

GottFA80

Gottlieb System80 MPU based on FPGA

Hardware v. 4.00

Software Version 0.99

user manual

ralf@lisy.dev

v4.00_01 25.07.2021

Table of contents

| | |
|--|----|
| Important remark..... | 3 |
| 1. Introduction..... | 3 |
| 2. Quickstart | 4 |
| 3. Put the GottFA80 Image to the SD card | 5 |
| 3.1. Win32DiskImager | 5 |
| 4. Installation..... | 6 |
| 5. Dip Switch Settings | 6 |
| 5.1. S1..... | 6 |
| 5.1.1. Dips 1..7 : game select..... | 6 |
| 5.1.2. Dip 8 : 80B game..... | 6 |
| 5.2. S2 | 6 |
| 5.2.1. S2-Dip1 -> Freeplay | 6 |
| 5.2.2. S2-Dip2 -> init nvram..... | 6 |
| 5.2.3. S2-Dip3 -> slam contact open..... | 6 |
| 5.2.4. S2-Dip4 -> slam contact closed..... | 6 |
| 6. boot sequence..... | 7 |
| 6.1. phase 1: init | 7 |
| 6.2. phase 2: SD card read..... | 7 |
| 6.3. phase 3: program execution..... | 7 |
| 7. programming the FPGA | 8 |
| 7.1. programmer software | 8 |
| 7.2. install the driver for your USB Blaster | 8 |
| 7.3. program your FPGA | 9 |
| 8. structure of SD card..... | 12 |
| 8.1. rom file structure for GottFA80..... | 12 |

Important remark

By using GottFA80 it is possible to damage your pinball machine. As this is a private project with NO commercial interest the author accepts no liability for any damage that may arise by using GottFA80!

1. Introduction

- GottFA80 is a 100% hobby project. This makes the solution cheap, depending on where you buy your components it is possible to create your Gottlieb replacement MPU for less than 50€.

What do you need?

- Basic soldering skills
- Possibility to read/write micro SD cards
- A PC with an USB port in order to be able to program the FPGA
- Gottlieb rom images (not included due to Copyright limitation)

2. Quickstart

1. Download latest versions of the SD card Image and the FPGA program from lisy.dev
2. Write the image to a SD card
3. Add Gottlieb roms
4. Program the FPGA
5. Configure switch 'game select' according to your pinball (Appendix A)
6. Replace your original System80 MPU with GottFA80
7. Switch the Game ON
8. Enjoy

3. Put the GottFA80 Image to the SD card

On my website you will find the latest version of GottFA80 as a (zipped) image file. After unpacking the image can be put on a SD card. I do recommend using Win32DiskImager for doing that.

This article uses content from the eLinux wiki page [RPI Easy SD Card Setup](#), which is shared under the [Creative Commons Attribution-ShareAlike 3.0 Unported license](#)

3.1. Win32DiskImager

- Insert the SD card into your SD card reader. You can use the SD card slot if you have one, or an SD adapter in a USB port. Note the drive letter assigned to the SD card. You can see the drive letter in the left hand column of Windows Explorer, for example **E:**
- Download the Win32DiskImager utility from the [Sourceforge Project page](#) as an installer file, and run it to install the software.
- Run the Win32DiskImager utility from your desktop or menu.
- Select the LISY image file you extracted earlier.
- In the device box, select the drive letter of the SD card. Be careful to select the correct drive: if you choose the wrong drive you could destroy the data on your computer's hard disk! If you are using an SD card slot in your computer, and can't see the drive in the Win32DiskImager window, try using an external SD adapter.
- Click 'Write' and wait for the write to complete.
- Exit the imager and eject the SD card.

4. Installation

GottFA80 boards have the same connectors and same mounting holes as the original Gottlieb System80 MPUs, so replacing of the board can be done in seconds.

5. Dip Switch Settings

5.1. S1

5.1.1. Dips 1..7 : game select

Here you can select what game GottFA80 should run. This depends on the roms placed on the SD card. See later chapter for an explanation of the structure of the SD card content.

5.1.2. Dip 8 : 80B game

If you want to emulate a 80B game put Dip8 to on

5.2. S2

5.2.1. S2-Dip1 -> Freeplay

With set to ,ON' GottFa will enable 'Freeplay'. By 'press and hold' the credit button for more than two seconds a coin insert for the left coin chute is simulated.

5.2.2. S2-Dip2 -> init nvram

With set to ,ON' GottFa will initialize the nvram ram during boot for the selected game to zero. This is useful if you want to reset ALL ram content. (recommended for the first start of the game)

5.2.3. S2-Dip3 -> slam contact open

With set to ,ON' the slam will be simulated as 'open' (needed for late 80B games)

5.2.4. S2-Dip4 -> slam contact closed

Most System80 games need the Slam contact to be closed and will not boot in case the slam contact is open. With set to ,ON' the slam will be simulated as 'closed' independent of the real status.

6. boot sequence

6.1. phase 1: init

Immediately after switching on the pinball with GottFA80 inserted you will see the following output on the display of your pinball machine

Player 1: version of the FPGA program running

Player 2: value of selected game on S1

Player 3: lisy.dev unique identifier for FPGA based MPUs

Note: with the current version there is no GottFA boot message for 80B games.

6.2. phase 2: SD card read

GottFA80 tries to read the SD card content, if this fails the red LED 'SD card error' will go ON.

6.3. phase 3: program execution

The code indicated by the Dip switch 'game select' is read from the SD card and executed. If the code runs (regular interrupts are seen) the second LED on the Cyclone IV card (on the backside) will go ON.

7. programming the FPGA

To program the FPGA you need the Quartus Programmer.

7.1. programmer software

It can be downloaded from the Intel Website for free. You just need to create a user account.

<https://fpgasoftware.intel.com/13.0sp1/?edition=web>

Go to Additional Software and Download **Quartus II Programmer and SignalTap II**

Note: Use version 13.0sp1! **Do not use newer versions of the programmer.** The used FPGA for BallyFA is quite old (and therefore cheap) but is not supported by latest versions of the programmer.

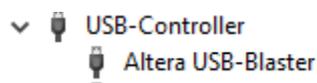
7.2. install the driver for your USB Blaster

When connecting your USB Blaster the first time it will not be recognized correctly by Windows.



You also need to install the driver for your USB Blaster. The driver comes together with the installation of the programmer and is located in the 'driver' subdirectory.

- Right click on the entry in the device manager and choose 'update driver' (Treiber aktualisieren)
- Choose 'search for driver on this PC' (auf dem Computer nach Treibersoftware suchen)
- For a default installation select 'C:\altera\13.0sp1\qprogrammer\drivers'
- Confirm installation
- Now the Altera USB Blaster should be visible und 'USB-Controller'

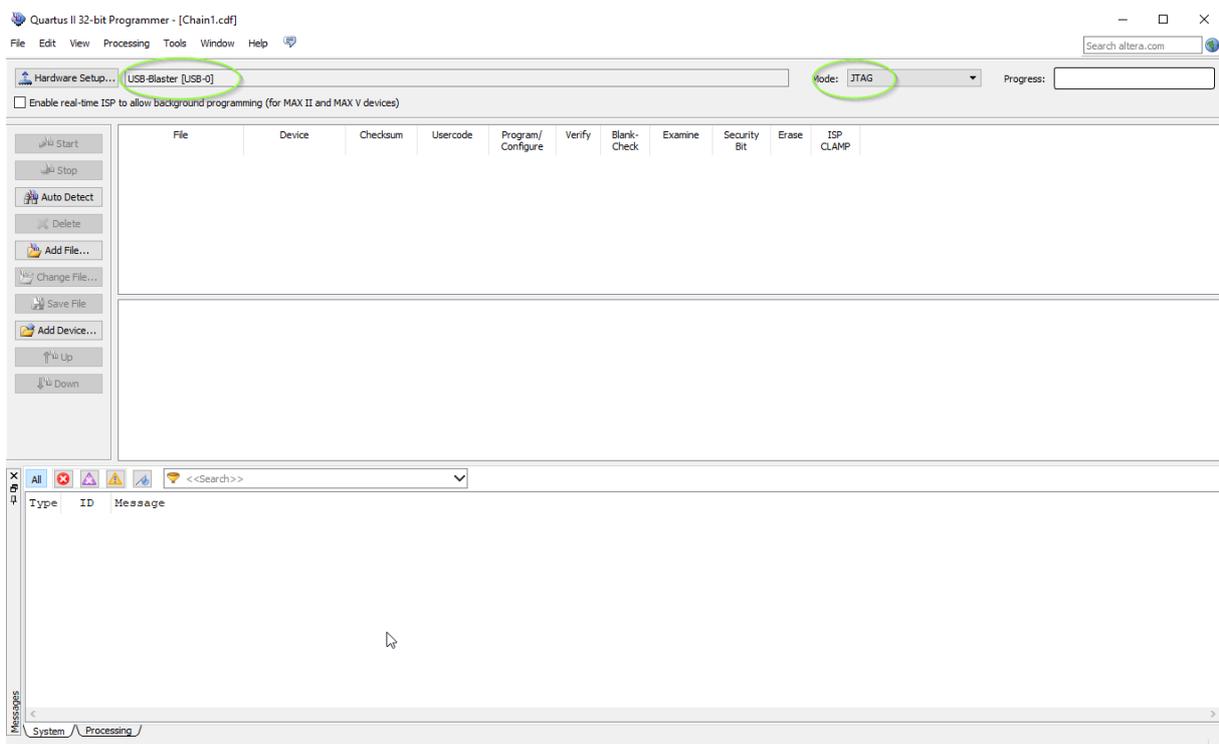


7.3. program your FPGA

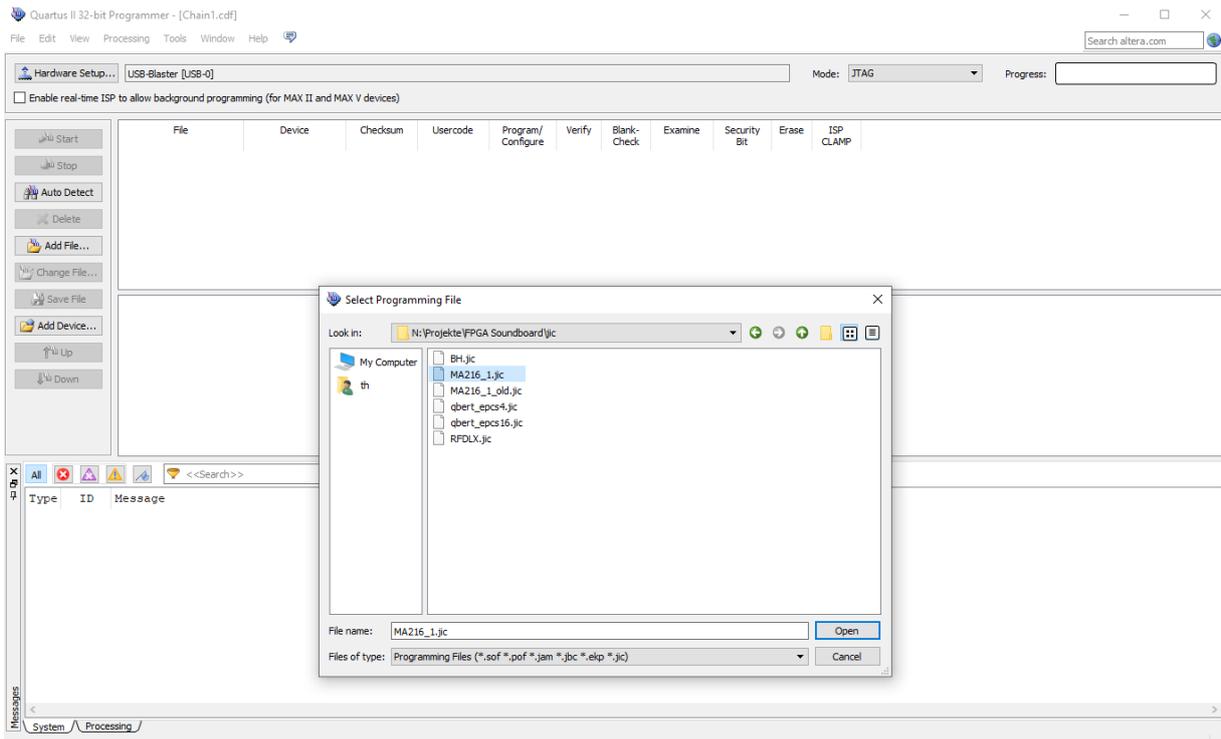
- 1) Connect the USB Blaster to the PC



- 2) Start the programmer, make sure in the Hardware setup ,USB-Blaster' is visible and the Mode is set to ,JTAG'

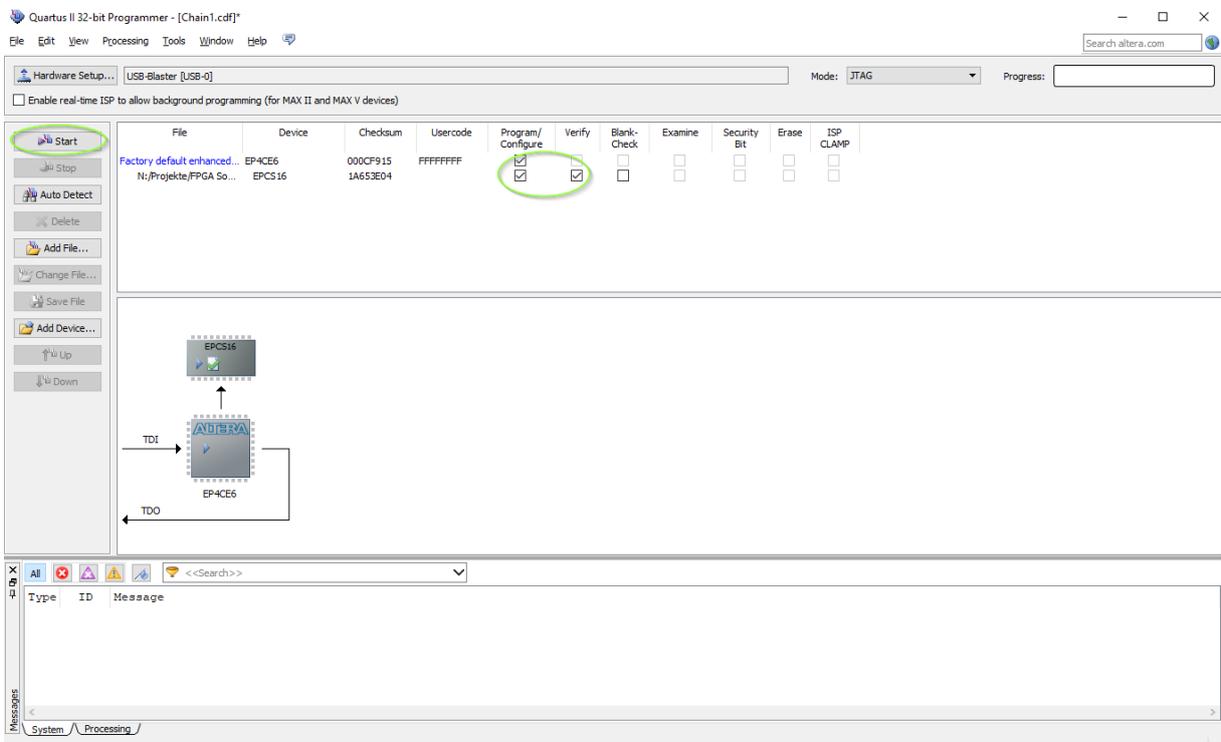


- 3) Select ,Add File' and choose the right ,jic' file for your game



4) Power the FPGA via USB and connect the USB Blaster to the ,programming' connector on the Gosof PCB

5) Select ,Program/Configure' and ,Verify'



6) Push start, wait for progress 100%

The screenshot displays the Quartus II 32-bit Programmer software interface. The main window shows the hardware setup for a USB-blaster (USB-0) connected to an EP4CE6 device. The progress bar at the top right indicates that the operation is 100% successful. Below the hardware setup, a table lists the device details and the status of various operations. The bottom panel shows a log of messages detailing the programming process, including device configuration, erasing, and programming steps.

| File | Device | Checksum | Usercode | Program/Configure | Verify | Blank-Check | Examine | Security Bit | Erase | ISP CLAMP |
|-----------------------------|--------|----------|----------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Factory default enhanced... | EP4CE6 | 000CF915 | FFFFFFFF | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| N:\Projekte\FPGA So... | EP4CE6 | 1A653E04 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

```
Messages
Type ID Message
209060 Started Programmer operation at Thu Feb 11 16:26:20 2021
209016 Configuring device index 1
209017 Device 1 contains JTAG ID code 0x020F10DD
209007 Configuration succeeded -- 1 device(s) configured
209018 Device 1 silicon ID is 0x14
209044 Erasing ASP configuration device(s)
209023 Programming device(s)
209021 Performing CRC verification on device(s)
209011 Successfully performed operation(s)
209061 Ended Programmer operation at Thu Feb 11 16:26:40 2021
```

That's it!

8. structure of SD card

Due to limitations of the SD card read routine in the FPGA (it does read fix sector numbers instead of looking for filenames) it is necessary to use my SD-card image (128 Mbyte). You can write the image to a SD-card of your choice.

8.1. rom file structure for GottFA80

Each rom file has a size of 16Kbyte and must have the 'game rom' within the first 8 Kbyte and the 'System rom' in the second 8 Kbyte. All Gottlieb Games have a 8Kbyte system rom while the size of the game roms depends on the game. Early Gottlieb games (Panthera, Spiderman, Circus, Counterforce and Star Race) will come with a game rom of only 1Kbyte. Games as James Bond and later will come with a 2Kbyte game rom. Late System80B (Excalibur, Bad Girls, Big House, Hot Shots Bone Buster and Night Moves) come with a 4Kbyte game rom.

For each game GottFA read a 16Kbyte block from the SD card, so you need to fill the gaps depending on the game you are using.

Example for Black Hole:

Gottlieb Black Hole has a 2Kbyte game.rom (668-4.cpu) und two systems roms 4Kbyte each (U2_80.bin & U3_80.bin). To create a rom file for GottFa (GottFA80_BH.img) you can use the following command on a windows system:

Copy /b 668-4.cpu + 668-4.cpu + 668-4.cpu + 668-4.cpu + U2_80.bin + U3_80.bin GottFA80_BH.img

The multiple copies of 668-4.cpu are just to fill the 6Kbyte gap between the game rom and the system rom.

Example for Raven.

Gottlieb Raven (System80B) has a 2Kbyte game.rom (prom2.cpu) and a 8Kbyte system.rom (prom1.cpu). To create a rom file for GottFa (GottFA80_Raven.img) you can use the following command on a windows system:

Copy /b prom2.cpu + prom2.cpu + prom2.cpu + prom2.cpu + prom1.cpu GottFA80_Raven.img

Example for Bad Girls.

Gottlieb Raven (System80B) has a 4Kbyte game.rom (prom2.cpu) and a 8Kbyte system.rom (prom1.cpu). To create a rom file for GottFa (GottFA80_BadGirls.img) you can use the following command on a windows system:

Copy /b prom2.cpu + prom2.cpu + prom1.cpu GottFA80_BadGirls.img

Example for Rock (also valid for Bounty Hunter, Chicago Cubs Triple Play, Rock Encore)

Gottlieb Raven (System80B) has a 8Kbyte system.rom (prom1.cpu) and **no separate game.rom**. To create a rom file for GottFa (GottFA80_Rock.img) you can use the following command on a windows system:

Copy /b prom1.cpu + prom1.cpu GottFA80_Rock.img