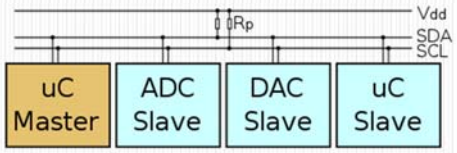
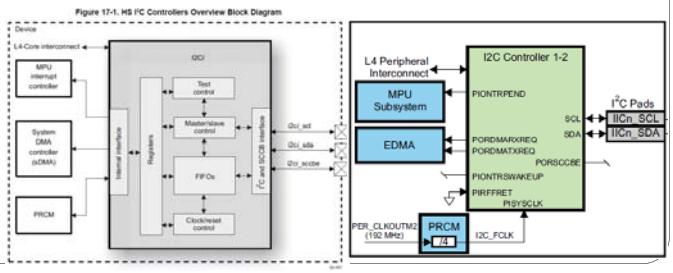
01-4 I2C

Interfacing with sensors over a serial bus



I²C

- "two-wire interface" standard
- Used to attach low-speed peripherals to embedded systems
- Beagle xM has four I²C controllers (Section 17 of TRM)
- The Bone has three (Section 21)



Hardware - xM

- You can see which ones are configured at boot time

```
beagle$ dmesg | grep i2c
[    0.082092] omap_i2c omap_i2c.1: bus 1 rev4.0 at 2600 kHz
[    0.100433] omap_i2c omap_i2c.3: bus 3 rev4.0 at 100 kHz
[    2.294616] input: twl4030_pwrbutton as
/devices/platform/omap/omap_i2c.1/i2c-1/1-0049/twl4030_pwrbutton/input/input1
[    2.295440] i2c /dev entries driver
```

Two buses each running at different speeds

Time in seconds

Hardware - bone

- You can see which ones are configured at boot time

```
beagle$ dmesg  | grep i2c
[    0.069359]  omap_i2c.1: alias fck already exists
[    0.085082] omap_i2c omap_i2c.1: bus 1 rev2.4.0 at 100 kHz
[    0.259664]  omap_i2c.3: alias fck already exists
[    0.259942] omap_i2c omap_i2c.3: bus 3 rev2.4.0 at 100 kHz
[    0.641936] i2c /dev entries driver
```

- Two buses, same speed.

i2c - bone

Table 11. Expansion Header P9 Pinout

SIGNAL NAME	PIN	CONN		PIN	SIGNAL NAME
	GND	1	2	GND	
	VDD_3V3EXP	3	4	VDD_3V3EXP	
	VDD_5V	5	6	VDD_5V	
	SYS_5V	7	8	SYS_5V	
PWR_BUT*		9	10	A10	SYS_RESETn
UART4_RXD	T17	11	12	U18	GPIO1_28
UART4_TXD	U17	13	14	U14	EHRPWM1A
GPIO1_16	R13	15	16	T14	EHRPWM1B
I2C1_SCL	A16	17	18	B16	I2C1_SDA
I2C2_SCL	D17	19	20	D18	I2C2_SDA
UART2_TXD	B17	21	22	A17	UART2_RXD
GPIO1_17	V14	23	24	D15	UART1_TXD
GPIO3_21	A14	25	26	D16	UART1_RXD
GPIO3_19	C13	27	28	C12	SPI1_CS0
SPI1_D0	B13	29	30	D12	SPI1_D1
SPI1_SCLK	A13	31	32	VDD_ADC(1.8V)	
AIN4	C8	33	34	GNDA_ADC	
AIN6	A5	35	36	A5	AIN5
AIN2	B7	37	38	A7	AIN3
AIN0	B6	39	40	C7	AIN1
CLKOUT2	D14	41	42	C18	GPIO0_7
	GND	43	44	GND	
	GND	45	46	GND	

Pin MUX

- Is the MUX set to output i2c?

```
beagle$ cd /sys/kernel/debug/omap_mux
beagle$ ls | grep i2c
i2c0_scl
i2c0_sda
beagle$ grep i2c2_sda *
spi0_sclk:signals:
        spi0_sclk | uart2_rxd | i2c2_sda | NA | NA | NA | NA | gpio0_2
uart0_rxd:signals:
        uart0_rxd | spi1_cs0 | d_can0_tx | i2c2_sda | NA | NA | NA | gpio1_10
uart1_ctsn:signals:
        uart1_ctsn | NA | d_can0_tx | i2c2_sda | spi1_cs0 | NA | NA | gpio0_12
```

- Which one is it?

Pin MUX

- Cat each file to see

```
beagle$ cat spi0_sclk
name: spi0_sclk.gpio0_2 (0x44e10950/0x950 = 0x0037), b NA, t NA
mode: OMAP_PIN_OUTPUT | OMAP_MUX_MODE7
signals: spi0_sclk | uart2_rxd | i2c2_sda | NA | NA | NA | NA | gpio0_2
beagle$ cat uart0_rxd
name: uart0_rxd.uart0_rxd (0x44e10970/0x970 = 0x0030), b NA, t NA
mode: OMAP_PIN_OUTPUT | OMAP_MUX_MODE0
signals: uart0_rxd | spi1_cs0 | d_can0_tx | i2c2_sda | NA | NA | NA |
gpio1_10
beagle$ cat uart1_ctsn
name: uart1_ctsn.i2c2_sda (0x44e10978/0x978 = 0x0033), b NA, t NA
mode: OMAP_PIN_OUTPUT | OMAP_MUX_MODE3
signals: uart1_ctsn | NA | d_can0_tx | i2c2_sda | spi1_cs0 | NA | NA |
gpio0_12
```

Hardware – TC74

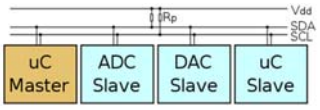
- Goal: Interface to a TC74 temp sensor

Parameter Name	Value
Typical Accuracy (°)	0.5
Max Input/ Supply Current (µA)	350
Max. Accuracy @ 25° (°)	2
Temp. Range (°C)	-40 to +125
Operating Voltage Range (V)	2.7 to 5.5
Device Description	Serial Output Temp Sensor

http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010749#1

2-wire bus

- The two wires are
  - Serial Clock (SCLK on the data sheet, SCL on the Beagle), is an input to the TC74 and is used to clock data into and out of the TC74.
  - Serial Data (SDA on both), is bidirection and carries the data to and from the TC74.
- The only other two pins on the TC74 that you need to use are the Power Supply (Vdd) and Ground.



Software - bone

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

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

I have 2, TMP102's and an LED matrix.

- The TMP102's are at **1001 000** and **1001 001**
- Convert to hex **0x48** and **0x49**

Registers

- Each TC74 has two registers

Command	Code	Function
RTR	0x00	Read Temperature (TEMP)
RWCR	0x01	Read/Write Configuration (CONFIG}

- Read with **$ i2get -y 3 0x48 0**
- **0x18** which is 24C or 75.2F

I²C via C – myi2cget.c

```
int main(int argc, char *argv[]) {
        char *end;
        int res, i2cbus, address, size, file;
        int daddress;
        char filename[20];

        /* handle (optional) flags first */
        if(argc < 3) {
                fprintf(stderr,
                        "Usage:  %s <i2c-bus> <i2c-address> <register>\n",
                        argv[0]);
                exit(1);
        }
        i2cbus   = atoi(argv[1]);
        address  = atoi(argv[2]);
        daddress = atoi(argv[3]);
        size = I2C_SMBUS_BYTE;
```

I²C via C

```
sprintf(filename, "/dev/i2c-%d", i2cbus);
file = open(filename, O_RDWR);
if (file < 0) {
    if (errno == ENOENT) {
        fprintf(stderr, "Error: Could not open file "
        "/dev/i2c-%d: %s\n", i2cbus, strerror(ENOENT));
    } else {
        fprintf(stderr, "Error: Could not open file "
            "`%s': %s\n", filename, strerror(errno));
        if (errno == EACCES)
            fprintf(stderr, "Run as root?\n");
    }
    exit(1);
}
```

I²C via C

```
if (ioctl(file, I2C_SLAVE, address) < 0) {
    fprintf(stderr,
            "Error: Could not set address to 0x%02x: %s\n",
                    address, strerror(errno));
    return -errno;
}

res = i2c_smbus_write_byte(file, daddress);
if (res < 0) {
    fprintf(stderr, "Warning - write failed, filename=%s,
            daddress=%d\n", filename, daddress);
}
res = i2c_smbus_read_byte_data(file, daddress);
close(file);
```

myi2ctest

- See **`exercises/i2c/myi2ctest.c`** for an example that controls an LED grid
- See **`exercises/i2c/i2c-tools-3.1.0`** for source code for ic2 tools