

STM32

Schon seit längerem wollte ich mich mit den Cortex M Prozessoren beschäftigen, weil ich in vielen Bereichen an die Leistungsfähigkeit der ATMEL Prozessoren gestossen bin. Sei es Dateisysteme, Prozessorgeschwindigkeit, Busbreite, Flash und RAM Größe usw.

Als dann der Franzis Verlag mit einem sehr interessanten Projekt auf mich zu kam, war das für mich der Auslöser, mich mit diesen Prozessoren zu beschäftigen.

Also habe ich mir aus China ein kleines Entwicklungsboard besorgt.



Natürlich braucht man dazu auch einen entsprechenden Programmer. Auch den hab ich mir aus China kommen lassen.

Hier mal die Daten:

STM32 ARM Minimum Systemplate STM32F103C8T6

STM32F103C8T6 Prozessor.

- Cortex M3
- 72MHz
- Gehäuse LQFP48 (Deswegen sind nur 2 komplette Ports herausgeführt.)
- 64kb Flash
- 20kb RAM
- 32 kHz Oszillator für eine RTC mit Kalibrierungsmöglichkeit
- Vbat für RTC
- 2x12Bit ADC mit 16 Kanälen, Temperatursensor
- 7-Kanal DMA Controller für Timer, ADC, SPIs, I2Cs and USARTs
- 5V tolerante IO's. 16 ext. Interruptvektoren
- Debug über SWD. (Debug Wire für Programmierung und Debuggen mit nur 2 Leitungen.)
- 7 Timer, davon 3 16-bit Timer, jeder mit bis zu 4 IC/OC/PWM oder Pulsezähler und Encoder eingang (Drehgeber)
- 16-bit, motor control PWM timer mit Notaus
- 2 Watchdog timers
- 24-bit SysTick Timer
- bis zu 9 Kommunikationsinterfaces, 2x I²C, 3x USART, 2x SPI 18MBit, CAN 2.0B Aktiv, USB 2.0 Fullspeed
- Hardware CRC Berechnung

Das Board hat die minimale Bestückung. D.h. alle Pins sind einfach auf Steckleisten herausgeführt. 8 MHz Quarz ist drauf (werden dann intern über PLL auf 72MHz gebracht) 3V3 Versorgung und USB Anschluss. Noch ein Resettaster und ein Umschalter für die Spannungsversorgung (USB/SWD) Boot Jumper und extra Anschlüsse für die erste USART und der 20-Polige JTAG (wo auch die SWD Signale anliegen) sind auch auf dem Board. Und zum Testen ist noch 1 LED am PC13 Signal und auf dem Board ist bereits der Uhrenquarz vorgesehen, aber nicht bestückt.

Leider war es dem Verkäufer nicht möglich mir einen Schaltplan zu besorgen. Ich habe zwar ein Archiv mit etlichen Schaltplänen und Codebeispielen von dem Verkäufer bekommen, aber der richtige Schaltplan war nicht dabei. Schade, vielleicht werde ich das Board mal reengineerieren

Aus diesem Grund war der erste Anschluss etwas abenteuerlich. Ich habe 2 Stunden gebraucht, um

die erste Verbindung mit dem Board herzustellen. Aber jetzt klappt es.

Das erste BLink Programm war dann auch in 1 Stunde erledigt. Auch da fehlte mir die Informationen, wo jetzt diese sch LED angeschlossen ist. PC13 ist der richtige Pin.

Nachdem auch diese Hürde geschafft wurde, habe ich schnell noch mein USB→Seriell Adapter angeschlossen und konnte da bereits nach 2 Minuten Daten an den PC senden. Der 1. Usart ist auch bei diesem Board extra herausgeführt.

Bishar habe ich alles mit der ([Coocox IDE](#)) gemacht. Version 1.7.7. Diese IDE basiert auf dem Eclipse CDT Framework. Ich mag Eclipse eigentlich sehr, aber Coocox ist doch recht langsam und vor allem stürzt es öfters mal ab. Da die aktuelle Version 2.0 leider den STM32F103 noch nicht unterstützt, mußte ich auf die ältere Version 1.7.7. gehen. Sobald die 2.0 den STM32F103 unterstützt, werde ich die IDE nochmal probieren.

Deswegen werde ich mal das nächste Projekt mit der [em:Blocks IDE](#) versuchen. Diese basiert auf code:Blocks.

Aber auch da hat man mit Versionskonflikten zu kämpfen. Die Versionen für die Standardbibliotheken STDLib von STM und CMSIS (ARM) sind schon recht alt, und sollten direkt nach dem ERzeugen des Projekts gegen die aktuellen ausgetauscht werden.

From:

<https://wkla.no-ip.biz/ArduinoWiki/> - **Arduino im Modellbau**

Permanent link:

<https://wkla.no-ip.biz/ArduinoWiki/doku.php?id=stm32:stm32>

Last update: **2018/11/04 10:51**

