Light and sound in de lessen wiskunde

Dr Didier Deses*

Samenvatting

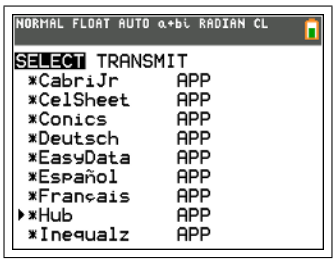
In deze workshop zullen we een aantal concrete voorbeelden uitwerken waarbij het gebruik van de **TI-Innovator Hub** via licht en geluid een meerwaarde kan bieden in de lessen wiskunde.

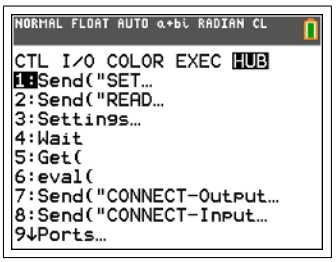
Inhoudsopgave

*Leerkracht wiskunde KA Koekelberg, medewerker aan het departement wiskunde van de VUB, stuurgroep T^3-Vlaanderen

1 Inleiding

Voor je aan de slag kan met de **TI-Innovator Hub** moet je eerst de meegeleverde `app Hub` op je **TI-84 Plus Color** zetten. Dit kan vanaf een computer of via een andere **TI-84 Plus Color** waarop het reeds geïnstalleerd is. Eenmaal de `app` ook op jouw **TI-84 Plus Color** staat, zal ze het menu `[prgm]` in de programmeereditor uitbreiden met een nieuw submenu `[HUB]` waarin de commando's staan om met de **TI-Innovator Hub** te communiceren.





Eenmaal dit gedaan is, kan je in een programma instructies doorsturen naar de **TI-Innovator Hub** . Dit gebeurt door berichten en commando's te zenden via `Send`. Wij zullen de volgende berichten gebruiken.

- `SET LIGHT ON TIME` t: De LED wordt voor t seconden aangezet.
- `SET COLOR` r g b: De kleur LED wordt aangezet in een bepaalde kleur.
- `SET SOUND` f `TIME` t: Produceert een toon met frequentie f gedurende t seconden.

Je vindt deze berichten (of delen ervan) onder `prgm` `[HUB]` `[Send("SET...]` en `prgm` `[HUB]` `[Settings...]`. Andere combinaties van deze berichten zijn ook mogelijk maar we zullen ze hier niet gebruiken.

Naast de berichten naar de **TI-Innovator Hub** zijn er ook enkele commando's die gebruikt zullen worden. Deze vind je onder `prgm` `[HUB]`.

- `Send(`*bericht*`)`: Verstuurt een bericht naar de **TI-Innovator Hub** .
- `Wait` t: Pauzeert de uitvoer van het programma gedurende t seconden.
- `eval(`*uitdrukking*`)`: Dit commando wordt binnen een bericht gebruikt om een uitdrukking te evalueren en om te zetten naar een getal. Dan pas wordt het bericht verstuurd naar de **TI-Innovator Hub** .

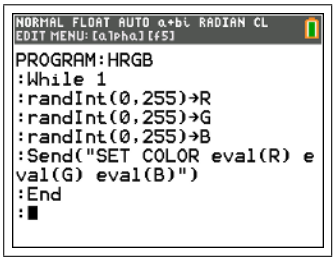
2 Licht

2.1 RGB-waarden

Uit de optica weet men dat elke kleur die ons oog kan waarnemen, gevormd kan worden door een mengsel van rood, groen en blauw licht. Daarom bestaat in de informatica een scherm uit pixels, elke pixel is opgemaakt uit drie LEDs: een rode, een groene en een blauwe. Op de **TI-Innovator Hub** is zo één opstelling van drie LEDs beschikbaar. Op deze wijze kunnen we één pixel nabootsen.

De intensiteit van elke afzonderlijke LED wordt gegeven door een getal tussen 0 en 255, dit komt intern overeen met één byte (=8 bits). Elke kleur komt dus overeen met een RGB-code, dit zijn drie waarden (R, G, B), telkens tussen 0 en 255, die aangeven hoeveel van elke basiskleur aanwezig is. We kunnen dus meer dan 16 miljoen kleuren maken (reken dit uit).

Met onderstaand programma wordt telkens een willekeurige waarde voor R, G en B genomen. We bekomen een disco flikkerlichtje.



Met `While 1` wordt een oneindige lus gestart. Daarna worden met `math` `[prb]` `[randInt]` voor R, G en B willekeurige waarden tussen 0 en 255 gekozen. Tenslotte worden met `SET COLOR eval(R) eval(G) eval(B)` de kleur LEDs aangezet in de juiste kleurintensiteiten.

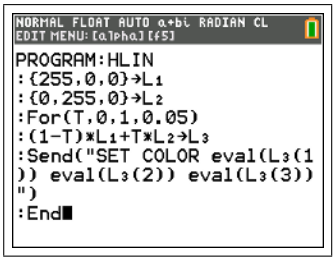
2.2 Lineaire interpolatie: de regenboog

De mens kent al zeer lang de kleuren van de regenboog: rood, oranje, geel, groen, blauw, indigo en violet. In werkelijkheid varieert de kleur echter op continue wijze. De verschillende 'banden' zijn een gevolg van de beperkingen van het menselijk oog. We zullen de kleur LEDs gebruiken om de opeenvolging van kleuren in een regenboog te simuleren.

We wensen dus van een bepaalde kleur naar een andere kleur te gaan en dit op een continue manier. Hiervoor beschouwen we een kleur als een punt in $\mathbb{R}^3$. We noteren $\vec{k}_1(r_1, g_1 b_1)$ en $\vec{k}_2(r_2, g_2, b_2)$. De rechte door deze twee punten wordt gegeven door de parametervergelijking

$$\vec{p} = (1-t)\vec{k}_1 + t\vec{k}_2$$

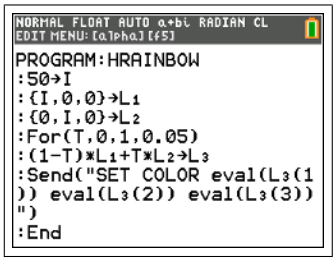
Maar deze parametervergelijking stelt meer voor. Indien $t \in [0,1]$ dan zal $\vec{p}$ een punt zijn dat op het lijnstuk $[\vec{k}_1, \vec{k}_2]$ ligt, we noemen $\vec{p}$ een lineaire interpolatie van $\vec{k}_1$ en $\vec{k}_2$. Voor $t = 0$ is dit $\vec{k}_1$, voor $t = 1$ is dit $\vec{k}_2$. Wanneer de parameter t van 0 naar 1 gaat, zal $\vec{p}$ het lijnstuk op continue wijze doorlopen van $\vec{k}_1$ tot $\vec{k}_2$. Op deze wijze stelt $\vec{p}$ een kleurovergang voor. We programmeren dit als volgt.

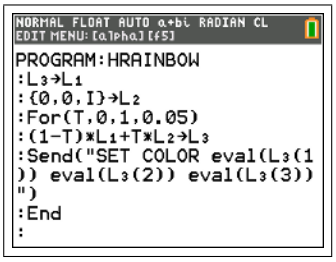


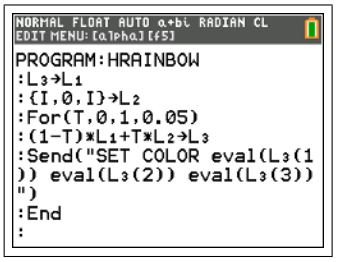
- Eerst worden twee kleuren (hier rood en groen) gegeven in de lijsten L_1 en L_2.
- Er wordt een `For`-lus gestart waarbij de parameter T van 0 tot 1 gaat met een kleine stap.
- De lineaire interpolatie wordt bepaald en onthouden in L_3.
- De waarden van L_3 worden gebruikt om de kleur LEDs aan te zetten.

Merk op dat de lineaire interpolatie wel een reeks getallen tussen 0 en 255 zal geven, maar dat deze niet noodzakelijk natuurlijke getallen zijn. Gelukkig is de **TI-Innovator Hub** hierop voorzien, alle waarden worden afgerond.

Nu we de overgang tussen twee kleuren kunnen maken, hoeven we voor een regenboog slechts van rood naar groen naar blauw naar violet te gaan. We gebruiken `2nd` `[rcl]` `[prgmHLIN]` om het vorige programma driemaal te kopiëren. Daarna passen we de kleuren aan. Met de parameter I bepalen we de intensiteit van de gebruikte kleuren.



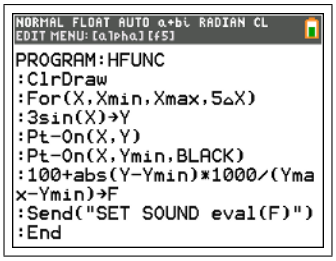


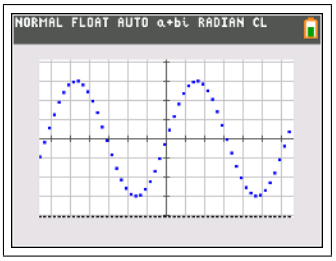


3 Geluid

3.1 Geluid van functies

De **TI-Innovator Hub** staat toe om een bepaalde toon op een zekere frequentie af te spelen. We zullen dit gebruiken om met een functie geluid te maken. Het programma ziet er als volgt uit.





3.2 De gelijkzwevende stemming

Reeds sinds Pythagoras heeft men getracht muziek in een wiskundig formalisme te beschrijven. Pythagoras merkte op dat een snaar aanslaan zorgde voor een welbepaalde toon. Wanneer de snaar wordt gehalveerd krijgt men een consonante toon, juist één octaaf hoger. Pythagoras ging verder de snaar opdelen volgens welbepaalde breuken en bekwam aldus de Pythagoraanse stemming. In deze stemming kent men de gewoonlijke toonladder die 17 eeuwen lang werd gebruikt. Pas aan het begin van de middeleeuwse polyfonie merkte men dat de verhoudingen niet voor een regelmatige verdeling van de toonhoogten zorgden.

In de moderne muziektheorie wordt de gelijkzwevende stemming gebruikt. Hierbij merkt men op dat bij verhoging van één octaaf de frequentie verdubbelt. Gezien er twaalf halve tonen zijn in één octaaf worden de frequenties per halve toon telkens vermenigvuldigd met $\sqrt[12]{2}$. We bekomen een meetkundige rij:

$$f_{n+1} = \sqrt[12]{2} f_n$$

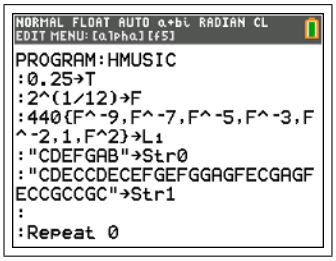
Het was ondermeer de bruggeling Simon Stevin die aan het einde van de 16de eeuw voor deze stemming pleitte. We zullen deze stemming ook gebruiken om met de **TI-Innovator Hub** muziek te spelen. Als standaard vertrekt men vaak van de la met frequentie 440Hz.

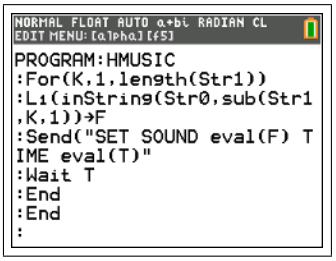
1	2	3	4	5	6	7	8	9	10	11	12
do	#do	re	#re	mi	#mi	fa	sol	#sol	la	#la	si
C	#C	D	#D	E	#E	F	G	#G	A	#A	B
$2^{-\frac{9}{12}}$	$2^{-\frac{8}{12}}$	$2^{-\frac{7}{12}}$	$2^{-\frac{6}{12}}$	$2^{-\frac{5}{12}}$	$2^{-\frac{4}{12}}$	$2^{-\frac{3}{12}}$	$2^{-\frac{2}{12}}$	$2^{-\frac{1}{12}}$	1	$2^{\frac{1}{12}}$	$2^{\frac{2}{12}}$

In ons programma zullen we ons beperken tot de hele noten:

do	re	mi	fa	sol	la	si
C	D	E	F	G	A	B
$2^{-\frac{9}{12}}$	$2^{-\frac{7}{12}}$	$2^{-\frac{5}{12}}$	$2^{-\frac{3}{12}}$	$2^{-\frac{2}{12}}$	1	$2^{\frac{2}{12}}$

Het programma ziet er als volgt uit:





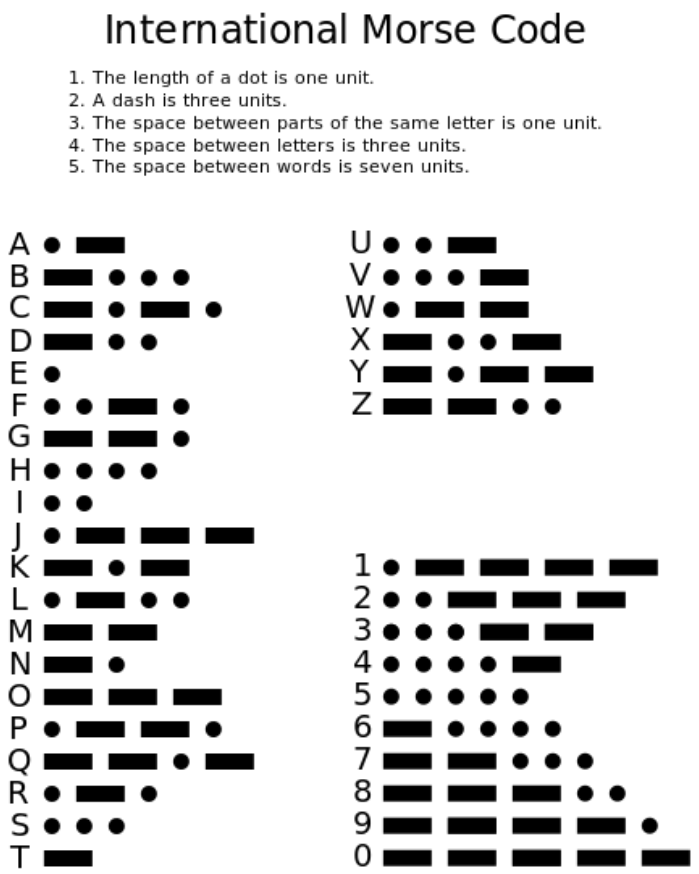
- Eerst wordt het tempo vastgelegd, elke noot zal $T = 0,25s$ duren. Het getal $\sqrt[12]{2}$ wordt eenmalig uitgerekend.
- De lijst L_1 zal de frequenties van de noten bevatten, de tekststring `Str0` de namen.
- In `Str1` komt het muziekstukje.
- Met `Repeat 0` wordt een oneindige lus gestart.
- Met een `For`-lus gaan we elke noot uit het muziekstukje af.
- `sub(Str1,K,1)` haalt de Kde noot uit `Str1`. Met `InString` wordt nagegaan welke positie de desbetreffende noot heeft in `Str0`. Deze positie wordt gebruikt als index voor L_1. Het resultaat is de frequentie F van de noot.
- Aan de hand van het bericht `SET SOUND eval(F) TIME eval(T)` wordt de noot afgespeeld, waarna het programma T seconden wacht vooralleer verder te gaan met de volgende noot.
- Uiteindelijk wordt ook de oneindige lus afgesloten en begint het muziekstukje opnieuw.

Dit programma leent zich uitstekend tot aanpassingen. Wat gebeurt er als het `Wait` commando wordt weggelaten? Kan je ook spaties toevoegen als pauzes? Hoe kan je de halve noten er toch in krijgen? ...

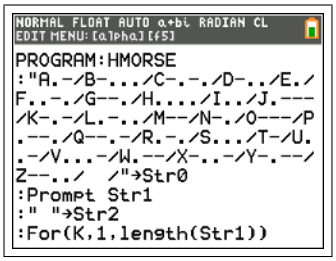
4 Licht en geluid

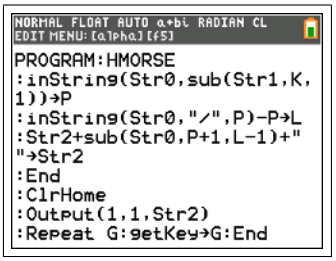
4.1 Cryptografie: Morsecode

Het Morse alfabet staat toe om met korte (.) en lange (-) signalen boodschappen door te zenden. Elke letter van het alfabet komt overeen met een reeks Morsetekens. De meest gebruikte letters hebben de kortste code (E=.) op deze manier is de Morsecode optimaal.



Onderstaand programma zet een willekeurige tekst om in Morse code.

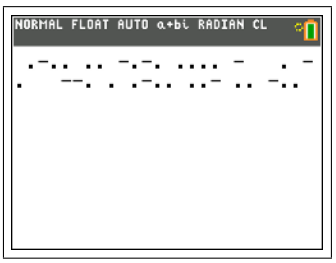




De werking is als volgt.

- Eerst wordt het volledige Morse alfabet opgeslagen in `Str0`. Let erop dat we de spatie ook omzetten in een spatie.
- De gebruiker kan nu een bericht ingeven. Dit wordt opgeslagen in `Str1`.
- `Str2` zal het bericht in Morsecode bevatten. Aanvankelijk is dit een lege tekststring.
- Met een `for`-lus gaan we de letters van het bericht af.
- Via `sub(Str1,K,1)` halen we de Kde letter uit het bericht. Met `inString` wordt bepaald waar de letter zich bevindt in het Morse alfabet, we onthouden deze positie in P.
- Met `inString(Str0,"/",P)` zoeken we vanaf de positie P de eerstvolgende /. Van deze positie trekken we P af, L is dus de lengte van de Morsecode voor de $k$de letter, tot aan de volgende /.
- We voegen nu de Morsecode voor de Kde letter (`sub(Str0,P+1,L-1)`) toe aan `Str2` samen met een spatie om de letters van elkaar af te zonderen. Merk op dat we hierbij de letter zelf (positie P en het eindteken / (op positie L verder) niet meenemen.
- We herhalen dit voor elke letter uit het bericht.
- Tenslotte tonen we de boodschap in Morsecode op een leeg scherm.
- Met een finale `Repeat`-lus wordt gewacht tot de gebruiker op een toets duwt.

Het ingeven van een bericht zal het gemakkelijkst gebeuren indien `2nd` `[A-lock]` wordt gebruikt.



We kunnen het programma nu uitbreiden om via de **TI-Innovator Hub** de Morsecode te verzenden als lichtsginaal en als geluidsignaal.

```
NORMAL FLOAT AUTO a+bi RADIAN CL
EDIT MENU: [alpha] [f5]
PROGRAM:HMORSE
:0.03→T
:Repeat 0
:For(K,1,length(Str2))
:If sub(Str2,K,1)="."
:Then
:Send("SET LIGHT  ON  TIME
 eval(T)")
:Send("SET SOUND 440 TIME
eval(T)")
```

```
NORMAL FLOAT AUTO a+bi RADIAN CL
EDIT MENU: [alpha] [f5]
PROGRAM:HMORSE
:Wait 2T
:End
:If sub(Str2,K,1)="-"
:Then
:Send("SET LIGHT  ON  TIME
 eval(3T)")
:Send("SET SOUND 440 TIME
eval(3T)")
:Wait 4T
```

```
NORMAL FLOAT AUTO a+bi RADIAN CL
EDIT MENU: [alpha] [f5]
PROGRAM:HMORSE
:End
:If sub(Str2,K,1)=" ":Wait
 2T
:End
:End
:
:
:
:
```

- Één tel T zal overeenkomen met $0,03$ seconden.
- Met `Repeat 0` wordt een oneindige lus gestart die het bericht zal blijven zenden.
- Opnieuw zorgt een `for`-lus dat alle tekens van het bericht worden afgegaan.
- Indien een `.` wordt gevonden, laten we het lichtje voor één tel branden. We zenden ook voor dezelfde periode een geluid uit aan $440Hz$. We wachten nadien twee tellen (één voor het signaal en één als pauze tussen twee signalen).
- Indien een `-` wordt gevonden, laten we het lichtje voor drie tellen branden. We zenden ook voor dezelfde periode een geluid uit aan $440Hz$. We wachten nadien vier tellen (drie voor het signaal en één als pauze tussen twee signalen).
- Indien een spatie wordt gevonden, wachten we twee tellen. Dit zorgt ervoor dat de pauzes tussen letters en woorden gerespecteerd worden.

4.2 De lichtsensor

Op de **TI-Innovator Hub** zit ook een lichtsensor. Met het bericht `READ BRIGHTNESS` en het commando `Get` kan de lichtintensiteit gemeten worden. Dit geeft een waarde tussen 0 en 100. In volgend programma gebruiken we dit om de intensiteit van de kleur LEDs te regelen als het donker wordt en bovendien ook een toon van het omgevingslicht te laten afhangen.

```
NORMAL FLOAT AUTO a+bi RADIAN CL
EDIT MENU: [alpha] [f5]
PROGRAM:HBRIGHT
:While 1
:Send("READ BRIGHTNESS")
:Get(B):Disp B
:(1-B/100)*255→I:10*B→F
:Send("SET COLOR eval(I) e
val(I) eval(I)")
:Send("SET SOUND eval(F)")

:End
```

- Een oneindige lus wordt gestart.
- De lichtintensiteit wordt gelezen, opgeslagen in B en afgedrukt.
- Met de formules wordt het licht I feller als donkerder het is, de toon F wordt dan lager.
- Tenslotte worden de kleur LEDs aangeschakeld en wordt de toon gespeeld.