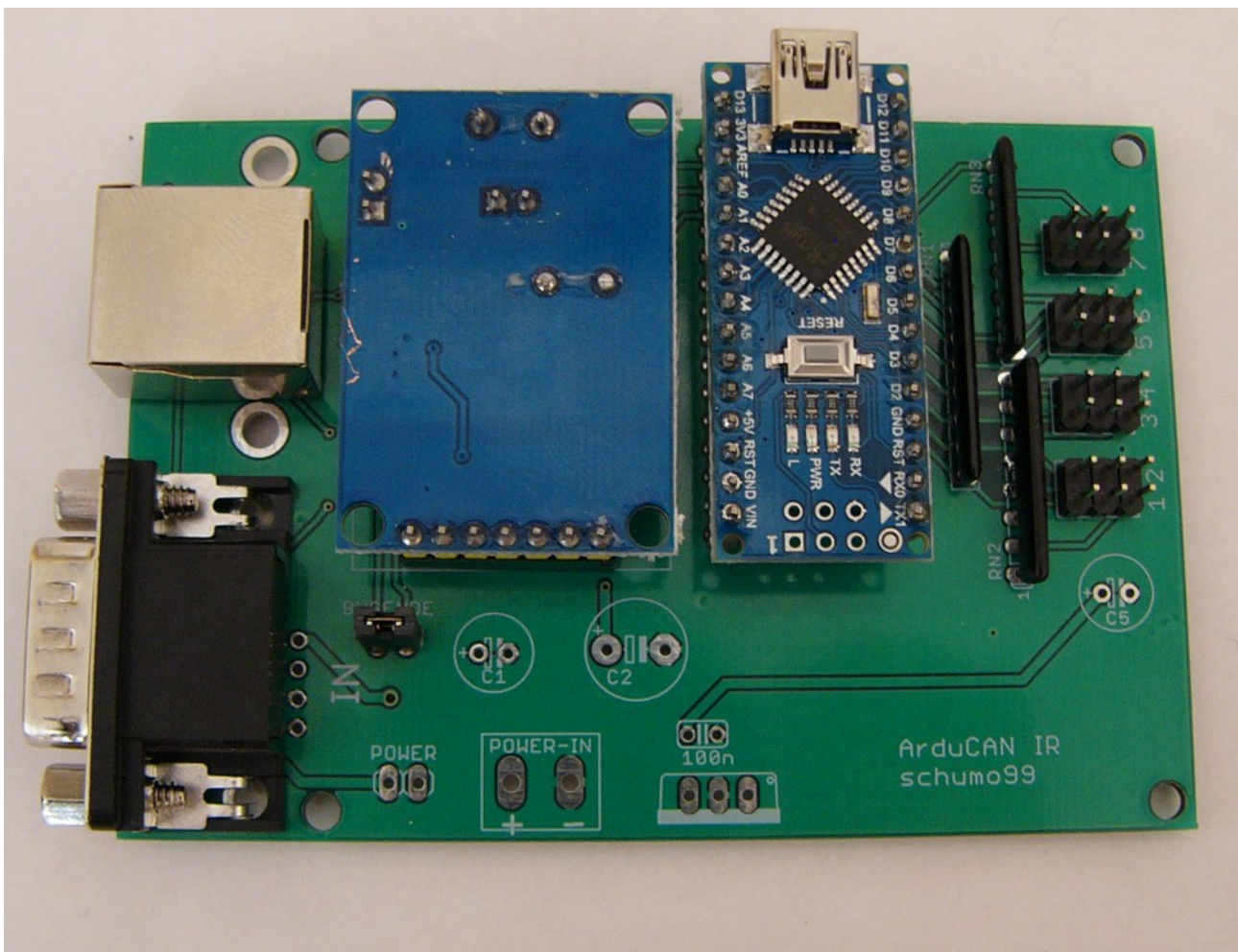


„ArduCAN-IR“ auf Arduinobasis by schumo99

Die vorliegende Decoder („ArduCAN-IR“) verfügt über 8 Fremdlicht unempfindliche Eingänge für Reflexlichtschranken. Trotz der enormen Umempfindlichkeit gegenüber Störlicht, ist die Empfindlichkeit doch so hoch, dass auch weiter entfernte Objekte ohne spezielle Kennzeichnung (wie z.B. helle Punkte) erkannt werden. Dadurch kann das gesamte Rollmaterial ohne Anpassungen sicher detektiert werden. Als Lichtschranken können der TCRT5000 oder eine Kombination aus SFH309FA und SFH4346 verwendet werden. Diese ist aus meiner Sicht leicht im Gleisbett zu verstecken. Andere Lichtschranken sind jedoch ebenfalls möglich.

Der Decoder funktioniert auf Basis eines Arduino nano und ist sehr leicht aufzubauen. Es sind keine SMD-Lötkenntnisse erforderlich. Die Gesamtstromaufnahme inkl. 8 Sensoren beträgt ca. 65mA.



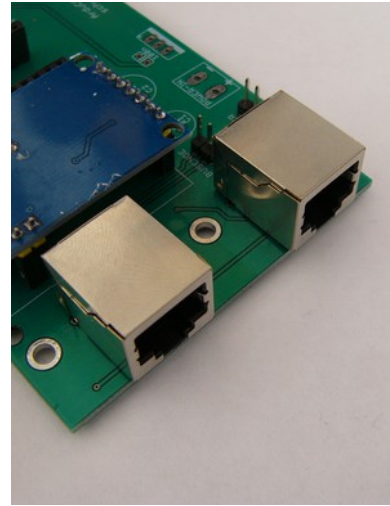
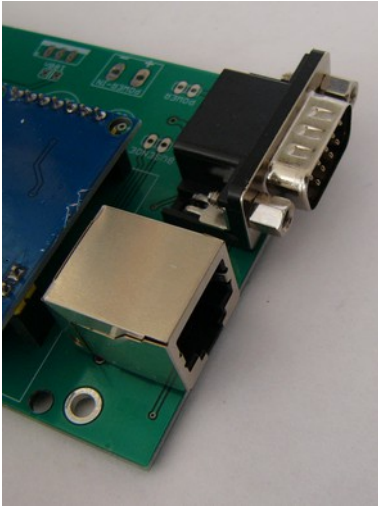
Viel Freude mit dem Nachbau!

Der Verfasser übernimmt keine Verantwortung für Funktionsfähigkeit oder etwaige Schäden, die durch die Nutzung der Schaltung entstehen. Der Aufbau erfolgt auf eigene Verantwortung.

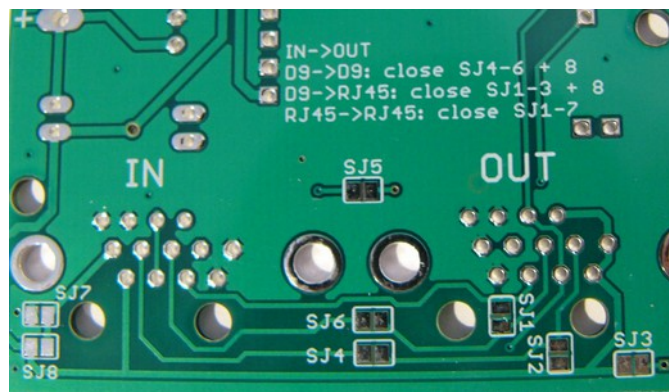
Anschlüsse

CAN-Bus

Die Platine bietet die Möglichkeit über D-Sub Stecker oder über RJ45-Stecker (Standard CAT-Netzwerk-kabel) angeschlossen zu werden. Es ist auch ein Mischbetrieb möglich.



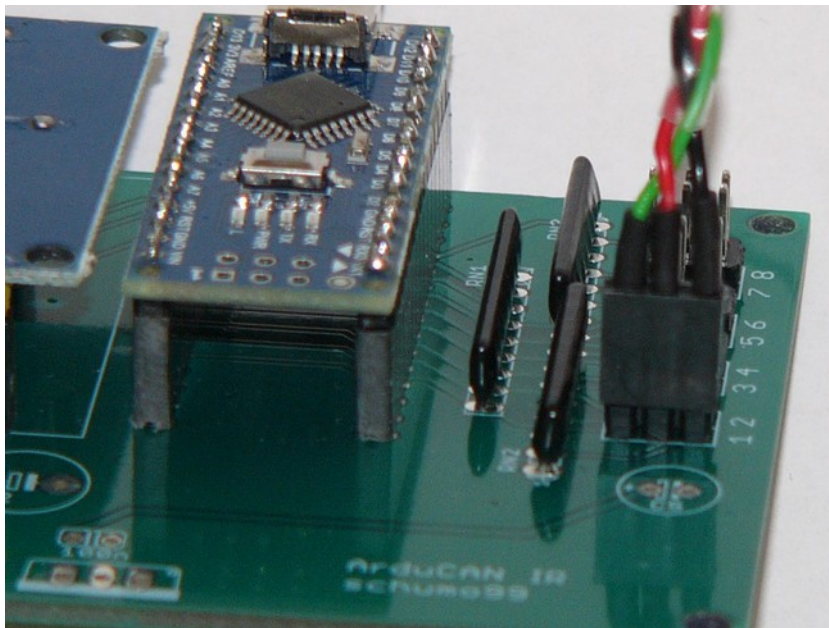
Ja nach gewünschter Konfiguration müssen auf der Unterseite der Platine die entsprechenden Lötjumper geschlossen werden. Welche Jumper bei welcher Kombination zu schließen sind, ist praktischerweise direkt auf der Platine aufgedruckt.



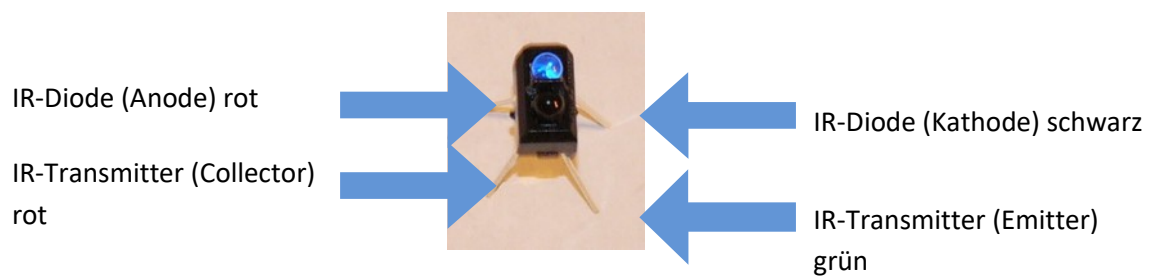
Eingänge

Der ArduCAN IR verfügt über 8 Eingänge für Reflexlichtschranken. Diese sind wie folgt beschaltet:

Anschlusspin	Empfohlene farbige Kennzeichnung	Elektronische Bedeutung
Pin auf der Boardaussenseite	schwarz	Minus-Pol der Sendediode
Mittlerer Pin	rot	Gemeinsamer + Pol
Pin auf der Boardinnenseite (dem Arduino zugewandt)	grün	Minus-Pol der Empfangsdiode



Für TCRT5000-Sensoren bedeutet dies folgende Verdrahtung:



Achtung: Die Beschaltung unterscheidet sich vom ReflexIR für s88!!

Stromversorgung

Die Stromversorgung erfolgt standardmäßig aus dem CAN-Bus. Der Jumper „Power“ bleibt standardmäßig offen. Dadurch wird die ganze Platine mit dem Strom des CAN-Bus versorgt.

Folgende Möglichkeiten stehen zur Verfügung:

Power-Jumper	Power-IN	Bedeutung/Verwendung
egal	Nichts angeschlossen	Standardbetriebsart Der „ArduCAN IR“ und alle Anschlüsse werden über den CAN-Bus mit Strom versorgt.
Geschlossen	CAN-Busspannung , direkt von der CAN-Bus- Schnittstelle entnommen	Um Leitungsverluste des CAN-Bus auszugleichen, kann die Busspannung erneut eingespeist werden. Achtung: Keine andere Stromversorgung verwenden
Offen	Netzteil	Der „ArduCAN IR“ und alle Anschlüsse werden über das angeschlossene Netzteil mit Strom versorgt. (siehe auch *)
Offen + Lötjumper „+5VIN“ auf der Unterseite geschlossen	Netzteil (5V)	Nur mit I/O-Steckerleiste: Die gesamte Stromversorgung des ArduCAN und aller angeschlossenen Komponenten erfolgt durch das externe Netzteil.

* Der „ArduCAN IR“ arbeitet intern mit 5V, welche auch den Lichtschranken zur Verfügung gestellt werden. Wenn die CAN-Busspannung deutlich über 12V liegt, ist der interne Spannungswandler überlastet. Man merkt das daran, dass der Arduino sehr warm wird, oder es im Extremfall sogar kurze Aussetzer gibt. Dann muss die interne Spannung von 5V mittels eines DC/DC-Wandlers erzeugt werden. Dies ist auf der Platine bereits vorgesehen. Als Spannungswandler kann z.B. ein LM7805 verwendet werden. C1 und C2 (z.B. 100µF) und der 100nF Kondensator müssen dann auch bestückt werden. Alternativ kann auch ein komplettes step-down Reglermodul , welches pincompatibel zum 7805 sein muss, benutzt werden. Die Kondensatoren sind dann nicht nötig, schaden aber auch nicht.

Inbetriebnahme

Der „ArduCAN IR“ wird an den CAN-Bus angeschlossen; entweder über D-Sub Stecker oder Standard-Netzwerkkabel, je nachdem welche Buchsen auf der Platine verbaut wurden. Der mit „Busende“ gekennzeichnete Jumper dient dazu den CAN-Bus zu terminieren. Dieser ist am Ende eines CAN-Bus Stranges zu setzen. Die o.g. Details zur Stromversorgung bitte ebenfalls beachten.

Nachdem der „ArduCAN IR“ mit Strom versorgt ist, blinken die LEDs „RX“ und „TX“ des Arduinos für circa fünf Sekunden recht schnell. In dieser Zeit kann durch Drücken und Gedrückthalten des Arduinotasters ein Reset aller Einstellungen des Bausteines ausgelöst werden (Bitte erst Drücken, wenn die LEDs schon 4-5x geblinkt haben). Wenn der Reset abgeschlossen ist, leuchten die beiden LEDs „RX“ und „TX“ dauerhaft. Nun kann der Taster wieder losgelassen werden und der Baustein geht normal in Betrieb. Dies ist an der dauerhaft leuchtenden LED „TX“ zu erkennen. Wird während des Betriebes der Taster des Arduinos kurz gedrückt, so wechselt der Baustein in den Modus zum Einlernen der Knoten-Nummer. Dies wird durch ein Blinken der LED „TX“ signalisiert und „rocrail“ zeigt den entsprechenden Dialog am Bildschirm an. Nach erfolgtem Absenden der neuen Knoten-Nummer durch die Steuerungssoftware (z.B. rocrail) leuchtet die LED „TX“ wieder dauerhaft.

Signal-LED	Bedeutung
„RX“ und „TX“ blinken schnell für circa 5 Sekunden	Initialisierung des Prozessors. Resetbefehl durch Drücken und Gedrückthalten des Tasters möglich. Taster erst drücken, wenn „RX“ und „TX“ schon 4-5x geblinkt haben
„TX“ leuchtet dauerhaft	Normaler Betriebsmodus
„TX“ blinkt	Lerne Knotennummer
„RX“ und „TX“ leuchten dauerhaft	Reset beendet → Taster loslassen
„RX“ und „TX“ blinken langsam	Prozessorfehler! Entweder Software nicht richtig eingespielt oder Probleme mit der Spannungsversorgung.

Aufbau

Beim Design der Platine wurde auf einfachen Aufbau geachtet. Daher sind keine SMD-Bauteile einzulöten. Stattdessen müssen hauptsächlich vorgefertigte Baugruppen über Buchsenleisten miteinander verbunden werden.

Anzahl	Bauteil	Verwendung
1x	Arduino Nano	Mikroprozessor
1x	CAN-Bus Modul MCP2515 TJA105	CAN-Transceiver
2x	Buchsenleiste 15-polig	Zum Aufstecken des Arduino nano
1x	Buchsenleiste 7-polig	Zum Aufstecken des CAN-Transceivers
2x	Buchsenleiste 2-polig	Zum Aufstecken des CAN-Transceivers
2x	D-Sub Stecker/Buchse oder RJ45-Buchse	Zum Anschluss an den CAN-Bus
1x	RN1: Widerstandsnetzwerk 22k z.B. SIL9-8 22k	Pull-down Widerstand für alle Eingänge (Achtung auf Einbaulage achten. Der markierte Pin ist gemeinsame Masse)
2x	RN2+RN3: Widerstandsnetzwerk 150 Ohm z.B. SIL8-4 150	Vorwiderstand für IR-Dioden
1x	Steckerleiste 2-Pin	Terminierung des CAN-Busses
8x	Steckerleiste 2x3 Pin	Anschlüsse für IR-Sensoren
8x	TCRT5000 oder SFH309FA+SFH4346 oder	Reflexlichtschranken

Bauteile einlöten und zum Schluss den programmierten Arduino und den CAN-Transceiver aufstecken.

Der Arduino muss vorher mit der Software „ArduCAN_IR.hex“ programmiert werden. Die Fuses sind so zu setzen, dass der Reset-Pin als I/O-Pin aktiviert wird. Achtung: Werden die Fuses falsch gesetzt, ist es möglich, dass der Arduino irreparabel beschädigt ist! Außerdem empfiehlt es sich einen Bootloader zu benutzen, da sonst keine weitere Programmierung des Arduinos (z.B. Updates) mehr möglich ist. Wer es sich nicht zutraut, biete ich gerne an seinen/ihren Arduino zu programmieren oder einen bereits entsprechend programmierten Baustein zur Verfügung zu stellen.

Der Verfasser übernimmt keine Verantwortung für Funktionsfähigkeit oder etwaige Schäden, die durch die Nutzung der Schaltung entstehen. Der Aufbau erfolgt auf eigene Verantwortung.