Programmeren van Jota Game board

Introductie

Deze handleiding beschrijft hoe je de PIC Microcontroller op het Jota Game board – opnieuw – kunt programmeren.

Benodigdheden

Hiervoor heb je drie zaken nodig:

1) Een programmer voor de PIC Microcontroller
2) Software voor de PIC programmer
3) Software image voor de PIC Microcontroller, de Jota Game board software

De software voor het Jota Game board is geschreven in JAL (Just Another Language). Als je de software voor het Jota Game board wilt wijzigen dien je ook de broncode (JAL bestanden) van het Jota Game board en de JAL libraries met JAL compiler te hebben. Zie verderop in deze handleiding waar je die kunt downloaden.

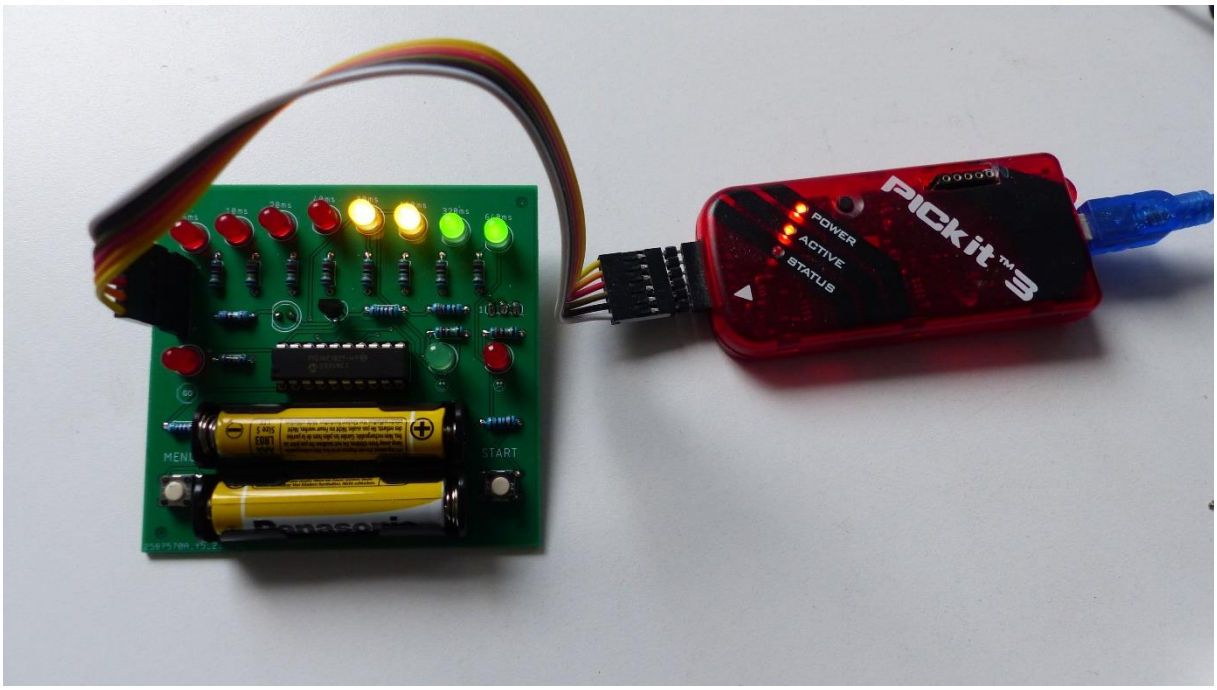
Programmer

In deze handleiding wordt gebruik gemaakt van de PICKit3. Deze programmer wordt niet meer door Microchip ondersteund maar is nog zeker op veel plaatsen te koop en prima te gebruiken voor het programmeren van de PIC Microcontroller die op het Jota Game board zit. De programmmer heeft de volgende aansluitingen:

Legend:

1 – Lanyard Loop
2 – USB Port Connection
3 – Pin 1 Marker
4 – Programming Connector
5 – Indicator LEDs
6 – Push Button

De programmer wordt over het algemeen geleverd met een USB kabel voor aansluiting aan je PC en een platte kabel die je aansluit aan de Programming Connector (nr. 4 in het voorgaande plaatje). Deze platte kabel sluit je straks 1 op 1 aan op connector SV1 van het Jota Game board. Dat ziet er dan als volgt uit:



Software voor de programmer

Zoals aangegeven wordt deze software niet meer door Microchip ondersteund maar je kunt hier nog steeds de PIC mee progammeren die op het Jota Game Board zit.

De gratis Windows programmer software van Microchip is moeilijk te vinden. Daarom kun je deze ook downloaden van de Jallib op GitHub: https://github.com/jallib/jallib/tree/master/tools/pickit3

Software image voor de PIC

Dit is een zogenaamde hexfile die je inleest in de programmer software, die deze dan gebruikt om de PIC te programmeren.

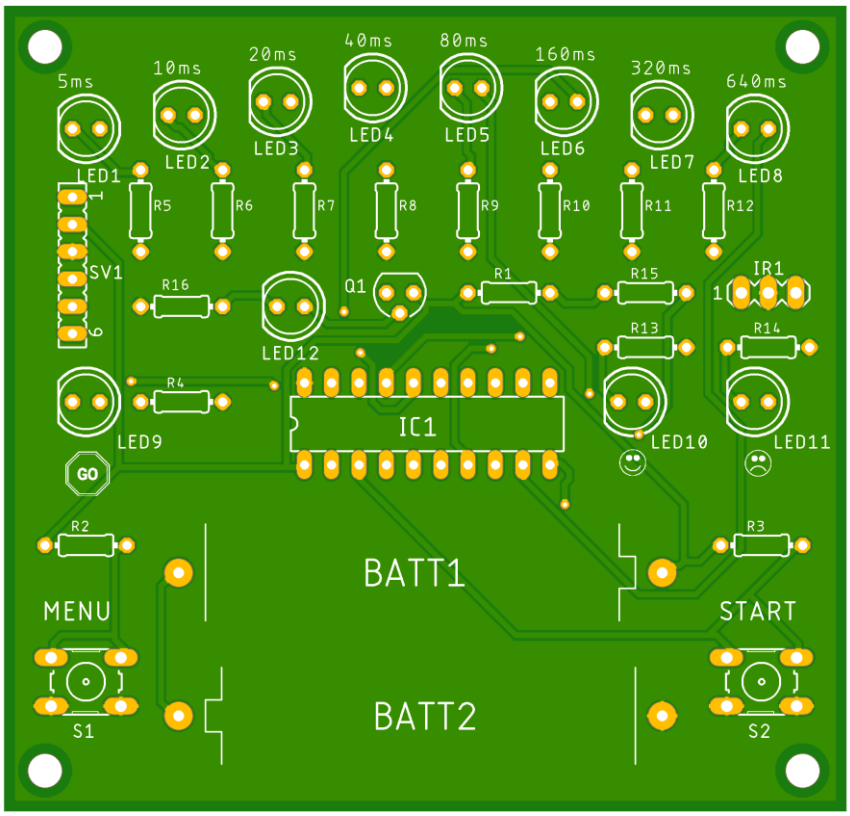
Alles broncode van het Jota Game board en de bijbehorende hexfile kun je downloaden van GitHub via deze link: https://github.com/RobJansen62/Jota_Game/tree/main/software

Het hex bestand wat je nodig hebt heet: **16f1829_jota_game.hex**

Programmeren van de PIC Microcontroller

Verbindt de PIC programmer via de USB kabel met de PC.

Sluit dan de programmer met de platte kabel aan op het Jota Game board aan. Let er hierbij op dat pin 1 van de programmer (zie nr. 3 van het plaatje terug) verbonden wordt met pin 1 van de connector SV1 op het Jota Board. Dit is de positie net onder LED1.

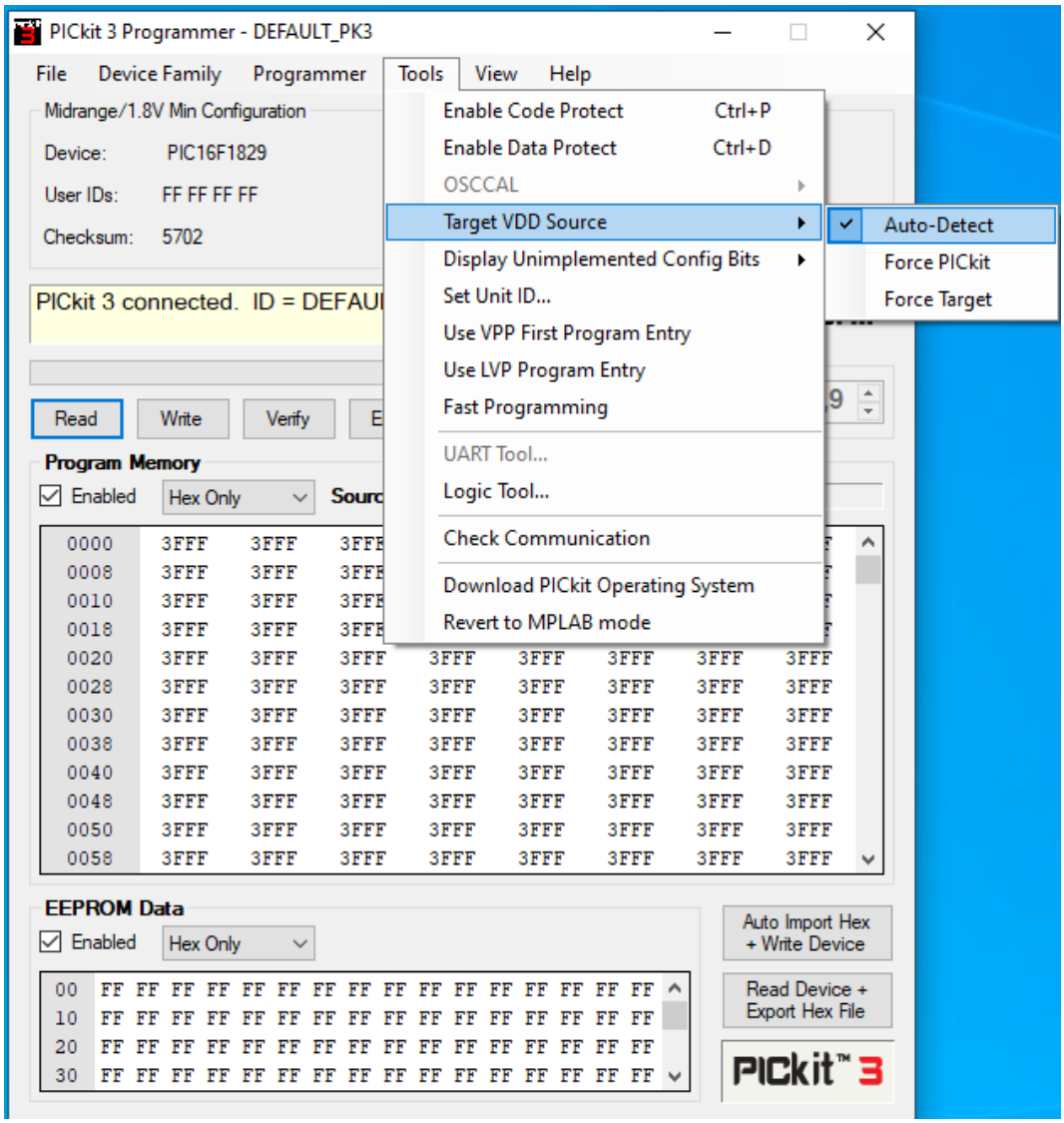


Er zijn 2 manieren om de PIC te programmeren:

1) Zonder geinstalleerde batterijen. De programmer levert dan de spanning om de PIC te programmeren. De programmer levert echter niet voldoende vermogen om het hele Jota Game board te laten werken.
2) Met geinstalleerde batterijen. De batterijen leveren dan de spanning om de PIC te programeren. Deze optie dien je te gebruiken als je naast het programmeren ook de software even wilt testen voordat je de programmer loskoppelt.

Je kunt de programmersoftware zo instellen dat hij automatisch detecteert of er batterijen geinstalleerd zijn of niet.

Start de programmer software. Voordat we de PIC kunnen programmeren moeten we eerst de programmersoftware zo instellen dat hij automatisch detecteert of er spanning op het Jota Game board staat of niet. Dit kun je aangeven via **Tools → Target VDD Source → Auto-Detect**.

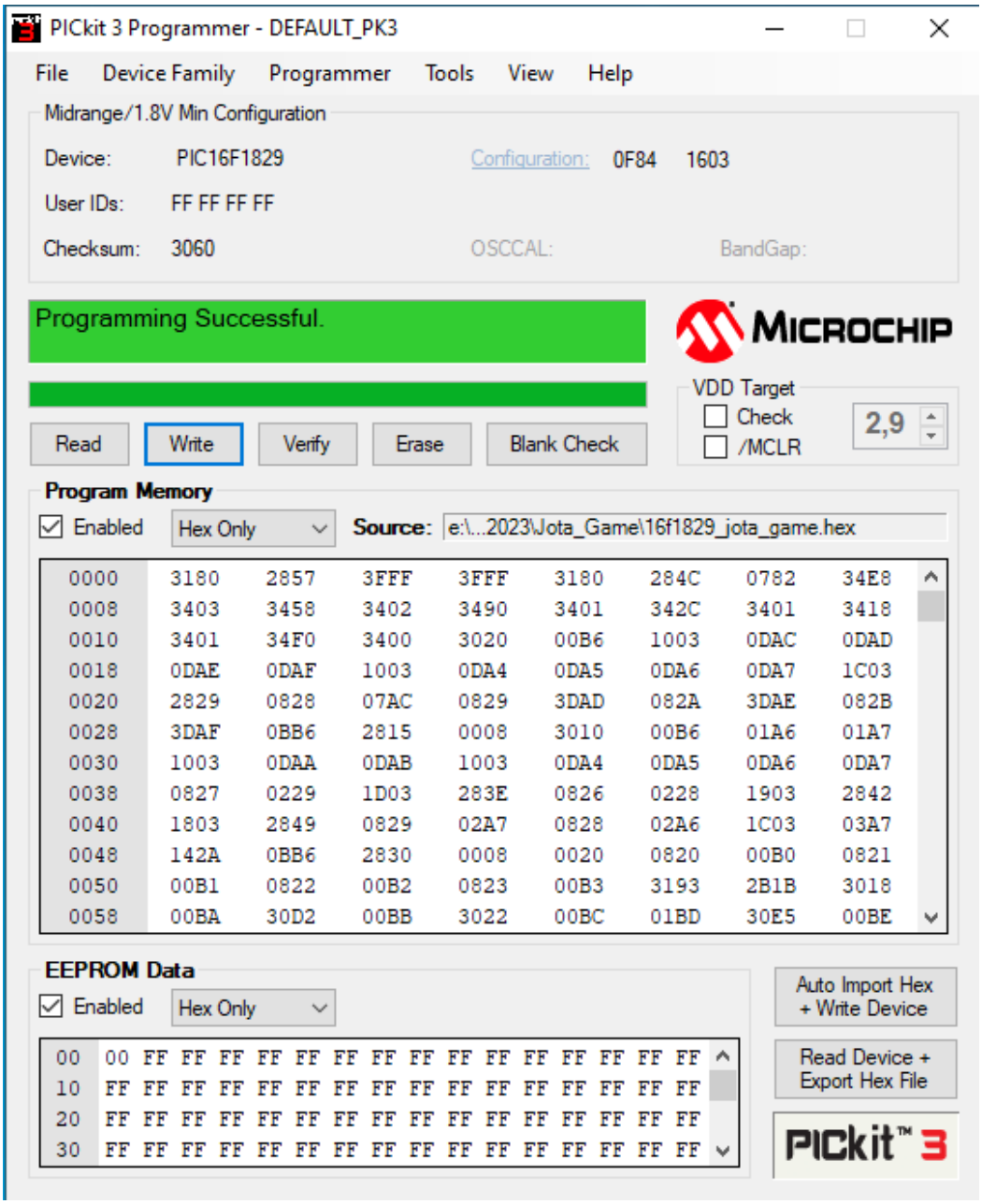


Voordat je een nieuw programma laadt dien je eerst het oude programma in de PIC te wissen met de **Erase** knop.

Laadt nu het bestand **16f1829_jota_game.hex** in via **File → Import Hex file**.

Druk na het succesvol inladen van de hexfile op de **Write** knop. De PIC wordt nu geprogrammeerd en er wordt ook gecontroleerd dat het programmeren gelukt is.

Het resultaat ziet dan als volg uit:



Als het programmeren om 1 of andere reden mislukt, wis de PIC dan, laadt het bestand opnieuw in en programmeer de PIC nog een keer.

Wijzigen van de Jota Board software

Hiervoor moet je de JAL libraries en de JAL compiler installeren op je computer.

Om snel een indruk te krijgen van de installatie en het gebruik van JAL, kijk dan even naar deze video: https://www.youtube.com/watch?v=FMLX5y0D5HE

Je hebt het volgende nodig:

- De Jallib software bestaande uit libraries, sample programma's en de JAL compiler. Deze kun je downloaden van de JAL Website: http://justanotherlanguage.org/downloads
- Als je de laatst gereleaste versie download kun je kiezen uit een Windows Installer of een zip-file. Als je de allernieuwste maar nog niet gereleaste versie wilt hebben dan kun je de laatst zogenaamde bee-package downloaden. Dit is dan alleen een zipfile.
  Run de Window installer of pak de zipfile uit en zet de Jallib directory op een eigen gekozen locatie op je computer.
- De JAL broncode van het huidige Jota Game Board. Deze kun je downloaden van GitHub. Daar vind je ook andere documentatie zoals die van het board, etc. In de 'software' directory vind je alle broncode, zie: https://github.com/RobJansen62/Jota_Game/tree/main/software

Broncode aanpassen

Als je een Windows gebruiker bent dan kun je gebruik maken van het programma JALEdit wat – alleen – bij de laatste release wordt meegeleverd en niet in de zip-file. Je kunt ook gebruik maken van Visual Studio Code wat naast Windows ook beschikbaar is voor Linux gebruikers en MAC gebruikers. In de 'doc' folder van de Jallib installatie vind je het **Jallib_Tutrial_Book**. Daarin staat o.a. wat je moet doen om JAL met Visual Studio Code te gebruiken. Er is namelijk een JAL extension voor Visual Studio Code beschikbaar waarmee je o.a. syntax highlighting hebt wat helpt bij het programmeren.

Uiteraard kun je ook een gewone platte text editor gebruiken zoals Notepad++.

Wat kun je aanpassen?

Omdat je de broncode hebt kun je alles aanpassen, zo kun je bijvoorbeeld:

- Je eigen LED patroon maken.
- De moelijkheidsgraad van sommige spelletjes aanpassen. Voor bijvoorbeld het spel 'Simon Says' is het maximaal aantal LEDs dat je moet onthouden op 8 gezet. In het JAL broncodebestand **games_and_actions.jal** kun je dit aantal vergroten door de constante **MAX_SIMON_SAYS** te veranderen van 8 in bijvoorbeeld 10.
- De werking van een spelletje aanpassen.
- Je eigen spel maken, met of zonder infra-rood communicatie.

Voorbeeld van een softwareaanpassing

Het lopend menu heeft 8 keuzemogelijkheden. Bij het maken van de software was ook een reactietijdspel gemaakt net als het Jota Game maar dan zonder dat je binnen een vooraf bepaalde tijd moest reageren. Omdat echter zeeslag was toegevoegd, is dit spel niet in gebruik. De software bestaat echter nog wel. Laten we als voorbeeld het Jota Game van de keuzemogelijkheden uit het lopend menu veranderen zodat we dit reactiespel kunnen gebruiken in plaats van het Jota Game. Het spel heet **reaction_timer** en bevindt zich in het bestand **games_and_actions.jal**. De software voor het afhandelen van de actie uit het lopend menu is te vinden in het hoofdprogramma **16f1829_jota_game.jal**.

In het hoofdprogramma vind je dit stukje code:

```
MENU_5:
block
   jota_game()
   menu_selection = MENU_MENU
end block
```

Als je hier de code **jota_game()** verandert in **reaction_timer()** dan wordt bij keuze menu 5 een ander spel gestart. Wat je hier doet is dat je een ander stukje programma aanroept en daarmee in dit geval een ander spel.

Broncode compileren

Als je jouw wijziging(en) wilt testen moet je de aangepaste broncode opnieuw compileren om een nieuwe hexfile te maken. Je moet altijd het hoofdprogramma **16f1829_jota_game.jal** compileren. Na succevol compileren wordt een hexfile aangemaakt met de naam **16f1829_jota_game.hex** .

Er zijn JAL compilers beschikbaar voor Windows, Linux en MAC. Kijk in de README.txt bestand in de Jallib 'compiler' directory hoe deze compilers heten.

In het Jallib_Tutorial_Book staat hoe je de compiler kunt starten vanuit JALEdit en Visual Studio Code. Als je gebruik maakt van een platte text editor, zoals Notepad++, start je het compileren voor Windows als volgt (pad aanpassen naar het pad waar je Jallib en de source files hebt hebt gezet):

C:\Jallib\compiler\jalv2_64.exe "C:\Jota_Game\16f1829_jota_game.jal" -s "C:\Jalllib\lib"

Testen van je wijzigingen

Na het aanpassen en succesvol compileren van je programma kun je de nieuwe software testen. Zorg ervoor dat er battijen in het Jota Game board zitten.

Volg nu weer de stappen zoals eerder beschreven:

1. Wis de PIC Microcontroller met de **Erase** knop
2. Laad je nieuwe hexfile in de programmersoftware
3. Programmeer de PIC Microcontroller met de **Write** knop

Het programma gaat nu vanzelf lopen en kun je jouw wijzigingen testen. Je kunt de programmer aangesloten laten totdat je zeker weet dat alles naar behoren werkt. Als dat gelukt is dan kun je de programmer loskoppelen.

EEPROM data en het infra-rood adres

Voor de infra-rood communicatie wordt een adres gebruikt. Dit adres staat standaard op 1 en kan door de gebruiker worden aangepast, zie hiervoor de handleiding van het Jota Game board. Het adres wordt opgeslagen in de EEPROM van de PIC Microcontroller. Bij het (her) programmeren van de PIC Microcontroller wordt de EEPROM gewist waarna deze weer na het opstarten van het Jota Game board door de software automatisch op adres 1 wordt gezet door de Jota Game software.

Broadcast adres

Het morse spel heeft de mogelijheid om op meer dan 1 adres te reageren. Normaal gesproken reageert het spel alleen op het door de gebruiker ingestelde adres. Er is een speciaal broadcast adres (adres 0) waar alle Jota Game board op reageren voor het morse spel. Dit adres is niet met de hand in te stellen maar kan alleen ingesteld worden met behulp van de programmer software.

In de source code van het bestand **16f1829_jota_game.jal** vind je dit stukje code:

```
; Special code. If you want to use the BROADCAST_ADDRESS address for the morse
; game then remove the comments before the pragma of the following line.
; pragma eedata BROADCAST_ADDRESS; This will place it at EEPROM address 0.
```

Door het de punt comma (;) weg te halen voor het **pragma** commando en door het vinkje van **EEPROM Data** in de progammer software aan te zetten, zal adres 0 worden opgeslagen in EEPROM. Dit is alleen nodig voor de zender van de morse codes want de ontvangers reageren altijd op hun eigen adres en op dit broadcast adres.

Door gebruik te maken van dit broadcast adres is het dus niet nodig dat alle ontvangers van het morse spel hetzelfde adres hoeven te hebben dus kun je dit spel spelen met een groep.

Voor het zeeslag spel werkt het broadcast adres niet. Dat komt omdat je bij zeeslag altijd 1 op 1 speelt met iemand anders en niet met een groep.

Meer informatie over JAL

Je vindt meer informatie over JAL op de JAL Website: http://justanotherlanguage.org/

Als je vragen hebt over het gebruik van JAL, stel die dan op de **Jallist Google Group**. Dit is een internationale groep dus je vragen moet je in het Engels stellen.

Er is ook nog een **Jallib Google Group**. Deze is bedoeld om problemen met JAL libraries of de JAL compiler te melden. Ook dit is een internationale groep.

Veel plezier met het programmeren van het Jota Game board en met programmeren van de PIC Microcontroller met de taal JAL!