



programmeren met

flow.it

Inhoud

Inleiding.....	3
<i>Flow.it</i> configureren.....	4
Voor de eerste keer	7
Een project beginnen	9
Routines in <i>flow.it</i>	11
Verwijzingen in <i>flow.it</i>	12
Instructies in <i>flow.it</i>	13
De flowchart wijzigen.....	15
De flowchart of code verschuiven	16
Een flowchart printen of exporteren	17

Inleiding

Flow.it is educatieve software. Er bestaat veel software die voor leerlingen het coderen vereenvoudigt, maar de weg van functionele wens naar functionerende code wordt daarmee niet aangeboden. *Flow.it* is bedoeld om dit gat te dichten.

Met *flow.it* gaan het ontwerpen van een programma en het coderen ervan hand in hand. Terwijl leerlingen nadenken over wat er nodig is en wanneer wat moet gebeuren, zorgt *flow.it* ervoor dat er een bijbehorend raamwerk in c++ ontstaat. Met behulp van sjablonen kan *flow.it* worden geconfigureerd om met code-bibliotheken voor bijvoorbeeld actuatoren en sensoren samen te werken. *Flow.it* is bedoeld om programma's voor populaire micro-boards zoals de Arduino en de Raspberry te maken.

Flow.it volgt de programmeeromgeving van de Arduino-IDE. Het biedt twee vaste routines: de 'setup' en de 'loop'. Om de code geschikt te maken voor bijvoorbeeld de g++ compiler op de Raspberry, moet onderaan de code nog het volgende worden toegevoegd:

```
int main( int argc, char** argv)
{
    setup();
    while( true ) loop();
    return 0;
}
```

Overigens bestaat er een uitgebreide code-bibliotheek met sensoren en actuatoren, die dit automatisch doet: DeviceLib. Voor de Arduino werkt deze samen met de Arduino-IDE en op de Raspberry met de Geany-IDE. Flow-it is in wezen gemaakt als flowchart-interface voor de DeviceLib-bibliotheek, maar is eenvoudig met andere bibliotheken uit te breiden.

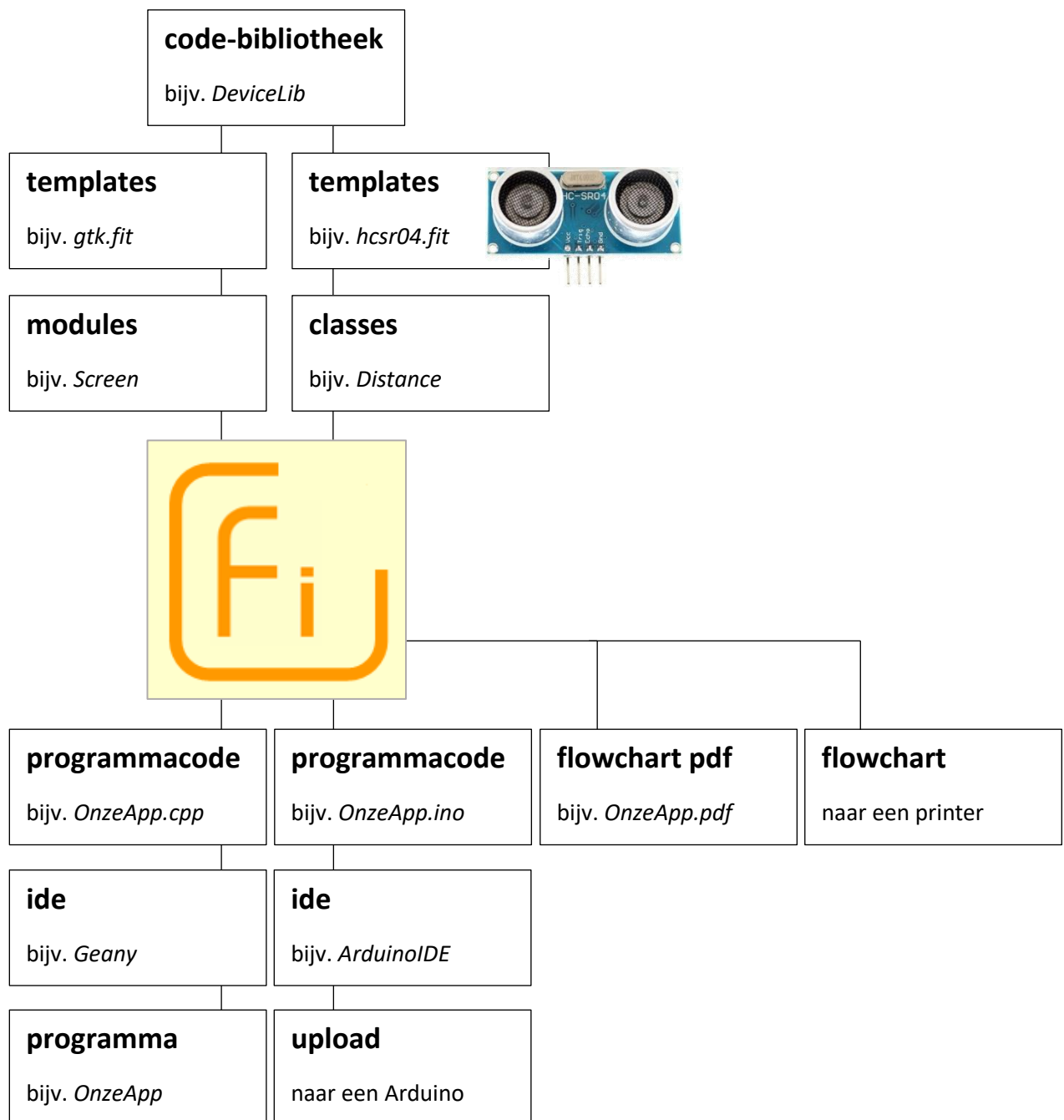
Tenslotte iets over het installeren van *flow.it*: Dit is niet nodig. Alle benodigde bestanden staan in de programma-map *FlowIt*¹. Je kunt de map *FlowIt* naar believen ergens op de harde schijf kopiëren². Je start *flow.it* vanuit de programma-map door te dubbelklikken op *FlowIt.exe* (Windows) dan wel *FlowIt* (Raspberry). Het is handig om een snelkoppeling op je bureaublad te plaatsen.

¹ Voor Windows staan ook de hulpprogramma's in de map *FlowIt*. Op een Raspberry zijn de benodigde hulpbestanden om *flow.it* te starten standaard aanwezig. Wanneer dit niet zo blijkt te zijn, kan het systeem als volgt worden gerepareerd. Start de verkenner vanaf de commandoprompt met *sudo pcmanfm* om superuser-rechten te krijgen. In de map *FlowIt* vind je een submap *usr_lib_arm-linux-gnueabi*. Verplaats de bestanden uit deze map naar de map */usr/lib/arm-linux-gnueabi*.

² Onder Windows kun je *flow.it* ook vanaf een usb starten. Linux staat dit niet toe.

Overzicht

Flow.it is afhankelijk van bestaande hulpcode. Over het algemeen vind je deze code terug in code-bibliotheken. Op basis van de te gebruiken hulpcode worden template-bestanden voor *flow.it* gemaakt, die de modules en classes ervan beschrijven. Hoe dit gaat, vind je terug in de paragraaf “*Flow.it* configureren”. Met behulp van de templates levert *flow.it* programmacode voor programmeeromgevingen (ide's) die er programma's voor de Arduino of de Raspberry van maken.



Flow.it configureren

Deze paragraaf is alleen van belang voor wie zelfgemaakte of andere bibliotheken aan *flow.it* wil toevoegen. Wie dat niet wil: *Flow.it* is voorgeconfigureerd voor de bibliotheek DeviceLib. Download DeviceLib van www.github.com/BM4GIT/DeviceLib, installeer de bibliotheek en ga door naar de volgende paragraaf.

Hoewel *flow.it* als zelfstandig programma kan werken, zal het toevoegen van code-bibliotheken zorgen voor meer programmeer-gemak. *Flow.it* kan via template-bestanden worden geconfigureerd om met code-bibliotheken samen te werken. Standaard worden er templates voor de DeviceLib-bibliotheek meegeleverd. Door deze bibliotheek voor de Arduino-IDE of op de Raspberry te installeren, kun je met *flow.it* programma's maken die van een reeks aan sensoren en actuatoren gebruik maakt.

Flow.it is object-georiënteerd. De templates zijn gebaseerd op zogenaamde classes. Het aanmaken van een template is vrij eenvoudig. In een template-bestand maak je aan *flow.it* bekend, welke class je toevoegt, in welk header-bestand deze te vinden is en of er een using-regel nodig is om met de routines van de class te kunnen werken. Verder moeten de te gebruiken routines van de class worden gedeclareerd.

Volg het onderstaande voorbeeld uit de DeviceLib-bibliotheek om zelf een template voor *flow.it* te maken.

Bekijk de inhoud van het header-bestand:

```
// header file: hcsr04.h

namespace HCSR04
{

class Distance : public Sensor
{
public:
    Distance();
    ~Distance();
    void setPin( uint8_t trigger, uint8_t echo);
    void read();
    uint16_t cm();
    uint16_t inch();
private:
    int8_t      m_trigger;
    int8_t      m_echo;
    int8_t      m_cm;
};

}
```

Op basis van dit header-bestand, wordt het volgende sjabloon met bijv. 'Kladblok' toegevoegd:

```
<template>
<type>sensor</type>
<include>hcsr04.h</include>
<using>HCSR04</using>
<class>Distance</class>
<function>setPin( 10, 11)</function>
<function>read( )</function>
<function>dataReady( )</function>
<function>cm( )</function>
<function>inch( )</function>
</template>
```

Bij het item `<type>` mogen drie types worden gebruikt: 'sensor', 'actuator' en 'storage'. Dit is overigens alleen van invloed op de flowchart-afbeelding en niet op de code. In het voorbeeld zijn de parameters van een routine vervangen door standaardwaardes om aan te geven wat voor soort parameter er wordt verlangd. Bij een goed gedocumenteerde bibliotheek, kunnen leerlingen dit ook in de documentatie terugvinden. In plaats van standaardwaardes kun je er voor kiezen om de parameters in woorden weer te geven kloppend met de documentatie. In het geval van de `setPin`-routine, kan de betreffende regel ook zo worden beschreven:

```
<function>setPin( trigger, echo)</function>
```

Bij het maken van de flowchart krijgt de leerling het volgende te zien en moet zelf de parameters vervangen door de juiste waardes:

```
setPin( 10, 11)
of
setPin( trigger, echo)
```

Sla het nieuwe template-bestand op met de uitgang '.fit' (*flow.it* template) in de map *Templates* van de programma-map van *flow.it*.

Het kan zijn dat een code-bibliotheek functies bevat, die niet in een class zijn ondergebracht. *Flow.it* verwacht dat deze functies in de template als module worden beschreven. In dat geval komen de items `<type>` en `<class>` te vervallen en moet de template het item `<module>` bevatten.

Bijvoorbeeld ziet de scherm-module van de bibliotheek DeviceLib er als volgt uit:

```
<template>
<module>Screen</module>
<include>ArduinoGtk.h</include>
<function>setTitle( "Naam")</function>
<function>setSize( 800, 600)</function>
<function>setPosition( 10, 10)</function>
<function>setFullScreen( true)</function>
<function>setPaperColor( 255, 255, 255, 255)</function>
<function>create( tButton, "ID", "Naam")</function>
...
</template>
```

Voor de eerste keer

Wanneer je *flow.it* voor het eerst start, krijg je een korte uitleg over hoe je met *flow.it* werkt.

The screenshot shows the FlowIt application window. On the left, a 'Project openen' (Open Project) dialog box is open, showing a file explorer with 'My Computer' and 'Gebruiker' (User) folders. The 'File name' field is empty, and the 'Files of type' is set to 'FlowItProjecten (*.fip)'. The main window displays a flowchart editor with a toolbar at the top containing icons for file operations and a 'setup loop' button. The flowchart is divided into three panels: 'VERWUZZING' (Flowchart) on the left, 'routines en verwijzingen' (Routines and References) in the middle, and 'code' (Code) on the right. The 'VERWUZZING' panel contains a vertical stack of flowchart elements: a start node, a decision diamond (ALS), a loop diamond (ZOLANG), a repeat diamond (HERHAAL), a wait oval (WACHT), and a do oval (DOEN). The 'routines en verwijzingen' panel shows a 'setup' routine with a 'eind' (end) node. The 'code' panel shows the code for the 'setup' and 'loop' routines. Annotations in Dutch explain the interface: 'eerste scherm --- scherm daarna' (first screen --- screen after) points to the 'Project openen' dialog; 'begin aan een project exporteer de flowchart of de code verschuif de flowchart of de code declareer een actuator of sensor' (start a project export the flowchart or the code move the flowchart or the code declare an actuator or sensor) points to the toolbar; 'klik op het rondje naast een declaratie en sleep naar een rondje in de flowchart' (click on the circle next to a declaration and drag to a circle in the flowchart) points to the 'eind' node in the 'routines en verwijzingen' panel; 'code flowchart declaraties' (code flowchart declarations) points to the 'code' panel.

eerste scherm --- scherm daarna

begin aan een project
exporteer de flowchart of de code
verschuif de flowchart of de code
declareer een actuator of sensor

routines en verwijzingen

klik op het rondje naast een declaratie en sleep naar een rondje in de flowchart

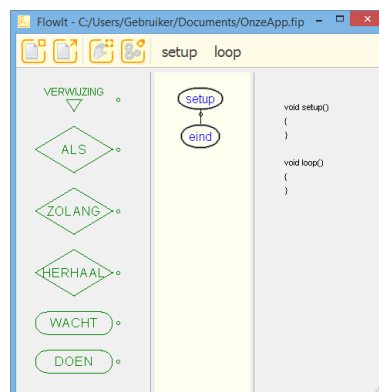
code
flowchart
declaraties

Klik op de enig beschikbare knop (de project-knop):



De eerste keer dat je op de knop drukt, moet je in het bestanden-venster bij *File name* een naam voor je project invullen. Druk op *Open* om aan je project te beginnen. Vanaf nu start *flow.it* automatisch op in het project waaraan je voor het laatst hebt gewerkt. Alleen als je een nieuw project wilt beginnen of van project wilt veranderen druk je weer op de project-knop.

Het project-scherm ziet er als volgt uit:



Je zult tevergeefs zoeken naar een knop om je project op te slaan. *Flow.it* slaat je project automatisch na iedere handeling op. Er gaan dan ook geen gegevens verloren wanneer de computer of *flow.it* er plotseling mee stopt (geen programma is volmaakt) en je kunt het zelf ook niet per ongeluk verkeerd doen bij het opslaan (geen mens is volmaakt).

Een project beginnen

Een project begint met het uitwerken van een plan. Wat wil je maken, wat heb je daarvoor nodig en hoe ga je het aanpakken? Stel dat je een zelfrijdende kar wilt maken die automatisch stopt als er iets voor komt en weer verder rijdt als de route vrij is. Je kunt zo'n kar maken met een motor en een afstandsensor. In *flow.it* wordt het declareren van de motor en afstandsensor dan de eerste actie.

Om een actuator (bijv. een motor), een sensor (bijv. een afstandsensor) of een interface (bijv. naar een database) toe te voegen aan je programma klik je op de declaratie-knop:



Flow-it toont vervolgens het declaratie-venster:

In het declaratie-venster krijg je de keus uit volgende opties:

1. Een routine declareren.
2. De declaratie van een routine verwijderen.
3. Een variabele declareren.
4. De declaratie van een variabele verwijderen.
5. Een onderdeel (zoals een actuator, sensor) declareren.
6. De declaratie van een onderdeel verwijderen.

Ad 1. Vul bij het veld 'Routine' de naam voor de routine in. Geef de routine een begrijpelijke naam waaraan je herkent wat hij doet. De naam van een routine mag uit letters, cijfers en het '_'-teken bestaan. Spaties of speciale tekens (zoals %, (, \$ of +) zijn niet toegestaan. De naam moet beginnen met een letter.

Ad 2. Kies in het veld 'Routine' de routine die je wilt verwijderen.

Ad 3. Vul bij het veld 'Variabele' de naam voor de variabele in. Geef de variabele een begrijpelijke naam waaraan je herkent waarvoor hij bestemd is. De regels voor een correcte naam zijn hetzelfde als bij een routine. Verder moet je nog het type van de variabele kiezen. Het type bepaalt welk soort informatie je in een variabele bewaart (bijvoorbeeld een getal of een tekst).

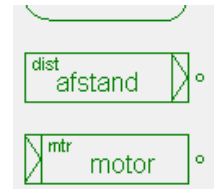
Ad 4. Kies in het veld 'Variabele' de variabele die je wilt verwijderen.

Ad 5. Vul bij het veld 'Naam' de naam voor het onderdeel in. Geef het onderdeel een begrijpelijke naam waaraan je herkent wat het doet. Kies bij het 'Type' of je een actuator, sensor of interface wilt toevoegen. Daarna kies je bij het veld 'Onderdeel' welke class of module je uit de code-bibliotheek (zoals DeviceLib) gaat gebruiken. Je vindt in de beschrijving van de bibliotheek welke class of module geschikt is voor je doel. Wanneer je onderdeel een class blijkt te zijn, is het nodig om bij het veld

'Declaratie' een declaratie-naam in te vullen die *flow.it* in de code moet gebruiken. De regels voor een correcte declaratie-naam zijn hetzelfde als bij een routine. Van sommige apparaten bestaan verschillende versies, zodat de code-bibliotheek moet weten met welke versie je aan de slag gaat. In dat geval moet je nog bij het veld 'Using' een using-regel opgeven. Deze regel is in de documentatie van de bibliotheek terug te vinden.

Ad 6. Kies in het veld 'Naam' het onderdeel dat je wilt verwijderen.

Wanneer de benodigde gegevens van het declaratie-venster zijn ingevuld, klik je op de knop 'Doen'. Een nieuwe routine zal dan op de pagina-balk en in de code (rechter kolom) worden weergegeven. Een nieuwe sensor, actuator of interface vind je in de declaratie-kolom (linker kolom, zie afbeelding hiernaast) terug. Een nieuwe variabele verschijnt alleen in de code.

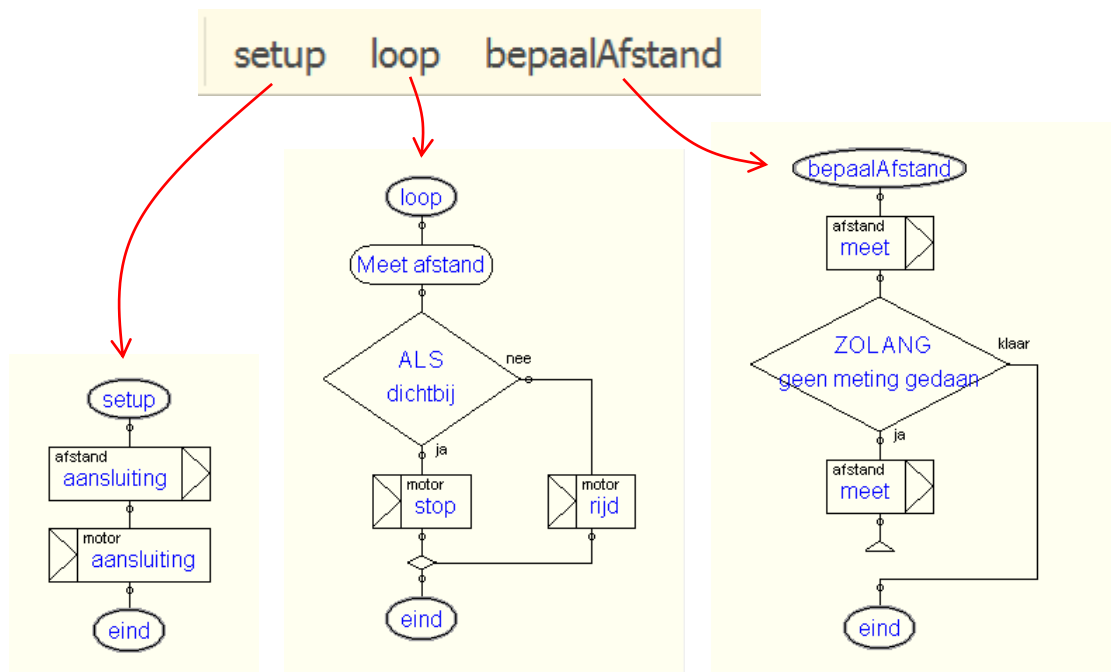


Let op! Wanneer je heel veel apparaten declareert, zal de lijst langer worden dan het scherm. Dit is geen probleem. Met de toetsen 'pijl omhoog' en 'pijl omlaag' scroll je de lijst naar boven en naar beneden. Je kunt ook de toetsen 'home' en 'end' gebruiken om naar het begin en einde van de lijst te springen.

Routines in *flow.it*

Een programma is onderverdeeld in routines. Twee routines zijn altijd aanwezig in *flow.it*: de 'setup' en de 'loop' (spreek uit: *loep* – dit is engels voor 'herhalen'). Verder kun je routines toevoegen en verwijderen. Dit is in de vorige paragraaf besproken.

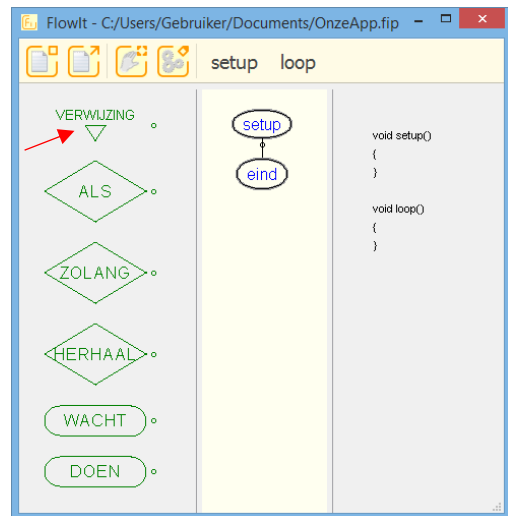
Door op één van de routines te klikken, verschijnt de bijbehorende flowchart in de flowchart-kolom (middelste kolom). De code-kolom (rechter kolom) bevat altijd de volledige code van het hele programma.



Verwijzingen in *flow.it*

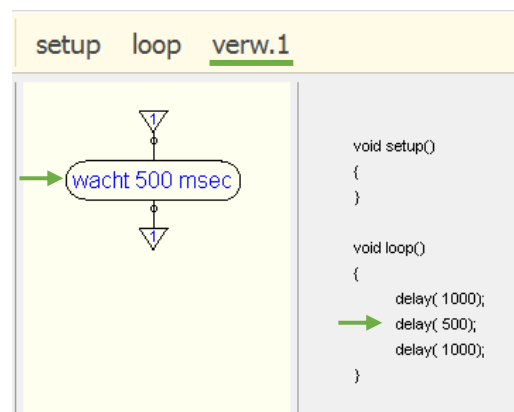
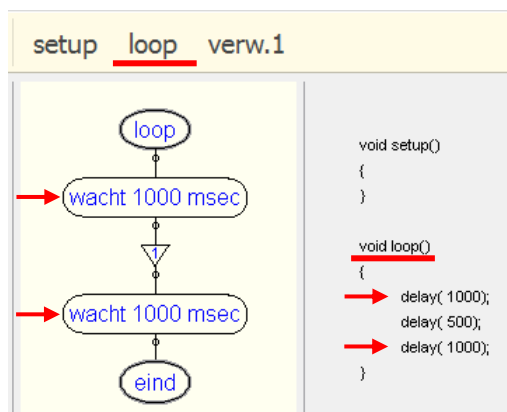
In *flow.it* is de flowchart-kolom (middelste kolom) lichtgeel van kleur. Deze geeft een pagina op A4-formaat weer. Naar gelang de flowchart in de lengte en breedte groeit, past de flowchart-kolom zich aan. Als de flowchart langer of breder wordt dan een A4-tje zullen er delen uit het gele gebied lopen en over de grijze achtergrond worden afgebeeld. Deze delen zijn bij het afdrukken of exporteren naar pdf niet zichtbaar.

Om te voorkomen dat een flowchart te lang of te breed wordt, kun je verwijzingen invoegen. Bovenaan de declaratie-kolom (linker kolom) vind je de declaratie voor verwijzingen. Verwijzingen worden automatisch genummerd. Net als de routines worden verwijzingen op de pagina-balk geplaatst en ook net als bij de routines wordt de bijbehorende flowchart op een aparte pagina weergegeven.



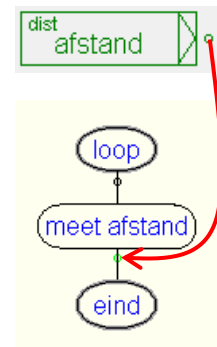
Wel is er een belangrijk verschil tussen een routine en een verwijzing. Een routine wordt als zodanig in de code apart geplaatst, terwijl je de code van een verwijzing-pagina op de plaats van de verwijzing terugvindt.

Let in het onderstaande voorbeeld op positie van de code 'delay(500)' in de code-kolom. Deze instructie hoort bij 'wacht 500 msec' in de flowchart en staat op een verwijzing-pagina. De verwijzing staat tussen beide keren 'wacht 1000 msec', waarbij de code 'delay(1000)' hoort.



Instructies in *flow.it*

Aan een flowchart voeg je instructies toe door de bijbehorende declaratie vanuit de declaratie-balk naar de plek te slepen waar je de instructie wilt invoegen. Als je bijvoorbeeld een instructie met de afstand-sensor wilt toevoegen, dan versleep je de declaratie daarvan. Hierbij zijn de kleine knooppjes (cirkeltjes) naast de declaraties en in de lijnen van de flowchart van belang. Door op een knooppje naast een declaratie te klikken, pak je de betreffende declaratie op. Houd de muisknop ingedrukt en versleep de declaratie naar een knoop in de flowchart. Wanneer de knoop in de flowchart oplicht, kun je de muisknop loslaten om de instructie daar in te voegen.



Bij het loslaten van de muisknop verschijnt het instructie-venster. Hoewel qua uiterlijk meestal hetzelfde, past het venster zich aan de versleepte declaratie aan. Wanneer je een sensor, actuator of interface hebt versleept, vind je bij het veld 'Functie' een lijst met functies die je ter beschikking staan voor het betreffende onderdeel. Bij de ALS- en ZOLANG-instructie kun je bovendien bij het veld 'Van declaratie' kiezen voor welk onderdeel de instructie geldt. De lijst met functies past zich dan aan het onderdeel aan. Bij de DOEN-instructie vind je bij het veld onder 'Algemene code invoeren' een lijst met de routines en variabelen die je eerder eens hebt toegevoegd. De instructies kunnen in het tekstvak naar behoeven worden aangepast. Hieronder vind je een voorbeeld om bij een ZOLANG-instructie de afstand te checken. Eerst wordt de 'cm()' -functie geselecteerd en daarna aangevuld met '< 50'. De check wordt dan 'cm() < 50'.

Tekst in flowchart:
dichtbij

Code-tekst:

☒ Een functie van een apparaat aanroepen

Van declaratie: dist

Functie: setPin(10, 11)
setPin(10, 11)
read()
dataReady()
cm()
inch()

☐ Algemene code invoeren

OK

Tekst in flowchart:
dichtbij

Code-tekst:

☒ Een functie van een apparaat aanroepen

Van declaratie: dist

Functie: cm() < 50

☐ Algemene code invoeren

OK

Je kunt ook het 'niet'-teken van c++ (uitroepteken) gebruiken, zoals hieronder bij '!dataReady()'.

The image shows two identical dialog boxes side-by-side, used for configuring a flowchart step. Each dialog has two main sections: 'Tekst in flowchart:' and 'Code-tekst:'.
In the left dialog:
- 'Tekst in flowchart:' contains the text 'meting gedaan'.
- 'Code-tekst:' has the radio button 'Een functie van een apparaat aanroepen' selected. Below it, 'Van declaratie' is set to 'dist' and 'Functie' is set to 'dataReady()'.
- There is an unselected radio button 'Algemene code invoegen' with an empty text field below it.
- An 'OK' button is at the bottom right.
In the right dialog:
- 'Tekst in flowchart:' contains the text 'geen meting gedaan'.
- 'Code-tekst:' has the same radio button selection and 'Van declaratie' set to 'dist'. However, 'Functie' is set to '!dataReady()'. A red arrow points to the exclamation mark in this dropdown.
- The 'Algemene code invoegen' option and its text field are also present.
- An 'OK' button is at the bottom right.

De HERHAAL- en WACHT-instructie hebben ieder hun eigen invulvenster.

The image shows two separate input dialog boxes.
The left dialog is for the 'HERHAAL' (Repeat) instruction. It features a numeric spinner set to '10' followed by the unit 'keer'. Below the spinner are two buttons: 'Vul in' (Input) and 'Annuleer' (Cancel).
The right dialog is for the 'WACHT' (Wait) instruction. It features a numeric spinner set to '1000' followed by the unit 'msec'. Below the spinner are two buttons: 'Vul in' (Input) and 'Annuleer' (Cancel).

De flowchart wijzigen

Een flowchart wijzig je door met de muis over een instructie te bewegen. De rand van de instructie licht dan op. Om de instructie te verwijderen druk je op de 'Delete'- of 'Backspace'-toets (ook wel met 'Del' of '←' aangeduid). Om de instructie te verplaatsen, versleep je hem naar een knoepje in de flowchart. Om de instructie te wijzigen klik je erop, waarna het instructie-venster verschijnt.

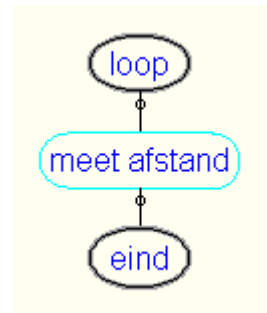
Let op! In *flow.it* bestaan de ALS-, ZOLANG-, TOTDAT- en HERHAAL-instructies uit wat we compound statements noemen. In de flowchart begint een compound statement bij de 'ja' en 'nee'-lijn en eindigt bij een klein ruitje of driehoekje. In de code staat een compound statement tussen accolades: { ... }. Wanneer je een ALS-, ZOLANG-, TOTDAT- of HERHAAL-instructie verwijdert, gaan ook alle instructies van het bijbehorend compound statement verloren.

Routines kun je niet verplaatsen, maar je kunt ze op dezelfde manier verwijderen als een instructie. De 'setup'- en 'loop'-routine kun je echter niet verwijderen.

Let op! Als je een routine verwijdert, gaat de hele pagina met instructies verloren. Hetzelfde geldt voor de verwijzingen.

Tenslotte is het mogelijk om ook een declaratie uit de declaratie-kolom (linker kolom) te verwijderen. Hier is het niet genoeg om met de muis over de declaratie te bewegen. De rand licht pas op als je op de declaratie klikt, waarna de declaratie is geselecteerd. Vervolgens druk je op de 'Delete'- of 'Backspace'-toets (ook wel met 'Del' of '←' aangeduid). Zolang een declaratie nog in de instructies van de flowchart wordt gebruikt, kun je hem niet verwijderen.

NB. Routines en declaratie kun je net als variabelen ook via het declaratie-venster verwijderen. Dit is in de paragraaf 'Een project beginnen' beschreven.

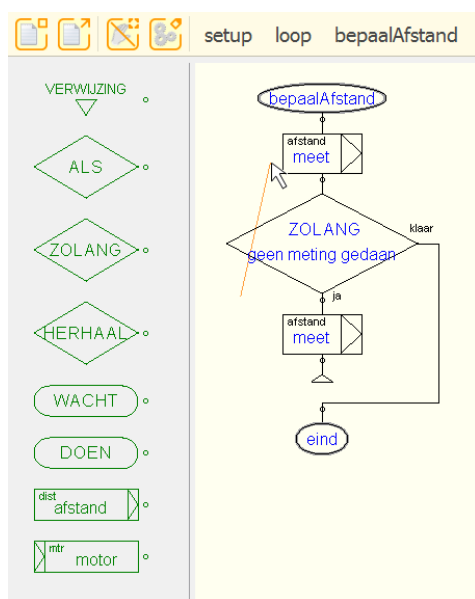


De flowchart of code verschuiven

Flowcharts die een heel A4-tje beslaan en ook de langere code, passen niet helemaal op het scherm. Met de verschuif-knop (links) kun je het gewenste deel van de flowchart of code in beeld schuiven. Om het verschuiven te stoppen klik je op de stop-knop (rechts).



Je verschuift het beeld met slepen. Een oranje lijn geeft de richting en afstand van het verschuiven aan. In het voorbeeld hieronder zal de flowchart schuin naar boven worden verplaatst.



Om de flowchart en de code weer normaal in beeld te krijgen, klik je ergens in de kolom met declaraties (linker kolom).


NB. De lijst met declaraties in de linker kolom kun je omhoog of omlaag scrollen. Gebruik daarvoor de toetsen 'pijl omhoog' en 'pijl omlaag'.

Een flowchart printen of exporteren

Voor het printen of exporteren klik je op de exporteer-knop:



Hiermee roep je het exporteer-scherm op:

Met de twee bovenste opties (voor Arduino of Raspberry exporteren) kun je de code uit de code-kolom opslaan met de extensie 'ino' of 'cpp'. De derde optie slaat de flowchart op als pdf-bestand. Klik op de -knop om een andere bestand-locatie te kiezen. De naam van het bestand is altijd gelijk aan de naam van het *flow.it*-project. Gebruik de vierde optie om de flowchart uit te printen. Kies één van de beschikbare printers.

Let op: De Arduino-IDE verwacht dat een programmabestand in een map met dezelfde naam als het bestand wordt geplaatst. Hier zorgt *flow.it* automatisch voor. Ook wanneer je code voor de Raspberry exporteert, doet *flow.it* dit automatisch. Bij het compileren en bouwen van het programma zul je namelijk de objectbestanden het liefst bij elkaar in één map houden. Voor de Raspberry vraagt *flow.it* bovendien of er een Geany-project moet worden aangemaakt. Geany is de standaard programmeeromgeving op de Raspberry.