The Cubesat Internal bus: The I2C

Description: The purpose of this document is to describe the internal bus on the Cubesat. The internal bus has been chosen to be the I2C−bus “Interconnected Integrated Circuit”. The document will describe the purpose of the I2C and the design of a suitable Protocol for the data link layer.

Responsible group: OBC 732, **01gr732@control.auc.dk**, paen00,
Subsystem: On Board Computer & Camera Unit.
Date: 12.11.01
Rev.: 1.5
File name: I2C−bus_v_1.5pdf
Path: **ftp://??????.auc.dk/documentation**

Literature:
http://www−us.semiconductors.philips.com/i2c/facts/

Internal bus on Cubesat (I2C)

In order to obtain communication between the different subsystems on the AAU cubesat, the Philips I2C bus has been selected. I2C or Interconnected Integrated Circuit was developed by Philips in 1992, it has now become the world standard, and is currently implemented in over 1000 different ICs.

Here are some of the features of the I2C−bus:

Only two bus lines are required; a serial dataline (SAD) and a serial clock line (SCL)

· Each device connected to the bus is software addressable by a unique address and simple master/slave relationships exist at all times; masters can operate as master−transmitters or as master−receivers.

· It s a true multi−master bus including collision detection and arbitration to prevent data corruption if two or more masters simultaneously initiate data transfer

· Serial 8−bit oriented bi−directional data transfers can be made at up to 100 kbit/s in the Standard−mode, up to 400 kbit/s in the Fast−mode, or up to 3.4 Mbit/s in the High−speed mode

· On−chip filtering rejects spikes on the bus data line to preserve data integrity

· Extremely low current consumption.

· High noise immunity.

· Wide supply voltage range.

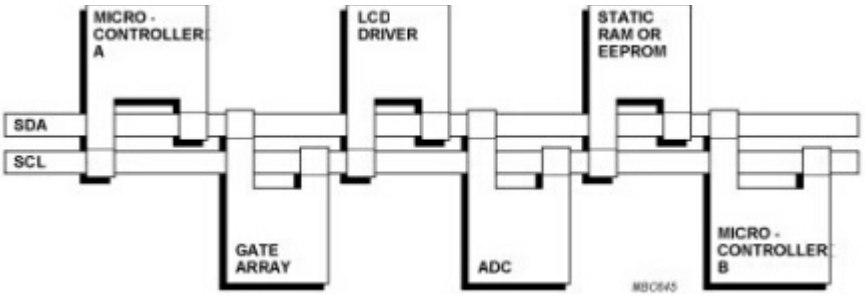
· Wide operating temperature range.

The I2C−bus supports any IC fabrication process (NMOS, CMOS, bipolar). Two wires, serial data (SDA) and serial clock (SCL), carry information between the devices connected to the bus. Each device is recognized by a unique address (whether it’s a microcontroller, LCD driver, memory or keyboard interface) and can operate as either a transmitter or receiver, depending on the function of the device. Obviously a LCD driver is only a receiver, whereas a memory can both receive and transmit data. In addition to transmitters and receivers, devices can also be considered as masters or slaves when performing data transfers (see Table 1). A master is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time any device addressed is considered a slave.

Table 1 Definition of I2C−bus terminology

Term	*DESCRIPTION*
Transmitter	The device which sends data to the bus
Receiver	The device which receives data from the bus
Master	The device which initiates a transfer, generates clock signal and terminates transfer.
Slave	The device addressed by a master
Multi−master	More than one master can attempt to control the bus at the same time without corrupting the message
Arbitration	Procedure to ensure that, if more than one master simultaneously tries to control the bus, only one is allowed to do so, and the winning message is not corrupted
Synchronization	Procedure to synchronize the clock signals of two or more devices

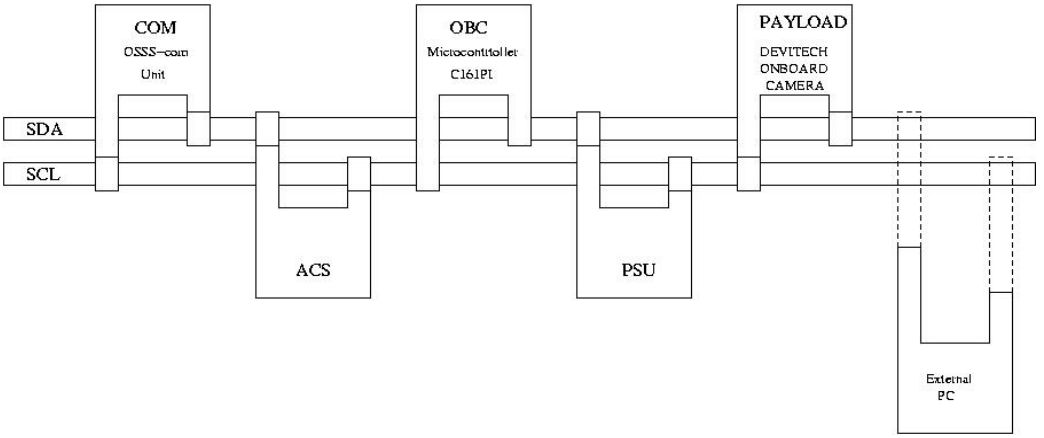
The I2C−bus is a multi−master bus. This means that more than one device capable of controlling the bus can be connected to it. As masters are usually micro−controllers, let s consider the case of a data transfer between two microcontrollers connected to the I2C−bus (see Fig.1).



If two or more masters try to put information onto the bus, the first to produce a "1" when the other produces a zero will lose the arbitration. The clock signals during arbitration are a synchronized combination of the clocks generated by the masters using the wired−AND connection to the SCL line.

Generation of clock signals on the I2C−bus is always the responsibility of master devices; each master generates its own clock signals when transferring data on the bus. Bus clock signals from a master can only be altered when they are stretched by a slow−slave device holding−down the clock line, or by another master when arbitration occurs.

CUBESAT I2C−BUS structure



The Cubesat structure consists internally of five different modules (COM, OBC, ACS, PSU and Payload). The last module on figure 1 is an external PC which is used for testing and simulation of the system.

I2C Protocol

The I2C protocol consists of the following.

Start:
The start condition is produced by a master, who wants to use the bus.
It is made by holding the SCL line high, while changing the SDA line form high to low.

Address:
Every device on the I2C−Bus has its own unique 7 bit address. The first 4 bit of the address is determined by the type of device connected to the I2C−Bus and is set by the manufacture of the IC. The last three bits are programmable, which allows 8 identical devices to be connected to the I2C−Bus.

R/W:

Read/Write is set by the Master, and determines whether the master wants to read from the slave or write to it. 1 for read and 0 for write.

Acknowledge:

Data transfer with acknowledge is obligatory. The acknowledge−related clock pulse is generated by the master. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse.
The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse, in order to confirm that it is willing to communicate.

Header:

Length:

The first three bits of the header determine how many data packages the header will be followed by (up to 8 * 8 bit data packages).

Header	No of data Packages
000xxxxx	1
001xxxxx	2
010xxxxx	3
011xxxxx	4
100xxxxx	5
101xxxxx	6
110xxxxx	7
111xxxxx	8

Module:

The device module number. E.g. The PSU temp probe nr. 2 (2*8 bit data) [00100010].
The total amount of modules you can address is 2^5 = 32
Data:

The data on the SDA line must be stable during the HIGH period of the clock. The HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW

Stop:

A LOW to HIGH transition on the SDA line while SCL is HIGH

defines a STOP condition.

Checksum:

The data package following the header is an 8 bit CRC Checksum.
The checksum is calculated in the following way.
All the data packages send in one transmission is added together, the 8 bit result of this calculation yields the CRC checksum. The received data is calculated in the same way and the last calculated checksum is subtracted from the “original” checksum. The result now indicates weather the transmitted data is corrupted. If the result is different from 0 the data is corrupted and will be retransmitted.

Send data		Received data		
[01101101]		[01101101]		
+		+		
[10110101]		[10110101]		
=		=		
1[00100010]	−	1[00100010]	=	0 The data is ok.

Communication Master−Slave Slave−Master:

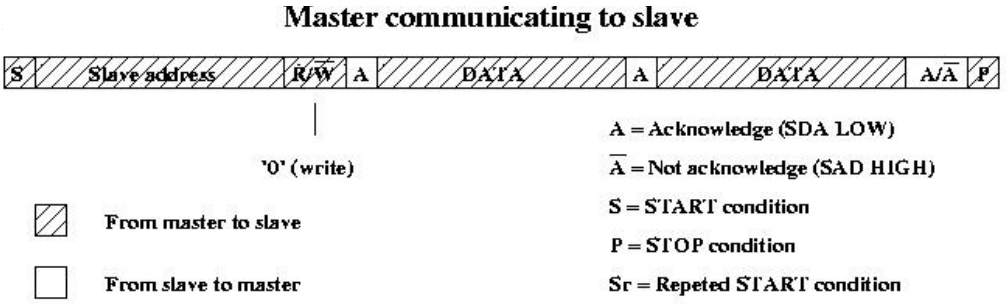
Possible data transfer formats are:
· Master−transmitter transmits to slave−receiver. The transfer direction is not changed (see Fig.2).

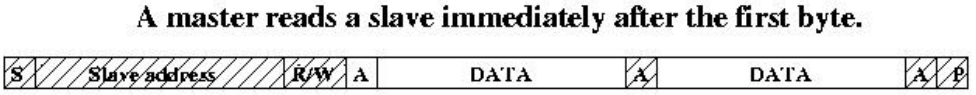
· Master reads slave immediately after first byte (see Fig.2). At the moment of the first acknowledge, the master− transmitter becomes a master− receiver and the slave−receiver becomes a slave−transmitter. This first acknowledge is still generated by the slave. The STOP condition is generated by the master, which has previously sent a not−acknowledge (A).

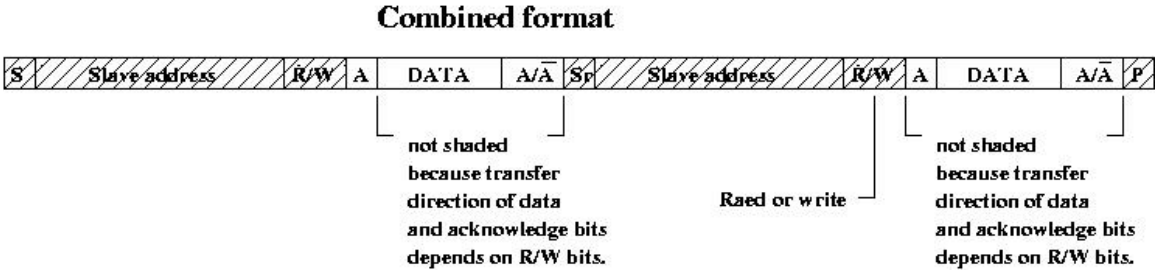
· Combined format (see Fig.2). During a change of direction within a transfer, the START condition and the slave address are both repeated, but with the R/W bit reversed. If a master receiver sends a repeated START condition, it has previously sent a not−acknowledge (A).

NOTES:
1. Combined formats can be used, for example, to control a serial memory. During the first data byte, the internal memory location has to be written. After the START condition and slave address is repeated, data can be transferred.
2. All decisions on auto−increment or decrement of previously accessed memory locations etc. are taken by the designer of the device.
3. Each byte is followed by an acknowledgment bit as indicated by the A or A blocks in the sequence.

4. I 2 C−bus compatible devices must reset their bus logic on receipt of a START or repeated START condition such that they all anticipate the sending of a slave address, even if these START conditions are not positioned according to the proper format.
5. A START condition immediately followed by a STOP condition (void message) is an illegal format.







Conducting housekeeping

When collecting housekeeping from the different units the master only needs to address a unit and set the R/W condition to “Read”, The unit then delivers all its housekeeping da ta to the MCU. Each sensor on a unit is identified by the module number transmitted in t he header.

Controlling Units

When controlling a unit from the MCU, the MCU will of course have to send data along with the address, header and checksum.

Further information about the I2C−bus can be found at.
http://www−us.semiconductors.philips.com/i2c/facts/