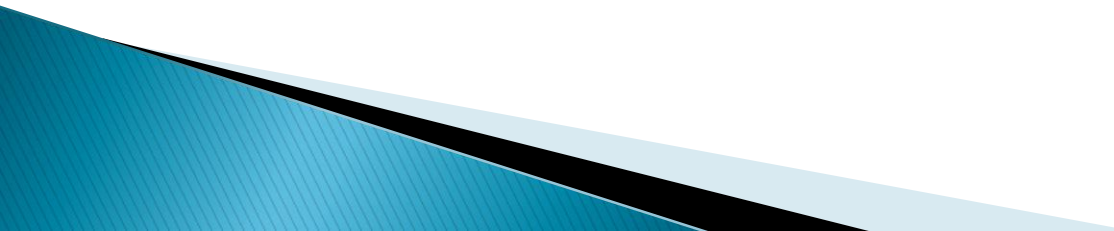


CAR – Dag 3

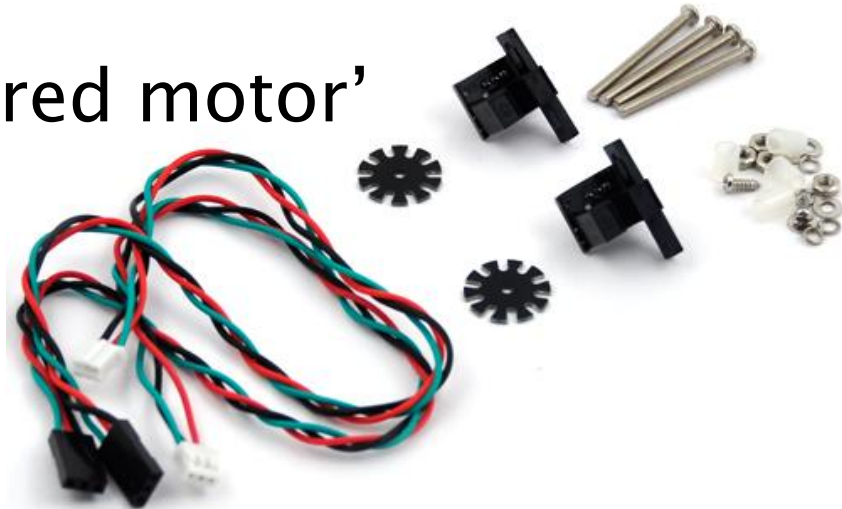
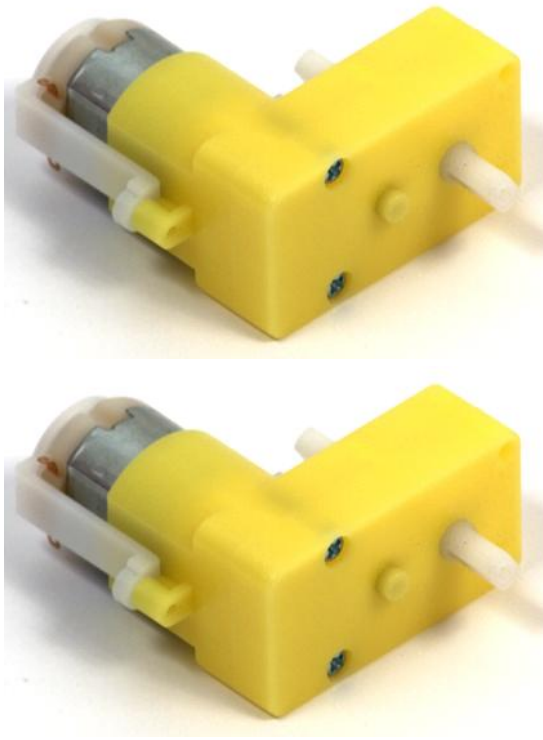
C, Arduino & Robots

Vooraf: 3 extra workshops

- ▶ Bij voldoende belangstelling
 - september
 - oktober en
 - december
 - ▶ November is de roborama wedstrijd.
 - ▶ Onderwerpen:
 - Uitlezen van encoders.
 - Gebruik encoder data voor robotbesturing.
- 

De robot (1: magician update)

- ▶ 'encoders' + 2 x 'dc geared motor'
- ▶ 'Batterijhouder 6xAA'



www.iprototype.nl
artikelnummers:

- 52779634
- 2x 60982364
- 66377015

De robot (2: alternatief)

- ▶ 'Robot platform 2wd' + 'encoders'

www.iprototype.nl

artikelnummers:

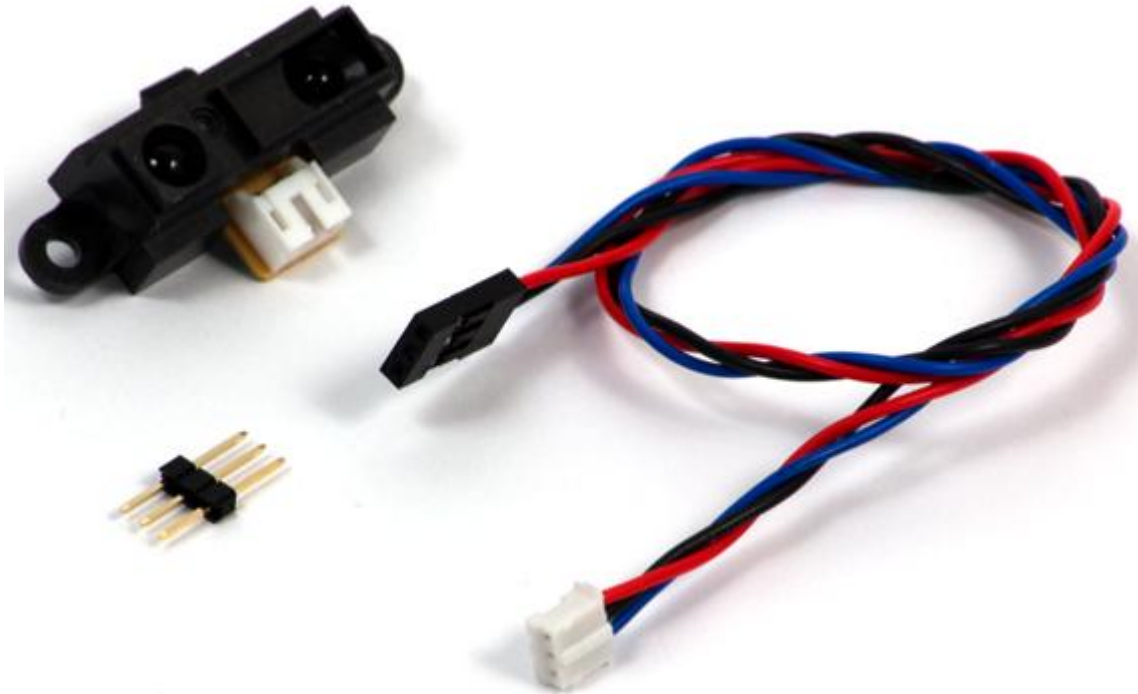
- 32328702
- 52779634



Voor de volgende keer

- ▶ 'IR Afstand sensor 80cm'

www.iprototype.nl
artikelnummer
• 57333680



Huiswerk...

- ▶ Voorspel de output van:

```
les_2_p40_puzzel

void setup() {
  Serial.begin(115200);

  int t;
  t = 29 / 10;
  Serial.println(t);

  t = (29 / 10) * 5;
  Serial.println(t);

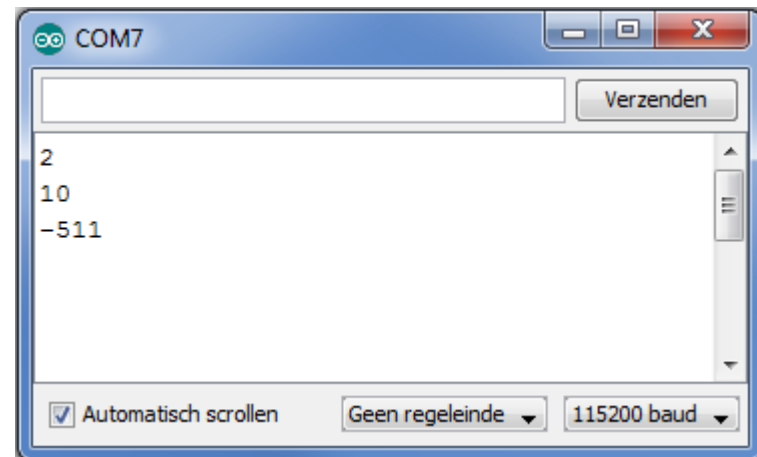
  t = 255 * 255;
  Serial.println(t);
}

void loop() {
}
```

Huiswerk...

- ▶ Voorspel de output van:

```
les_2_p40_puzzel  
  
void setup() {  
  Serial.begin(115200);  
  
  int t;  
  t = 29 / 10;  
  Serial.println(t);  
  
  t = (29 / 10) * 5;  
  Serial.println(t);  
  
  t = 255 * 255;  
  Serial.println(t);  
}  
  
void loop() {  
}
```



Huiswerk...

- ▶ 1 B – druk op knop: led aan; druk op knop: led uit.

Flank detectie

Variabele om vorige
toestand te onthouden

Bewaar huidige
toestand voor
volgende keer

Acties

Ingang 1x uitlezen en
resultaat bewaren

Test op verandering

(optioneel) test op
richting flank

```
static bool VorigeKnop;  
bool Knop = digitalRead(11);  
if (Knop != VorigeKnop) {  
    // flank (knop ingedrukt of losgelaten)  
    VorigeKnop = Knop;  
    if (Knop == 0) {  
        // knop is nu ingedrukt  
        Serial.println("De knop is ingedrukt");  
        ToggleLed();  
    }  
}
```

Huiswerk...

- ▶ 1 B – druk op knop: led aan; druk op knop: led uit.

```
les_3_p10_toggle_led_met_knop_blocking

void setup() {
  pinMode(13, OUTPUT);      // led output
  pinMode(11, INPUT_PULLUP); // knop input & pull-up
}

void loop() {

  while (digitalRead(11) != HIGH) {} // wacht op knop los

  while (digitalRead(11) != LOW) {}  // wacht op knop ingedrukt

  digitalWrite(13, HIGH);           // LED aan

  while (digitalRead(11) != HIGH) {} // wacht op knop los

  while (digitalRead(11) != LOW) {}  // wacht op knop ingedrukt

  digitalWrite(13, LOW);            // LED uit

}
```

Toestanden (states)

Toestand: waar zijn we mee bezig.

```
les_3_p10_toggle_led_met_knop_blocking

void setup() {
  pinMode(13, OUTPUT);      // led output
  pinMode(11, INPUT_PULLUP); // knop input & pull-up
}

void loop() {

  while (digitalRead(11) != HIGH) {} // wacht op knop los

  while (digitalRead(11) != LOW) {}  // wacht op knop ingedrukt

  digitalWrite(13, HIGH);           // LED aan

  while (digitalRead(11) != HIGH) {} // wacht op knop los

  while (digitalRead(11) != LOW) {}  // wacht op knop ingedrukt

  digitalWrite(13, LOW);            // LED uit
}
```

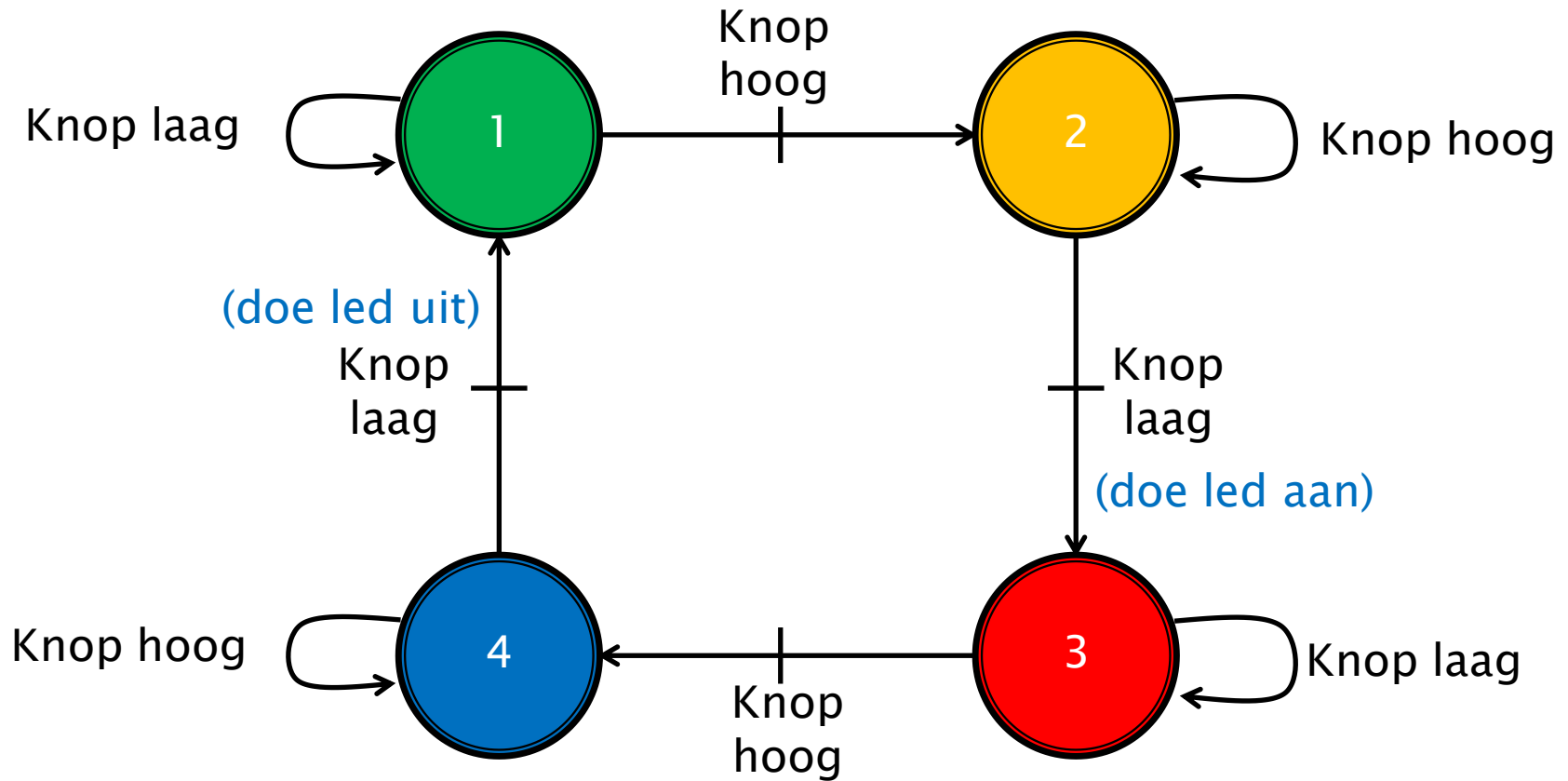
1

2

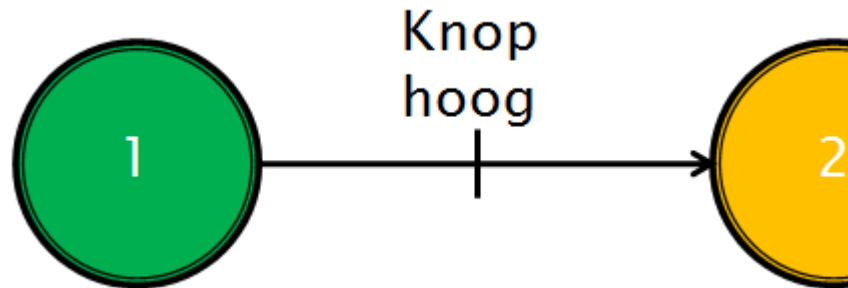
3

4

Toestanden (states)



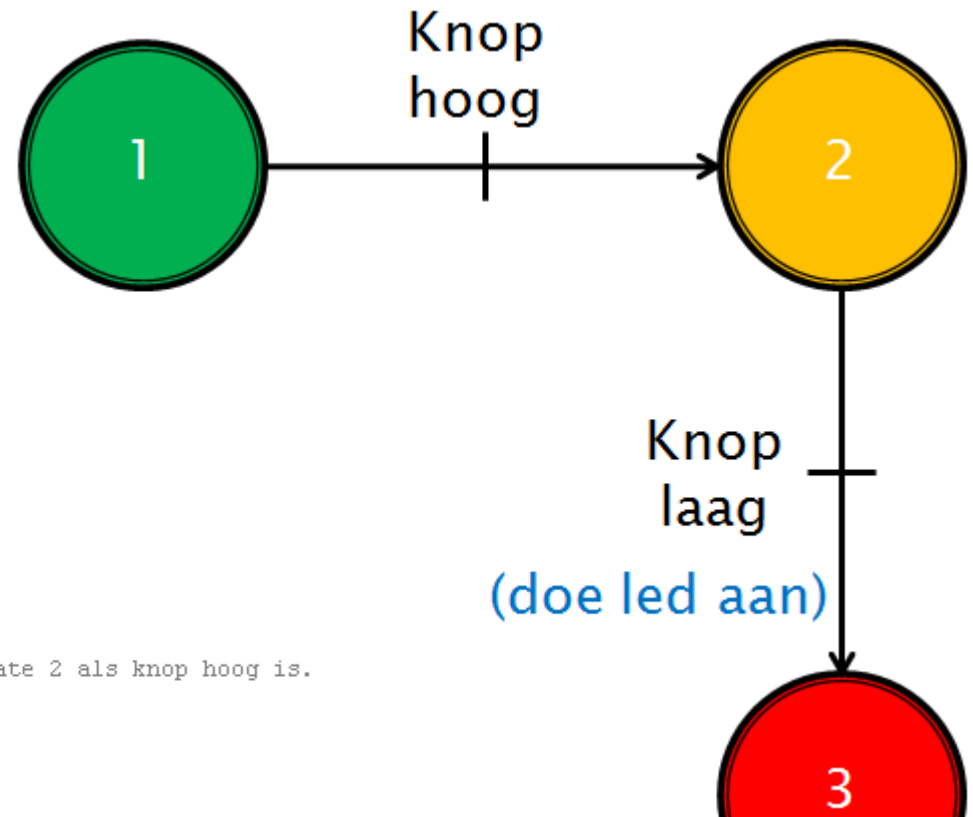
Eerste state



```
int State = 1;
void loop() {
  delay(10);

  if (State == 1) {
    if (digitalRead(11) == HIGH) {
      State = 2;          // ga naar state 2 als knop hoog is.
    }
  }
}
```

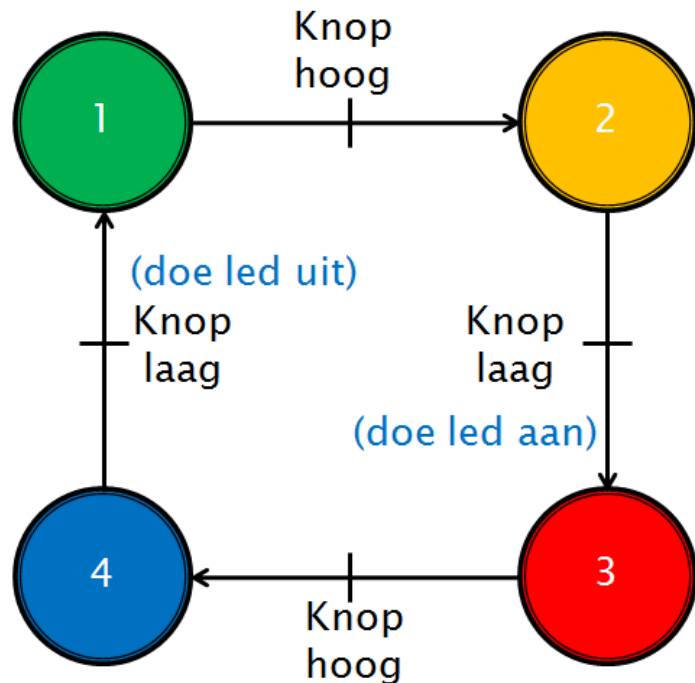
Tweede state



```
int State = 1;
void loop() {
  delay(10);

  if (State == 1) {
    if (digitalRead(11) == HIGH) {
      State = 2;          // ga naar state 2 als knop hoog is.
    }
  }
  if (State == 2) {
    if (digitalRead(11) == LOW) {
      State = 3;          // ga naar state 3 als knop laag is.
      digitalWrite(13, 1); // en zet de led aan
    }
  }
}
```

Oefening – state 3 en 4



les_3_p26_states

```
void setup() {
  pinMode(13, OUTPUT);          // led output
  pinMode(11, INPUT_PULLUP);    // knop input & pull-up

  // start serial
  Serial.begin(115200);
  Serial.println("Hallo arduino wereld.");
}

int State = 1;
void loop() {
  delay(10);

  Serial.println(State);

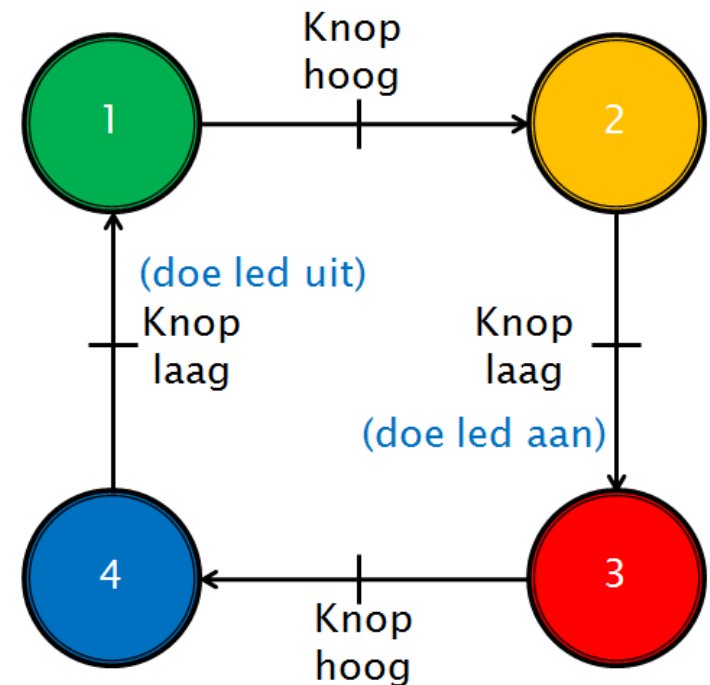
  if (State == 1) {
    if (digitalRead(11) == HIGH) {
      State = 2;                // ga naar state 2 als knop hoog is.
    }
  }
  if (State == 2) {
    if (digitalRead(11) == LOW) {
      State = 3;                // ga naar state 3 als knop laag is.
      digitalWrite(13, 1);      // en zet de led aan
    }
  }
}
```


Oefening – het resultaat

```
int State = 1;
void loop() {
  delay(10);

  Serial.println(State);

  if (State == 1) {
    if (digitalRead(11) == HIGH) {
      State = 2;          // ga naar state 2 als knop hoog is.
    }
  }
  if (State == 2) {
    if (digitalRead(11) == LOW) {
      State = 3;          // ga naar state 3 als knop laag is.
      digitalWrite(13, 1); // en zet de led aan
    }
  }
  if (State == 3) {
    if (digitalRead(11) == HIGH) {
      State = 4;          // ga naar state 4 als knop hoog is.
    }
  }
  if (State == 4) {
    if (digitalRead(11) == LOW) {
      State = 1;          // ga naar state 1 als knop laag is.
      digitalWrite(13, 0); // en zet de led uit
    }
  }
}
```

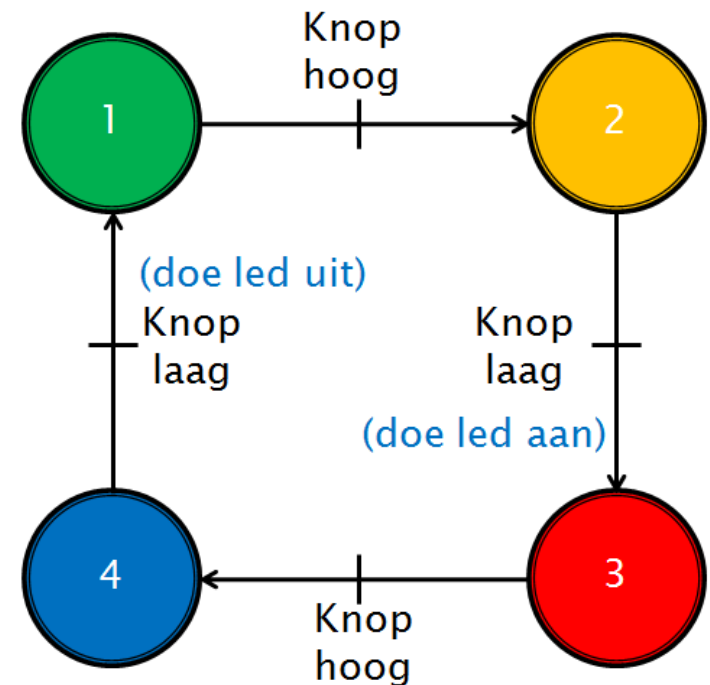


Oefening – het resultaat

```
int State = 1;
void loop() {
  delay(10);

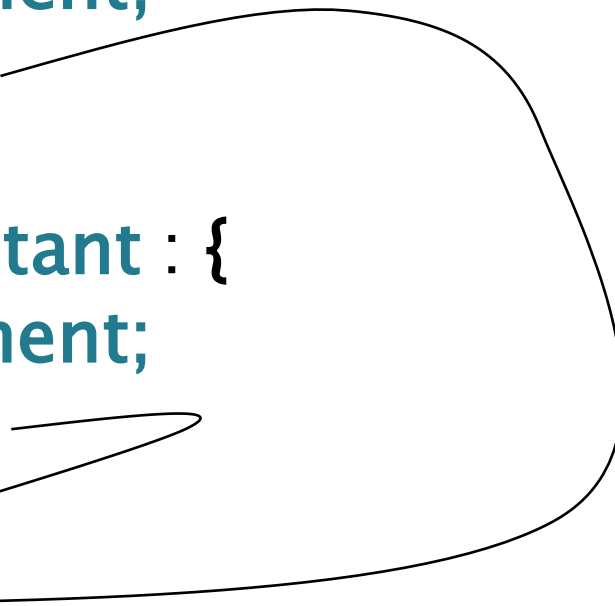
  Serial.println(State);

  if (State == 1) {
    if (digitalRead(11) == HIGH) {
      State = 2;          // ga naar state 2 als knop hoog is.
    }
  }
  if (State == 2) {
    if (digitalRead(11) == LOW) {
      State = 3;          // ga naar state 3 als knop laag is.
      digitalWrite(13, 1); // en zet de led aan
    }
  }
  if (State == 3) {
    if (digitalRead(11) == HIGH) {
      State = 4;          // ga naar state 4 als knop hoog is.
    }
  }
  if (State == 4) {
    if (digitalRead(11) == LOW) {
      State = 1;          // ga naar state 1 als knop laag is.
      digitalWrite(13, 0); // en zet de led uit
    }
  }
}
```



switch / case

```
switch ( expression ) {  
    case constant : {  
        statement;  
        break;  
    }  
    case constant : {  
        statement;  
        break;  
    }  
}
```



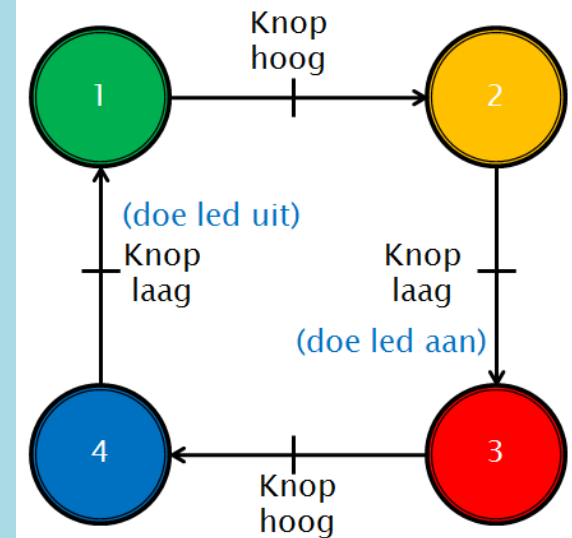
The diagram illustrates the execution flow of a switch statement. Two curved arrows originate from the 'break;' statements within the case blocks and point to the closing curly brace of the switch statement, indicating that the execution exits the switch block upon reaching a break statement.

switch / case – voorbeeld

```
int State = 1;

void loop() {

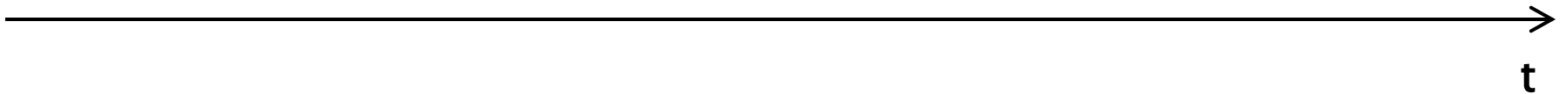
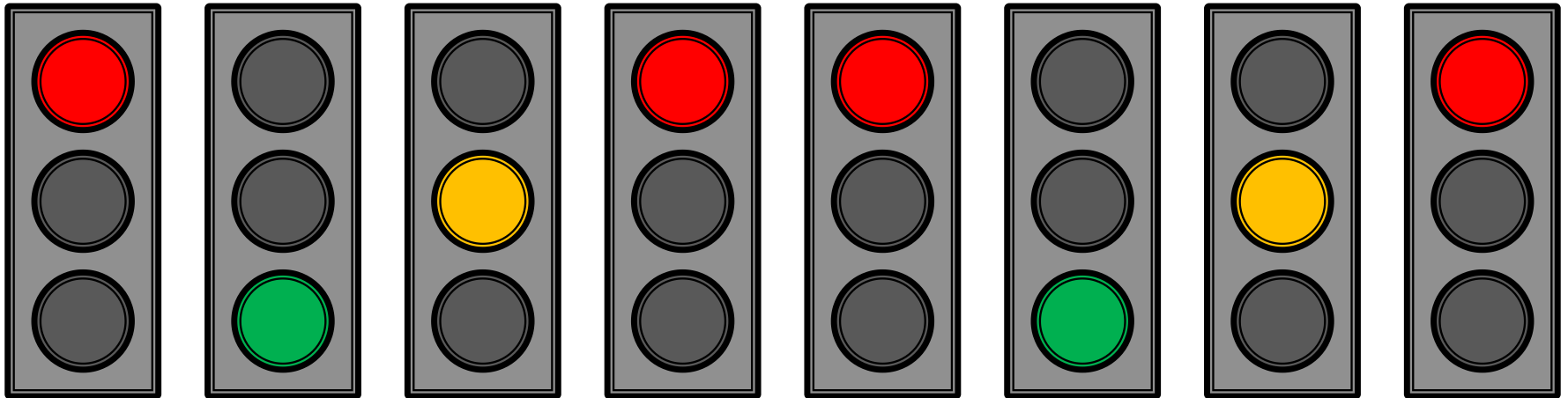
  switch(State) {
    case 1 : { // wacht op knop los
      if (digitalRead(11) == HIGH) {
        State = 2;
      }
      break;
    }
    case 2 : { // wacht op knop ingedrukt
      if (digitalRead(11) == LOW) {
        digitalWrite(13, HIGH); // LED aan
        State = 3;
      }
      break;
    }
    case 3 : { // wacht op knop los
      if (digitalRead(11) == HIGH) {
        State = 4;
      }
      break;
    }
  }
```



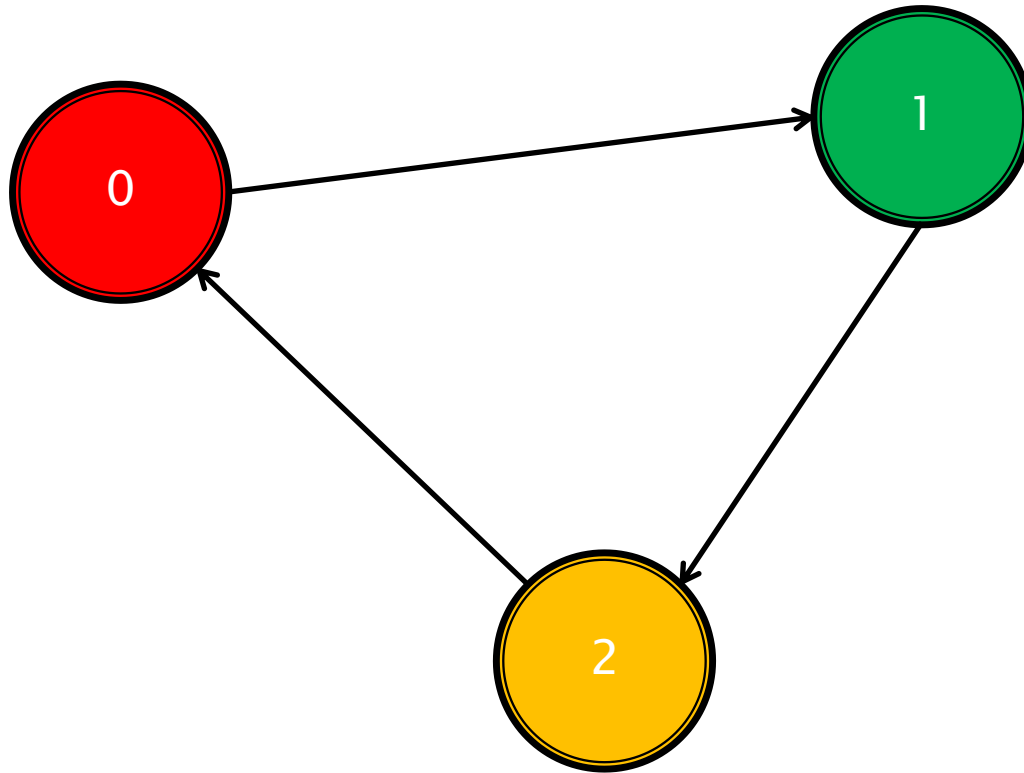
```
    case 4 : { // wacht op knop ingedrukt
      if (digitalRead(11) == LOW) {
        digitalWrite(13, LOW); // LED uit
        State = 1;
      }
      break;
    }
  }

  default : {
    State = 1;
    break;
  }
} // einde van switch
}
```

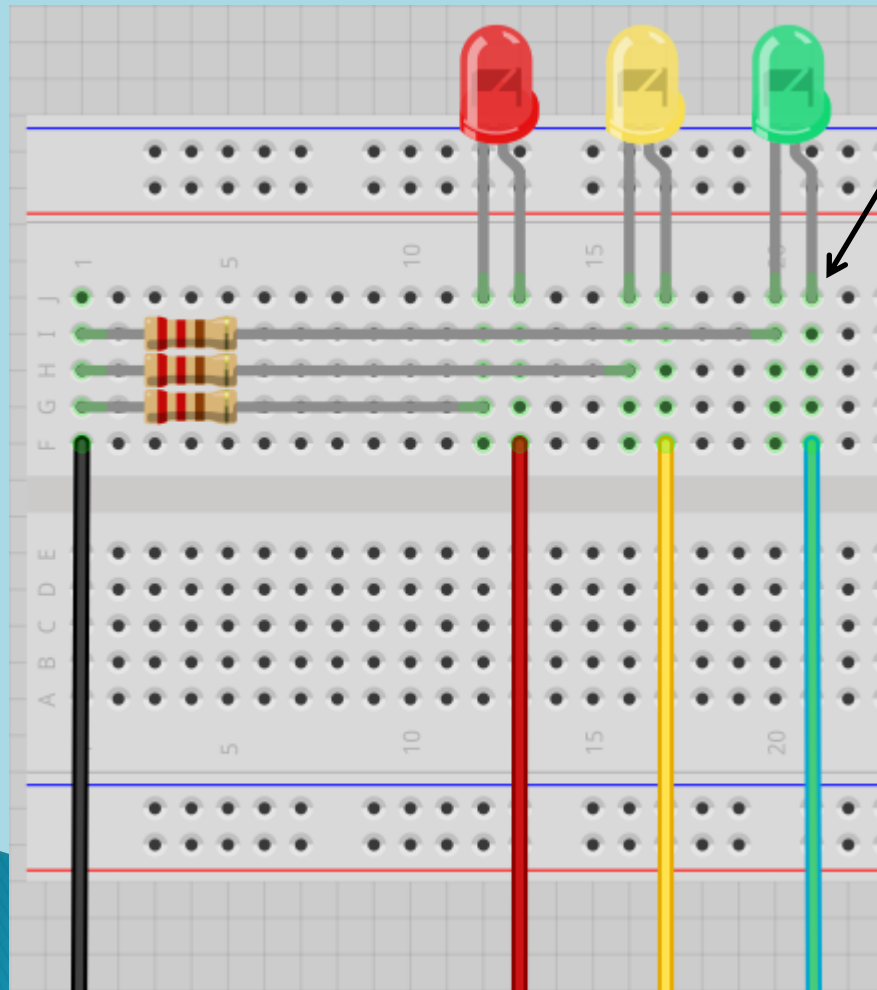
Verkeerslicht



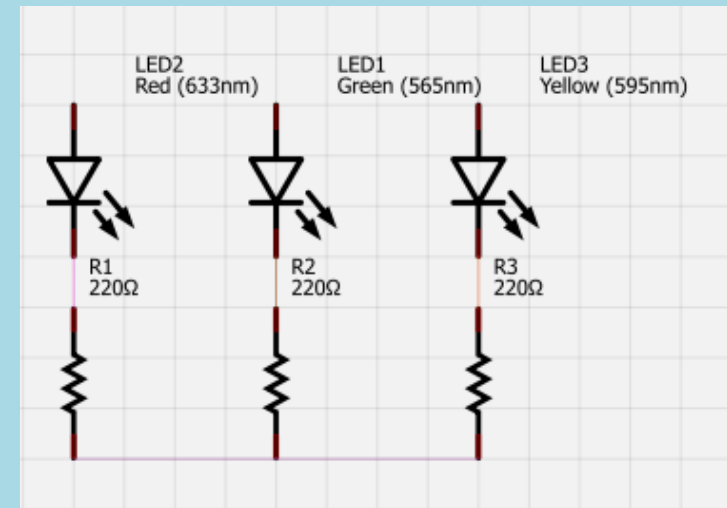
Verkeerslicht (states)



Oefening – verkeerslicht



Lange pin rechts



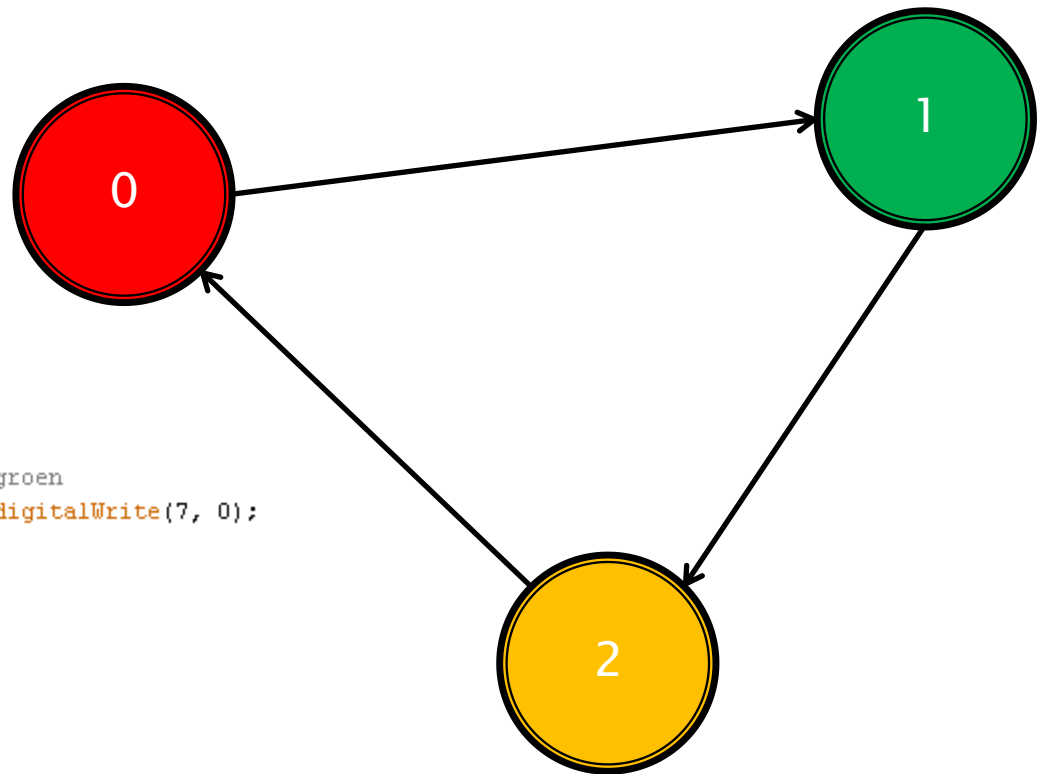
les_3_p35_led_test

```
void setup() {  
  pinMode(5, OUTPUT);           // rode led  
  pinMode(6, OUTPUT);           // gele led  
  pinMode(7, OUTPUT);           // groene led  
  
  // start serial  
  Serial.begin(115200);  
  Serial.println("Hallo arduino wereld.");  
}
```


Oefening – verkeerslicht 2

les_3_p40_stoplicht_1

```
void setup() {  
  pinMode(5, OUTPUT);    // rode led  
  pinMode(6, OUTPUT);    // gele led  
  pinMode(7, OUTPUT);    // groene led  
  
  // start serial  
  Serial.begin(115200);  
  Serial.println("Hallo arduino wereld.");  
}  
  
void loop() {  
  delay(100);  
  
  StoplichtTakt();  
}  
  
void StoplichtTakt()  
{ static int State = 0;  
  
  switch(State) {  
    case 0 : { // rood  
      // rood          geel          groen  
      digitalWrite(5, 1); digitalWrite(6, 0); digitalWrite(7, 0);  
      State = 1;  
      break;  
    }  
  }  
}
```

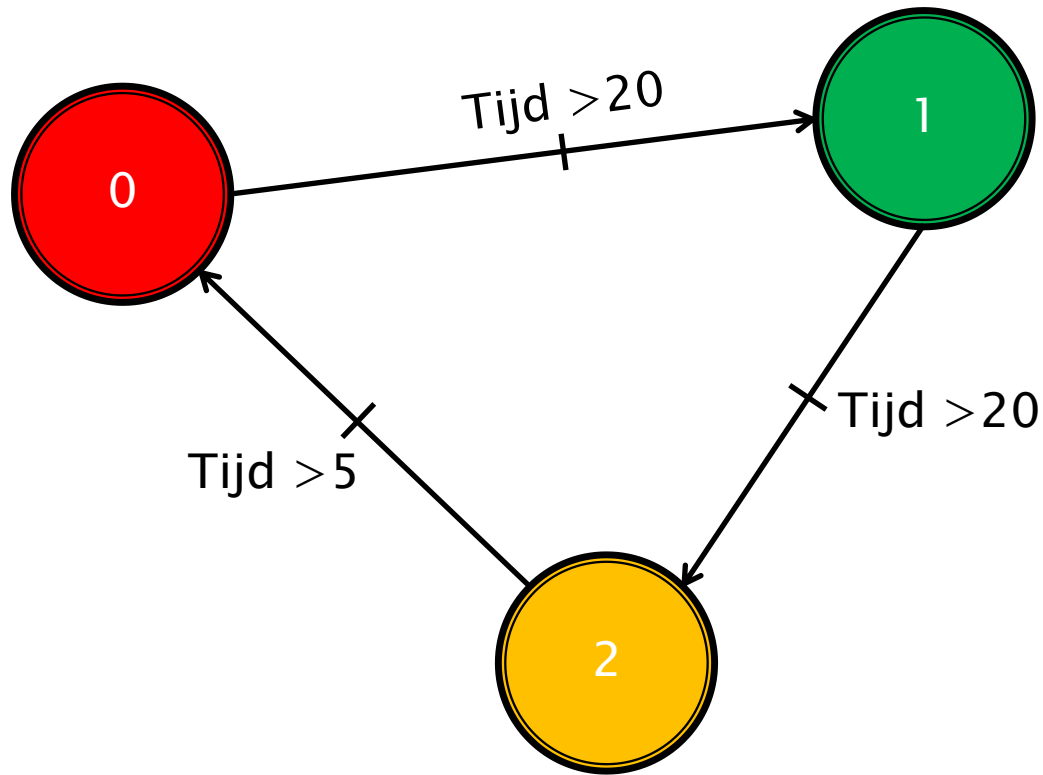


Oefening – resultaat

```
void StoplichtTakt()
{ static int State = 0;

  switch(State) {
    case 0 : { // rood
      // rood          geel          groen
      digitalWrite(5, 1); digitalWrite(6, 0); digitalWrite(7, 0);
      State = 1;
      break;
    }
    case 1 : { // groen
      // rood          geel          groen
      digitalWrite(5, 0); digitalWrite(6, 0); digitalWrite(7, 1);
      State = 2;
      break;
    }
    case 2 : { // geel
      // rood          geel          groen
      digitalWrite(5, 0); digitalWrite(6, 1); digitalWrite(7, 0);
      State = 0;
      break;
    }
    default : {
      Serial.print("Ongeldige state ");
      Serial.println(State);
      State = 0;
      break;
    }
  } // einde van switch
}
```

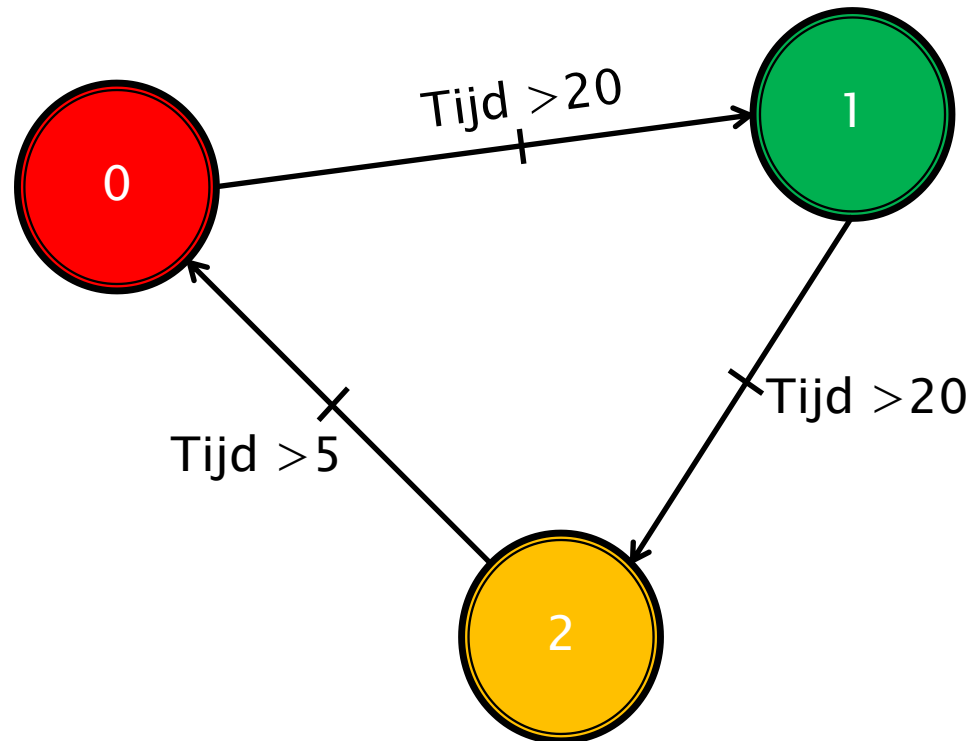
Verkeerslicht (tijd)



Oefening – verkeerslicht3

- Voeg wachttijd toe aan besturing voor het verkeerslicht.

```
void StoplichtTakt()  
{ static int State = 0;  
  static int Tijd;  
  
  Tijd = Tijd + 1;  
  Serial.println(Tijd);  
  
  switch(State) {  
    case 0 : { // rood licht  
      // rood      geel      groe  
      digitalWrite(5, 1); digitalWrite(6, 0); digi  
  
      if (Tijd > 20) {  
        Tijd = 0;  
        State = 1;  
      }  
      break;  
    }
```



Oefening – verkeerslicht3

les_3_p50_stoplicht_3

```
void setup() {
  pinMode(5, OUTPUT);    // rode led
  pinMode(6, OUTPUT);    // gele led
  pinMode(7, OUTPUT);    // groene led

  // start serial
  Serial.begin(115200);
  Serial.println("Hallo arduino wereld.");
}

void loop() {
  delay(100);

  StoplichtTakt();
}

void StoplichtTakt()
{ static int State = 0;
  static int Tijd;

  Tijd = Tijd + 1;
  Serial.println(Tijd);

  switch(State) {
    case 0 : { // rood licht
      // rood      geel      groen
      digitalWrite(5, 1); digitalWrite(6, 0); digitalWrite(7, 0);

      if (Tijd > 20) {
        Tijd = 0;
        State = 1;
      }
      break;
    }
  }
```

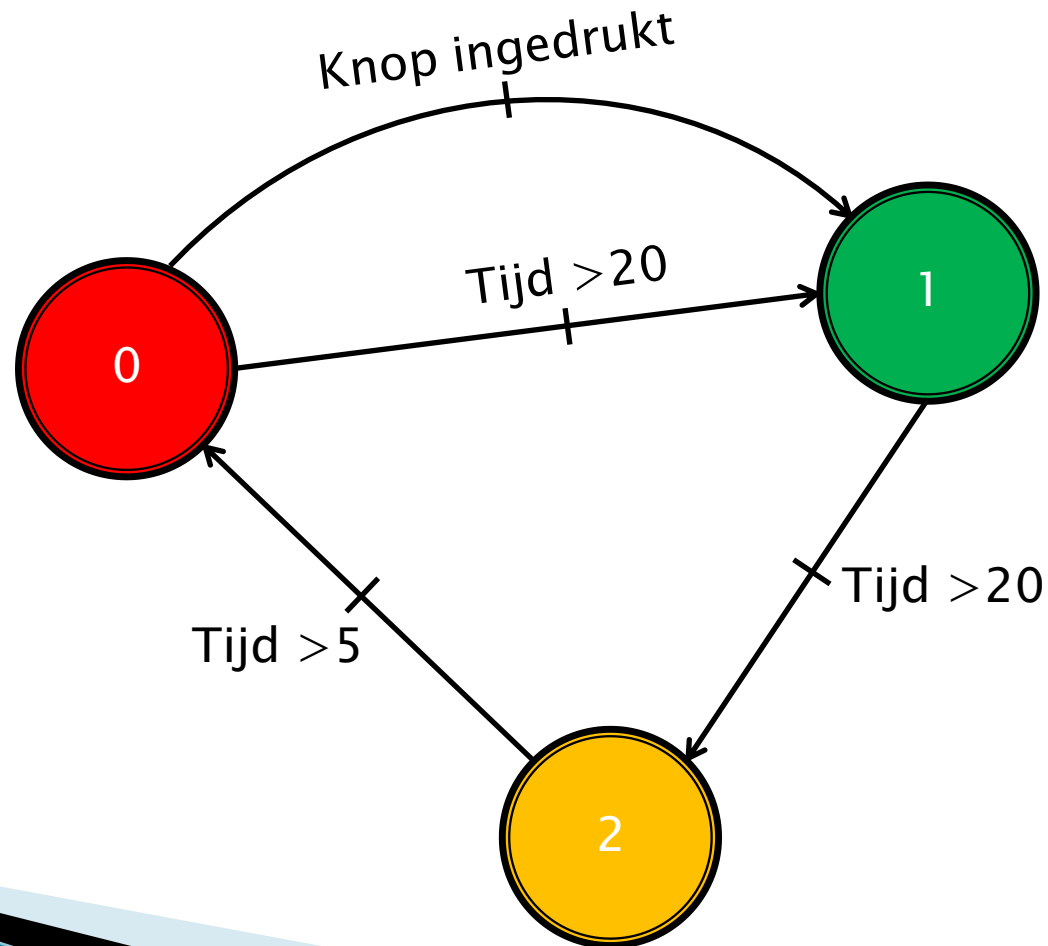
```
    case 1 : { // groen licht
      // rood      geel      groen
      digitalWrite(5, 0); digitalWrite(6, 0); digitalWrite(7, 1);
      if (Tijd > 20) {
        Tijd = 0;
        State = 2;
      }
      break;
    }
    case 2 : { // geel / oranje licht
      // rood      geel      groen
      digitalWrite(5, 0); digitalWrite(6, 1); digitalWrite(7, 0);
      if (Tijd > 5) {
        Tijd = 0;
        State = 0;
      }
      break;
    }
    default : {
      Serial.print("Ongeldige state ");
      Serial.println(State);
      Tijd = 0;
      State = 0;
      break;
    }
  } // einde van switch
}
```

Voor thuis

- ▶ 1. Bestudeer 'flank detectie' patroon.

Voor thuis

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



Voor thuis...

► 3. 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
}
```

Voor thuis...

- ▶ 4. Bonus: Rateltikker voor blinden.
 - Gebruik led op D13
 - Groen licht = snel knipperen
 - Oranje, Rood licht = langzaam knipperen

Tips:

- Maak losse functie voor het knipperen van de led
- Roep deze functie aan in loop()
- Knippersnelheid afhankelijk van globale variabele
- Waarde globale variabele instellen in StoplichtTakt()