Workshop AVR

Een klein stapje naar 8-bits microcontrollers

Door Ad Boerma

Voor wie is dit? Doel?

Voor iedereen die nieuwsgierig is naar AVR microcontrollers.

Specifieker: Voor mensen die al wel een beetje kunnen programmeren maar het ook wel eens met microcontrollers willen doen en dan bij gebrek aan kennis van elektronica afhaken.

Doel: Basiskennis opdoen met enkele basiscomponenten rondom een AVR microcontroller. Beetje schema's kunnen lezen. Allerlei how-to's en artikeltjes beter kunnen begrijpen.

Niet (direct) het doel: Studie Elektrotechniek, advanced class microcontrollers, world domination

Wat gaan we zien

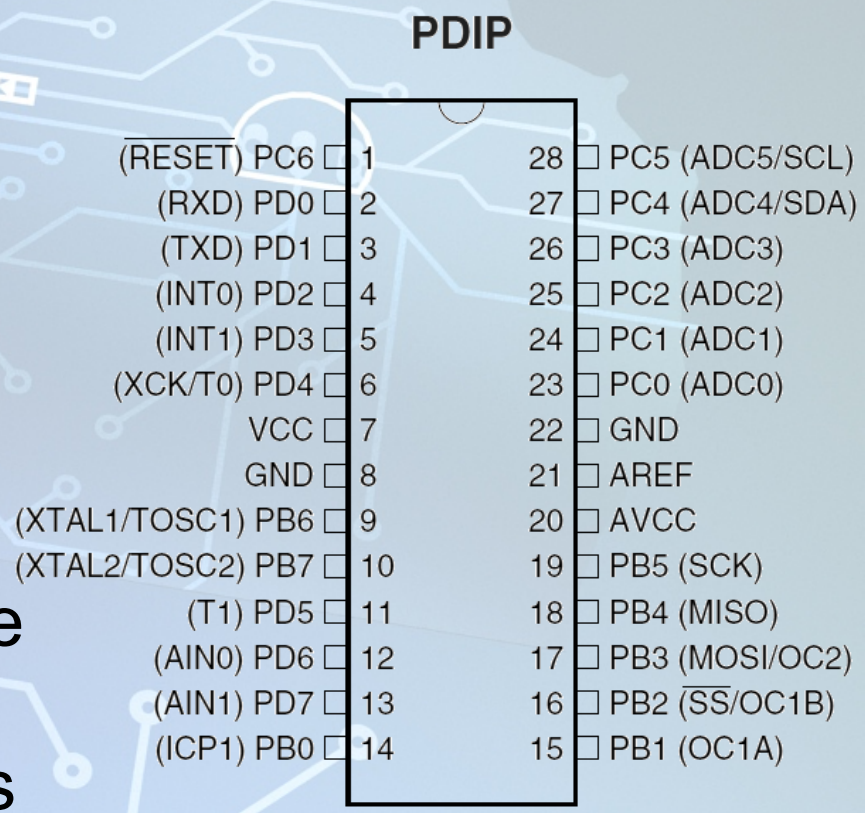
Atmel AVR 8-bits microcontrollers
Atmega8 in detail
Elektronica - Weerstand, Condensator, Led (met voorschakelweerstand, Diode, Transistor, IC
Programmeren (hardware/software)
Eerste voorbeeld: Knipperled
Rekenen met bitjes en bytes

Atmel AVR 8-bits microcontroller

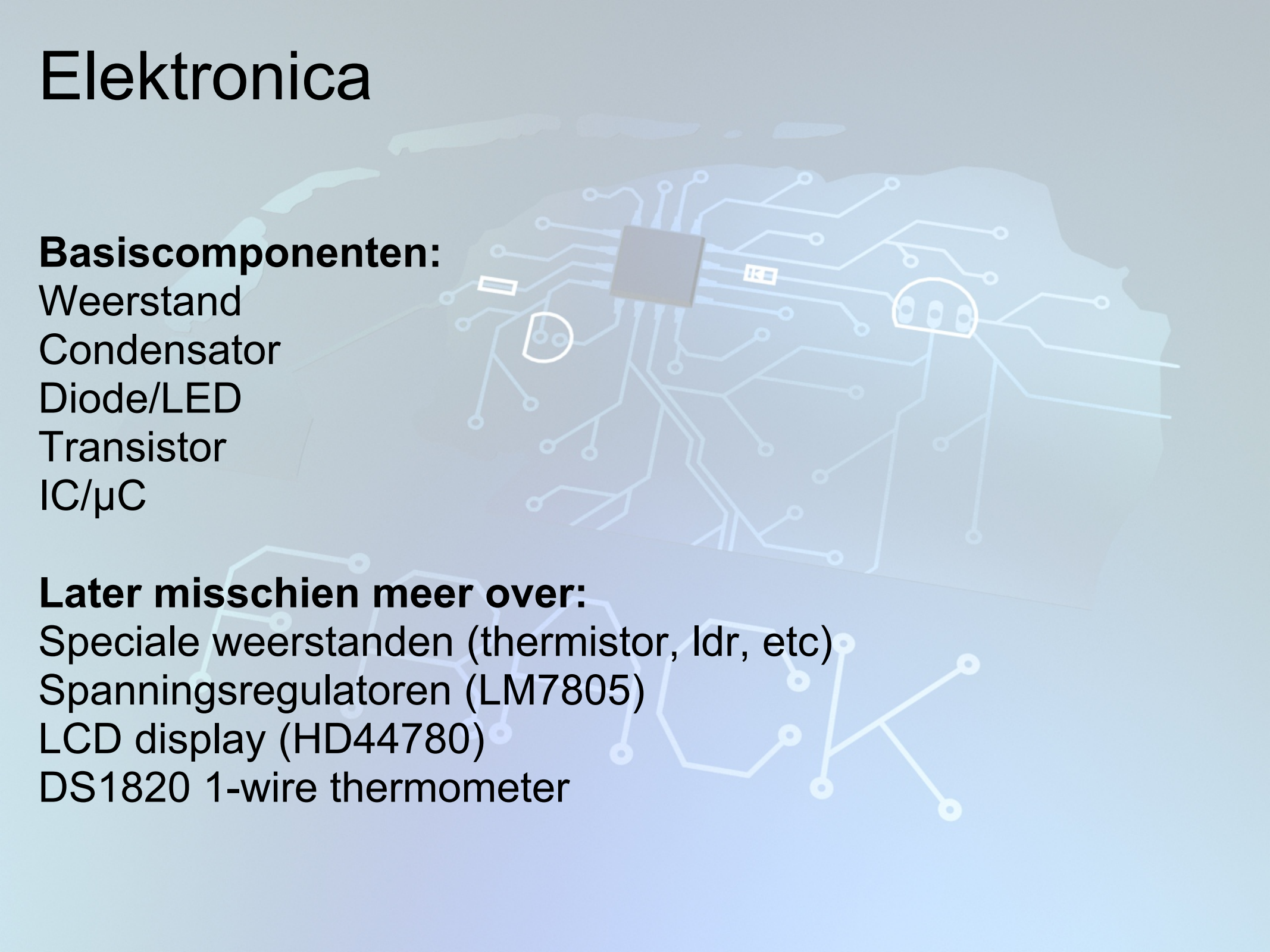
*"De **AVR** is een 8 bit-RISC-microcontroller (µC) ontwikkeld door Atmel in 1996. De AVR was één van de eerste microcontroller-families die gebruik maakte van on-chip Flashgeheugen voor programmaopslag in plaats van PROM, EPROM of EEPROM.*
De AVR is een Modified Harvard-architectuurapparaat waarbij het programma en de data worden opgeslagen in aparte fysieke geheugensystemen."
- Wikipedia

Atmega8

8Kbytes of Flash program memory
512Bytes EEPROM
1Kbyte Internal SRAM
Up to 16MIPS Throughput at 16MHz
Two 8-bit Timer/Counters
One 16-bit Timer/Counter
Three PWM Channels
6-channel 10-bit ADC in PDIP package
Internal Calibrated RC Oscillator
External and Internal Interrupt Sources
23 Programmable I/O Lines



Elektronica



Basiscomponenten:
Weerstand
Condensator
Diode/LED
Transistor
IC/µC

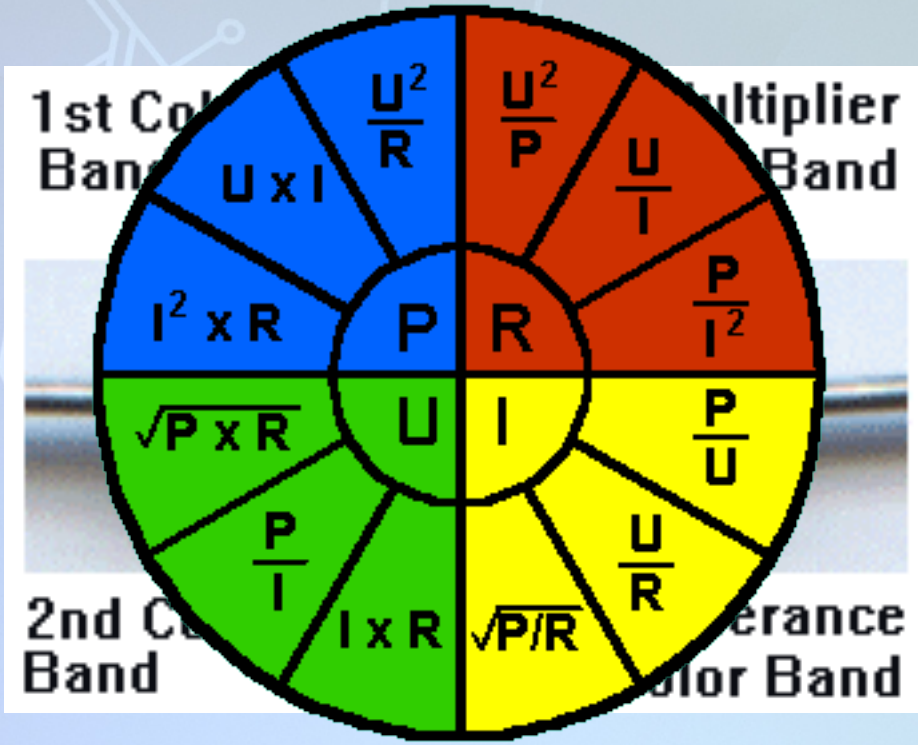
Later misschien meer over:
Speciale weerstanden (thermistor, ldr, etc)
Spanningsregulatoren (LM7805)
LCD display (HD44780)
DS1820 1-wire thermometer

Elektronica: Weerstand

Een weerstand beperkt de doorvoer. Een weerstand heeft een waarde van 1 ohm (R) als een spanning (U) van 1 volt over de component leidt tot een stroom (I) van 1 ampère.

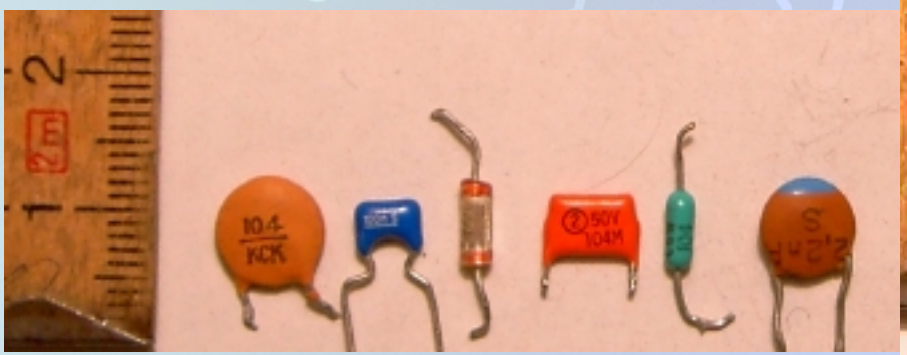
$U = I * R$ $I = U / R$ $R = U / I$

Kleur	1e	2e	Factor	Tol
Zwart	0	0	x1	
Bruin	1	1	x10	1%
Rood	2	2	x100	2%
Oranje	3	3	x1000	3%
Geel	4	4	x10.000	4%
Groen	5	5	x100.000	
Blauw	6	6	x1.000.000	
Paars	7	7	x10.000.000	
Grijs	8	8		
Wit	9	9		
Goud			x0.1	5%
Zilver			x0.01	10%



Elektronica: Condensator

Een condensator slaat elektrische lading op (uitgedrukt in farad). Deze lading kan in een klap worden losgelaten (denk aan het opladen van een flitser van een fototoestel. Maar vaak worden condensatoren gebruikt om verstoringen weg te filteren (voedingslijnen).
De keramische condensator is klein, de elektrolyt-condensator (of elco) is groter maar hierbij moet gelet worden op de polariteit.

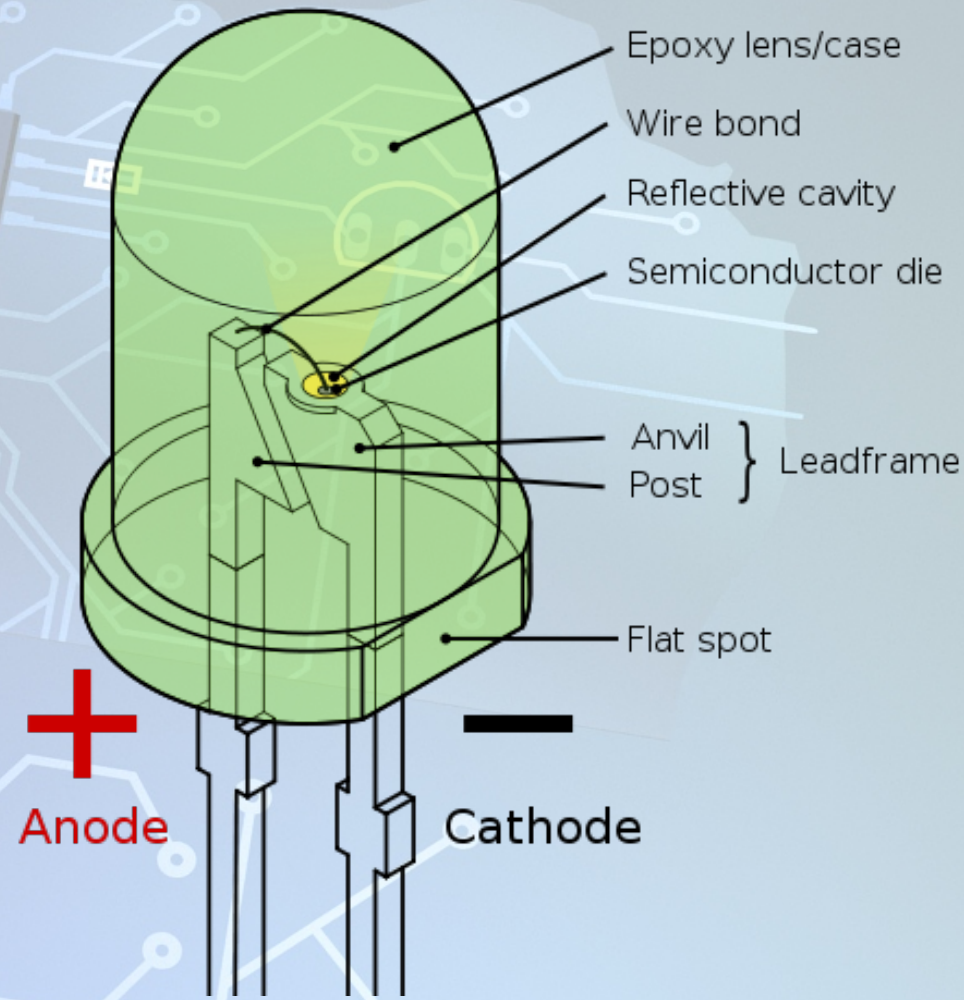


Elektronica: LED

Light Emitting Diode
Elke kleur heeft een ander voltage
Anode is de lange poot
Cathode is de korte poot
Cathode heeft een vlak kantje

Een gewoon ledje heeft altijd een voorschakelweerstand nodig!!
(doe je dit niet kan je de led en je microcontroller stuk maken door een te hoge stroom)

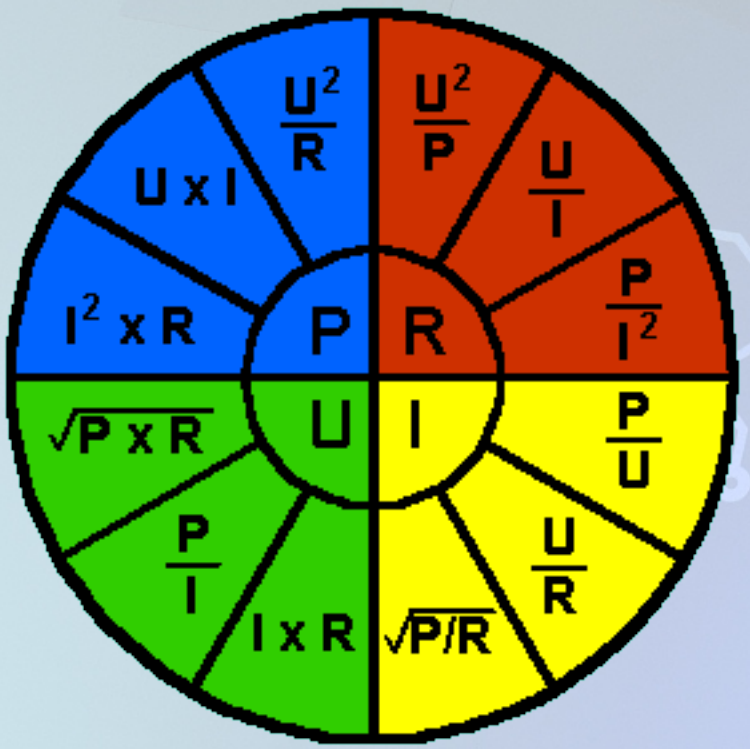
Standaard rood: 1.7V
Standaard groen: 2.2V
Standaard blauw: 4.5V



(afbeelding van Wikipedia)

Elektronica: Voorschakelweerstand

Een LED heeft (meestal) een voorschakelweerstand nodig
Voorschakelweerstand berekenen kan ook makkelijk online
Voor Android is er ElectroDroid
Maar het kan ook met de hand



R = U / I
De *voedingsspanning* (V) minus de *spanningsval* (V) over de led gedeeld door de *stroom* (A) geeft de benodigde *voorschakelweerstand* (Ω)

(Voeding(V) - LED(V)) / LED(A) = R
(5V - 2V) / 0.020A = 150Ω
(5V - 1.7V) / 0.015A = 220Ω

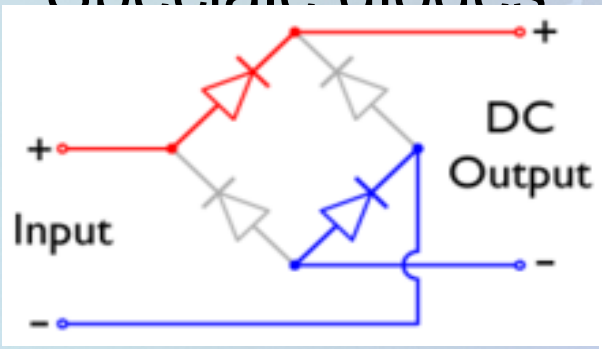
E12 bevat oa: 150Ω, 180Ω, 220Ω, 270Ω

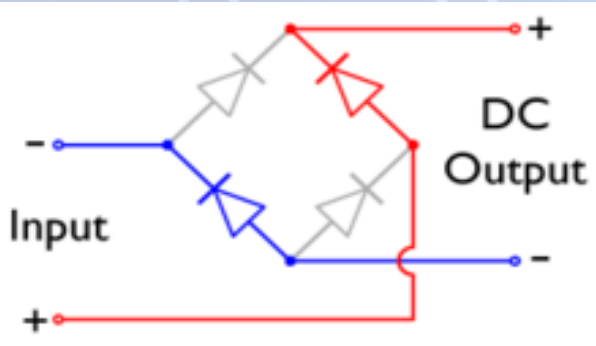
Elektronica: Diode

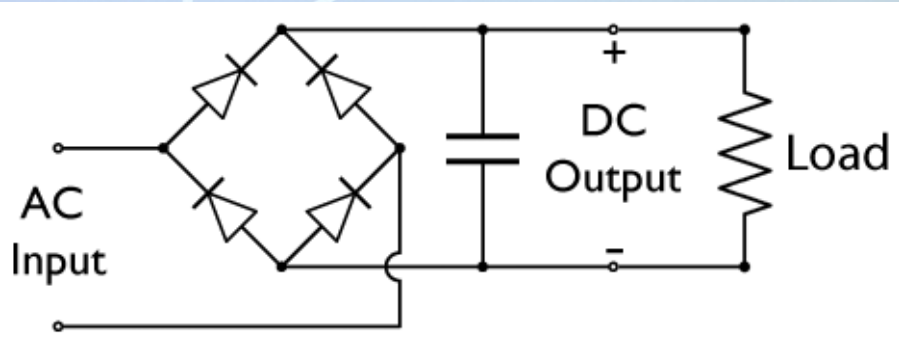
Een diode is een een-richtings-versperring. Zo kan een diode gebruikt worden tegen het verkeerdom aansluiten van een stroombron, maar hou er rekening mee dat ze wel een beetje van je voltage afsnoepen.

Een bekende toepassing van diodes is een gelijkrichter. Hiermee kan wisselspanning omgezet worden naar gelijkspanning.

Speciale diodes:





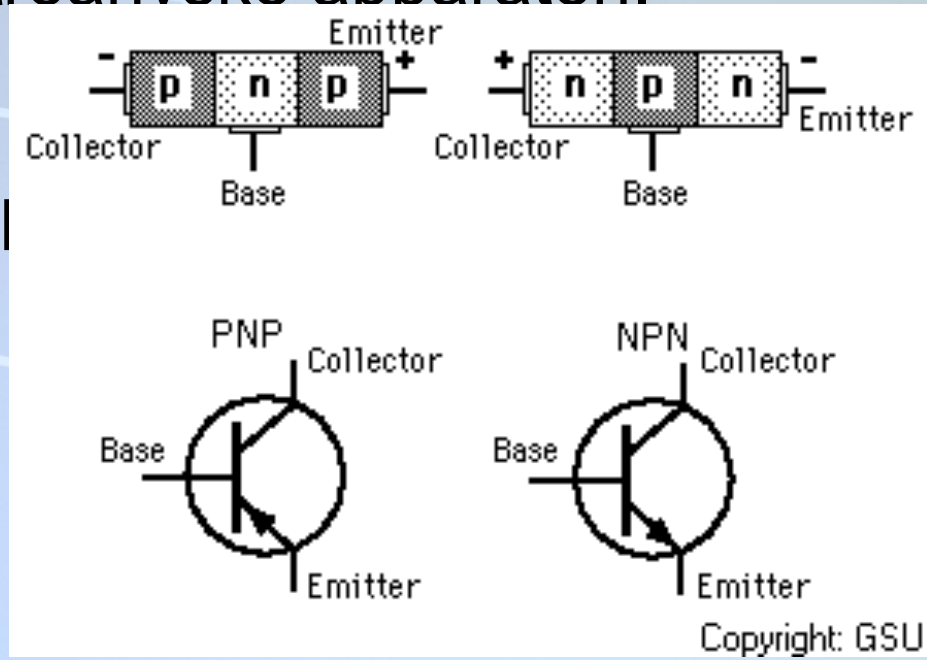


Elektronica: Transistor

Een transistor kan gezien worden als een elektrische aan/uit-schakelaar. Deze kan ook gebruikt worden als versterker.

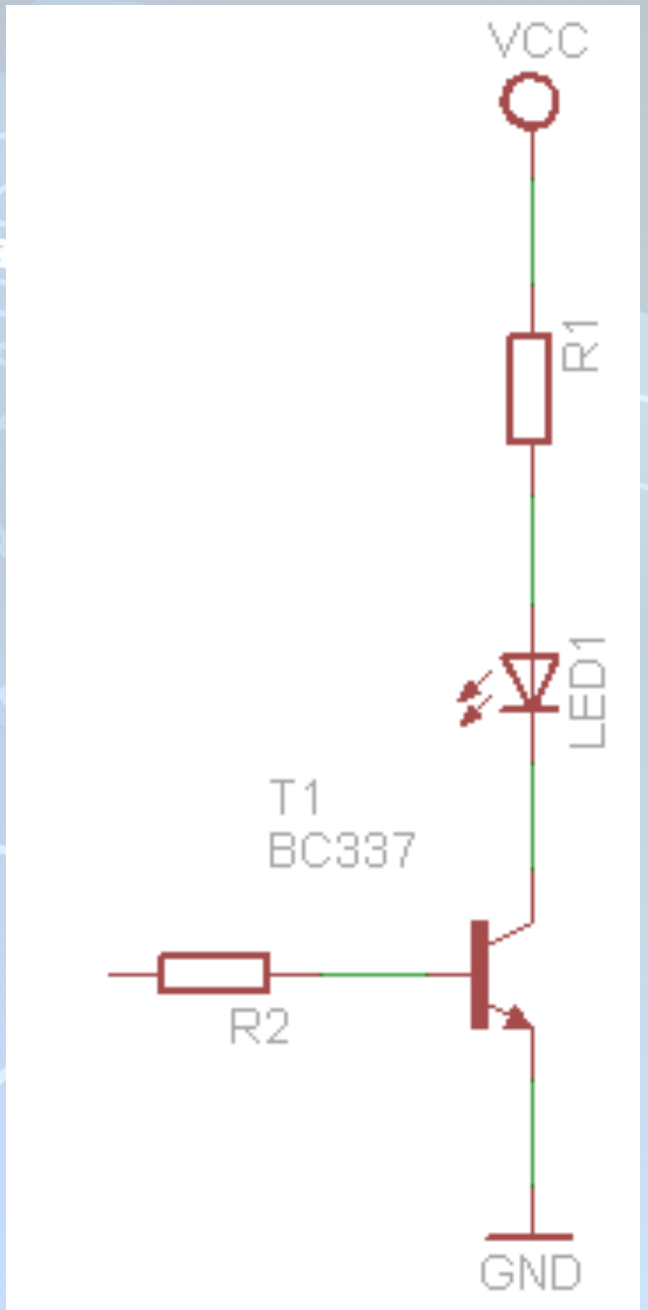
"De transistor is de wichtichste aktive healgelieder binnen de elektroanika. Hy fersterket of skeakelt benammen elektroanyske sinjalen. De transistor is de fûnemintele boustien fan kompjûters en in soad oare elektroanyske apparaten."
- Aldus de Fryske Wikipedia

Bipolaire transistor in twee smaken: I



Elektronica: Transistor

Dit is een voorbeeld van een NPN-transistor. Op deze manier kan er met een klein stroompje van de microcontroller een veel zwaardere last worden geschakeld dan door de pootjes van de µC mag.



Elektronica: IC/µC

De afkorting IC staat voor Integrated Circuit

Dit kunnen de meest uiteenlopende soorten circuits zijn, van simpel tot zeer complex. Ook het uiterlijk varieert van een klein zwarte knobbeltje in de zakjapanner tot iets complex als de CPU van een moderne PC.

Voor de hobby is de (P)DIP de meest gebruikte 'package' Deze (Plastic) Dual Inline Package is de bekende chip met pootjes aan beide zijden.

Programmeren

Het programmeren van een µC wordt meestal in assembler of C gedaan, al bestaan ook Basic-varianten. Door gebruik te maken van C, met een speciale versie van gcc is het mogelijk om met vrijwel elk (desktop)platform code te compileren voor de AVR µC's. Ook in de keuze van IDE/editor is men erg vrij.

Voor Windows™ is er de veel gebruikte AVR Studio 4 van Atmel zelf, met WinAVR als C-compiler.

Onder Linux is er gedit/vim/Emacs/geany/codeblocks of zelfs Eclipse met een AVR-plugin. Het compilen kan daarna met avr-libc, binutils-avr en gcc-avr.

Programmeren

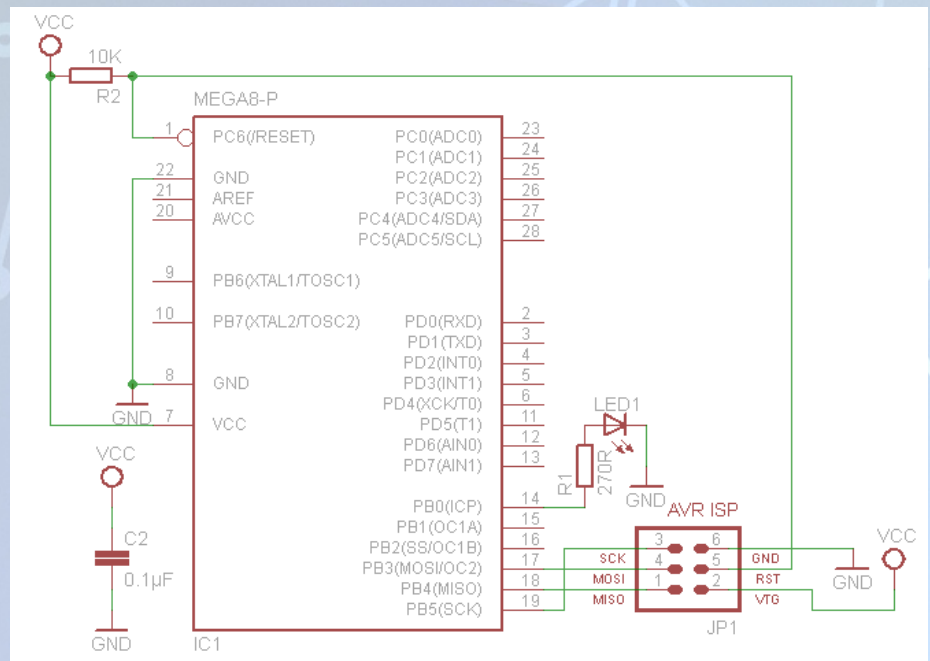
Om uiteindelijk de gecompilde binaire (hex)file in de µC te krijgen is er een stukje hardware nodig om de boel aan elkaar te knopen. Dat kan het makkelijkst met een In System Programmer (ISP) en een stukje software zoals AVRdude.

Deze ISP's zijn in diverse smaakjes te krijgen, met USB, seriële en parallelle verbindingen. De USB-versie kan vaak ook meteen als 5V-voeding misbruikt worden.
De Arduino's gebruiken een bootloader waardoor het programmeren op een nog hoger niveau kan plaatsvinden. De µC bevat een soort mini-OS-je die de rest van de µC kan volschrijven met programma. Hierbij gebruik je andere ontwikkelpakketten.

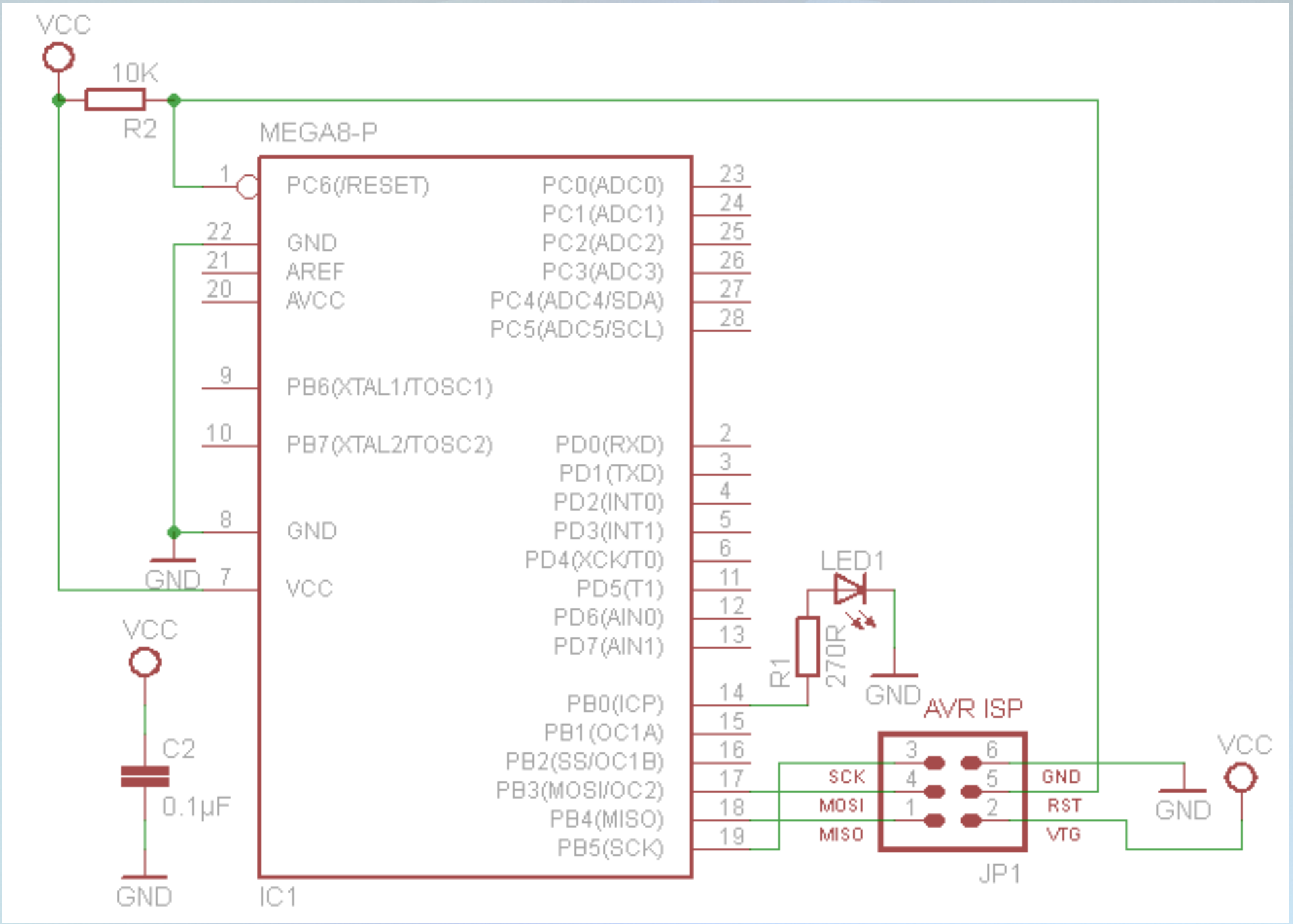
Eerste voorbeeld: Knipperled

Het eerste voorbeeld is een led die we gaan laten knipperen op een vaste interval. Dit is eigenlijk het equivalent van 'Hello World!' voor microcontrollers.

Hardware:
1x µC, 1x weerstand (220-330 Ohm), 1x LEDje, paar draadjes



Eerste voorbeeld: Knipperled



Eerste voorbeeld: Knipperled

```
#define F_CPU 1000000UL  // 1 MHz int osc

#include <util/delay.h>  // lib voor niks doen
#include <avr/io.h>      // registers -> namen
int main (void) {        // Beginpunt
  DDRB = 0x01;           // 0=ingang / 1=uitgang
  while (1) {            // oneindige lus
    PORTB |= 0x01;       // PORTB OR 0b00000001
    _delay_ms(500);
    PORTB &= ~0x01;      // PORTB AND 0b11111110
    _delay_ms(500);
  }
}
```

Eerste voorbeeld: Knipperled

```
Binair tellen: 128  64  32  16  8  4  2  1

0b00000001 = 1
0b00000010 = 2
0b00000011 = 3
0b00000100 = 4
0b00000101 = 5
0b00000110 = 6
0b00000111 = 7
0b00001000 = 8
0b10110100 = 180 = 0xB4

Hexadecimaal: 0 1 2 3 4 5 6 7 8 9 A B C D E F
```

Eerste voorbeeld: Knipperled

```
vier = 0b00000100;
vier = 4;
vier = 0x04;
vier = _BV(PB2);
vier = (1<<2);
```

```
0b00110101          0b00110101
0b00011000 OR       0b00011000 AND
-----------         -----------
0b00111101          0b00010000
```

Eerste voorbeeld: Knipperled

LEDje aan:

```
PORTB |= 0x01;     // PORTB OR 0b00000001
```

LEDje uit:

```
PORTB &= ~0x01;     // PORTB AND 0b11111110
```

```
0b00110100          0b00110101
0b00000001 OR       0b11111110 AND
-----------         -----------
0b00110101          0b00110100
```

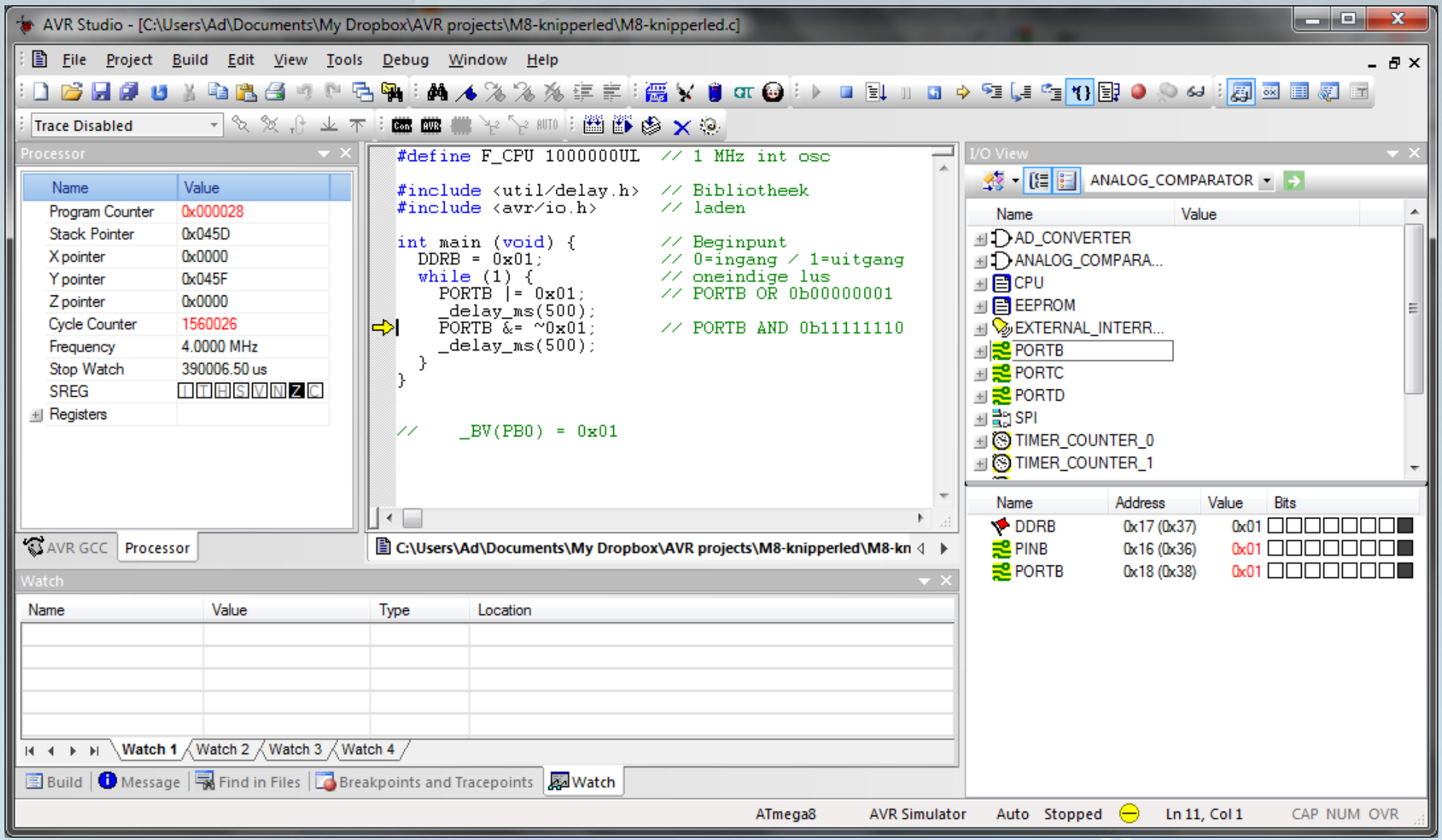
Eerste voorbeeld: Knipperled

```
#define F_CPU 1000000UL  // 1 MHz

#include <util/delay.h>  // lib voor niks doen
#include <avr/io.h>      // registers -> namen
int main (void) {        // Beginpunt
  DDRB = 0x01;           // 0=ingang / 1=uitgang
  while (1) {            // oneindige lus
    PORTB |= 0x01;       // PORTB OR 0b00000001
    _delay_ms(500);
    PORTB &= ~0x01;      // PORTB AND 0b11111110
    _delay_ms(500);
  }
}
```

Eerste voorbeeld: Knipperled

```
Build started 24.6.2011 at 23:50:13
avr-gcc  -mmcu=atmega8 -Wall -gdwarf-2 -Os -std=gnu99 -funsigned-char -funsigned-bitfields -fpack-
struct -fshort-enums -MD -MP -MT M8-knipperled.o -MF dep/M8-knipperled.o.d  -c  ../M8-knipperled.c
avr-gcc -mmcu=atmega8 -Wl,-Map=M8-knipperled.map M8-knipperled.o     -o M8-knipperled.elf
avr-objcopy -O ihex -R .eeprom -R .fuse -R .lock -R .signature  M8-knipperled.elf M8-knipperled.hex
avr-objcopy -j .eeprom --set-section-flags=.eeprom="alloc,load" --change-section-lma .eeprom=0 --no-
change-warnings -O ihex M8-knipperled.elf M8-knipperled.eep || exit 0
avr-objdump -h -S M8-knipperled.elf > M8-knipperled.lss

AVR Memory Usage
----------------
Device: atmega8

Program:     102 bytes (1.2% Full)
(.text + .data + .bootloader)

Data:          0 bytes (0.0% Full)
(.data + .bss + .noinit)


Build succeeded with 0 Warnings...
```

Eerste voorbeeld: Knipperled



Eerste voorbeeld: Knipperled

```
C:\...\M8-knipperled\default>avrdude -p m8 -c usbasp -U flash:w:M8-knipperled.hex

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.04s

avrdude: Device signature = 0x1e9307
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed
         To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: reading input file "M8-knipperled.hex"
avrdude: input file M8-knipperled.hex auto detected as Intel Hex
avrdude: writing flash (102 bytes):

Writing | ################################################## | 100% 0.85s


avrdude: 102 bytes of flash written
avrdude: verifying flash memory against M8-knipperled.hex:
avrdude: load data flash data from input file M8-knipperled.hex:
avrdude: input file M8-knipperled.hex auto detected as Intel Hex
avrdude: input file M8-knipperled.hex contains 102 bytes
avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 0.53s


avrdude: verifying ...
avrdude: 102 bytes of flash verified

avrdude: safemode: Fuses OK

avrdude done.  Thank you.
```

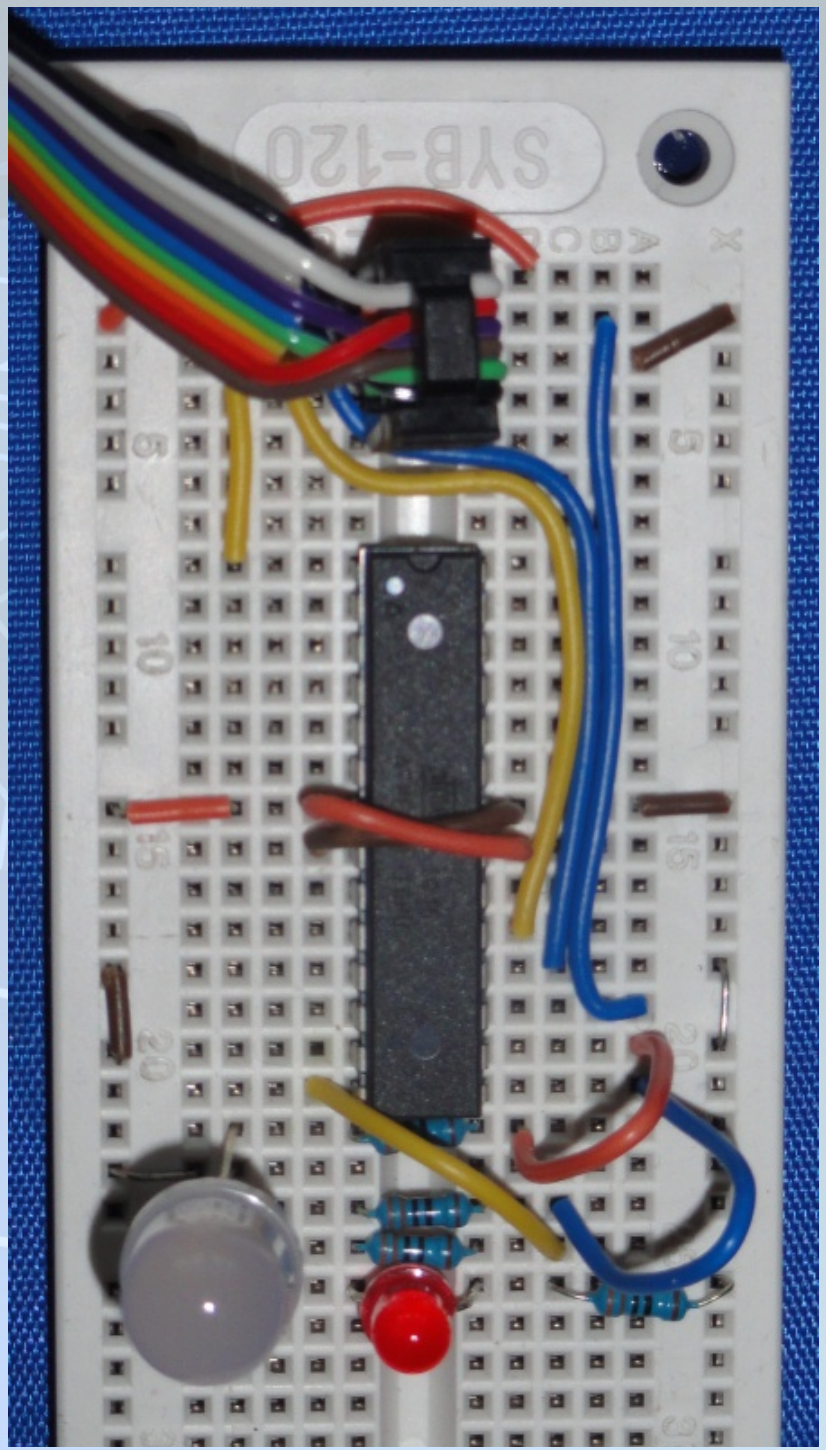
Here's one I prepared earlier

Programmeerpoort

Atmel atmega 8

Voeding

RGB-led met RGB aangesloten op 3 weerstanden
Rode led direct tussen VCC en GND



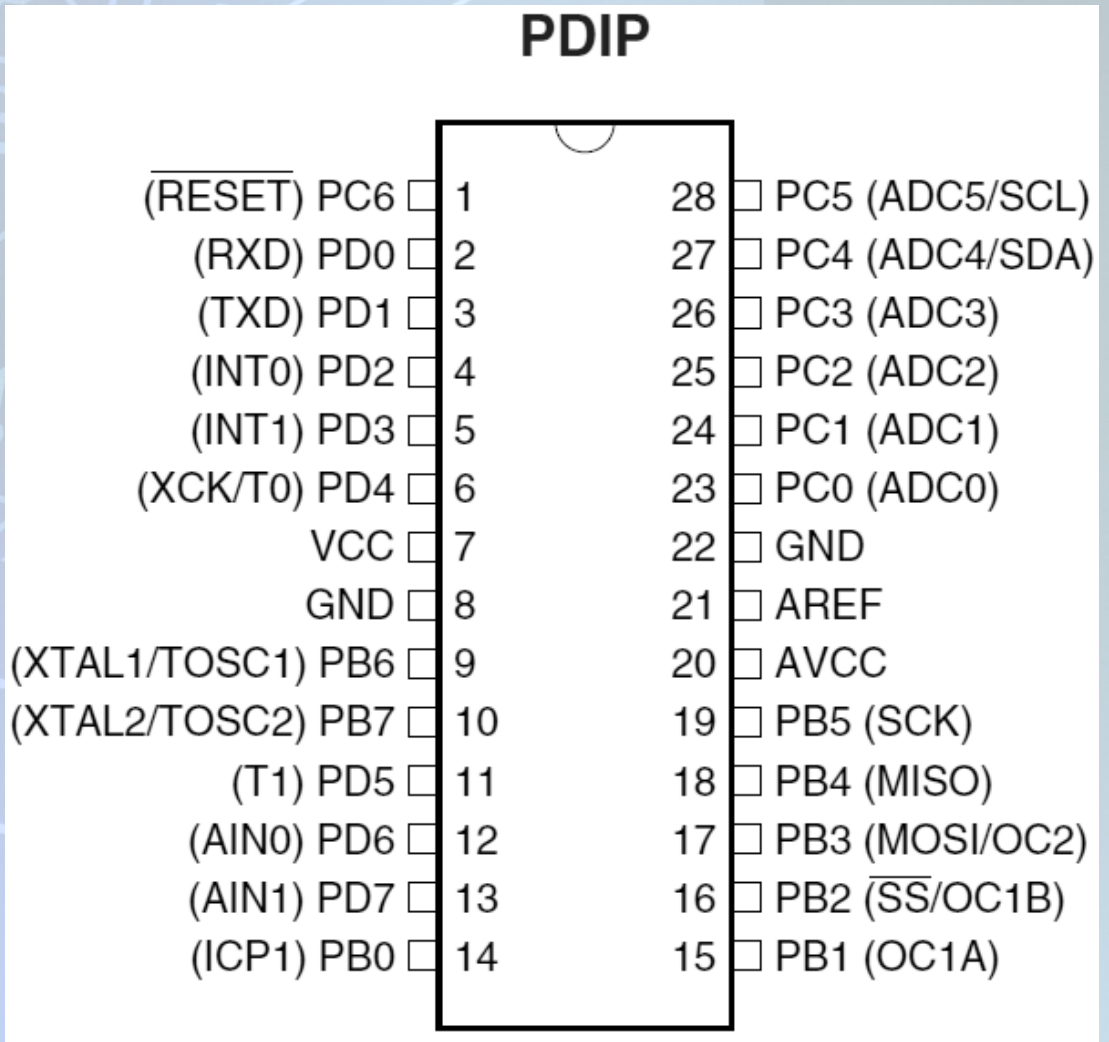
Pitfalls en aandachtspunten

Je Atmel-etje vind 5V genoeg en kan slechts 40mA per poot verdragen (300mA in totaal over Vcc en GND)
Condensatortjes om te ontstoren zijn handig, zelfs noodzakelijk bij analoge metingen (met smoorspoeltje erbij)
Fuses: je 'bios'-settings, verkloot je die kan je er niet meer bij
Probeer niet je breadboard én via USB én los te voeden
Weerstanden beperken de stroom, bij LED's, bij transistoren, bij bijna alles
Datasheets, datasheets, datasheets, Google en datasheets

Volgende keren...

Lees eens een datasheet

Mogelijke onderwerpen:
Mooi maken van breadboard
Inputs gebruiken
RGB-ledje aansturen
Analoge ingang gebruiken
LCD display
8-24V -> 5V voeding
Communicatie 1-wire
DS1820
The end...




FRACK