

Arduino en I2C – Dag 3

Huiswerk

1. Vragen over logging?
2. DS3231 datasheet?

I2C bus – De details

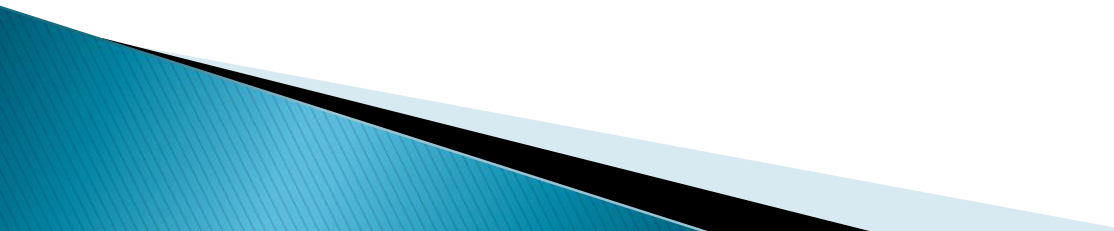
Waar je een keer tegenaan zult lopen...

- ▶ Clock stretching
- ▶ Meerdere voedingsspanningen
- ▶ Pull-up & bus-capaciteit

Clock stretching



Clock stretching

- ▶ Optioneel voor slave
 - ▶ Master moet dit ondersteunen
 - ▶ Pause tot de clock weer hoog gaat
 - ▶ Geeft extra verwerkingstijd voor slave.
 - ▶ Kost bandbreedte
- 

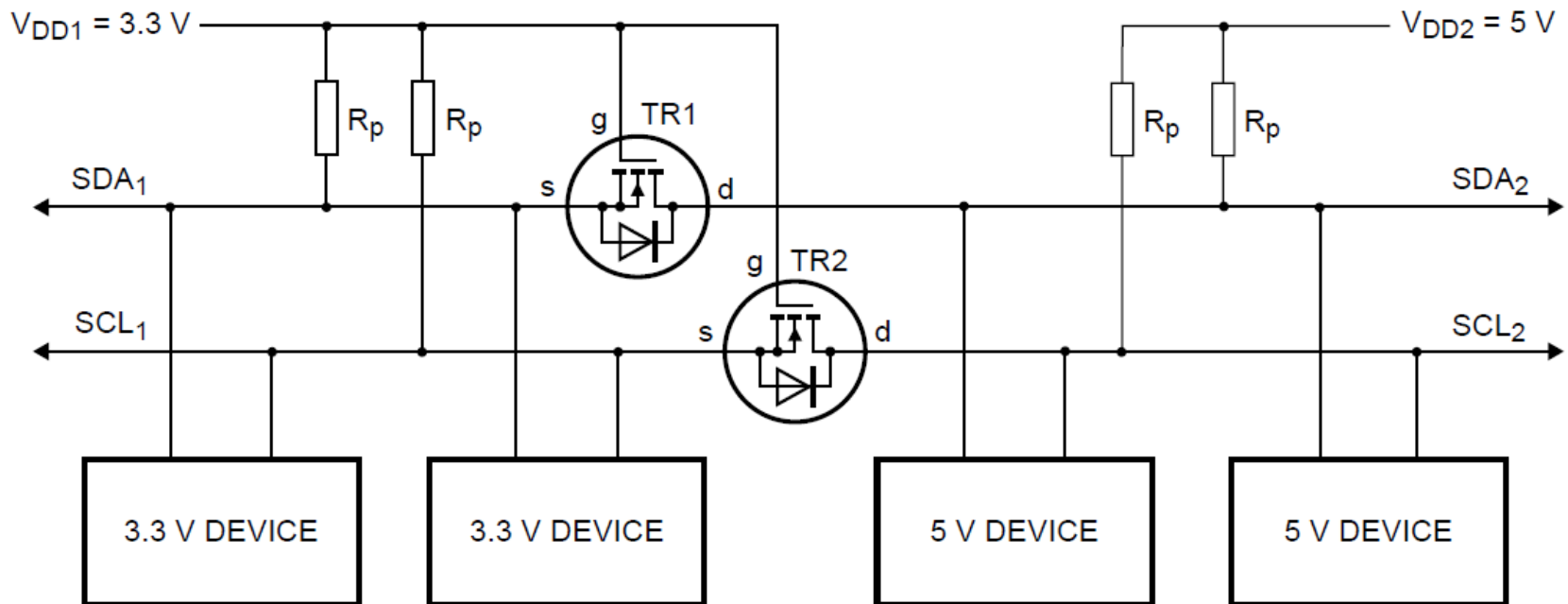
Verschillende voedingsspanningen

- ▶ Devices met verschillende voedingsspanning.

Bijvoorbeeld:

- ▶ 3V3 master (ARM) → PCF8574 → 5V LCD
- ▶ 5V master (Uno) → 3V3 sensor
 - BH1750 Light Sensor
 - VL53L0X Distance Sensor
 - BMP280 Digital Pressure Sensor
 - (en dit worden er steeds meer)

Verschillende voedingsspanningen



mgk879

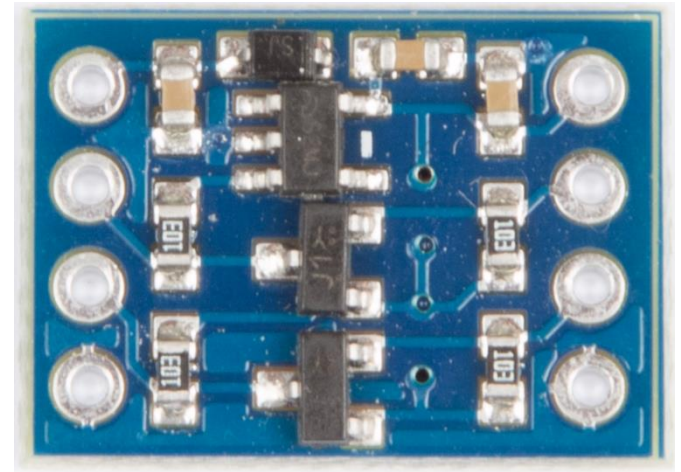
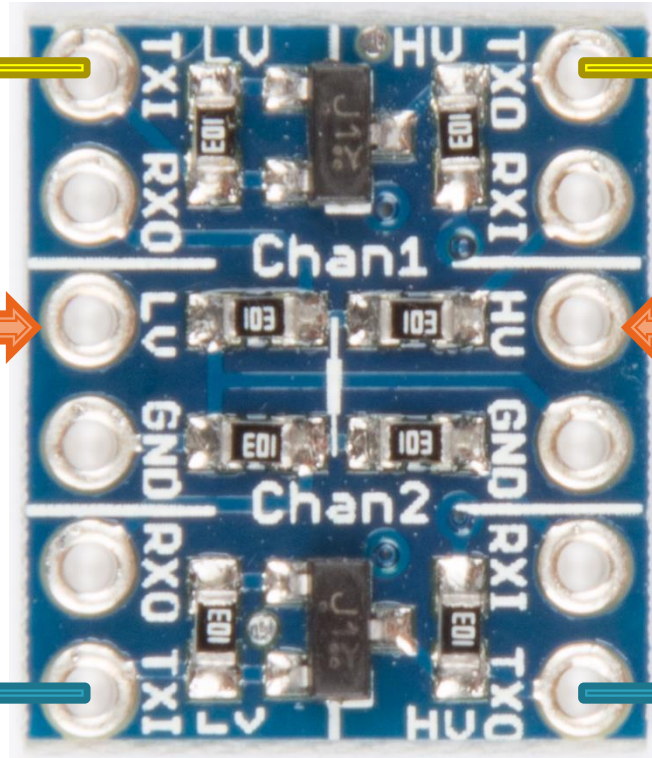
Fig 1. Bidirectional level shifter circuit connecting two different voltage sections in an I²C-bus system

Level shifter

SDA

3V3

SCL



3V3

5V



Pull-up & bus-capaciteit

- ▶ Flank omhoog steil genoeg.
- ▶ 'Laag' niveau $< 0.4V$ ($\Rightarrow 3 \text{ mA max}$)
- ▶ 1.5k–3k (5V, 3mA, 100kHz, 400pF)

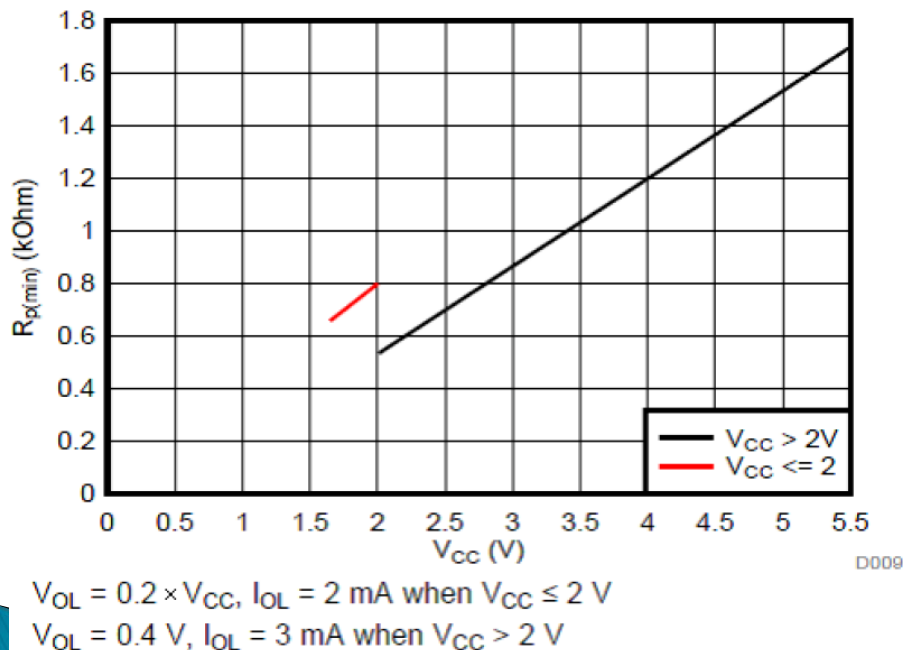


Figure 2. Minimum Pullup Resistance [R_p (min)] vs Pullup Reference Voltage (V_{cc})

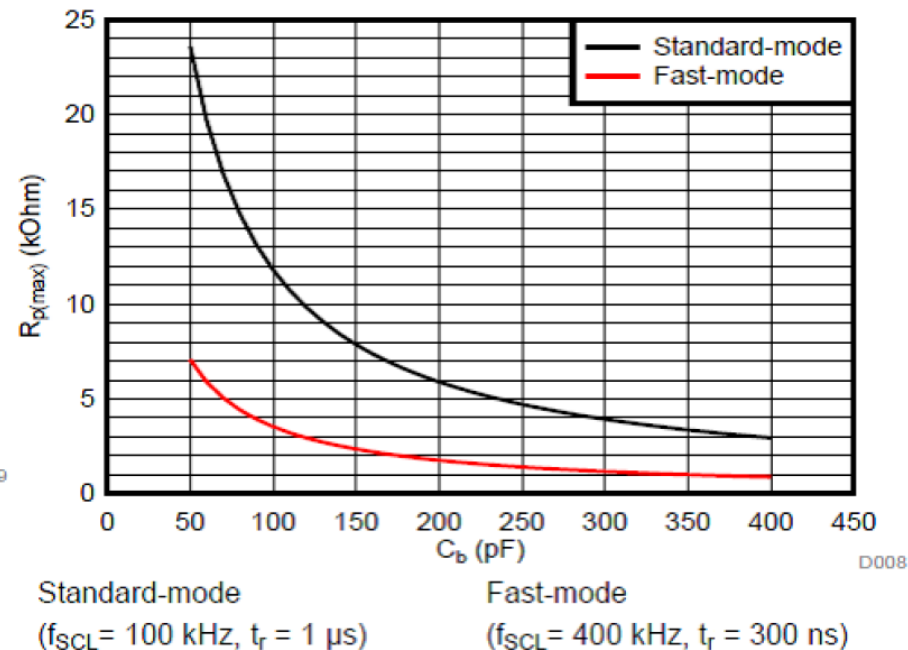


Figure 3. Maximum Pullup Resistance [R_p (max)] vs Bus Capacitance (C_b)

I2C Pull-up

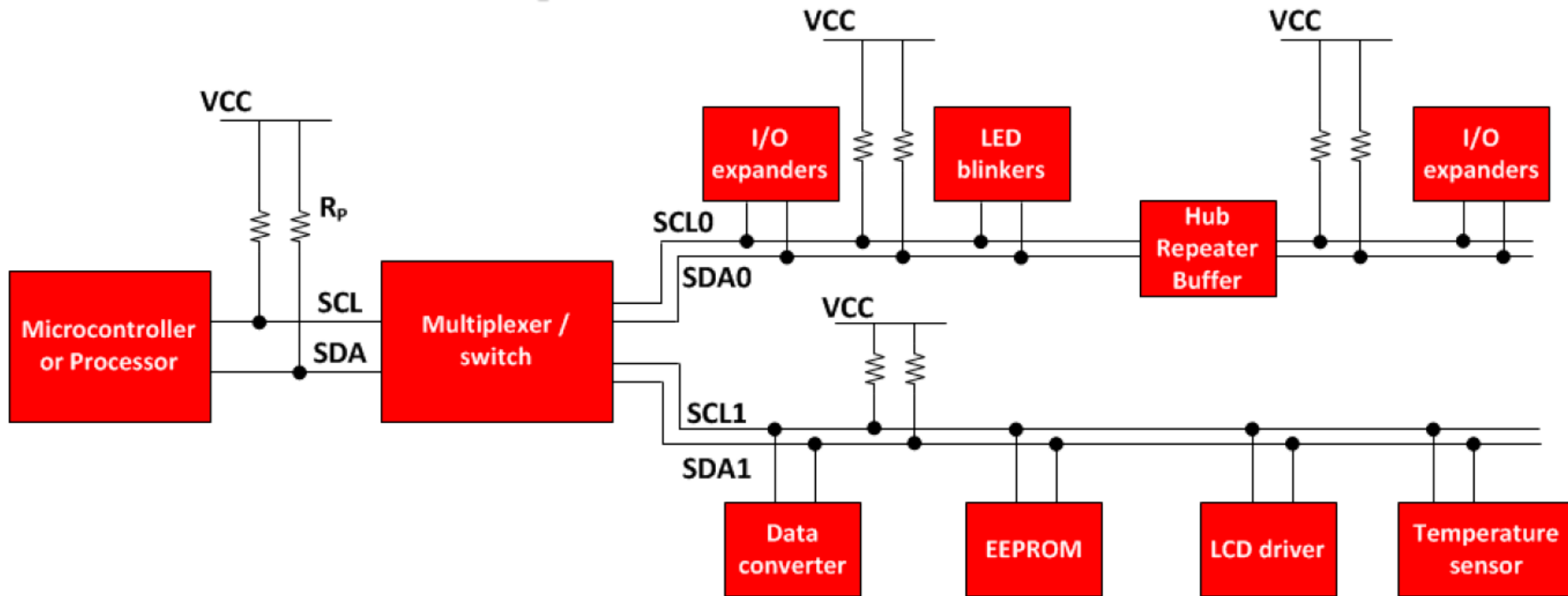


Figure 1. Application Example Showing I2C Communication Between the Different IC's on a System and With Pullup Resistors on I2C Bus

DS3231 – Real Time Clock

DS3231

Extremely Accurate I²C-Integrated
RTC/TCXO/Crystal

Benefits and Features

- Highly Accurate RTC Completely Manages All Timekeeping Functions



- Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100
- Accuracy $\pm 2\text{ppm}$ from 0°C to $+40^{\circ}\text{C}$
- Accuracy $\pm 3.5\text{ppm}$ from -40°C to $+85^{\circ}\text{C}$



- Digital Temp Sensor Output: $\pm 3^{\circ}\text{C}$ Accuracy
- Register for Aging Trim
- $\overline{\text{RST}}$ Output/Pushbutton Reset Debounce Input
- Two Time-of-Day Alarms



- Programmable Square-Wave Output Signal
- Simple Serial Interface Connects to Most Microcontrollers
 - Fast (400kHz) I²C Interface
- Battery-Backup Input for Continuous Timekeeping

DS3231 – Registers

[illegible]

Toolkit commando's deel 1

Register adres	Read byte	Read word	Hex dump (1-32 bytes)
0	rnb	rnw	rn
1 (8 bits)	rbb	rbw	rb
2 (16 bits)	rwb	rww	rw

rbb <slave> <register>

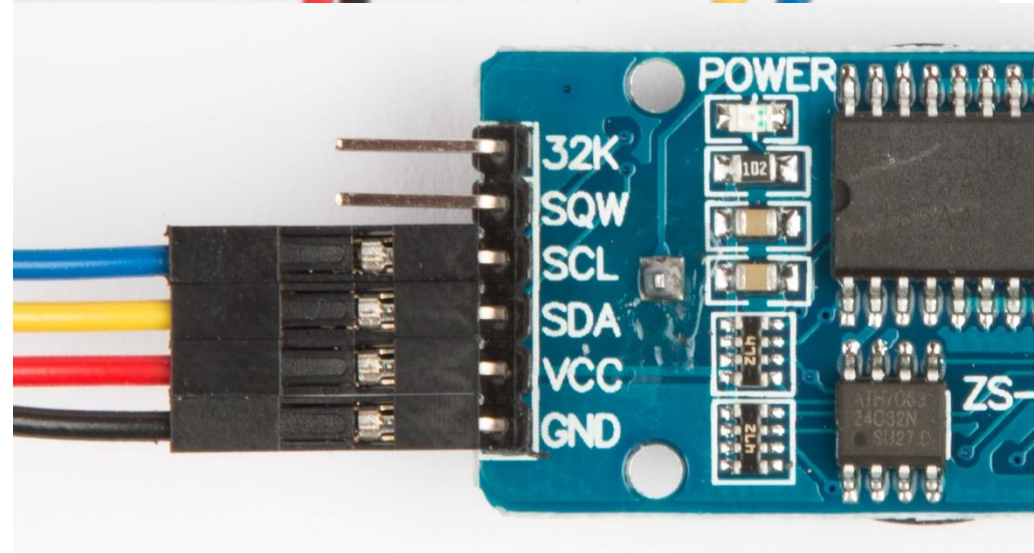
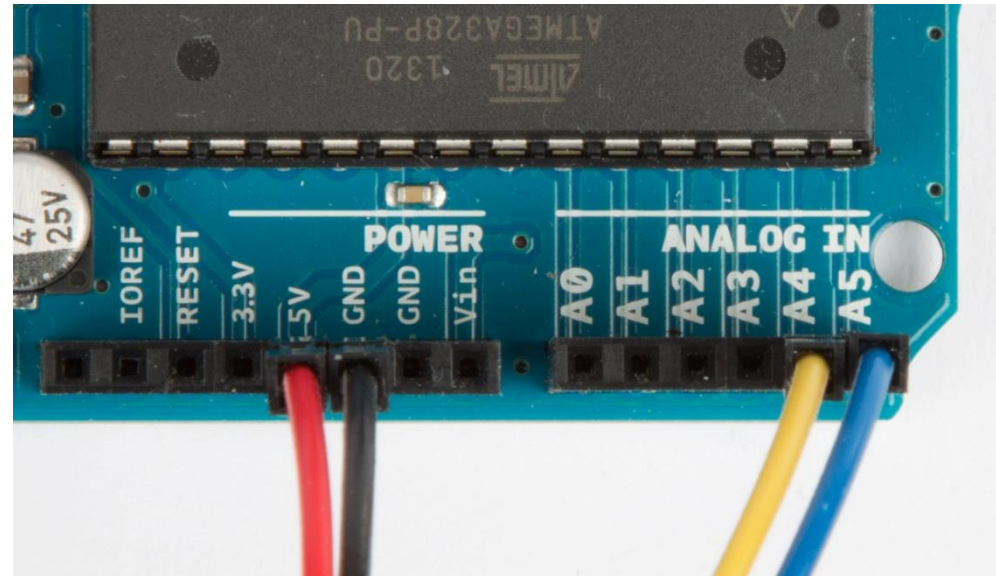
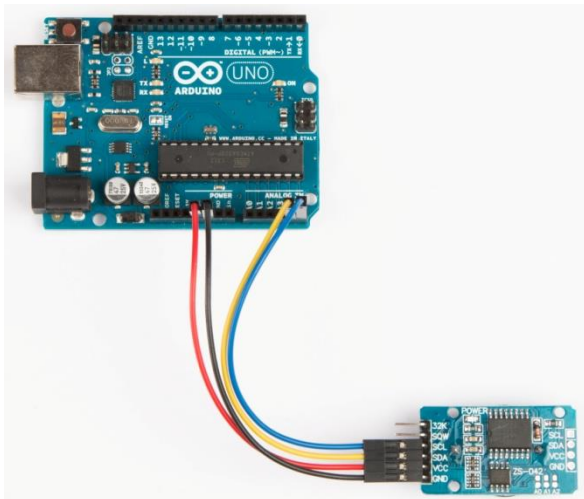
rbw <slave> <register>

rb <slave> <register> <#bytes>

Register adres	Write byte
0	wnb
1 (8 bits)	wbb
2 (16 bits)	wwb

RTC & I2CmTk – Oefening

- ▶ Commando's
 - scan
 - rb 0x68 0 16
 - wbb 0x68 0 0



Real Time Clock oefening (-)

ADDRESS	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 LSB	FUNCTION	RANGE
00h	0	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12/24	AM/PM	10 Hour	Hour				Hours	1–12 + AM/PM 00–23
			20 Hour							
03h	0	0	0	0	0	Day			Day	1–7
04h	0	0	10 Date		Date				Date	01–31
05h	Century	0	0	10 Month	Month				Month/ Century	01–12 + Century
06h	10 Year				Year				Year	00–99

rbb 0x68 0

read van slave 0x68, byte adres 0

rb 0x68 0 16

hex dump, slave 0x68,
16 bytes vanaf byte adres 0

wbb 0x68 0 0

Zet seconden op 0

Probeer ook register 1, minuten.

BCD Binary Coded Decimal

- ▶ Subset van hex
- ▶ Geen A...F
- ▶ 13 hexadecimaal
=> 19 decimaal
=> 13 BCD

1

3

0	0	0	1	0	0	1	1
---	---	---	---	---	---	---	---

0	0	0	1	0	0	1	1
---	---	---	---	---	---	---	---

Toolkit commando's deel 2

Commando	Parameters	Omschrijving
?	-	Print opstart-tekst van toolkit (met versie)
debug	n	Stel debug niveau in (variabele I2cDebug, 0 = uit, 1 = beetje, 2 = veel)
scan	-	Scan de I2C bus op actieve slaves
ram	-	Print hoeveelheid RAM die nog vrij is
fill	Addr Size Value	Vul eeprom met slave address <Addr> en grootte <Size> (in bytes, veelvoud van 16) met waarde <Value>.
logdump	-	LogDump (voor debug van logging)
logclear	-	LogClear – wis eeprom als voorbereiding op nieuwe logreeks

Commando	Parameters	Omschrijving
logwrite	-	Schrijf een logbericht (in een test-formaat) naar de eeprom
logread	-	Lees de logberichten (in een test formaat) uit de eeprom
epatroon	-	Schrijf een test-patroon van 32 bytes naar de eeprom
edump	-	Dump de eerste 64 bytes van de eeprom
tijd	-	Print de huidige tijd van de Real Time Clock
tijd	uu mm ss	Stel de tijd van de Real Time Clock in
blink	-	Knipper 1x met de led op de PCF8574
running	n	Toon n keer het looplicht patroon op de uitgangen van de PCF8574

Toolkit functies deel 1*

```
void HexDump(const void *Data, int Length);  
void BusScan();  
bool AddressProbe(int I2cSlaveAddress);  
void EepromFill(int Slave, int Size, int Value);  
byte Bcd(byte Decimal);
```

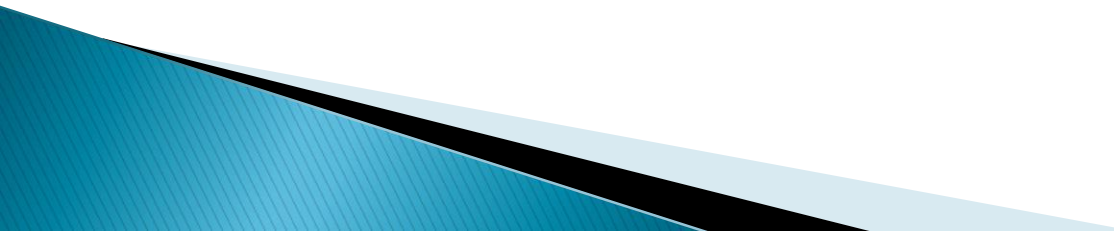
```
bool I2cSendReceive(byte I2cSlaveAddress, byte TxCount,  
                    byte RxCount, const byte *TxBuffer, byte *RxBuffer);
```

```
void LogDump();  
void LogStart();  
void LogWrite(const byte *Data);  
void LogReadOpen();  
bool LogRead(byte *Data);
```

```
char CGet();
```

*zie 'I2C Master Toolkit documentatie' voor deel 2.

Toolkit

- ▶ Handig om interactief een device te verkennen.
 - ▶ Testen, diagnose i2c bus bij problemen.
 - ▶ Een aantal handige functies om te gebruiken in je programma.
 - ▶ En.. je hebt de broncode als je precies wilt weten hoe het zit.
- 

Lees tijd – Oefening (a)

```
case 'a' :
{
    // opgave a

    // Schrijf 1 byte, lees 3 bytes
    I2cTxBuffer[0] = 0;           // Eerste te lezen register
    boolean r = I2cSendReceive(0x68, 1, 3, I2cTxBuffer, I2cRxBuffer);

    if (r == true) { // succes ?
        HexDump(I2cRxBuffer, 3); // Dump 3 bytes
    }
    else {
        printf("I2C fout\n");
    }
    break;
}
```

- ▶ Toegift (b): print tijd als 'hh:mm:ss'

Lees tijd – Resultaat (b)

- ▶ print tijd als 'hh:mm:ss'

```
case 'b' :
{
    // Defening b
    I2cTxBuffer[0] = 0;           // register nummer
    r = I2cSendReceive(DS3231_I2C_ADDRESS, 1, 3, I2cTxBuffer, I2cRxBuffer);
    if (r == true) { // succes ?
        printf("Tijd: %02x:%02x:%02x\n", I2cRxBuffer[2], I2cRxBuffer[1], I2cRxBuffer[0]);
    }
    else {
        printf("I2C fout\n");
    }
    break;
}
```

Temperatuur uitlezen – Oefening (c)

- ▶ Register 0x11 (17) is temperatuur in graden.
- ▶ Lees & print de temperatuur.

```
case 'c' :
{
    // Oefening c
    I2cTxBuffer[0] = 0x11;    // register nummer
    r = I2cSendReceive(DS3231_I2C_ADDRESS, 1, 1, I2cTxBuffer, I2cRxBuffer);
    if (r == true) { // succes ?
        printf("Temperatuur: %d graden\n", I2cRxBuffer[0]);
    }
    else {
        printf("I2C fout\n");
    }
    break;
}
```

Benefits and Features

- Digital Temp Sensor Output: $\pm 3^{\circ}\text{C}$ Accuracy

Temperatuur

ADDRESS	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 LSB	FUNCTION	RANGE
11h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	MSB of Temp	—
12h	DATA	DATA	0	0	0	0	0	0	LSB of Temp	—

Benefits and Features

- Digital Temp Sensor Output: $\pm 3^{\circ}\text{C}$ Accuracy

- ▶ Accuracy = Nauwkeurigheid
- ▶ Resolution = stapgrootte
- ▶ Repeatability = herhaalbaarheid

Temperatuur uitlezen – Oefening (–)

ADDRESS	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 LSB	FUNCTION	RANGE
11h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	MSB of Temp	—
12h	DATA	DATA	0	0	0	0	0	0	LSB of Temp	—

Benefits and Features

- Digital Temp Sensor Output: $\pm 3^{\circ}\text{C}$ Accuracy

rbb 0x68 0x11

read byte van slave 0x68, byte
adres 0x11

rbw 0x68 0x11

read word van slave 0x68,
byte adres 0x11

Temperatuur uitlezen – Resultaat

ADDRESS	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 LSB	FUNCTION	RANGE
11h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	MSB of Temp	—
12h	DATA	DATA	0	0	0	0	0	0	LSB of Temp	—

Benefits and Features

- Digital Temp Sensor Output: $\pm 3^{\circ}\text{C}$ Accuracy

rbb 0x68 0x11

rbw 0x68 0x11

```
I2c read slave 0x68, Reg: 17 (0x11), Data: 23 (0x17)
```

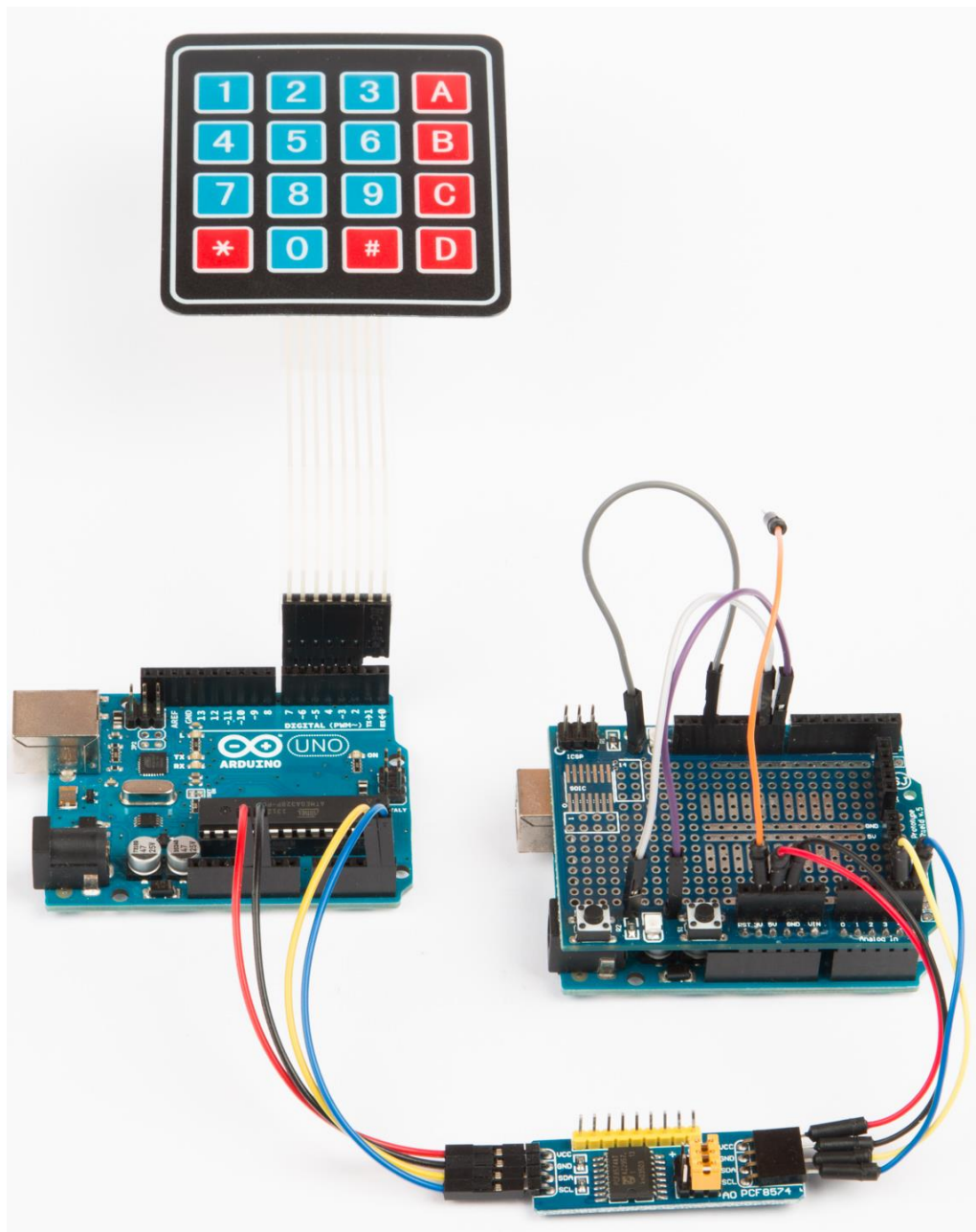
```
TkCmd 'rbb' gereed.
```

```
I2c read slave 0x68, Reg: 17 (0x11), Data: 5952 (0x1740)
```

```
TkCmd 'rbw' gereed.
```

DS3231 – Registers

[illegible]



Huiswerk

1. Test je eigen I2C slave met de toolkit.

Voor de C liefhebbers:

2. Print tijd als hh:mm:ss (Toegift b)
3. Programma: print temperatuur in hoge resolutie (0.25 graden)