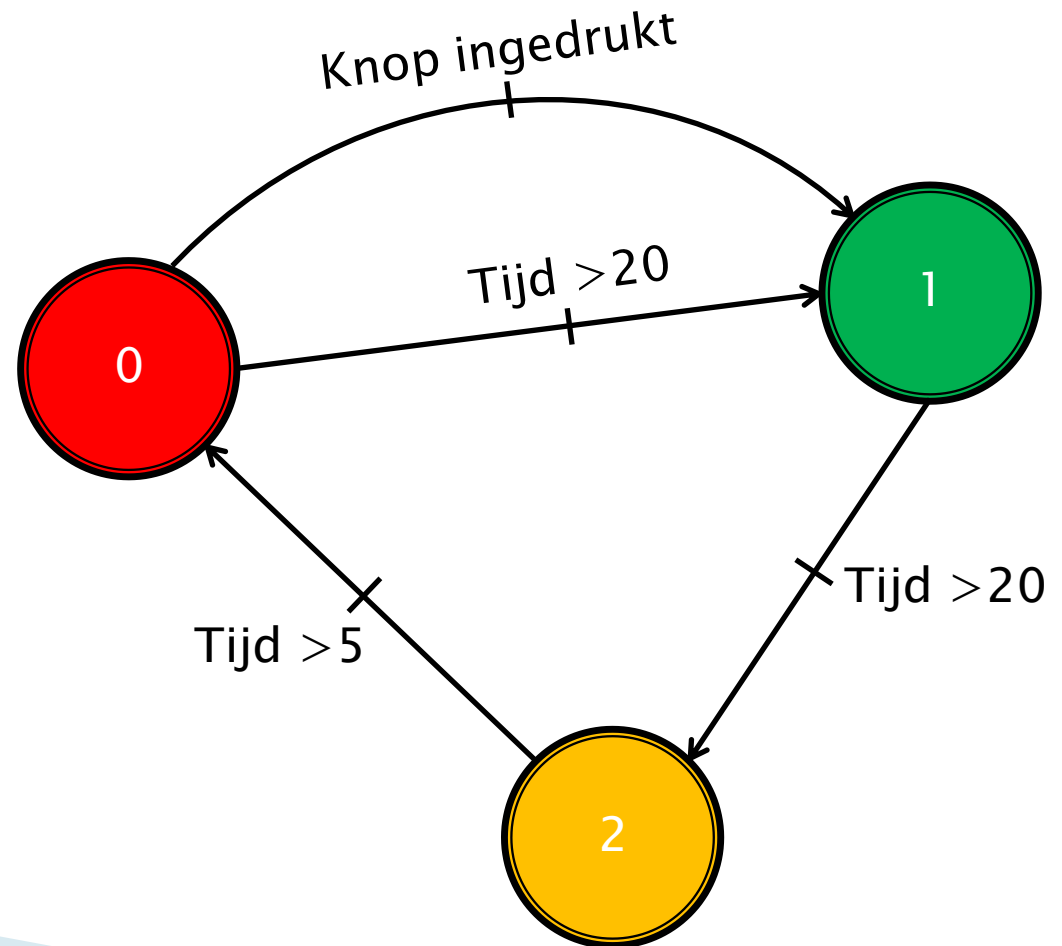


CAR – Dag 4

C, Arduino & Robots

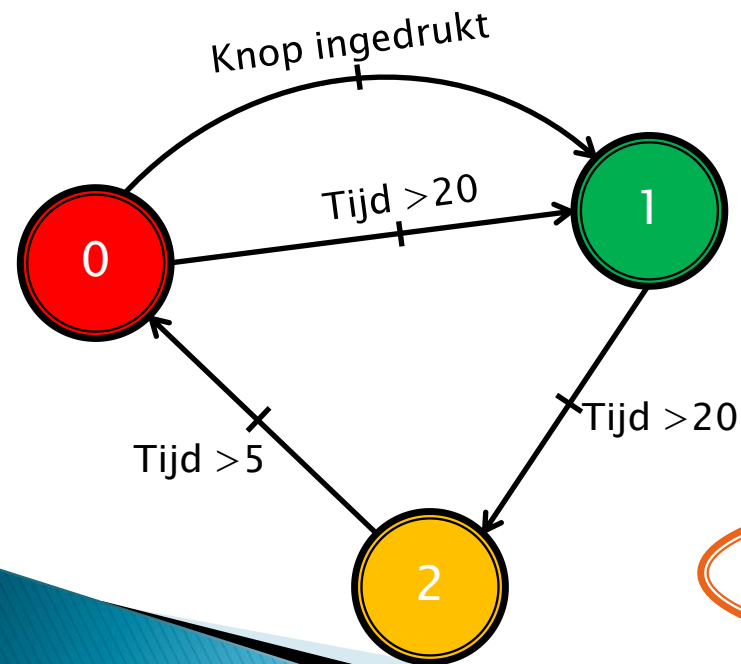
Huiswerk

- ▶ 1. Drukknop voor groen licht (als het rood is).



Huiswerk...

- ▶ Drukknop voor groen licht (als het rood is).



les_4_p10_stoplicht_3

```
void setup() {  
  pinMode(5, OUTPUT);           // rode led  
  pinMode(6, OUTPUT);           // gele led  
  pinMode(7, OUTPUT);           // groene led  
  
  pinMode(11, INPUT_PULLUP);    // knop input met pull-up  
  
  // start serial  
  Serial.begin(115200);  
  Serial.println("Hallo arduino wereld.");  
}  
  
int Toestand = 0;  
int Tijd = 0;  
  
void loop() {  
  delay(100);  
  Tijd = Tijd + 1;  
  Serial.println(Tijd);  
  
  switch(Toestand) {  
    case 0 : { // rood licht  
      // rood          geel          groen  
      digitalWrite(5, 1); digitalWrite(6, 0); digitalWrite(7, 0);  
  
      if (Tijd > 20) {  
        Tijd = 0;  
        Toestand = 1;  
      }  
  
      if (digitalRead(11) == 0) { // knop ingedrukt?  
        Tijd = 0;  
        Toestand = 1;  
      }  
  
      break;  
    }  
  }  
}
```

Huiswerk...

- Voorspel de output

les_3_p60_puzzel

```
void setup() {  
  // start serial  
  Serial.begin(115200);  
  Serial.println("Hallo arduino wereld.");  
}  
  
int State = 7;  
  
void loop() {  
  delay(100);  
  
  Serial.print("****");  
  
  switch(State) {  
    case 0 : {  
      Serial.println("Nul ");  
      State = 1;  
    }  
    case 1 : {  
      Serial.println("Een ");  
      State = 2;  
      break;  
    }  
    case 2 : {  
      Serial.println("Twee");  
      State = 0;  
    }  
    default : {  
      Serial.println("Ongeldige state");  
      State = 1;  
      break;  
    }  
  } // einde van switch  
}
```

Huiswerk...

```
Hallo arduino wereld.  
***Ongeldige state  
***Een  
***Twee  
Ongeldige state  
***Een  
***Twee  
Ongeldige state  
***Een  
***Twee  
Ongeldige state  
***Een  
***Twee  
Ongeldige state
```

les_3_p60_puzzel

```
void setup() {  
    // start serial  
    Serial.begin(115200);  
    Serial.println("Hallo arduino wereld.");  
}  
  
int State = 7;  
  
void loop() {  
    delay(100);  
  
    Serial.print("****");  
  
    switch(State) {  
        case 0 : {  
            Serial.println("Nul ");  
            State = 1;  
        }  
        case 1 : {  
            Serial.println("Een ");  
            State = 2;  
            break;  
        }  
        case 2 : {  
            Serial.println("Twee");  
            State = 0;  
        }  
        default : {  
            Serial.println("Ongeldige state");  
            State = 1;  
            break;  
        }  
    } // einde van switch  
}
```

De C manier van printen: printf

- Duidelijke manier om informatie te geven vanuit je programma.

les_4_p20_printf

```
// to support printf
int my_putc(char c, FILE *t) {
    Serial.write(c);
}
```

```
// the setup routine runs once when you press reset:
void setup() {
    // start serial
    Serial.begin(115200);
```

```
fdevopen( &my_putc, 0); // device 0 (stdout) output naar my_putc()
```

```
printf("Opstarten gereed.");
}
```

```
int Seconde;
void loop() {
    delay(1000);
    Seconde = Seconde + 1;
```

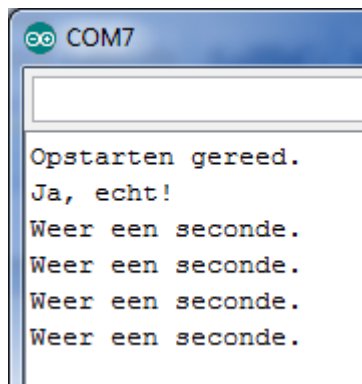
```
printf("Weer een seconde.");
}
```

Toevoegen

Gebruiken

De C manier van printen: printf

► NewLine (\n)



```
COM7
Opstarten gereed.
Ja, echt!
Weer een seconde.
Weer een seconde.
Weer een seconde.
Weer een seconde.
```

```
les_4_p25_printf_If$

// to support printf
int my_putc(char c, FILE *t) {
    Serial.write(c);
}

// the setup routine runs once when you press reset:
void setup() {
    // start serial
    Serial.begin(115200);

    fdevopen( &my_putc, 0); // device 0 (stdout) output naar my_putc()

    printf("Opstarten gereed.\nJa, echt!\n");
}

int Seconde;
void loop() {
    delay(1000);
    Seconde = Seconde + 1;

    printf("Weer een seconde.\n");
}
```

De C manier van printen: printf

► Meer escape karakters

Escape sequence	Connotation
<code>\n</code>	Newline (Line Feed)
<code>\r</code>	Carriage Return
<code>\t</code>	Horizontal Tab
<code>\\</code>	Backslash
<code>\'</code>	Single quotation mark
<code>\"</code>	Double quotation mark
<code>\0</code>	Null (string terminator)
<code>\xhh</code>	<u>Hexadecimal representation</u>

http://en.wikipedia.org/wiki/Escape_sequences_in_C

De C manier van printen: printf

- ▶ Variabele (waarde) als parameter.
- ▶ %d => decimale waarde in format string
- ▶ %x => hexadecimale waarde

Netjes onder elkaar:

- ▶ %8d => gebruik (minimaal) 8 posities
- ▶ %08d => gebruik (minimaal) 8 posities met voorloop nullen.

les_4_p30_printf_int

```
int Seconde;  
void loop() {  
    delay(1000);  
    Seconde = Seconde + 1;  
  
    printf("Er zijn %d seconden verstreken sinds reset.\n", Seconde)  
}
```

while-loop

```
int teller = 0;  
while (teller < 6) {  
    teller = teller + 1;  
    printf("Teller is %d\n", teller);  
}
```

Oefening: while-loop (1)

les_4_p50_while

```
// to support printf
int my_putc(char c, FILE *t) {
    return Serial.write(c);
}

// the setup routine runs once when you press reset:
void setup() {
    // start serial
    Serial.begin(115200);

    fdevopen( &my_putc, 0); // device 0 (stdout) output naar my_putc()

    printf("Opstarten gereed.\n");
}

void loop() {

    int teller = 0;
    while (teller < 6) {
        teller = teller + 1;
        printf("Teller is %d\n", teller);
    }

    delay(5000);
}
```

Resultaat: while-loop (1)

les_4_p50_while

```
// to support printf
int my_putc(char c, FILE *t) {
    return Serial.write(c);
}

// the setup routine runs once when you press reset:
void setup() {
    // start serial
    Serial.begin(115200);

    fdevopen( &my_putc, 0); // device 0 (stdout) output naar my_putc()

    printf("Opstarten gereed.\n");
}

void loop() {

    int teller = 0;
    while (teller < 6) {
        teller = teller + 1;
        printf("Teller is %d\n", teller);
    }

    delay(5000);
}
```

COM16

```
Opstarten gereed.
Teller is 1
Teller is 2
Teller is 3
Teller is 4
Teller is 5
Teller is 6
```

Oefening: while-loop (2)

les_4_p51_while

```
// to support printf
int my_putc(char c, FILE *t) {
    return Serial.write(c);
}

// the setup routine runs once when you press reset:
void setup() {
    // start serial
    Serial.begin(115200);

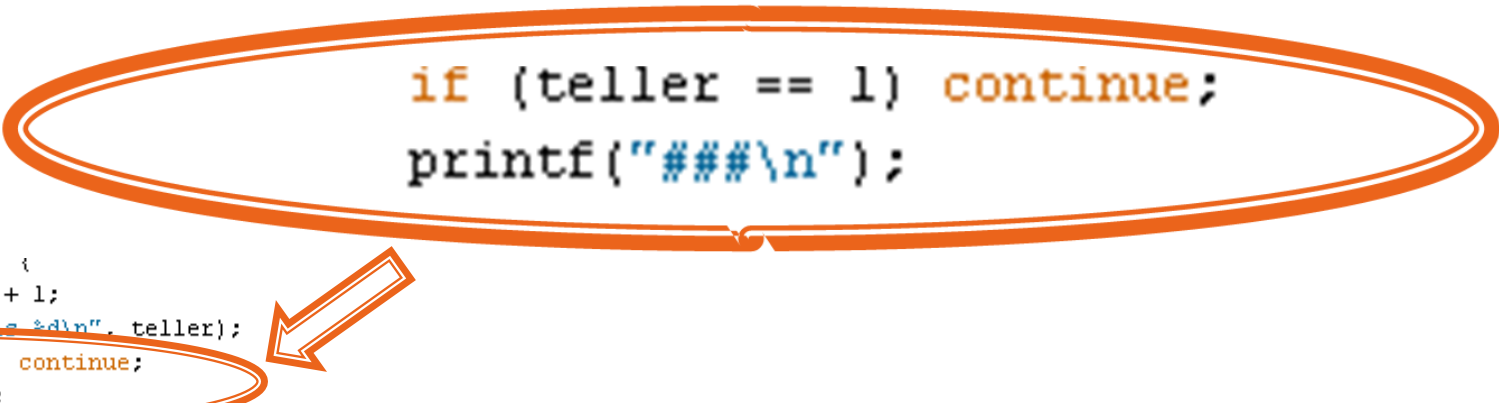
    fdevopen( &my_putc, 0); // device 0 (stdout) output naar my_putc()

    printf("Opstarten
}

void loop() {

    int teller = 0;
    while (teller < 6) {
        teller = teller + 1;
        printf("Teller is %d\n", teller);
        if (teller == 1) continue;
        printf("###\n");
    }

    delay(5000);
}
```



The diagram consists of a large orange oval that encloses the code block `if (teller == 1) continue;` and the line `printf("###\n");` in the `loop()` function. A white arrow with an orange outline points from this oval to a smaller orange oval that encloses the code block `if (teller == 1) continue;` and the line `printf("###\n");` in the `while` loop. This visualizes the logic of the `continue` statement, showing how it skips the rest of the current iteration and jumps to the start of the next iteration.

Resultaat: while-loop (2)

les_4_p51_while

```
// to support printf
int my_putc(char c, FILE *t) {
    return Serial.write(c);
}

// the setup routine runs once when you press reset:
void setup() {
    // start serial
    Serial.begin(115200);

    fdevopen( &my_putc, 0); // device 0 (stdout) output naar my_putc()

    printf("Opstarten gereed.\n");
}

void loop() {

    int teller = 0;
    while (teller < 6) {
        teller = teller + 1;
        printf("Teller is %d\n", teller);
        if (teller == 1) continue;
        printf("###\n");
    }

    delay(5000);
}
```

COM16

Opstarten gereed.
Teller is 1
Teller is 2

Teller is 3

Teller is 4

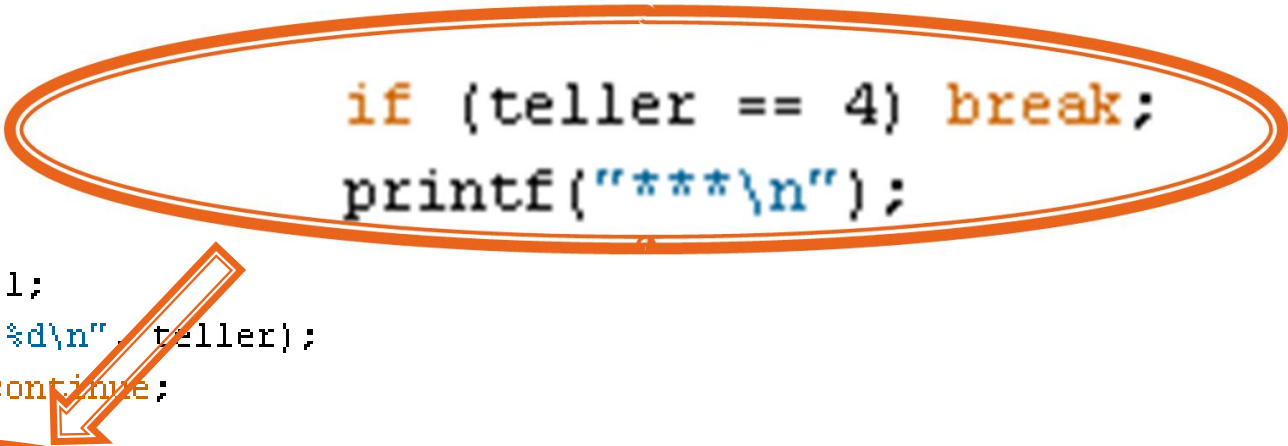
Teller is 5

Teller is 6
###

Oefening while-loop (3)

les_4_p52_while

```
void loop() {  
  
    int teller = 0;  
    while (teller < 6) {  
        teller = teller + 1;  
        printf("Teller is %d\n", teller);  
        if (teller == 1) continue;  
        printf("###\n");  
        if (teller == 4) break;  
        printf("***\n");  
    }  
  
    delay(5000);  
}
```



Resultaat while-loop (3)

les_4_p52_while

```
void loop() {  
  
    int teller = 0;  
    while (teller < 6) {  
        teller = teller + 1;  
        printf("Teller is %d\n", teller);  
        if (teller == 1) continue;  
        printf("###\n");  
        if (teller == 4) break;  
        printf("***\n");  
    }  
  
    delay(5000);  
}
```

COM16

```
Opstarten gereed.  
Teller is 1  
Teller is 2  
###  
***  
Teller is 3  
###  
***  
Teller is 4  
###  
Teller is 1  
Teller is 2  
###  
***  
Teller is 3  
###
```


Voor thuis: while-loop (4)

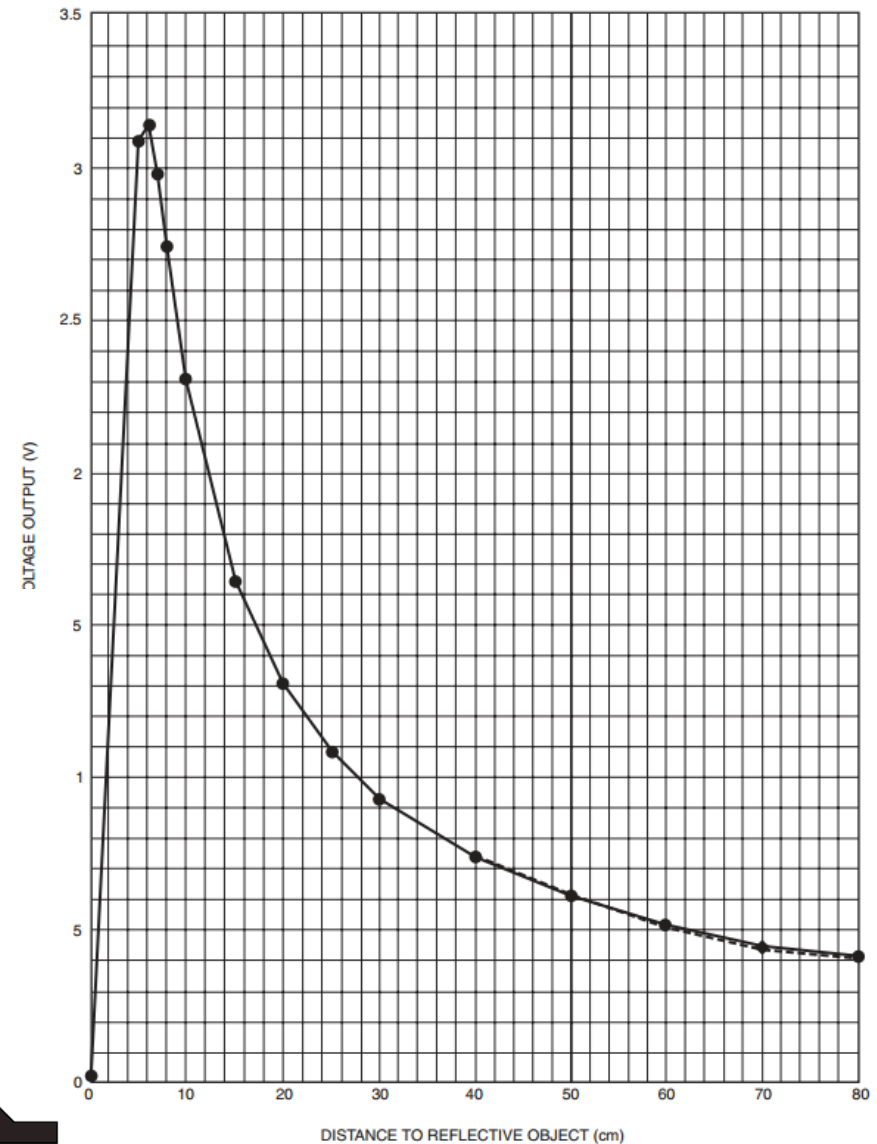
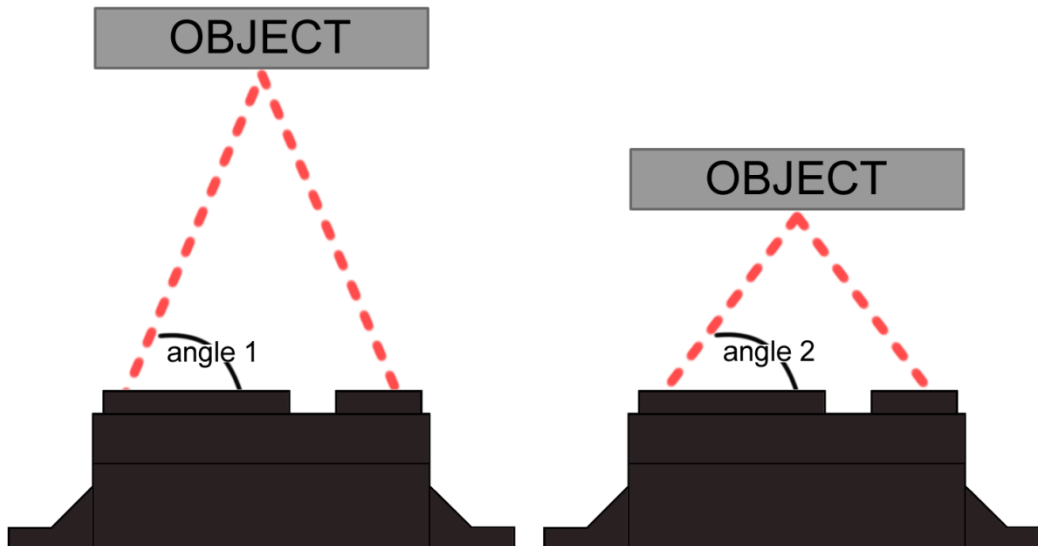
Test het programma met verkorte notatie.

```
teller = teller + 1;  
doet hetzelfde als  
teller += 1;  
doet hetzelfde als  
++teller;  
doet hetzelfde als  
teller++;
```

Niet helemaal hetzelfde,
maar in dit geval hetzelfde
resultaat.

Sharp sensor

- ▶ Driehoeksmeting
- ▶ Analoge uitgang



Analoge ingang

- ▶ Lees analoge waarde
- ▶ Bereik 0...1023 voor 0...5V
- ▶ Pin A0..A5

les_4_p60_analoog_in

```
// the setup routine runs once when you press reset:
void setup() {
  // start serial
  Serial.begin(115200);

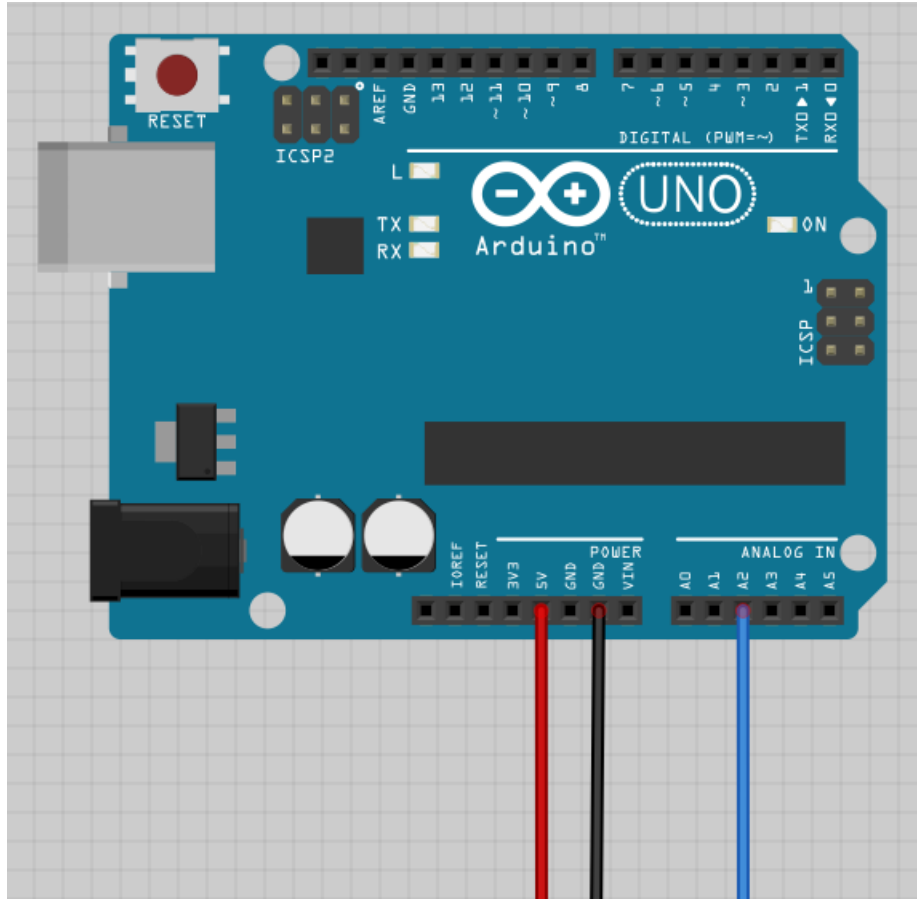
  fdevopen( &my_putc, 0); // device 0 (stdout) output naar my_putc()

  printf("Opstarten gereed.\n");
}

void loop() {
  delay(200);

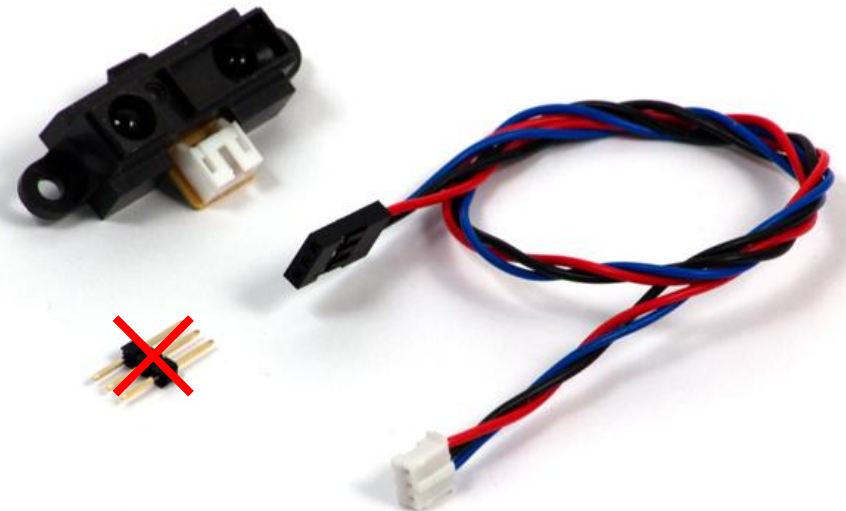
  int SensorValue = analogRead(A2);
  printf("Sensor input is %d\n", SensorValue);
}
```

Oefening: Analoge ingang



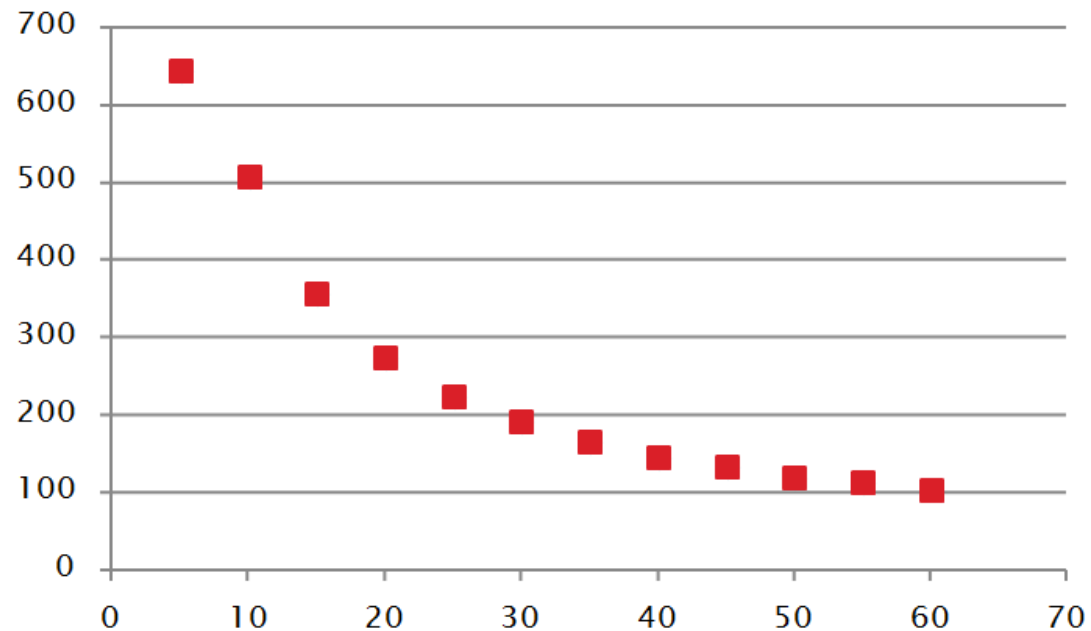
les_4_p60_analog_in

```
void loop() {  
  delay(200);  
  
  int SensorValue = analogRead(A2);  
  printf("Sensor input is %d\n", SensorValue);  
}
```



Sharp sensor – linear 1

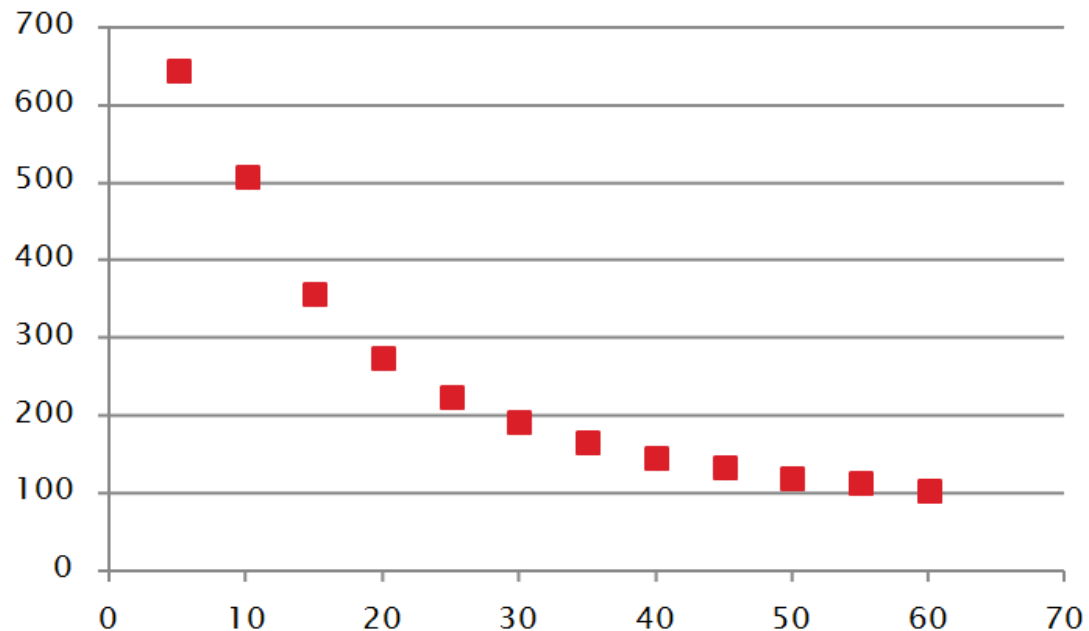
Afstand	ADC
5	646
10	510
15	358
20	277
25	225
30	193
35	168
40	148
45	136
50	122
55	115
60	106



Sharp sensor – linear 2

$L = \text{Constante} / \text{ADC}$

Afstand	ADC
5	646
10	510
15	358
20	277
25	225
30	193
35	168
40	148
45	136
50	122
55	115
60	106

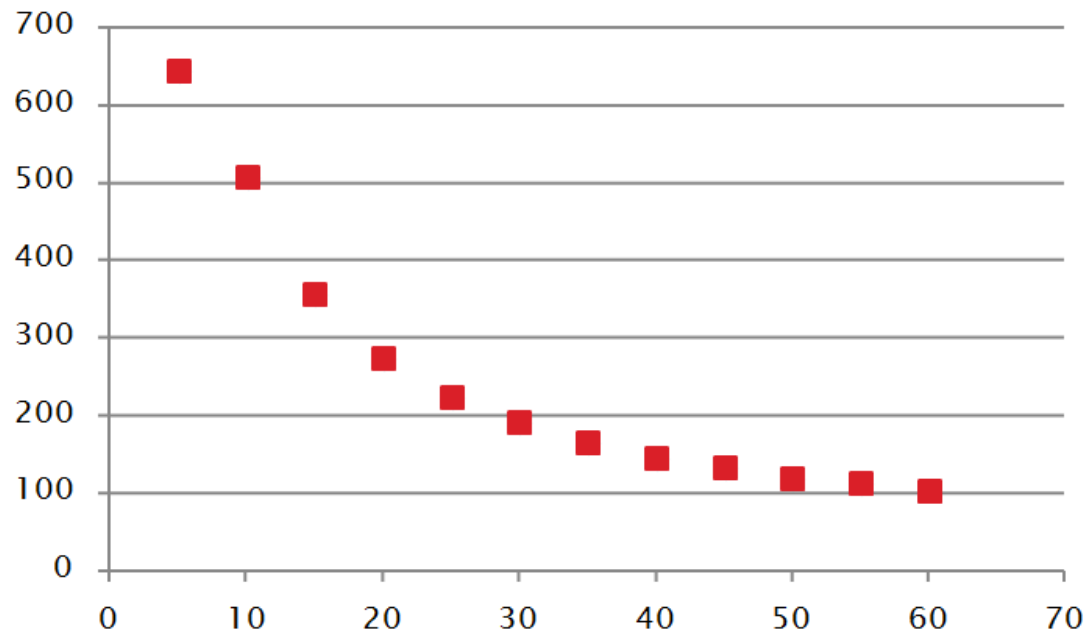


Sharp sensor – linear 3

$L = \text{Constante} / \text{ADC}$

$\text{Constante} = L * \text{ADC}(L)$

Afstand	ADC
5	646
10	510
15	358
20	277
25	225
30	193
35	168
40	148
45	136
50	122
55	115
60	106



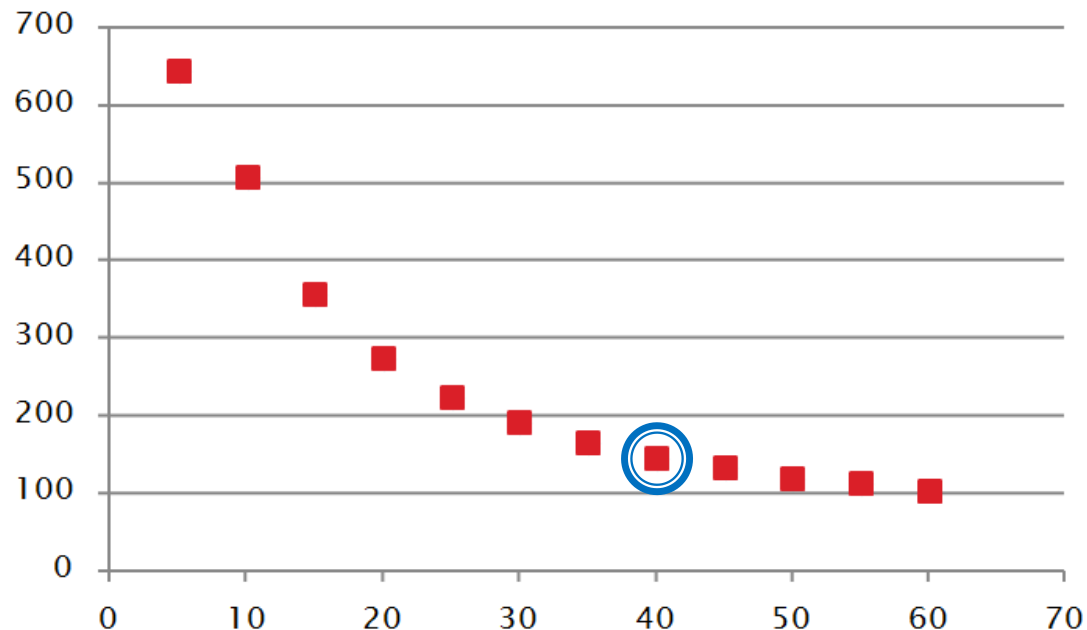
Sharp sensor – lineair 4

$L = \text{Constante} / \text{ADC}$

$\text{Constante} = L * \text{ADC}(L)$

$\text{Constante} = 40 * 148$

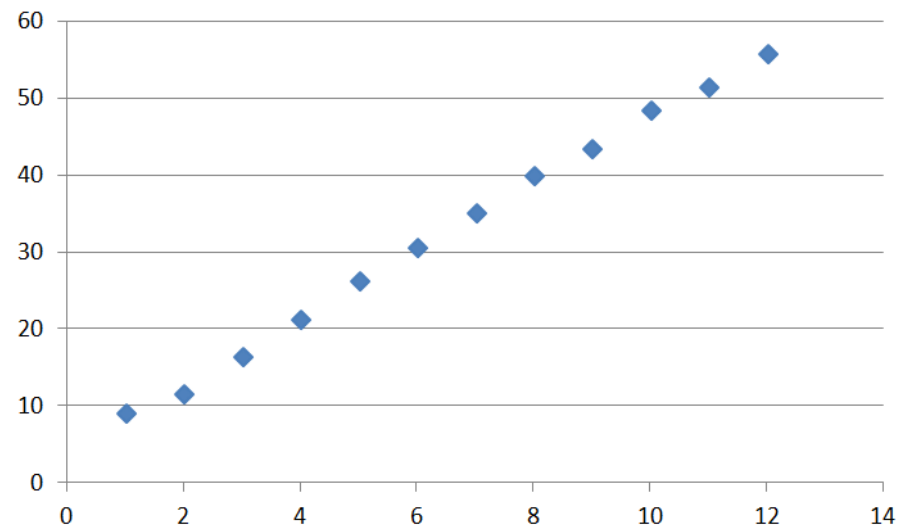
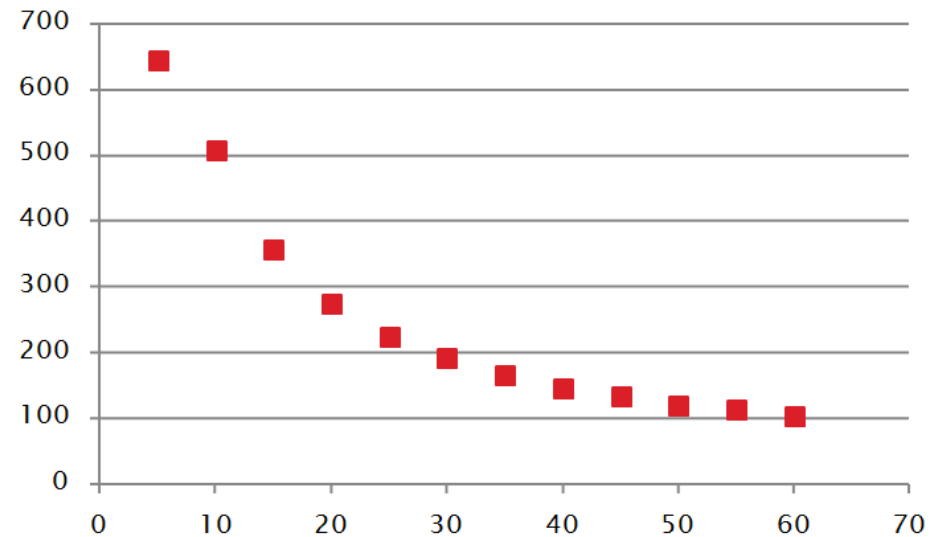
Afstand	ADC
5	646
10	510
15	358
20	277
25	225
30	193
35	168
40	148
45	136
50	122
55	115
60	106



Sharp sensor – linear 5

$$L = (40 * 148) / \text{ADC}$$

Afstand	ADC	L
5	646	9.164087
10	510	11.60784
15	358	16.53631
20	277	21.37184
25	225	26.31111
30	193	30.67358
35	168	35.2381
40	148	40
45	136	43.52941
50	122	48.52459
55	115	51.47826
60	106	55.84906



Oefening – Analooog 2

Formule: $\text{Afstand} = (40 * 148) / \text{SensorValue}$

Opgave:

- ▶ Reken de waarde van de analoge ingang (SensorValue) om naar Afstand (in cm).
- ▶ Print SensorValue en Afstand.

```
printf("Sensor input is %d, Afstand: %d\n", SensorValue, Afstand);
```

Resultaat- Analooog 2

les_4_p65_analoog_cm

```
// to support printf
int my_putc(char c, FILE *t) {
    return Serial.write(c);
}

// the setup routine runs once when you press reset:
void setup() {

    // start serial
    Serial.begin(115200);

    fdevopen( &my_putc, 0); // device 0 (stdout) output naar my_putc()

    printf("Opstarten gereed.\n");
}

void loop() {

    int SensorValue = analogRead(A2);
    int Afstand = (40*148) / SensorValue;

    printf("Sensor input is %d, Afstand %d\n", SensorValue, Afstand);
}
```

Afstand functie

les_4_p86_analoog_cm

```
// to support printf
int my_putc(char c, FILE *t) {
    return Serial.write(c);
}

// the setup routine runs once when you press reset:
void setup() {

    // start serial
    Serial.begin(115200);

    fdevopen( &my_putc, 0); // device 0 (stdout) output naar my_putc()

    printf("Opstarten gereed.\n");
}

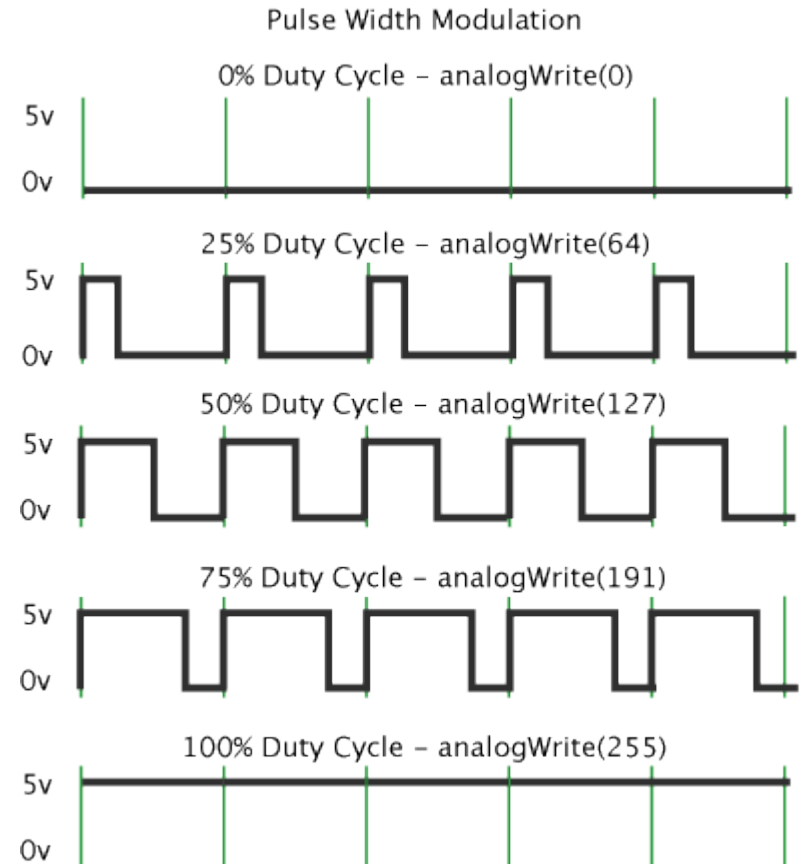
void loop() {

    int Centimeters = SharpAfstand(A2);
    printf("Sensoor Afstand: %d\n", Centimeters);
}

int SharpAfstand(int Pin)
{
    int SensorValue = analogRead(Pin);
    int Afstand = (40*148) / SensorValue;
    return Afstand;
}
```

PWM uitgang

- ▶ PWM = Pulsebreedte.
- ▶ ‘Soort’ analoge output
- ▶ Sturing van motoren en dimmen van led's
- ▶ Bereik 0...255
- ▶ pins 3, 5, 6, 9, 10, and 11



Oefening –PWM

- ▶ Stuur de PWM uitgang aan met $3 * \text{afstand}$ (in cm)
- ▶ Gebruik de functie `analogWrite()`
- ▶ led op uitgang D6
- ▶ print sensorwaarde (raw), afstand & pwm waarde

Resultaat- Analooq in & PWM

les_4_p70_analoog_in_pwm

```
// to support printf
int my_putc(char c, FILE *t) {
    return Serial.write(c);
}

// the setup routine runs once when you press reset:
void setup() {

    // start serial
    Serial.begin(115200);

    fdevopen( &my_putc, 0); // device 0 (stdout) output naar my_putc()

    printf("Opstarten gereed.\n");
}

void loop() {

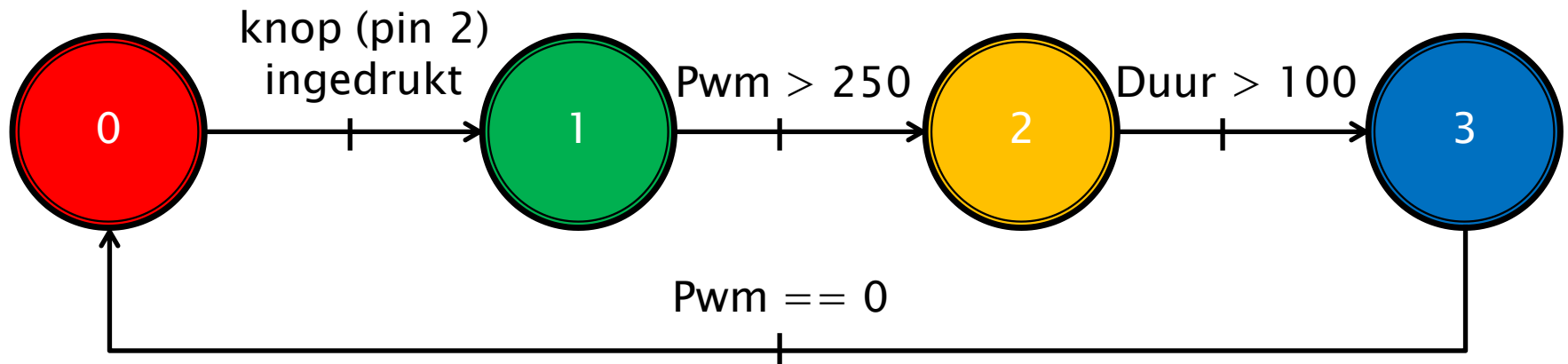
    int SensorValue = analogRead(A2);
    int Afstand = (40*140) / SensorValue;
    int PwmValue = Afstand * 3;

    analogWrite(6, PwmValue);

    printf("Sensor input is %d, Afstand %d, Pwm output is %d\n", sensorValue, Afstand, PwmValue);
}
```

Voor thuis

- ▶ 1. Maak de onderstaande statemachine voor PWM aansturing van de LED.
- ▶ Loop tijd is 10 ms.



State acties:

- ▶ State 0: $Pwm = 0$
- ▶ State 1: verhoog Pwm in stapjes van 2
- ▶ State 2: verhoog teller Duur met 1, te beginnen bij 0.
- ▶ State 3: verlaag Pwm in stapjes van 1.

Voor thuis...

► 2. Voorspel de output

Opmerking: optionele opgave, wordt volgende keer niet besproken.

les_4_p80_puzzel

```
int my_putc(char c, FILE *t) {
    return Serial.write(c);
}

void setup() {
    Serial.begin(115200);
    fdevopen(&my_putc, 0); // device 0 (stdout) output naar my_p
    printf("Opstarten gereed.\n");

    printf("* %d *\n", 123);
    printf("* %6d *\n", 123);
    printf("* %06d *\n", 123);
    printf("* %-6d *\n", 123);
    printf("* 0x%02x *\n", 123);
    printf("* 0x%02X *\n", 123);

    int x = 7;
    while (x) {
        printf("x: %d\n", x);
        x--;
    }

    int y = -3;
    while (y < 100) {
        printf("y: %d\n", y);
        y += 7;
    }

    while (1) {
        delay(1000);
        printf("lus \"while (1)\"\\n");
    }
}

void loop() {
    delay(1000);
    printf("Hoofdlus.\n");
}
```

Voor thuis...

3. Bouw de robot.

- ▶ Volg de beschrijving.
- ▶ Batterijhouder op 'chassis bottom', aan de voorzijde (beperkte gewicht op kogel-wieltje).
- ▶ Gebruik volle alkaline batterijen (geen NiMh / NiCd oplaadbare batterijen).
- ▶ Ardiuno eventueel voorzichtig vastmaken op 'chassis up'. Let op: geen kortsluiting maken.
- ▶ Motoren aansluiten op +/- schroef-aansluitingen op het motorshield
- ▶ Motor test programma MotorTest_Rev3.ino
- ▶ Controleer of beweging wiel overeen komt met tekst op console

Voor thuis...

3. Bouw de robot (vervolg).

Sharp sensor

- ▶ horizontaal monteren

Dag 5 oefeningen:

- ▶ eerst: sharp sensor recht vooruit
- ▶ vervolg: schuin naar rechts voor (ca 45 graden).

Problemen? bericht op

https://groups.google.com/forum/#!forum/hcc_robotmc

zodat we het voor de start van de volgende les
op kunnen lossen!

Tot slot

Volgende keer starten we een uur vroeger, om
12.00

