User Documentation

MAXwel

MAX ii Workbench for Education and Learning

Michael Hohenstein

August 19, 2024

Contents

1 Hardware

1.1 Pinmap

To access the onboard LEDs, buttons and switches, knowing the connected GPIO pin of the FPGA is necessary. An overview about the accessories placed on the board is depicted in Fig. 1.1. This picture also numbered all of them for later identification where the corresponding pinmap is given in the subsequent sections.

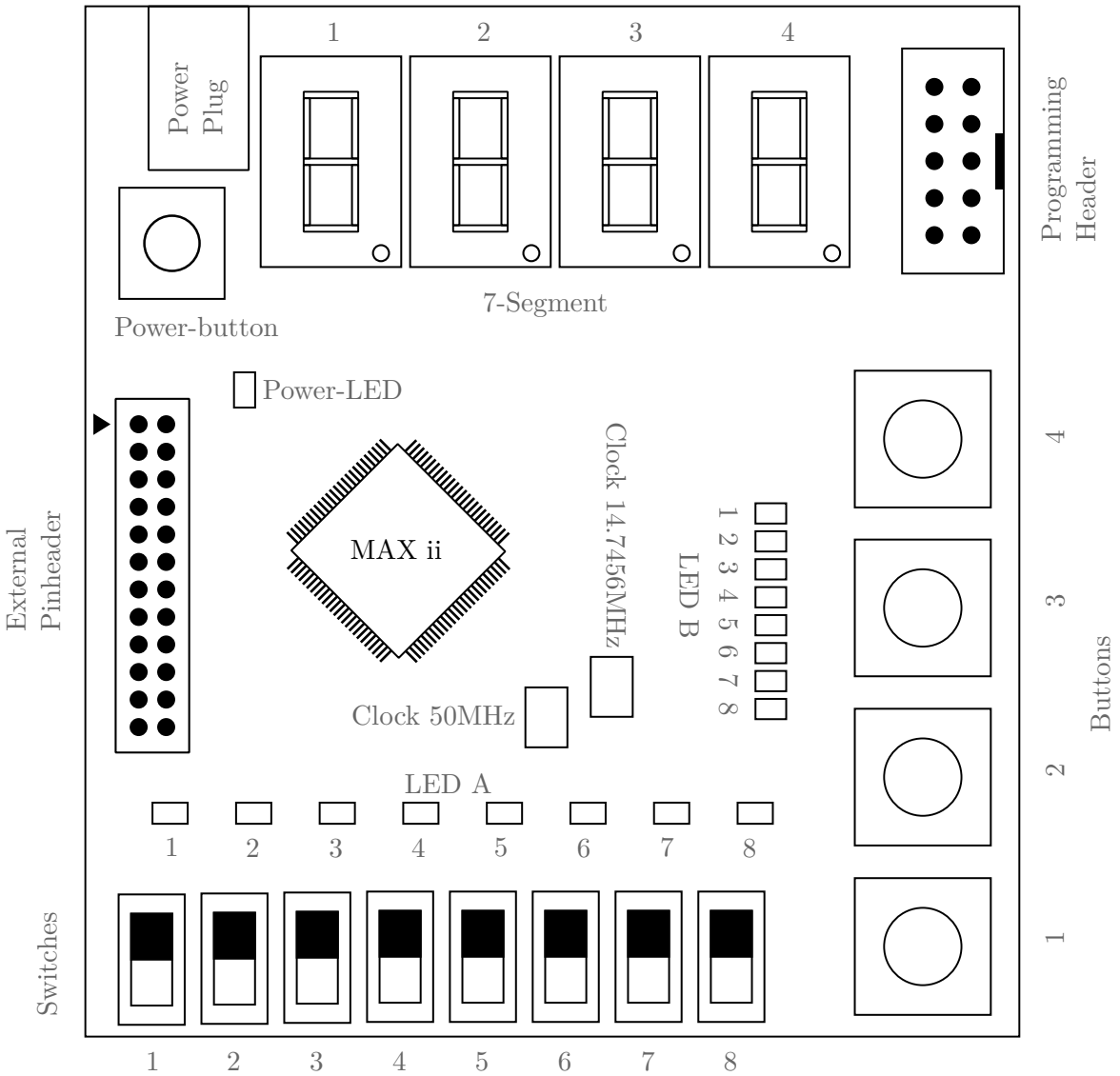

Figure 1.1: Floorplan of the MAXwel with labeled peripheral devices.

1.1.1 Power

The power supply is connected to the board via a barrel jack. A power button is placed next to the jack to turn the board on and off. To indicate the power state, a power LED is placed on the board that always lights when the board is powered.

1.1.2 Clocks

The board has two clock sources, a 50 MHz oscillator and a 14.7456 MHz oscillator. Both are connected to special GPIO pins that are dedicated for clock signals. These pins provide a low skew and jitter free clock signal and provide access to the internal global clock network of the FPGA.

Clock	FPGA Pin
50 MHz	PIN_12
14.7456 MHz	PIN_14

1.1.3 LEDs

The board has 16 LEDs, divided into two rows (namely A and B) of 8 LEDs each, that can be controlled by the FPGA. To turn an LED on the corresponding pin has to be set to high.

LED number	LED A FPGA Pin	LED B FPGA Pin
1	PIN_86	PIN_18
2	PIN_88	PIN_17
3	PIN_90	PIN_16
4	PIN_92	PIN_15
5	PIN_96	PIN_8
6	PIN_98	PIN_7
7	PIN_100	PIN_6
8	PIN_2	PIN_5

1.1.4 Buttons and Switches

The board is equipped with 4 buttons and 8 switches. The buttons are debounced with a 8 kHz low pass filter and should drive the input pin to high when pressed.

Button number	FPGA Pin
1	PIN_3
2	PIN_4
3	PIN_19
4	PIN_20

The switches are also debounced with a 8 kHz low pass filter and should drive the input pin to high when the switch is in the upper position.

Switch number	FPGA Pin
1	PIN_85
2	PIN_87
3	PIN_89
4	PIN_91
5	PIN_95
6	PIN_97
7	PIN_99
8	PIN_1

1.1.5 Seven Segment Display

The MAXwel is equipped with a four digit seven segment display. The seven segment displays have a common anode: to turn on a segment the corresponding pin has to be set to high. The labels for the segments are depicted in Fig. 1.2.

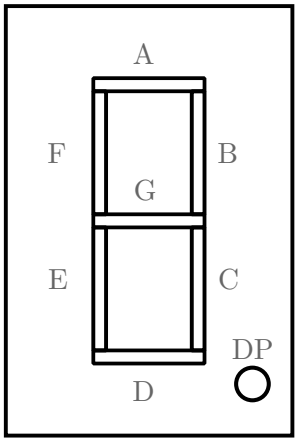

Figure 1.2: Seven segment display with labeled segments.

Segment Name	Segment 1	Segment 2	Segment 3	Segment 4
A	PIN_56	PIN_48	PIN_38	PIN_28
B	PIN_54	PIN_44	PIN_36	PIN_26
C	PIN_55	PIN_47	PIN_37	PIN_27
D	PIN_58	PIN_50	PIN_40	PIN_30
E	PIN_62	PIN_52	PIN_42	PIN_34
F	PIN_57	PIN_49	PIN_39	PIN_29
G	PIN_61	PIN_51	PIN_41	PIN_33
DP	PIN_53	PIN_43	PIN_35	PIN_21

1.1.6 External pin header

The orientation of the external pin header is determined by the reference triangle. All Pins on the pin header are connected directly to the GPIO pins of the FPGA so take care when connecting external devices to the board. Do not exceed the voltage range of 0 V to 3.3 V and do not draw too much current from the pins. The pin header is depicted in Fig. 1.3.

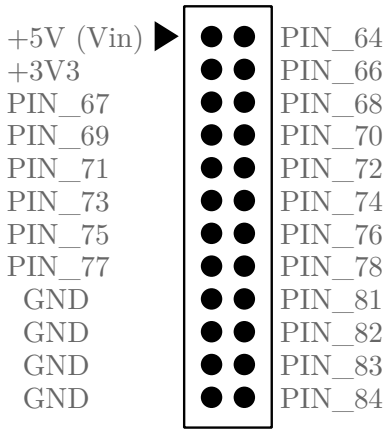

Figure 1.3: External pin header with labeled pins.

1.1.7 Programming header

The programming header is used to program the FPGA with a JTAG programmer. The pinout of the programming header is depicted in Fig. 1.4. The *TCK* pin is connected to a 10 kΩ pull down resistor and the *TDI*, *TMS* and *TDO* pins are connected to a 10 kΩ pull up resistor.

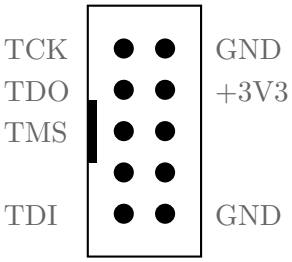

Figure 1.4: Programming header with labeled pins.

1.2 Development

The hardware development was done using the KiCAD EDA software. There, the Circuit is first drawn as an abstract schematic diagram. Afterwards each component in the schematics needs a corresponding footprints assigned for the circuit board. These footprints are afterwards places on the circuit board editor and traces are routed between them, guided by the schematic diagram.

For this project some custom schematic symbols and footprints were used which are locates in the project directory. The fabrication process was finalised using the Fabrication Toolkit Plugin. Furthermore a Design Rule file from the circuit board manufacturer was placed in the project directory to make sure, the clearance between traces, vias an pads on the circuit board match the fabrication requirements.

The main directory of this project contains the shell-script *git_filters.sh* to install some git filters for this repository to provide better git integration for the KiCAD files.

1.3 Components

This is a list of all the components on the circuit boards, their datasheets and a reference for part ordering.

table 1.1 shows the components that were ordered with the circuit board which can be done by selecting the "PCB assembly" options when ordering the circuit board. For this project the PCB assembly was done by JLCPCB and all components were ordered from the same page but they are being supplied by LCSC. After uploading the Gerber files[1], the BOM[2] and the CPL[3] files, the components can be selected from a table and added to or removed form the order. Components that were not identified directly can be selected form a list at this stage in the ordering process. All SMD components except the LEDs were ordered with the circuit board because they are too small for hand soldering (and it's not worth the time). The LEDs were ordered separately because they were not available at the time of ordering. Each THT component was removed from the order and soldered afterwards to keep the cost of the circuit board low.

Component	Datasheet / Information	LCSC Component Number
MAX ii EPM240T100C5N	MAX ii Device Handbook, EPM240 Device page	C10041
AMS1117-3.3	AMS1117 Datasheet	C6186
50 MHz MHz Crystal Oszillator	Datasheet	C387308
14.7456 MHz MHz Crystal Oszillator	Datasheet	C7431374
Capacitors and Resistors		The capacitors and resistors are standard components and are automatically added to the catalog without any component search involved.

Table 1.1: All components ordered with the circuit board

All the other components were bought from another supplier and are listed in table 1.2. Some of these components were already available and were therefore not ordered. The source for them is still listed in the table.

[1]The Gerber files contain the manufacturing floorplan for the circuit board
[2]BOM: Bill of Material, a list of all components on the circuit board
[3]CPL: Centroid Pick List, a list of all components and their positions on the circuit board

Component	Datasheet / Information	Component source
100 μF Capacitors	5 mm Diameter, 11 mm height, 2 mm pitch	RS-Component Number: 228-6650
LEDs Red	SMD 0805	RS-Component Number: 692-0941
7-Segment Displays		RS-Component Number: 877-1564
Power Plug		RS-Component Number: 259-6422
5 V Power Supply		RS-Component Number: 121-7115
Programming Header	2.54 mm JTAG 10-pin	Private Inventory, Source
Bush Buttons	12 mm $\times$ 12 mm $\times$ 5 mm	Private Inventory, Source
Sliding Switch	SS12D10	Private Inventory, Source
Programming Adapter	Altera USB Blaster	Private Inventory, Source
Power LED	SMD 0805	Private Inventory
External Pinheader	Dupont 2.54 mm	Private Inventory
JTAG Programmers	Altera USB Blaster	Private Inventory, Source

Table 1.2: All the other components soldered afterwards

2 Software

2.1 Installing the Toolchain

In order to use the FPGA Chip, the Quartus Prime Lite Design-Software[1] needs to be installed. For users of an Arch based Linux Distribution there is already a package for this available in the AUR.

After downloading and starting the installation some additional parameters need to be specified. For Linux, [/opt] is the preferred installation directory. Furthermore it is important that the Device files for the *MAX ii* Chips is selected for installation. The rest of the installation process should be straight-forward.

2.1.1 Linux udev device rules

The JTAG Programming adapter, formally known as „USB Blaster“, is regularly only available with read and write permissions for the root user. To make the device available for a user without higher privileges the file **/etc/udev/rules.d/51-usbblaster.rules** needs to created with the following content:

```
1 SUBSYSTEM=="usb", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6001", MODE="0666"
2 SUBSYSTEM=="usb", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6002", MODE="0666"
3 SUBSYSTEM=="usb", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6003", MODE="0666"
4 SUBSYSTEM=="usb", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6010", MODE="0666"
5 SUBSYSTEM=="usb", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6810", MODE="0666"
```

After rebooting the operating system the changes are available for the udev device daemon.

2.2 Software Usage

This section explains the usage of the Quartus Software as well as the usage of the two provided template projects.

2.2.1 Quartus How-To

The Quartus splash screen offers two frequently needed options: *New Project Wizard* and *Open Project*. For the template Projects the *Open Project* option opens a file manager that is used to open the project file. The Project file is located in the directory of the project and has the file extension *.qpf*.

Project Navigator

The left side of the Quartus window shows the *Project Navigator* which is used to navigate through the project. In its default state it shows a hierarchy of the project files but it can be switched to a file view as shown in Fig. 2.1.

The files can be edited by double-clicking on them in the project navigator.

[1] Not to mix up with Quartus Prime Pro!

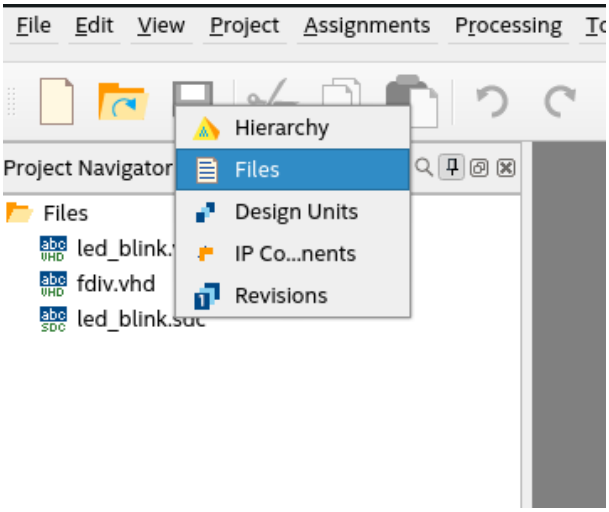

Figure 2.1: The Quartus project navigator allows to navigate through the project files.

Synthesis

To synthesize a design the Quartus Software has two methods to start the synthesis process. The first method is to click on the *Start Compilation* button in the toolbar. The second method is to double click on the *Compile Design* entry in the *Tasks* window. The *Tasks* window is located in the bottom left corner of the Quartus window. Both options are highlighted in Fig. 2.2. The synthesis process can take a few minutes depending on the complexity of the design. Any errors that occur during the synthesis process are shown in the *Messages* window which is located in the bottom of the Quartus window.

Programming the FPGA

The synthesized design can be uploaded using the *Programmer* tool. The programmer tool can be opened by clicking on the *Programmer* button in the toolbar or double clicking the *Program Device* entry in the *Tasks* window. Both options are highlighted in Fig. 2.3.

The dialog window of the programmer tool is shown in Fig. 2.4. There, the *Hardware Setup* can be configured to select a programmer device. Afterwards, the checkbox of the "CFM" file (the synthesis result) has to be checked for the option *Program/Configure*. By clicking on the *Start* button the design is uploaded to the FPGA. The important buttons are highlighted in Fig. 2.4.

2.2.2 Template Projects

The template projects have been created using the *New Project Wizard*. During this setup, the target device needs to be specified. For the MAXwel board it is the *EPM250T100C5* Chip. Afterwards an empty project should be available. The Input / Output pins of the FPGA can be configured by selecting the *Pin Planner* in the *Assignments* menu. This allows to assign the pins to the desired signals, which is already done in the template projects. It is important that the pins assigned to the Switches and Buttons have the value in the *I/O Standard* column set to *3.3V Schmitt Trigger Input* whereas the other pins can stay at the default value *3.3V LVCMOS*. The informations specified in the pin planner are also located in the *.qsf* file of the project.

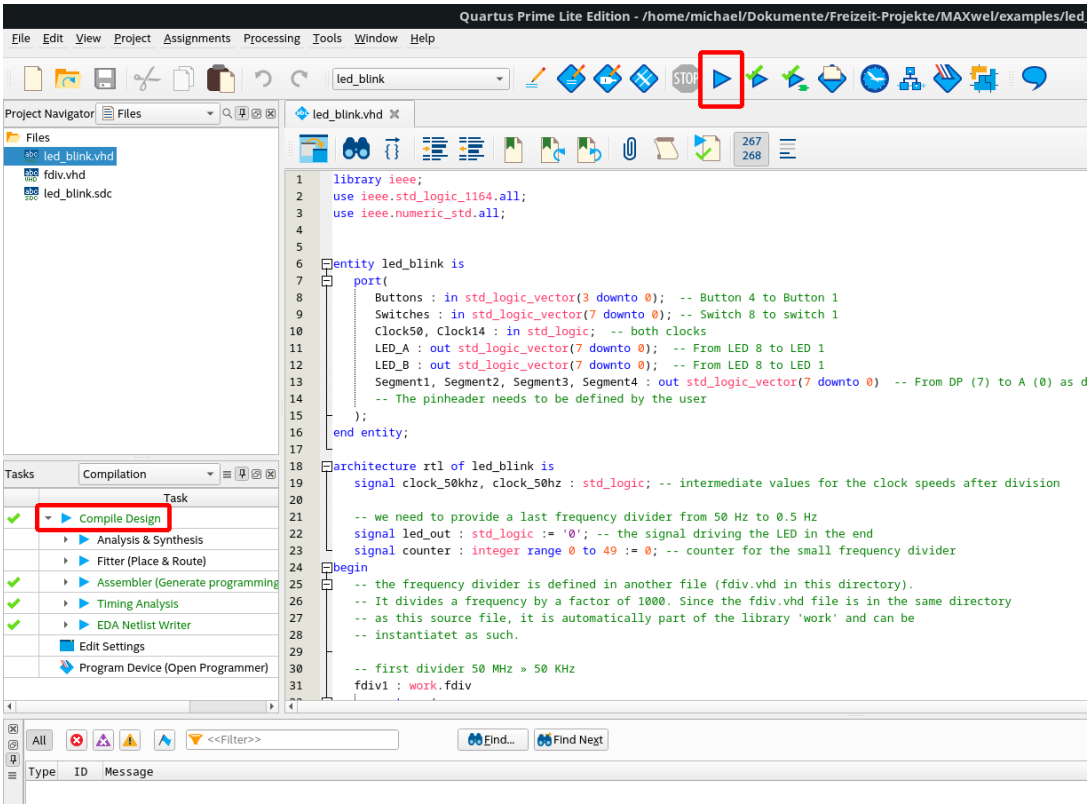

Figure 2.2: There are two ways to start a synthesis

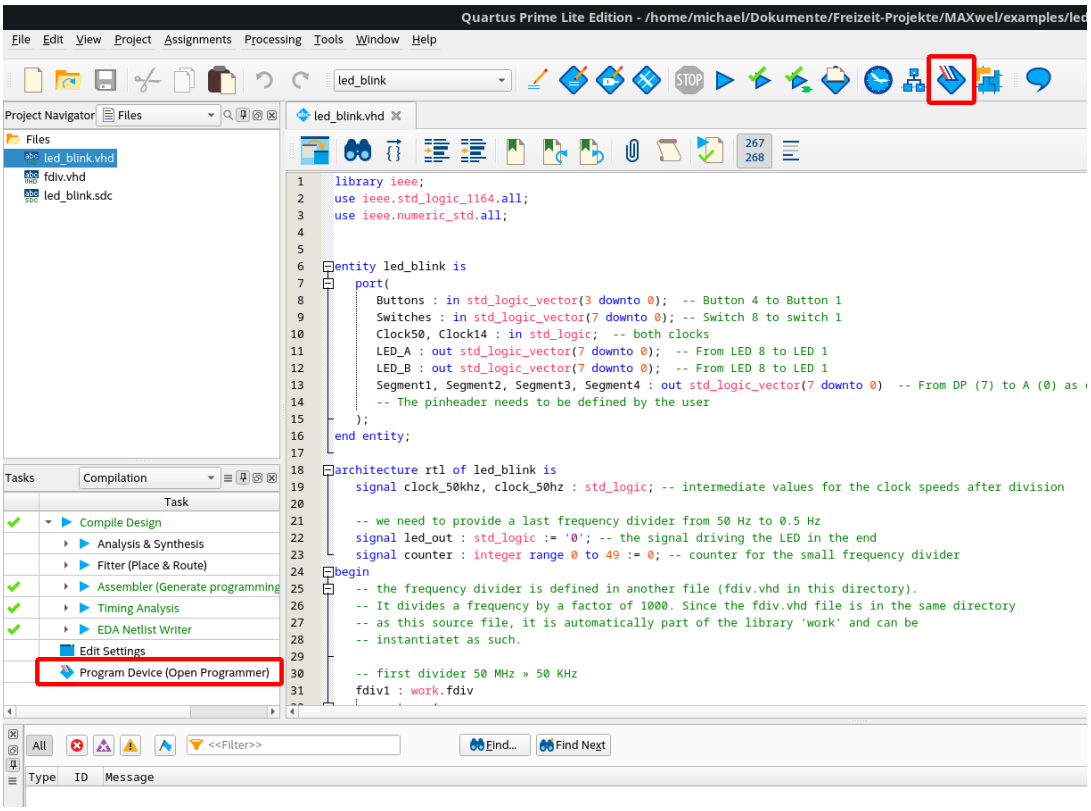

Figure 2.3: Possibilities to open the programmer tool.

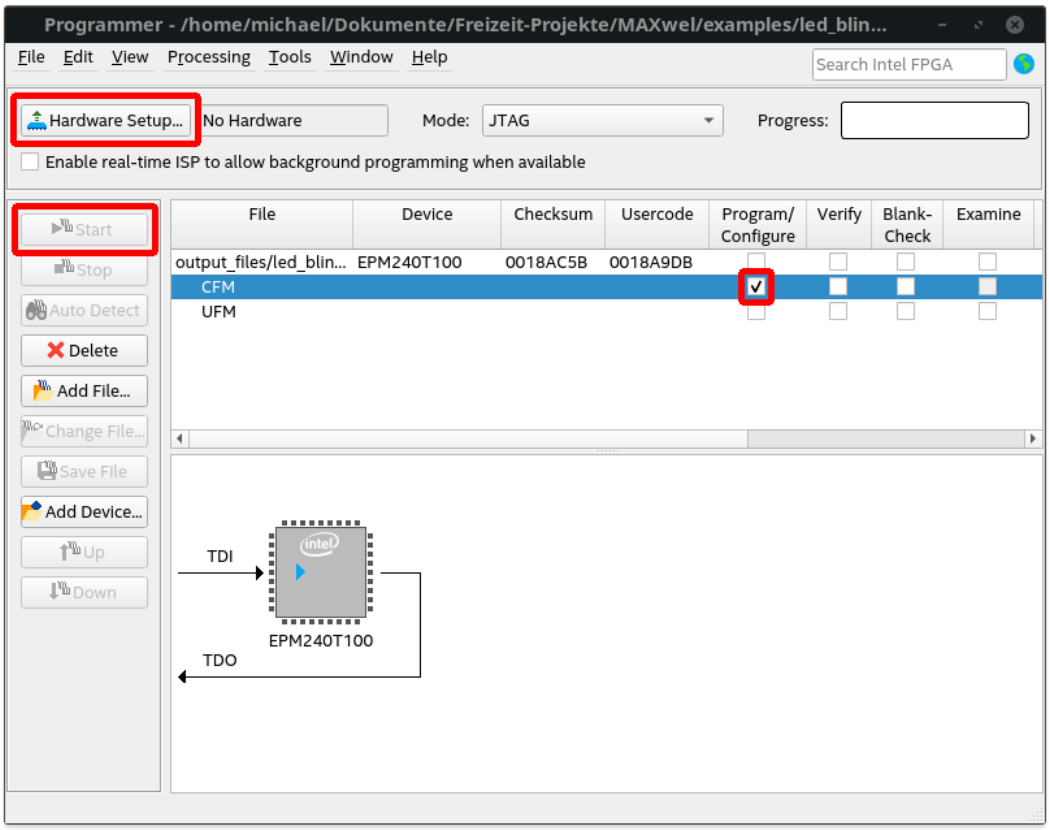

Figure 2.4: Important sections of the Programmer dialog