I2C Sysfs and Device Node APIs

Using I2C from User Applications

Bill Gatliff
`bgat@billgatliff.com`

Freelance Embedded Systems Developer

Overview

Roadmap:

- What is “i2c”?
- The `sysfs` interface
- The device node interface
- Examples

What is “I2C”?

Inter-Integrated Circuit bus:

- Multi-master serial bus
- Two wires, low-speed operation
- SCL and SDA
- “Two-wire Interface” (TWI)

SMBus:

- Subset of official i2c
- Stricter electrical, protocol specifications

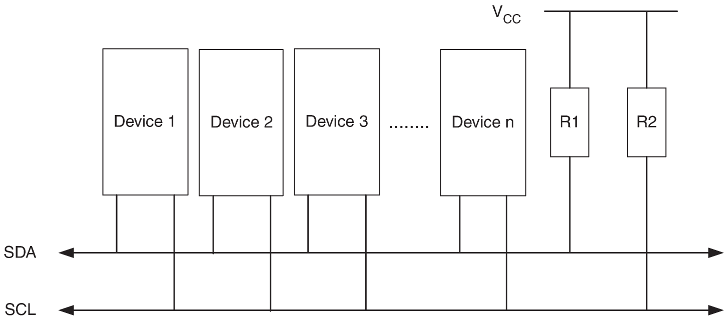
What is “I2C”?

Manifestations:

- Memory devices
- Sensors, data converters
- Battery charging circuits
- General-purpose, low-cost I/O

What is “I2C”?

Figure 21-1. TWI Bus Interconnection



What is “I2C”?

Arbitration:

- Any device can be a master!
- Any device can be a slave!

Protocol design is key:

- The hardware can detect access collisions
- You must prevent them in order to transfer data

What is “I2C”?

Pull-up resistors:

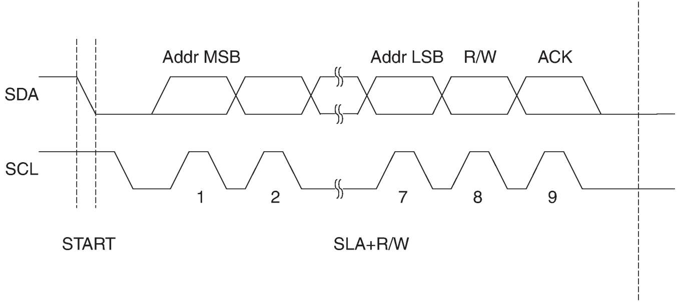
- Important (both size and presence)!
- Devices only pull lines to GND
- Clock stretching, ACK, etc.

The source of many reported problems:

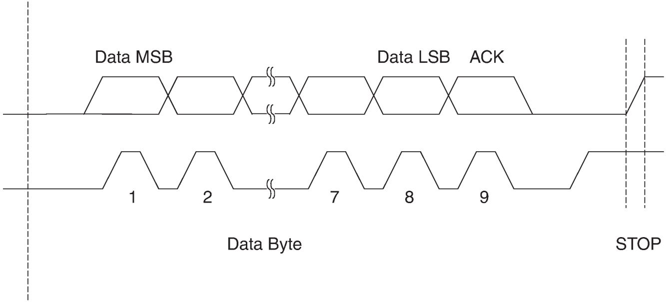
- “What pull-up resistors?”

What is “I2C”?

Figure 21-6. Typical Data Transmission



What is “I2C”?



Using I2C

Requires a bus adapter:

- Bi-directional GPIO can be made to work
- True bus adapter peripherals are ideal

Challenging features:

- Address recognition, especially during sleep
- Arbitration, collision detection
- Cycle “stretching”

Using I2C

Linux provides:

- Some bus adapter peripheral drivers
- A basic, “bit-bang” GPIO implementation
- Kernel API

And for user applications:

- Interface based on `char` device nodes, `ioctl()`
- Feedback via `sysfs` attributes (`/sys/class/i2c-adapter/...`)

Using I2C

```
# ls -CF /sys/class/i2c-adapter
i2c-0/
# ls -CF /sys/class/i2c-adapter/i2c-0
0-005a/  device@  i2c-dev:i2c-0@  name  subsystem@  uevent
# ls -CF /sys/class/i2c-adapter/i2c-0/0-005a/
bus@  modalias  name  subsystem@  uevent

# ls -CF /sys/class/i2c-dev/
i2c-0/
# ls -CF /sys/class/i2c-dev/i2c-0/
dev  device@  name  subsystem@  uevent
# cat /sys/class/i2c-dev/i2c-0/dev
89:0
```

Using I2C

A lot of information:

- What adapters are available
- What their major, minor numbers are
- What devices are connected to each adapter, if known

Remember:

- The “device” is the bus adapter!
- Each “chip” is a member of the bus
- (Chips can of course offer their own interfaces)

Using I2C

To communicate with a chip:

- Call `open()` on the adapter's device node
- Use `ioctl()` to specify chip address
- Use `ioctl()` to read, write the chip

`#include<i2c-dev.h>`

- For `i2c_smbus_read_byte()`, etc.
- See `lm-sensors` project, `i2c-tools` source code

jmbapp.c

```
#include <fcntl.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>

#include "i2c-dev.h"

int main (void)
{
  int file;
```

jmbapp.c

```c
int adapter_nr = 12;
char filename[20];

snprintf(filename, sizeof(filename),
         "/dev/i2c-%d", adapter_nr);
file = open(filename, O_RDWR);
if (file < 0) {
  perror("Could not open device");
  exit(1);
}
```

jmbapp.c

```
    if (ioctl(file, I2C_SLAVE, addr) < 0) {
      perror("Could not set I2C_SLAVE");
      exit(2);
    }

    __s32 v = 0xdeadbeef;
```

jmbapp.c

```
    v = i2c_smbus_read_byte(file);
    if (v < 0) {
      perror("i2c_smbus_read_word failed (2)");
      exit(3);
```

jmbapp.c

```
# gcc -Wall -g -O2 -I ./ -o jmbapp jmbapp.c

# ./jmbapp
i2c_smbus_read_byte: de
# ./jmbapp
i2c_smbus_read_byte: ad
# ./jmbapp
i2c_smbus_read_byte: be
# ./jmbapp
i2c_smbus_read_byte: ef
```

Recap

The I2C sysfs interface:

- Informational
- Tells about adapters, drivers and devices

The I2C device (node) interface:

- Interface to an i2c bus adapter
- Uses `ioctl()` to talk to chips
- Needs `#include<i2c-dev.h>`

I2C Sysfs and Device Node APIs

Using I2C from User Applications

Bill Gatliff
`bgat@billgatliff.com`

Freelance Embedded Systems Developer

Assignment

Read sensor status byte:

- Adapter 0, bus address 0x40
- See `jmbapp.c`

Format:

- Low nibble is actual state
- High nibble is previous state