

# Arduino Quick Reference Card

v1.0 NL, 29 november 2011, door Ernest Neijenhuis PA3HCM

## Structuur

```
// declaraties en includes
void setup() {
  // Deze code wordt alleen bij het
  // opstarten uitgevoerd
}
void loop() {
  // Deze code blijft doorgaan totdat
  // de spanning eraf wordt gehaald
}
```

## Syntax

```
// Dit is één regel met commentaar

/*
  Dit zijn
  meerdere regels
  met commentaar
*/

{} - Code wordt gegroepeerd met accolades
; - Iedere regel met code eindigt met een
    puntkomma
```

## Variabelen

**int** Integer, een getal van 2 bytes (16 bits). Heeft geen getallen achter de komma en heeft een waarde tussen -32,768 en 32,767.

**long** Net als **int**, maar dan 4 bytes (32 bits) en heeft een bereik van -2,147,483,648 t/m 2,147,483,647.

**boolean** Binair, oftewel waar of onwaar. Gebruikt slechts 1 bit geheugen.

**float** voor getallen met cijfers achter de komma. Kost 4 bytes (32 bits) RAM en heeft een bereik van -3.4028235E+38 t/m 3.4028235E+38.

**char** Een enkel teken, maakt gebruik van de ASCII code (bijv. 'A' = 65). Gebruikt 1 byte (8 bits) geheugen. De Arduino ziet een string als een array van `char`'s.

## Rekenen

**=** (toewijzing) maakt iets gelijk aan iets anders (bijv.  $X = 10 * 2$  (x heeft nu de waarde 20))

**%** (modulo) geeft de rest als het nummer gedeeld wordt door een ander getal (bijv.  $12 \% 10$  (geeft 2))

**+** (optellen)

**-** (aftrekken)

**\*** (vermenigvuldigen)

**/** (delen)

## Vergelijken

**==** (gelijk aan) (bijv.  $12 == 10$  is FALSE of  $12 == 12$  is TRUE)

**!=** (ongelijk aan) (bijv.  $12 != 10$  is TRUE of  $12 != 12$  is FALSE)

**<** (kleiner dan) (bijv.  $12 < 10$  is FALSE of  $12 < 12$  is FALSE of  $12 < 14$  is TRUE)

**>** (groter dan) (bijv.  $12 > 10$  is TRUE of  $12 > 12$  is FALSE of  $12 > 14$  is FALSE)

## Control structures

```
if( conditie ){ ... }
else if( conditie ){ ... }
else { ... }
```

Voert de code tussen de eerste accolades uit als de conditie `true` is, of anders de code tussen de tweede accolades als de tweede conditie waar is, of anders de code tussen de derde accolades achter `else`.

```
for(int i = 0; i < #aantal; i++){ ... }
```

Voert de code tussen de accolades een aantal keren uit. Optellen gaat met `++`, je kunt ook aftellen met `--`.

```
delay(tijd);
```

Wacht `tijd` milliseconden.

## Digitaal

```
pinMode(pin, mode);
```

Stel een pin in als `INPUT` of `OUTPUT`. `Pin` is het pin nummer, tussen 0 en 19 (analog 0-5 zijn 14-19).

```
digitalWrite(pin, waarde);
```

Als een pin ingesteld is als `OUTPUT`, kan de output op `HIGH` (+5 volt) of `LOW` (0 volt) worden gezet.

```
int digitalRead(pin);
```

Als een pin is ingesteld als `INPUT`, kan deze uitgelezen worden, het resultaat is `HIGH` of `LOW`.

## Analoog

```
analogWrite(pin, waarde);
```

De pinnen 3, 5, 6, 9, 10 en 11 ondersteunen PWM (pulse width modulation). Met deze functie stel je de duty cycle in: 0 geeft 0% duty cycle (ongeveer 0 volt) and 255 geeft 100% duty cycle (5 volt)

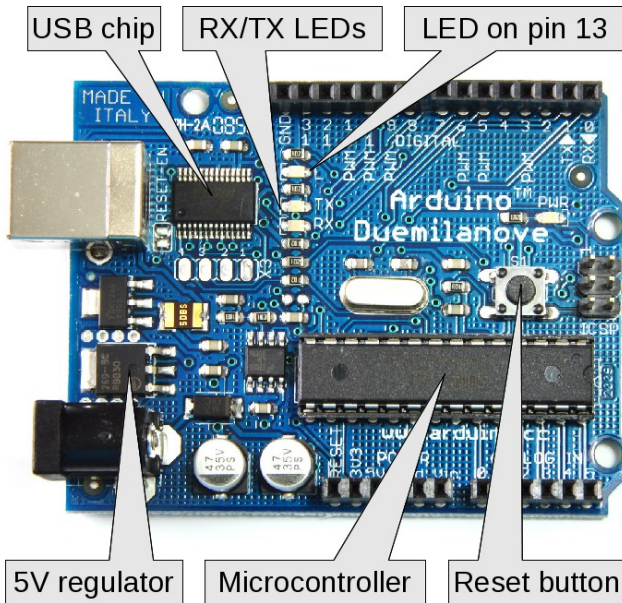
```
int analogRead(pin);
```

Lees de waarde uit van de analog input pin. De waarde is tussen 0 (0 volt) en 1023 (5 volt).

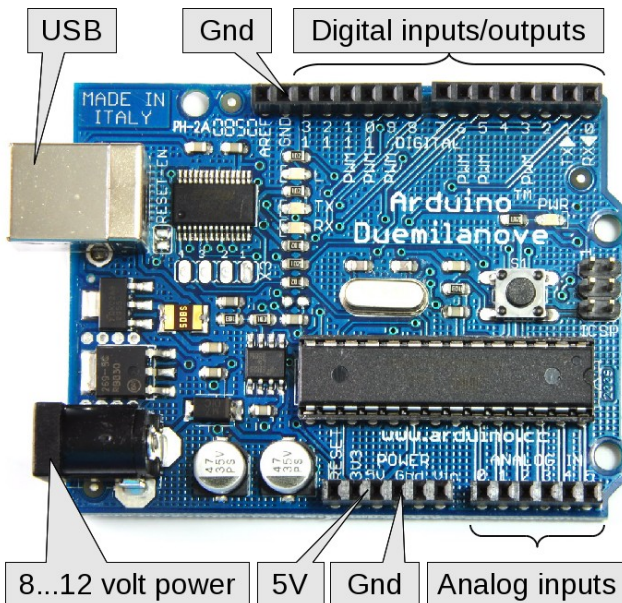
# Arduino Quick Reference Card

v1.0 NL, 29 november 2011, door Ernest Neijenhuis PA3HCM

## Componenten



## Aansluitingen



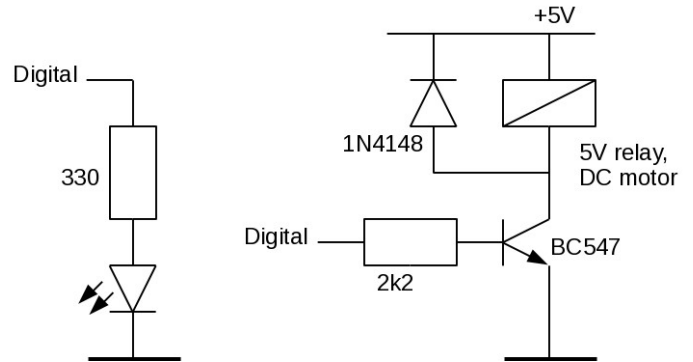
## Spanning en stroom

Ext. spanning: 6...20Vdc (aanbevolen 8...12Vdc)  
Max. stroom per I/O pin: 40mA

## Eenvoudige output schakelingen

Links: LED

Rechts: 5V relais of klein DC motortje

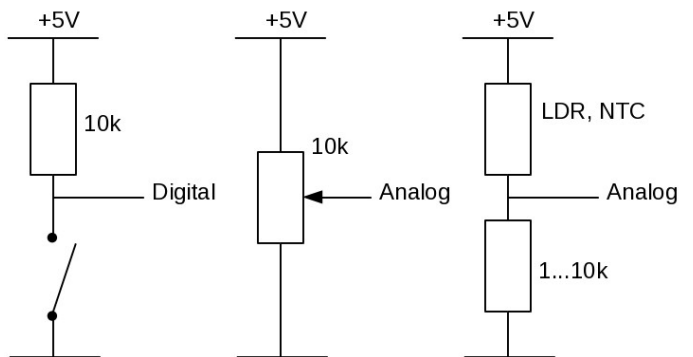


## Eenvoudige input schakelingen

Links: drukknop of schakelaar

Midden: potmeter

Rechts: Licht- of temperatuurafhankelijke weerstand

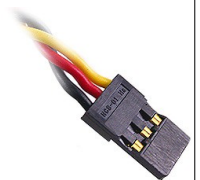


## Servo aansluitingen

Zwart of bruin = GND

Rood = +5V

Wit of oranje = PWM



## Colofon

Versie: 1.0 NL

Uitgebracht: 29 November 2011

Auteur: Ernest Neijenhuis PA3HCM

Meeste tekst overgenomen uit "Experimenter's Guide for Arduino" door oomlout.com