

Arduino C++ #1

(C) Richèl Bilderbeek 

February 13, 2014

0.1 Overzicht

1. Doel
2. Arduino IDE
3. 'setup' en 'loop'
4. Fouten
5. data typen
6. if
7. for

0.2 Doel

- Fouten leren lezen
- Internet zoektermen
- Basis van programmeren

0.3 Arduino IDE

- Doel: gemakkelijk programmeren voor beginners (NB: voor gevorderden minder geschikt)
- taal: C++, volgens de C++03 standaard (niet C++0x, niet C++11)
- bibliotheken: C, Arduino
- voegt zelf nog dingen toe (zoals de main functie)

0.4 Van code naar machinetaal

- Code is bedoeld voor mensen om te lezen en te schrijven
- Een chip begrijpt alleen maar machinetaal
- Een compiler doet de vertaling van code naar machinetaal (NB: er zijn meer programma's die helpen, die noem ik vandaag allemaal 'compiler')
- De compiler is erg precies, en dat is goed
- De compiler heeft altijd gelijk!

0.5 Minimum programma

```
void setup() {}  
void loop() {}
```

0.6 Fouten

De code:

```
//(alle code gewist)
```

Foutmelding:

```
core.a(main.cpp.o):  
In function 'main': /[...] / main.cpp:11:  
undefined reference to 'setup'  
collect2: error: ld returned 1 exit status
```

Kern:

```
undefined reference to 'setup'
```

Wat bedoelt de compiler:

De compiler heeft gehoord dat 'setup' moet bestaan, maar kan deze niet vinden

0.7 Opdracht

- Start 'Examples | Basics | Bare minimum'
- Probeer zoveel mogelijk verschillende fouten te maken, alleen in 'setup' ('loop' is immers precies hetzelfde)
- Noteer (1) de code (2) de kern van de foutmelding (3) wat je denkt dat de compiler bedoelt (4) laat ruimte over voor later; als je snapt wat de
- Ik vind er met gemak achttien!
- Deze fouten zul je nog vaak tegen komen!

0.8 Fouten

<pre>//Een letter weg oid setup() {} vid setup() {} vod setup() {} voi setup() {} voidsetup() {} void etup() {} void stup() {} void seup() {} void setp() {} void setu() {}</pre>	<pre>void setup ({} void setup) {} void setup () } void setup () { //Een woord of tekencombinatie weg setup() {} void () {} void setup {} void setup ()</pre>
---	---

0.9 Variabelen en data typen

- Variabele: iets wat een computer moet onthouden, iets dat een mens kan lezen
- int: een geheel getal, bijvoorbeeld: een pin nummer, een aantal seconden

```
const int pin_led = 2;
```

- double: een gebroken getal, bijvoorbeeld: een voltage

```
const double voltage = 3.3; // Volt
```

- bool: een ja/nee, bijvoorbeeld: is de drukknop ingedrukt?

```
const bool is_ingedrukt = /* iets ingewikkelds */;
```

- nog veel meer

0.10 Ombouwen van data type

- Soms gebeurt dit stiekum of per ongeluk
- Maar dit kan ook open en expres, met `static_cast`

```
const int waarde
    = analogRead(A0);
const double voltage
    = static_cast<double>(waarde)
      / 1024.0;
```

0.11 if statement

- Als je iets soms wel en soms niet wil

```
if (analogRead(A0) < 512) { digitalWrite(2,HIGH); }  
else { digitalWrite(2,LOW); }
```

Teken	Spreek uit
==	is gelijk aan
!=	is ongelijk aan
<	is kleiner dan
<=	is kleiner of gelijk dan
>	is groter dan
>=	is groter of gelijk dan

0.12 for loop

- Als je een teller wilt laten lopen

```
for (int teller=0; teller!=255; ++teller)
{
    analogWrite(2,teller);
}
```

- Lees dit als:
 1. Zet een int genaamd 'teller' op de beginwaarde nul
 2. Doe de code tussen accolades zo lang 'teller' niet gelijk is aan 255
 3. Na de code tussen accolades: tel één op bij teller ('++teller') en ga naar stap 2

0.13 Testen

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  const double x = 2.0 / 7.0; Serial.println(x);
  const int i = 123; Serial.println(i);
  const bool b = true; Serial.println(b);
  for (int i=0; i!=255; ++i) Serial.println(i);
}
```

0.14 Opdracht

- Zoek een toepassing voor een if statement
- Zoek een toepassing voor een for loop