Programming the Jota Game board

Introduction

This manual describes how to (re-)program the PIC Microcontroller on the Jota Game board.

Supplies

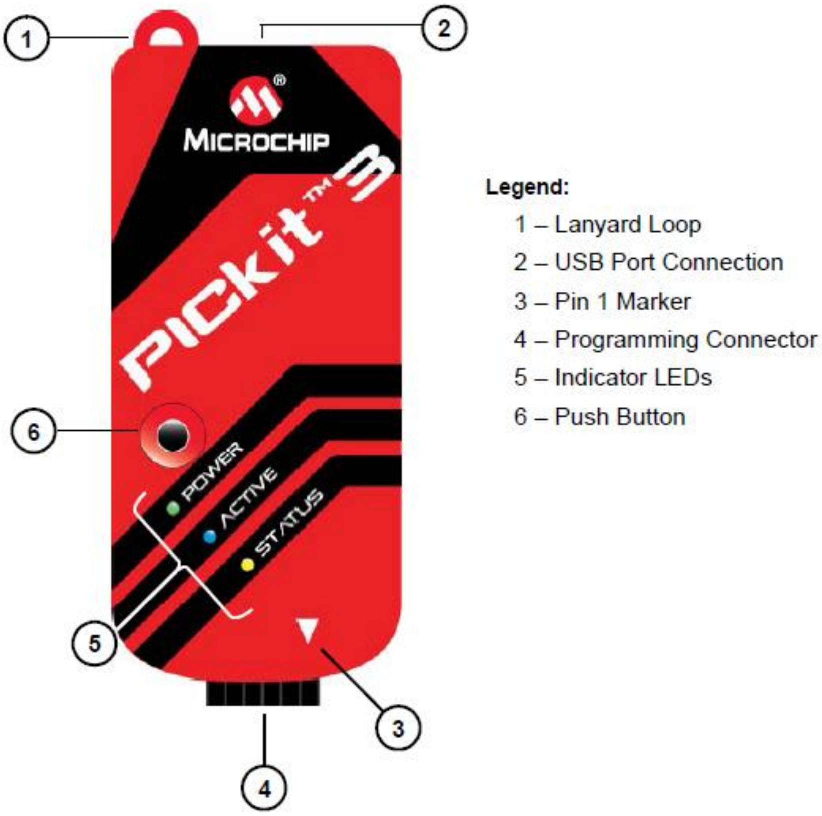
You need to have the following three items:

1) A programmer for the PIC Microcontroller
2) Software for the PIC programmer
3) Software image for the PIC Microcontroller, the Jota Game board software

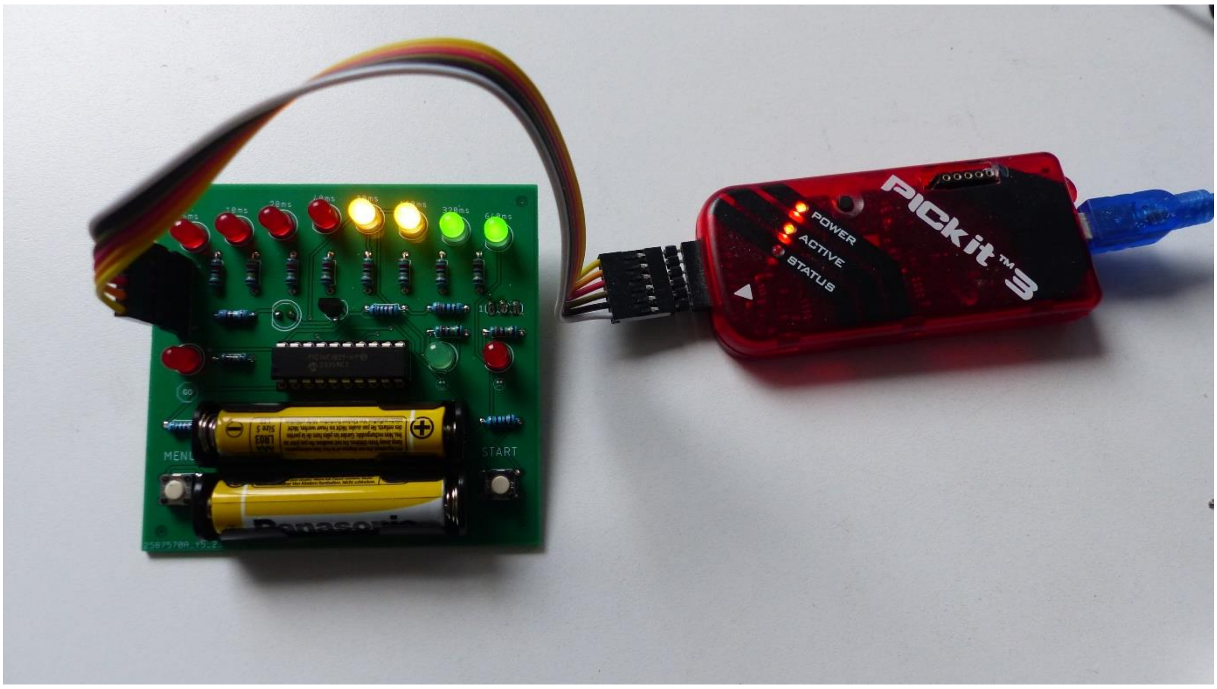
The software of the Jota Game board is written in the JAL programming language (Just Another Language). If you want to modify the software for the Jota Game board you need to have the source files (JAL files) of the Jota Game board including the JAL libraries and the JAL compiler. In the next sections you wil find the information where you can download these files.

Programmer

In this manual the PICKit3 programmer is used. This programmer is no longer supported by Microchip but can still be purchased from various sources and can be used the (re-) program the PIC Microcontroller on the Jota Game board. The programmer has the following connections:



The programmer is normally supplied with a USB cable to connect the programmer to your PC. Also a flat cable is provided which you need to connect to the Programming Connector (number 4 in the previous picture). This flat cable is then connected directly to connector SV1 on the Jota Game Board. This then looks as follows:



Software for the programmer

As mentioned earlier the software for this programmer is no longer supported by Microchip but you can still use it to program the PIC on the Jota Game board.

The fre Windows programmer software is not easy to find on the Interne. That's why you can also download a copy of this software from the Jallib repository on GitHub: https://github.com/jallib/jallib/tree/master/tools/pickit3

Software image for the PIC

This image is called a hexfile and you need to load it into the programmer software which uses it to program the PIC Microcontroller.

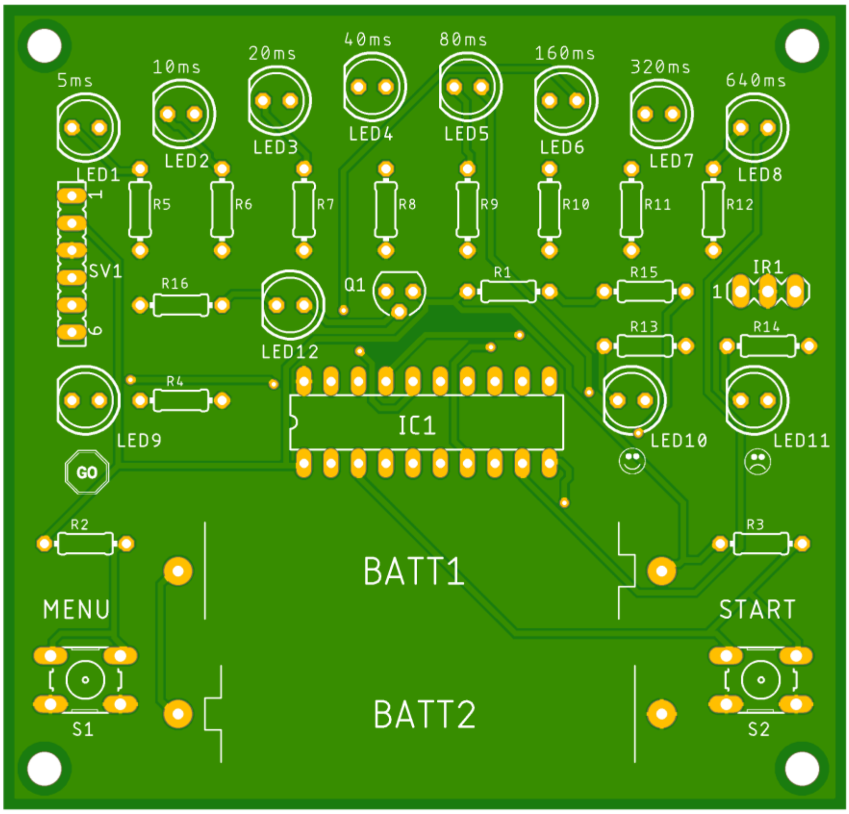
All source files of the Jota Game board including the hexfile can be download from GitHub using this link: https://github.com/RobJansen62/Jota_Game/tree/main/software

The hexfile is called: **16f1829_jota_game.hex**

Programming the PIC Microcontroller

Connect the PIC programmer to the PC using the USB cable.

Connect the flat cable between the programmer and the connector SV1 on the Jota Game board. Make sure that pin 1 of the programmer (see number 3 in the picture shown earlier) is connected to pin 1 of connector SV1 on the Jota Game board. Pin 1 is the pin just below LED1.

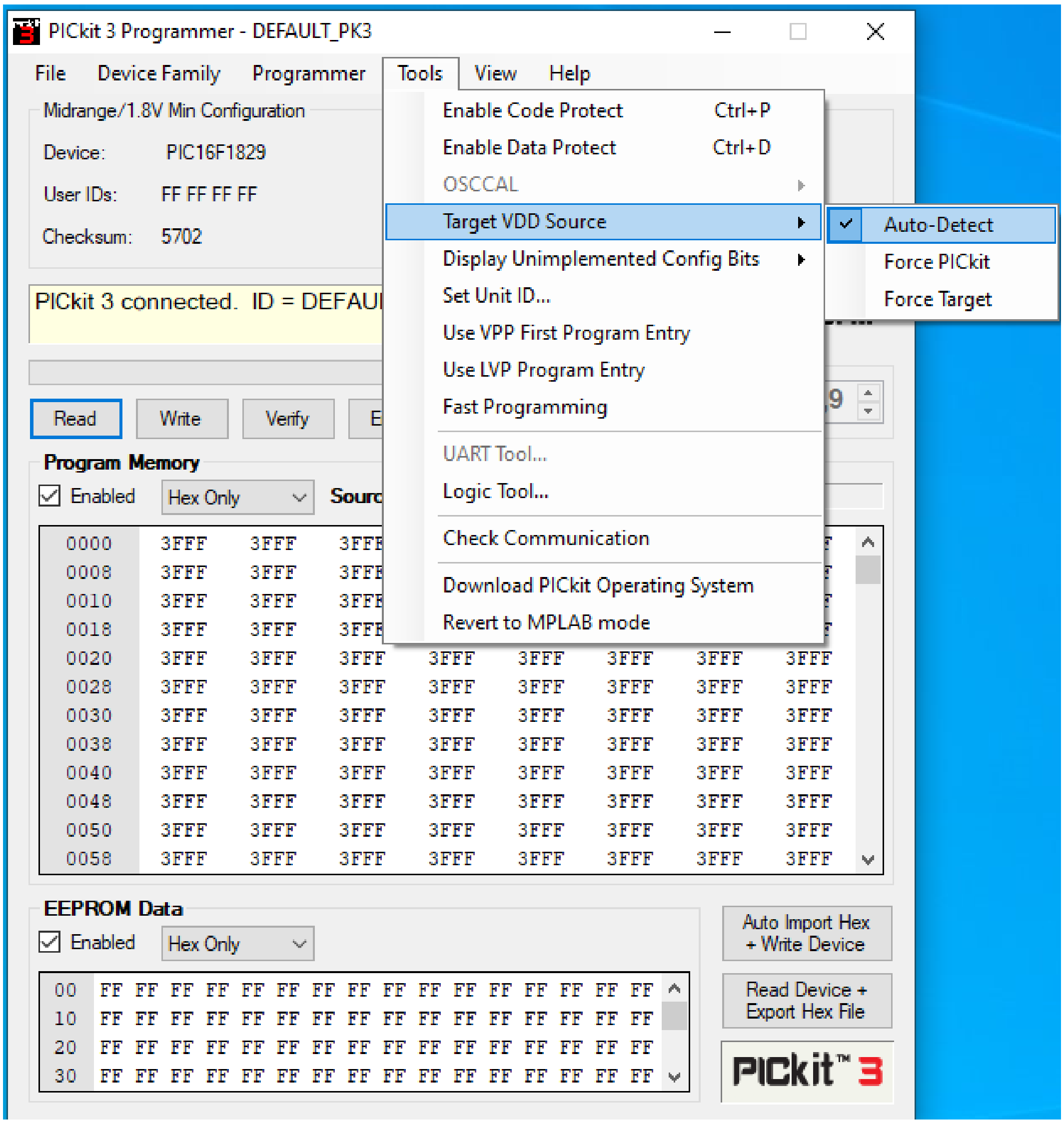


There are 2 ways to program the PIC Microcontroller:

1) Without installed batteries. In this situation the programmer will supply the required voltage to program the PIC. The programmer is not capable of supplying enough power to operate the Jota Game board.
2) With batteries installed. In this situation the batteries supply the voltage to program the PIC. You need to use tins option if you want to test the software by using the Jota Game board before you disconnect the programmer.

You can configure the PIC programmer software in such a way that it will automatically detect if there are batteries installed or not

Start the programmer software. Before we can program the PIC we need to configure the programmer software in such a way that it automatically detects if there is already a supply voltage present on the Jota Gam board. You can selec this via: **Tools → Target VDD Source → Auto-Detect**.



Midrange/1.8V Min Configuration

Device: PIC16F1829

User IDs: FF FF FF FF

Checksum: 5702

PICkit 3 connected. ID = DEFAU

Program Memory: ☑ Enabled Hex Only

```
0000  3FFF  3FFF  3FFF
0008  3FFF  3FFF  3FFF
0010  3FFF  3FFF  3FFF
0018  3FFF  3FFF  3FFF
0020  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0028  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0030  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0038  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0040  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0048  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0050  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0058  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
```

EEPROM Data: ☑ Enabled Hex Only

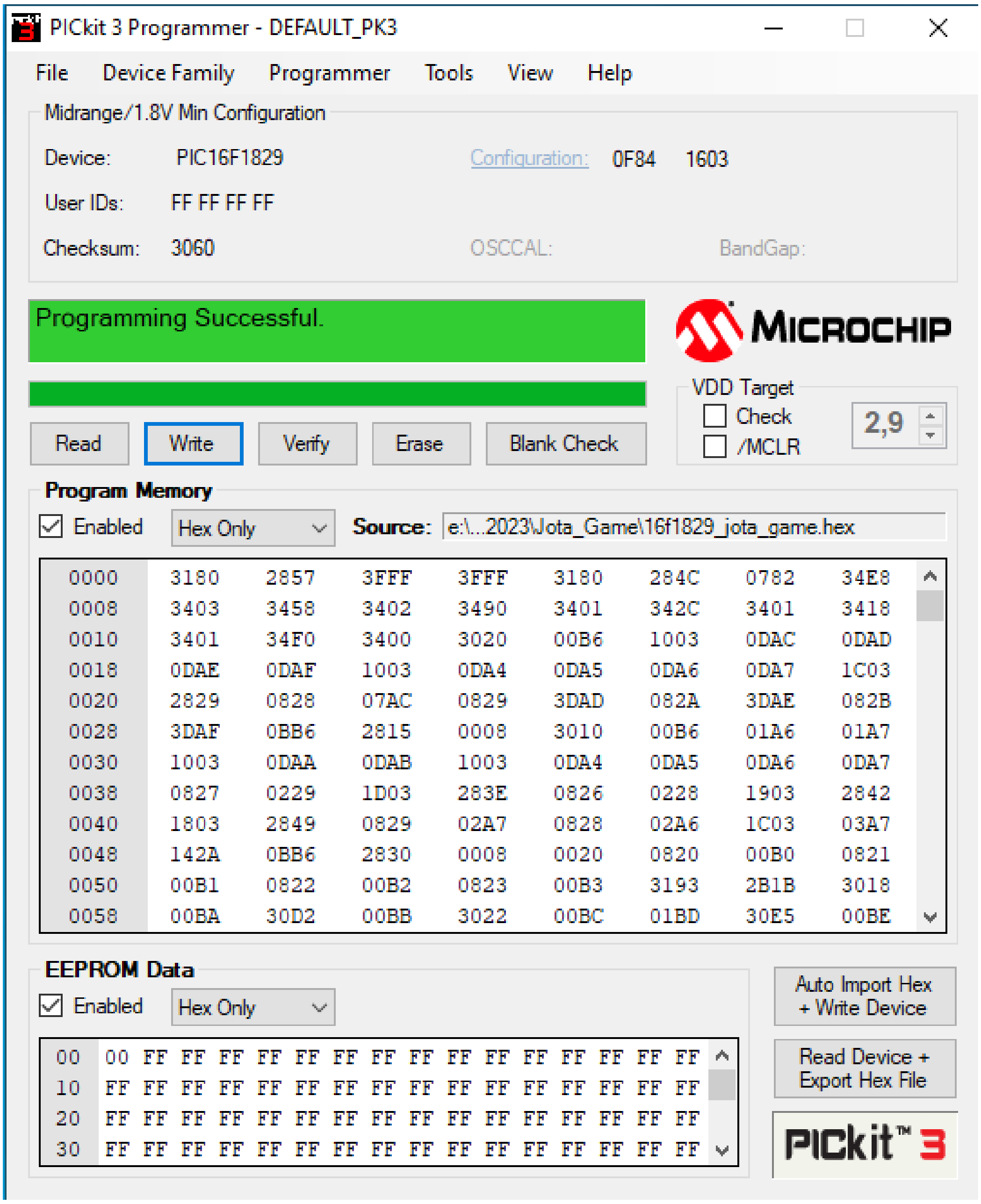
```
00  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

Before loading a new program in the PIC you first need to erase the existing program in the PIC by using the **Erase** button.

Now laod the file: **16f1829_jota_game.hex** via **File → Import Hex file**.

After succesfully loading the hexfile push the Write button. The PIC is now programmed and the programming is verified as to see if programming was succesful.

The result then looks as follows:


PICkit 3 Programmer - DEFAULT_PK3
File
Device Family
Programmer
Tools
View
Help
MICROCHIP
Read
Write
Verify
Erase
Blank Check
Auto Import Hex + Write Device
Read Device + Export Hex File
PICkit™ 3

Midrange/1.8V Min Configuration

Device: PIC16F1829

Configuration: 0F84 1603

User IDs: FF FF FF FF

Checksum: 3060

OSCCAL:

BandGap:

Programming Successful.

VDD Target

□ Check

□ /MCLR

2,9

Program Memory

☑ Enabled

Hex Only

Source: e:\...2023\Jota_Game\16f1829_jota_game.hex

```
0000  3180  2857  3FFF  3FFF  3180  284C  0782  34E8
0008  3403  3458  3402  3490  3401  342C  3401  3418
0010  3401  34F0  3400  3020  00B6  1003  0DAC  0DAD
0018  0DAE  0DAF  1003  0DA4  0DA5  0DA6  0DA7  1C03
0020  2829  0828  07AC  0829  3DAD  082A  3DAE  082B
0028  3DAF  0BB6  2815  0008  3010  00B6  01A6  01A7
0030  1003  0DAA  0DAB  1003  0DA4  0DA5  0DA6  0DA7
0038  0827  0229  1D03  283E  0826  0228  1903  2842
0040  1803  2849  0829  02A7  0828  02A6  1C03  03A7
0048  142A  0BB6  2830  0008  0020  0820  00B0  0821
0050  00B1  0822  00B2  0823  00B3  3193  2B1B  3018
0058  00BA  30D2  00BB  3022  00BC  01BD  30E5  00BE
```

EEPROM Data

☑ Enabled

Hex Only

```
00  00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

If programming fails for some reason, erase the PIC again, load the hexfile again and program the PIC again.

Changing the Jota Game board software

In order to be able to do this you need to download the JAL Libraries and the JAL compiler and install them on your computer.

To get an impression on how this works have a look at the following video: https://www.youtube.com/watch?v=FMLX5y0D5HE

You need the following:

- The Jallib software, consisting of libraries, sample programs and the JAL compiler. You can download this from the JAL Website: http://justanotherlanguage.org/downloads
- If you download the latest release version of Jallib you can choose for a Window Installer program or a zip-file. If you want the latest, but not yet released, version you can download the so-called bee-package. This is always a zip-file
- The JAL source files for the Jota Game board. You can download these from GitHub. You will also find other documentation there like the documentation of the Jota Game board. In the directory 'software' you will find all source files : https://github.com/RobJansen62/Jota_Game/tree/main/software

Changing the source code

When using the Windows installer you can make use of the program JALEdit. This program is only delivered with the Windows installer but not in the zip-file. You can also make use of Visual Studio Code which is available for Windows, Linux and Mac. In the 'doc' folder of the Jallib installation you find the **Jallib_Tutrial_Book**. This document describes which steps you need to follow to use JAL in Visual Studio Code. There is a JAL extension for Visual Studio Code which offers syntax highlight which makes programming a little bit easier.

Of course you can also use a plain text editor like Notepad++.

What can you change?

Because you have the source code you can change anything, for example:

- Your own LED patterns.
- The difficulty of some of the games. You can, for example, change the game 'Simon Says' and limit the number of LEDs that you have to remember to 8. In the JAL source **games_and_actions.jal** you can make this number larger by changing the constant **MAX_SIMON_SAYS** from 8 to 10.
- Change the functionality of the games
- Make your own game, with or without infra-red communcation.

Example of a software modification

The walking menu has 8 options you can select from. When making this software a reaction game was made just like the Jota Game but then without the need to react within a certain given time. Because of adding the game battle ship, this game is not in used. The software, however, still exists. So let's take as example to change which game you can select from the walking menu by replacing the Jota Game by this reaction game so that we can use this game. The game is called **reaction_timer** which can be found in the file **games_and_actions.jal**. The software for handling the actions in the walking menua can be found in the main **16f1829_jota_game.jal**.

In the main program you find the following piece of code:

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

When you replace the code **jota_game()** by the code **reaction_timer()** then the reaction timer game will be selected for menu 5. So what you are doing here is that you are calling another piece of software which then activates another game.

Compiling the source code

After making changes to the source file you need to test it. For that you need to compile all source files as to make a new hexfile. You always need to compile the main program **16f1829_jota_game.jal**. After succesful compilation a new hexfile is created with the name **16f1829_jota_game.hex**.

There are JAL compilers for Windows, Linux and Max. Have a look at the README.txt file in the Jallib 'compiler' directory to see how these compilers are called.

In the **Jallib_Tutorial_Book** it is described how to start the compiler from JALEdit and from Visual Studio Code. If you are using a flat text editor like Notepad++ you start the compiler for Windows as follows (change the path to the directory whery you have placed Jallib and teh source files on your computer):

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

Testing your changes.

After succesful compilation of your changes you can test it. Make sure that the batteries are installed in the Jota Game board.

1. Erase the PIC Microcontroller using the **Erase** button
2. Load the new hexfile in the programmer software
3. Program the PIC Microcontroller using the **Write** button

The program will start and you can test your changes. The programmer can stay connected until you have verified that all changes work as expected. If you are done you can disconnect the programmer.

EEPROM data andthe infra-red address

An address is used for the infra-red communication. The default addres is set to 1 and can be changed by the user, see the manual of the Jota Game board. The address is stored in the EEPROM of the PIC Microcontroller. When (re-) programming the PIC Microcontroller, this EEPROM is erased. After powering up the Jota Game board this address is automatically set to 1 by the Jota Game software.

Broadcast address

The morse game has the option to react to more than 1 address. In normal operation the game only responds to the programmed infra-red address. There is a special broadcast address (address 0) to which all Jota Game board respond for the morse game. This address cannot be set manually but can

In the source code of the file **16f1829_jota_game.jal** you find this piece of 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.
```

By removing the semicolon (;) before the **pragma** statement and by selecting the **EEPROM Data** option in the programmer software adres 0 willl be stored in EEPROM. This is only needed for the sender of the morse codes since the receivers will always respond to their own address and the broadcast address.

By using this broadcast address it is not required that the receivers of the morse game all have the same address so you can use the game with a group.

The broadcast adddress does not work for the battle ship game. This because you always play this game 1 on 1 and not with a group.

Meer information about JAL

More information about JAL can be found on the JAL Website: http://justanotherlanguage.org/

When you have questions on how to use JAL, you can post it on the **Jallist Google Group**. This is an international group so make sure to ask your question in English

There is also a **Jallib Google Group**. This group is meant to report problems with JAL libraries or the JAL compiler. This is also an international group.

Have fun programming the Jota Game board and the PIC Microcontroller with the JAL programming language!