I2C

Chris Simmonds

Embedded Linux Conference Europe 2011

Overview

- The I2C bus
- Access from user space
- Writing code to access I2C

I2C

- Stands for “Inter IC” = IIC = I-squared-C
- Simple 2-wire bus for communicating short distances
- Relatively slow (100 KHz standard, 400 KHz fast option)
- SMBus (System Management Bus) is a sub-set of I2C: most SMBus devices will work with I2C
- Typical uses
  - Communicating with temperature monitoring and power management chips
  - Controlling camera sensor chips

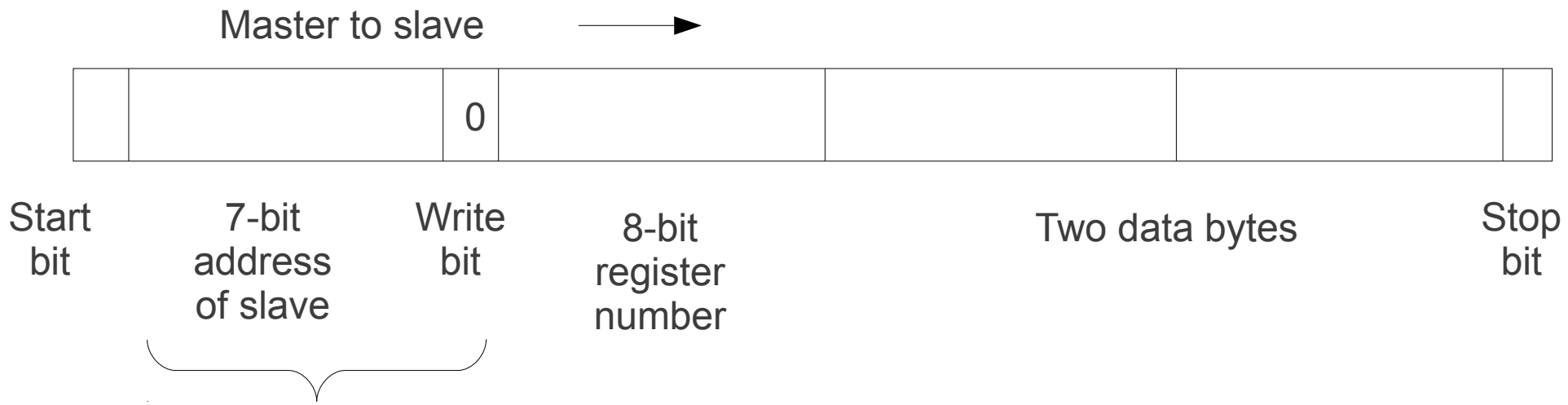
I2C Addresses

- Master-slave protocol
- 7-bit address: 128 nodes per bus
  - 16 reserved addresses, 112 nodes in practice
- peripheral addresses set by manufacturer
  - read the data sheet!

I2C transaction: write

This is a typical I2C write operation. In this case the peripheral chip expects 3 bytes of data

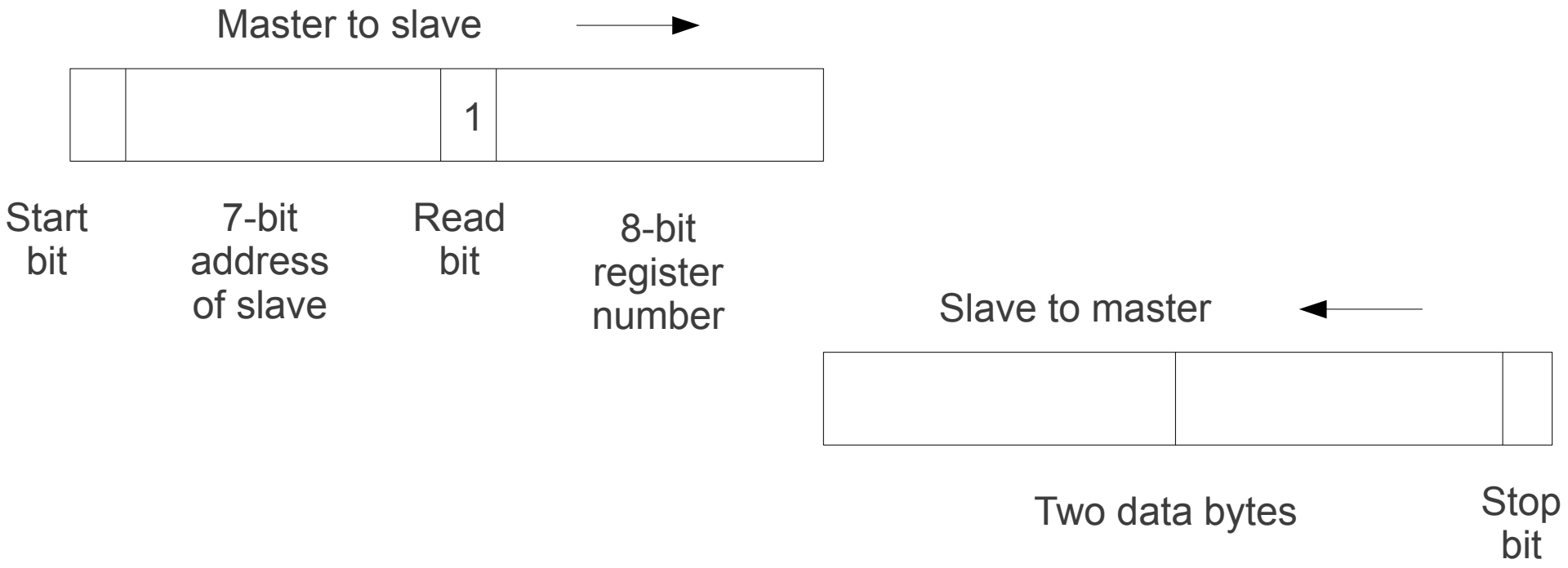
byte 0: Register address, 0..255
bytes 1 and 2: 16 bits of data



The address and read/write bit are often expressed as a single byte so that the bottom bit of the address is 0 for write and 1 for read

I2C transaction: read

To read, the master sends a read request, slave responds with data bytes



More complex transactions

- The previous slides illustrate a 3 byte transfer
  - Could be up to 32 bytes
- Some devices may allow auto-increment addressing
- Some allow back-to-back transactions
  - instead of sending a stop bit you send a start bit to begin the next transaction

User space: /dev/i2c*

- A typical embedded device has several I2C buses each connected to an I2C adapter
- Each adapter has an entry in /dev
  - For example, this has 4:

```
# ls -l /dev/i2c*
crw-rw---- 1 root i2c 89, 0 Jan  1 00:18 /dev/i2c-0
crw-rw---- 1 root i2c 89, 1 Jan  1 00:18 /dev/i2c-1
crw-rw---- 1 root i2c 89, 2 Jan  1 00:18 /dev/i2c-2
crw-rw---- 1 root i2c 89, 3 Jan  1 00:18 /dev/i2c-3
```

The i2c-utils package

- i2c-utils contains
  - **i2cdetect** - list i2c adapters and probe bus
  - **i2cdump** - dump data from all registers of an I2C peripheral (warning: dangerous!!)
  - **i2cget** - read data from an I2C device
    - i2cget <bus> <chip> <register>
  - **i2cset** - write data to an I2C device
    - i2cset <bus> <chip> <register> <value>

Information in /sys/bus/i2c

/sys/bus/i2c/devices lists all adapters and i2c slave devices

```
# ls /sys/bus/i2c/devices/
0-0033  0-0042  2-0018  2-0029  2-0034  2-0068  3-005d  i2c-1  i2c-3
0-0040  0-0070  2-001e  2-0033  2-005c  3-005c  i2c-0   i2c-2
```

/sys/bus/i2c/drivers lists drivers that are attached to the various devices

```
# ls /sys/bus/i2c/drivers
av8100      dev_driver  lp5521              lsm303dlh_m         tc3589x
bh1780      dummy       lps001wp_prs_sysfs  stmpe               tps6105x
bu21013_ts  l3g4200d    lsm303dlh_a         synaptics_rmi4_i2c
```

Accessing I2C registers in a program

```
#include <i2c-dev.h>
#include <sys/ioctl.h>

#define I2C_ADDRESS 0x5d
#define CHIP_REVISION_REG 0x10

main ()
{
    int f_i2c;
    int val;

    /* Open the adapter and set the address of the I2C device we want to talk to */
    f_i2c = open ("/dev/i2c-1", O_RDWR);
    ioctl (f_i2c, I2C_SLAVE, I2C_ADDRESS);

    /* Read 16-bits of data from a register */
    val = i2c_smbus_read_word_data (f, CHIP_REVISION_REG);
    printf (“Sensor chip revision %d\n”, val);

    close (f);
}
```

Summary

- I2C is a common on both embedded and non-embedded devices
- Master-slave architecture
- Master polls slaves when it wants to read or write data values