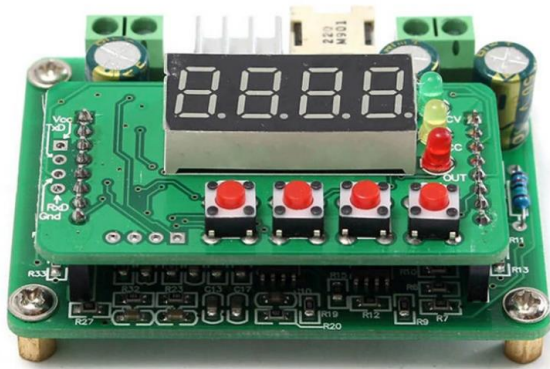


Programmierung des Odyssey 2

Der Odyssey 2 wird in 3 Stufen programmiert. Im ersten Schritt wird die MCU (PIC) programmiert, danach der Bootloader in den FPGA geschrieben und im letzten Schritt die eigentliche Firmware (oder auch mehrere verschiedenen Firmware) ebenfalls in den FPGA.

Grundbedingung ist, dass der Odyssey 2 funktioniert. Ich denke das ist selbstverständlich. Nur wissen wir dies zu diesem Zeitpunkt meistens noch nicht.

Wird der Odyssey 2 mit Spannung versorgt (12V) dann nimmt die unprogrammierte Platine einen Strom von ca. 3-4mA auf. Gemessen mit dem Mini Step Down DC-DC Wandler von Aliexpress. Optimal auch für Leute ohne geregeltm Labornetzteil. Strom und Spannung können eingestellt werden. Ich verwende es mit einer Lithium-Ionen-Batterie. Das Labornetzteil ist mir oft zu laut. Leistungswerte: 0-36V (Eingang 0-40V) und Ausgangsstrom maximal 3A. Preis: ca. 8-9 Dollar.



1. MCU Programmierung (PIC16F1827)

Dazu verwenden wir den Pickit 3 USB Programmer und die Software «Pickit 3 Programmer»

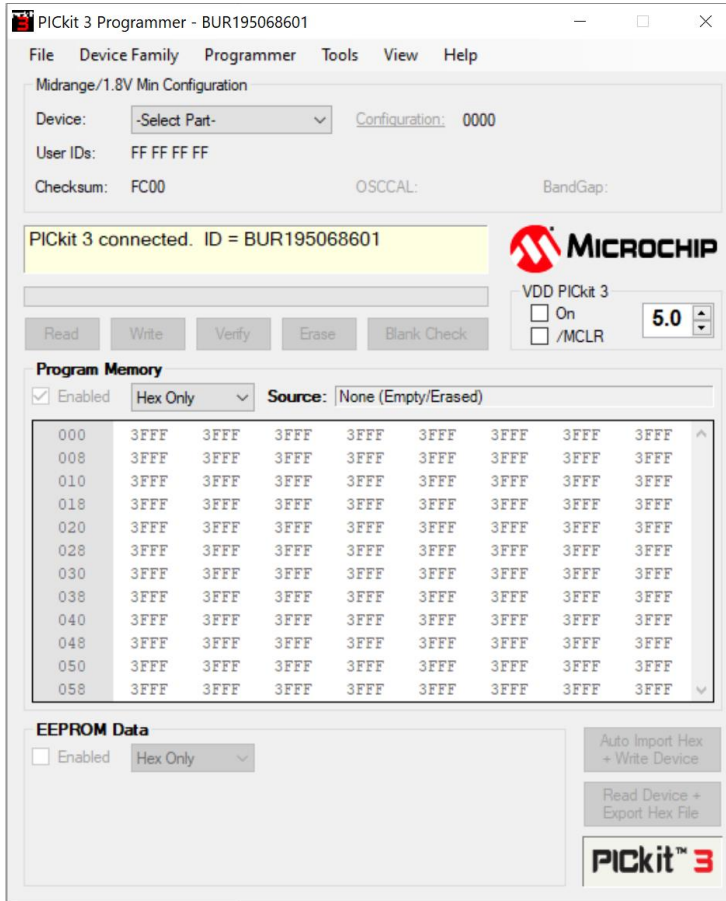


wir verbinden den Programmer mit dem Odyssey 2 und zwar mit der dafür vorgesehenen Pfostenleiste. Der Pin links aussen auf dem Odyssey 2 Board (MCLR) ist Pin1, auf dem Programmer wird PIN1 mit einem weissen Dreieck gekennzeichnet. Der Programmer hat 6 Pins, wir benötigen aber nur 5. Der 6. «hängt» in der Luft.

Der Odyssey 2 muss für die Programmierung der MCU nicht mit Spannung versorgt sein. Der Programmer versorgt den Pci mit der notwendigen Betriebsspannung. Wir können also die Versorgungsspannung am Odyssey 2 wieder entfernen.

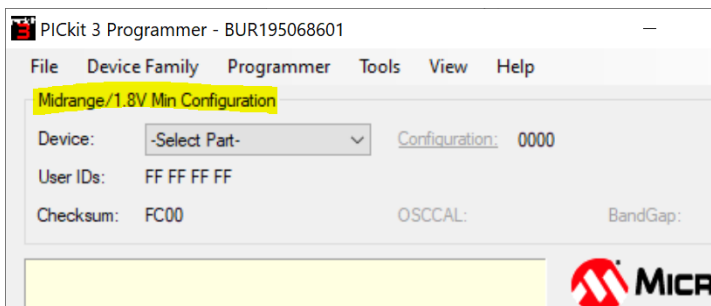
Ich gehe davon aus, dass Treiber Software für den Pickit 3 bereits geladen ist und dass dieser von der Software erkannt wird.

Wir starten nun die Pickit3 Software:



Nun klicken wir auf «Programmer» und wählen da «Manual Device Select» falls diese Option nicht bereits angewählt ist.

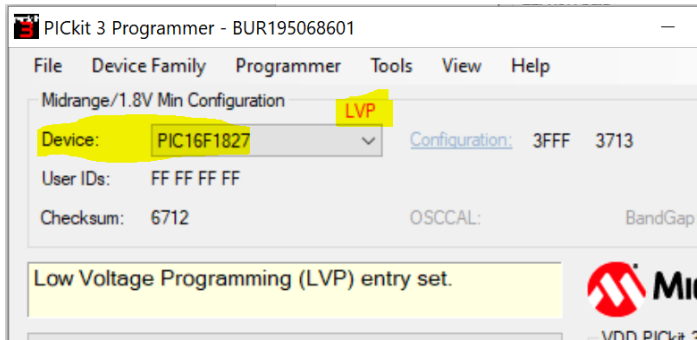
Als nächstens klicken wir auf «Device Family» und wählen da «Midrange -> 1.8V» aus.



Nun wähle unter «Device» den korrekten Controller aus: PIC16F1827

Bei älteren Boards ist es ausserdem wichtig, dass «LVP» (LVP=Low Voltage Programming) benutzt wird. Dies weil ein Pin mit dem FPGA verbunden ist und dieser nicht mehr als 5V verträgt. Beim aktuellen Board (V1.2) ist dies aber nicht mehr so, es würde also nichts

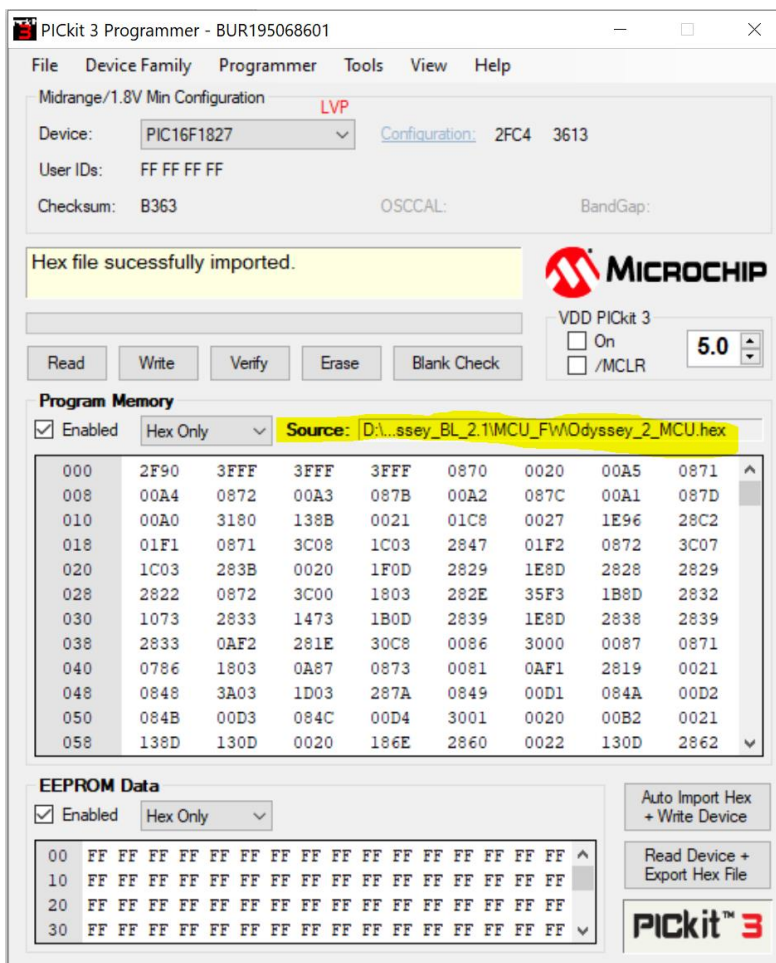
passieren. Trotzdem mache ich das so, man weiss ja nie 😊
Also unter dem Menü «Tools» den Punkt «User LVP Programm Entry» anwählen.



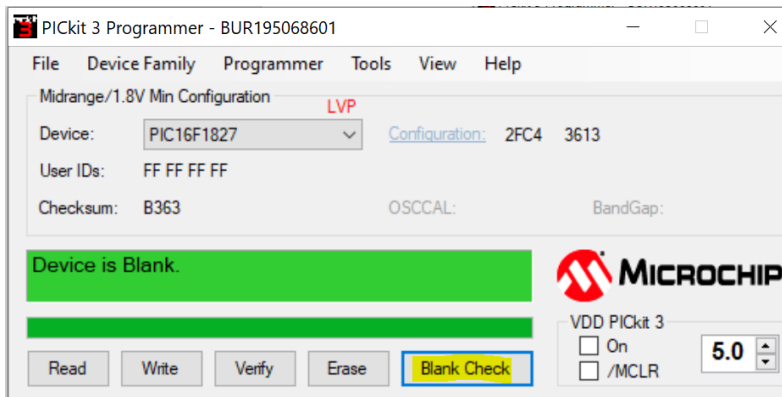
Nun haben wir die Pickit3 Software korrekt vorbereitet und können nun die MCU Firmware laden.

Dazu wählen wir «File» und dann «Input HEX»

Hier wählen wir die .hex Datei welche sich im jeweiligen Bootloader ZIP befindet. In meinem Fall ist das die Datei Odyssey_2_MCU.hex

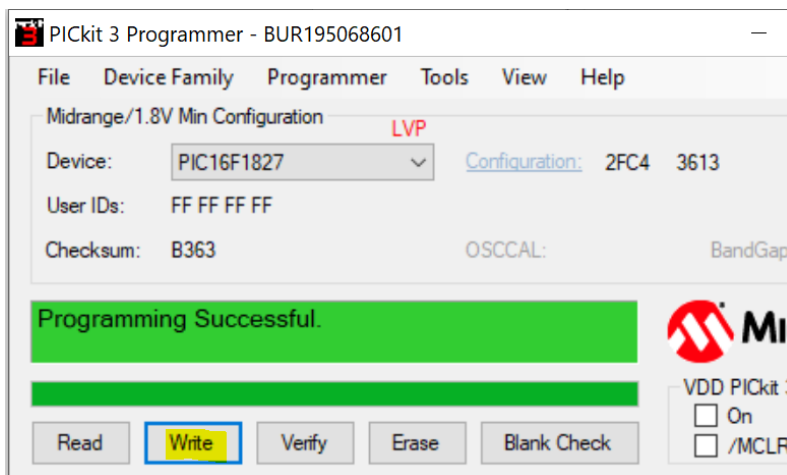


Nun klicken wir auf «Blank Check» und überprüfen ob der PIC auch wirklich leer ist, was wir bei einem neuen Device ja vermuten.



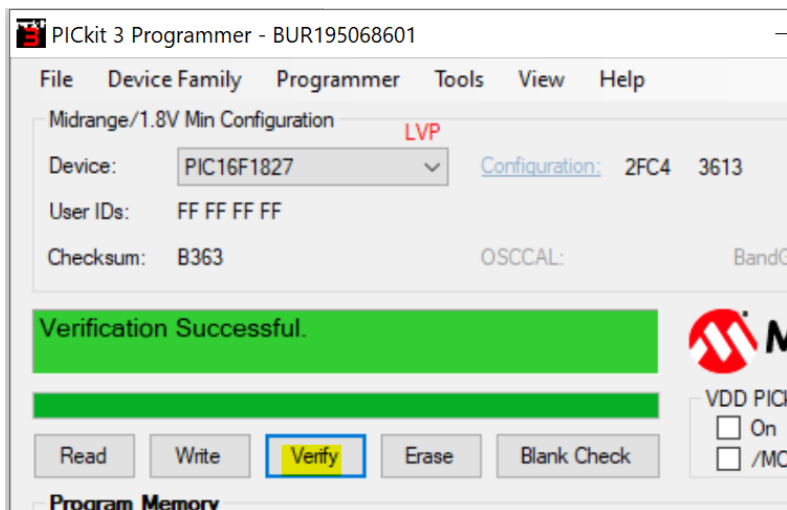
Es kommt wie erwartet. Der PIC ist leer.

Nun drücken wir «Write» und schreiben die Firmware in den PIC:



Wenn der Schreibvorgang erfolgreich war, dann sollte es so aussehen wie oben.

Zur Überprüfung kann nun noch der Vergleich des Programmspeichers im PIC mit unserem File gemacht werden:



Wenn auch das so aussieht, dann haben wir den PIC erfolgreich programmiert. Der Picket3 benötigen wir ab jetzt nicht mehr und kann vom Odyssey 2 entfernt werden.

Verbinden wir nun den Odyssey 2 mit 12V, müsste die Stromaufnahme auf 6-7mA angestiegen sein.

Nun drücken wir auf die Einschalttaste auf dem Odyssey 2 (STBY). Die Stromaufnahme steigt nun auf ca. 315mA und auf dem OLED Display steht «FPGA ERROR».

>>Bild «FPGA ERROR»<<

Damit wissen wir nun, dass der PIC funktioniert und dass er die Spannungsversorgung zum FPGA erfolgreich eingeschaltet hat. Der Fehler kommt aber davon, dass der FPGA noch keinen Bootloader hat und so nicht startet. Alle 4 LED's oberhalb des FPGA sollten nun leuchten (keine blinkt).

Nun sind wir bereit für den 2. Schritt!

Programmierung des Bootloaders in den FPGA

Dazu benötigen wir nun den «USB Blaster»

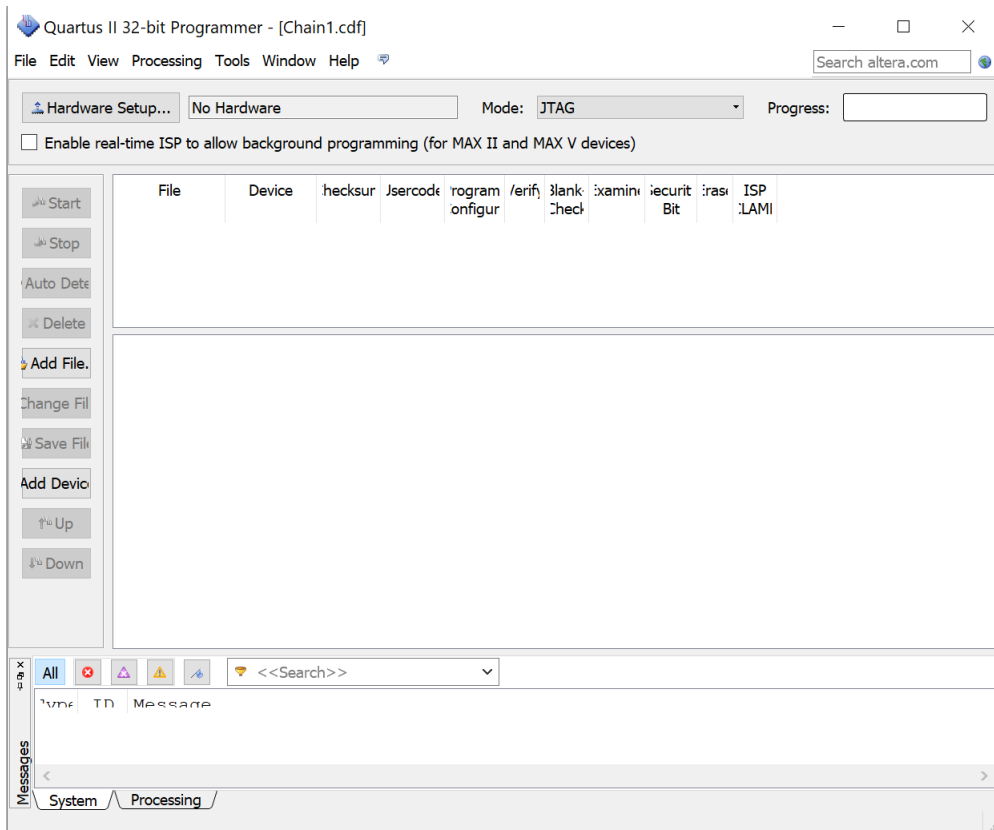


und die Software «Quartus II – 23bit Programmer». Diese kann unter folgendem Link heruntergeladen werden:

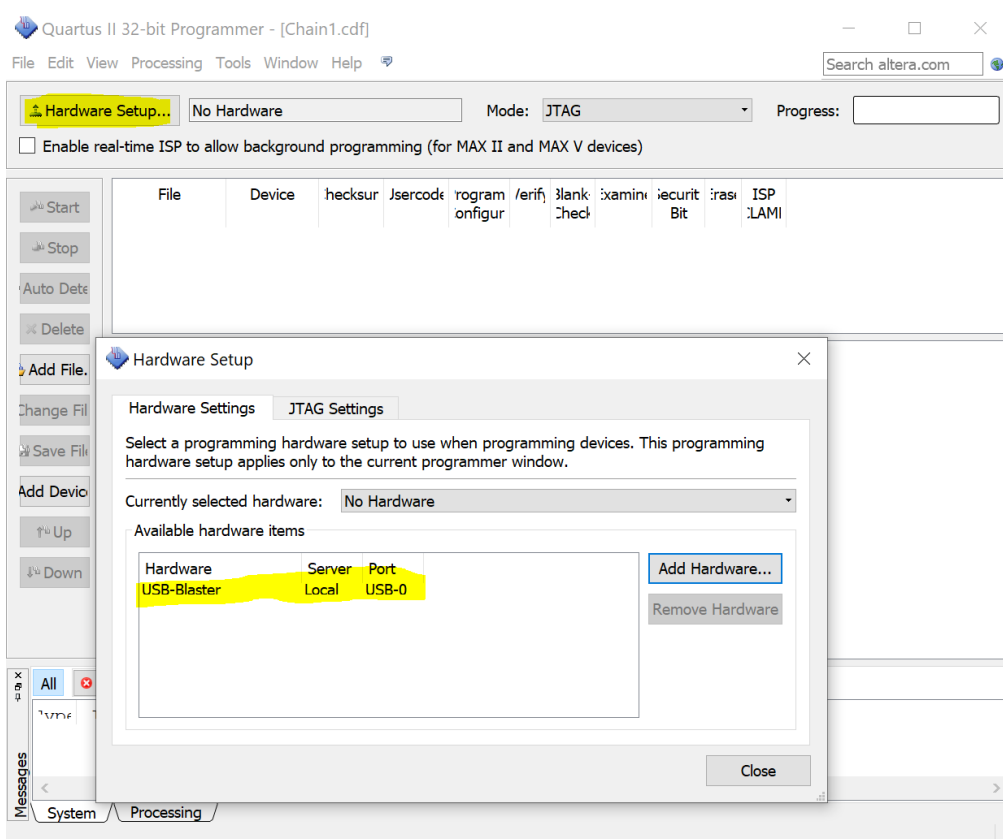
<https://www.intel.com/content/www/us/en/programmable/downloads/software/software/121.html>

Auch hier gehe ich davon aus, dass der USB Treiber bereits installiert ist und dass die Programmiersoftware den USB Blaster auch erkennt.

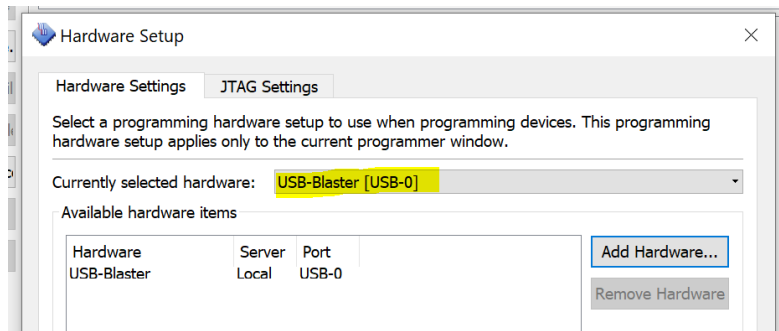
Wir starten nun den Software und erhalten diesen Stand:



Wir drücken nun auf «Hardware Setup» und unser USB Blaster sollte gefunden werden:

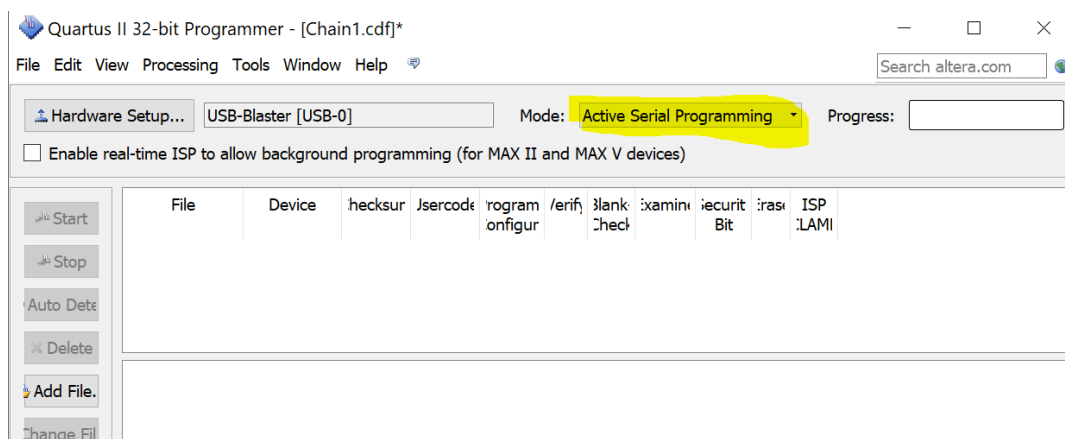


Unter «Currently selected Hardware» wählen wir nun unseren USB Blaster aus:



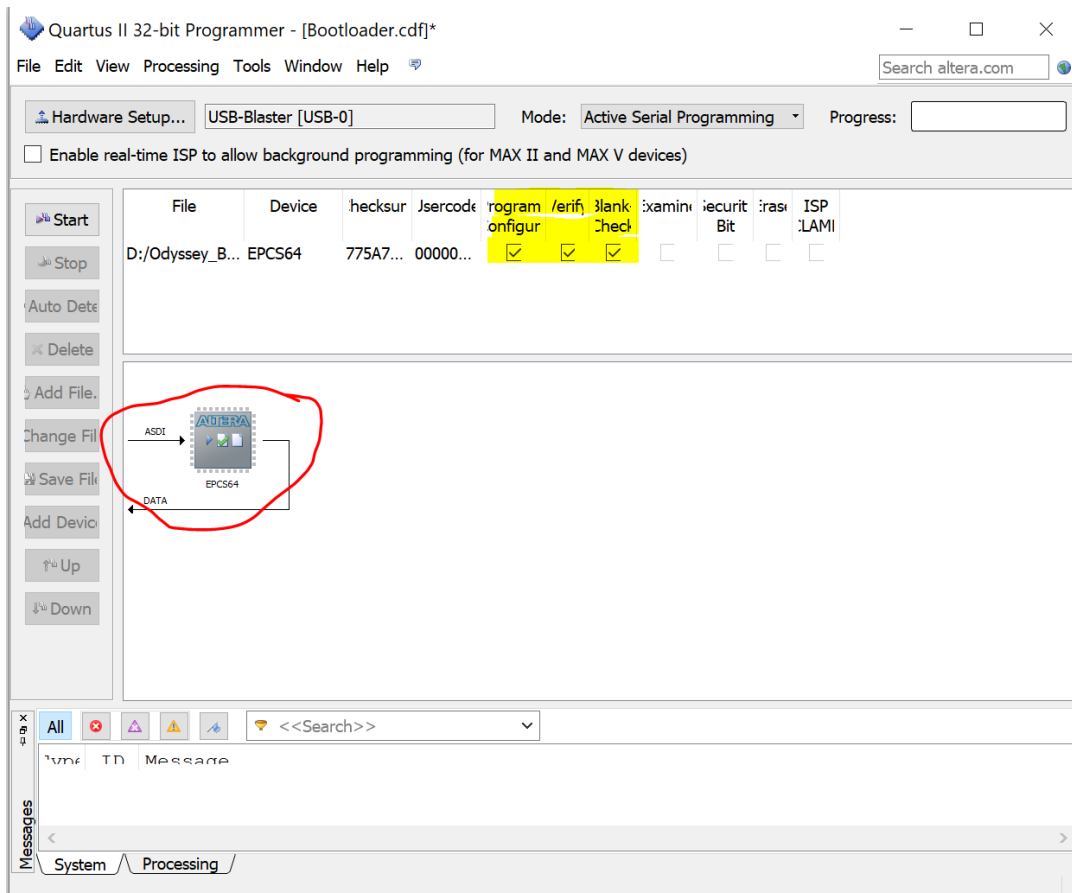
Danach drücken wir «Close»

Nun stellen wir noch den Programmiermode auf «Active Serial Programming» um:



Damit sind die Einstellungen im Quartus Programmer gemacht. Es kann nun also das Bootloader File geladen werden.

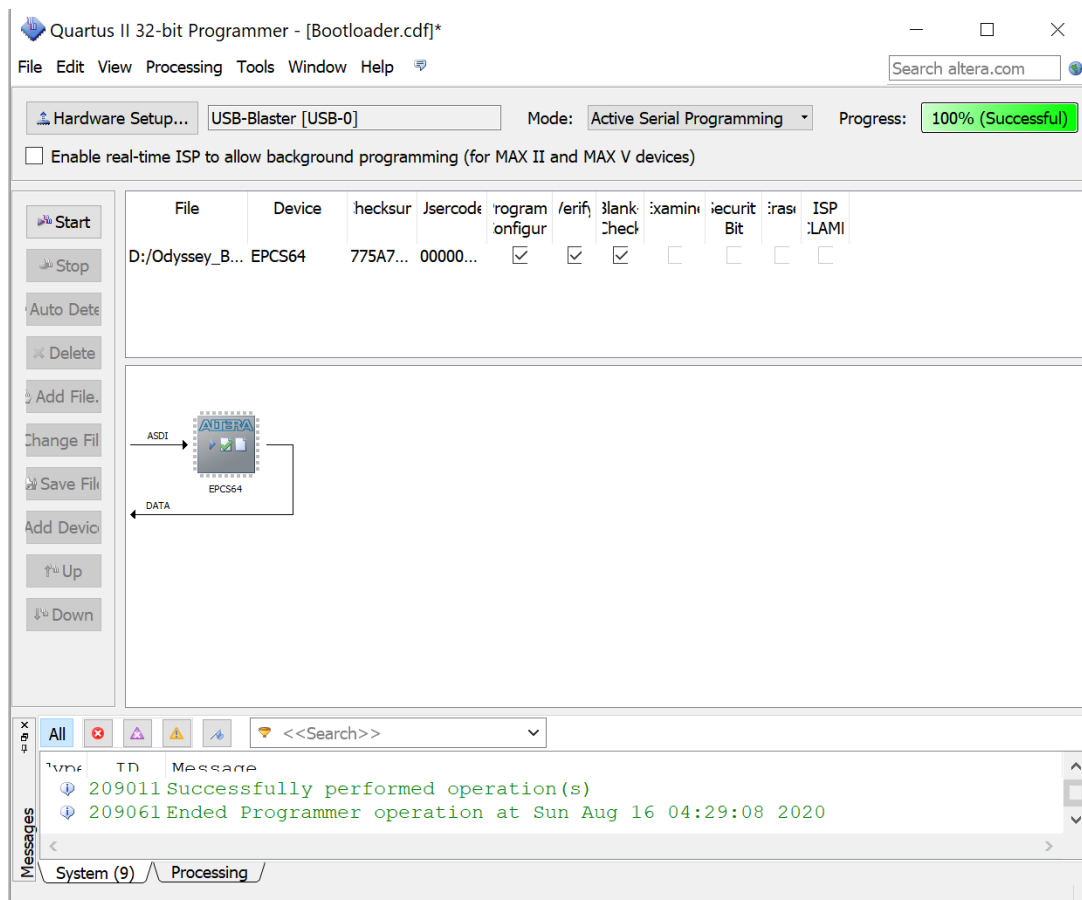
Unter «File» -> «Open» öffnest Du das entsprechende Bootloader File. In meinem Fall das File bootloader.pof aus dem Odyssey 2 Bootloaer 2.1.



Wir klicken noch die gelb markierten Kästchen an und damit wird «Start» aktiv. D.h. wir können den FPGA programmieren.

Wir drücken auf Start (der Programmiervorgang dauert ca. 20-30 Sekunden):

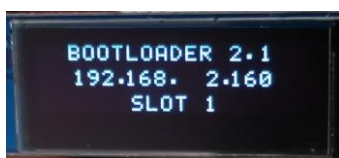
Danach sollten wir diesen Bildschirm erhalten:



Hier noch das vollständige LOG des Programmiers:

```
Info (209060): Started Programmer operation at Sun Aug 16 04:28:46 2020
Info (209018): Device 1 silicon ID is 0x16
Info (209044): Erasing ASP configuration device(s)
Info (209020): Blank-checking device 1
Info (209024): Programming device 1
Info (209022): Performing verification on device 1
Info (209018): Device 1 silicon ID is 0x16
Info (209011): Successfully performed operation(s)
Info (209061): Ended Programmer operation at Sun Aug 16 04:29:08 2020
```

Der Odyssey rebootet nun selbständig. Wir sollten nun auf dem OLED Display folgendes lesen können:



Gratulation. Dein FPGA scheint zu funktionieren und kommuniziert mit der MCU. Die 4 LED's oberhalb des FPGA's blinken nun folgendermassen:

LED1: 2x

LED2: 5x

LED 3 und 4: je 1x am Schluss des Prozesses, zusammen mit LED1 und LED2.

Dies passiert nun zyklisch, da der Bootloader versucht die Firmware zu laden, welche ja noch nicht programmiert ist.

Die Stromaufnahme schwank nun zwischen ca. 290mA und 320mA

Der Odyssey ist nun im Bootloader Modus. Mit dem nächsten und letzten Schritt laden wir nun die eigentliche Firmware in den Odyssey. Der USB Blaster kann nun wieder entfernt werden. Dazu den Odyssey 2 ausschalten und von der Spannung trennen.

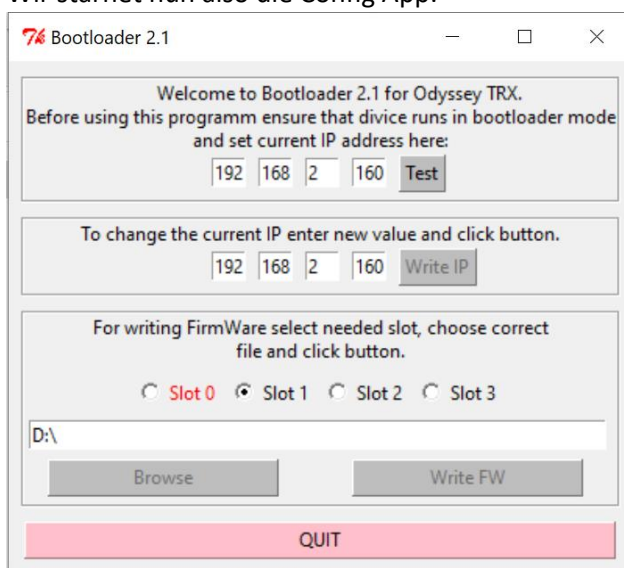
3. Programmierung der Firmware

Die Firmware wird mit der Software «BootLoader_2.1.pyw» geladen. Dies war zumindest bei mir etwas knackig. Ich hatte vorerst Mühe Python auf dem Windows 10 zum Laufen zu bringen. Nun klappt es. Ich werde dazu aber nicht näher eingehen, wie Python installiert wird, findet man im Internet.

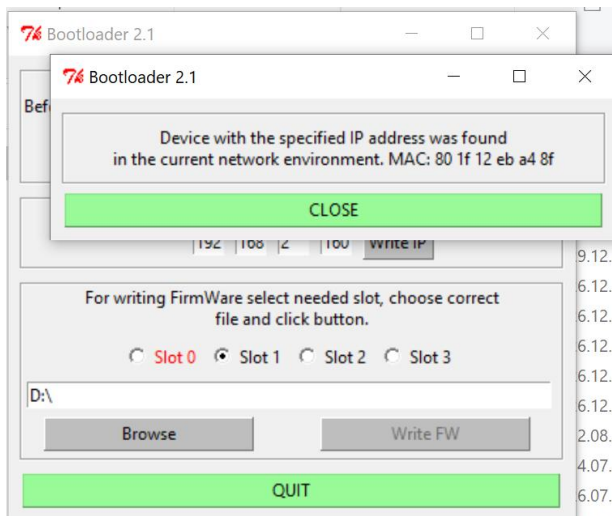
Der Odyssey 2 muss nun per Ethernet Kabel mit dem PC verbunden werden. Die entsprechende Netzwerkschnittstelle muss eine IP-Adresse bekommen im gleichen Subnet wie der Odyssey 2, beispielsweise 192.168.2.100. Die IP-Adresse des Odyssey 2 kann später über die BootLoader_2.1.pyw Applikation geändert werden.

Damit der Odyssey in den Bootloader Modus geht, muss nun an der «Key Buchse» ein Kopfhörer angesteckt werden, oder ein Dummy Stecker bei dem alle Kontakte miteinander verbunden sind (Kurzschluss) Damit geht der Odyssey beim Einschalten in den Bootloader Modus und wird von der Applikation erkannt.

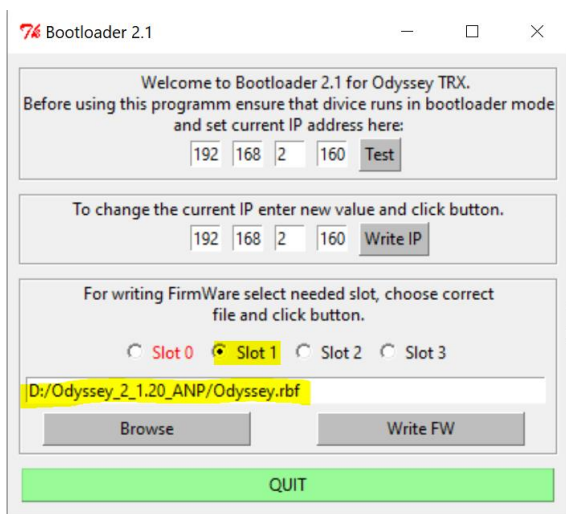
Wir starten nun also die Config App:



Wir drücken auf «TEST» und es sollte so aussehen:

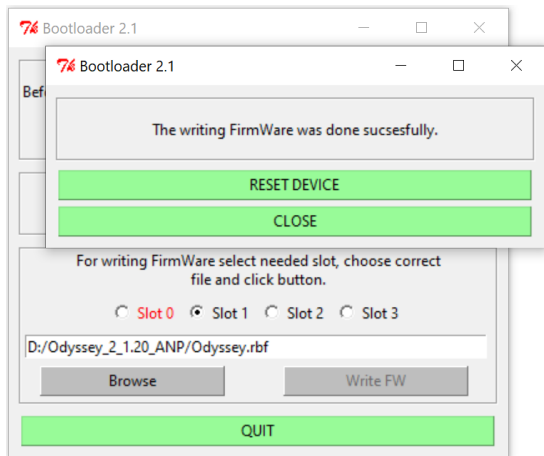


Test muss gemacht werden, da wir ansonsten das Firmware File nicht laden können. Wir klicken nun auf «Browse» und laden unser Firmware File, das wir in Slot 1 speichern möchten. Es handelt sich dabei um das File mit der Endung .rbf (nicht .pof):



Über den Punkt «Write FW» schreiben wir nun die Firmware in den Odyssey 2.

Wenn wir diese Ausgabe erhalten, dann haben wir es geschafft. Der Odyssey 2 ist programmiert.



Im OLED Display wird nun zuerst der Bootloader mit Version angezeigt, danach die Firmware die aus dem aktiven Slot geladen wird (im aktuellen Fall Slot 1). Kurz darauf ist der Odyssey 2 gestartet und im OLED Display steht dauerhaft «ODYSSEY»

Später sollten da noch Informationen zum SWR und Power angezeigt werden. Ob das jedoch realisiert wird, weiss ich nicht. Ist aber so von David angekündigt worden.

Die verschiedenen Slots

In Slot 1 bis 3 können jeweils verschiedene Firmware Versionen geladen werden und beim Start des Odysseys ausgewählt werden. Aktuell wird immer aus Slot 1 gestartet. Den Slot umstellen kann man über ein am Mic Anschluss angeschlossenes Yaesu MH30 und dessen Zusatz Tasten auslösen. Dies sollte während dem Bootvorgang möglich sein, das schaffte ich aber bis jetzt nie, da der Ody sehr schnell bootet. Wenn also der Slot umgestellt werden soll, dann wieder Kopfhörer oder Dummy Klinke in «Key» einstecken und den Odyssey 2 einschalten. Der Bootvorgang bleibt nun beim Bootloader stehen. Mit den Key Tasten des Mikrofons kann nun der Slot umgestellt werden. Danach Odyssey 2 ausschalten, Kopfhörer/Dummy Klinke aus «Key» Buchse entfernen und wieder einschalten. Nun wird die Firmware aus dem zuvor gewählten Slot gestartet.

Wie einleitend erwähnt, habe ich Slot 1 mit der 1.20 ANP belegt und auf Slot 2 eine 1.11 AUP Firmware. Also eine mit dem neuen Protokoll eine mit dem alten Protokoll. Damit wird der Odyssey zu verschiedenen SDR Software Programme kompatibel.

Damit ist der Transceiver komplett.

Mit aktiver SDR Software und einem aktiven RX Kanal springt die Stromaufnahme nun auf ca. 450mA (Wert muss ich nochmals ermitteln. Ist so aus dem Gedächtnis hingeschrieben, aber die Stromaufnahme steigt auf jeden Fall, sobald der TRX in Betrieb ist.

TX: Da meiner TX mässig aktuell noch nicht fertig ist, es fehlen immer noch die 2 Trafos, ich hatte noch nicht die Lust dazu gefunden, diese zu wickeln, kann ich dazu noch keine Angaben machen.

Viel Spass!

Version 0.1, 16.08.2020 HB9TRT