

# Actuators

- Uitgangen
- Geluid
- Weergave
- Motor

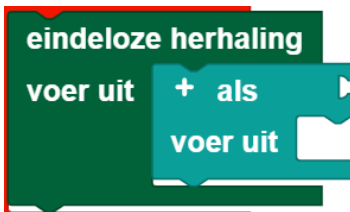
# Uitgangen

## Het blok Start elke keer

Het blok **Start elke keer** biedt de mogelijkheid om een programma te laten draaien, wanneer aan een voorwaarde is voldaan. Dit werkt dus net als een verschil tussen gevallen, maar wordt tijdens het gehele verloop van het programma niet slechts eenmaal doorlopen, maar elke keer wanneer aan de voorwaarde wordt voldaan. Het blok **Start elke keer**:



is een afkorting voor onderstaande constructie:



Je kunt in het blok **Start elke keer** van de categorie Uitgangen alle voorwaarden uit juist deze categorie gebruiken.

**Aanwijzing:** Het programmagedeelte van het blok **Start elke keer** moet kort worden gehouden en mag geen blokkerende openingshandelingen of eindeloze loops bevatten, zodat dit deel van het programma snel kan worden afgewerkt.

## LED's



## Instellen

Met de blokken **stel LED ... in** en **stel LED-helderheid ... in** kun je de LED aan- en uitzetten respectievelijk hun helderheid op een bepaalde waarde (van 0 tot 512) instellen.

## Openen

Met het blok **haal LED-helderheid** kan de helderheid van een LED geopend en als waarde verder verwerkt worden.

## Opvragen

Met de blokken **is LED ...** en **is LED-helderheid ...** kun je de activiteit respectievelijk de helderheid van een LED als voorwaarde gebruiken. In het voorbeeld hebben wij de helderheid van de LED op 512 ingesteld, voor zover zij deze helderheid niet al heeft.



## Motoren

Het symbool op de motorblokken is plaatsvervangend voor alle motoren, die geen encoder- of servomotoren zijn.

### Instellen

Met het blok **stel motortoerental in op [] ...** kun je het toerental van een motor op een bepaalde waarde (van 0 tot 512) instellen.

### Openen

Met het blok **haal motortoerental** kan het toerental van een motor geopend en als waarde verder verwerkt worden.

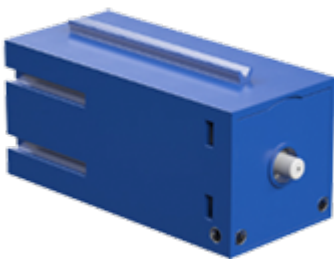
## Opvragen

Met de blokken **draait motor** en **is motortoerental ...** kun je de activiteit respectievelijk het toerental van een motor als voorwaarde gebruiken.

### Stoppen

Met het blok **stop motor ...** is het mogelijk om een motor te stoppen.

## Compressor



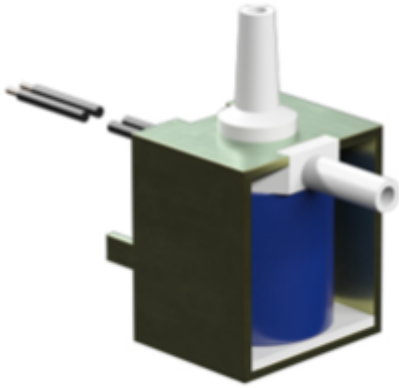
### Instellen

Met het blok **stel compressor [] in** kun je de compressor in- of uitschakelen.

## Opvragen

Met het blok **is compressor []** kun je de activiteit van een compressor als voorwaarde gebruiken.

# Magneetklep



## Instellen

Met het blok **stel magneetklep [] in** kun je de magneetklep in- of uitschakelen. Hier betekent "aan" dat de klep open staat en "uit", dat de klep gesloten is.

## Opvragen

Met het blok **is magneetklep []** kun je de activiteit van een magneetklep als voorwaarde gebruiken.

# Geluid

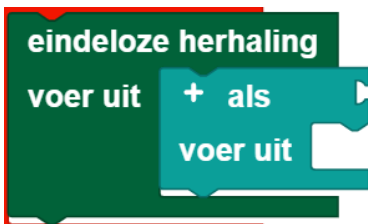
De TXT 4.0 Controller heeft een ingebouwde luidspreker en biedt daarmee de mogelijkheid om audiobestanden af te spelen.

## Het blok Start elke keer

Het blok **Start elke keer** biedt de mogelijkheid om een programma te laten draaien, wanneer aan een voorwaarde is voldaan. Dit werkt dus net als een verschil tussen gevallen, maar wordt tijdens het gehele verloop van het programma niet slechts eenmaal doorlopen, maar elke keer wanneer aan de voorwaarde wordt voldaan. Het blok **Start elke keer**:



is een afkorting voor onderstaande constructie:



Je kunt in het **blok Start elke keer** van de categorie Geluid alle voorwaarden uit juist deze categorie gebruiken.

**Aanwijzing:** Het programmagedeelte van het blok **Start elke keer** moet kort worden gehouden en mag geen blokkerende openingshandelingen of eindeloze loops bevatten, zodat dit deel van het programma snel kan worden afgewerkt.

## Afspelen

### Vooraf geïnstalleerde audiobestanden

Met het onderstaande blok kun je een van de 29 vooraf geïnstalleerde audiobestanden afspelen. Het gewenste audiobestand kun je via het dropdown-menu (klein driehoekje) kiezen. Het is bovendien mogelijk om het geluid in een permanente loop af te spelen. Daarvoor moet je een vinkje achter het symbool voor de permanente loop zetten.



### Eigen audiobestanden

Wanneer je een eigen geluidsfragment wilt afspelen kun je het blok



gebruiken. Om je eigen geluidsfragmenten in het blok in te leggen moet je:

1. met de controller verbonden zijn.
2. het IP-adres van de controller in de browser invoeren (hierbij moet het IP-adres worden gekozen dat ook voor het verbinden met de controller is gebruikt).
3. op de geopende pagina USER: ft, PASSWORD: fischertechnik invoeren.
4. de map geluidsfragmenten openen en daar via het plusteken het gewenste audiobestand naar de controller uploaden (belangrijk: het audiobestand moet beschikbaar zijn in het wav-formaat).
5. In het ROBO Pro Coding-blok onder het pad "./bestandsnaam.wav" aangeven.

Ook hier heb je de optie op het geluidsfragment in een permanente loop af te spelen.

## Opvragen

Om op te vragen of een audiobestand moet worden afgespeeld maak je gebruik van het blok **geef geluidsfragment weer**. Dit kan als voorwaarde in het programma worden gebruikt.

## Stoppen

Om een geluidsfragment te stoppen wordt eenvoudig het blok **stop geluidsfragment** in het programma gebruikt.

# Weergave

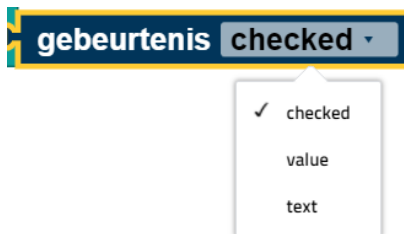
Met de blokken Categorie display kan het beeldscherm van de TXT 4.0 Controller vorm gegeven en bruikbaar gemaakt worden. Dit gebeurt in twee stappen:

1. Configureren, hetgeen wil zeggen dat
  - een nieuw bestand uit de Categorie display openen en via de pagina's Symbool met behulp van het plusteken linksboven
  - de gewenste elementen naar het raster trekken (dit geeft het configureerbare gedeelte van het display weer)
  - en zo nodig de specificaties aanpassen.
2. Programmeren, wil zeggen
  - in het hoofdprogramma met behulp van de blokken Categorie display de werking van interacties met het display programmeren.

## Blokken

### Opvragen van gebeurtenissen

Het blok **Gebeurtenis** [] vraagt de weergavewaarde van een element op. Dit blok kan alleen in de gebeurtenisprogramma's worden gebruikt. In deze gebeurtenisprogramma's heeft het blok automatisch betrekking op de gebeurtenis in het programma waarin het wordt gebruikt. Het geschikte type voor de weergavewaarden kan via het dropdown-menu (klein driehoekje) worden gekozen:



## Tekstveld

Met het element Tekstveld kun je een tekst op het beeldscherm plaatsen. Het symbool in de weergaveconfigurator is het etiket. Wanneer je dit symbool in het raster trekt, wordt rechts een venster geopend. Hier kan onder Inspecteur

- de grootte van het tekstveld in pixels,
- de positie van het tekstveld in pixels (de hoek van het tekstveld linksboven ligt op het aangegeven punt),
- de naam van het tekstveld en
- de inhoud van het tekstveld (deze tekst wordt bij het starten van het display weergegeven)

worden vastgelegd.

Met het blok **stel tekstveld Tekst ... in** kan de afgebeelde tekst in de loop van het programma worden gewijzigd.

## Invoeren

Het element **Invoer** maakt het mogelijk dat gebruikers via de controller tekst invoeren. Het bijbehorende symbool in de weergaveconfigurator is het "T"-teken. Wanneer je dit symbool in het raster trekt, wordt rechts een venster geopend. Hier kan onder Inspecteur

- de grootte van het invoerveld in pixels,
- de positie van het invoerveld in pixels (de hoek van het invoerveld linksboven ligt op het aangegeven punt),
- de naam van het invoerveld en
- de inhoud van het invoerveld (deze tekst wordt bij het starten van het display weergegeven)

worden vastgelegd.

Met het blok **stel invoerveld Tekst ... in** kan de afgebeelde tekst in de loop van het programma worden gewijzigd.

## Invoerprogramma

Het invoerprogramma draait wanneer een invoer is beëindigd. Dit wordt afzonderlijk van het hoofdprogramma geschreven. Variabelen werken over het geheel genomen via beide programma's. Het invoerprogramma draait in het blok **wanneer een invoer is beëindigd**. Het blok **Gebeurtenis []** wordt in het invoerprogramma op "Tekst" gezet. In dit voorbeeld wordt de variabele **Naam** op de ingevoerde tekst gezet, en wordt dan in het hoofdprogramma gebruikt om de ingevoerde tekst weer te geven:



## Meetinstrument

De meetinstrument-functie kan waarden (geen waarden lager dan 1) weergeven. Het bijbehorende symbool in de displayconfigurator is de schaalverdeling. Wanneer je dit symbool in het raster trekt, wordt rechts een venster geopend. Hier kan onder Inspecteur

- de grootte van het meetinstrument in pixels,
- de positie van het meetinstrument in pixels (de hoek van het meetkaliber linksboven ligt op het aangegeven punt),
- de naam van het meetinstrument,
- de uitlijning van het meetinstrument,
- het waardegebied dat door het meetinstrument wordt weergegeven, en
- de waarde van het meetinstrument, die bij het starten van het display wordt weergegeven

worden vastgelegd.

Met het blok **stel meetinstrument in op waarde ...** kan het meetinstrument op de ingevoerde waarden worden ingesteld. Deze waarde moet binnen het vooraf gedefinieerde waardebereik liggen. Wanneer de waarde buiten het waardebereik ligt wordt, afhankelijk van het feit of de waarde te hoog of te laag is, een van de grenzen van het waardebereik weergegeven.



# Statusregel

De statusindicator geeft de activiteit van iets aan. Afhankelijk van de status brandt deze ("actief") of doet dat niet ("niet actief"). Het symbool in de displayconfigurator is een brandende diode. Wanneer je dit symbool in het raster trekt, wordt rechts een venster geopend. Hier kan onder Inspecteur

- de grootte van de statusregel in pixels,
- de positie van de statusregel in pixels (de hoek van de statusindicator linksboven ligt op het aangegeven punt),
- de naam van de statusregel,
- de kleur van de statusregel en
- of de statusregel aan het begin "actief" of "niet actief" moet zijn,

worden vastgelegd.

Met het blok **stel statusregel actief [] in** kan de statusregel geactiveerd respectievelijk uitgeschakeld worden. In het dropdown-menu (klein driehoekje) kan worden gekozen of de statusregel op "actief" of "niet actief" moet worden ingesteld.

# Schuifregelaar

De schuifregelaar retourneert de waarden afhankelijk van zijn positie. De positie kan daarbij door de gebruiker met het touchscreen worden veranderd. De waarde kan met het blok **Gebeurtenis []** worden opgevraagd, zodra de schuifregelaar niet meer wordt bewogen. De opgevraagde waarde is een decimaal getal. Wil je dat de waarde van de schuifregelaar als heel getal wordt weergegeven, dan moet je het **afronden**-blok gebruiken. Het bijbehorende symbool voor de schuifregelaar is de streep met de cirkel. Wanneer je dit symbool in het raster trekt, wordt rechts een venster geopend. Hier kan onder Inspecteur

- de grootte van de schuifregelaar in pixels,
- de positie van de schuifregelaar in pixels (de hoek van de schuifregelaar linksboven ligt op het aangegeven punt)
- de naam van de schuifregelaar,
- de activiteit van de schuifregelaar,
- de uitlijning van de schuifregelaar,
- het waardebereik dat door de schuifregelaar wordt afgedekt en
- de waarde waarop de regelaar bij het starten van het display staat

worden vastgelegd.

Met het blok **stel de schuifregelaarwaarde ... in** kun je de schuifregelaar naar een andere waarde verschuiven.

Met **stel schuifregelaar geactiveerd [] in** kun je de activiteit via het dropdown-menu (klein driehoekje) omwisselen.

# Schuifregelaarprogramma

Het schuifregelaarprogramma draait, nadat de schuifregelaar werd verschoven. Dit wordt afzonderlijk van het hoofdprogramma geschreven. Variabelen werken over het geheel genomen via beide programma's. Het schuifregelaarprogramma draait in het blok **wanneer de schuifregelaar beweegt**. Het blok **Gebeurtenis []** wordt in schuifregelaarprogramma "value" gezet. In dit voorbeeld wordt het toerental van de motor met de schuifregelaar geregeld. De waarde van de schuifregelaar moet afgerond zijn, aangezien de motor alleen hele getallen als toerental accepteert:



## Button

De button is een veld met tekst, die kan worden ingedrukt. Wanneer je op de button drukt begint het buttonprogramma te draaien, zodra deze weer wordt losgelaten. Het bijbehorende symbool voor de button is het vierkant met daarop de tekst "OK". Wanneer je dit symbool in het raster trekt, wordt rechts een venster geopend. Hier kan onder Inspecteur

- de grootte van de button in pixels,
- de positie van de button in pixels (de hoek van de button linksboven ligt op het aangegeven punt),
- de naam van de button,
- de tekst die op de button staat en
- de activiteit van de button

worden vastgelegd.

Met **stel button geactiveerd** [] in kun je de activiteit via het dropdown-menu (klein driehoekje) omwisselen.

## Buttonprogramma

Het buttonprogramma begint te draaien zodra de button niet meer wordt ingedrukt. Dit wordt afzonderlijk van het hoofdprogramma geschreven. Variabelen werken over het geheel genomen via beide programma's. Het buttonprogramma draait in het blok **wanneer de button is ingedrukt**. Het blok **Gebeurtenis** [] kan niet in het buttonprogramma worden gebruikt, aangezien de button geen retourneringswaarde heeft. In dit voorbeeld wordt de LED geactiveerd wanneer de button wordt ingedrukt.



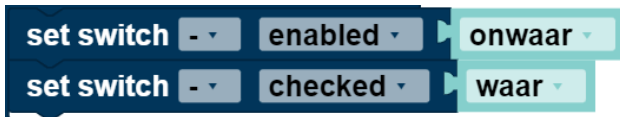
## Schakelaar

De schakelaar kan twee standen aannemen en staat altijd in precies een van deze beide standen. Afhankelijk van de stand geef deze **waar** of **onwaar** weer. Het bijbehorende symbool voor de schakelaar is het ovaal met de punt. Wanneer je dit symbool in het raster trekt, wordt rechts een venster geopend. Hier kan onder Inspecteur

- de grootte van de schakelaar in pixels,
- de positie van de schakelaar in pixels (de hoek van de schakelaar linksboven ligt op het aangegeven punt),
- de naam van de schakelaar,
- de tekst die naast de schakelaar staat,
- de activiteit van de schakelaar en
- de toestand waarin de schakelaar bij het starten van het programma moet staan

aangepast worden.

Het blok



neemt twee functies voor zijn rekening. Je kunt of de activiteit (enabled in het dropdown-menu selecteren) of de toestand (checked in het dropdown-menu selecteren) op **waar** of **onwaar** instellen.

## Schakelaarprogramma

Het schakelaarprogramma gaat elke keer draaien wanneer de schakelaar wordt omgezet. Dit wordt afzonderlijk van het hoofdprogramma geschreven. Variabelen werken over het geheel genomen via beide programma's. Het schakelaarprogramma draait in het blok **wanneer de schakelaar is omgezet**. Het blok **Gebeurtenis []** wordt in het schakelaarprogramma op "checked" ingesteld en geeft **waar** of **onwaar** weer. Dit voorbeeldprogramma schakelt de LED in wanneer de schakelaar is omgezet, in het andere geval wordt de LED uitgeschakeld:



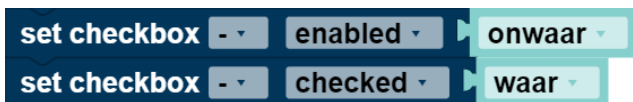
## Selectievakje

Het selectievakje kan twee standen aannemen en staat altijd in precies een van deze beide standen. Afhankelijk van de stand geeft deze **waar** of **onwaar** weer. Het symbool voor het selectievakje is het vierkant met het vinkje. Wanneer je dit symbool in het raster trekt, wordt rechts een venster geopend. Hier kan onder Inspecteur

- de grootte van het selectievakje in pixels,
- de positie van het selectievakje in pixels (de hoek van de schakelaar linksboven ligt op het aangegeven punt),
- de naam van het selectievakje,
- de tekst die naast het selectievakje staat,
- de activiteit van het selectievakje en
- de toestand waarin het selectievakje bij het starten van het programma moet staan

worden vastgelegd.

Het volgende blok neemt twee functies voor zijn rekening. Via het dropdown-menu (klein driehoekje) kan worden aangegeven, welke je gebruikt. Je kunt of de activiteit (enabled in het dropdown-menu selecteren) of de toestand (checked in het dropdown-menu selecteren) op **waar** of **onwaar** instellen.



## Selectievakjeprogramma

Het selectievakjeprogramma gaat elke keer draaien wanneer het selectievakje wordt aangeklikt. Dit wordt afzonderlijk van het hoofdprogramma geschreven. Variabelen werken over beide programma's. Het selectievakjeprogramma draait in het blok **wanneer het selectievakje wordt omgezet**. Het blok **Gebeurtenis []** wordt in het schakelaarprogramma op "checked" ingesteld en geeft **waar** of **onwaar** weer. Dit

voorbeeldprogramma schakelt de LED in wanneer een vinkje in het selectievakje wordt gezet, in het andere geval wordt de LED uitgeschakeld.



# Motor

## Het blok Start elke keer

Het blok **Start elke keer** biedt de mogelijkheid om een programma te laten draaien, wanneer aan een voorwaarde is voldaan. Dit werkt dus net als een verschil tussen gevallen, maar wordt tijdens het gehele verloop van het programma niet slechts eenmaal doorlopen, maar elke keer wanneer aan de voorwaarde wordt voldaan. Het blok **Start elke keer**:



is een afkorting voor onderstaande constructie:



Je kunt in het blok **Start elke keer** van de categorie Motor alle voorwaarden uit juist deze categorie gebruiken.

**Aanwijzing:** Het programmagedeelte van het blok **Start elke keer** moet kort worden gehouden en mag geen blokkerende openingshandelingen of eindeloze loops bevatten, zodat dit deel van het programma snel kan worden afgewerkt.

## Motor

Het symbool op de motorblokken is plaatsvervangend voor alle motoren, die geen encoder- of servomotoren zijn.

## Instellen

Met het blok **stel motortoerental in op [] ...** kun je het toerental van een motor op een bepaalde waarde (van 0 tot 512) instellen. Via het dropdown-menu (klein driehoekje) kan de draairichting worden geselecteerd.

## Openen

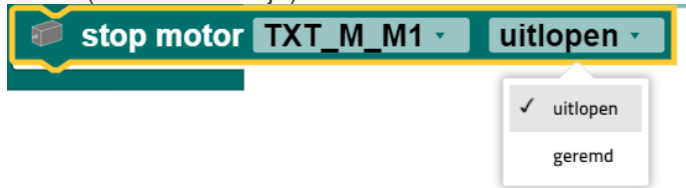
Met het blok **haal motortoerental** kan het toerental van een motor geopend en als waarde verder verwerkt worden.

## Opvragen

Met de blokken **draait motor** en **is motortoerental ...** kun je de activiteit respectievelijk het toerental van een motor als voorwaarde gebruiken.

## Stoppen

Met het blok **stop motor** [] is het mogelijk om een motor te stoppen. Daarbij biedt het blok **stop motor** [] de opties om een motor direct of geleidelijk tot stilstand te brengen. De gewenste optie kun je via het dropdown-menu (klein driehoekje) kiezen:



## Servomotor



### Instellen

Met het blok **stel stand op ... in** kun je de stand van een motor op een bepaalde waarde (van 0 tot 512) instellen. 0 en 512 zijn de waarden voor de maximale uitslag naar rechts en links. Bij de waarde 256 staat de servomotor dienovereenkomstig in het midden.

### Openen

Met het blok **open stand** kan de stand van een servomotor geopend en als waarde verder verwerkt worden.

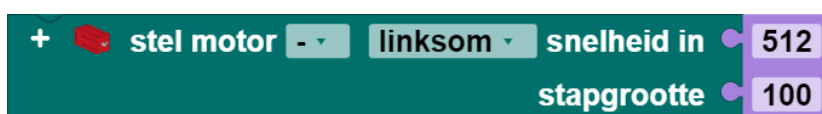
## Encodermotor



De encodermotor heeft dezelfde functie als een normale motor, maar heeft bovendien de mogelijkheid om het aantal omwentelingen te tellen en meerdere motoren synchroon aan te sturen. Een omwenteling wordt daarbij in ~64 stappen onderverdeeld.

### Instellen

Met het blok



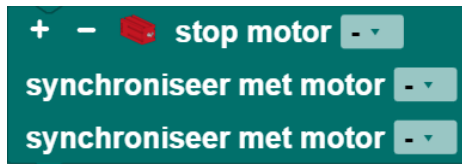
kan het toerental van een motor op een bepaalde waarde (van 0-512) worden ingesteld. Via het dropdown-menu (klein driehoekje) kan de draairichting worden geselecteerd. Bovendien kan het aantal stappen worden

ingevoerd, die de motor moet afleggen. In dit voorbeeld draait de motor 100 stappen, dus één en een derde omwenteling. Zoals in het voorbeeld is te zien heeft dit blok een plusteken, met behulp waarvan meerdere motoren synchroon kunnen worden aangestuurd. Het is mogelijk om motoren aan de hand van een master of een extensie te synchroniseren, een overkoepelende synchronisatie bijv. tussen motoren van de master en een extensie is niet mogelijk.

**Aanwijzing: Snel opeenvolgende synchronisatieprocessen, zoals bijv. door een loop mogelijk zijn (zie voorbeeld), kunnen de synchroniteit beïnvloeden of zelfs compleet verhinderen.**

## Stoppen

Met het blok **stop motor** ... wordt een motor gestopt. Wil je meerdere motoren gelijktijdig stoppen, dan kun je met het linker plusteken in het blok maximaal drie andere motoren toevoegen.



## Opvragen

Het blok **heeft de positie bereikt** wordt gebruikt om het bereiken van de positie als voorwaarde te gebruiken. Met positie wordt hier de eindpositie van een encodermotor na een volledig beëindigde stap bedoeld.