

Software für Arduino: Ein universeller DCC-Zubehördecoder

Dies ist eine Software für einen DiY-DCC-Zubehördecoder auf Arduinobasis. Für den Aufbau sind nur Grundkenntnisse in der Arduino-Programmierung notwendig. Der eigentliche Sketch muss normalerweise nicht angepasst werden. Das Verhalten kann über Konstantendefinitionen in einer Parameterdatei eingestellt werden.

Bei entsprechender externer Beschaltung und mit einer geeigneten Zentrale können die CV-Werte sowohl geschrieben, als auch gelesen werden.

Die Hardware muss selbst erstellt werden, ein Bausatz existiert nicht. Beispiel-Schaltpläne sind im Anhang vorhanden.

Die folgende Zubehörtypen können (auch gemischt) eingerichtet werden. Pro Typ können 2 Portausgänge konfiguriert werden.

- Servos (opt. mit Relais zur Polarisierung von Weichen)
- Doppelspulenantriebe
- statische Ausgänge
- blinkende Ausgänge (opt. Wechselblinker)

Bis zu 8 (aufeinanderfolgende) Zubehöradressen sind ansteuerbar
die 1. Adresse ist per Programmierung einstellbar

Das Verhalten der Funktionalität wird über CV-Programmierung festgelegt:

- bei Servoausgängen die Endlagen und die Geschwindigkeit
- bei Doppelspulenantrieben die Einschaltzeit der Spulen.
- bei blinkenden Ausgängen das Blinkverhalten

Die parametrierbaren Eigenschaften des Decoders lassen sich über eine ‚Steuerdatei‘ (DCC_Zubehordecoder.h) einstellen, ohne den eigentlichen Sourcecode der .ino-Datei ändern zu müssen.

Parametrierbar sind:

- Portbelegung für Startverhalten
- Zahl und Typ der ansteuerbaren Zubehöradressen
- die Ports für die Zubehöranschlüsse
- initiale Grundeinstellungen für die jeweiligen Zubehörtypen (diese können per CV-Programmierung überschrieben werden.

Inbetriebnahme

Zur Inbetriebnahme kann der Decoder in verschiedene Betriebsmodi gebracht werden. Dies wird über Spannungspegel an 2 analogen Eingängen (Standard:A5/A4) beim Hochlauf bestimmt. Die im folgenden mit A5 bzw. A4 bezeichneten Eingänge lassen sich per Einstellung in der .h-Datei auch auf andere Analogeingänge verlegen.

A5:

- 5V (nur Pullup) normaler Betriebsmodus, kein PoM
- 3,3V (Spannungsteiler 1:2) PoM immer aktiv, Adresse immer aus defaults
- 1,6V (Spannungsteiler 2:1) IniMode: CV's werden immer auf init-Werte aus .h-Datei gesetzt
- 0V Programmiermodus / PoM (1. Empfangenes Telegramm bestimmt Adresse)

A4:

- Ist A4 beim Programmstart auf 0, werden alle CV's einschl. interner

Statuswerte auf die Defaults zurückgesetzt

Justierung der Servo-Ausgänge:

Die Endlagen der Servos sind in CV's gespeichert. Zur Justierung kann ein Drehencoder angeschlossen werden. Dies vereinfacht die Einstellung. Der Drehencoder wirkt immer auf den Servo und die Servolage, die als letztes über einen DCC-Befehl eingestellt wurde. Sobald der Servo über einen DCC-Befehl wieder verstellt wird, wird die aktuell per Encoder eingestellte Position in den CV's gespeichert.

Wird A4 während der Einstellung auf 0 gezogen, wird der aktuell vom Drehencoder beeinflusste Servo in die Mittellage gebracht. Sobald der Encoder wieder bewegt wird, bewegt sich das Servo wieder zur vorhergehenden Position.

Ist einem Servo ein Relais zugeordnet, so schaltet dieses in der Mitte des eingestellten Stellweges um.

CV-Werte

folgende CV's werden vom Zubehördecoder verwendet:

CV-Nr	Bedeutung
1/9	1. Adresse (Decoderadresse oder Weichenadresse)
29	Betriebsarteneinstellung nach DCC (Decodertyp, Adressierungsart: Decoder / Output-Adressierung)
47	Kennung für Erstinitiierung, allgemeine Optionen die für den gesamten Decoder gelten Bit7-4 = 0x50 andere Werte führen beim Hochlauf zur Grundinitiierung Bit0=1: Automatische Adresserkennung im Programmiermodus Bit1=1: Roco-Modus (dcc-Adresse 0 = Weichenadresse 1) Bit1=0: Std-Modus (dcc-Adresse 4 = Weichenadresse 1)
48/49	PoM-Adresse
50-54	Parameter für 1. Weichenadresse
55-59	Parameter für 2. Weichenadresse
...	
Bedeutung der CV's bei den verschiedenen Funktionen (CV-Nummern für 1. Weichenadresse)	
	Servo:
50	Bit0 = 1: AutoOff der Servoimpulse bei Stillstand des Servo
51	Position des Servo für Weichenstellung '0' (in Grad, 0...180)
52	Position des Servo für Weichenstellung '1' (in Grad, 0...180)
53	Geschwindigkeit des Servo
54	aktuelle Weichenstellung (nicht manuell verändern!)
	Doppelspulenantrieb: (derzeit nur mit automatischer Abschaltung)
50	Bit0 = 1: Spulenausgang automatisch abschalten = 0: Spulenausgang über DCC-Befehl abschalten

51	Einschaltdauer der Spule (in 10ms Einheiten)
52	minimale Ausschaltdauer der Spule (in 10ms Einheiten)
53	-
54	aktuelle Weichenstellung (nicht manuell verändern!)
	statischer/Blinkender Ausgang (derzeit nur statisch möglich)
50	Bit0 = 1: Blinken, 0: statisch Bit1 = 1: Beim Blinken starten erst beide Leds dann Wechselblinken Bit2 = 1: mit weichem Auf/Abblenden (Pins müssen PWM-fähig sein)
51	Einschaltzeit des Blinkens (10ms Einheiten)
52	Ausschaltzeit des Blinkens (10ms Einheiten)
53	Auf/Abblendezeit
54	aktueller Zusatnd (nicht manuell verändern!)

Schaltbild:

Beispiel einer Decoderschaltung für 4 Servos (3 mit Polarisierung), 1 Doppelspulenantrieb und 2 statischen Ausgängen.

Diese Schaltbild passt zum Beispielkonfigurationsfile im Sketchverzeichnis

