

Maintenance Guide Blitz Armor

Afbeelding met elektronica, overdekt, Elektronisch apparaat, Kantoorapparatuur

Automatisch gegenereerde beschrijving

|  |  |  |
| --- | --- | --- |
| Version | Name | Description |
| 1.0 | Tjebbe van Bergen | First version |
| 1.1 | Frenk van de Nieuwegiessen | Added microcontroller parts |

Content

[Getting started 3](#_Toc163127698)

[Uploading software 3](#_Toc163127699)

[Troubleshooting 3](#_Toc163127700)

[Hardware 3](#_Toc163127701)

[Electric layout 4](#_Toc163127702)

[PCB layout 5](#_Toc163127703)

[Physical connection 5](#_Toc163127704)

[Shield 5](#_Toc163127705)

[Band cable 6](#_Toc163127706)

[Joystick and buttons 6](#_Toc163127707)

[Software 7](#_Toc163127708)

[Changing ROM 7](#_Toc163127709)

# Getting started

To play the game you need to do a few things:

1. Plug the VGA cable into the FPGA.
2. Plug the other side of the VGA cable into the monitor or TV.
3. Plug the USB cable into the FPGA.
4. Plug the USB cable into the microcontroller.

If it’s the first time using the hardware go to [uploading software](#_Uploading_software). For instructions on flashing the hardware.

# Uploading software

To upload software to the microcontroller and FPGA you need two programs.

Download [Vivado](https://www.xilinx.com/support/download.html) version 2023.2 default compiler.

If the user wants to edit the code and upload it to the microcontroller:

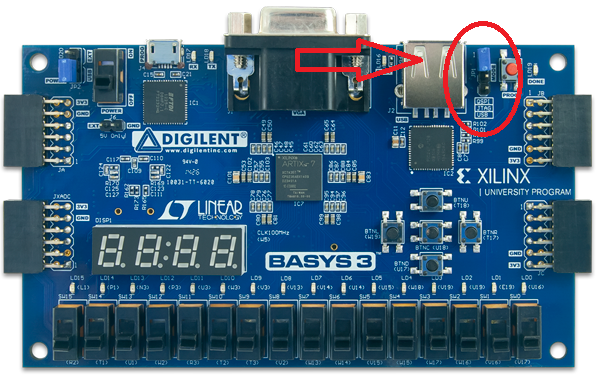
Download [cubeIDE](https://www.st.com/en/development-tools/stm32cubeide.html) version 1.15.0 default compiler.

If the user only wants to upload the game code to the microcontroller:

Download [STM32CubeProgrammer](https://www.st.com/en/development-tools/stm32cubeprog.html?dl=redirect) version 2.16.0.

To upload the software to the FPGA you need this bit file. And flow the instructions of [Programming the Basys3 using Quad SPI](https://digilent.com/reference/learn/programmable-logic/tutorials/basys-3-programming-guide/start)

When the basys3 board is powered but does not show anything you can use the usb stick in the casing of the game. On this stick there is the bitfile of the VHDL project. To use this stick, you need to change the jumper on the top right of the basys3 board from JTAG to USB. After changing the jumper, you need to power the basys3 board and insert the usb stick into the usb port. When done correctly, this VHDL code will work.



To upload the software to the microcontroller you need to download it from [GitHub](https://github.com/Joluzut/Retro_game). Download the .bin or .elf file if you only want to upload the software. Download the folder game\_files containing the .h and .c files and the .ioc file if you also want to edit the software.

To import the project into STM32CubeIDE go to file -> new -> create a new stm32 project from an existing file (.ioc). Select the .ioc file location and add the .h and .c manually to the project.

Use [this guide](https://youtu.be/oJ0D5Xye6Hg?si=lJ0Z60KXdUG5yWnB&t=344) to help you flash the STM 32 with STM32CubeIDE.

# Troubleshooting

Powering up the hardware is best done by first powering the FPGA and after receiving VGA signal on the monitor, power up the STM32

Reconnect microcontroller if game is hanging or not loading.

Look at the [connectors](#_Physical_connection) and fix loose connections.

[Flash](#_Uploading_software) STM and FPGA with the newest version of the code.

# Hardware

Necessary hardware for this project:

* Basys 3 FPGA.
* STM 32 nucleo f030r8.
* 2 push buttons.
* 1 joystick with four built-in buttons.
* 8 cables with connector for joystick. Preferable 2 color’s. 0,75mm^2 multicore
* 4 cables for the push buttons. Preferable 2 color’s. 0,50mm^2
* 1 band cable 12 lanes.

## Electric layout

Afbeelding met tekst, Rechthoek, diagram, schermopname

Automatisch gegenereerde beschrijving

## PCB layout

Afbeelding met tekst, schermopname, diagram, stroomkring

Automatisch gegenereerde beschrijving

## Physical connection

### Shield

To connect the Arduino shield to the STM you just need to align with the pins right. The PINs of the Arduino shield are mostly the same as the STM. So check if the PINs are correct and then the shield should fit.

### Band cable

Plug the band cable on the 12 male pines on the STM shield with the red side of the cable to the USB connector.Afbeelding met elektronica, Elektrische bedrading, Elektronische engineering, kabel

Automatisch gegenereerde beschrijving

Plug the other side of the band cable in the FPGA with the orientation and port next to the USB A port of the picture. So the red line on the 3.3V lines of the FPGA. Afbeelding met elektronica, Elektrische bedrading, Elektronische engineering, kabel

Automatisch gegenereerde beschrijving

### Joystick and buttons

Connect the side with the connector for the joystick on the joystick like in the picture. Afbeelding met overdekt, rood

Automatisch gegenereerde beschrijving

It is preferred to use 2 colours for the ground and connection cable.

Put the other side of the cable in the STM shield. Look at the layout [electrical](#_Electric_layout) and [PCB](#_PCB_layout) layout for the right connection.

Afbeelding met stroomkring, Elektronische engineering, Stroomkringonderdeel, Elektronisch onderdeel

Automatisch gegenereerde beschrijving

### Yellow Buttons

connect the cables of the yellow buttons in the blue connector. Look at [PCB](#_PCB_layout) and [electrical](#_Electric_layout) layout for the correct connection

If you want to replace the cable of the yellow buttons look closely at the black piece, because one side is + and the other side is -. Afbeelding met kabel, Elektrische bedrading, geel, persoon

Automatisch gegenereerde beschrijving

# Software

### C and H files overview

bullet.h/c -- physics and hit detection/handling of the bullet

button.h/c -- input handler

global.h/c -- global defines and other variables

grondMap.h/c -- map generation and visibility

highscore.h/c -- highscore calculation and data

letterBox.h/c -- onscreen text write

main.h/c -- main state machine

mainmenu.h/c -- main menu data

player.h/c -- player movement and collisions handler

spiData.h/c -- spi write command

stm32f0xx\_hal\_conf.h/c -- auto generated by stm32cube IDE

stm32f0xx\_it.h/c -- auto generated by stm32cube IDE

## Changing ROM

The only ROM file is for the letters on the screen.

Use this guide to change it:

Go to sources

Go to the ROM IP in these instance it is called Alphabet\_Rom\_instAfbeelding met tekst, schermopname, software, Webpagina

Automatisch gegenereerde beschrijving

Double click on the file.

A window opens up, got to other options

Afbeelding met tekst, schermopname, software, Webpagina

Automatisch gegenereerde beschrijving

And now you can change the COE file. By giving the path to the desired COE file. The COE file used can be found on the [GitHub](https://github.com/Joluzut/Retro_game/tree/feature/knop/COE%20File).

### Changing text

Open global.c and edit any of the char variables.

letterBox.c contains also char variables but these get edited by the code and are not fully editable.

### Changing sound

How to change sound:

Open vivado.

Change any of the tones

### Changing tank

We would not recommend changing the physical aspect of the tank because the tank is just an array of different colours.

But you can change the colour of the tank by: