

Hoe voeg ik code samen?

(C) Richèl Bilderbeek 

August 15, 2014

0.1 Overzicht

1. Standaard volgorde
2. Regels samenvoegen
3. Conclusie

0.2 Standaard volgorde

- Includes
- Globale variabelen
- Functies, waaronder 'setup' en 'loop'

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12,11,5,4,3,2);

void ZegHallo() { lcd.print("Hello World!"); }

void setup() { lcd.begin(16, 2); }

void loop() { ZegHallo(); }
```

0.3 Samenvoegen includes

- Hou de includes bij de includes
- Elke volgorde is goed

```
#include <Servo.h>
```

en

```
#include <LiquidCrystal.h>
```

wordt bijvoorbeeld

```
#include <LiquidCrystal.h>
```

```
#include <Servo.h>
```

0.4 Samenvoegen globale variabelen

- Hou de globale variabelen bij de globale variabelen
- Houdt de volgorde aan van de losse programma's
- Check of bijvoorbeeld pinnummers niet dubbel worden gebruikt

```
const int pin_servo{9};
```

en

```
const int pin_led_red{8};
```

```
const int pin_led_red{pin_led_red + 1};
```

wordt

```
const int pin_servo{9};
```

```
const int pin_led_red{10};
```

```
const int pin_led_red{pin_led_red + 1};
```

0.5 Samenvoegen normale functies

- Hou de niet-setup en niet-loop functies bij de functies
- Houdt de volgorde aan van de losse programma's

```
void DraaiServo() { /* */}
```

en

```
void ZetAan() { /* */}
```

```
void ZetUit() { /* */}
```

```
void Knipper() { /* Roept ZetAan en ZetUit aan */ }
```

wordt

```
void DraaiServo() { /* */}
```

```
void ZetAan() { /* */}
```

```
void ZetUit() { /* */}
```

```
void Knipper() { /* Roept ZetAan en ZetUit aan */ }
```

0.6 Samenvoegen setup functies

- Voeg de code van de setup functie samen tot een setup functie
- Houdt de volgorde aan van de losse stukken code

```
void setup() { my_servo.attach(pin_servo); }
```

en

```
void setup() { const int breedte{16}; const int hoogte{2};  
my_lcd.begin(breedte, hoogte); }
```

wordt

```
void setup() {  
my_servo.attach(pin_servo);  
const int breedte{16}; const int hoogte{2};  
my_lcd.begin(breedte, hoogte);  
}
```

0.7 Samenvoegen loop functies

- Voeg de code van de loop functie samen tot een loop functie
- Houdt de volgorde aan van de losse stukken code
- Bedenk wat je wilt dat het samengevoegde programma doet

0.8 Voorbeeld

```
void loop ()
{
  for (int pos{0}; pos!=180; ++pos)
  {
    myservo.write(pos);
    delay(1000);
  }
}

en

void loop ()
{
  Serial.println(analogRead(pin_sensor));
  delay(1000);
}
```

0.9 Naieve samenvoeging

```
void loop()
{
  for(int pos{0}; pos!=180; ++pos)
  {
    myservo.write(pos);
    delay(1000);
  }
  Serial.println(analogRead(pin_sensor));
  delay(1000);
}
```

0.10 Juiste samenvoeging

```
void loop()
{
  for(int pos{0}; pos!=180; ++pos)
  {
    Serial.println(analogRead(pin_sensor));
    myservo.write(pos);
    delay(1000);
  }
}
```

0.11 Conclusie

- De includes zijn gemakkelijk samen te voegen
- Let op bij het samen voegen van globale variabelen
- Normale functies zijn gemakkelijk samen te voegen
- De setup functies zijn gemakkelijk samen te voegen
- Bij het samenvoegen van de loop functies moet het meest worden nagedacht