

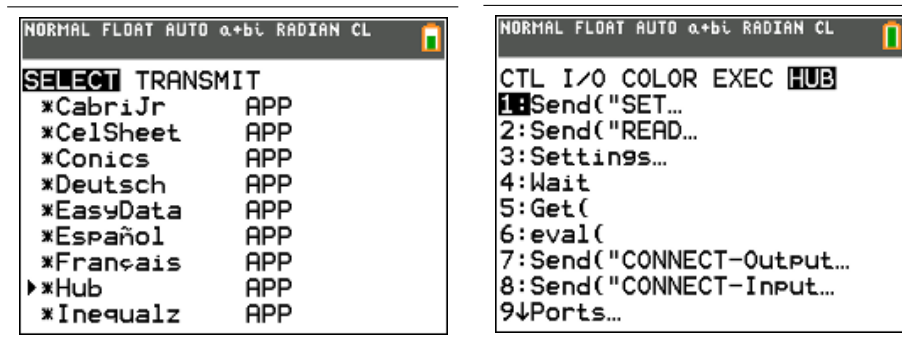
Introductie tot de **TI-Innovator Hub**

Inhoudsopgave

1	Inleiding	2
2	Licht	4
3	Geluid: random muziek	6

1 Inleiding

Voor je aan de slag kan met de **TI-Innovator Hub** moet je eerst de meegeleverde appHub op je **TI-84 Plus CE-T** zetten. Dit kan vanaf een computer of via een andere **TI-84 Plus CE-T** waarop het reeds geïnstalleerd is. Als eenmaal de app ook op jouw **TI-84 Plus CE-T** staat, zal ze het menu [prgm] in de programmeeditor uitbreiden met een nieuw submenu [HUB] waarin de commando's staan om met de **TI-Innovator Hub** te communiceren.



De TI-Innovator Hub heeft enkele basis functies ter beschikking. Deze werkende oplossingen zijn te programmeren:.. een RGB (Rood-Groen-Blauw) LED, een enkele rode LED, een helderheidssensor en ook een luidspreker zijn ingebouwd in de TI-Innovator™ Hub.

Inhoud van de doos

TI-Innovator™ Hub met ingebouwde onderdelen

- 1 Een sensor aan de onderkant van de Hub- voor het meten van de helderheid van het licht kan afgelezen worden als "HELDERHEID" in Hub- instructie-regels.
 - 2 Rode LED is adresseerbaar als "LICHT" in Hub- instructie-regels.
 - 3 Luidspreker (aan de achterkant van Hub-, niet afgebeeld) is adresseerbaar als "GELUID" in Hub- instructie-regels.
- Ã Rood-Groen-Blauwe LED is adresseerbaar als "KLEUR" in Hub- instructie-regels.

Tevens zichtbaar aan de voorkant van de hub zijn:

- A Groen LED-lampje voor externe voeding
- B Groen LED-lampje voor voeding,
- C Rood LED-lampje voor fouten.

Updaten van de Hub-software

De TI-Innovator™ Hub bevat software, TI-Innovator™ Sketch, die Hub-opdrachten interpreteert en communiceert met ingebouwde apparaten en aangesloten modules. Met een op on-line tool kunt u de update uitvoeren voor de Sketch. Geüpdatete versies bevatten bugfixes en zorgen ervoor dat uw TI-Innovator™ Hub kan communiceren met de nieuwste componenten.

Ga voor de nieuwste versie van de TI-Innovator™ Sketch naar de volgende website:

<https://education.ti.com/go/innovator>

Vragen over de Hub-software

Wat is de TI-Innovator™ sketch?

De 'sketch' is de software op de TI-Innovator™ Hub die communiceert met de grafische rekenmachine, de opdrachten verwerkt en de externe onderdelen bestuurt.

Moet ik de sketch bijwerken op de TI-Innovator™ Hub?

De beste resultaten krijgt u als u steeds gebruikmaakt van de meest recente versie van de TI-Innovator™-sketch. Als u op de hoogte wilt blijven van updates voor de TI-Innovator™ Hub, registreert u uw product bij education.ti.com/register of raadpleegt u de TI-Innovator-website op education.ti.com/go/innovator.

Wat is de meest recente versie van de sketch?

De beste resultaten krijgt u als u steeds gebruikmaakt van de meest recente versie van de TI-Innovator-sketch. U kunt altijd de meest recente versie van de sketch vinden op education.ti.com/go/innovator.

Waarom zou ik een update van de sketch uitvoeren?

Er zijn enkele verschillende redenen om een upgrade van de sketch uit te voeren.

1. Om de meest recente versie van TI te ontvangen die mogelijk nieuwe functionaliteit bevat.
2. Om de TI-sketch te herstellen na het laden van een aangepaste sketch – Dit is alleen nodig voor geavanceerde gebruikers die een alternatieve sketch gebruiken.

Hoe kan ik de sketch op de TI-Innovator™ Hub laden?

De sketch kan worden bijgewerkt via de updatesoftware voor de TI-Innovator Hub. Deze software kan gratis worden gedownload van de TI-website.

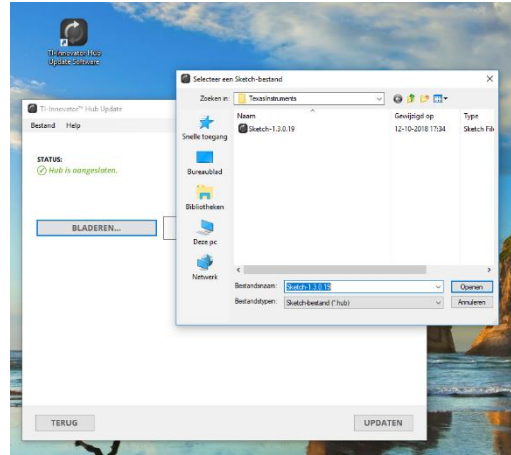
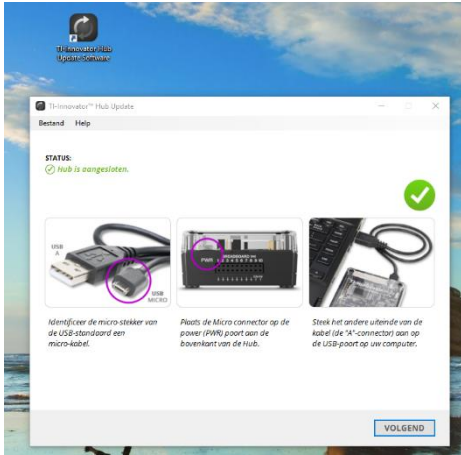
Kan ik meerdere TI-Innovator Hubs tegelijk bijwerken?

Met de updatesoftware voor de TI-Innovator Hub kan slechts één Hub tegelijk worden bijgewerkt. De toepassing is echter zodanig ontworpen dat u meerdere Hubs na elkaar kunt bijwerken zonder de software opnieuw te hoeven starten.

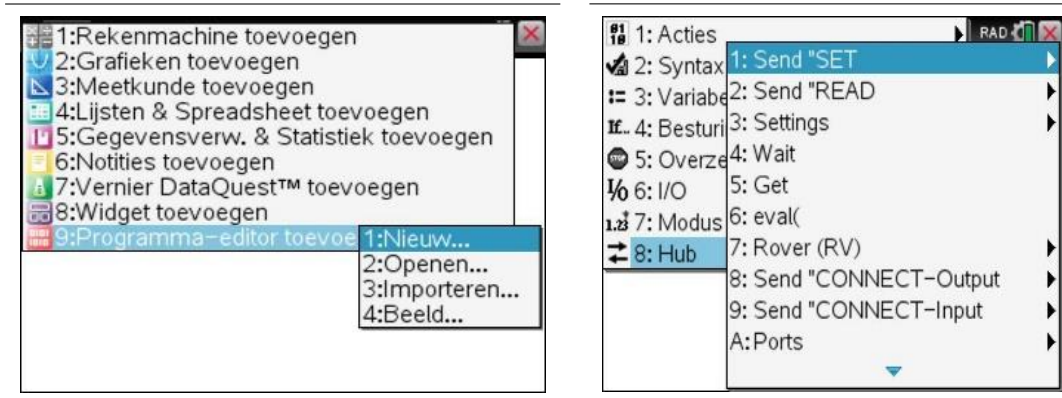
Kan de sketch die op de TI-Innovator™ Hub wordt geleverd worden bewerkt zodat functionaliteit wordt toegevoegd terwijl deze nog steeds werkt met de TI-rekenmachine? Is de sketch open source?

De code voor de sketch die op de TI-Innovator™ is geladen is niet gepubliceerd om te worden gewijzigd of bewerkt door anderen. Als u de compatibiliteit wilt handhaven tussen de TI-Innovator™ Hub en TI-rekenmachineproducten, gebruikt u alleen de officieel gepubliceerde sketch voor TI-Innovator™ Hub.

Wanneer de Ti-Innovator Hub wordt bijgewerkt moet deze eerst correct worden aangesloten aan de computer. Zie linker plaatje. Daarna kan het sketch programma gekozen worden(rechter plaatje) waarna de toets **updaten** gekozen dient te worden.



Voor de **TI-Nspire** moet je zorgen dat je besturingssysteem up-to-date is. Onder de programma-editor kan je een nieuw programma aanmaken. Onder de knop menu kan je dan een submenu [8 : Hub] terugvinden.



Nu kunnen we, zowel voor **TI-84 Plus CE-T** als **TI-Nspire**, 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 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 staan 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

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 opgevoerd 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).

Hoe doe je dit op de TI-84 Plus CE-T ?

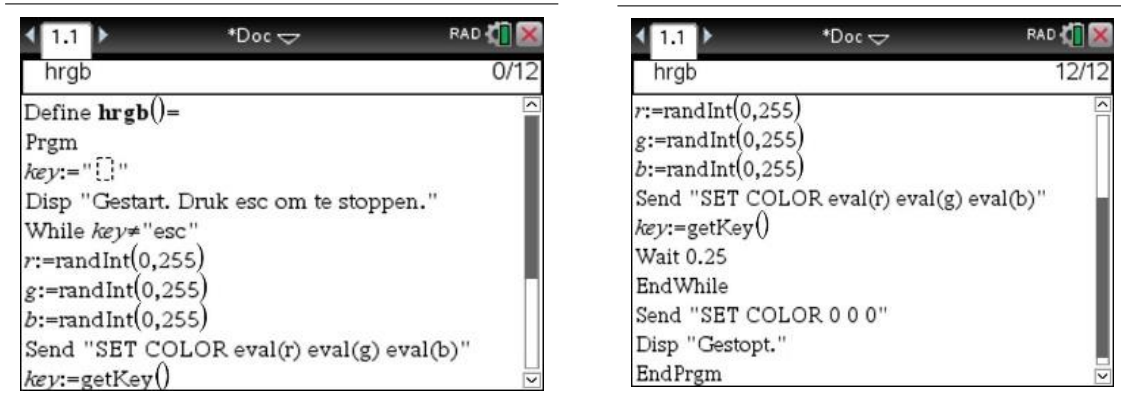
Met onderstaand programma wordt telkens een willekeurige waarde voor R , G en B genomen. We krijgen dan een disco knipperlichtje.

```
NORMAL FLOAT AUTO a+bi RADIAN CL
EDIT MENU: [alpha] [F5]
PROGRAM:HRGB
:While 1
:randInt(0,255)→R
:randInt(0,255)→G
:randInt(0,255)→B
:Send("SET COLOR eval(R) e
val(G) eval(B)")
:End
:█
```

- 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 en wordt de lus gesloten.

Hoe doe je dit op de TI-Nspire ?

Met onderstaand programma wordt ook telkens een willekeurige waarde voor R , G en B genomen. We bekomen een disco flikkerlichtje. We gaan net nog een stapje verder dan bij het programma uit de **TI-84 Plus CE-T** en laten het programma eindigen als op `esc` wordt gedrukt.



```
Define hrgb()=
Prgm
key:=""
Disp "Gestart. Druk esc om te stoppen."
While key≠"esc"
r:=randInt(0,255)
g:=randInt(0,255)
b:=randInt(0,255)
Send "SET COLOR eval(r) eval(g) eval(b)"
key:=getKey()
Wait 0.25
EndWhile
Send "SET COLOR 0 0 0"
Disp "Gestopt."
EndPrgm
```

- Met `While key = "esc"` wordt een lus gestart die blijft lopen zolang we niet op `esc` drukken. Daarom dat we de variabele `key` eerder in het programma ook al leegmaakten met `key:=""`.
- Daarna worden met `[randInt]` voor r , g en b willekeurige waarden tussen 0 en 255 gekozen.
- Vervolgens worden met `SET COLOR eval(r) eval(g) eval(b)` de kleur LEDs aangezet in de juiste kleurintensiteiten. We gebruiken `eval()` om de waarde uit de betreffende variabele te halen voor het commando `SET COLOR`.
- Daarna zullen we met `key:=getKey()` kijken of er op een knop is gedrukt. Denk er aan dat de voorwaarde om onze `While`-lus te sluiten hier aan gekoppeld is.
- Omdat we niet willen dat ons discolicht te snel verandert, laten we het programma 0,25 seconden wachten.
- Als we op `esc` drukken, wordt de lus doorbroken.
- Ten slotte gaan we met `SET COLOR 0 0 0` de LEDs weer uitzetten en displayen we dat het programma is gestopt.

3 Geluid: random muziek

De **TI-Innovator Hub** staat toe om een bepaalde toon op een zekere frequentie af te spelen.

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 krijgen 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	fa	#fa	sol	#sol	la	#la	si
C	#C	D	#D	E	F	#F	G	#G	A	#A	B
<u>9</u>	<u>8</u>	<u>7</u>	<u>6</u>	<u>5</u>	<u>4</u>	<u>3</u>	<u>2</u>	<u>1</u>		<u>1</u>	<u>2</u>

2^{-12} 2^{-12} 2^{-12} 2^{-12} 2^{-12} 2^{-12} 2^{-12} 2^{-12} 2^{-12} 2^{-12} 1 2^{12} 2^{12}

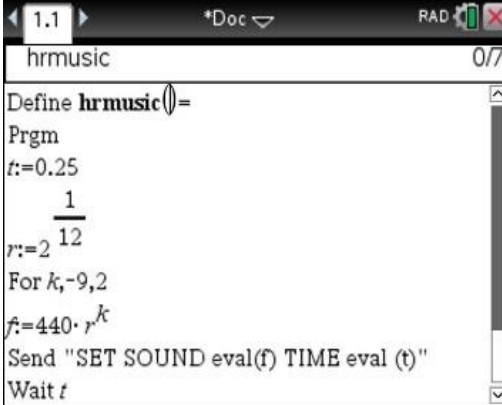
Hoe doe je dit op de TI-84 Plus CE-T ?

```
NORMAL FLOAT AUTO a+bi RADIAN CL
EDIT MENU: [alpha] [Phi] [F5]
PROGRAM:HRMUSIC
:0.25→T
:2^(1/12)→R
:While 1
:440R^randInt(-9,2)→F
:Send("SET SOUND eval(F) T
IME eval(T)")
:Wait T
:End
```

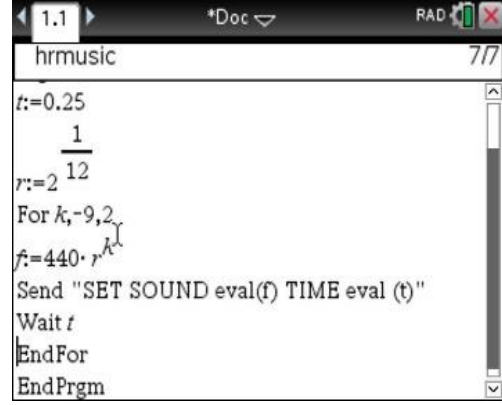
- Eerst wordt het tempo vastgelegd, elke noot zal $T = 0,25$ s duren. Het getal $\sqrt[12]{2}$ wordt eenmalig uitgerekend.
- Met `While 1` wordt een oneindige lus gestart.
- Er wordt een willekeurig getal gekozen tussen -9 en 2 , waarmee de frequentie van de noot wordt berekend.
- De noot wordt gedurende een tijd T afgespeeld, het programma wacht vooraleer verder te gaan met de lus.

Hoe doe je dit op de TI-Nspire ?

In het voorbeeld voor de **TI-Nspire** werken we niet met een oneindige `while` maar met een `for` om alle 12 tonen van de toonladder af te gaan.



```
1.1 *Doc RAD  
hrmusic 0/7  
Define hrmusic()=  
Prgm  
t:=0.25  
  1  
r:=2 12  
For k,-9,2  
  f:=440·rk  
  Send "SET SOUND eval(f) TIME eval (t)"  
  Wait t  
EndPrgm
```



```
1.1 *Doc RAD  
hrmusic 7/7  
t:=0.25  
  1  
r:=2 12  
For k,-9,2  
  f:=440·rk  
  Send "SET SOUND eval(f) TIME eval (t)"  
  Wait t  
EndFor  
EndPrgm
```

Binnen elk programma kan een ander programma worden aangeroepen.
Als voorbeeld het onderstaand programma om uiteindelijk het morse geluid weer te geven op de Ti-innovator Hub.

Programma MSOUND

```
Input " Geef tijdeenheid (in s)" ,T
Input " Geef toonhoogte (in Hz)" ,F
```

```
lengte(Str5)→A
For(I,1,A
Output(8,3,sub(Str5,I,1)
If sub(Str5,I,1)="/"
Wait 3*T

If sub(Str5,I,1)="."
Then
Send("SET SOUND eval(F) TIME eval(T)")
Wait T
End
If sub(Str5,I,1)="-"
Then
Send("SET SOUND eval(F) TIME eval(3*T)")
Wait 3*T
End
End
End
WisHome
```

Vul van te voren wel STR5

Hub-Programmeren op de TI CE Grafische rekenmachine

Opmerking: Deze instructies zijn van toepassing op de TI CE grafische rekenmachine. Voor vergelijkbare instructies voor TI-Nspire™ CX-technologie, zie volgende pagina.

De TI-Innovator™ Hub reageert op TI Basic-programmeerinstructies, zoals **Send** en **Get**.

- **Send** - Verstuur instructiereeksen naar de Hub- voor het besturen van apparaten of het opvragen van informatie.
- **Get** - Haalt informatie op die opgevraagd is vanaf de Hub-.
- **eval** - levert het resultaat van een expressie als een tekenreeks. Vooral bruikbaar binnen de Hub-instructiereeks in **Send**-instructies.
- **Wait** - Pauzeert de uitvoering van een programma gedurende een ingesteld aantal seconden.

Voorbeelden van code: TI CE Grafische rekenmachine

Gewenste actie	Programmacode
Zet het ingebouwde rode LED aan ("LIGHT").	Send ("SET LIGHT ON")
Speel een toon op 440Hz tone af via de ingebouwde luidspreker ("SOUND") gedurende 2 seconden.	Send ("SET SOUND 440 TIME 2")
Zet de blauwe ingebouwde RGB LED ("COLOR") op een helderheid van 100%.	Send ("SET COLOR.BLUE 255")
Lees en toon de huidige waarde van de lichtsensor van het board ("BRIGHTNESS"). De reikwijdte is 0% t/m 100%.	Send ("READ BRIGHTNESS") Get (A) :Disp A

Hub-Programmeren op basis van TI-Nspire™ CX-technologie

Opmerking: Deze instructies gelden voor de TI-Nspire™ CX-technologie. Voor vergelijkbare instructies voor de TI CE grafische rekenmachine, zie vorige pagina.

De TI-Innovator™ Hub reageert op TI Basic-programmeerinstructies, zoals **Send** en **Get**.

- **Send** - Verstuur instructiereeksen naar de Hub- voor het besturen van apparaten of het opvragen van informatie.
- **Get** en **GetStr** - halen informatie op die aangevraagd is van de Hub-.
- **eval()** - levert het resultaat van een expressie als een tekenreeks. Alleen geldig in de instructies **Send**, **Get**, en **GetStr**.
- **Wait** - Pauzeert de uitvoering van een programma gedurende een ingesteld aantal seconden.

Voorbeelden van code: TI-Nspire™ CX-technologie

Gewenste actie	Programmacode
Zet het ingebouwde rode LED aan ("LIGHT").	Send "SET LIGHT ON"
Speel een toon op 440Hz tone af via de ingebouwde luidspreker ("SOUND") gedurende 2 seconden.	Send "SET SOUND 440 TIME 2"
Zet de blauwe ingebouwde RGB LED ("COLOR") op een helderheid van 100%.	Send "SET COLOR.BLUE 255"
Lees en toon de huidige waarde van de lichtsensor van het board ("BRIGHTNESS"). De reikwijdte is 0% t/m 100%.	Send "READ BRIGHTNESS" Get a: Disp a

Het uiteindelijke programma voor morse codes zelf bestaat eigenlijk uit twee onderdelen.

- 1) Invoeren van tekst welke omgezet moet worden in morse code
- 2) Morse code omzetten naar tekst

Voor beide geldt dat dit kan met behulp van de Ti Innovator Hub middels geluid of licht.

Er dienen een aantal zaken gedefinieerd te worden zoals bijvoorbeeld het vast leggen van de karakters (A t/m Z en 0 t/m 9)

```
Input "Bericht ?",Str1
lengte(Str1→L
Output(4,3,"OMZETTEN...
For(A,1,L
sub(Str1,A,1→ Str2
If Str2="A
".- → Str4
If Str2="B
"-...→ Str4
If Str2="C
"-.-.→ Str4
etc.....
```

De lengte van de code dient bepaald te worden en kan bijvoorbeeld middels onderstaande structuur:

```
If lengte(Str4):Then
If lengte(Str5):Then
Str5+Str4 → Str5
Else
Str4 → Str5
End
```

Als dit allemaal vastgelegd is kan er binnen het hoofd programma een ander programma worden aangeroepen zoals het eerder genoemde MSOUND.

```
.....
Prgm MSOUND
Disp Str5
```