

# Das erste FPGA-Projekt mit Quartus II Webedition und dem DE2-115 Board

Die Programmierung eines FPGAs mit der Quartus-Software ist für den Anfänger erst mal nicht so einfach zu bewerkstelligen. Mit dieser Anleitung wird anhand eines einfachen Beispiel Schritt für Schritt gezeigt, wie es geht.

Grundidee: Auf dem DE2-115 Board sind u.a. 4 Taster (Keys) und 18 rote Leds als Eingabe- und Ausgabemöglichkeiten vorhanden. In unserem Beispiel soll durch das Drücken des Key0 die LedR0 aufleuchten.

- 1. Schritt** Beim Starten der Software kann für ein neues Projekt der Project Wizard verwendet werden. Dies erleichtert die Erstellung sehr.



- 2. Schritt** Auf der ersten Seite wird zuerst das Arbeitsverzeichnis ausgewählt. Es empfiehlt sich, für jedes Projekt vorher einen eigenen Ordner anzulegen. Dabei unbedingt auf Sonderzeichen, Umlaute und auch Zahlen verzichten. U.U. Bekommt man sonst Fehlermeldungen im Programm.  
Wichtig ist auch der letzte (rot markierte) Eintrag. Dieser Name muss später in der Schnittstellenbeschreibung genau so eingegeben werden, sonst kommt es zu einer Fehlermeldung. Für unser Projekt wählen wir hier **Led\_top** als Namen. Weiter mit **Next**. Die Seite 2 kann mit **Next** übersprungen werden.



New Project Wizard

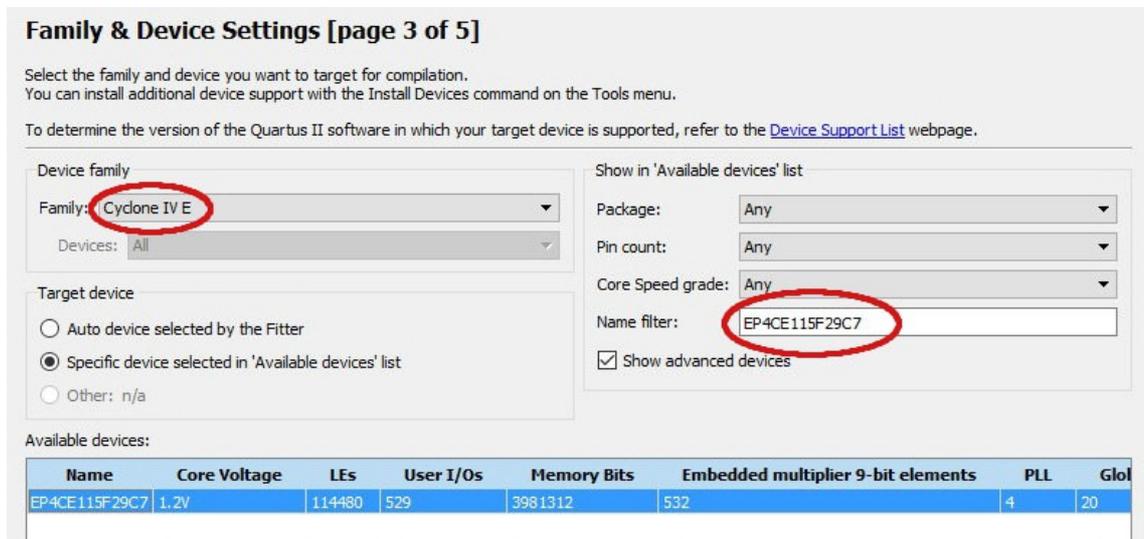
### Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?

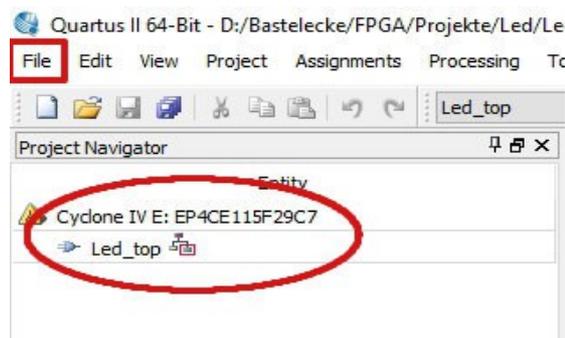
What is the name of this project?

What is the name of the top-level design entity for this project? This name is case sensitive

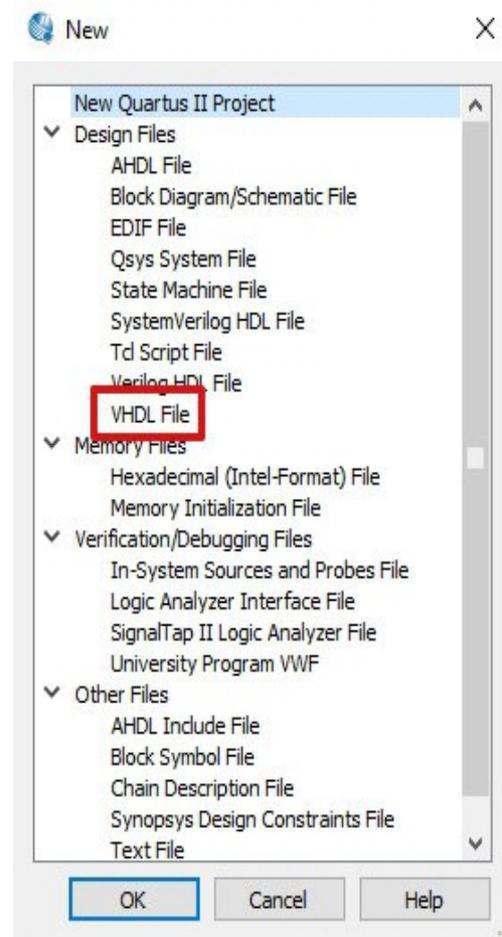
**3. Schritt** Auf der Seite 3 muss zuerst das richtige FPGA eingestellt werden. Auf dem DE2-115 Board ist ein **Cyclone IV E**-FPGA verbaut. Die genaue Bezeichnung lautet **EP4CE115F29C7**. Unbedingt in die Auswahl unten auf das angezeigte Device klicken, bis es blau hinterlegt wird. Mit **Finish** wird an dieser Stelle der Wizard beendet.



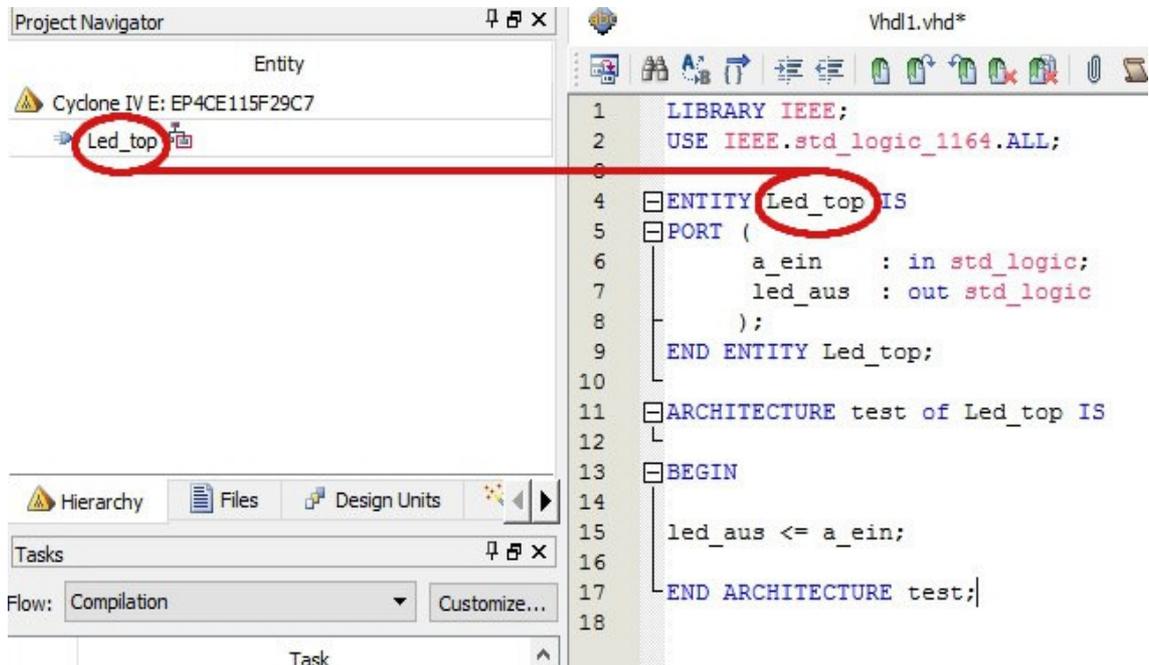
**4. Schritt** Wenn alles richtig eingestellt wurde, findet man im Projekt-Navigator nun das richtige FPGA und das Top-Level Entity „Led-top“.



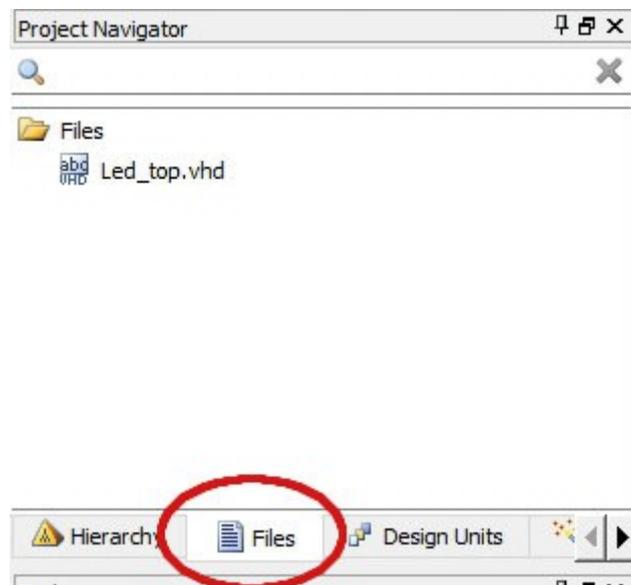
Mit **File New** öffnet sich ein neues Fenster. Ausgewählt wird **VHDL File** und mit **OK** bestätigen.



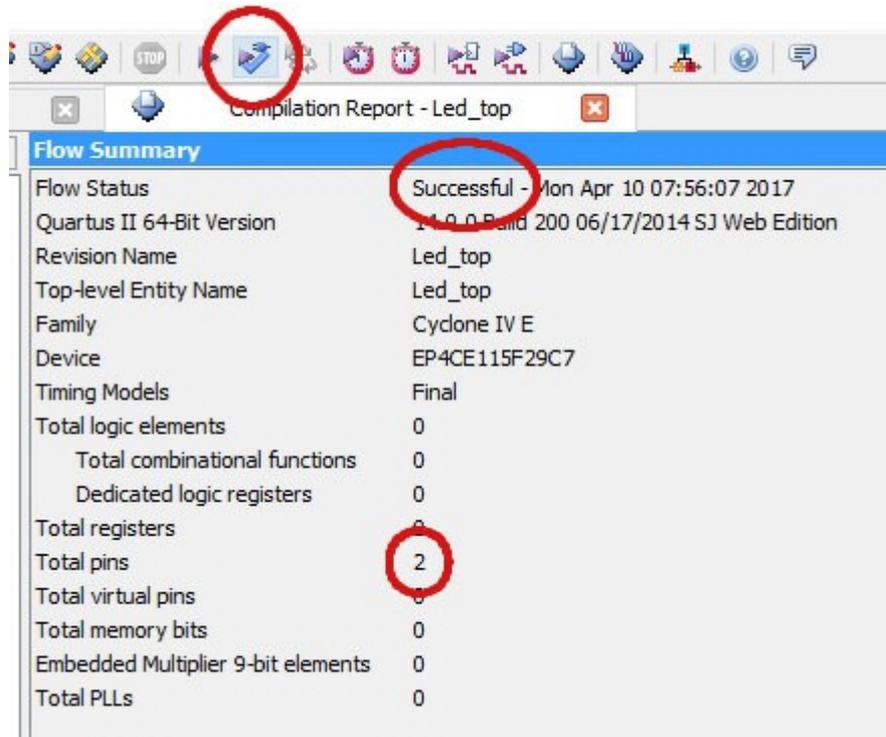
**5. Schritt** In dem neuen Fenster öffnet sich der VHDL-Editor. In diesen gibt man das unten abgebildete Programm ein. Dabei muss der Name nach dem Schlüsselwort ENTITY unbedingt mit dem Namen links im Programm-Navigator übereinstimmen! Mit **File Save..as** wird das VHDL-Programm unter **Led\_top.vhd** gespeichert.



Klickt man im Projekt-Navigator auf **Files**, so sieht man nun das gespeicherte VHDL-Programm **Led\_top.vhd**.



**6. Schritt** Nun ist unser Projekt für den ersten Testlauf bereit. Mit der unten markierten Schaltfläche wird dieser gestartet. Dabei wird sowohl die richtige Syntax des VHDL-Files überprüft, als auch die Zuordnung der Ein- und Ausgabepins vorbereitet. Wenn alles richtig eingegeben wurde erscheint bei Flowstatus „Successful“ und bei Total pins die Zahl 2.

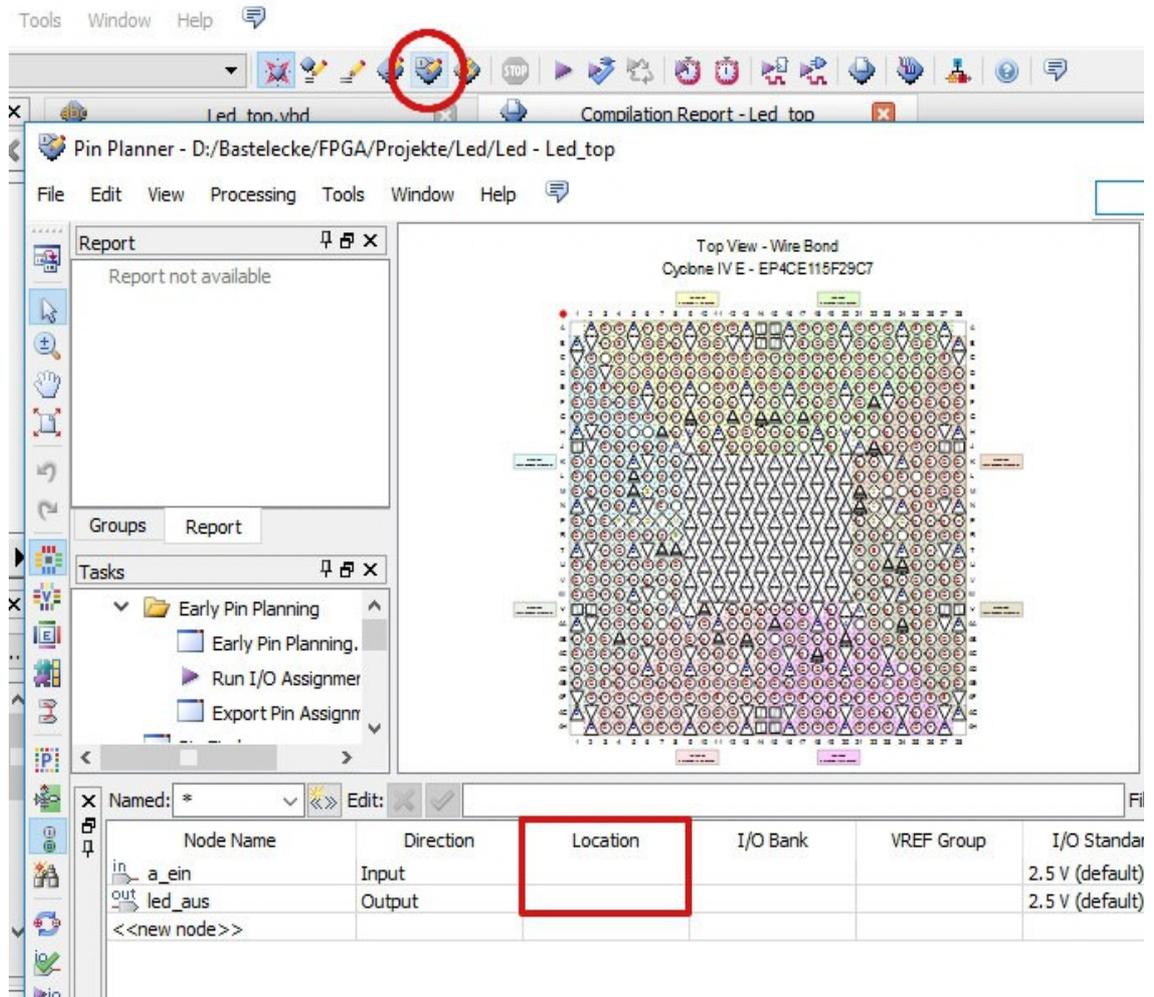


Jetzt müssen die im VHDL-Programm verwendeten Pins noch den physikalischen Pins auf dem DE2-115 Board zugeordnet werden. Wir ein anderes Board verwendet, so müssen die dort angegebenen Pins benutzt werden. Aus dem Handbuch des DE2-115 ist folgendes zu entnehmen:

Key0	PIN_M23
LEDR0	PIN_G19

Falls man andere Leds oder Taster verwenden möchte, gibt es für das DE2-115 Board eine tabellarische Übersicht aller Pins.

**7. Schritt** Mit der unten markierten Schaltfläche wird der PIN-Planer geöffnet. Unten findet man nun unsere beiden Pins *a\_ein* und *led\_aus* wieder. Klickt man nun in das Feld **Location** und gibt bei *a\_ein* dort **M23** an und bestätigt mit **Return** so ist der oben angesprochene PIN\_M23 dem Signal physikalisch zugeordnet. Das Gleiche macht man mit *led\_aus* und **G19**. Die hinter **Location** liegenden Felder werden automatisch ausgefüllt. Der Pin-Planer kann am Ende einfach ohne weitere Speicherung geschlossen werden. Nun ist das Projekt fertig zur Kompilation.



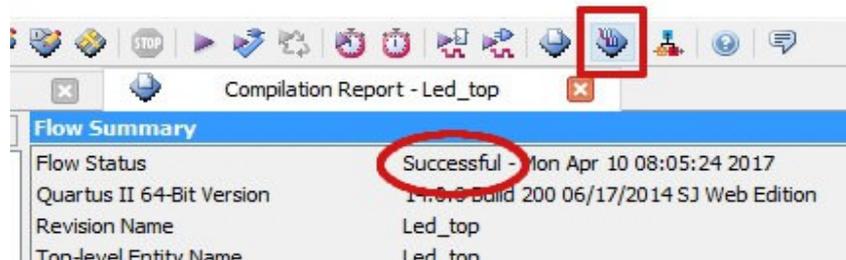
So sieht die fertige Pin-Planung aus.

Node Name	Direction	Location	I/O Bank
in_a_ein	Input	PIN_M23	6
out_led_aus	Output	PIN_G19	7
<<new node>>			

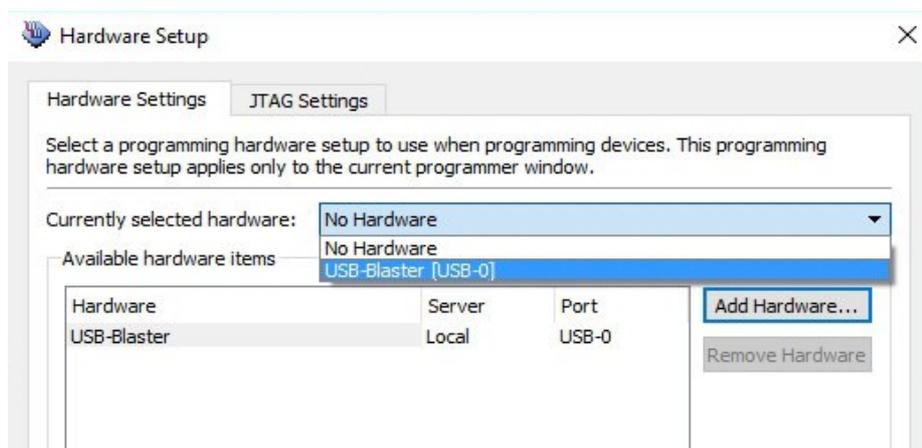
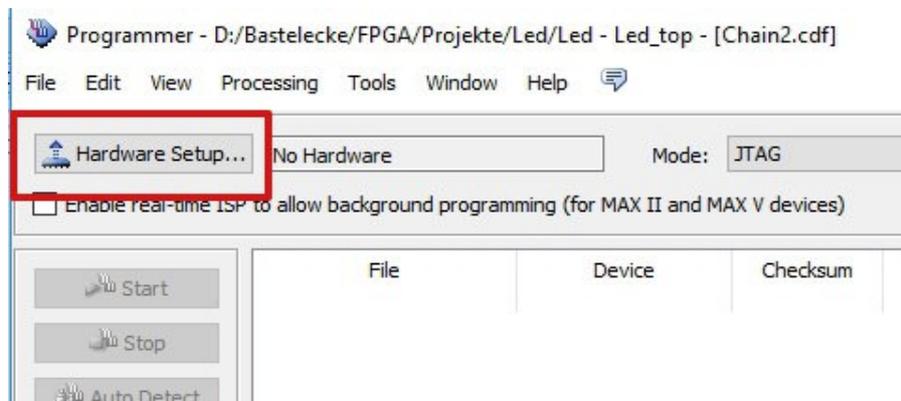
**8. Schritt** Die Kompilation wird mit der unten markierten Schaltfläche gestartet und dauert, je nach Rechner mehrere Minuten.



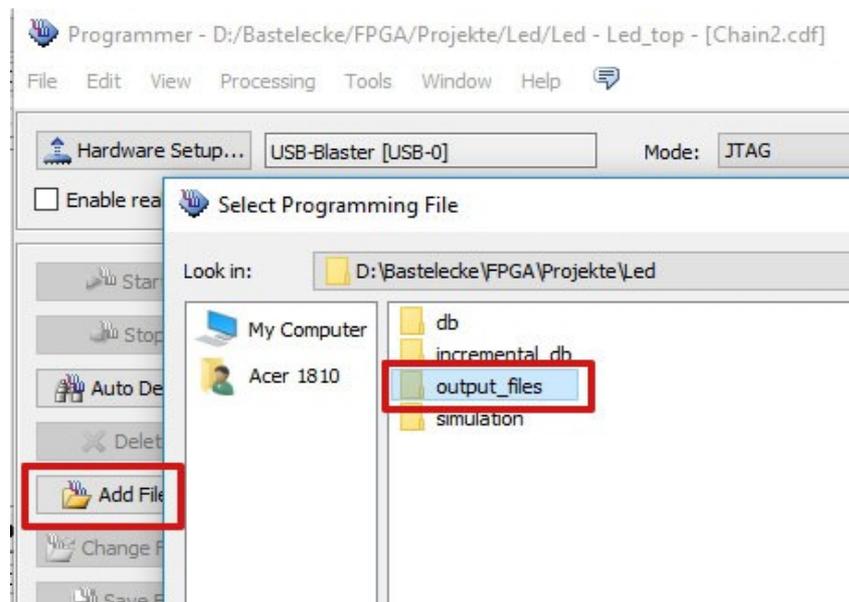
**9. Schritt** Ist sie erfolgreich gewesen („Successful“), so kann das Programm nun auf das FPGA-Board übertragen werden. Dazu wird die oben markierte Schaltfläche benutzt.



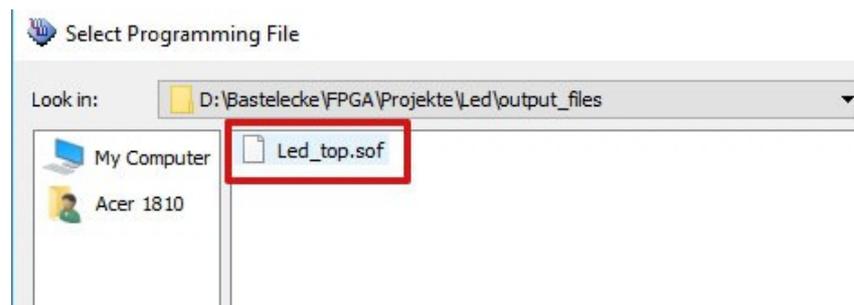
Beim ersten Benutzen des Programmers muss die verwendete Hardware richtig eingestellt werden. Mit **Hardware Setup...** öffnet sich eine neues Fenster, wo der **USB-Blaster** im Auswahlménü erscheinen soll. Falls dieser bei der Installation nicht mit installiert wurde, muss dies noch nachgeholt werden. Das USB-Kabel muss an den ganz linken USB-Port auf dem DE2-115 Board angeschlossen sein. Das Fenster wird mit **Close** geschlossen.



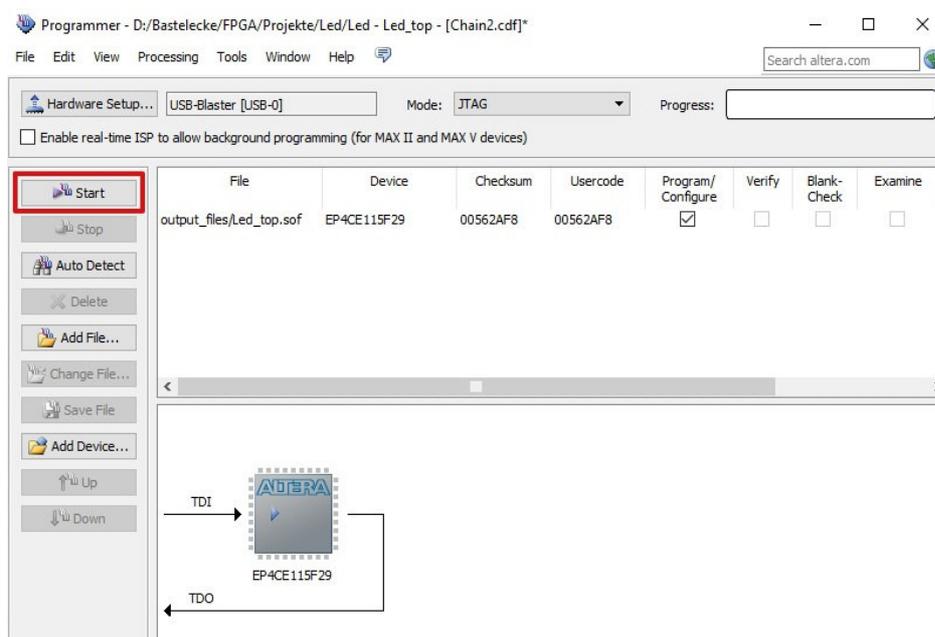
Mit **Add File** öffnet sich nun ein neues Fenster. Unser Programmer-File befindet sich im Ordner **output\_files**.



Das File **Led\_top.sof** wird ausgewählt.



Mit **Start** wird die Übertragung gestartet. Der kleine Schalter auf dem DE2-115 Board „Run/PROG“ (links unten auf dem Board) muss dabei aus „RUN“ stehen.



## Ergebnis

Auf dem Board leuchtet nun die **Led0** auf. Drückt man auf den Taster **Key0**, so erlischt die Led. Dieses zuerst einmal unerwartete Verhalten liegt an der Beschaltung der Taster. Dieser ist im ungedrückten Zustand mit einem Widerstand auf High-Level gezogen (Logisch „1“) und beim Drücken wird er auf GND (Logisch „0“) gezogen. Damit entspricht der ungedrückte Taster dem logischen Wert „1“ (die Led leuchtet) und der gedrückte dem logischen Wert „0“ (die Led ist aus).

Wenn man die übliche Funktionsweise möchte, muss man im VHDL-File nur die Zeile

```
led_aus <= a_ein;
```

in

```
led_aus <= not a_ein;
```

ändern.