

# CAR – Patronen

C, Arduino & Robots

# 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 int VorigeKnop;  
int 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();  
    }  
}
```

# Vaste interval (1)

(voor een code block in een functie)

Variabele om volgende moment te onthouden

Tijd 1x uitlezen en bewaren

Bereken tijdverschil & test resultaat

Bepaal volgend moment

```
static int NextTakt;  
int ms = millis();  
if ((ms - NextTakt) > 0) {  
    NextTakt = ms + 10; // zet tijd voor volgende interval  
  
    // hier de periodieke acties voor deze interval  
}
```

Acties

# Vaste interval (2)

## (begin van een functie)

Variabele om volgende moment te onthouden

Bereken tijdverschil & test resultaat

Tijd1x uitlezen en bewaren

```
static int NextTakt;  
int ms = millis();  
if ((ms - NextTakt) < 0) return;  
NextTakt = ms + 10; // zet tijd voor volgende interval  
  
// hier de acties
```

Bepaal volgend moment

Acties

Verlaat functie als tijd nog niet voorbij is

# State machine (deel 1)

Variabele om huidige toestand te onthouden, eerste keer automatisch 0

Variabele om vorige toestand te onthouden, eerste keer ongelijk aan State

Vlag die aangeeft of we voor de eerste keer in deze toestand (terug) zijn.

```
void StateMachineTakt()  
{  
    static int State, PrevState = -1;  
    bool NewState = false;  
  
    // rapporteer status bij state overgang  
    if (PrevState != State) {  
        PrevState = State;  
        printf("StateMachineTakt State: %d\n", State);  
        NewState = true;  
    }  
}
```

State gewijzigd?

Onthoud state voor volgende keer

Meld nieuwe state

Zet vlag voor gebruik in states

# State machine (deel 2)

Switch naar actieve state

```
// state machine
switch(State) {
    case 0 : // State: stil staan
    {
        ...
        break;

    case 1 : // State: wandvolgen
    {
        ...
        break;

    default :
    {
        printf("StateMachineTakt: ongeldige state %d\n", State);
        State = 0;
        break;
    }
} // einde van switch
}
```

Twee states  
(zie volgende slide)

Eindig case altijd  
met break;

Meld ongeldige state.

Default voor  
afvangen van  
ongeldige states

Ga terug naar  
start-state

Markeer sluit-accolade switch

# State machine (deel 3)

Case met state omschrijving  
als commentaar

Gebruik 'NewState' voor code die eenmalig,  
'OnEntry', uitgevoerd moet worden

Acties, het  
echte werk  
van deze state

1 of meer condities  
voor state overgangen.

Omschrijving conditie

Conditie

'OnExit' actie (per conditie)

Altijd break;

Nieuwe state (met omschrijving)

```
case 0 : // State: stil staan
{
    if (NewState) {
        // OnEntry actie
        Teller = 10;
    }

    // State actie
    Teller--;
    digitalWrite(13, true); // led aan

    // Conditie
    if (Teller <= 0) { // Tijd voorbij?
        digitalWrite(13, false); // led uit
        State = 1; // wandvolgen
    }
    break;
}
```