- Hw 02, Due Tuesday
- Hw 03, Due Next Tuesday
-

Today's Topics:

- Project Ideas
- I2C
- LED Matrix

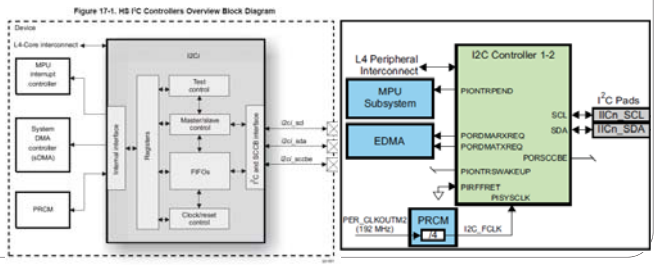
02-2 I2C

Interfacing with sensors over a serial bus



I²C

- "two-wire interface" standard
- Used to attach low-speed peripherals to embedded systems
- The Bone has three I²C controllers (Section 21 of TRM)



Hardware - Bone

Display Kernel Log Messages

- You can see which ones are configured at boot time

```
bone$ dmesg | grep i2c
[    1.577027] omap_i2c 44e0b000.i2c: could not find pctldev for node
/ocp/l4_wkup@44c00000/scm@210000/pinmux@800/pinmux_i2c0_pins, deferring
probe
[    1.577893] omap_i2c 4802a000.i2c: bus 1 rev0.11 at 400 kHz
[    1.578556] omap_i2c 4819c000.i2c: bus 2 rev0.11 at 400 kHz
[    2.200784] i2c /dev entries driver
[    2.438435] input: tps65217_pwr_but as
/devices/platform/ocp/44e0b000.i2c/i2c-0/0-0024/input/input0
[    2.464756] omap_i2c 44e0b000.i2c: bus 0 rev0.11 at 400 kHz
```

Three buses each running at same speeds

dmesg -Hw

Human readable

Wait for more input

```
[  +0.147158] IPv6: ADDRCONF(NETDEV_CHANGE): SoftAp0: link becomes ready
[Sep 8 08:20] fbtft: module is from the staging directory, the quality is unknown,
you have been warned.
[  +0.016886] fbtft_device: module is from the staging directory, the quality is
unknown, you have been warned.
[  +0.001700] spidev spi1.0: spidev spi1.0 24000kHz 8 bits mode=0x00
[  +0.000026] spidev spi1.1: spidev spi1.1 24000kHz 8 bits mode=0x00
[  +0.000188] spidev spi1.0: Deleting spi1.0
[  +0.009426] fbtft_device: GPIOS used by 'adafruit28':
[  +0.000026] fbtft_device: 'reset' = GPIO113
[  +0.000008] fbtft_device: 'dc' = GPIO116
[  +0.000022] spidev spi1.1: spidev spi1.1 24000kHz 8 bits mode=0x00
[  +0.000014] spi spi1.0: fb_ili9341 spi1.0 32000kHz 8 bits mode=0x00
[  +0.043846] fb_ili9341: module is from the staging directory, the quality is
unknown, you have been warned.
[  +0.344838] Console: switching to colour frame buffer device 40x30
[  +0.000694] graphics fb0: fb_ili9341 frame buffer, 320x240, 150 KiB video
memory, 16 KiB DMA buffer memory, fps=20, spi1.0 at 32 MHz
[Sep 8 08:22] usb 1-1.4: USB disconnect, device number 8
[Sep 8 08:34] usb 1-1.1: USB disconnect, device number 3
[  +0.000030] usb 1-1.1.1: USB disconnect, device number 9
```

Hardware - Bone

- You can see which ones are configured at boot time

```
bone$ dmesg | grep i2c
[    1.577027] omap_i2c 44e0b000.i2c: could not find pctldev for node
/ocp/l4_wkup@44c00000/scm@210000/pinmux@800/pinmux_i2c0_pins, deferring
probe
[    1.577893] omap_i2c 4802a000.i2c: bus 1 rev0.11 at 400 kHz
[    1.578556] omap_i2c 4819c000.i2c: bus 2 rev0.11 at 400 kHz
[    2.200784] i2c /dev entries driver
[    2.438435] input: tps65217_pwr_but as
/devices/platform/ocp/44e0b000.i2c/i2c-0/0-0024/input/input0
[    2.464756] omap_i2c 44e0b000.i2c: bus 0 rev0.11 at 400 kHz
```

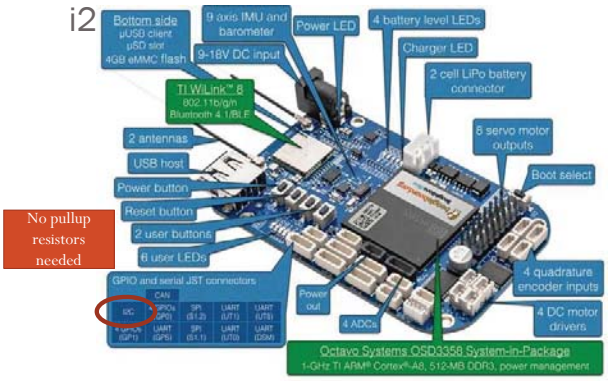
Three buses each running at same speeds

Time in seconds

Sept-2016

```
[    2.105100] i2c /dev entries driver
[    2.228438] input: tps65217_pwr_but as
/devices/platform/ocp/44e0b000.i2c/i2c-0/0-0024/input/input0
[    2.258422] omap_i2c 44e0b000.i2c: bus 0 rev0.11 at 400 kHz
[    2.262395] omap_i2c 4819c000.i2c: bus 2 rev0.11 at 100 kHz
[   94.721560] omap_i2c 4802a000.i2c: bus 1 rev0.11 at 100 kHz
```

i2



The 0th I2C bus can't be used for other digital I/O operations without interfering with that function, but you can still use it to add other I2C devices at available addresses.
The 1st I2C bus is available for you to configure and use.

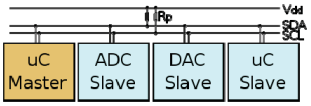
Hardware – TMP101

- Goal: Interface to a TMP101 temp sensor

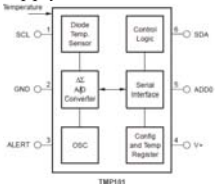
Parameter Name	Value
Typical Accuracy (°)	±2.0°C from −25°C to +85°C (max) ±3.0°C from −55°C to +125°C (max)
Supply Current (µA)	45µA, 0.1µA Standby
Resolution	9- to 12-bits,
Operating Voltage Range (V)	2.7V to 5.5V
Device Description	Serial Output Temp Sensor

http://www.ti.com/lit/gpn/tmp101

2-wire bus



- The two wires are
  - Serial Clock (SCL), is an input to the TMP101 and is used to clock data into and out of the TMP101.
  - Serial Data (SDA), is bidirectional and carries the data to and from the TMP101.
- The only other two pins on the TMP101 that you need to use are the Power Supply (Vdd) and Ground.



Software - bone

Bus number

- See what's on a bus with **`i2cdetect`**

```
bone$ i2cdetect -y -r 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- 48 49 -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: 70 -- -- -- -- -- -- --
```

I have 2, TMP101's and an LED matrix.

- The TMP101's are at **`1001 000`** and **`1001 001`**
- Convert to hex **`0x48`** and **`0x49`**

Registers

- Each TMP101 has four registers

Table 2. Pointer Addresses of the TMP100 and TMP101 Registers

P1	P0	REGISTER
0	0	Temperature Register (READ Only)
0	1	Configuration Register (READ/WRITE)
1	0	T_{LOW} Register (READ/WRITE)
1	1	T_{HIGH} Register (READ/WRITE)

- Read with **`$ i2cget -y 1 0x48 00`**
- **`0x18`** which is 24C or 75.2F

Table 6. Configuration Register Format

BYTE	D7	D6	D5	D4	D3	D2	D1	D0
1	OS/ALERT	R1	R0	F1	F0	POL	TM	SD

Registers

Table 2. Pointer Addresses of the TMP100 and TMP101 Registers

P1	P0	REGISTER
0	0	Temperature Register (READ Only)
0	1	Configuration Register (READ/WRITE)
1	0	T_{LOW} Register (READ/WRITE)
1	1	T_{HIGH} Register (READ/WRITE)

- Read with `$ i2cget -y 1 0x48 01`
- `0x80` which is `1000 0000`

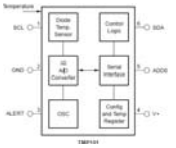


Table 6. Configuration Register Format

BYTE	D7	D6	D5	D4	D3	D2	D1	D0
1	OS/ALERT	R1	R0	F1	F0	POL	TM	SD

SD – Shutdown Mode
TM - Thermostat Mode
POL-Polarity
F1/F0 – Fault Queue
R1/R0 – Converter Resolution
OS – OS/Alert

Table 8. Resolution of the TMP100 and TMP101

R1	R0	RESOLUTION	CONVERSION TIME (typical)
0	0	9 Bits (0.5°C)	40ms
0	1	10 Bits (0.25°C)	80ms
1	0	11 Bits (0.125°C)	160ms
1	1	12 Bits (0.0625°C)	320ms

Read TMP101

```
#!/usr/bin/env python3
# Read a TMP101 sensor
# sudo apt install python3-smbus

import smbus
import time

bus = smbus.SMBus(1)
address = 0x49

while True:
    temp = bus.read_byte_data(address, 0)
    print (temp, end="\r")
    time.sleep(0.25)
```

Must install

Bus Number

Address of device

Register to read

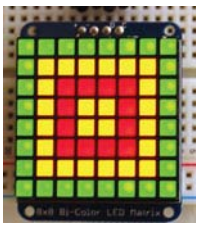
- smbus commands

```
read_byte_data(int addr,char cmd)
write_byte_data(int addr,char cmd, char val)
write_byte_data(int addr,char cmd, char val)
write_i2c_block_data(int addr, char cmd, long vals[])
```

http://www.raspberry-projects.com/pi/programming-in-python/i2c-programming-in-python/using-the-i2c-interface-2

LED Matrix

- In the lab you will be interfacing an I2C LED matrix
- Some are bicolor (red and green)
- Some are single color
- Both interface the same way: i2c
- How many wires to you need to control 128 LEDs?



https://www.adafruit.com/products/902

https://www.adafruit.com/products/871

exercises/displays/matrix8x8/i2cmatrix.py

```
#!/usr/bin/env python3
import smbus
import time
bus = smbus.SMBus(1)  # Use i2c bus 1
matrix = 0x70         # Use address

# The first byte is GREEN, the second is RED.
smile = [0x00, 0x3c, 0x00, 0x42, 0x28, 0x89, 0x04, 0x85,
    0x04, 0x85, 0x28, 0x89, 0x00, 0x42, 0x00, 0x3c
]
frown = [0x3c, 0x00, 0x42, 0x00, 0x85, 0x20, 0x89, 0x00,
    0x89, 0x00, 0x85, 0x20, 0x42, 0x00, 0x3c, 0x00
]
neutral = [0x3c, 0x3c, 0x42, 0x42, 0xa9, 0xa9, 0x89, 0x89,
    0x89, 0x89, 0xa9, 0xa9, 0x42, 0x42, 0x3c, 0x3c
]
```

exercises/displays/matrix8x8/i2cmatrix.py

```
# Start oscillator (p10)
bus.write_byte_data(matrix, 0x21, 0)
# Disp on, blink off (p11)
bus.write_byte_data(matrix, 0x81, 0)
# Full brightness (page 15)
bus.write_byte_data(matrix, 0xe7, 0)
```

exercises/displays/matrix8x8/i2c

```
bus.write_i2c_block_data(matrix, 0, frown)
time.sleep(delay)

bus.write_i2c_block_data(matrix, 0, neutral)

for fade in range(0xe0, 0xef, 1):
    bus.write_byte_data(matrix, fade, 0)
    time.sleep(delay/10)

bus.write_i2c_block_data(matrix, 0, smile)
```

Writing one column

```
bus.write_i2c_block_data(matrix,
                         0x04, [0xff]);
```

- Turns on the third column of *green* LEDs, without writing the other columns
- Try it!

LED Matrix

- Goal: Etch-a-Sketch on the LED Matrix!