Microcontrollers

Les 1

Inhoud cursus

Wat zullen we leren in deze cursus?

- programmeren in C op het Arduino platform
- binaire getallen
- seriële communicatie
- digitale input en output
- functies en bibliotheken
- interrupts
- timers

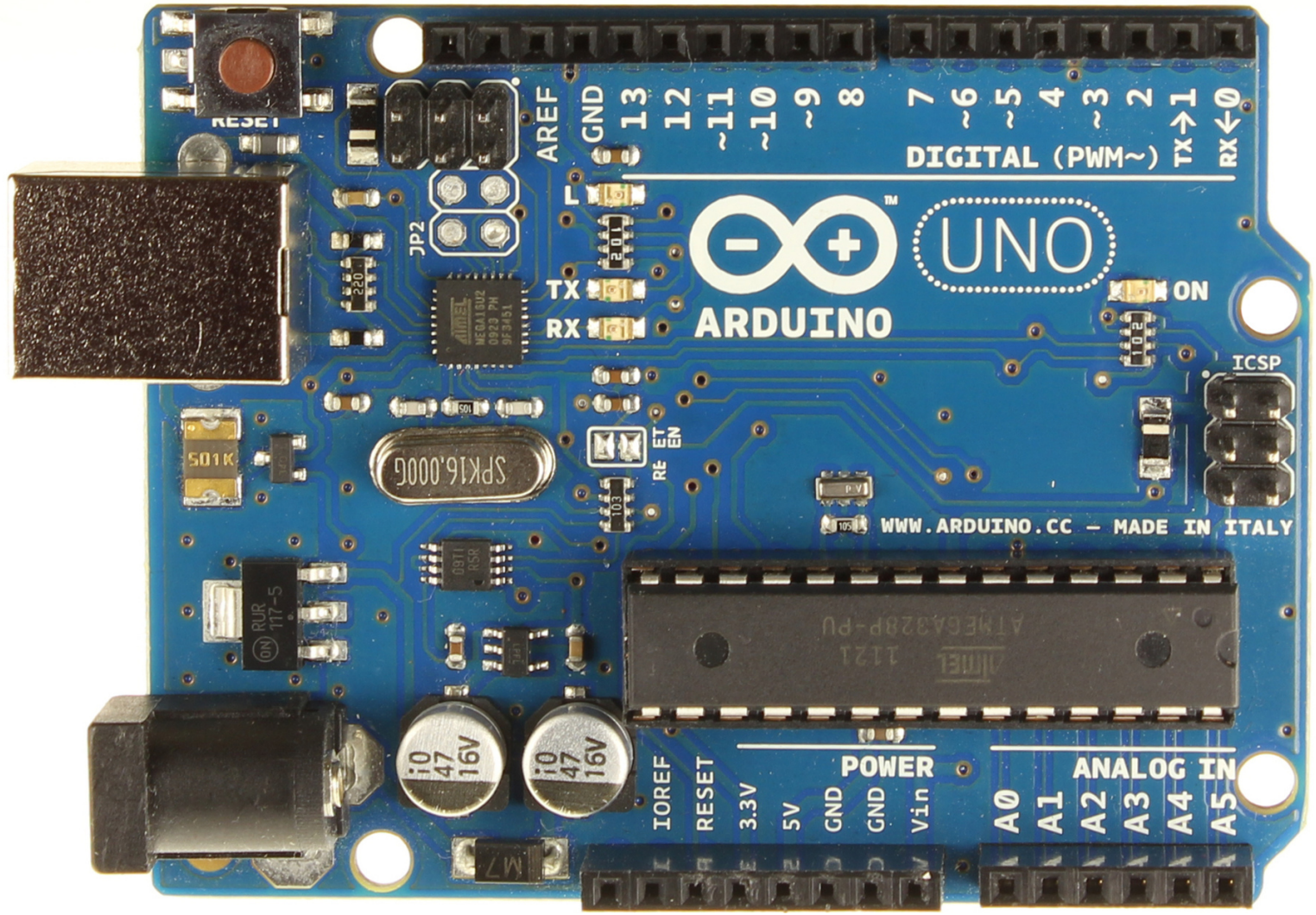
Evaluatie theorie

- schriftelijk examen op einde semester
- theorie + korte oefeningen
- gesloten boek

Evaluatie labo

- bolletjessysteem
- na iedere oefening controle door lesgever
- pas starten met volgende oefening als vorige OK
- tweemaal individueel labo (week 6 en 12)

Arduino

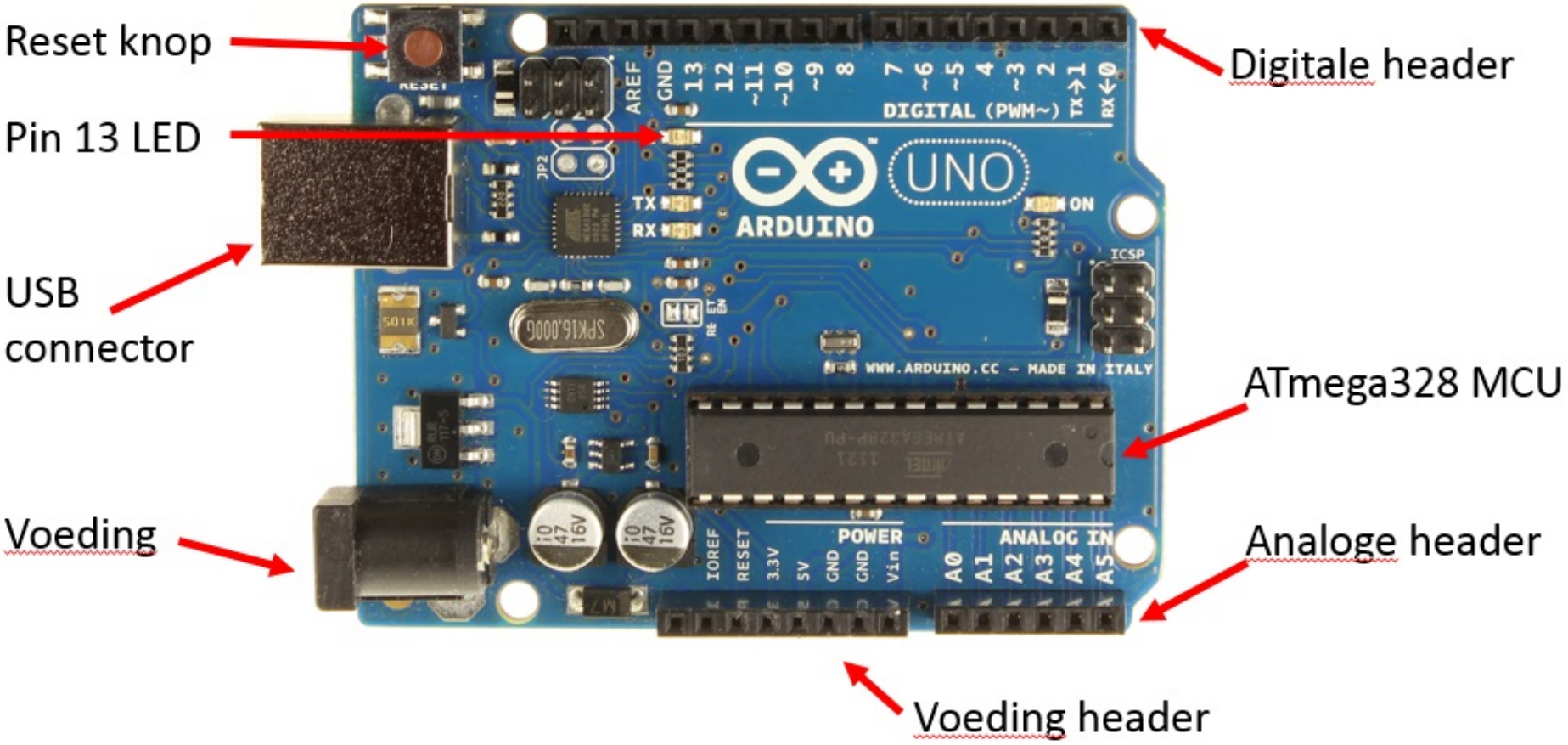

RESET
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
JP2
TX
RX
ARDUINO
UNO
ON
ICSP
WWW.ARDUINO.CC – MADE IN ITALY
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

Arduino Uno

Bordje beschikt o.a. over:

- eenvoudig microcontroller-bordje
- gebaseerd op Atmega328
- USB ondersteuning door extra IC
- bevat zelf weinig I/O
- heel veel shields ter beschikking

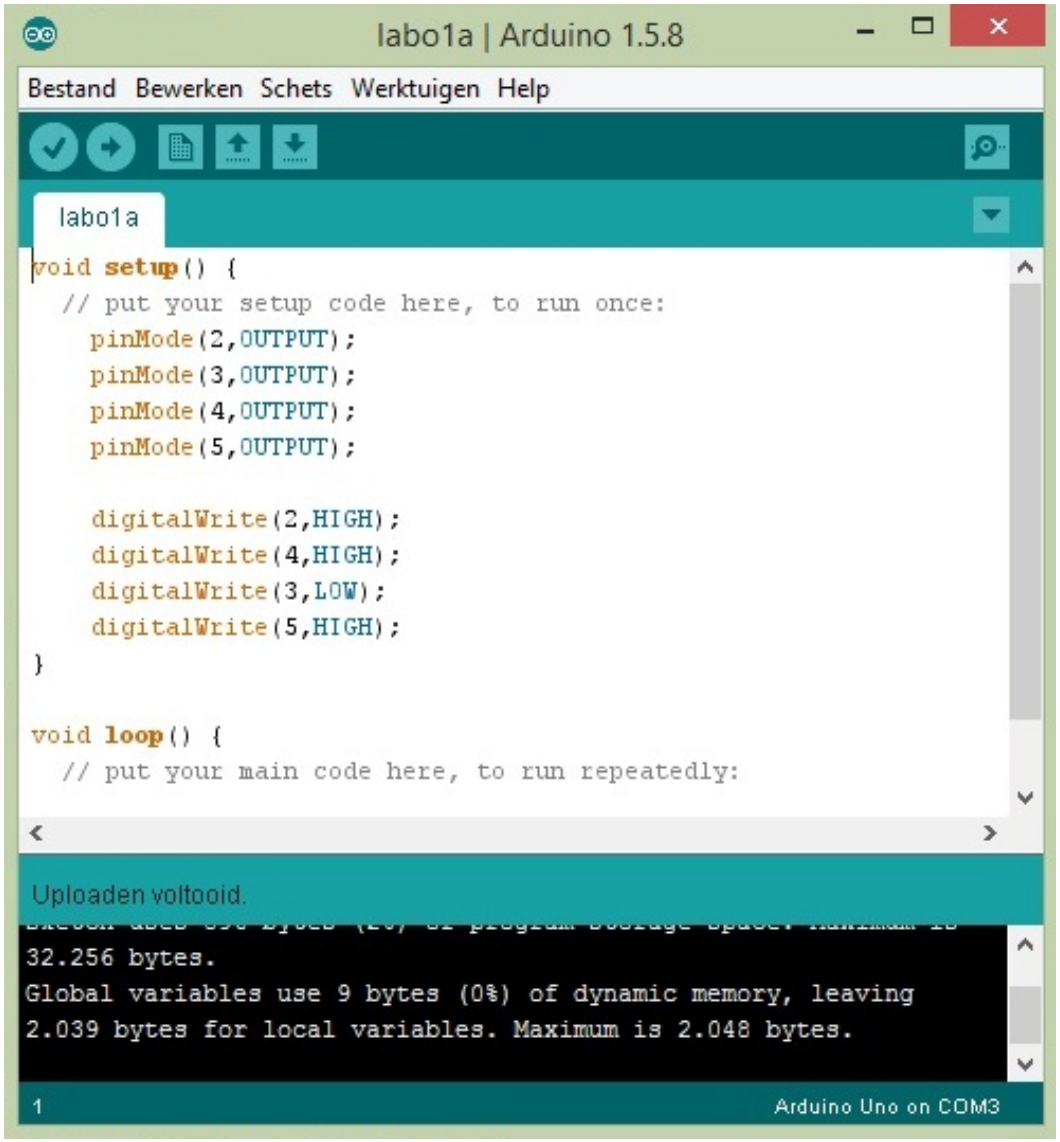
Arduino


Reset knop
Pin 13 LED
USB connector
Voeding
Digitale header
ATmega328 MCU
Analoge header
Voeding header
UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN
WWW.ARDUINO.CC – MADE IN ITALY

Werking

- bordje eenvoudig te programmeren via USB
- kan ook gevoed worden via USB (of extern)
- speciale IDE voor Arduino
- geen extra hardware (programmer) of software nodig

Arduino IDE


labo1a | Arduino 1.5.8
Bestand Bewerken Schets Werktuigen Help
labo1a
void setup() {
// put your setup code here, to run once:
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
digitalWrite(2,HIGH);
digitalWrite(4,HIGH);
digitalWrite(3,LOW);
digitalWrite(5,HIGH);
}
void loop() {
// put your main code here, to run repeatedly:
Uploaden voltooid.
32.256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving
2.039 bytes for local variables. Maximum is 2.048 bytes.
1
Arduino Uno on COM3

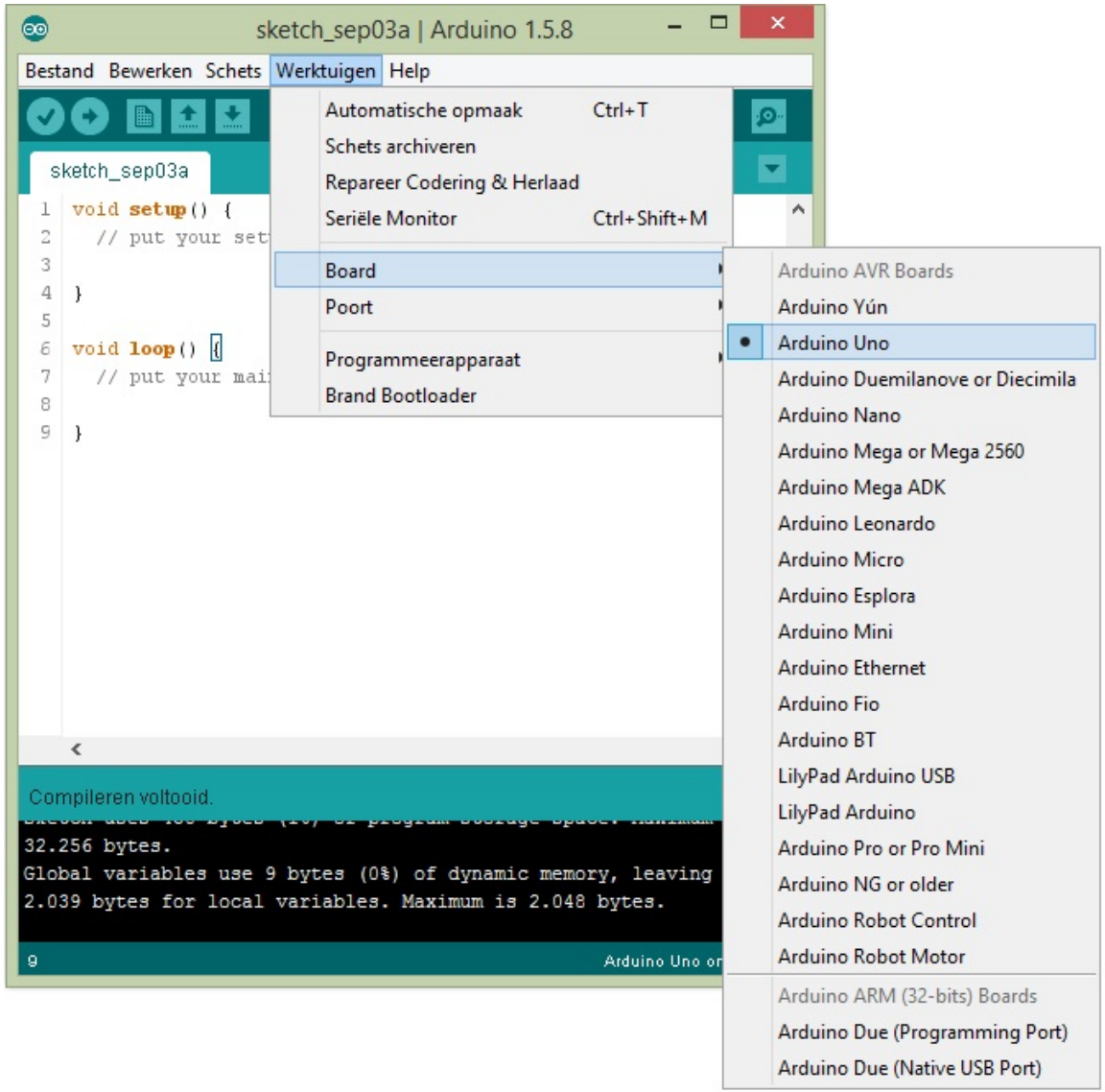
Arduino IDE

- bevat alles om een Arduino te programmeren
- gratis te downloaden vanaf website Arduino
- heel veel codevoorbeelden
- code op bordje uploaden gebeurt vanuit IDE
- geen debugger, kan deels vervangen worden door seriële monitor

Installatie

- hardware: bordje + USB-kabel
- software: IDE downloaden
- bordje aansluiten
- driver installeren (virtuele seriële poort)
- start IDE
- bordje + seriële poort instellen

Instelling bordje



Open source

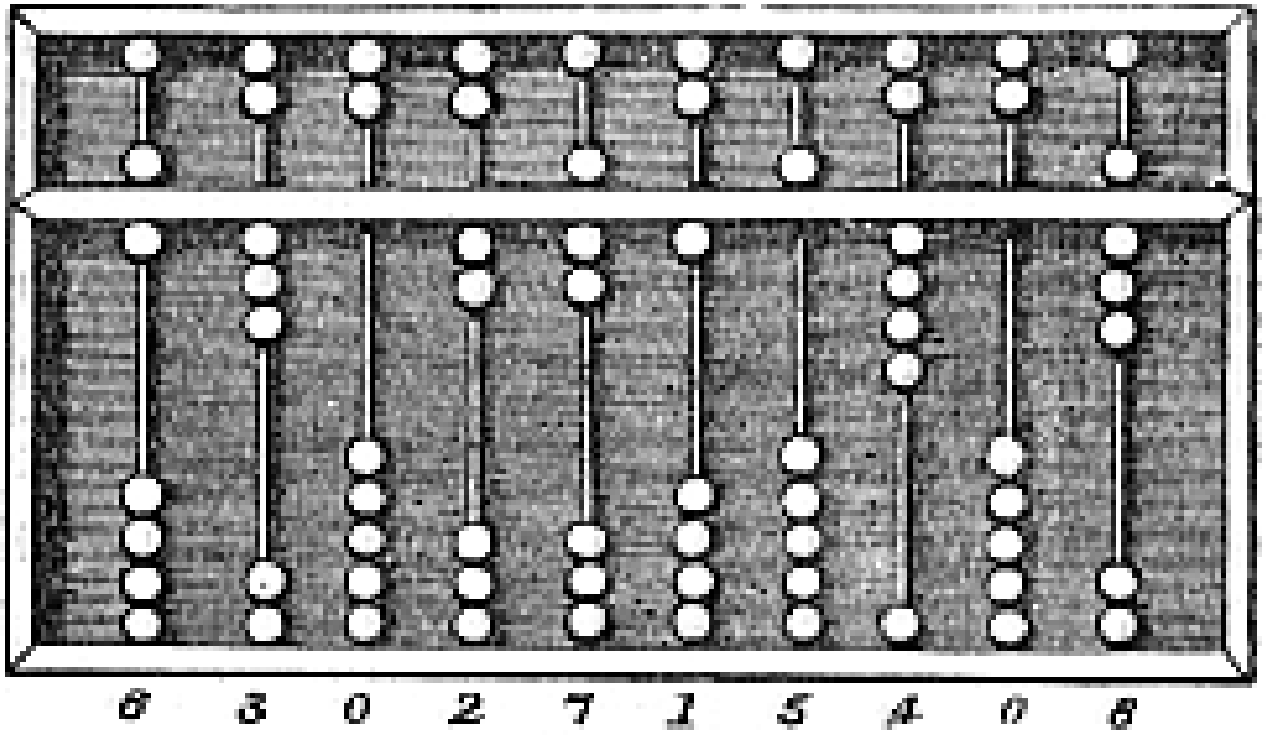
- Arduino is open source hardware
- iedereen mag dit dus namaken
- gevolg: heel populair
- heel veel klonen van andere merken

Geschiedenis

- het mechanisch tijdperk
- het elektrisch tijdperk
- belangrijke uitvindingen
- het microprocessor tijdperk

Het mechanisch tijdperk

De abacus = telraam om snelle en logische berekeningen te maken



Charles Babbage

- principe van moderne computer wordt aan hem toegeschreven
- voorheen: decimale stelsel (van 0 tot 9)
- Babbage maakte gebruik van binair stelsel (0 en 1)
- maakte concept van eerste programmeerbare (mechanische) reken-machine in 1849
- techniek stond nog niet ver genoeg om te bouwen

Het elektrisch tijdperk

- Zuse bouwde eerste programmeerbare rekenmachine:
  - in 1938 de mechanische rekenmachine Z1
  - in 1941 de elektromagnetische computer Z3
- de colossus (1943)
  - eerste elektronische computer met 1500 elektronenbuizen

Belangrijke uitvindingen

transistor : uitgevonden in 1948

- veel kleiner in warmteproductie en afmetingen dan een elektronenbuis

geïntegreerd circuit (IC) : meerdere transistors in één chip

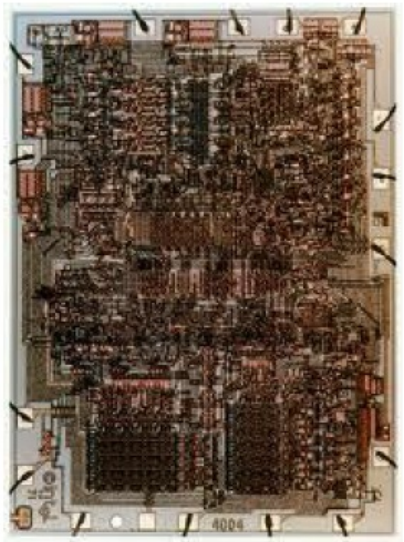
- het werd mogelijk om een complete processor op één chip te integreren
- computers werden veel goedkoper

Het microprocessor tijdperk

- Intel werd opgericht in 1968
- ontwikkelde de 4004 (4-bits) en 8008 (8-bits) CPU voor een calculator
- eerste uit één chip bestaande CPU's
- enorm veel (onverwachte) belangstelling
- de 8080 werd ontwikkeld als general-purpose CPU

Systemen worden complexer

Evolutie processoren



Intel 4004 - 1971
2300 transistors
10 µm technologie
12 mm²

Intel Six Core i7 - 2011
2,270,000,00 transistors
32 nm technologie
434 mm²

Idee complexiteit

De 8086: 1978 , 29.000 transistors

Idee complexiteit

De Pentium: 1993 , 3.300.000 transistors

Idee complexiteit

De Pentium 4: 2000 , 55.000.000 transistors

Idee complexiteit

De Core i7: 2011 , 1.200.000.000 transistors

MIPS

- snelheid kan aangeduid worden met klokfrequentie
- niet elke processor heeft evenveel klokpulsen nodig per instructie
- betere maatstaf is de MIPS-eenheid
- MIPS staat voor **miljoen instructies per seconde**
- MIPS = kloksnelheid in MHz / gem. aantal klokpulsen per instructie
- een 80286 haalde aan 6Mhz een MIPS van 1
- de Intel Core i7 Extreme Edition heeft een MIPS van 147600
- tegenwoordig is ook het aantal MIPS per Watt belangrijk

Oefeningen MIPS

Bereken het aantal MIPS van de volgende processoren:

- microprocessor A: 1GHz - 100 klokpulsen/instructie
- microprocessor B: 2Ghz - 250 klokpulsen/instructie
- microprocessor C: 3GHz - 500 klokpulsen/instructie

Basiszaken Arduino

- poorten
- C-taal
- datatypes
- eerste programma

Digitaal: poort richting

- input: om informatie in te lezen
- output: om iets aan te sturen
- wordt ingesteld met `pinMode()`
- één pin per keer instellen
- input: met of zonder pull-up weerstand

Voorbeelden

```
1  pinmode(13, OUTPUT);          // uitgang
2  pinmode(2,INPUT);             // ingang (laag)
3  pinmode(3,INPUT_PULLUP);      // ingang (hoog)
4
5  digitalWrite(13, HIGH);       // schrijven
6  if (digitalRead(2) == LOW) ... // lezen
```

C-taal

- universele taal
- wordt op bijna elke processor ondersteund
- Arduino gebruikt *vereenvoudigde* versie
- veel zit in bibliotheken, bv. `digitalWrite()`
- ondersteunt geen objecten

Datatypes

Voornaamste datatypes:

boolean : true of false

byte : 0..255

int : -32768..32767

word : 0..65535

long : -2 miljard .. 2 miljard

float : kommagetallen

Cross-platform datatypes

Deze datatypes zijn op elk platform even groot:

uint8_t : unsigned integer 8-bits

int8_t : signed integer 8-bits

uint16_t : unsigned integer 16-bits

int16_t : signed integer 16-bits

uint32_t : unsigned integer 32-bits

int32_t : signed integer 32-bits

Eerste programma: blink

```
1  // wordt eenmalig uitgevoerd
2  void setup() {
3    pinMode(13, OUTPUT);  // pin 13 output
4  }
5
6  // wordt oneindig herhaald
7  void loop() {
8    digitalWrite(13, HIGH);   // LED aan
9    delay(1000);              // wacht 1s
10   digitalWrite(13, LOW);    // LED uit
11   delay(1000);              // wacht 1s
12 }
```

`setup()` **en** `loop()`

`setup()`

- wordt éénmalig uitgevoerd in het begin
- ideaal voor instellingen
- taken die éénmalig moet gebeuren

`loop()`

- wordt na `setup()` uitgevoerd
- wordt onderbroken door interrupts
- wordt herhaald tot reset

Werking

Volgende code is niet zichtbaar in de IDE:

```
1  int main()
2  {
3     setup();
4     while(true)
5     {
6        loop();
7        serialEventRun();
8     }
9  return 0;
10 }
```

Datavoorstellingen

- de bit
- talstelsels
- dataorganisatie
- karaktercodes

De bit

- informatie in computer wordt voorgesteld als patronen van bits
- een bit kan de waarde 0 of 1 aannemen
- daarbij is 0 $= false$ en 1 $= true$
- elektronica is ideaal om met aan/uit toestand te werken
- 0V $=$ 0 en 5V $=$ 1 (tegenwoordig is deze spanning lager)

Het decimale talstelsel

- het decimale talstelsel wordt veel gebruikt
- tien cijfers van **[0..9]**
- bv. het getal 156 kan iedereen ontleden:
- $1.10^2 + 5.10^1 + 6.10^0 = 100 + 50 + 6$
- positie komt overeen met de plaats t.o.v. decimale punt
- om het decimale talstelsel aan te tonen schrijft men ook: 156_d

Oefening

Ontleed telkens de volgende decimale getallen:

- 4785
- 86,7
- 0,3654

Het binaire talstelsel

- nummersysteem ontwikkeld om met binaire logica te werken
- principe hetzelfde als decimaal maar slechts twee cijfers **[0..1]**
- het getal 101_b komt overeen met:
- $1.2^2 + 0.2^1 + 1.2^0 = 4 + 0 + 1 = 5_d$

Hoofdgetallen

Enkele veelvouden van twee:

$$\begin{aligned} 100000_b &= 32_d &&= 2^5 \\ 10000_b &= 16_d &&= 2^4 \\ 1000_b &= 8_d &&= 2^3 \\ 100_b &= 4_d &&= 2^2 \\ 10_b &= 2_d &&= 2^1 \\ 1_b &= 1_d &&= 2^0 \end{aligned}$$

Oefening

Bereken de respectievelijke decimale waarde:

- 11_b
- 100_b
- 1010_b
- 1100_b
- 101010_b
- 11110000_b

Van decimaal naar binair

Dit kan door het getal telkens door twee te delen:

- stel we willen 6_d omrekenen
- we delen dit telkens door 2 en schrijven de rest op:

$$\begin{aligned} 6/2 &= 3 \text{ rest } 0 \\ 3/2 &= 1 \text{ rest } 1 \\ 1/2 &= 0 \text{ rest } 1 \end{aligned}$$

- uitkomst is dus 110_b (omgekeerde volgorde)

Oefening

Bereken de respectievelijke binaire waarde:

- 3_d
- 7_d
- 4_d
- 14_d
- 33_d
- 523_d
- 302_d

Het hexadecimale talstelsel

- binaire getallen zijn lastig om mee te werken
- vandaar dat men meestal 4 cijfers samenvat (4 bits = nibble)
- 4 binaire cijfers (van 0000_b tot 1111_b) = 16 mogelijke waarden
- nieuw talstelsel: hexadecimaal met 16 cijfers **[0..F]**
- voordeel 1: getallen worden veel korter
- voordeel 2: men kan telkens per 4 groeperen

De hexadecimale cijfers

Binair	Decimaal	Hexadecimaal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Conversie binair - hexadecimaal

Van binair naar hexadecimaal:

- groepeer per 4 bits en reken dit groepje om
- $1111101_b = 0111\ 1101_b = 7D_h$

Van hexadecimaal naar binair:

- schrijf eenvoudig cijfer per cijfer in binair formaat
- $3FA_h = 0011\ 1111\ 1010_b = 1111111010_b$

Oefening

Vul onderstaande tabel aan:

Binair	Decimaal	Hexadecimaal
	16	
		FF
11011		
		17
	84	
1010110		
		A3E
	317	
		9F9F

Dataorganisatie

Bit (1_b)

- komt overeen met één binair cijfer, 0 of 1 dus

Nibble (1011_b of B_h)

- komt overeen met 4 binaire, of één hexadecimaal cijfer

Byte ($1010\ 1110_b$ of AE_h)

- komt overeen met 8 binaire, of 2 hexadecimale cijfers

Word ($1101\ 1100\ 0001\ 1010_b$ of $DC1A_h$)

- komt overeen met 16 binaire, of 4 hexadecimale cijfers

Karaktercodes

- elke computer gebruikt een verzameling karakters
- ook deze worden inwendig binair voorgesteld
- minimaal bestaat een set uit: 26 letters, 10 cijfers en enkele leestekens
- elk karakter wordt een nummer toegekend: dit is de karaktercode
- men kan gebruik maken van een 6-, 7-, 8- of 9-bits code
- bij 8-bits (1 byte) zijn 256 karakters mogelijk

ASCII

- ASCII is een 7-bits karaktercode (128 karakters)
- het 8ste bit wordt meestal voor foutdetectie gebruikt
- buiten de VS meer karakters nodig, ook 8ste bit gebruiken
- dit is de extended ASCII code
- helaas is dit niet één standaard

ASCII code

Dec	Hex	Char
07	07	BEL
10	0A	LF
13	0D	CR
32	20	spatie
33	21	!
48	30	0
49	31	1
65	41	A
66	42	B
97	61	a
98	62	b

Unicode

- ASCII bevat enkel letters uit het Engels
- Latin-1 bevat ook letters met accenten (West-Europa)
- Unicode bevat meerdere schriften
- eerste versie bood ruimte aan 65536 tekens (2^{16})
- later uitgebreid naar één miljoen (o.a. voor Chinees)
- aantal bytes per teken 2 (UCS-2) of 4 (UCS-4)
- ruimteverspillend bij ons: UTF-8 compatibel met 7-bit ASCII