Sensoren und Aktoren

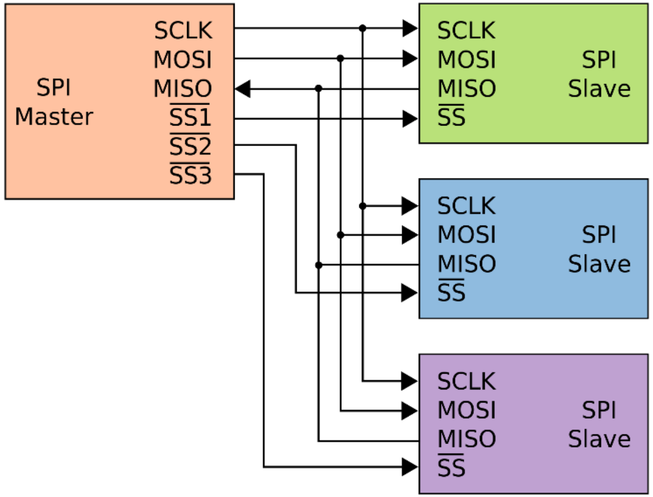
Wahlpflichtfach 5. Semester Elektrotechnik

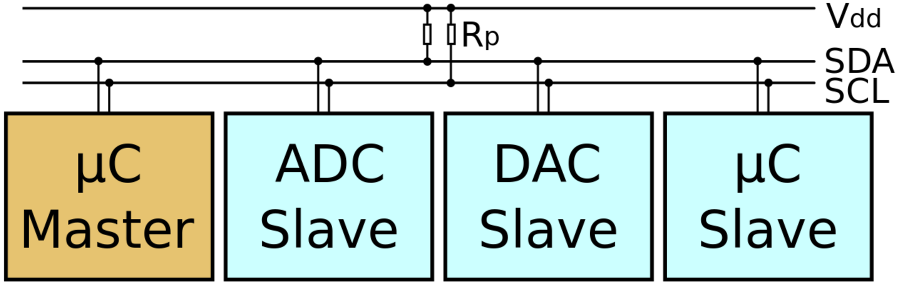
Prof. Dr. Felix Hüning

FB Elektrotechnik und Informationstechnik
FH Aachen

Examples

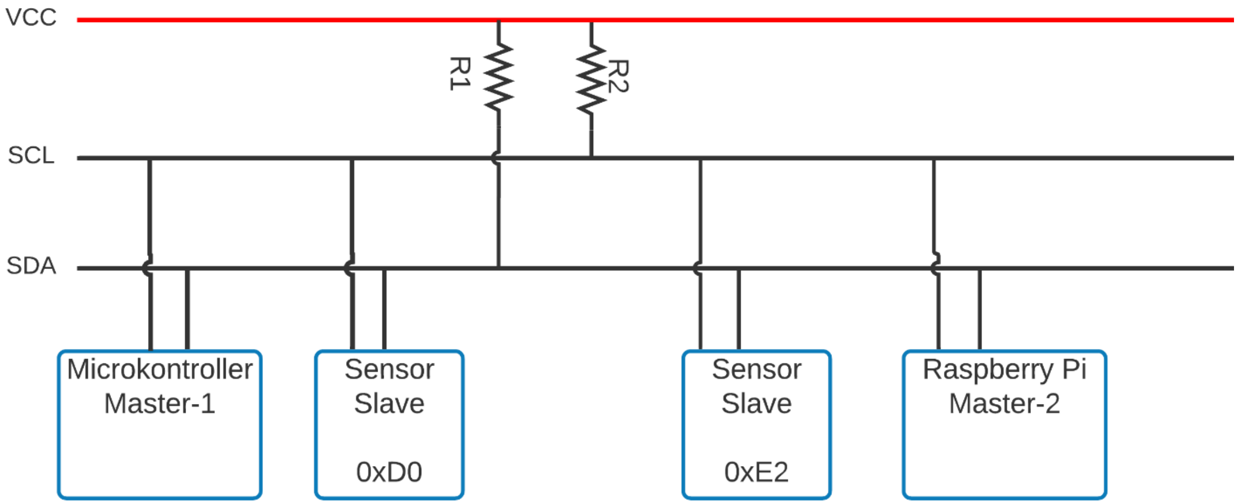
- Frequently used data busses for short range communication
  - SPI: Serial Peripheral Interface
  - I²C: Inter-Integrated Circuit





Von en:user:Cburnett - Own work made with Inkscape, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=1472017

I²C-Bus

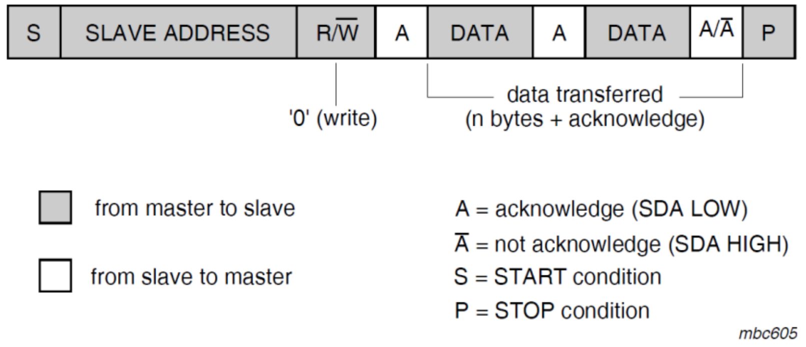


- 2 lines
    - SCL – clock (driven by master)
    - SDA – data line
- Open drain: Pull-up resistors to VCC

➡ Active low: units pull line to GND

- Standard: https://www.nxp.com/docs/en/user-guide/UM10204.pdf

Communication schedule: Master transmits



S	SLAVE ADDRESS	R/$\overline{W}$	A	DATA	A	DATA	A/$\overline{A}$	P

▨ from master to slave

□ from slave to master

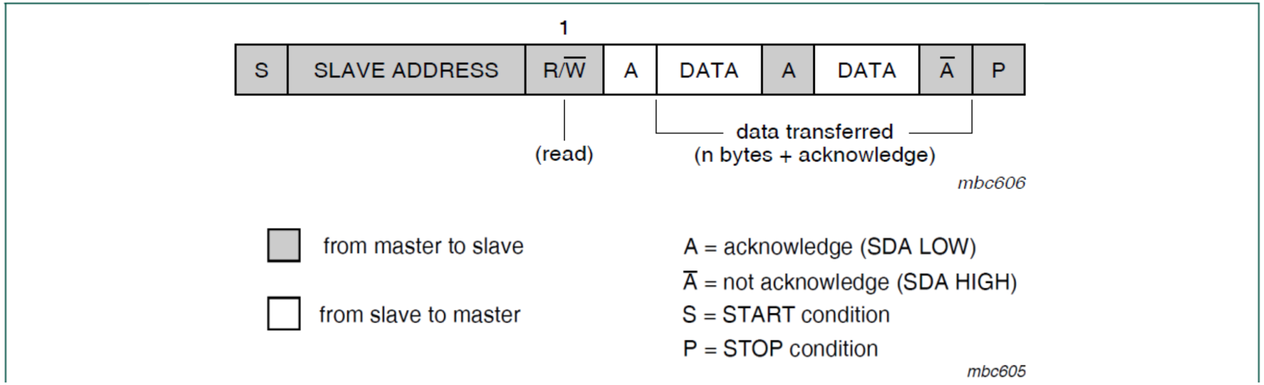
A = acknowledge (SDA LOW)
$\overline{A}$ = not acknowledge (SDA HIGH)
S = START condition
P = STOP condition

Fig 11. A master-transmitter addressing a slave receiver with a 7-bit address (the transfer direction is not changed)

Quelle: I²C Standard

- Master starts communication (Start-Condition)
- Master sends slave addresse (7Bit)
- Master indicates transmit or receive: (RW-Bit)
- Slave sends acknowledgment (Acknowledge, ACK)
- Master send data in 8 Bit units, Slave acknowledges by ACK
- End of transfer by master (Stop-Condition)

Communication schedule: Slave transmits



		1						
S	SLAVE ADDRESS	R/$\overline{W}$	A	DATA	A	DATA	$\overline{A}$	P

(read)

data transferred (n bytes + acknowledge)

▨ from master to slave

□ from slave to master

A = acknowledge (SDA LOW)

$\overline{A}$ = not acknowledge (SDA HIGH)

S = START condition

P = STOP condition

Quelle: I²C Standard

- Master starts communication (Start-Condition)
- Master sends slave addresse (7Bit)
- Master indicates transmit or receive: (RW-Bit)
- Slave sends acknowledgment (Acknowledge, ACK)
- Slave sends data in 8 Bit units, Master acknowledges
- After last data byte: Master sends NACK
- Master stops communication (Stop-Condition)

Start and stop condition

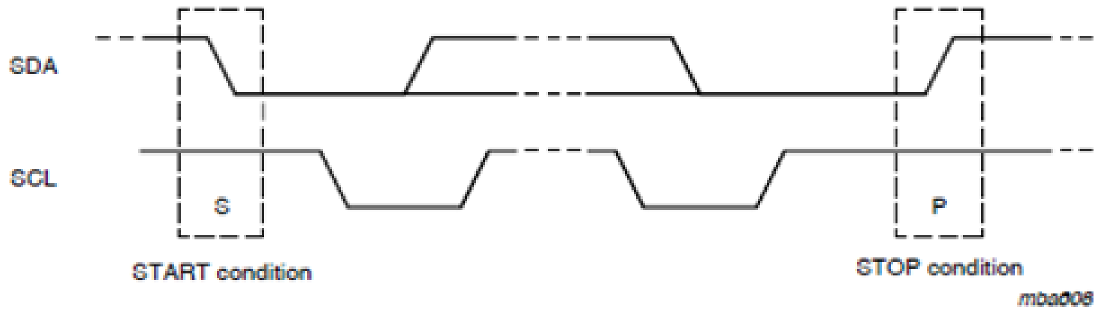


Fig 5. START and STOP conditions

Quelle: I²C Standard

- Bus not busy: SDA and SCL are HIGH
- **Start condition:**
  - Master pulls SDA to LOW
  - SCL stays HIGH
- **Stop condition:**
  - SDA is pulled to HIGH
  - At the same time SCL is HIGH

Validity of bits

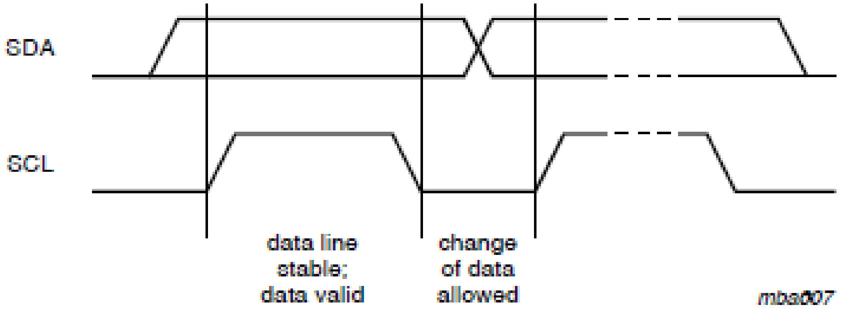
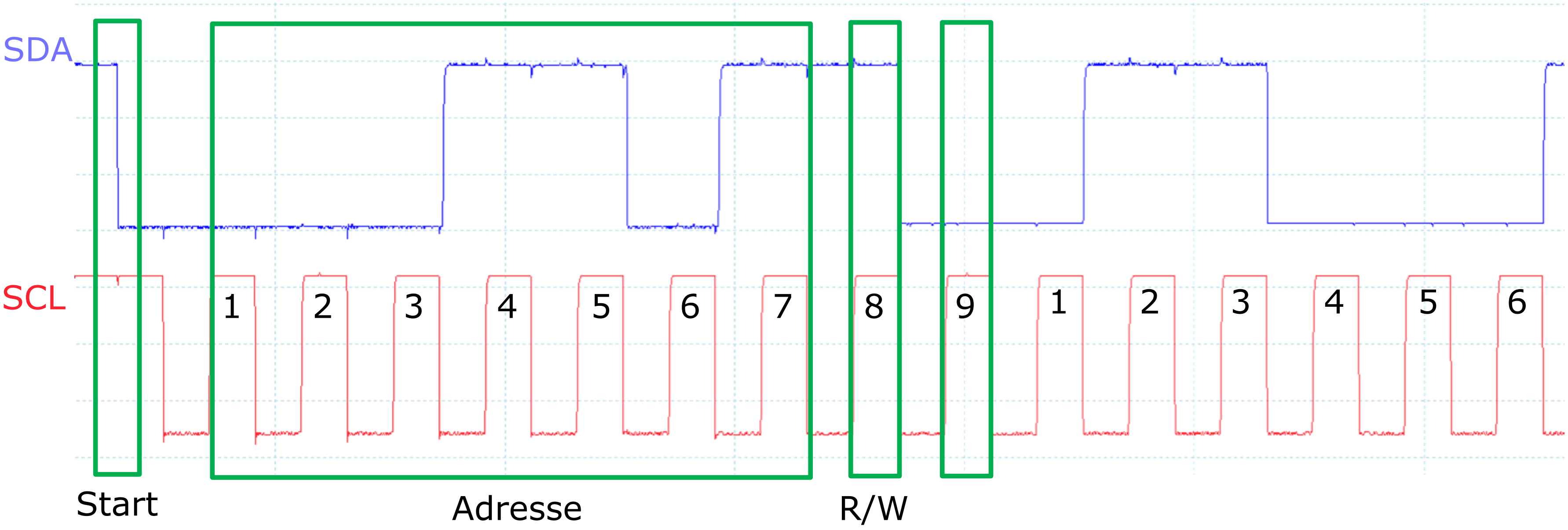


Fig 4. Bit transfer on the I²C-bus

Quelle: I²C Standard

- During data transmission:
  Changes of SDA during Clock LOW only
- Start und Stop condition
  - Clock HIGH and SDA level changes

Addressing of slave devices

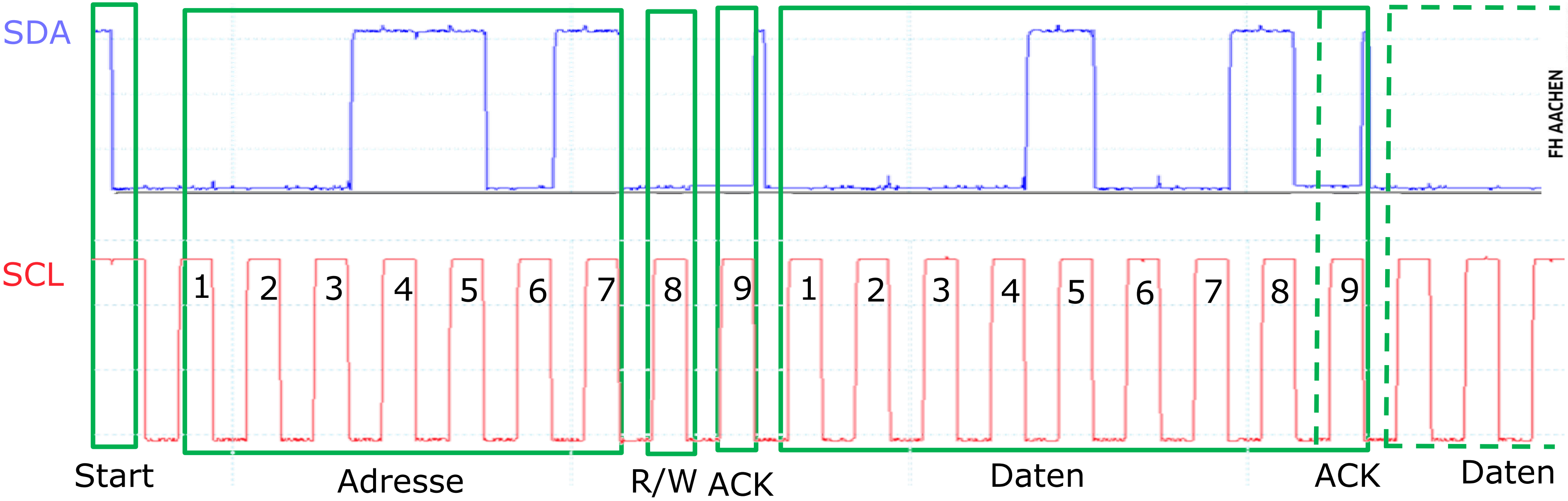


- Start condition
- Address field (7-Bit)

1	2	3	4	5	6	7	Wert
0	0	0	1	1	0	1	0x0D

- R/W Bit (0-Write, 1-Read)
- ACK by slave: slave pulls SDA to LOW

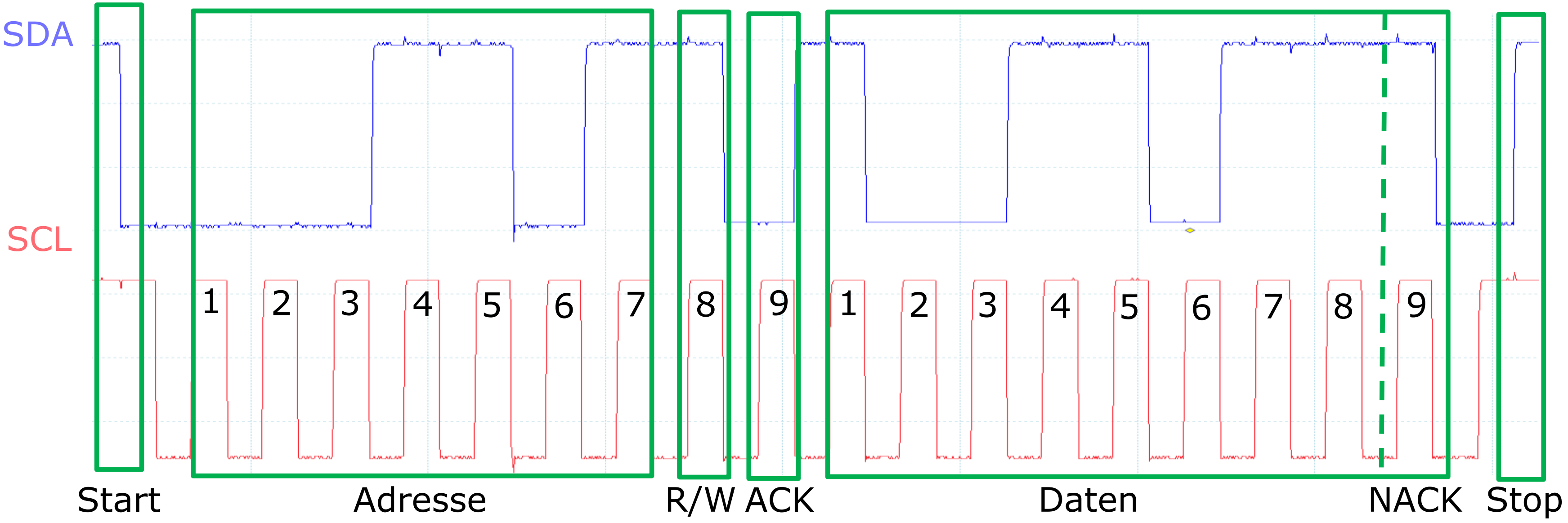
Data transfer: data to slave



- Start condition
- Address field (7-Bit) + R/W Bit (0-Write, 1-Read) + ACK
- 1 or more 8-Bit data fields + ACK/NACK

  Data sent: **b00001001** oder **0x09**
- Stop condition

Data transfer: data from slave



- Start condition
- Address field (7-Bit) + R/W Bit (0-Write, 1-Read) +ACK
- 1 or more 8-Bit data fields + ACK/NACK

  Data from sensor: **b1001 1011 = 0x9B**
- Stop condition

Communication

- Start
- Addresse + R/W-Bit from master
- Receiver ties SDA to LOW for one clock cycle (ACK) (9. cylce)
- ACK mandatory for all but the last byte

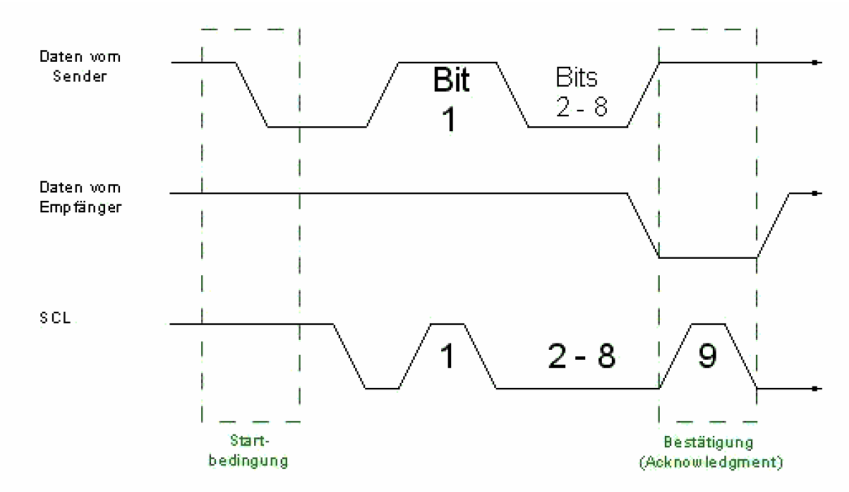
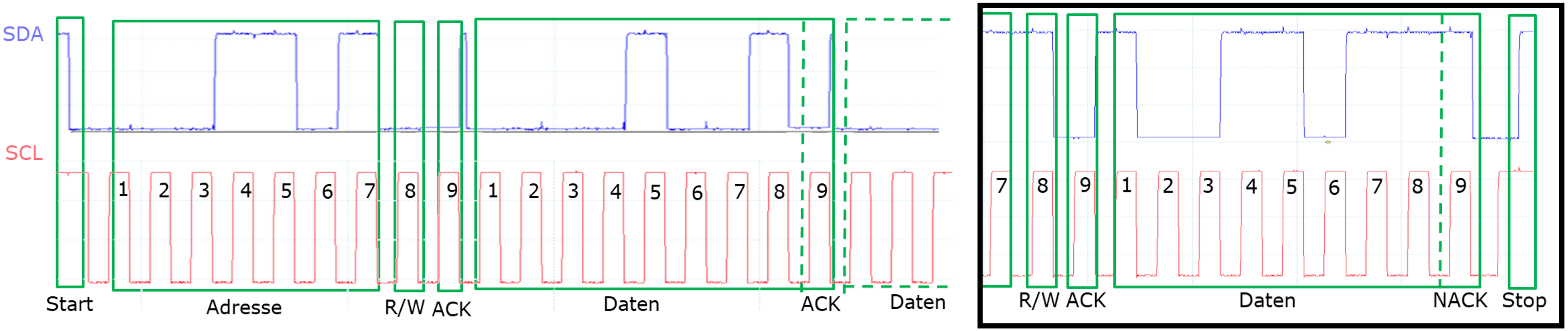


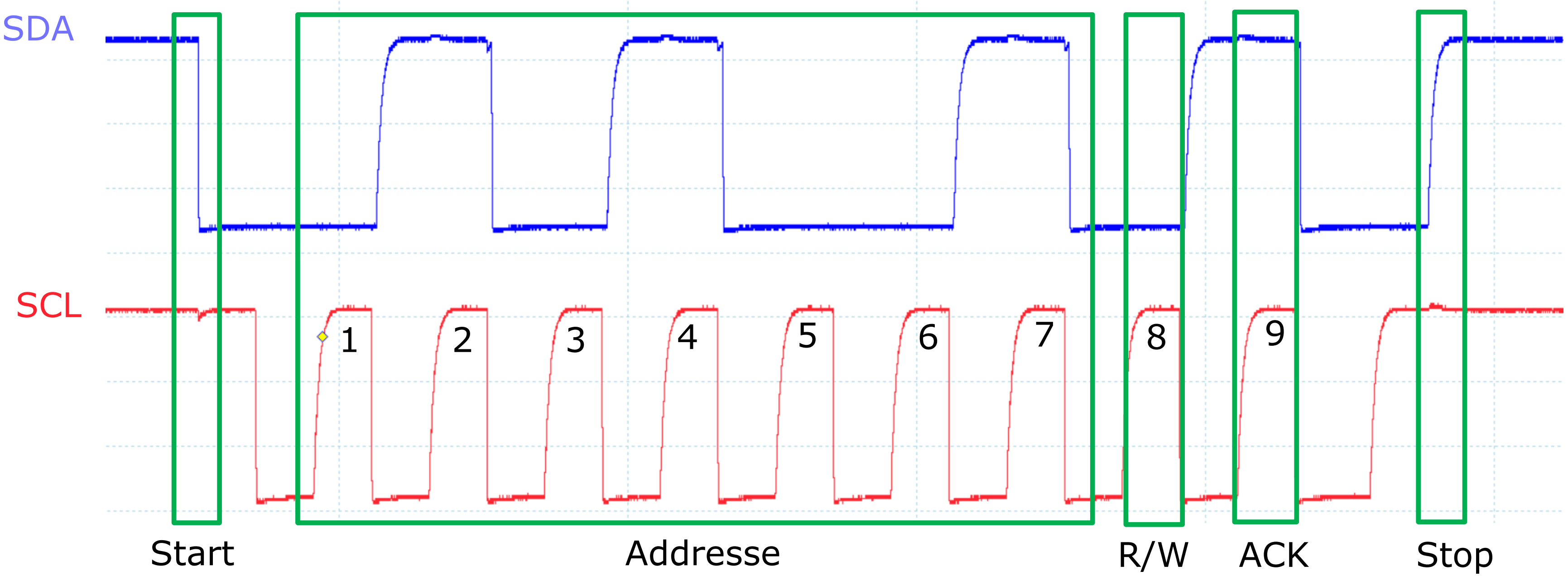
Bild [10]: Bestätigung bei der Datenübertragung
Quelle: Roboternetz.de

ACK, NACK and No ACK



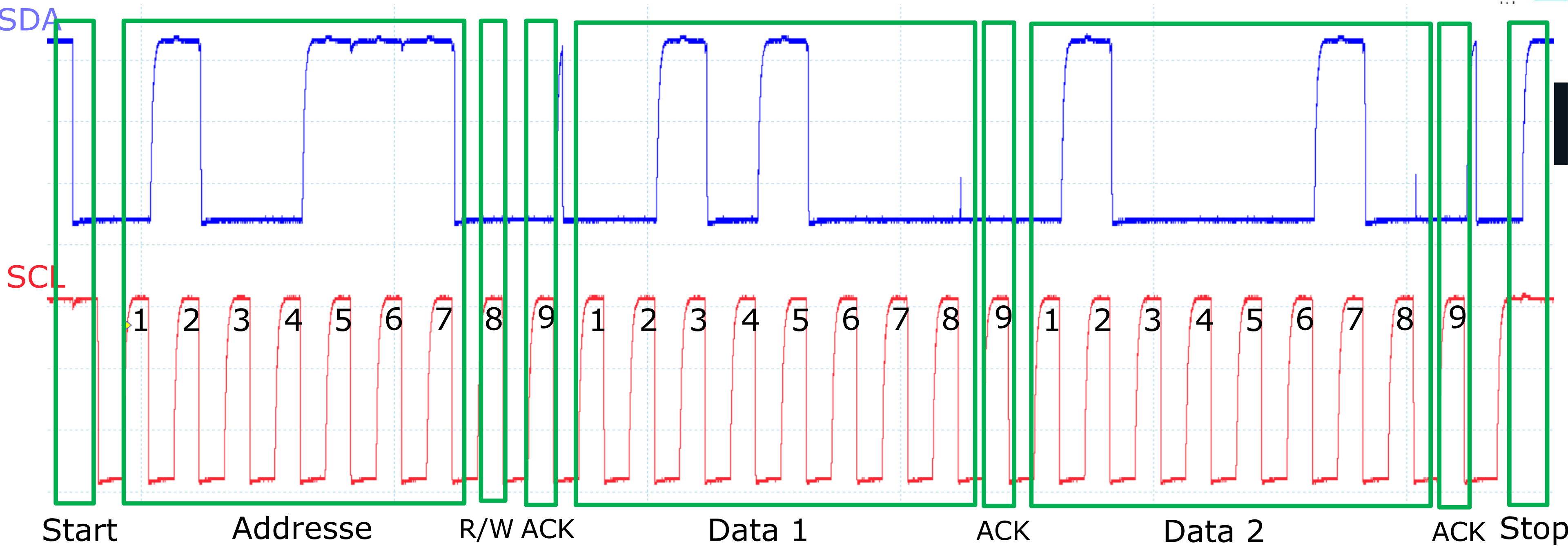
- ACK (SDA is LOW) by receiver
- At the end of a data transfer to the master the master sends NACK (SDA = HIGH), indicating all data were received
- Reasons for missing ACK (no ACK)
  - No device with given address available
  - Missing synchronization (e.g. slave missed one clock cycle)

Decoding an I2C signal



- Address: b0101001 = 0x29
- R/W: Low, Write: Master sends to Slave
- No ACK from Slave
- Stop condition
- No response from slave, wrong addresse or slave not connected

Decoding an I2C signal



- Address: b0100111 = 0x27
- R/W: Low, Write: Master sends to slave + ACK from slave
- Data1: b0010 1000 = 0x28 + ACK from Slave
- Data2: b0100 0010 = 0x42 + ACK from Slave
- Stop condition
- Valid data transfer

Additional features of the standard

Additional features of I2C

- Different data rates
  (Fast Mode, High Speed, …)
- 10-Bit addressing and sub-adressierung
- Multi-Master-Mode
- Repeated Start-Signal
- Clock Streching

Quelle: Wikipedia

Data rates

	Datenrate	
Standard Mode	Up to 100kbit/s	Bidirectional
Fast Mode	Up to 400kbit/s	Bidirectional
Fast mode+	Up to 1 Mbit/s	Bidirectional
High Speed Mode	Up to 3.4 Mbit/s	Bidirectional
Ultra Fast Mode	Up to 5.0 Mbit/s	Unidirectional

- Most sensors supprt Standard Mode
- Fast Mode, Fast Mode+ and High-speed devices are downward compatibel
- Ultra-Fast Mode is not downward compatibel

Communication: addressing

- I^{2}C addresses are often device specific
- Same devices on on bus system: address is split into fixed address and sub-address

	Fixed address				Sub-address		
Adresse	0	0	1	0	x	x	x

- Fixed address of device (4bit) and sub-addresse (3bit)
- Adressing via solder option or jumper

Communication: dedicated addresses

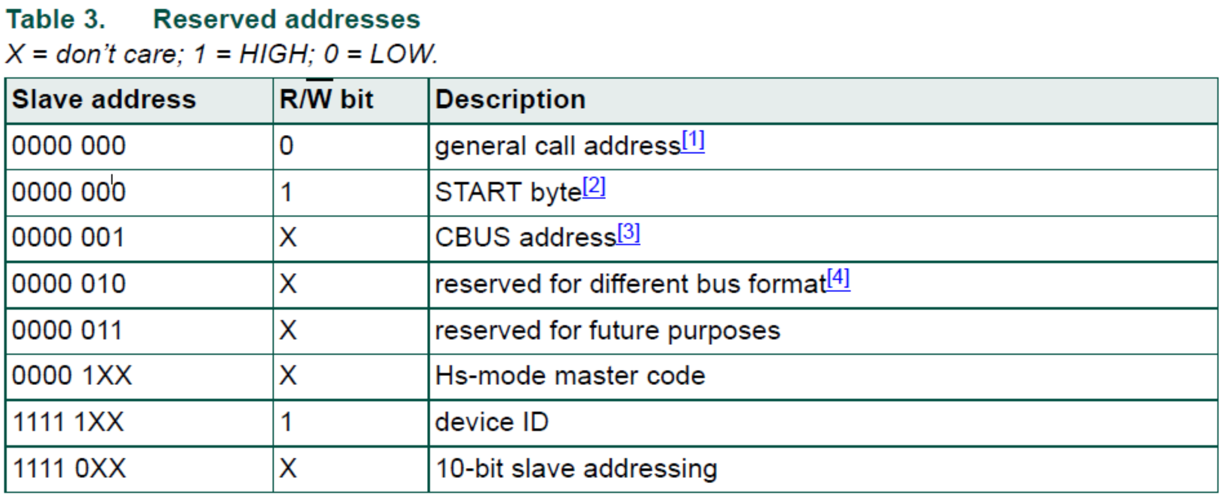


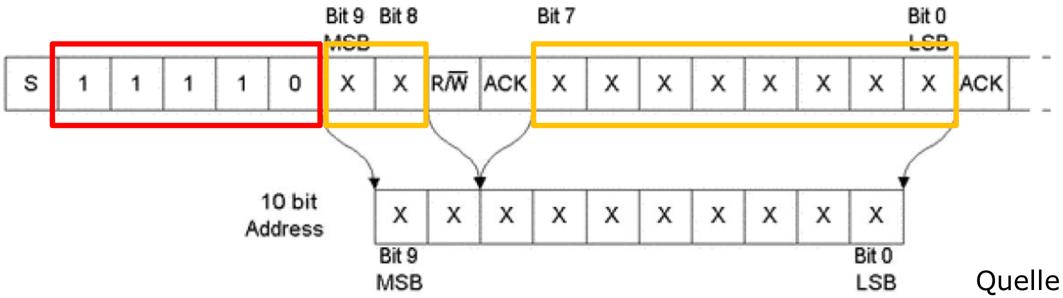
Table 3. Reserved addresses

X = don't care; 1 = HIGH; 0 = LOW.

Slave address	R/$\overline{W}$ bit	Description
0000 000	0	general call address[1]
0000 000	1	START byte[2]
0000 001	X	CBUS address[3]
0000 010	X	reserved for different bus format[4]
0000 011	X	reserved for future purposes
0000 1XX	X	Hs-mode master code
1111 1XX	1	device ID
1111 0XX	X	10-bit slave addressing

- General call address: Used for reset, address transmission
- Start Byte: High frequency polling for devices polling the bus
- Device ID: Read supplier and device ID
- 10-bit adressing

Communication: 10bit addressing



Quelle: I2C-bus.org

- Indicator for 10 bit addressing
- First 2 bits of 10 bit address
- R/W bit and Ack
- First data field with remaining 8 bit of 10 bit address

Multi-Master

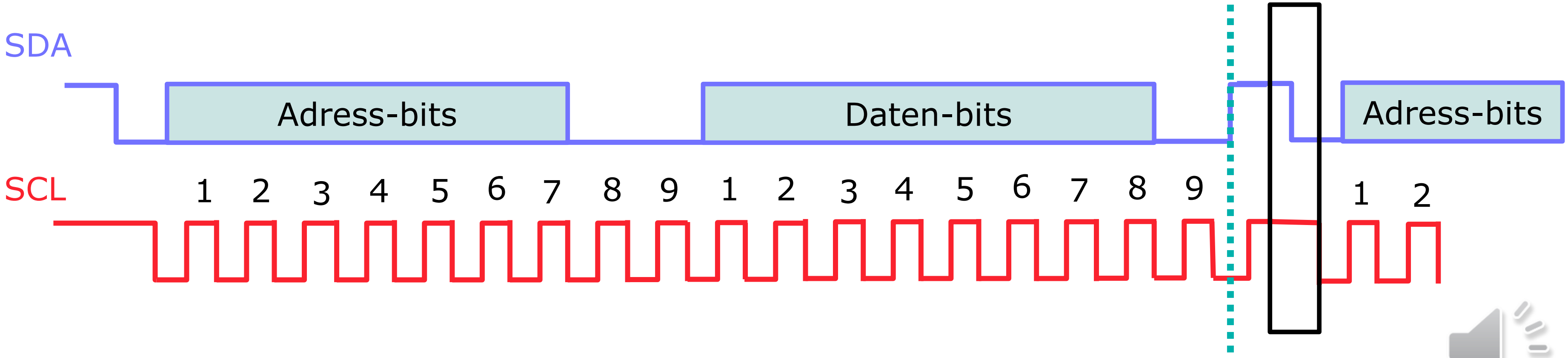
- Has to be supported by all nodes
- Mixed singel and multi master devices may lead to disturbed communication

- Mutli-Master devices sense SCL and SDA
  - Check whether bus is free (> 4.7 ns High)
  - Wait for stop condition before start of transmission
  - Interruption of transmission in case of an unexpected low level

Repeated Start (Sr)

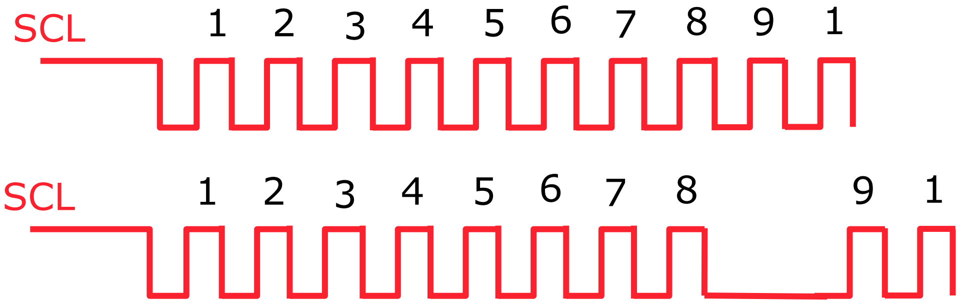
- Start without previous stop
- Read and write alternating (without releasing the bus)
- Example: Read memory
  1. Master writes address of memory cell (I²C Write)
  2. Master reads memory from slave (I²C Read)
- Stop condition between 1. and 2. would release the bus
- Another master can start access

➡ Race condition



Clock Stretching

- Option to stop or delay I2C communication
- Slave ties clock to low
- Slave has more time to send Ack



Troubleshooting

- No communication

- Possible reasons:
  - Bus seems busy (SDA and SCL not HIGH)
  - Unexpected restart of MCU (Master) during communication
  - Electromagnetic disturbance
- Slave waits for missing clock signals
- Slave doesn‵t react on start condition

- Remedy: Send several clock signals (e.g. 16) and stop signal after reset to put slave back to receive mode

Troubleshooting:

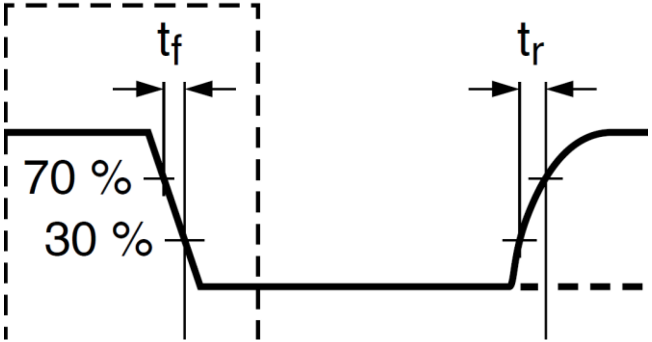
- Occasionally no ACK from Slave

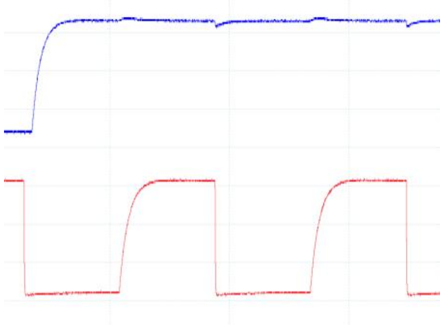
- Possible reasons:
  - Signal levels are not reached in time (to slow)
  - ➡ Slave misses one or more clock cycles
  - Electromagnetic disturbance was read as clock cycle

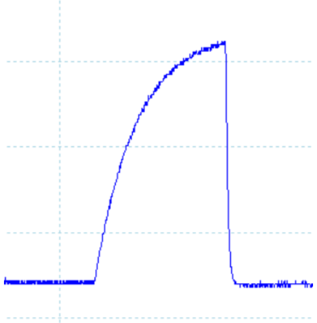
- Check signal level and rise and fall time wrt specification

- Remedy: Modification of pull-up resistors

Fall and Rise Time:







- Signal edges have to be in line with specification
- System setup important (devices, cable length, pull up resistors)
    - Longer cable, more devices, pull up resistors with high resistance increase rise time
- If t_r is to big final signal level is not reached in time

Electrical characteristics

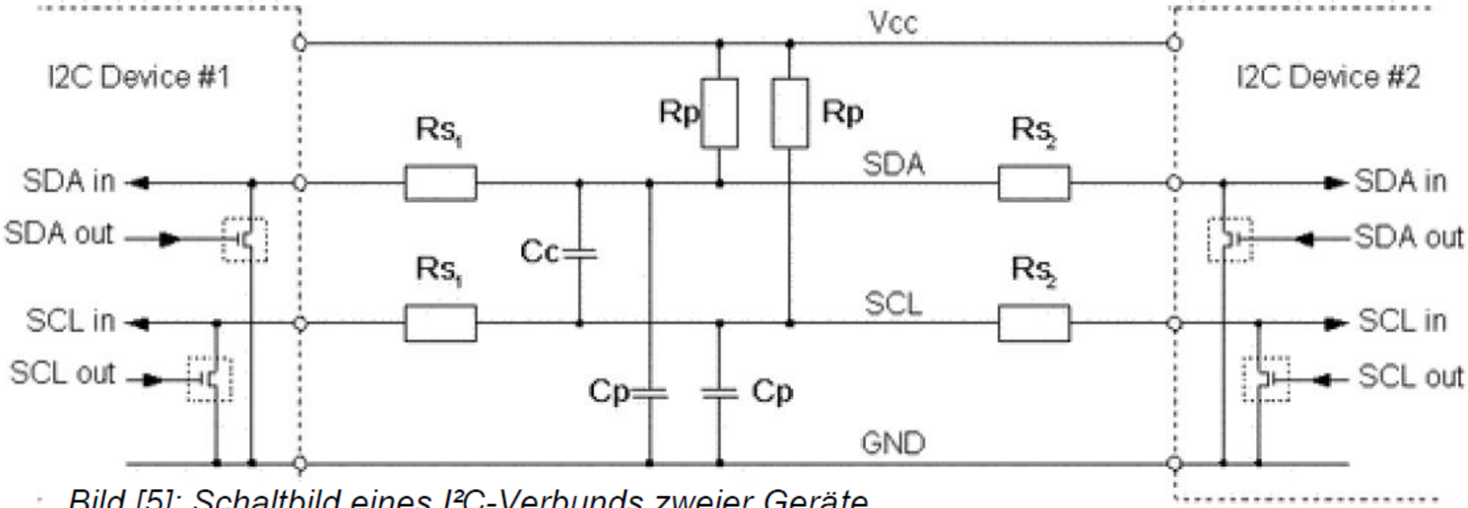
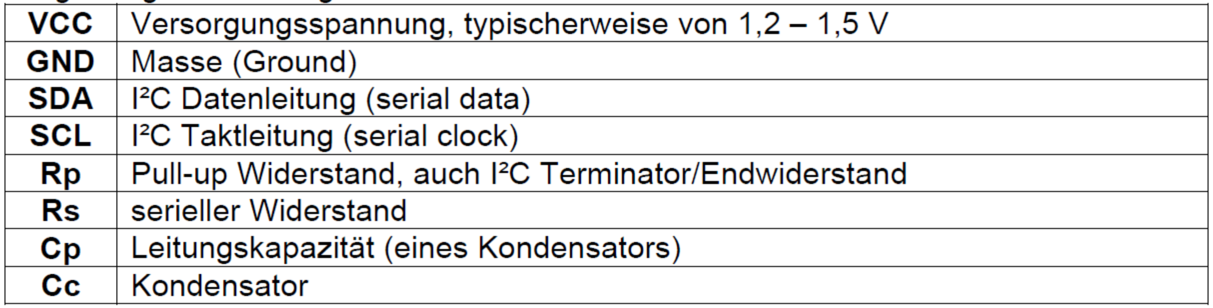


Bild [5]: Schaltbild eines I²C-Verbunds zweier Geräte
Quelle: I²C-bus.org

Zugehörige Abkürzungen:



VCC	Versorgungsspannung, typischerweise von 1,2 – 1,5 V
GND	Masse (Ground)
SDA	I²C Datenleitung (serial data)
SCL	I²C Taktleitung (serial clock)
Rp	Pull-up Widerstand, auch I²C Terminator/Endwiderstand
Rs	serieller Widerstand
Cp	Leitungskapazität (eines Kondensators)
Cc	Kondensator

Specification pull up calculation

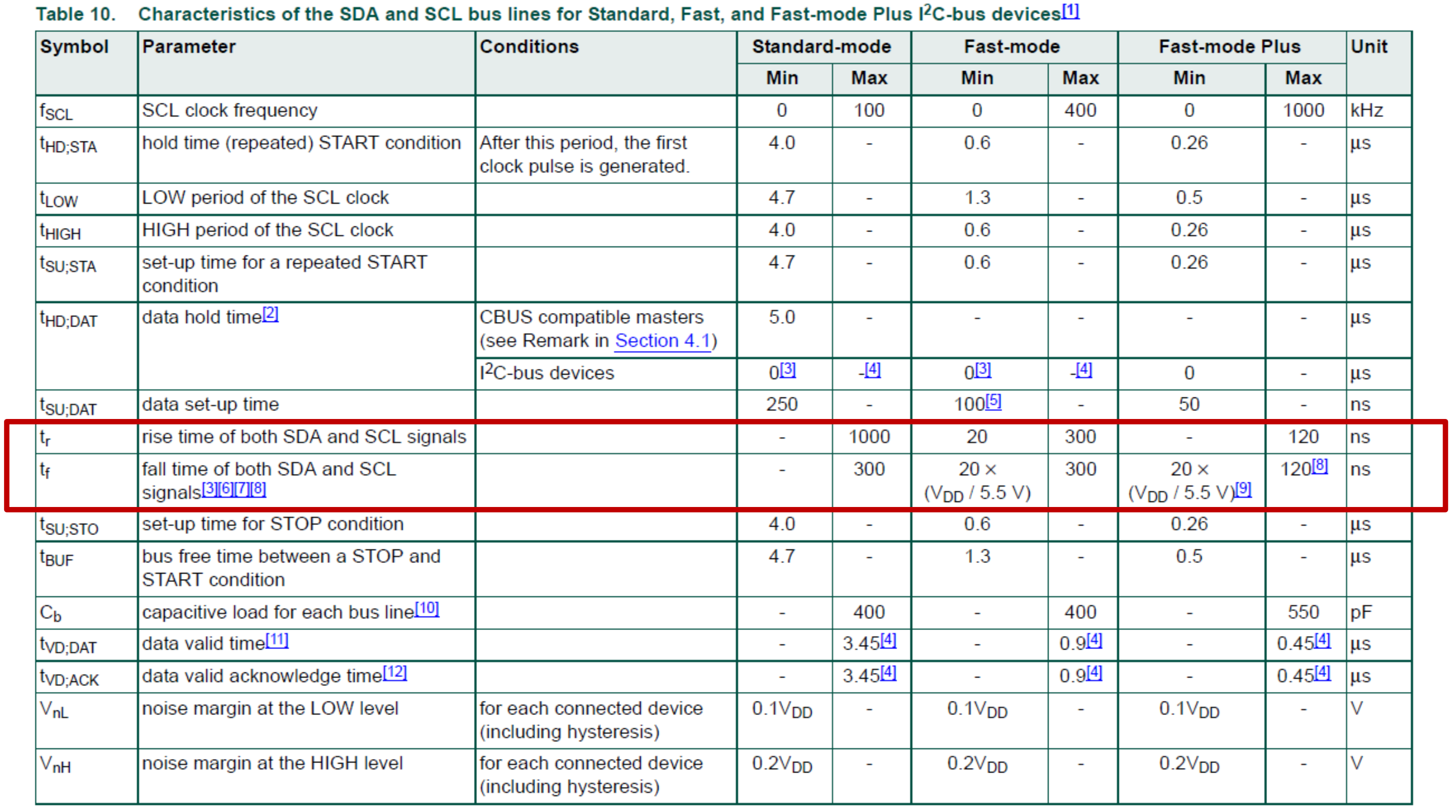


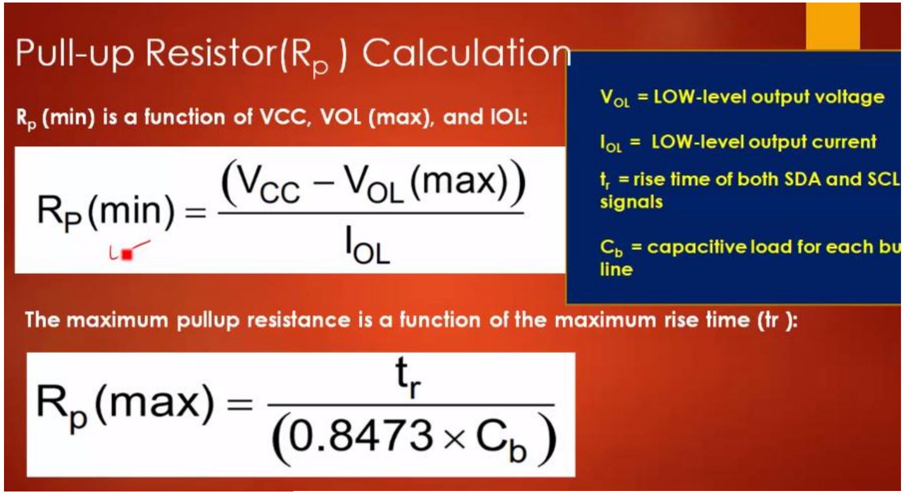
Table 10. Characteristics of the SDA and SCL bus lines for Standard, Fast, and Fast-mode Plus I²C-bus devices[1]

Symbol	Parameter	Conditions	Standard-mode		Fast-mode		Fast-mode Plus		Unit
			Min	Max	Min	Max	Min	Max	
f_{SCL}	SCL clock frequency		0	100	0	400	0	1000	kHz
$t_{HD;STA}$	hold time (repeated) START condition	After this period, the first clock pulse is generated.	4.0	-	0.6	-	0.26	-	µs
t_{LOW}	LOW period of the SCL clock		4.7	-	1.3	-	0.5	-	µs
t_{HIGH}	HIGH period of the SCL clock		4.0	-	0.6	-	0.26	-	µs
$t_{SU;STA}$	set-up time for a repeated START condition		4.7	-	0.6	-	0.26	-	µs
$t_{HD;DAT}$	data hold time[2]	CBUS compatible masters (see Remark in Section 4.1)	5.0	-	-	-	-	-	µs
		I²C-bus devices	0[3]	-[4]	0[3]	-[4]	0	-	µs
$t_{SU;DAT}$	data set-up time		250	-	100[5]	-	50	-	ns
t_r	rise time of both SDA and SCL signals		-	1000	20	300	-	120	ns
t_f	fall time of both SDA and SCL signals[3][6][7][8]		-	300	20 × (V_{DD} / 5.5 V)	300	20 × (V_{DD} / 5.5 V)[9]	120[8]	ns
$t_{SU;STO}$	set-up time for STOP condition		4.0	-	0.6	-	0.26	-	µs
t_{BUF}	bus free time between a STOP and START condition		4.7	-	1.3	-	0.5	-	µs
C_b	capacitive load for each bus line[10]		-	400	-	400	-	550	pF
$t_{VD;DAT}$	data valid time[11]		-	3.45[4]	-	0.9[4]	-	0.45[4]	µs
$t_{VD;ACK}$	data valid acknowledge time[12]		-	3.45[4]	-	0.9[4]	-	0.45[4]	µs
V_{nL}	noise margin at the LOW level	for each connected device (including hysteresis)	$0.1V_{DD}$	-	$0.1V_{DD}$	-	$0.1V_{DD}$	-	V
V_{nH}	noise margin at the HIGH level	for each connected device (including hysteresis)	$0.2V_{DD}$	-	$0.2V_{DD}$	-	$0.2V_{DD}$	-	V

- Specification I2C: max_t_r = 1000 ns, max t_f=300 ns

Specification pull up calculation

- Spezifikation I2C: max_trise = 1000ns, V_OL =0.4V
- Cb: Measured or from data sheet



Pull-up Resistor(R_p) Calculation

R_p (min) is a function of VCC, VOL (max), and IOL:

$$R_P(min) = \frac{(V_{CC} - V_{OL}(max))}{I_{OL}}$$

V_{OL} = LOW-level output voltage

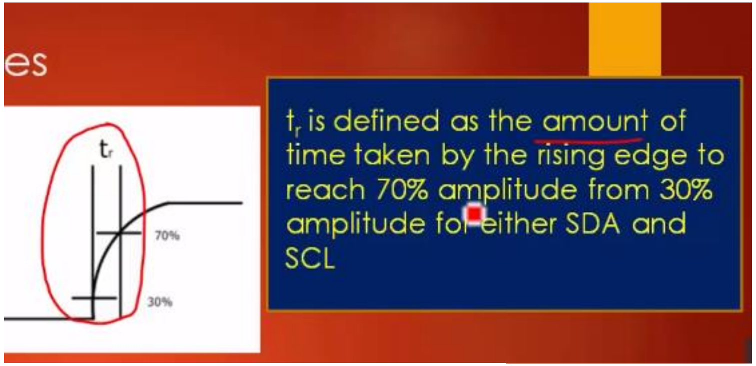
I_{OL} = LOW-level output current

t_r = rise time of both SDA and SCL signals

C_b = capacitive load for each bu line

The maximum pullup resistance is a function of the maximum rise time (tr):

$$R_p(max) = \frac{t_r}{(0.8473 \times C_b)}$$



es

t_r is defined as the amount of time taken by the rising edge to reach 70% amplitude from 30% amplitude for either SDA and SCL

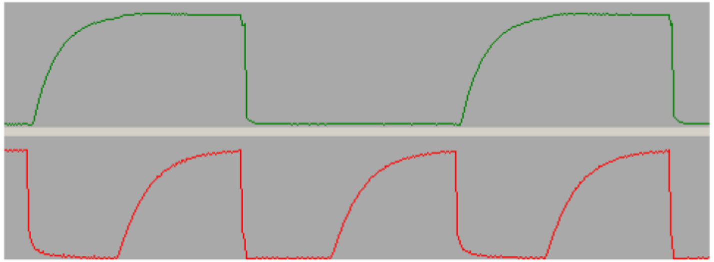


Bild [6]: SDA (oben) und SCL (unten), Rp = 10 kΩ, Cp = 300 pF, SCL-Takt läuft mit 100 kHz (normal)
Quelle: I²C-Bus.org

FH Aachen
Fachbereich Elektrotechnik und Informationstechnik
Prof. Dr. rer. nat. Felix Hüning
Eupener Straße 70
52066 Aachen
T +49 (0)241 6009 51979
F +49 (0)241 6009 52191
huening@fh-aachen.de
www.huening.fh-aachen.de