

DCC Zentrale

Die Digitalzentrale ist das Steuerungselement einer digitalen Modellbahn. Die Digitalzentrale bedient diverse Handregler und versorgt die Fahrzeuge mit den nötigen Informationen. Bei 2L= hat sich das DCC - Digital Command Control - durchgesetzt. Dabei gibt die Zentrale ein DCC Signal aus, eine Bitfolge als Datenstrom. Dieses Rechtecksignal wird von einem [Booster](#) als Gleissignal verstärkt und auf die Gleise gegeben. In dem Signal befinden sich also die Daten und gleichzeitig werden die Fahrzeuge darüber mit Spannung versorgt. Durch den Takt von fast 10kHz und einer Bitfolge auch im Ruhezustand (d.h. es werden gerade keine Daten übertragen) ergibt sich ein Effektivwert am Gleis, mit dem die Dekoder in den Fahrzeugen betrieben werden, aus denen die Motoren, die Beleuchtung ihre Spannung bezieht.

Neben den käuflichen Digital-Zentralen für die Modellbahn gibt es auch eine Reihe Selbstbauprojekte. Die meisten Selbstbauprojekte beruhen auf den 8 Bit Controllern von Microchip, entweder AVR oder PIC. Ein sehr umfangreiches Beispiel ist die [Z21PG](#) von Philipp Gahtow oder eine etwas einfachere Variante die [DCC-Ex](#). Nicht unerwähnt lassen möchte ich die Arbeit von W. Kufer mit dem [OpenDCC](#) Projekt, eine Quelle der Informationen mit vielen guten Beschreibungen, der Protokolle und der Hardware.

Die 8 Bit AVR Controller sind zwar leistungsfähig, aber gerade in Bezug auf den Speicher und der Netzwerkfähigkeit kommen diese schnell an ihre Grenzen. Die Z21PG und DCC-Ex sind beide Open Source, alle Quelltexte sind veröffentlicht. Auch ich habe erst mit einer Z21PG auf Basis eines ATmega2560 begonnen und wollte mir eigentlich auch für eine Selbstbau-Zentrale eine Leiterplatte machen. Mehr oder weniger per Zufall bin ich in einer meiner vielen Schubladen über den Antennenanalyzer nach EU1KY gestolpert, den ich fürs andere Hobby mal gebaut hatte. Der besteht im wesentlichen aus einem [ST Discovery Modul](#) mit einem STM32 ARM Controller.

So habe ich mich etwas näher mit diesen Controllern beschäftigt und auch die Entwicklungsumgebung dazu installiert, die aus der STM32CubeIDE, dem STM32CubeProgrammer und STM32CubeMX besteht. Alle Tools sind kostenlos auf der Website von STMicroelectronics zu bekommen. Ich habe mich dann für ein Nucleo-144 Board mit dem STM32F746ZG Controller entschieden. Mittlerweile gibt es einen Nachfolger davon, aber das schöne bei den STM32 Controllern ist die Skalierbarkeit, vom kleinen 8 Pin Controller bis zum F746 mit 204 Pins. Nach den ersten Schritten mit CubeIDE bin ich dann auf Visual Studio Code mit PlatformIO und STMduino gewechselt.

Stück für Stück setze ich den vorhandenen Arduino Code der Z21PG für einen ATmega auf die STM32 ARM Plattform um. Da die Boards gleich Ethernet haben, I/O Pins ohne Ende, tolle AD und DA Wandler, mehrere serielle Schnittstellen, I²C, SPI usw. usf. (einfach mal ins [Datenblatt](#) schauen), sind die für diese Anwendung wirklich gut geeignet. Dazu kommt noch der viele Speicher mit 32Bit Breite, die hohe Verarbeitungsgeschwindigkeit der ARM Controller und der extrem günstige Preis der Boards. Aktive Boards mit Ethernet wären diese hier und es ist im Prinzip egal, welches man wählt, die haben alle den gleichen Formfaktor und das gleiche Layout, um diese auf einer Trägerplatine mit Schnittstellen zu integrieren.

<input type="checkbox"/>	NUCLEO-F207ZG	ACTIVE	STM32F207ZG	Arduino Uno, ST Morpho, ST Zio, USB Micro-AB FS OTG	100Mbps (IEEE 1588)	ST-LINK V2-1, USB Micro-B
<input type="checkbox"/>	NUCLEO-F439ZI	ACTIVE	STM32F439ZI	Arduino Uno, ST Morpho, ST Zio, USB Micro-AB FS OTG	100Mbps (IEEE 1588)	ST-LINK V2-1, USB Micro-B
<input type="checkbox"/>	NUCLEO-F756ZG	ACTIVE	STM32F756ZG	Arduino Uno, ST Morpho, ST Zio, USB Micro-AB FS OTG	100Mbps (IEEE 1588)	ST-LINK V2-1, USB Mini-B
<input type="checkbox"/>	NUCLEO-F767ZI	ACTIVE	STM32F767ZI	Arduino Uno, ST Morpho, ST Zio, USB Micro-AB FS OTG	100Mbps (IEEE 1588)	ST-LINK V2-1, USB Micro-B
<input type="checkbox"/>	NUCLEO-H563ZI	ACTIVE	STM32H563ZI	Arduino Uno, ST Morpho, ST Zio, USB Type-C FS	100Mbps (IEEE 1588)	ST-LINK V3E, USB Type-C
<input type="checkbox"/>	NUCLEO-H723ZG	ACTIVE	STM32H723ZG	Arduino Uno, ST Morpho, ST Zio, USB Micro-AB FS OTG	100Mbps (IEEE 1588)	ST-LINK V3E, USB Micro-B
<input type="checkbox"/>	NUCLEO-H753ZI	ACTIVE	STM32H753ZI	Arduino Uno, ST Morpho, ST Zio, USB Micro-AB FS OTG	100Mbps (IEEE 1588)	ST-LINK V2-1, USB Micro-B
<input type="checkbox"/>	NUCLEO-H755ZI-Q	ACTIVE	STM32H755ZI	Arduino Uno, ST Morpho, ST Zio, USB Micro-AB FS OTG	100Mbps (IEEE 1588)	ST-LINK V2-1, USB Micro-B

Erste Versuche liefen bisher sehr erfolgversprechend. Hier ein erstes Foto des Nucleo-144 Boards mit OLED Display und EEPROM. Ich habe hier ein EEPROM für die Speicherung von Konfigurationsdaten vorgesehen, auch wenn man im Flash des Controllers speichern könnte. Nur ist der Flashspeicher nicht beliebig oft wieder beschreibbar und bei den Controllern liegt der Wert bei 10.000x im Gegensatz zum EEPROM mit > 100.000x. Das die Anzeige vom Display so seltsam aussieht, liegt an der akt. niedrigen Wiederholrate, das wird sich noch ändern. Das Board ist über Netzwerk zu erreichen und auch die Smartphone-App fürs iPhone für die Steuerung kann dieses bereits als DCC-Zentrale erkennen.

Demnächste mehr in diesem Theater.

From:

<https://isnix.de/> - **It's boring when it works!**

Permanent link:

<https://isnix.de/doku.php?id=modellbahn:zentrale>

Last update: **2024-03-16 18:28**

