

Coocox-IDE in Kombination mit stm32F4Discovery und Beispielprojekten

Installation der IDE:

- Download der CoIDE von der Seite http://www.coocox.org/CooCox_CoIDE.htm#
- Installation der CoIDE
- Download der GCC-Toolchain von der Seite https://launchpad.net/gcc-arm-embedded/4.6/4.6-2012-q2-update/+download/gcc-arm-none-eabi-4_6-2012q2-20120614.exe
- Installation der GCC-Toolchain
- Starten der CoIDE und in der Menueleiste auf „Project“ → „Set Toolchain-Path“ klicken
- Nun den Toolchain Path angeben. Dies ist der bin-Ordner der zuvor installierten GCC-Toolchain. Also beispielsweise der Ordner: „C:\Program Files\GNU Tools ARM Embedded\4.6 2012q2\bin“.
- Download des neuesten ST-Link Treibers von der Seite http://www.st.com/internet/com/SOFTWARE_RESOURCES/SW_COMPONENT/SW_DRIVER/st-link_v2_usbdriver.zip
- Installation des ST-Link Treibers
- Für den STM32F4 werden Datenblätter benötigt:
- Das Reference-Manual:
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/REFERENCE_MANUAL/DM00031020.pdf
- Das Datasheet:
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/DM00037051.pdf
- Die Beschreibung des STM32F4Discovery-Boards:
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/USER_MANUAL/DM00039084.pdf

Nun ist alles eingerichtet und die Beispielprojekte können durch Doppelklick auf die entsprechende Projektdatei mit der Endung .cob geöffnet werden.

Beschreibung der Beispielprojekte:

Hinweis: Der interessante Beispielcode steht immer in der „main.c“

Projektordner	Beschreibung
STM32F4_GPIO_INPUT_OUTPUT	Die blaue Taste auf dem stm32f4Discovery wird abgefragt und eine LED geht an, sofern die Taste gedrückt ist
STM32F4_PWM	Erzeugt zwei PWM-Signale per Timer
STM32F4_TIMER_INTERRUPT	Ein Timer erzeugt einen Interrupt und eine LED blinkt
STM32F4_RANDOM_GENERATOR	Der Zufallsgenerator wird initialisiert und es wird eine Zufallszahl auf der USART ausgegeben.
STM32F4_USART	Ganze Strings mit Endezeichen werden interruptgesteuert empfangen und dann zurückgesendet.
STM32F4_ADC	Der ADC wird angesteuert und gewandelte Werte über die USART versendet.
STM32F4_DAC	Der DAC wird angesteuert und ein Spannungswert wird ausgegeben
Hinweis zur Verwendung des DMA: Immer Optimierungsstufe „OPTIMIZE MOST O-3“ einstellen, da es sonst Probleme geben kann.	
STM32F4_DMA_USART	Ein String wird per DMA über die USART ausgegeben
STM32F4_DMA_ADC	Der ADC misst zwei Eingangskanäle und per DMA werden diese abgespeichert und per USART versendet.
STM32F4_DMA_DAC	Eine Sinuskurve wird mittels DMA und DAC erzeugt
STM32F4_DMA_PORT	Simple Beispiel für einen DMA-Transfer, welcher eine LED zum Blinken bringt.
STM32F4_DMA_PWM	Beispiel wie per DMA praktisch beliebig viele PWM-Signale auf beliebigen Ausgabepins erzeugt werden können ohne Interrupt und ohne CPU-Belastung.

Öffnen/Kompilieren eines Beispielprojektes:

- Öffnen des Ordners „STM32F4_GPIO_INPUT_OUTPUT“
- Doppelklick auf die entsprechende Projektdatei mit der Endung .cob
- In der Menueleiste auf „Debug“ → „Debug Configuration“ → „STM32F4_GPIO_INPUT_OUTPUT.configuration“
- Den Adapter auf „ST-Link“ den Port auf „SWD“ und die Max Clock auf „1M“ und dann auf „Apply“ klicken und dann auf „Close“
- In der Menueleiste auf „Projekt“ → „Rebuild“ klicken. Nun müsste das Projekt kompiliert werden. Falls es nicht funktioniert kontrollieren ob der „Toolchain Path“ richtig angegeben ist (Siehe Beschreibung erste Seite in diesem Dokument)
- Nach dem Kompilieren das stm32F4Discovery Board anschließen und auf der Menueleiste auf „Flash“ → „Program Download“.
- Wenn man nun auf die Blaue Taste drückt sollte die rote LED aufleuchten

Falls etwas nicht klappen sollte gibt es ausführlichere Hinweise auf der Seite
<http://www.coocox.org/Index.html>