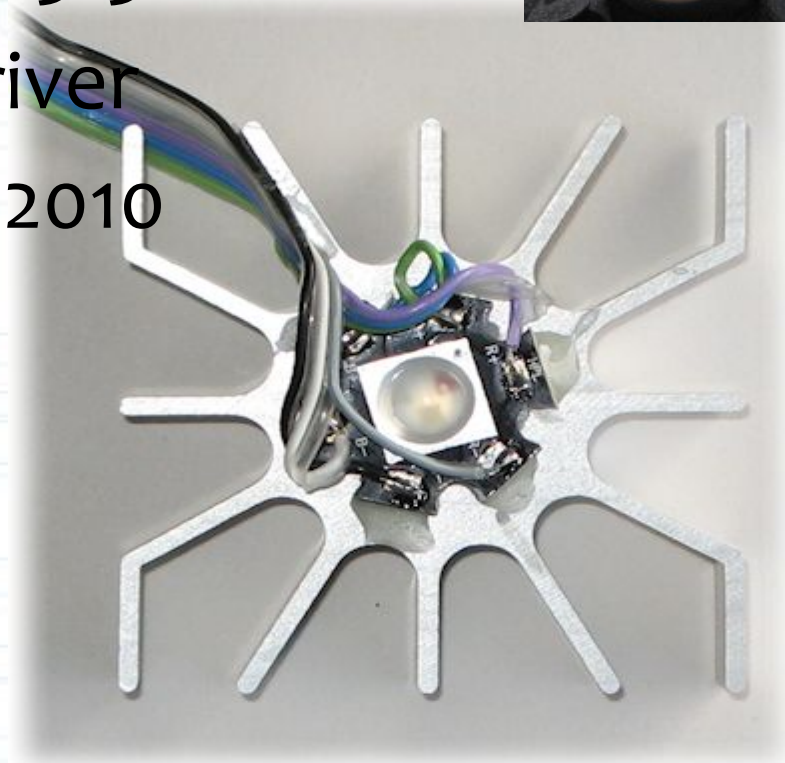


EmSE presentatie



Starten met de ATtiny2313:
Een RGB powerLED driver
Kees Moerman, 13 juli 2010



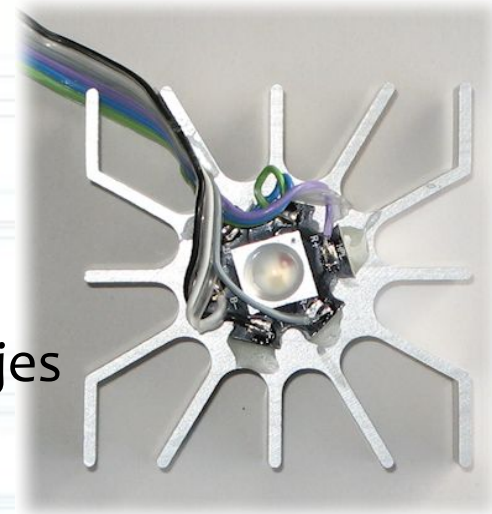
PDIP/SOIC

(RESET/dW) PA2	1	20	VCC
(RXD) PD0	2	19	PB7 (UCSK/SCL/PCINT7)
(TXD) PD1	3	18	PB6 (MISO/DO/PCINT6)
(XTAL2) PA1	4	17	PB5 (MOSI/DI/SDA/PCINT5)
(XTAL1) PA0	5	16	PB4 (OC1B/PCINT4)
(CKOUT/XCK/INT0) PD2	6	15	PB3 (OC1A/PCINT3)
(INT1) PD3	7	14	PB2 (OC0A/PCINT2)
(T0) PD4	8	13	PB1 (AIN1/PCINT1)
(OC0B/T1) PD5	9	12	PB0 (AIN0/PCINT0)
GND	10	11	PD6 (ICP)



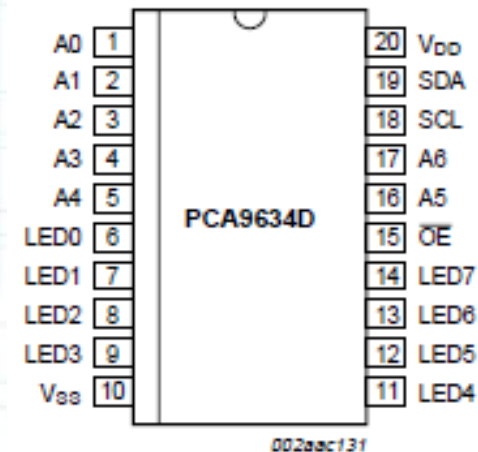
Het probleem

- Ik wil een LED controller maken
 - Voor zelfgemaakte RGB power-LED lampjes
- Om aan mijn thuisnetwerk te hangen
 - Inprogrammeren kleureffecten, overgangen en zo
 - Rustig inslapen/wakker worden
- Ik heb een netwerkcontroller met I²C-bus
 - Zowel een NSLU2 als een Rabbit bord
- En er zijn I²C PWM-controller chips
 - NXP PCA9634, 8-kanaals 8-bit PWM



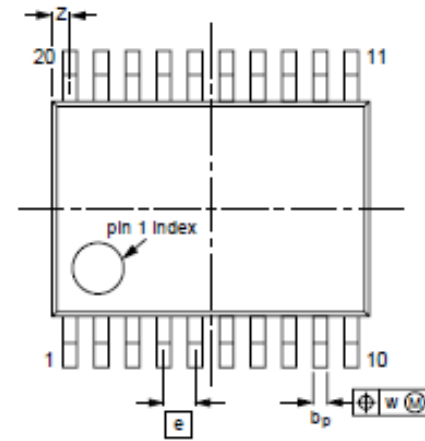
Problem solved

- Einde presentatie...



Het probleem – deel 2

- Er zijn I²C PWM-controller chips
 - NXP PCA9634, 8-kanaals 8-bit PWM
- Maar, alleen leverbaar in TSSOP20 behuizing of kleiner
 - Pitch ('e') 0.65 mm; voor mij niet te doen qua solderen
 - Of ik moet fotografisch printen gaan etsen en zo
- **En** niet via normale particuliere kanalen
 - Alleen DigiKey, met €18 adminkosten plus €???
 - verzendkosten; minimaal 5 stuks
 - RS, Farnell, etc hebben/leveren niet in kleine aantallen
 - Tenzij ik beloof later 'in productie' >= 2000 stuks af te nemen
- Geen bruikbare alternatieven gevonden



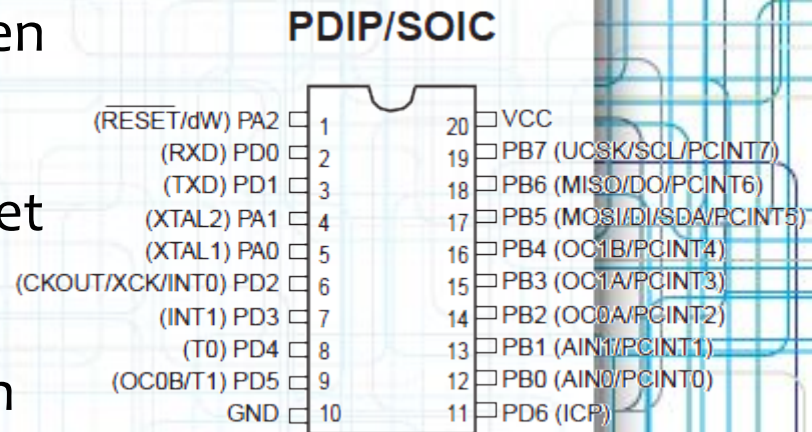
Hoe op te lossen??

... Zelf maken

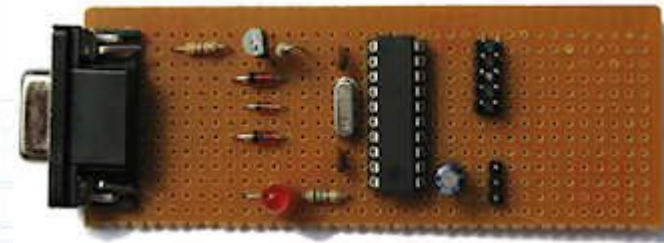
- Ik ben een hobbyist; kan ik het niet zelf maken?
- Bijvoorbeeld via een simpele microcontroller
 - Kan zelf functionaliteit bepalen
 - Kan zelf behuizing kiezen
 - Moet zelf programma maken
 - Is veel leuker ...
- Wat zijn de voorwaarden
 - De chip moet goed verkrijgbaar (en betaalbaar) zijn
 - De chip moet voldoende mogelijkheden hebben
 - Bij voorkeur I²C support, PWM outputs, timers
 - Goede ontwikkeltools en programmer
 - Liefst C/C++ en grafische IDE

Processorkeuze

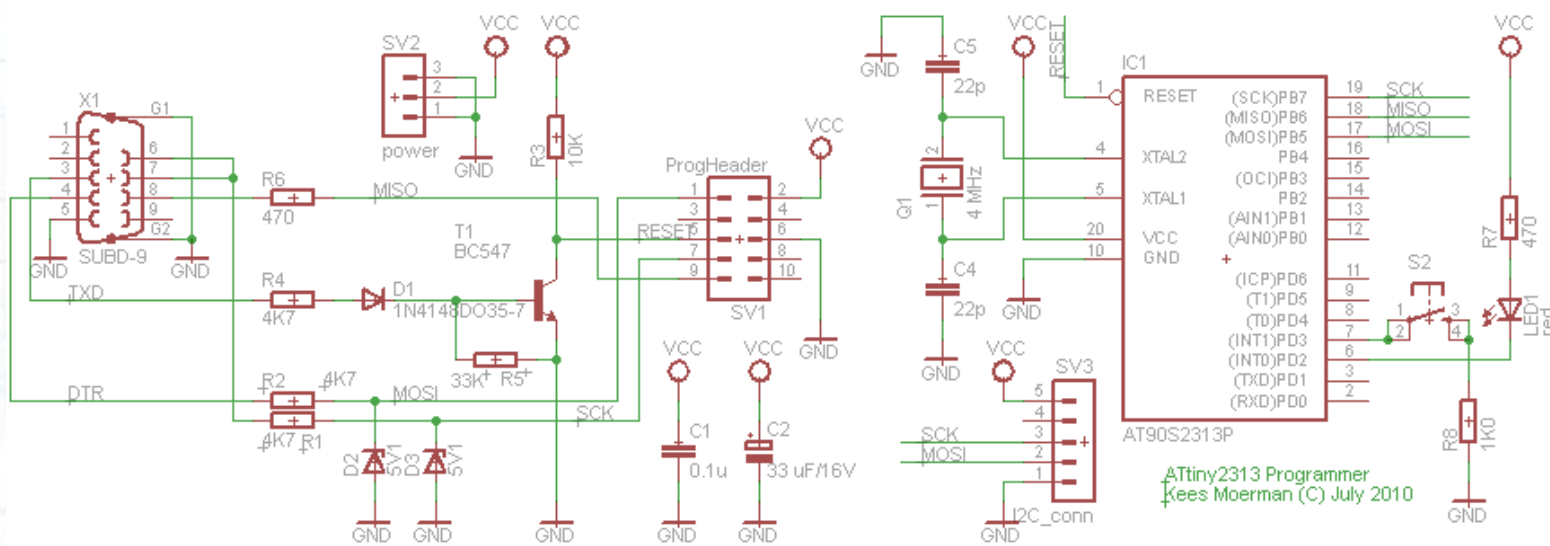
- Op zoek naar een programmeerbare controller
 - 8 bit; goede tools; veel gebruikt; handige behuizing
 - Low power; eenvoudig te programmeren
- Keuze verkleind naar PIC of AVR
 - Aan de hand van wat bladeren op het net
- Gekozen voor Atmel ATtiny2313
 - Veel gebruikt → veel info over te vinden
 - Deel van uitgebreide Atmel familie
 - Veel varianten; later handig voor andere projecten
 - C/C++ programmeerbaar
 - Eenvoudig en low-cost te programmeren
 - Low power/low voltage, handig voor andere projecten
 - I²C, PWM outputs, DIL-20 behuizing, < €2.50



Programmer

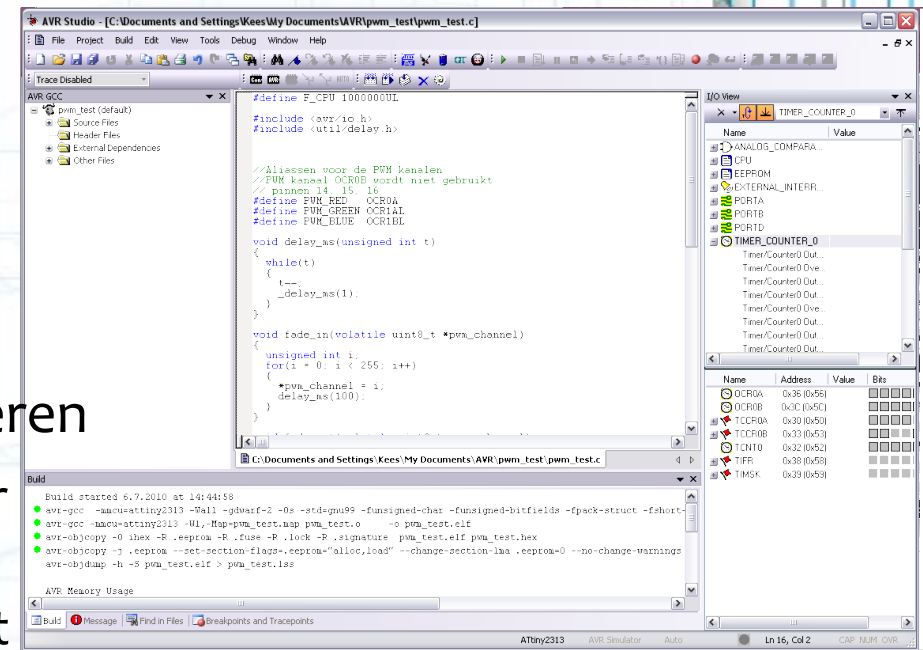


- Wat is nodig: een programmer
 - Low cost, bruikbaar op PC onder Windows XP en/of Ubuntu
- Gekozen voor seriële programmer
 - Parallele versie lastig in gebruik en onveilig
 - Access naar parallele poort in XP afgeschermd
 - Ongebufferd → opblazen van PC poort
- Gebaseerd op schema Olimex AVR-PG1
 - Wat minimale wijzigingen; ook als app.bord bruikbaar



Ontwikkelomgeving

- Wil in C programmeren
 - Niet weer een nieuwe assembler leren
 - Lezen pak ik snel genoeg op, maar alle details om zelf te schrijven...
Ik kijk wel naar de compiler output
 - Win-AVR: GCC compiler voor AVR 8-bit serie
 - Goede documentatie, ook over runtime library AVR-libc
- Wil een IDE omgeving
 - Geen ingewikkelde command prompts
 - AVR Studio 4; integreert met GCC compiler
 - Heeft ook emulator voor PC-only debugging
- Programmer: PonyProg (grafisch), of toch avrdude?
 - Avrdude zit in Win-AVR package, maar wel command line
 - Toch voor gekozen, een programma minder te installeren



Mijn eerste programma

- De 'Hello World' voor embedded: knipperLED
 - Simpel C programma
 - Code size 116 bytes incl runtime
 - 'F7' in IDE build executable
 - Issue: instellen CPU klok...
 - Draait in een keer
 - Programmeren → knippert
- Tweede programma was PWM
 - Voorbeeldcode van internet
 - 3-kanaals gebruik makend van timer-based PWM in hardware
 - <http://expand.xs4all.nl/avr/rgb.html>
 - Werkt ook in een keer, zelfs met de powerLEDs
 - Outputs inverterend programmeren ivm driver (sheet 11)

```
int main()
{ DDRD = (1 << PD2);
  while(1)
  { _delay_ms(500);
    PORTD |= (1 << PD2);
    _delay_ms(500);
    PORTD &= ~(1 << PD2);
  }
}
```

Zeer snelle start (enkele dagen)!!

Gevoel voor tijd

- Timers
 - Timers worden gebruikt voor de PWM
 - Ik gebruik de timero overflow nu ook voor een interrupt
 - $4 \text{ MHz} / 8 / 256 = \sim 1.953 \text{ kHz}$
 - Verhoogt teller, toggelt LED
- Interrupts
 - Goed gesupport vanuit C
 - opties voor blocking/re-entrant etc

```
#include <avr/interrupt.h>

ISR(TIMER0_OVF_vect)
{
    pwm_timer++; // volatile!
    PORTD ^= 1 << PD2;
}

...
TIMSK |= (1 << TOIE0);
Sei(); // global enable
...
```

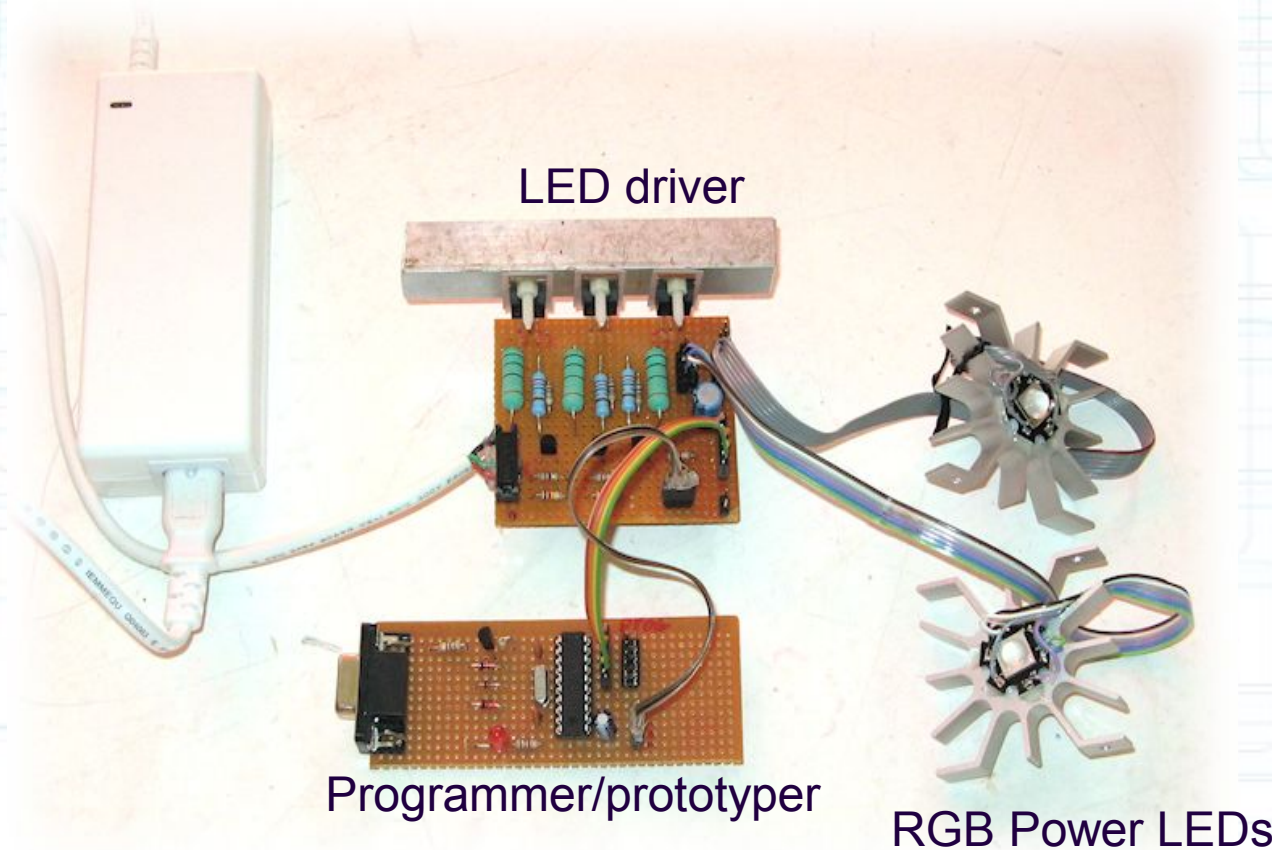
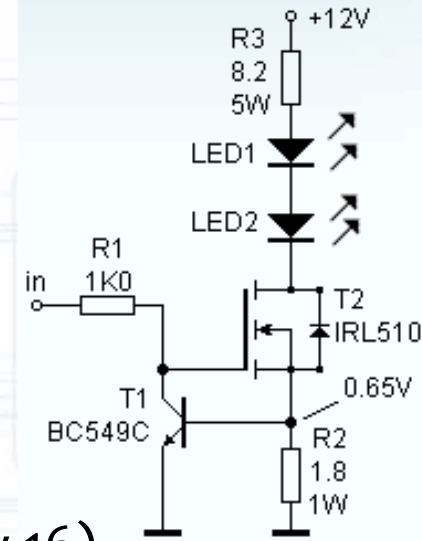
I²C, daar ging het om

- Atmel noemt I²C: TWI, Two-Wire Interface
 - Trade-mark issues voorkomen
- Zoeken om libraries te vinden
 - Vooral voor I²C slave
 - I²C eeprom voorbeeld gevonden
 - Volledig in C, interrupt-driven
 - Map LED/switch op eeprom memory buffer
 - Programma <512 bytes (25%)
 - Alternatief: circular buffers voor send/receive (I²C-dimmer voorbeeld)
 - Message passing protocol; voor high-level commando's?
- Aansluiten op I²C port op NSLU2 → verbinding!!!

```
void main(void)
{
    while(1)
    {
        // switch op #2, LED #3
        eeprom[2] = PIND;
        PORTD = eeprom[3];
    }
}
```


En nu verder

- LED power driver: zie hiernaast
 - Current limiting @ 350 mA/kanaal
- Compiler efficiency
 - Tips&tricks, zoals 8-bit code generatie (ipv 16)



Bedankt voor jullie aandacht
Meer info: www.keesmoerman.nl/attiny.html



Vragen?