# Robotkamp

## Programmeren FC6





Versie: 2

Datum: 18/02/2016

**Auteur: Frank Marchal** 

Doelgroep: 11 – 14 jaar

Inleiding:

Frank Marchal EDULAB



Dit document heeft tot doel de deelnemer van de **"bouw je eigen robot**" sessie of robotkamp te helpen bij het **programmeren** van de **robotkamp robot**. Na de mechanica dag en elektronica dag is het dus nu aan de **programmeerdag**.

De robot heeft deze code nodig om uiteindelijk iets nuttig te kunnen doen (bv. LEDs laten knipperen, schakelaars inlezen, geluid maken, rijden op een zwarte lijn...).

De robot wordt geprogrammeerd via een **USB kabel**, die verbonden is met een PC. Op de PC gaan we leren gebruik maken van de **Flowcode software**. De code komt terecht in de microcontroller.



Heb je achteraf nog vragen, dan kan je mij altijd bereiken op info@edulab.be

1. Korte kennismaking met een microcontroller

Frank Marchal EDULAB V2

Een microcontroller is te vergelijken met een **kleine computer**. Wanneer je deze niet vertelt wat hij moet doen, dan zal hij gewoon weg ook niets doen.

De microcontroller wordt dus via USB geprogrammeerd.

We kunnen aan de microcontroller signalen aanbieden (druk op de knop, LDR, lijnvolgers). Dit noemen we de **INPUTS**. Deze inputs worden door de controller afgevraagd en afhankelijk van het programma dat we hebben geschreven wordt er iets nuttig mee gedaan.

→ Zo werkt een koffiemachine bijvoorbeeld pas als je op de startknop drukt, wanneer er een tas is voorzien om de koffie in te doen, wanneer er water en koffiebonen aanwezig zijn in de machine. In dit geval moeten we 4 inputs hebben voordat de koffie kan gemaakt worden.



Frank Marchal EDULAB Een verwarmingselement, een motor die bonen maalt, een klep die opent en de koffie naar buiten laat komen zijn dus allemaal voorbeelden van OUTPUTS.

→ Bij onze robotkamp robot werkt dit net zo.



We kunnen deze info ook als volgt voorstellen:



#### The PIC microcontroller

De robot heeft als **inputs** de schakelaars, LDR, lijnvolgers.

De robot verwerkt alles in de microchip microcontroller PIC18F4455.

De robot heeft als **outputs** de LEDs, motoren en zoemer.

#### Hoe denkt nu een microcontroller?

Een microcontroller is een elektronische bouwblok die denkt, net als een computer. Hij werkt zijn instructies (wat hij moet uitvoeren) **stap per stap** af.

Bij de **LEGO NXT robot** zit er ook een microcontroller in het witte bakje. Via draden kunnen we INPUTS en OUTPUTS op de VERWERKINGS blok aansluiten. Het programma schrijven we in een LEGO flowcode omgeving.



Bij onze flowcode omgeving van onze robotkamp robot werkt dit net zo!

Als we bijvoorbeeld een **LED willen laten knipperen**, dan kan een programma er als volgt uitzien:



Deze programmaflow geeft aan dat we eerst LEDO **AAN** zetten. Daarna **wachten** we 1 seconde. Tenslotte doen we de LEDO weer **UIT** en wachten opnieuw 1 seconde.

Opdat het programma meer dan 1 keer wordt uitgevoerd stoppen we de ganse flow in een **LOOP**.

We zien duidelijk dat de controller stap voor stap denkt.

#### 2. Werking flowcode

We gebruiken flowcode v4.5 (installatie instructies en software vind je terug op de USB stick).

Flowcode is een **visuele programmeer omgeving.** Dit betekent dat we met het slepen van iconen (zie gele bouwblokken in bovenstaand voorbeeld) in een

bepaalde volgorde een programma makkelijk en overzichtelijk kunnen opbouwen. In de iconen kunnen we functies instellen (zie later). Daarna moeten we enkel nog op de "**download**" knop drukken en het programma wordt **in de robot geladen**.



#### De Formula Flowcode buggy simulator

We kunnen ons programma ook **uittesten op de PC** vóór dat we het laden in de robot. Dit doen we in een **SIMULATOR.** Bovenstaande figuur geeft aan hoe deze simulator er uit ziet. Je kan de drukknoppen, LEDs en lijnvolgers, alsook de motoren en LDR makkelijk testen met deze software.

#### Hoe wordt een flowcode programma opgebouwd?



\*Reset betekent dat de microcontroller opnieuw van het **BEGIN icoon** zijn programma uitvoert. We kunnen dit doen door op de **RESET button** op de PCB te drukken.



#### Programma's met de Flowcode buggy schrijven

(merk wel op dat de windows vensters **uit WIN7 zijn geknipt en geplakt**, dus de layout kan een beetje verschillend zijn, maar de werking is exact hetzelfde)

- 3. Mijn eerste programma: een LED laten knipperen
  - 1. Start Flowcode op. Klik hiervoor op het volgende icoon op de desktop:



2. Kies ervoor om een **nieuwe project** aan te maken.



3. Selecteer de "Target" tab. Selecteer nu de "PIC" chip reeks en als family "Misc". Selecteer tenslotte "Formula Flowcode buggy" en druk dan op OK.

Project Options
Project Options Choose a Target Project description General Options Configure Choose a target for this flowchart: Family Family FertO-40 F
ARM Matrix Proto Board (18F24K50) MIAC V2 MIAC V2 Microchip FSUSB Microchip PIC18 SK
OK Cancel

#### 4. Nu start de programmeeromgeving op.



Bovenstaande figuur geeft aan welke vensters er o.a. beschikbaar zijn in deze programmeeromgeving. De 3D simulatie omgeving is niet gebruikt.

Indien bepaalde vensters niet aanwezig zijn kan je deze altijd via het "**view**" **menu** aanzetten. Zorg dat je de **volgende vensters bij de start** van **elke oefening** toont op het flowcode scherm:

- 2D simulatie omgeving (Dashboard Panel)
- **Eigenschappen** iconen instellen (Panel Properties)
- Command toolbox
- Component toolbox



5. Klik op "**Mechatronics**" in het menu en selecteer de "**Formula Flowcode**" robot. Blijf de muis inhouden en **sleep** deze component op het dashboard panel.



Nu staat op het controle paneel **de simulator van de formula flowcode** robot. Deze simulator komt in grote mate overeen met wat onze robot kan. Het voordeel van deze simulator is dat we op een makkelijk niveau kunnen starten. Hij zorgt voor de nodige **macro's** (kleine hulpprogramma's om alles aan te sturen).



Mogelijk staat de robot simulator nog niet op de juiste plaats in het midden bij het openen van de component. Gebruik dan de "**select to move**" knop en **selecteer jouw robot**. Sleep hem naar het **midden** van het blad. Daarna houd je de **CTRL-toets** op je keyboard in en **scroll je met de muis voorwaarts** om de component **in te zoomen** tot dat alles goed zichtbaar is. 6. Klik links op het **LOOP** icoon (knop ingedrukt houden) en sleep dit icoon in het programmeer venster (knop loslaten). Binnen de loop blijft ons programma **altijd draaien**. Dit gaat door tot je op RESET duwt of de spanning afzet.



7. Sleep nu een **COMPONENT MACRO** icoon in de loop. Deze macro heeft op zich nog geen functie. Daarom is het icoontje nog leeg.



8. **Dubbelklik** op het "component macro" icoon. Nu krijg je het volgende venster:

Properties: Macro		
Display name: Call Component Macro		-
Macros Cor	le1	^
LEDOff PlayNote ReadIRSenso	or	-
Parameters:	ture Europeire	
	BYTE 0	•
Return Value:		
	OK & Edit Macro OK Can	cel

Selecteer de Formula Flowcode1 component in de linker kolom.

We willen graag een led laten knipperen op de robot. Kies daarom de macro **LEDOn** in de rechtse kolom. Merk op dat er nog heel wat andere macro's voor onze robot ter beschikking zijn. Deze komen later nog wel aanbod <sup>©</sup>

Tot slot moeten we aangeven **welke LED** we willen laten branden. Type daarom **"0"** voor LED nummer 0 in het parameter invulvak. Dit de eerste LED in de rij. Wil je dus LED 3 laten branden moet je "2" invullen in dit vak. In de elektronica beginnen we namelijk vanaf "0" en niet vanaf "1" te tellen.

Druk nu op OK.



Nu staat er wel in de macro wat er moet gebeuren: doe LED 0 aan.

9. Nu willen we de LED 0 bijvoorbeeld **1 seconde** lang laten branden. Dit kan door het **DELAY** (vertraging) icoon te slepen onder de LEDOn(0) macro.



De delay staat standaard op 1 ms (milliseconde = 1/1000 seconde). Wanneer je **dubbel klikt** op deze delay icoon kan je de tijd aanpassen naar 1 seconde.

Properties: Delay	1		<b>E</b>
Display name:			
Delay			-
Delay value or variable:			
1			•
microseconds	milliseconds	seconds	
0		ОК	Cancel

Selecteer "seconden" en laat de delaywaarde op "1" staan. Druk dan op OK.

10. Werk nu zelf de flowcode af zodat je ook de **LED 0 gaat uitzetten** gedurende 1 seconde. Zie voorbeeld code onderaan.



14

11. Nu kunnen we onze code gaan testen. Eerst gaan we **simuleren.** Om de simulatie te **STARTEN** moeten we enkel op de **rode driehoek** klikken of F5 drukken op het keyboard.



12. Als alles goed is geprogrammeerd kan je nu **LED 0 zien knipperen** op de FormulaFlowcode buggy simulator in het paneel venster. Dit gebeurt op het ritme van 1 seconde aan en 1 seconde uit.



13. De simulator kan je STOPPEN door op het rode vierkant te klikken. Wil je graag een programma laten PAUZEREN (niet helemaal beëindigen maar enkel laten wachten op een bepaalde plaats in de code) dan kan je op de PAUZE knop klikken. Deze bevind zich tussen START en STOP.



14. Je kan de simulatie tijd van jouw code zelf aanpassen door in het "simulatie debugger" venster de slider aan te passen (links is max snelheid van de processor, rechts is heel traag). Ondertussen zie je in jouw code een rood kadertje van icoon naar icoon springen. Dit geeft aan waar we ons in de code bevinden. Handig om fouten op te sporen in de code.

Simulation debugger			×
Simulation speed:		Simulation de	lay:
	-0-		Skip
Macro Calls		Show integers as hex	
Main		Expression	Value
		······ Expression	
<	Þ		

- 15. Jouw code kunnen we ook **testen op de robot**. Zorg daarom dat steeds <u>eerst</u> de simulator is GESTOPT.
- → Sluit nu de **robot via de USB kabel** aan op de PC.
- → Zet de spanning op van de robot (POWER knop AAN). Beide LEDs D0 en D1 moeten knipperen. Zo niet, druk dan op reset of check de spanning van de batterijen na. Mogelijk is de driver van de robot niet goed geïnstalleerd als de LEDs beiden constant flikkeren (zie device manager en driver op USB stick). Los dit probleem eerst op voordat je verder gaat.
- → Druk nu op de **download** knop (compile to chip):



→ Nu vraagt Flowcode jou om de code eerst te SAVEN (op te slaan) op een plaats op jouw PC.



Opdat je straks jouw eigen programma's mee naar huis kan nemen gaan we jouw code opslaan op de voorziene USB stick.

- → Plug daarom nu eerst jouw USB stick in de PC.
- → Sluit het pop-up venster van Windows waar deze voorstelt wat je wil gaan doen met jouw USB stick.
- → Klik nu op "ja" in Flowcode.
- → Nu krijg je de vraag om aan te geven waar je jouw code wil opslaan. Browse (verwijs) naar jouw USB stick en maak er een nieuwe directory (map) aan. Noem deze "flowcode oefeningen". Een nieuwe map kan je maken door eerst op "nieuwe map" te klikken in het window en dan de naam van de map in te typen. Daarna druk je op de "ENTER" toets om jouw naam te bevestigen.

Save As			0-8-	- <b>7</b> -		×
Computer + F	RANK	5.16GE	B (G:) 🕨 👻 🗸	Zoeken in FRAI	VKS 16GB (G:)	٩
Organiseren 🔻 🛛 Nieuwe map					== - (	0
📔 Afbeeldingen	*	Na	am		Gewijzigd op	*
Documenten			2012		1/05/2012 0.26	
👌 Muziek	/		2012		1/05/2012 9:30	
Video's			automatisatie		19/01/2012 9:50	
			books		9/03/2012 19:00	
💐 Thuisgroep			eigen ontwerpen		19/01/2012 9:52	
thuisgroup			films		5/02/2012 20:21	
Computer			games		3/05/2012 10:39	
Computer			kieswijzer		31/01/2012 14:1	9 ≣
Ecoverny (C:)			muziek		28/03/2012 11:1	2
	=		pcb's lln		12/01/2012 9:57	
HP_TOOLS (F:)	_		project SRF08		3/02/2012 10:04	
FRANKS 16GB (G:)			resultaten IIn		14/05/2012 12:4	6
🦋 My Web Sites on MSN			roburobot sw		30/01/2012 10:0	5
			tools		19/01/2012 10:1	8
👊 Netwerk			flowcode oefeningen		25/06/2012 14:2	6 -
	*	1			•	
Bestands <u>n</u> aam: knipperled						-
Opslaan als: Flowcode fo	r PIC f	iles (*.	.fcf)			•
📥 Mappen verbergen				Opslaan	Annuleren	

 → Dubbel klik dan op deze "flowcode oefeningen" map zodat je er in gaat staan.
 → Geef nu een naam aan jouw programma. Type daarom "knipperled" in het "bestandsnaam" invulvak.

🛃 Save As		_			x
😋 🗢 🛡 🕌 « FRANKS 16GB (G:) 🕨	flowcode oefeningen	• 4 <sub>7</sub>	Zoeken in f	ilowcode oefeningen	P
Organiseren 🔻 Nieuwe map					0
Afbeeldingen ^	Naam	^		Gewijzigd op	Ту
Documenten		Geen zoekre	sultaten.		
Video's					
🤣 Thuisgroep					
🖳 Computer					
🚢 Lokale schijf (C:)					
RECOVERY (D:)					
HP_TOOLS (F:)					
FRANKS 16GB (G:)					
💘 My Web Sites on MSN					
🗣 Netwerk					
-	<				- F
Bestandspaam: knipperled					-
Opslaan als: Elowcode for PIC fi	les (* fcf)				-
Interest and the second for the fit					
Mappen verbergen		(	<u>O</u> pslaan	Annuleren	

→ Druk dan op de knop "**opslaan**".

→ Nu wordt de code gedownload in de robot. Wanneer dit gebeurd is verschijnt er op het Flowcode download venster o.a. de volgende tekst:

Compiler Messages	
FINISHED	-
Building CASM file	
Memory Usage Report	
RAM available:2048 bytes, used:48 bytes (2.4%), free:2000 bytes (97.6%),	
Heap size:2000 bytes, Heap max single alloc:127 bytes	
ROM available:22528 bytes, used:530 bytes (2.4%), free:21998 bytes (97.6%)	
51100055	
- Launching the programmer	
C:\Program Files (x86)\Flowcode 6\tools\mLoader\mLoader.exe -ff "oefl.hex"	
Loading file	
File loaded from	
=> oefl.hex	
File sending	
File gent!	
FINISHED	
	Þ.
Cancel	

Als alles goed is gegaan moet er nu "File sent" en op het einde "Finished" staan.

16. Jouw code kunnen we nu gaan testen op de robot. <u>Na het downloaden van de</u> <u>code voert de robot **direct** jouw code uit.</u>



- → Als alles goed is gelopen zal nu LED 0 knipperen op het 1 seconde ritme.
- → Je kan ook de robot zo instellen dat de code pas start nadat er op de knop RB4 of RB5 op de PCB is gedrukt. Deze functie heet "wait for button".

De instelling voor het wel of niet direct werken van de robot na de download kan je terug vinden in de Properties van de Formule Flowcode robot. Zie de "**wait for button**" propertie.

Properties	×
FormulaFlowcode1	-
Properties + Position	
Component	
🔑 Handle	FormulaFlowcode1
Туре	Formula Flowcode
Properties	
	Yes 🚽 🚽
IR Sensitivity	Byte (8-bit)
Z Motor Balance (-100 - 100)	0
🚊 🕍 IR Thresholds (CheckIR macro)	)
	100

Deze instelling kan **enkel werken wanneer** je de flowcode macro "**initialise**" hebt toegevoegd aan het begin van jouw code. Dit is handig wanneer we met **motoren** gaan spelen. Zie daar voor **meer uitleg** over deze extra macro.

#### **Opmerkingen:**

Het is **mogelijk** dat de robot ook **vooruit gaat rijden** tijdens deze code test en dat er **meerdere LEDs gaan branden**. Dit komt omdat de waarden op de poorten van de microcontroller <u>na een reset niet altijd betrouwbaar z</u>ijn.

Dit kan je **verhelpen** door aan het begin van jouw code de uitgangen van de **poorten C, D en E op nul** te zetten. Een poort bestaat uit maximum 8 pinnen. Dit zijn de aansluitingen van de microcontroller met de buitenwereld (schakelaars, LEDs ...).

Voeg daarom het volgende **OUTPUT** icoon toe aan jouw code:



→ Dubbel klik op het OUTPUT icoon en selecteer PORT D.

Properties: Output	
Display name: Output	•
Variable or value:	Port:     PORTD ▼
Output to:	
⊚ Single Bit:	Entire Port:
0 -	Use Masking:
	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
0	OK Cancel

- → Zorg dat de variabele waarde op "0" staat en dat de "hele poort" is geselecteerd. Nu sturen we een "0" naar gans de PORT D (8 uitgangen op nul gezet om de LEDs uit te zetten).
- ➔ Druk dan OK.

→ Doe dit zelfde voor PORT C en PORT E om de motoren stil te leggen. Deze zijn aangesloten op PORT E.



- → Als deze code is toegevoegd download je de code nog maar eens in de robot. Je zal merken dat nu enkel LED 0 zal knipperen. Voeg daarom deze 3 OUTPUT iconen steeds toe aan al je volgende programma's.
- → Telkens als je iets verandert aan de code en je wil gaan downloaden, dan moet je even de code saven. Flowcode zal jou hier altijd achter vragen. Gewoon op "ja" klikken.

PROFICIAT, je hebt jouw eerste programma geschreven en getest 😊

#### 4. Andere programma's:

#### 4.1 Nog meer LEDs laten knipperen

Nu we weten hoe we 1 led kunnen laten knipperen willen we graag **meerdere LEDs tegelijk** laten knipperen.

Je zou telkens de MACRO "LEDon" kunnen gebruiken om de verschillende LEDs aan te sturen. Je kan ook de MACRO "**WriteLeds**" gebruiken en een waarde tussen 0 en 255 intypen.

Wat stelt dit getal voor?

De leds D0, D1, D2, D3 en D4 zijn 5 leds die aangesloten zijn op poort D. Je kan ze binair aansturen. Binair betekent dat we enkel met "1" of "0" kunnen werken. Elke pin van poort D moet dus een "1" of "0" krijgen om aan of uit gezet te worden.

Poort nr	D7	D6	D5	D4	D3	D2	D1	D0
decimaal	128	64	32	16	8	4	2	1
gewicht								

Merk op dat er geen LEDs zijn aangesloten op D5, D6 en D7 bij deze robot.

Als ik bijvoorbeeld alle 5 de LEDs tegelijk wil aanzetten moet ik de gewichten van elke pin optellen: 1 + 2 + 4 + 8 + 16 = 31

Als ik dus een "WriteLeds" MACRO gebruik en het getal "**31**" invul dan moeten alle LEDs aangaan.

Wil ik graag een andere combinatie van LEDs laten branden, dan moet ik enkel de **juiste gewichten bij elkaar optellen** en deze versturen.

Zo gaan **enkel de LEDs op D0 en D4** oplichten als ik het getal 1 + 16 = **17** verstuur.







Om al de **LEDs uit** te krijgen moet je een "**WriteLeds**" macro met de waarde "**0**" sturen.

#### UITDAGINGEN:

- 1. Maak een programma waarbij afwisselend LED D0 + D1 en dan D3 + D4 oplichten.
- 2. Maak een knight ridder. Hierbij lopen de LEDs ombeurten van links naar rechts en terug.

<u>Opmerking voor de begeleider:</u> spring nu even naar hoofdstuk 4.7 om ook eens te spelen met de 4 LEDs in het chassis. Het is namelijk veel mooier dat deze LEDs aangestuurd worden tijdens bijvoorbeeld het lijnvolgen. Dan heeft de IIn hier al ervaring mee.

#### 4.2 Schakelaars inlezen

Onze robot heeft 2 schakelaars: RB4 (rechts) en RB5 (links).

Deze schakelaars kunnen we inlezen en daarna kunnen we hier een actie aan koppelen. Zo kan je een LED pas laten branden als je op de knop drukt.

→ Maak eerst een nieuw programma door op NIEUW (New) te klikken in het menu BESTAND.



- → Kies opnieuw de FormulaFlowcode robot als doel.
- → Selecteer de FormulaFlowcode **simulator** ook in de **Mechatronics** menu.
- → Maak een LOOP. Voeg er de OUTPUTS aan toe die PORTC, D en E op nul zetten.
- → Deze vorige stappen herhaal je telkens weer bij een nieuwe oefening.
- → Voeg een macro toe in de LOOP.



→ Dubbelklik op de component macro en kies voor de macro "ReadSwitch".

Properties: Macro		
Display name: Call Component Macro		<b>•</b>
Macros	ponents 🛱 Simulation 🌈 Functions	
ReadLineSenso	or	*
Reverse     SetMotors     SpinLeft     SpinRight     Stop		II
Parameters:		
Name R Switch	Type Expression	
D Switch		
Return Value:(BYTE)		
0	OK & Edit Macro OK Cance	

- → Vul " 'L' " van Linkse schakelaar tussen 2 haakjes in in het parameter veld. Je mag ook een 0 invullen. Als je de " 'R' " gebruikt kan je de rechter schakelaar afvragen. Je kan ook een 1 invullen.
- → We willen graag te weten komen of de schakelaar is ingedrukt of niet. Dit doe je door de stand van de schakelaar te onderzoeken. De stand van de schakelaar wordt voorgesteld door een "1" (gesloten of ingedrukt) of "0" (open of los laten).
- → De waarde die we terug krijgen van de schakelaar (retour waarde) moeten we in een variabele bewaren. Een variabele is een plaats in het geheugen van de microcontroller waar we deze waarde kunnen in opslaan en verder gebruiken in onze code.

#### Hoe maak je een variabele?

- Druk op de onderste "variabelen" knop (neergaande pijl) van bovenstaande tekening.
- → Selecteer "Globals" als type variabele.

- Ports 🔀 Globals 🗵 Locals	
Globals	
Constants	_
b false	
b true	
✓ Variables	
Add new	
Delete unused	

- → Druk op de pijl langs "variables" en selecteer "Add new".
- ➔ Type de naam "schakelaar\_links" als nieuwe naam voor de variabele van de schakelaar. MERK OP dat de naam <u>uit 1 woord</u> moet bestaan (underscore gebruiken).
- → Kies als type een "**Bool**". Bool betekent dat we een waarde van 0 of 1 kunnen opslaan in de variabele. Druk dan OK.

Create a New Variable			
Name of new variable:			
schakelaar_links			
Initial value:			
Description:			
Variable type:			
Bool (either true, 1 or false, 0)			
Byte (number in the range 0 to 255)			
Int (number in the range -32768 to 32767)			
◯ UInt (number in the range 0 to 65535)			
Cong (number in the range -2147483648 to 2147483647)			
ULong (number in the range 0 to 4294967295)			
String (default size = 20)			
Floating point			
Object handle			
OK Cancel			

→ Dubbel klik nu op onze nieuwe variabele "schakelaar links":



→ Nu is onze variabele zichtbaar in het return value. Druk op OK om onze macro af te werken.

Properties: Macro	1.4	18070	
Display name:			
Call Component Macro			-
Macros Comp Macros Comp ReadLineSenso ReadSwitch ReadSwitch Reverse SetMotors SpinLeft SpinRight Stop WaitEorSwitch	ponents 🕂 S	imulation from Functions	* 
Parameters:	Type	Evonession	
Rewitch	DVTC	17.1	_
Return Value:(BYTE)			
schakelaar_links			•
0	OK & <u>E</u> dit Macr	o OK Cancel	

→ Voeg nu de onderstaande code toe om de inhoud van de schakelaar op LED 0 te plaatsen.

We selecteren bij de **WriteLeds** macro de variabele "**schakelaar\_links**". De inhoud van deze variabele wordt dan getoond op de ganse **PORT D** (alle LEDs).

-		2D. Dashboard	ranci			
	Properties: Macro		0.11			
	Display name: Call Component Macro				<b>_</b>	
	Macros Comp	oonents 📫 S	imulation	Fix Functions		
	SetMotors SpinLeft SpinRight Stop					
	WaitForSwitch					7 堀瓦 向
Н	Parameters:				Ē	
Н	Name	Туре	Expression			Constants
Ш	B LED_Byte	BYTE				b false
Ш						true
						b schakelaar_links



De linkse schakelaar wordt ingelezen en getoond op de LEDs (LED 0).

Frank Marchal EDULAB → Simuleer nu jouw code. Druk op de Left (linker) knop en zie of LED 0 aan/uit gaat.



- → Download jouw code en test ze op de robot. Save jouw code onder de naam "schakelaar\_inlezen" in de map "flowcode oefeningen" op de USB stick.
- → Druk op RB5 ( de rechter schakelaar, kort bij de voedingsknop). Nu moet enkel LED 0 oplichten. Laat je de knop los, dan gaat de LED ook terug uit.
- → Waarom licht slechts 1 LED op? Omdat we maar 1 pin van de microcontroller afvragen (pin 5 op poort B = RB5) wordt enkel de status van deze pin getoond. Eén pin stelt ook maar 1 bit voor, vandaar 1 LED die oplicht.

#### Uitdagingen:

- 1. Probeer nu zelf enkel de rechtse knop in te lezen en te tonen op PORT D.
- 2. Toon nu de rechtse knop enkel op LED 2 op PORT D.
- 3. Lees nu beide schakelaars in en toon de linkse schakelaar op LED 1 en de rechtse schakelaar op LED 3.

<u>Hoe maak je een programma</u> waarbij de LED **enkel oplicht als zowel de linkse als rechtse** schakelaar zijn ingedrukt?

We zijn hier op zoek naar een **logische EN (AND) schakeling**. Enkel als men op links EN rechts drukt zal de LED branden.



EN (AND) schakeling

In Flowcode moeten we hiervoor een **DECISION** (beslissing) toevoegen om een EN schakeling te kunnen realiseren (zie onderstaande code).

In de DECISION kunnen we onze voorwaarde instellen.

Properties: Decision	×
Display name:	
Decision	▼
lf:	
schakelaar_links = 1	▼
Swap Yes and No	
	OK Cancel

Als de schakelaar\_links = 1 dan volgt de code het "ja" pad in de flow.



In deze code wordt de linkse schakelaar afgevraagd. Als de schakelaar is ingedrukt (= 1) moet LED 1 aangaan. Anders gaat LED 1 (in feite alle LEDs) uit.

Als we nu de stand van beide schakelaars willen afvragen om een **EN (AND) poort** te realiseren krijgen we het volgende:



Hier vragen we eerst zowel de linkse als rechts schakelaar af. We gebruiken een '0' voor links en een '1' voor rechts in de ReadSwitch macro.

Daarna wordt in de Decision de volgende vraag gesteld:

#### (schakelaar\_links = 1) AND (schakelaar\_rechts = 1)

**AND** is het Engels voor EN-poort. De flowcode software werkt enkel met Engelse commando's. Let op de <u>haakjes</u> die aanduiden wat er <u>eerst moet uitgevoerd</u> worden.

**Enkel indien beide knoppen zijn ingedrukt** volgt de code het 'ja' pad. Dan zal LED D2 oplichten (zie Write LEDs(4)). Anders worden alle LEDs uitgezet.

Frank Marchal EDULAB V2

#### **Uitdaging:**

 Probeer nu eens een OF (OR) poort te ontwerpen. Dit wil zeggen dat bijvoorbeeld LED D2 gaat branden als 1 of meerdere knoppen zijn ingedrukt. De onderstaande tekening verduidelijkt de werking van de OR poort.



OR schakeling

4.3 Geluid maken

Buiten de LEDs en de schakelaars heeft onze robot o.a. een **zoemer** aan boord.



De zoemer stelt ons in staat om tonen en zelfs muziek te maken met de robot.

Frank Marchal EDULAB Flowcode heeft hiervoor de macro **PLAYNOTE** voorzien.

- → Maak eerst de voorbereidende code en voeg er een lege macro component aan toe (zie vorige hoofdstukken en onderstaande code).
- → Save jouw code onder de naam "tonen genereren".



Voorbereiden code

- → Open dan de macro component en kies de macro PLAYNOTE.
- → Bij deze macro moeten we nog **2 extra parameters** toevoegen:
- De toonhoogte (note).
- De **tijd** dat de toon moet te horen zijn (delay). Deze tijd is uitgedrukt in milliseconden (1/1000 seconde). Deze waarde kan van 0 tot 65535 gaan.

De toonhoogte wordt	weergeven in v	olgende tabel:
---------------------	----------------	----------------

note	value	BEGIN
G G# A	0 14 28	C FormulaFlowc PlayNote(64,
A# B	40 53	PlayNote(64,
C#	75 85	FormulaFlowc PlayNote(127
D# E	94 103 112	G FormulaFlowc PlayNote(127
F# G	120 127	A FormulaFlowc PlayNote(141
G# A A#	135 141 148	A FormulaFlowc PlayNote(141
B C C#	154 159 165	G FormulaFlowc PlayNote(127
D D# E	170 175 179	END
F	183	

Elke toon heeft een bepaalde waarde. De C (of do noot) kan je horen als je "64" als waarde invult in PLAYNOTE.

roperties: Macro			-
Dieplay pame:			
Call Component Macm	<u>,</u>		
Call Component Macro	,		
Macros 📋	Components	] Simulation   Fix Function	ons
🚊 🛅 FormulaFlov	vcode1		
CheckIR			ſ
Forward			
Initialise			l
LEDON			
	ensor		
Read DF	2		
Parameters:			
Name	Туре	Expression	
B Note	BYTE	127	
🔀 Delay_ms	UINT	100	
Return Value:			
Fretuint value.			
	OK & Edit		Cancel

Frank Marchal EDULAB Kies PLAYNOTE als macro, vul de noot waarde in (bijvoorbeeld 127), type dan een **komma** en vul dan de tijd in dat de noot moet gehoord worden (100 in dit voorbeeld).

Druk dan op OK en test jouw code (een PC met boxen kan deze toon laten horen).



Deze code kan je beluisteren door jouw **simulator** aan te zetten (rode pijl) of door de code te **downloaden in de robot**. Merk op dat de tonen van de robot juist klinken.

#### Uitdagingen:

- 1. Componeer jouw eigen lied door de juiste noten achter elkaar te zetten met de juiste intervallen er tussen (tijden).
- 2. Je kan ook de noten van de volgende muziek ingeven.



Voor de minder sterke muzikanten staan hieronder de volgende noten (zet ze om in cijfers, zie tabel):

C = 64, D, E, C x 2 E, F, G x 2 G, A, G, F, E, C x 2 C, lage G = 0, C x 2

Let ook op het ritme: zwarte noten met een horizontaal streepje (vlagje) zijn 2 keer sneller dan een zwarte noot zonder vlagje.

Witte noten zijn 2 keer trager dan een zwarte noot zonder vlagje.

Frank Marchal EDULAB

#### 4.4 Motoren laten draaien

Onze robot is voorzien van 2 flinke motoren. We kunnen deze motoren afzonderlijk of te samen vooruit en achteruit laten draaien. We kunnen de robot ook links en rechts laten draaien.

Hiervoor maken we gebruik van de volgende macro's:

- FORWARD (voorwaarts)
- **REVERSE** (achterwaarts)
- **STOP** (stop de motoren) (geen power in te stellen)
- SPINLEFT (draai naar links)
- **SPINRIGHT** (draai naar rechts)

Bij 4 van de 5 macro's moeten we enkel de kracht (**POWER**) van de motoren instellen. Dit doen we door een getal tussen 0 en 255 in te vullen in het invulveld POWER.

- **SetMotors** is een 6<sup>de</sup> macro waarbij we **afzonderlijk** de power van de linker en rechter motor kunnen instellen.

Display name:			
Call Component Macro	D		•
Macros	Components	Simulation <b>F</b> Functions	
Parameters:			
Name	Туре	Expression	
Damas	DVTC		

**Opmerking**: je zal merken dat **de robot pas gaat rijden vanaf een bepaalde waarde.** Toch worden op dit moment de motors al aangestuurd. Het gewicht van de robot en de tandwielen in de motoren zorgen echter voor de nodige **mechanische wrijving**. Dit is net zoals bij een locomotief: deze heeft het in begin heel moeilijk om van start te gaan en om alle wagons te bewegen. Daarna heeft hij minder kracht nodig om alles in gang te houden.

We moeten ook nog **voorkomen** dat de robot **direct gaat rijden** na de download van de code. Daarom voegen we een "**initialise**" flowcode macro toe aan het begin van de code.



Wanneer we nu ook nog in de **properties** van de flowcode robot de eigenschap "**wait for button**" **op "yes**" zetten, dan zal de robot wachten met de code uit te voeren totdat we op **RB4 of RB5** gedrukt hebben.

#### Zo voorkomen we dat de robot spontaan van tafel rijdt!

-	Туре	Formula Flowcode	
ш	The Properties		
		Yes 🚽 🚽	
	IR Sensitivity	Byte (8-bit)	
	Z Motor Balance (-100 - 100)	0	
	in Thresholds (CheckIR macro)		
	77 1.4	100	

De robot als **een slang** laten rijden:

Maak de volgende code aan en download deze in de robot of simuleer de code eens.



#### Opgelet: test de robot steeds in de bak of recht op! Anders rijdt hij mogelijk van de tafel.

#### Uitdagingen:

 Laat de robot in de simulator rijden in een vierkant (sleep hiervoor in het "tools" menu de "Formula Flowcode Racetrack" op het 2D dashboard panel). Nu kan je de code instellen zodat de robot mooi binnen de lijnen blijft rijden.



- 2. Zoek de snelheidswaarde uit waarbij de motoren net gaan draaien.
- 3. Laat de robot 2 seconden vooruit rijden, dan stoppen, weer 2 seconden vooruit rijden, weer stoppen, enz... in de bak.
- 4. Laat de robot in een vierkant rijden in de bak.

#### 4.5 LDR inlezen

Onze robot is voorzien van 2LDR's. LDR staat voor **Light Depending Resistance** = licht gevoelige weerstand. Wanneer er **licht valt op de LDR** gaat de **weerstandswaarde verkleinen** en wordt deze verandering door de microcontroller waargenomen.



Via de macro **ReadLDR** kunnen we enkel de **linkse** LDR inlezen.

Properties: Macro				
Display name:				
Call Component Ma	cro			<b>~</b>
Macros	Components	s 📑 Simulatio	n $f_{\mathbb{X}}$ Functions	
Components				
📄 👘 FormulaFle	owcode1			
Check	IR			
- Forwa	rd			=
Initialis	e			
LEDO	f			
LEDOr	1			
	ite Concern			
Readi	CSerisor			
ReadL	DK inoConcor			
	lic			-
Parameters:				
Name	Туре	Expression		
Return Value:(INT)				
LDB				<b>_</b>
0	OKA	Edit Macro	OK	Cancel
	UNA			Cancel

Maak de **variabele LDR (type Byte)** aan om de waarde van de LDR in te bewaren. Zie bij het hoofdstuk van de "schakelaars inlezen" hoe je weer een variabele kon aanmaken!

### Met de volgende code kan je de **linkse LDR waarde** toonbaar maken op jouw robot LEDs:



De waarde van de LDR zal liggen tussen 0 en 255.

De LEDs geven echter een **binaire waarde** weer. Wanneer er **veel licht** op de LDR valt moeten er **slechts enkele LEDs** aan zijn. Dan is de **spanning klein** aan de ingang van de microcontroller.

```
0000 0000 = 0 (heel veel licht op de LDR = kleine weerstand)

0000 0001 = 1

0000 0010 = 2

0000 0011 = 3

....

1111 1111 = 255 (geen licht op de LDR = grote weerstand)
```

We kunnen opnieuw, net als bij het commando **WriteLeds** de waardes telkens gaan uitrekenen, afhankelijk van het aantal LEDs die branden.

V2

Zie volgende figuur i.v.m. het uitrekenen van binaire naar decimale waardes.



Test het volgende programma om de robot enkel **vooruit te laten rijden** wanneer er met een **zaklamp op de linker LDR geschenen** wordt.



Nadat de LDR waarde is ingelezen wordt er gekeken of de waarde **kleiner is dan 80 (veel licht)**. Als dit zo is gaan de motoren met een kracht van 200 vooruit gestuurd worden. Zo niet blijft de robot stilstaan. De initialise macro dient om de robot niet zomaar te laten rijden als de code gedownload is. Eerst op RB4 of RB5 drukken.

V2



Je kan de **LDR waarde testen** bij Flowcode 6 door een **potentiometer** toe te voegen. Je moet dan de byte waarde van deze potentiometer inlezen in LDRL.

#### Uitdagingen:

- 1. Maak een programma waarbij telkens als er veel licht (zaklamp) valt op de robot, de robot zich 180 graden gaat draaien.
- 2. Probeer eens op de LEDs niet de binaire waarde te tonen maar telkens als er veel licht valt op de LDR, gaan er veel LEDs branden. Als er weinig licht op valt branden er minder LEDs. Hiervoor moet je meerdere decisions voorzien in de code.

#### 4.6 Een lijnvolger maken

Als volgende oefening voor beginners gaan we nu een **lijnvolger** bouwen. De bedoeling is dat de robot de **zwarte lijn gaat volgen** op het papier dat voorzien is in de bak.

Straks gaan we ook een **gezamelijke wedstrijd** houden om jouw robot zo snel mogelijk te laten rijden over die zwarte lijn.



Jonas en Stijn deden met hun eigen gebouwde robot met lijnvolger mee aan robocup junior.

We kunnen de robot een lijn laten volgen door gebruik te maken van de 2 infrarood sensors. Infrarood (IR) wil zeggen dat er een niet zichtbaar licht wordt uitgezonden, net als bij een afstandsbediening.

We zenden steeds hetzelfde IR licht uit.

Wanneer het licht wordt geschenen op een **wit oppervlakte** weerkaatst er heel veel licht terug. Dit maakt dat de robot een **binaire '0'** zal zien.

Wanneer de robot zijn IR licht op een **zwart oppervlakte** laat schijnen, komt er weinig licht terug en ziet de robot **een '1'.** 





2 IR sensors aan de onderzijde van de robot maken lijnvolgen mogelijk.

Om de IR sensors te kunnen inlezen maken we gebruik van de macro **ReadLineSensor.** 

Display name:		
Call Component Mac	o	
Macros	Components	Simulation fx Functions
Components	wcode1 t i Sensor R leSensor c	
Parameters:	l+	le i
Name	Туре	Expression
B Sensor	BYTE	1
Retum Value:(BYTE) IRleft		
INTERP		

We moeten wel de microcontroller nog vertellen welke lijnsensor er moet worden ingelezen. Voor de **linkse lijnsensor** moet je een **'1'** invullen in het parameters veld. Voor de **rechtse lijnsensor** moet je een **'0'** invullen.

De gelezen waarde van de lijnsensor stoppen we in een **zelf aangemaakte** variabele IRleft en IRright van het type Byte.

Test de volgende code uit om jouw IR sensors te testen.



De code laat zien dat we eerst de linkse lijnsensor inlezen. Als de sensor **verschillend is ( <> )** van 0 (dus o.a. een 1) dan zal led D4 aangaan. Anders gaat D4 terug uit.

Hetzelfde doen we voor de rechtse lijnsensor.

### Test de volgende lijnvolger code in de bak uit of op de testmat !

Je kan een lijnvolger maken met 1 lijnsensor:

Strategie:

We gebruiken **enkel de rechtse sensor**. Als de sensor op de zwarte lijn is (1) dan gaan we naar links. Als de sensor op het witte vlak komt gaan we naar rechts. We kunnen de motoren instellen met de **SetMotor** macro. **De hoeveelheid power / motor** bepaalt hier in welke richting de robot zal draaien. Neem je links 200 en rechts 255 als waarde, dan gaat de robot naar links rijden, want de rechter motor gaat harder draaien dan de linker.



Je kan ook een lijnvolger maken met 2 lijnsensoren (beste prestaties):

Strategie:

Laat de robot de **wit/zwarte rand volgen** van de lijn. Dan ziet de linkse sensor wit (0) en de rechtse sensor zwart (1). Dan gaat hij FORWARD.

Als de robot met <u>beide</u> sensoren op de zwarte lijn rijdt moet hij naar **links draaien** (SPINLeft). Rijdt hij met <u>beide</u> sensoren op de witte kant dan moet hij naar **rechts draaien** (SPINRight). De motor power kan je zelf naar keuze instellen.



Je kan met de snelheid experimenteren (0 tot 250 per motor)

Voeg ook de 4 chassis LEDs toe om aan te geven of de robot naar links of rechts wordt gestuurd. Dit geeft een mooi effect.

Simuleren van lijnvolgers kan door een knop te voorzien voor de linker en rechter sensor. Test dit zeker ook eens uit.

Frank Marchal EDULAB

#### 4.7 De 4 chassis LEDs aansturen

Als laatste oefening hebben we de 4 chassis LEDs nog te goed die we graag willen aansturen.

Deze LEDs zijn via een connector en draadjes verbonden met de PCB.



De LEDs worden niet aangestuurd met een macro. Matrix Multimedia (de bedenkers van het robotje) hebben deze LEDs niet in hun design voorzien.

Wij kunnen steeds nog zelf outputs aansturen met het icoon OUTPUT.

De 4 chassis LEDs zijn aangesloten op de 4 kleinste pinnen van PORT A: A0, A1, A2 en A3.

Properties: Output	×
Display name:	
Output	•
Variable or value:	Port: PORTA
Output to:	
Single Bit:	Entire Port:
0	Use Masking:
	7 6 5 4 3 2 1 0 <b>F F F F F F F</b>
0	OK Cancel

Van PORT A wordt enkel Bit 0 op '1' gezet. Dus pin A0 = 1 en de LED gaat branden.



Met dit programma gaan we, nadat we de motoren hebben uitgezet en de leds hebben afgezet, 1 voor 1 een led bij aanzetten op de PORT A.

#### UITDAGINGEN:

- 1. Zorg eens voor een disco op jouw robot (LEDs op eigen ritme laten knipperen).
- 2. Combineer jouw lijnvolger samen met deze 4 LEDs.
- 3. Zorg eens dat de LEDs gesimuleerd worden via een extra toegevoegde **LED array** op **Dashboard Panel**. Sluit de LEDs juist aan op de PORT A.

Frank Marchal EDULAB Deze basiscursus is uiteraard te kort om al de features uit te leggen van een microcontroller. Dit kan je leren in een TSO opleiding Elektriciteit - Elektronica.

Toch willen we jou al een eind op weg zetten. Daarom is er documentatie en testsoftware voorzien. Vraag maar aan jouw begeleider.

Neem ook eens een kijkje op de <u>www.edulab.be</u> website voor nog meer inspiratie of een nieuwe elektronica uitdaging.

Veel plezier met het verder programmeren en ontdekken van jouw robot. Proficiat dat je zo hebt volgehouden.

#### Techniek is FUN.





FRANK MARCHAL HOMMELHEIDE 45 B-3500 HASSELT +32 (0) 498 82 15 90 INFO@EDULAB.BE WWW.EDULAB.BE