

Arduino en I2C – Dag 2

Huiswerk

1. Arduino LED (pin 13) bedienen met input 0 van de PCF8574 (opgave 'y').

i2c_master_ws1

I2CmTk

```
// Defining y
Wire.requestFrom(PCF8574A_I2C_ADDRESS, 1);
if (Wire.available() == 1) {
    pinMode(13, OUTPUT);
    digitalWrite(13, Wire.read() & 1);
}
```

Huiswerk

2. PCF8574 LED laten knipperen (opgave 'z')

i2c_master_ws1

I2CmTk

```
// Defining z
static bool Knipper;
if (Knipper) {
    Wire.beginTransmission(PCF8574A_I2C_ADDRESS);
    Wire.write(0xFF-0x20); // Alle bits hoog, behalve bit 5
    Wire.endTransmission();
    Knipper = false;
} else {
    Wire.beginTransmission(PCF8574A_I2C_ADDRESS);
    Wire.write(0xFF);
    Wire.endTransmission();
    Knipper = true;
}
}
```

Clock snelheid

- ▶ Geen minimum
 - ▶ Standard-mode: 100 kbps
 - ▶ Fast-mode: 400 kbps
 - ▶ Fast-mode Plus: 1 Mbps
 - ▶ High-speed mode 3.4 Mbps
-
- ▶ Modes zijn backward compatible
 - ▶ Wij gaan uit van Standard-mode.

UM10204

I²C-bus specification and user manual

Rev. 6 — 4 April 2014

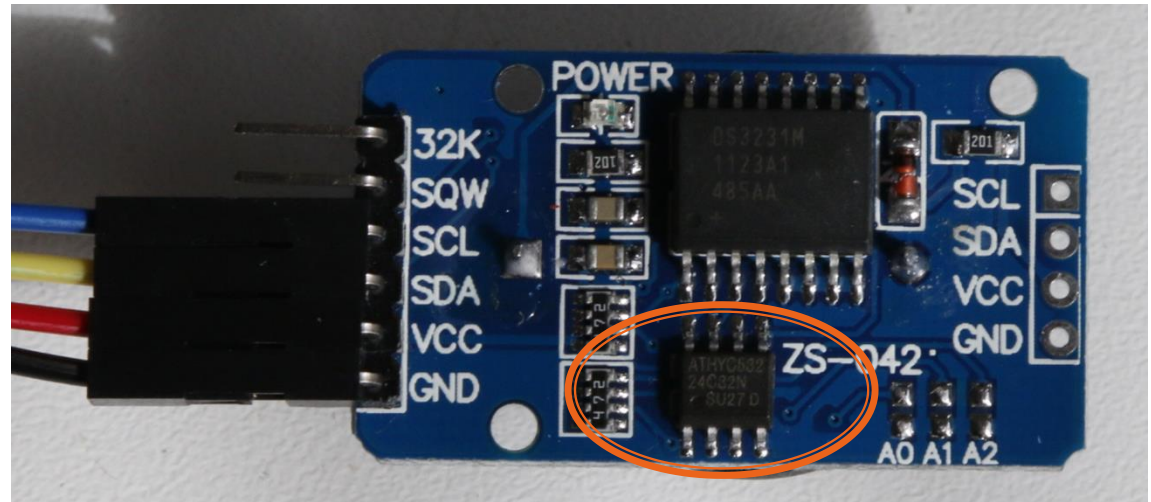
I2C Eeprom



2-Wire Serial EEPROM

32K (4096 x 8)

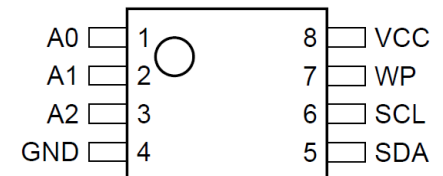
AT24C32



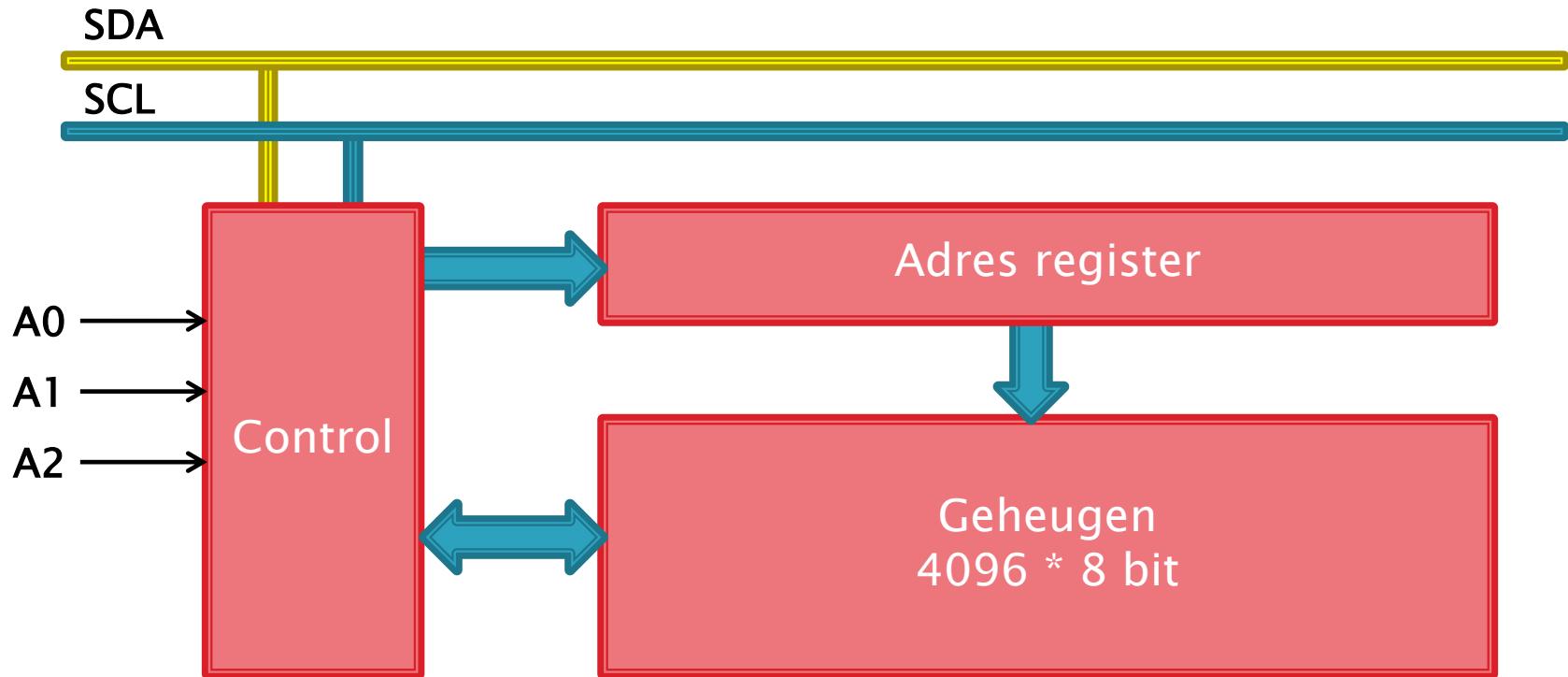
Pin Configurations

Pin Name	Function
A0 - A2	Address Inputs
SDA	Serial Data
SCL	Serial Clock Input
WP	Write Protect

8-Pin TSSOP

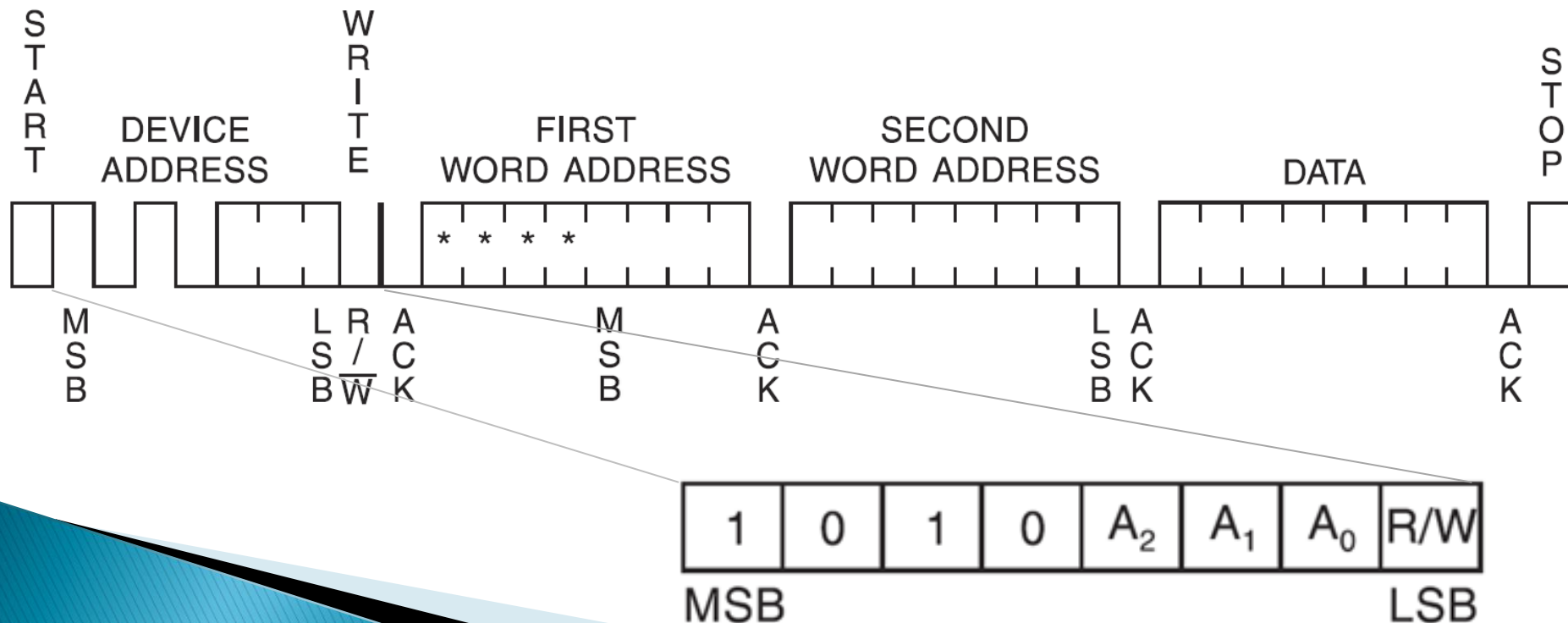


I2C Eeprom blokschema



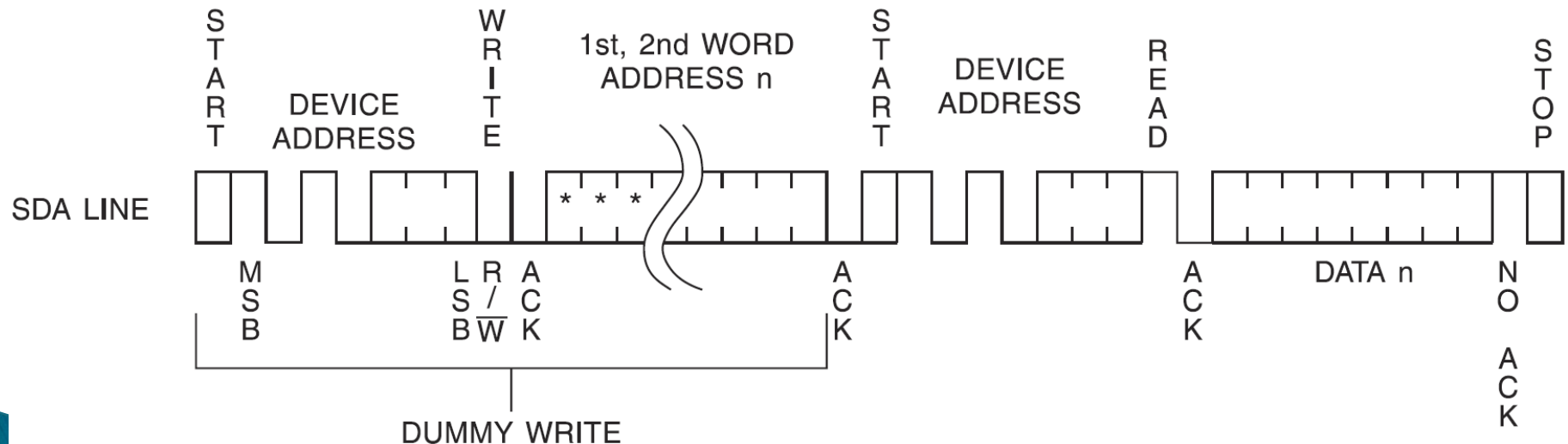
I2C – Eeprom write

- ▶ Device adressering (7+1 bit)
- ▶ Geheugenlocatie adressering (4+12bit)
- ▶ Data byte (8 bit)



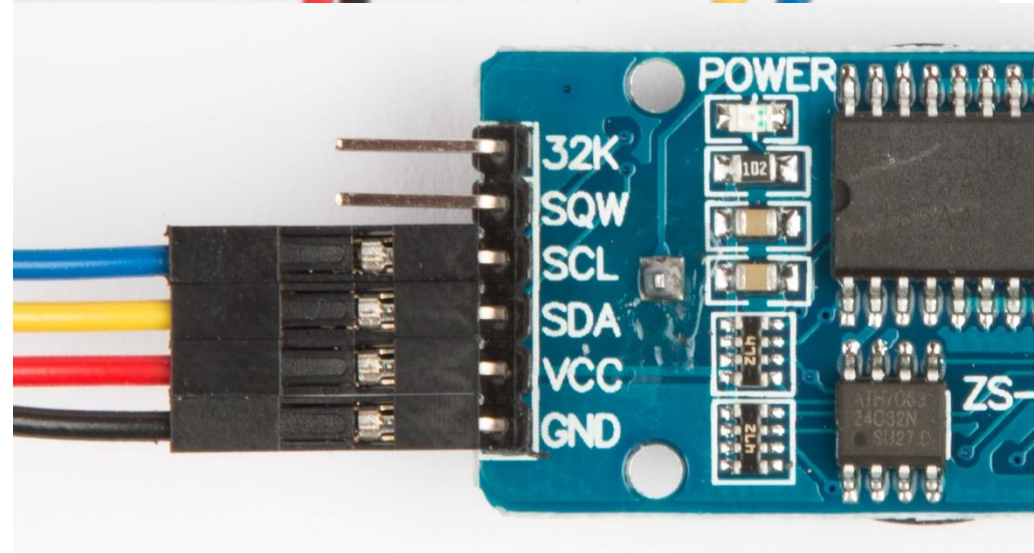
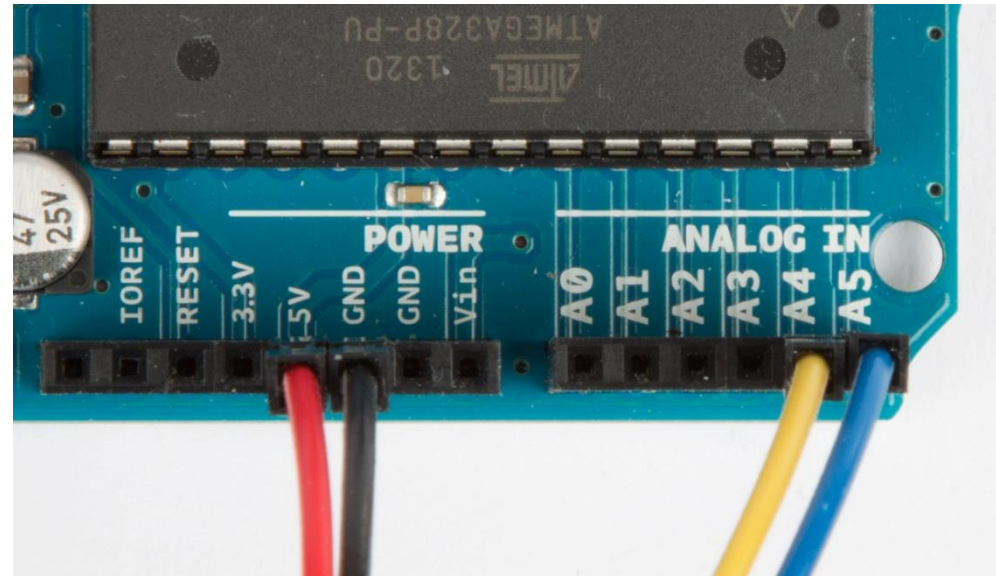
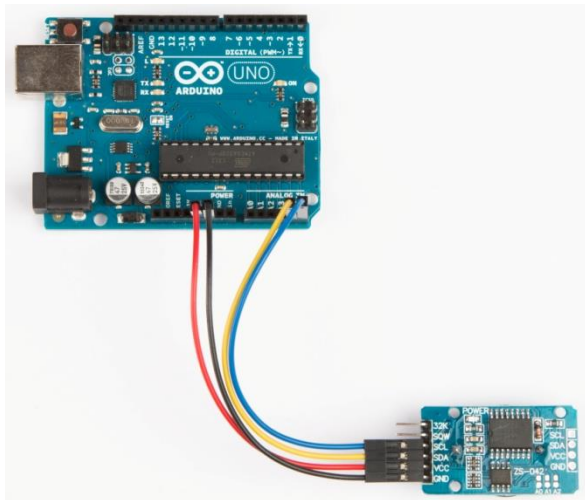
I2C – Eeprom read

- ▶ Device adressering (7+1 bit)
- ▶ Geheugenlocatie adressering (4+12bit)
- ▶ Device adressering (7+1 bit)
- ▶ Data byte (8 bit)



Eeprom & I2CmTk – Oefening

- ▶ Commando's
 - scan
 - edump
 - epatroom



Resultaat – scan

COM8

```
I2CmTk - I2C Master Toolkit versie 0.5
Ontwikkeld voor de Workshop 'Arduino & I2C'.
(c) 2016-2017 Karel Dupain & Joep Suijs
Gecompileerd: Nov 20 2016 13:06:33
.....
.....
.....57.....
.....68.....
TkCmd 'scan' gereed.
```

Resultaat – edump

```
I2CmTk - I2C Master Toolkit versie 0.5
Ontwikkeld voor de Workshop 'Arduino & I2C'.
(c) 2016-2017 Karel Dupain & Joep Suijs
Gecompileerd: Nov 20 2016 13:06:33
TkCmd 'epatroon' gereed.
0000: 44 65 20 77 6f 72 6b 73 68 6f 70 20 76 61 6e 20 De workshop van
0010: 4b 61 72 65 6c 20 65 6e 20 4a 6f 65 70 2e 00 00 Karel en Joep...
0020: de c0 de 00 ab ba ba be 00 ca fe f0 0d 00 be ef .....
0030: 00 c0 ff ee fe ed 00 fa ca de 00 de ad 00 c0 de .....
TkCmd 'edump' gereed.
```

rw <slave> <register> <#bytes>

Eeprom Write – Oefening (a)

- Schrijf een byte naar locatie 0x1E (30)

```
i2c_master_ws2  I2CmTk
case 'a' :
{
    // Oefening a
    Wire.beginTransaction(0x57); // slave adres
    Wire.write(0);               // hoge adres-byte
    Wire.write(0x1E);            // lage adres-byte
    Wire.write('*') ;            // data (0x2A)
    Wire.endTransmission();

    break;
}
```

- Toegift: 2 bytes schrijven (b).

```
I2CmTk - I2C Master Toolkit versie 0.5
Ontwikkeld voor de Workshop 'Arduino & I2C'.
(c) 2016-2017 Karel Dupain & Joep Suijs
Gecompileerd: Nov 20 2016 13:06:33
TkCmd 'epatroon' gereed.
0000: 44 65 20 77 6f 72 6b 73 68 6f 70 20 76 61 6e 20 De w
0010: 4b 61 72 65 6c 20 65 6e 20 4a 6f 65 70 2e 00 00 Kare
0020: de c0 de 00 ab ba ba be 00 ca fe f0 0d 00 be ef ....
0030: 00 c0 ff ee fe ed 00 fa ca de 00 de ad 00 c0 de ....
TkCmd 'edump' gereed.
```

Eeprom Write – Oefening (b)

- Schrijf twee byte vanaf locatie 0x1E (30)

```
i2c_master_ws2  I2CmTk
case 'b' :
{
    // Oefening b
    Wire.beginTransaction(EEPROM_I2C_ADDRESS);
    Wire.write(0);           // hoge adres-byte
    Wire.write(0x1E);        // lage adres-byte
    Wire.write('*') ;        // data (0x2A)
    Wire.write('*') ;        // data (0x2A)
    Wire.endTransmission();

    break;
}
```

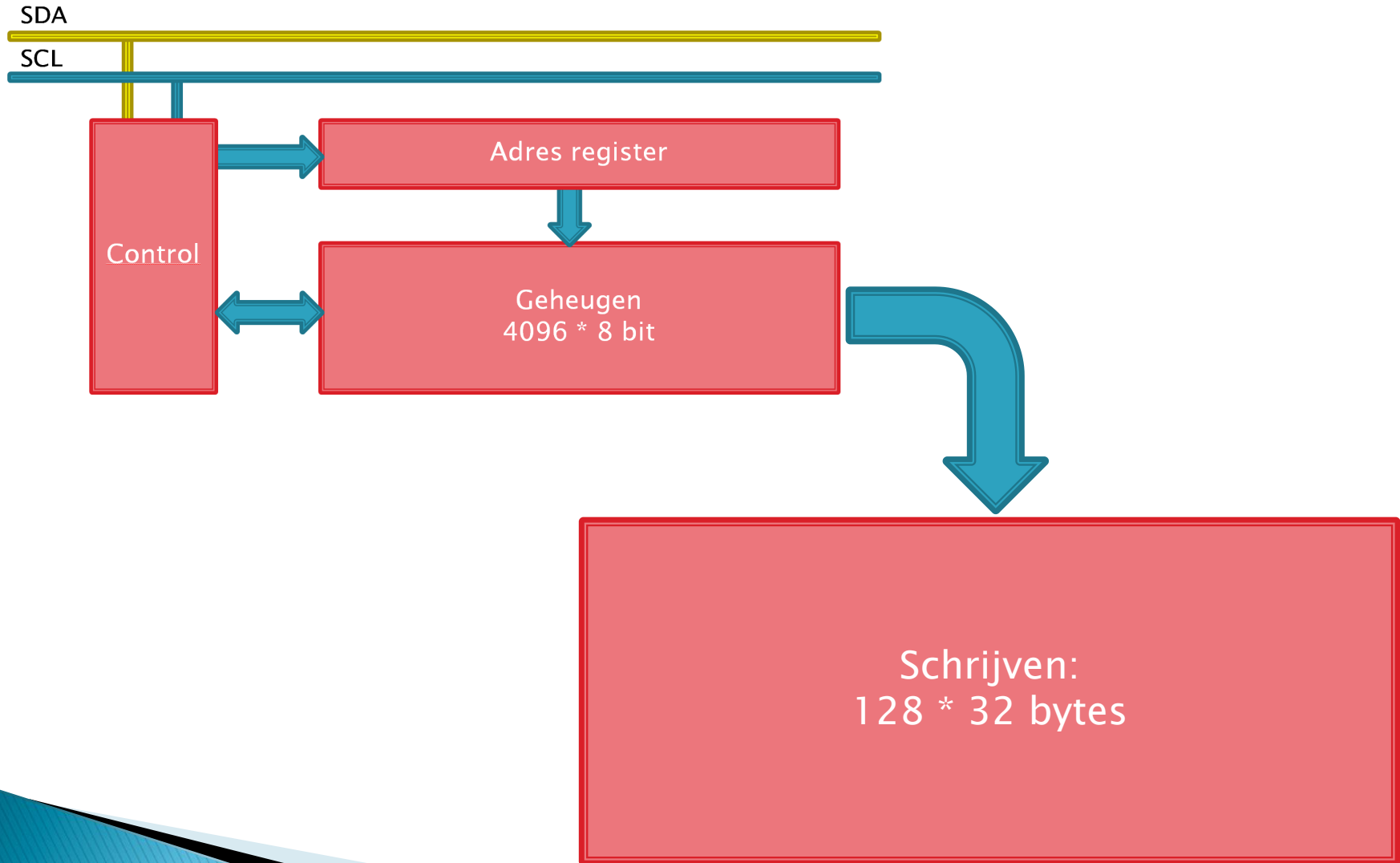
- Toegift: 3 bytes schrijven (c).

Eeprom Write – Resultaat (c)

```
// Oefening c -  
Wire.beginTransmission(EEPROM_I2C_ADDRESS);  
Wire.write(0);           // hoge adres-byte  
Wire.write(0x1E);        // lage adres-byte  
Wire.write('*') ;        // data (0x2A)  
Wire.write('*') ;        // data (0x2A)  
Wire.write('*') ;        // data (0x2A)  
Wire.endTransmission();
```

```
0000: 2a 65 20 77 6f 72 6b 73 68 6f 70 20 76 61 6e 20  *e workshop van  
0010: 4b 61 72 65 6c 20 65 6e 20 4a 6f 65 70 2e 2a 2a  Karel en Joep.**  
0020: de 00 de 00 ab ba ba be 00 ca fe f0 0d 00 be ef  .....  
0030: 00 c0 ff ee fe ed 00 fa ca de 00 de ad 00 c0 de  .....  
TkCmd 'edump' gereed.
```

Eeprom – Page write



Eeprom Write – Oefening (d)

- ▶ epatroon
- ▶ edump
- ▶ 'd'

i2c_master_ws2

I2CmTk

```
case 'd' :
{
    // Oefening d

    // Schrijf laatste 2 bytes van blok
    Wire.beginTransaction(EEPROM_I2C_ADDRESS);
    Wire.write(0); // hoge adres-byte
    Wire.write(0x1E); // lage adres-byte
    Wire.write('*') ; // data (0x2A)
    Wire.write('*') ; // data (0x2A)
    Wire.endTransmission();

    // Schrijf 1e byte van volgend blok
    Wire.beginTransaction(EEPROM_I2C_ADDRESS);
    Wire.write(0); // hoge adres-byte
    Wire.write(0x20); // lage adres-byte
    Wire.write('*') ; // data (0x2A)
    Wire.endTransmission();

    break;
}
```


Eeprom Write – resultaat (d)

I2CmTk - I2C Master Toolkit versie 0.5

Ontwikkeld voor de Workshop 'Arduino & I2C'.

(c) 2016-2017 Karel Dupain & Joep Smits

Gecompileerd: Nov 20 2016 10:33

TkCmd 'epatroon' gereed.

0000:	44	65	20	77	6f	72	6b	73	68	6f	70	20	76	61	6e	20	De workshop van
0010:	4b	61	72	65	6c	20	65	6e	20	4a	6f	65	70	2e	00	00	Karel en Joep...
0020:	de	c0	de	00	ab	ba	ba	be	00	ca	fe	f0	0d	00	be	ef
0030:	00	c0	ff	ee	fe	ed	00	fa	ca	de	00	de	ad	00	c0	de

TkCmd 'edump' gereed.

Commande 'd' uitgevoerd

0000:	44	65	20	77	6f	72	6b	73	68	6f	70	20	76	61	6e	20	De workshop van
0010:	4b	61	72	65	6c	20	65	6e	20	4a	6f	65	70	2e	2a	2a	Karel en Joep.**
0020:	de	c0	de	00	ab	ba	ba	be	00	ca	fe	f0	0d	00	be	ef
0030:	00	c0	ff	ee	fe	ed	00	fa	ca	de	00	de	ad	00	c0	de

TkCmd 'edump' gereed.

Schone start

Eeprom Write – Oefening (e)

- ▶ epatroon
- ▶ edump
- ▶ 'e'

i2c_master_ws2

I2CmTk

```
case 'e' :
{
    // Oefening e

    // Schrijf laatste 2 bytes van blok
    Wire.beginTransaction(EEPROM_I2C_ADDRESS);
    Wire.write(0); // hoge adres-byte
    Wire.write(0x1E); // lage adres-byte
    Wire.write('*') ; // data (0x2A)
    Wire.write('*') ; // data (0x2A)
    r = Wire.endTransmission();

    if (r != 0) {
        printf("Error: i2c fout1 (%d)\n", r);
        break;
    }

    // Schrijf 1e byte van volgend blok
    Wire.beginTransaction(EEPROM_I2C_ADDRESS);
    Wire.write(0); // hoge adres-byte
    Wire.write(0x20); // lage adres-byte
    Wire.write('*') ; // data (0x2A)
    r = Wire.endTransmission();

    if (r != 0) {
        printf("Error: i2c fout2 (%d)\n", r);
        break;
    }
    break;
}
```

Eeprom Write – resultaat (e)

I2CmTk - I2C Master Toolkit versie 0.5

Ontwikkeld voor de Workshop 'Arduino & I2C'.

(c) 2016-2017 Karel Dupain & Joep Suijs

Gecompileerd: Nov 20 2016 15:53:33

TkCmd 'epatroon' gereed.

0000:	44 65 20 77 6f 72 6b 73 68 6f 70 20 76 61 6e 20	De workshop van
0010:	4b 61 72 65 6c 20 65 6e 20 4a 6f 65 70 2e 00 00	Karel en Joep...
0020:	de c0 de 00 ab ba ba be 00 ca fe f0 0d 00 be ef
0030:	00 c0 ff ee fe ed 00 fa ca de 00 de ad 00 c0 de

TkCmd 'edump' gereed.

Error: i2c fout2 (2)

Commmando 'e' uitgevoerd

0000:	44 65 20 77 6f 72 6b 73 68 6f 70 20 76 61 6e 20	De workshop van
0010:	4b 61 72 65 6c 20 65 6e 20 4a 6f 65 70 2e 2a 2a	Karel en Joep.**
0020:	de c0 de 00 ab ba ba be	
0030:	00 c0 ff ee fe ed 00 fa	

TkCmd 'edump' gereed.

Schone start

Foutmelding
command 'e'

Wire.endTransmission();

0:success

1:data too long to fit in transmit buffer

2:received NACK on transmit of address

3:received NACK on transmit of data

4:other error

Eeprom Write – Oefening (f)

- ▶ epatroon
- ▶ edump
- ▶ 'f'

```
I2c_master_ws2 I2CmTk
case 'f' :
{
    // Oefening f

    // Schrijf laatste 2 bytes van blok
    Wire.beginTransaction(EEPROM_I2C_ADDRESS);
    Wire.write(0);           // hoge adres-byte
    Wire.write(0x1E);        // lage adres-byte
    Wire.write('*');         // data (0x2A)
    Wire.write('*');         // data (0x2A)
    Wire.endTransmission();

    while (AddressProbe(EEPROM_I2C_ADDRESS) == false) {
        // Wacht tot slave weer reageert (write cycle afgerond)
    }

    // Schrijf 1e byte van volgend blok
    Wire.beginTransaction(EEPROM_I2C_ADDRESS);
    Wire.write(0);           // hoge adres-byte
    Wire.write(0x20);        // lage adres-byte
    Wire.write('*');         // data (0x2A)
    Wire.endTransmission();
    break;
}
```



```
while (AddressProbe(EEPROM_I2C_ADDRESS) == false) {
    // Wacht tot slave weer reageert (write cycle afgerond)
}
```

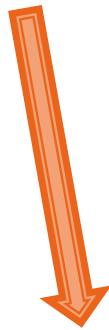
Eeprom Write- Resultaat (f)

Commmmando 'f' uitgevoerd

```
0000: 44 65 20 77 6f 72 6b 73 68 6f 70 20 76 61 6e 20 De workshop van
0010: 4b 61 72 65 6c 20 65 6e 20 4a 6f 65 70 2e 2a 2a Karel en Joep.**
0020: 2a c0 de 00 ab ba ba be 00 ca fe f0 0d 00 be ef *.....
0030: 00 c0 ff ee fe ed 00 fa ca de 00 de ad 00 c0 de .....
TkCmd 'edump' gereed.
```

Eeprom Read – ontvangstcontrole

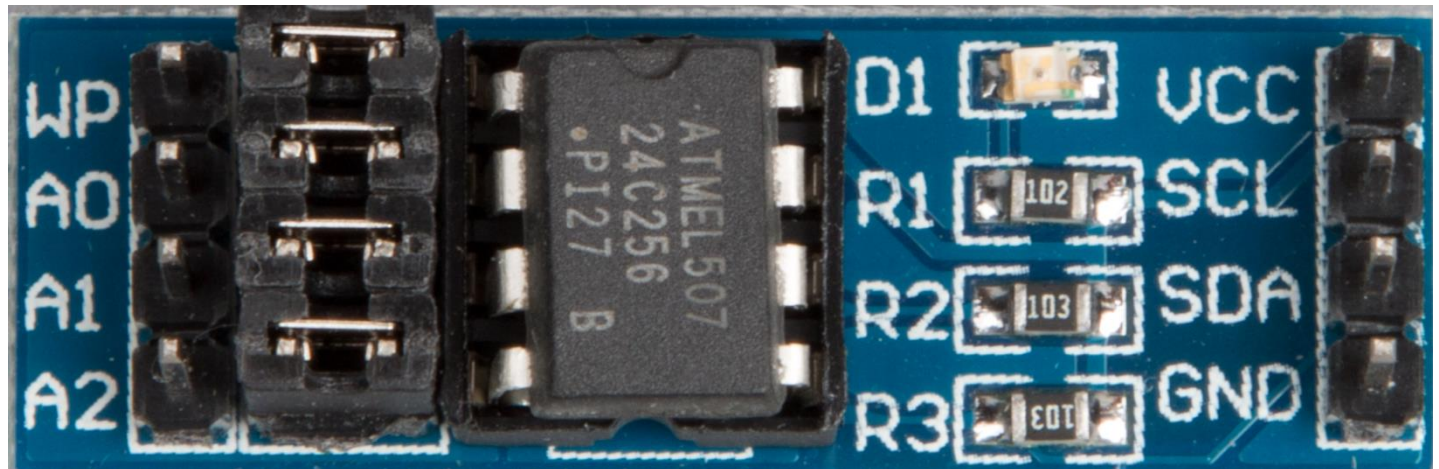
```
void edump()  
{  
    byte I2cTxBuffer[2];  
    byte I2cRxBuffer[16];  
  
    for (int a=0; a<64; a+=16) {  
  
        // Stuur gewenste adres  
        I2cTxBuffer[0] = 0;  
        I2cTxBuffer[1] = a;  
        Wire.beginTransaction(EEPROM_I2C_ADDRESS);  
        for (int i=0; i<2; i++) {  
            Wire.write(I2cTxBuffer[i]); //  
        }  
        int r = Wire.endTransmission();  
        if (r != 0) return; // error  
  
        // lees 16 data bytes  
        Wire.requestFrom(EEPROM_I2C_ADDRESS, 16);  
        if (Wire.available() != 16) return; // error  
  
        // Plaats data in de buffer  
        for (int i=0; i<16; i++) {  
            I2cRxBuffer[i] = Wire.read();  
        }  
  
        // Dump buffer  
        HexDump(I2cRxBuffer, 16, a);  
    }  
}
```



```
// lees 16 data bytes  
Wire.requestFrom(EEPROM_I2C_ADDRESS, 16);  
if (Wire.available() != 16) return; // error
```


Logging

- ▶ PageWrite grens
- ▶ Arduino grens
- ▶ Blokjes data, 1^e byte is vlag.



Logging commando's

- ▶ logclear -> LogStart(); – Eeprom wissen
- ▶ logwrite -> demo routine – schrijf 1 record
- ▶ logread -> demo – lees alle records
- ▶ logdump-> Voor debuggen...

```
// logging configuratie
#define LOG_BLOCK_SIZE      16 // Macht van 2, maximaal 32
#define LOG_MAX_DATA        15 // maximaal blocksize-1 en maximaal 29
#define LOG_NR_BLOCKS       256 // max eeprom grootte (bytes) / LOG_BLOCK_SIZE
#define LOG_EEPROM_ADDRESS 0x57 // I2C slave adres
```


Logging demofuncties

```
byte Data[LOG_MAX_DATA];
```

► logwrite

```
long Tijd = millis() / 1000; // tijd in seconde stappen  
Data[0] = Tijd / 256;  
Data[1] = Tijd;  
Data[2] = digitalRead(8);
```

```
LogWrite(Data); // schrijf naar log
```

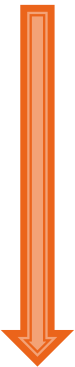
► logread

```
LogReadOpen(); // lees vanaf 1e record
```

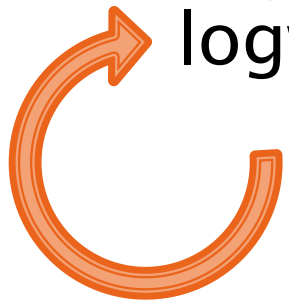
```
for (;;) {  
    bool r = LogRead(Data);  
  
    if (r) {  
        // success  
        int Tijd = Data[0] * 256L + Data[1];  
        int Pin8 = Data[2];  
        printf("Tijd: %d sec, pin8: %d\n", Tijd, Pin8);  
    }  
    else {  
        printf("Log end.\n");  
        return;  
    }  
}
```

Logging - Oefening (-)

logclear



logwrite



```
I2CmTk - I2C Master Toolkit versie 0.81
Ontwikkeld voor de Workshop 'Arduino & I2C'.
(c) 2016-2017 Karel Dupain & Joep Suijs
Gecompileerd: Feb 19 2017 15:44:23
Tijd: 23 sec, pin8: 1
Tijd: 26 sec, pin8: 1
Tijd: 26 sec, pin8: 1
Tijd: 27 sec, pin8: 1
Tijd: 28 sec, pin8: 1
Tijd: 102 sec, pin8: 1
Tijd: 115 sec, pin8: 1
Tijd: 120 sec, pin8: 1
Tijd: 128 sec, pin8: 1
Tijd: 132 sec, pin8: 1
Tijd: 776 sec, pin8: 1
Tijd: 782 sec, pin8: 1
Tijd: 10 sec, pin8: 1
Tijd: 1081 sec, pin8: 1
Tijd: 3342 sec, pin8: 1
Log end.
TkCmd 'logread' gereed.
```

Huiswerk

1. Maak een echte logging toepassing (robot, sensor).
2. Bekijk de datasheet van de DS3231