

Contents

1	Software	2
1.1	Quartus (Quartus Prime 18.1) Lite Edition.....	2
1.2	Download 7Zip:.....	2
1.3	Arduino IDE 2.0.0	2
1.4	Template	2
2	Wie legt man ein neues Projekt an und kompiliert dieses?	3
2.1	FHS-Project-Template Ordner "FHS_DIG_VidorProject" kopieren.....	3
2.2	Im neuen Pfad Quartus durch Doppelklick auf MKRVIDOR4000.qpf starten:	3
2.3	Jetzt können alle IOs des FPGAs im Schematic Design verwendet werden.	3
2.4	CMD-File	4
2.5	Kompilieren des Projekts durch Doppelklick auf ‘Compile Design’ im linken unteren Bereich der IDE starten.	4
2.6	Erzeugen von app.h	5
2.7	Arduino Sketch Template vorbereiten.....	5
2.8	Laden des Programms mittels Arduino Sketch.....	5
2.9	Aufbauen	6

1 Software

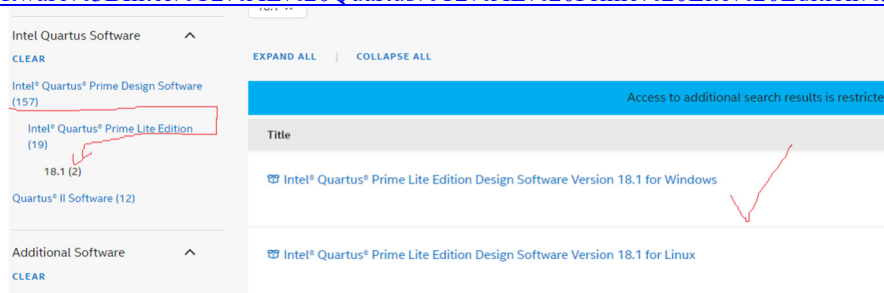
Die Folgende Anleitung ist für Windows ausgelegt!

1.1 Quartus (Quartus Prime 18.1) Lite Edition

Wird für Mac nicht angeboten...

Download unter:

<https://www.intel.com/content/www/us/en/collections/products/fpga/software/downloads.html?s=Newest&f:guidetmD240C377263B4C70A4EA0E452D0182CA=%5BIntel%C2%AE%20Quartus%C2%AE%20Prime%20Design%20Software%3BIntel%C2%AE%20Quartus%C2%AE%20Prime%20Lite%20Edition%3B18.1%5D>



Dauer ca. 2-5 Stunden je nach Downloadgeschwindigkeit 5Gb

Die Datei muss im Anschluss entpackt werden. Dafür wurde 7zip verwendet. Es geht aber auch jedes andere ZIP-Programm für .tar

1.2 Download 7Zip:

Download unter:

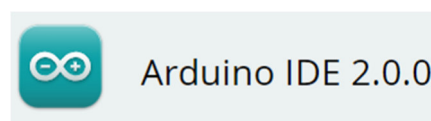
<https://www.7-zip.org/download.html>

Beim Starten die „FreeLizenz“ wählen.

1.3 Arduino IDE 2.0.0

Download unter:

<https://www.arduino.cc/en/software>



Dauer ca. 5 Minuten je nach Downloadgeschwindigkeit

1.4 Template

Download unter:

https://fhsalzburg.sharepoint.com/:f:/r/sites/TEAM-ITS-BAC1-FPGA/Freigegebene%20Dokumente/General/Arduino%20MKR%20Vidor%20Templates/00_FHS_DIG_VidorProject?csf=1&web=1&e=qbeKUb

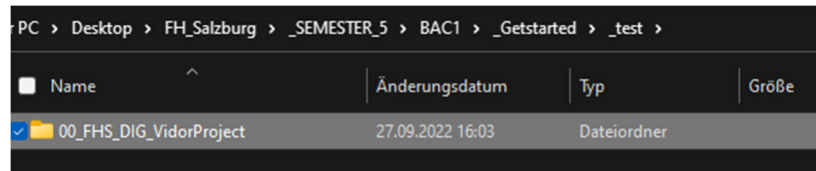
2 Wie legt man ein neues Projekt an und kompiliert dieses?

Weg über Schematic Editor / FHS Template.

Wichtig - Zu beachten:

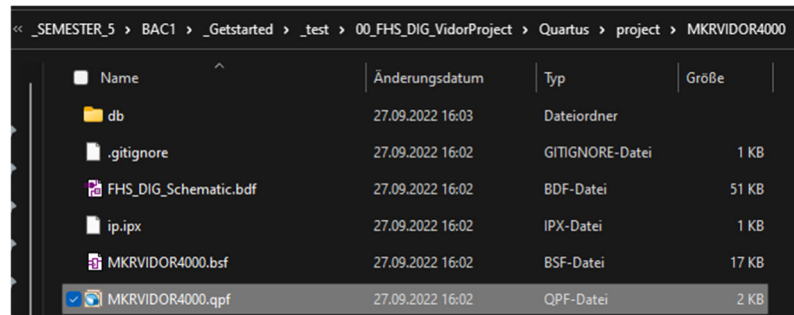
- 3.3V IOs. Anlegen von 5V kann die IOs zerstören.
- Nie denselben Pin als Output auf MCU und FPGA konfigurieren!

2.1 FHS-Project-Template Ordner "FHS_DIG_VidorProject" kopieren

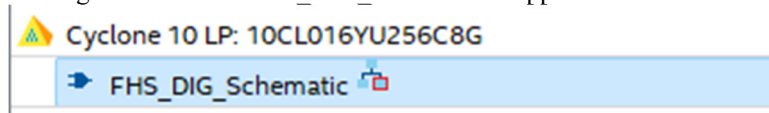


2.2 Im neuen Pfad Quartus durch Doppelklick auf MKRVIDOR4000.qpf starten:

- ..\Quartus\project\MKRVIDOR4000\MKRVIDOR4000.qpf

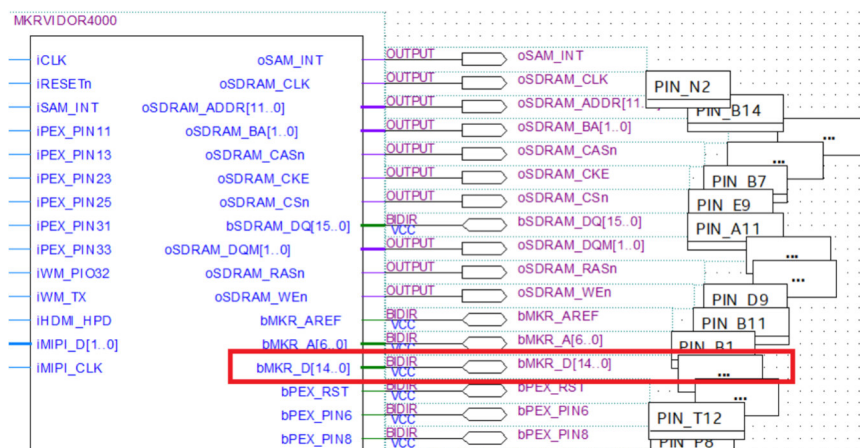


- Quartus zeigt möglicherweise "IP upgrade recommended" an. Dieses über das IP Upgrade Tool durchführen ("Perform Automatic Upgrade")
- Im Project Navigator links auf "FHS_DIG_Schematic" doppelklicken um das Design zu öffnen



2.3 Jetzt können alle IOs des FPGAs im Schematic Design verwendet werden.

Für uns interessant sind vor allem die digitalen IOs bMKR_D[14..0].

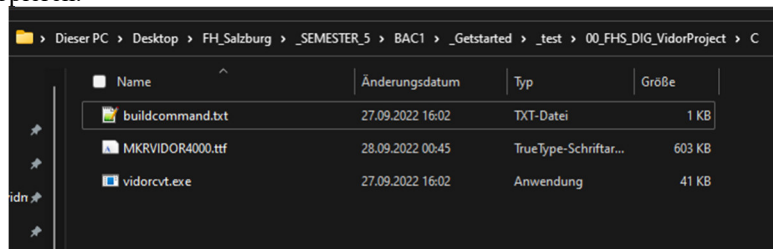


The screenshot shows a Windows File Explorer window with the address bar displaying the path: `\\BAC1\ -> _Getstartet -> _test -> 00_FHE_HD_VideoProjekt -> Querbus -> project -> MRVIDOR4000 -> output_files`. The left sidebar shows the navigation pane with 'Schneidbrett' selected. The main area displays a list of files in a table format:

Name	Änderungsdatum	Typ	Größe
MRVIDOR4000_arm.rpt	20.09.2022 00:45	PPT-Dates	15 KB
MRVIDOR4000_dome	20.09.2022 00:45	DONE-Dates	1 KB
MRVIDOR4000_Rr.rpt	20.09.2022 00:45	PPT-Dates	313 KB
MRVIDOR4000_fm.smssg	20.09.2022 00:45	MSG-Dates	1 KB
MRVIDOR4000_fm.summary	20.09.2022 00:45	SUMMARY-Dates	1 KB
MRVIDOR4000_flow.rpt	20.09.2022 00:45	PPT-Dates	11 KB
MRVIDOR4000_jam	20.09.2022 00:45	JAM-Dates	66 KB
MRVIDOR4000_jbc	20.09.2022 00:45	JBC-Dates	55 KB
MRVIDOR4000_jdi	20.09.2022 00:45	JO-Dates	1 KB
MRVIDOR4000_map.rpt	20.09.2022 00:44	PPT-Dates	115 KB
MRVIDOR4000_map.summary	20.09.2022 00:44	SUMMARY-Dates	1 KB
MRVIDOR4000_psn	20.09.2022 00:45	PSN-Dates	33 KB
MRVIDOR4000.pdf	20.09.2022 00:45	PDF-Dates	2,049 KB
MRVIDOR4000_rsf	20.09.2022 00:45	RSF-Dates	146 KB
MRVIDOR4000_sld	20.09.2022 00:45	SLD-Dates	1 KB
MRVIDOR4000_sof	20.09.2022 00:45	SOF-Dates	486 KB
MRVIDOR4000_sta.rpt	20.09.2022 00:45	PPT-Dates	350 KB
MRVIDOR4000_sta.summary	20.09.2022 00:45	SUMMARY-Dates	2 KB
MRVIDOR4000.stp	20.09.2022 00:45	TrueType-Schrift...	603 KB

2.6 Erzeugen von app.h

- `..\Quartus\project\MKRVIDOR4000\output_files\MKRVIDOR4000.ttf` in das C Verzeichnis des Projekts kopieren.



- Mit `cmd` in das C Verzeichnis des Projekts wechseln und `app.h` erzeugen: `vidorcvt < MKRVIDOR4000.ttf > app.h` (eventuell Antivirus Programm ausschalten)

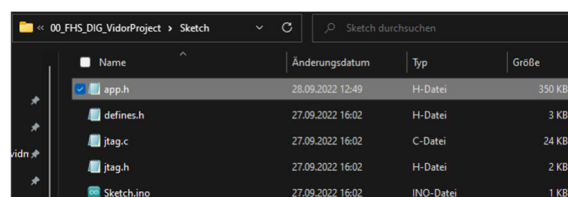
```
C:\Users\phili\Desktop\FH_Salzburg\SEMESTER_5\BAC1_Getstarted_test\00_FHS_DIG_VidorProject\C>vidorcvt < MKRVIDOR4000.ttf > app.h
C:\Users\phili\Desktop\FH_Salzburg\SEMESTER_5\BAC1_Getstarted_test\00_FHS_DIG_VidorProject\C>
```

- `app.h` enthält jetzt die selbe Information wie das `ttf` File (Hex FPGA Inhalt), allerdings mit benötigter inversierter Bitfolge.

Es handelt sich also quasi um einen Hex-Dump des FPGA-Inhalts.

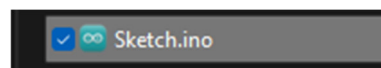
2.7 Arduino Sketch Template vorbereiten

Erzeugte 'app.h' nach '`...\Sketch\`' kopieren

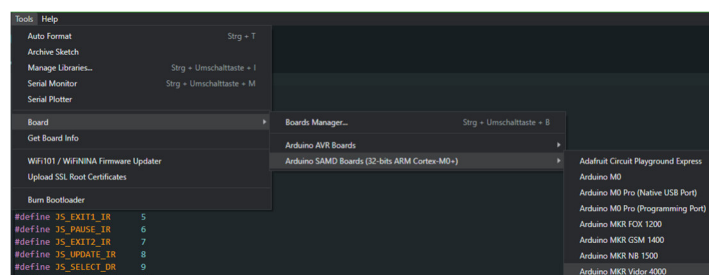


2.8 Laden des Programms mittels Arduino Sketch

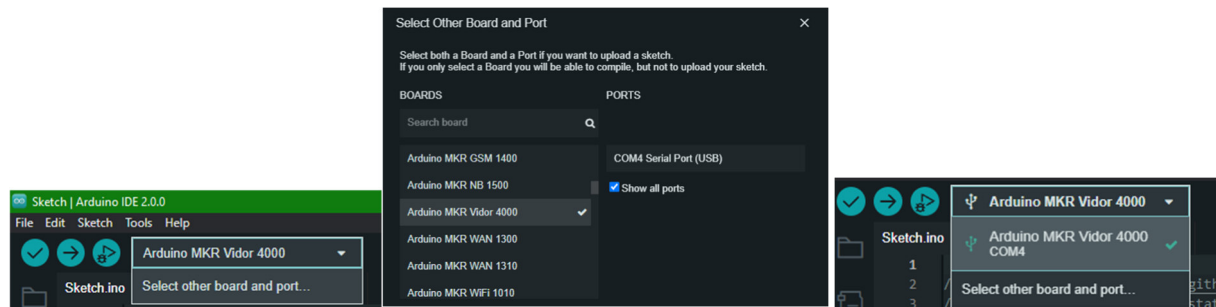
- '`\Sketch\Sketch.ino`' mit Arduino IDE öffnen



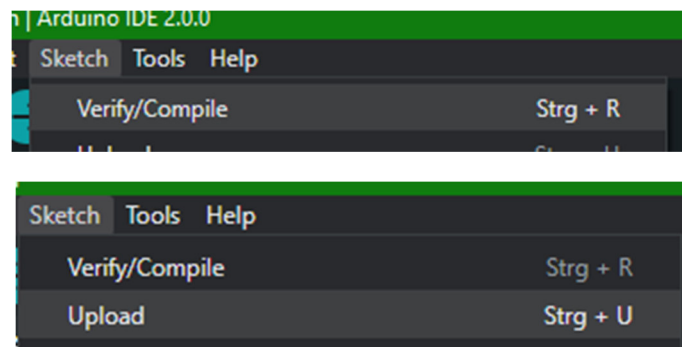
- Unter 'Tools - Board: - Board Manager' nach 'Arduino SAMD Boards' suchen und installieren, wenn nicht installiert.
- Verbindung herstellen mit 'Tools - Board: - Arduino SAMD Boards - Arduino MKR Vidor 4000'



- 'Tools - Port' - Port auswählen, an dem das Board verbunden ist



- 'defines.h' definiert, dass 'app.h' als Payload auf das FPGA geladen wird
- In der setup() Routine können optional Pins der MCU konfiguriert werden
- (Sollte für unser Digitaltechnik Projekt nicht der Fall sein)
- In der main loop() kann optional Code für die MCU implementiert werden
- (Sollte für unser Digitaltechnik Projekt nicht der Fall sein)
- FPGA über 'Sketch - Upload' programmieren



Achtung:

Nie denselben Pin als Output auf MCU und FPGA konfigurieren!

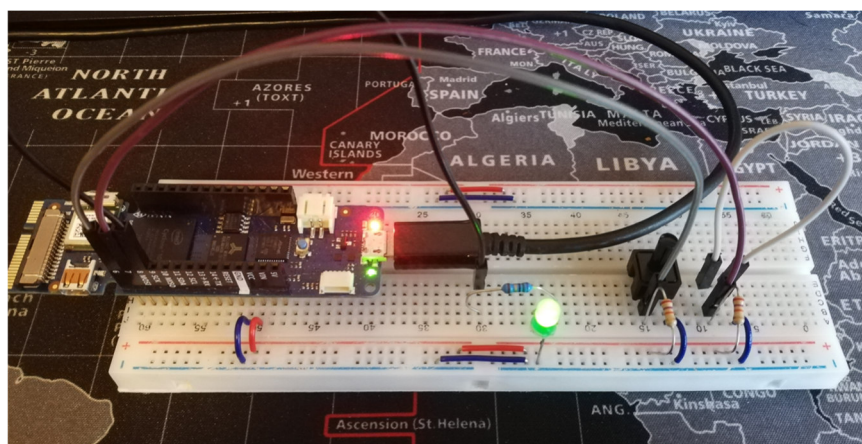
Pin mapping:

C:\Users\Hannes Waclawek\Documents\Git\Arduino MKR Vidor\VidorFPGA\constraints\MKRVIDOR4000

Pin IP folder:

C:\Users\Hannes Waclawek\Documents\Git\Arduino MKR Vidor\VidorFPGA\ip

2.9 Aufbau



Pin -> 6 & 7 Input mit PullUp (25kOhm laut Degenblatt FPGA)

Pin 8 -> Output

INFO: Logisch ist der Schalter mit Pullup ein Öffner → der verbaute Schalter ist somit invertiert!