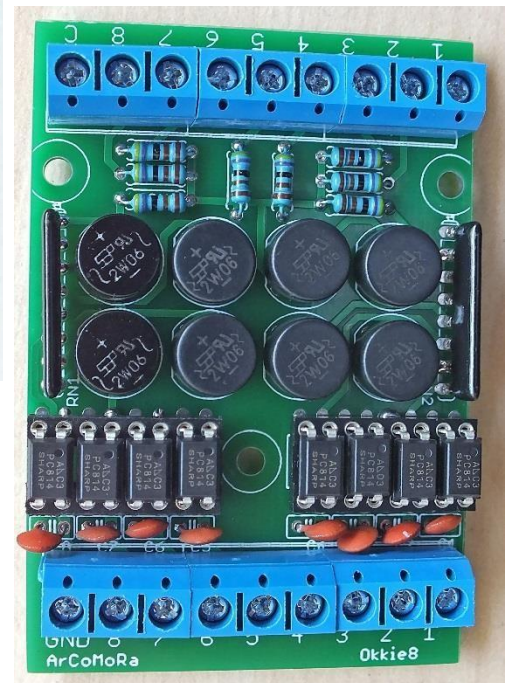
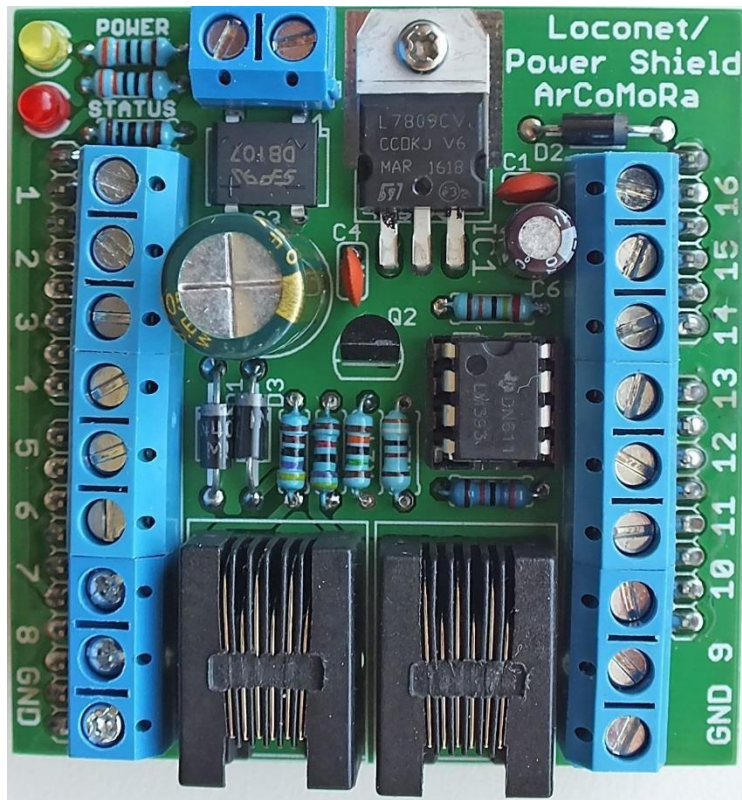


ARduinoLOCOnet

Der universelle Besetzmelder mit Arduino und LocoNet®



Author: Nico Teering

Juni 2020

Arloco Version 3.1 Dokument Version: 3.1DE

Info: info@Arcomora.com

Web: www.Arcomora.com/Arloco

LocoNet® is a registered trademark of DigiTrax, Inc



[LADEN SIE HIER DIE ARLOCO SOFTWARE HERUNTER](#)

Einleitung

Die Verwendung eines Arduino Mikrocontrollers im Modelleisenbahnbau ist vielen gut bekannt. Allerdings ist die eigentliche Anwendung für die meisten zu kompliziert. Die Programmierung oder einfache Anpassungen eines Programms werden immer noch als großes Problem angesehen. Infolgedessen bleibt der Arduino die Domäne einer sehr begrenzten Gruppe mit etwas mehr technischem Wissen. Das ist schade, denn mit dem Arduino können Sie Ihre gesamte Modelleisenbahn günstig automatisieren. Um den Arduino für jedermann zugänglich zu machen, habe ich daher eine Reihe von drei Programmen für Arduino erstellt. Diese reichen für alle Automatisierungen auf der Modelleisenbahn aus. Zusammen bilden sie das Arcomora-Konzept.

Arcomora steht für Arduino Controlled Model Railway. Bei einer digitalen Modelleisenbahn nach dem Arcomora-Konzept wird die Automatisierung vollständig mit Arduino-Mikrocomputern gesteuert. Es ist nicht notwendig, einen Arduino selbst zu programmieren. Als Protokolle kommen DCC und LocoNet® zum Einsatz. Das Arcomora-Konzept besteht eigentlich aus drei verschiedenen Arduino-Programmen mit jeweils eigener spezifischer Aufgabe:

- **Mardec**, ein multifunktionaler Arduino DCC-Decoder.
Damit können Sie bis zu 12 Servos und viele weitere Zubehörteile wie LEDs und Relais steuern. Der Mardec hat maximal 16 Ausgänge, aber auch 'Eingänge'. Damit können Sie das angeschlossene Zubehör auch ohne DCC-Signal steuern. Optional mit einstellbarer Verzögerungszeit.
- **ArSigDec**, Arduino Signal Decoder.
Der Arsigdec kann Signale mit bis zu 8 LEDs (65 LEDs beim Mega2560) und 10 Signalbegriffen pro Signal steuern. An einer DCC-Adresse können Sie zwei unterschiedliche Signalbegriffe darstellen. Arsigdec enthält auch eine Datenbank häufig verwendeter Signale. Sie können auch eigene Signalbilder zusammenstellen.
- **ArLoco**, Arduino Rückmelder mit LocoNet®
Ein 16-Kanal-Rückmelder (58 auf Mega2560) basierend auf LocoNet. Optimiert sowohl für Stromerkennung als auch Impulserkennung (z. B. Reedschalter oder Infrarotlichtschranke). Dafür wird ein separates LocoNet-Shield verwendet.

Ein wesentliches Merkmal dieser drei Komponenten ist, dass sie einfach zu konfigurieren sind, ohne dass Arduino-Kenntnisse erforderlich sind. Die Konfiguration erfolgt mit einem einfachen Frage-Antwort-Spiel auf Ihrem PC. Also keine Notwendigkeit für eine Zentrale oder Ärger mit komplizierten CVs. Diese Konfiguration wird meist fälschlicherweise als Programmierung bezeichnet.

Um Verwirrung zu vermeiden, verwenden wir den Begriff *Programmierung*, für das Schreiben von Arduino-Code und *Konfigurieren*, für das Einstellen der gewünschten Funktionen der angeschlossenen Komponenten.

Alle Arcomora-Komponenten haben daher einen Konfigurationsmodus und einen Betriebsmodus. Im Konfigurationsmodus kann man alles nach Belieben einstellen und im Betriebsmodus ist der Arduino betriebsbereit und man kann genau verfolgen, welche Aktionen am PC ausgeführt werden.

Die Installation der Software ist sehr benutzerfreundlich. Die gesamte benötigte Software wird auf einmal installiert. Auch das Hochladen der Programme auf den Arduino erfolgt fast vollständig automatisch. Mit ArCoMoRa ist es nicht notwendig, auch nur einen Buchstaben Arduino-Code zu schreiben.

Warum ein Arduino?

Der Hauptgrund ist der Preis. Ein (Klon-)Arduino kostet ohne USB-Kabel maximal 3,50 € (Stand 2020). Zusätzlich wird für den Mardec und Arsigdec ein Shield (Montageplatine) benötigt.

Dieser kostet 4,25 €. Man kann es aber auch ganz einfach selber machen.

Damit erhalten Sie einen 16-Kanal-DCC-Decoder oder einen Signaldecoder für 2 bis 8 Signale für weniger als 7 €. Für ArLoco ist ein LocoNet-Shield nötig; das kostet 4,25 €. Optional können Sie zwei 8-Kanal-Stromfühler-Platinen (Okkie, für je 5,50 €) daran anschließen.

Merkmale von ArLoco:

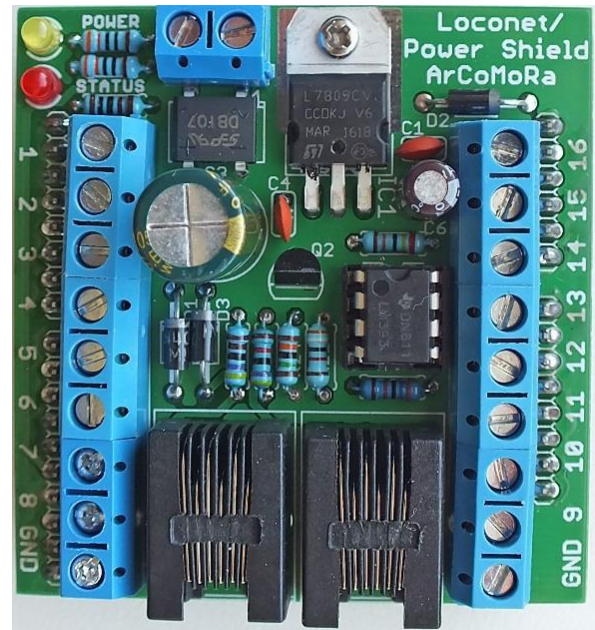
- Passend für Arduino MEGA2560 und Arduino UNO
- 16 Kanal (UNO) oder 58 Kanal (MEGA2560) Besetztmelder
- 256 Adressen möglich
- Adresse pro Kanal einstellbar.
- Verwendet LocoNet® für die Kommunikation mit Zentrale oder Computer.
- Umfangreiche Debugging-Möglichkeiten.
- Optimierte für Impulserkennung (Reed-Schalter, Infrarot etc.) und kontinuierliche Erkennung (Stromfühler)
- Einfache Konfiguration mit Ihrem PC.
- Kein Ärger mit CVs.
- Stromversorgung des Arduino über externes Netzteil oder LocoNet möglich.
- Geeignet für LocoNet-T und LocoNet-B
- Konfigurierbar pro Kanal für Impuls- oder Dauerdetektion. (auf Kanal 1 nur Impulserkennung)
- Kontinuierliche Erkennung:
 - Vorbeugung gegen Fehlalarme durch Störspitzen
 - Vorkehrung gegen kurze Stromunterbrechungen
 - Geeignet für Stromerkennung mit dem OKKIE-Stromerkennungsprint
- Impulserkennung:
 - Liefert konstanten Besetztimpuls von 1 Sek.
 - Eliminiert Kontaktprellen.
 - Vorkehrung gegen Doppelimpuls bei langsamer Fahrt mit Reed-Schaltern
 - Geeignet für Reed-Schalter, Infrarot-Erkennung etc.

Was wird benötigt?

- Ein Arduino UNO oder MEGA2560 Mikrocontroller
- Ein ArLoco Shield
- Eine LocoNet-Zentrale mit USB- oder Netzwerkanschluss oder eine LocoNet-Zentrale ohne USB oder Netzwerk und einen LocoNet-USB-Hub
- PC mit Modellbahnsteuerungsprogramm
- Optional: Stromfühler-Hardware wie z.B. OKKIE (zur Stromerkennung)

ArLoco Shield

- 2 LocoNet-Anschlüsse. Dadurch ist es möglich, mehrere ArLocos in Reihe zu schalten.
- Mit Stromversorgung für Arduino (AC/DC)
- 16 Schraubklemmen für die Eingänge
- 2 Schraubklemmen für GND-Anschluss
- Power-LED
- Status-LED



Okkie

- Platine zur Stromerfassung für 8 Gleisabschnitte
- 8 Schraubklemmen für die Eingänge
- 8 Schraubklemmen für die Ausgänge
- GND-Anschluss
- Gemeinsamer Gleisanschluss



Ein Bausatz für das ArLoco Shield und das Okkie kann unter www.arcomora.com/order bestellt werden.

Erkennungsmethoden

Das ArLoco besitzt zwei Arten der Besetztterkennung:

- Kontinuierliche Erkennung
- Pulserkennung

Kontinuierliche Erkennung

Bei kontinuierlicher Erkennung sendet ArLoco bei fallender Flanke einen LocoNet-Impuls mit Adressinformationen an die Zentrale. Der Abschnitt ist jetzt „belegt“.

Bei steigender Flanke sendet ArLoco erneut einen LocoNet-Impuls. Der Abschnitt ist jetzt „frei“.

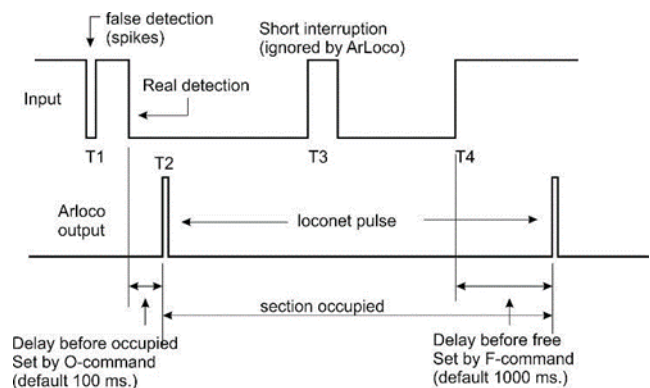
Daher sollte diese Methode zur Stromerkennung in Kombination mit einer Stromfühlerschaltung verwendet werden. Der Ausgang dieser Schaltung ist mit dem Eingang des ArLoco-Schildes verbunden.

Um falscher Detektion durch kurze Impulse zu verhindern, hat ArLoco zwei Timer eingebaut:

Wartezeit vor Besetzmeldung:

(delay before occupied)

Ein Abschnitt wird als belegt gemeldet, wenn der Eingang länger auf Low ist, als vom Timer „Wartezeit vor Besetzmeldung“ vorgegeben. Standardmäßig beträgt diese Zeit 100ms. Dies kann jedoch mittels „O“-Kommando geändert werden. Das verhindert falsche Meldungen durch kurze Spitzenimpulse. Der Impuls an T1 (siehe Bild) wird nicht erkannt, da die Dauer kürzer ist, als die Zeit von „Wartezeit vor Besetzmeldung“. Ein LocoNet-Puls (T2) wird gesendet nachdem die Wartezeit vor Besetzmeldung abgelaufen ist, da hier der Eingang länger auf „Low“ ist.



Wartezeit vor Freimeldung:

(delay before free)

Der Abschnitt wird freigegeben, wenn der Eingang länger, als durch Wartezeit vor Freimeldung vorgegeben, wieder auf „High“ ist. (Standardeinstellung 1000ms, änderbar mittels „F“-Befehl) Der High-Impuls T3 wird von ArLoco ignoriert, da er kürzer ist, als die Wartezeit vor Freimeldung.

Zum Zeitpunkt T4 plus der Zeit von Wartezeit vor Freimeldung wird der Abschnitt mittels LocoNet Puls freigegeben.

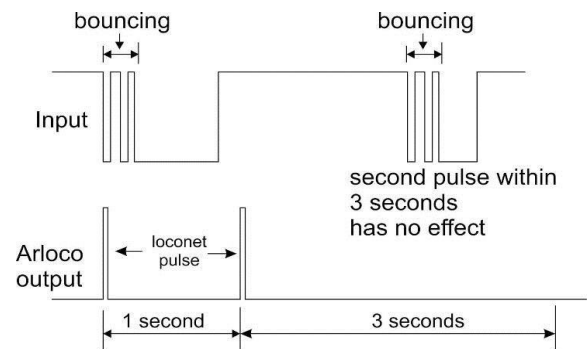
Auf einem idealen Gleis sollten die Zeiten beider Timer Null sein.

Beachte: Auf Sensor 1 ist keine kontinuierliche Erkennung, sondern nur Pulserkennung möglich!

Pulserkennung

Diese Methode ist für die Verwendung mit Reed-Kontakten gedacht. Wenn der Eingang Low wird, sendet ArLoco einen LocoNet „besetzt“ Impuls an die Zentrale.

Nach dem fixen Intervall von 1 Sekunde sendet ArLoco die LocoNet „frei“ Information. Das verhindert falsche Meldungen durch Kontaktprellen.



Das Tempo der Züge hat ebenfalls keinen Einfluss, auch wenn der Zug auf einem Reed-Kontakt anhält.

Da nur geringe Ströme involviert sind, besteht keine Gefahr von Spitzen. Daher wurden auch keine Vorkehrungen dagegen getroffen.

Nach dem zweiten LocoNet Puls wartet ArLoco drei Sekunden, bevor der Eingang wieder abgefragt wird. Das verhindert Doppelmeldungen bei Langsamfahrten. Diese Wartezeit kann mittels „W“-Kommando eingestellt werden.

Gemischte Erkennung

Standardmäßig sind alle Eingänge (außer Eingang 1) auf kontinuierliche Erkennung konfiguriert. Mittels des „S“-Befehls kann der Modus für jeden einzelnen Eingang (außer Eingang 1) individuell gesetzt werden.

Eingang 1 ist **immer** auf Pulserkennung eingestellt.

Adressierung

Der ArLoco hat 16 oder 58 Sensoreingänge. Jeder Sensor hat seine eigene spezifische Adresse. Der gesamte Adressbereich von ArLoco geht von Adresse 1 bis 256.

Eingänge des ArLoco können mittels „A“-Befehl deaktiviert werden. Dadurch verwenden diese keine Adressen.

Dies ist besonders nützlich für modulare Anlagen, bei denen jedes Modul seinen eigenen ArLoco haben muss, aber die Anzahl der pro Modul zu erkennenden Abschnitte begrenzt ist. Dadurch können mehr als 16 ArLocos verwendet werden.

Jedem ArLoco muss eine eigene Basisadresse („B“-Befehl) zugewiesen werden. Wenn Sie also z.B. nur 10 Adressen ab Adresse 25 verwenden möchten, stellen Sie die Basisadresse auf 25 und die Sensoren 11 bis 16 auf 'inaktiv'. Der erste aktive Sensor erhält die Basisadresse, der Zweite die Basisadresse +1, der Dritte Basisadresse+2 usw.

Dank dieser Methode ist ArLoco sehr flexibel im Umgang mit dem verfügbaren Adressraum.

Es ist ebenfalls möglich, jedem Sensor eine eigene Adresse zu geben. Beachte aber bitte, dass diese Adresse dann in keinem anderen ArLoco verwendet wird. Auch muss diese Adresse außerhalb des „eigenen“ Adressbereichs liegen.

Rechts ist eine Tabelle, die angibt, welcher Arduino Pin für welchen Sensor verwendet wird.

Beim UNO werden die Pins 1, 8, 12 & 13 nicht verwendet. Die Sensornummern sind am Shield aufgeführt.

Beim MEGA werden die Pins 1, 8, 12, 13 & 48 nicht verwendet.

Die Sensornummern 1 bis 16 sind am Shield aufgeführt. Die Pin-Nummern für die Sensoren 17 bis 58 sind am MEGA angegeben.

Achtung: Beginnend mit Pin Nummer 48 verschieben sich die Pin- und Sensornummern!

Wenn einem Sensor keine eigene Adresse zugewiesen wurde, gilt Folgendes:

Adresse = Sensornummer + Basisadresse – 1

Sensor	UNO pin	MEGA pin
1	0	0
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	9	9
9	10	10
10	11	11
11	14	54
12	15	55
13	16	56
14	17	57
15	18	58
16	19	59
17		17
18		18
19		19
20		20
21		21
22		22
23		23
24		24
25		25
26		26
27		27
28		28
29		29
30		30
31		31
32		32
33		33
34		34
35		35
36		36
37		37
38		38
39		39
40		40
41		41
42		42
43		43
44		44
45		45
46		46
47		47
48		49
49		50
50		51
51		52
52		53
53		A10
54		A11
55		A12
56		A13
57		A14
58		A15

Konfiguration des ArLoco

Nachdem die Software korrekt installiert wurde, kann die Konfiguration des ArLoco beginnen. Klicken sie auf die Verknüpfung „ARLOCO konfigurieren“ („Configure ARLOCO“)

Beim allerersten Start von ArLoco werden die Treiber mit jenem USB-Port verbunden, an dem der Arduino angeschlossen ist.

Beachte Bitte:

- Jede numerische Eingabe muss mit <enter> abgeschlossen werden.
- Ausschließlich Kommandos aus nur **einem** Buchstaben benötigen kein <enter>.
- Bei den meisten Eingaben wird der aktuelle Wert durch Eingabe von <enter> alleine nicht verändert. Der aktuelle Wert wird in Klammern angezeigt.
- Im Konfigurationsmodus leuchtet die gelbe LED am Shield (oder die eingebaute LED an Pin 13) permanent.
- Im Betriebsmodus ist diese LED aus, Sie blinkt, wenn ein LocoNet-Signal erkannt wird.
- Schalte NUMLOCK an der numerischen Tastatur ein.
- Für Befehle könne Groß- oder Kleinbuchstaben verwendet werden.
- Benutze **NUR** die Backspace-Taste, um numerische Eingaben zu korrigieren.

Alle Bildschirmfotos basieren auf dem Arduino UNO. Am MEGA2560 funktionieren sie jedoch ident.

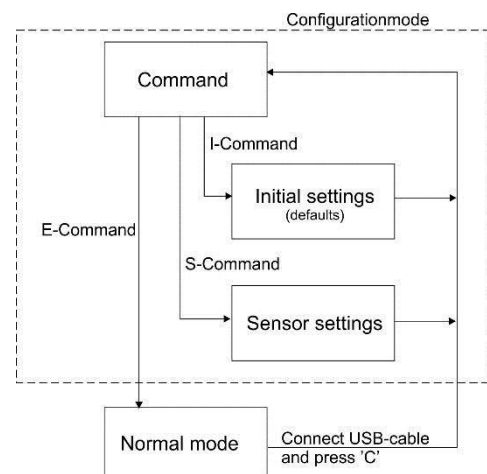
Du kannst nun ein Kommando eingeben.

Beim allerersten Start des ARLOCO oder nach einem kompletten Reset wird automatisch zuerst der I-Befehl aktiviert. (Siehe bei „I“-Befehl). Es werden verschiedene Default-Werte gesetzt. Bei jedem Start der Konfiguration wird automatisch eine Übersicht der Konfiguration angezeigt.

Die Kommandos

Wie der Mardec und der Arsigdec hat der ArLoco zwei Zustände - den Konfigurationsmodus und den Betriebsmodus. Der Konfigurationsmodus umfasst allgemeine Befehle, Sensorbefehle und Anfangseinstellungen. Durch Eingabe eines „?“ (Fragezeichen) erhält man einen Überblick über die verfügbaren Befehle.

```
Specify action (B/D/E/I/R/S/V/?): ?  
B = Set Base Address  
S = Set Sensor settings  
I = Set Initial(default) settings  
D = Set Debug mode  
V = View all sensors  
R = Reset ArLoco  
E = Exit to normal mode  
Specify action (B/D/E/I/R/S/V/?):
```



Jeder Befehl besteht aus einem einzelnen Buchstaben, der ohne <enter> eingegeben werden kann. Der Buchstabe stammt von der englischen Beschreibung des Befehls ab.

ACHTUNG: ArLoco 2.x kann **nicht** auf Version 3.0 upgedatet werden.

Die Art der Datenspeicherung im EEPROM hat sich radikal geändert. Das war nötig, damit jeder Sensor eine eigne Adresse bekommen kann.

I-Kommando (Initiale Einstellungen)

Das "I"-Kommando konfiguriert alle Standardeinstellungen. Es gibt keine Untermenüs, alle Einstellungen werden nacheinander abgefragt.

Das betrifft die folgenden Einstellungen.

- Eingabe der administrativen Nummer des ArLoco
- Standardtyp der Detektion (Puls oder kontinuierlich)
Alle Eingänge werden dadurch auf diese Methode eingestellt (Ausnahme Sensor 1, der immer Puls ist)
Auch mittels „M“-Befehl je Sensor.
- Die Basisadresse.
Auch mittels „B“-Befehl setzbar.
- Standard-Invertierung der Sensoren.
Standardmäßig wird ein „low“ (0 Volt) Signal als

```
Specify action (B/D/E/I/R/S/V/?): i
Specify number for this ARLOCO
Enter value from 1 to 100 (1): 1
This is your ARLOCO #1

Specify default detection method; 1=Continuous, 2=Pulse
Enter value from 1 to 2: 1
Detection method set to Continuous

Set Base address for this ArLoco
Enter value from 1 to 250 (1): 32
Base address set to 32

Default Inversion set to NOT Inverted
Change inversion ? Y/N (N=default) :

Specify Waittime for pulse detection (msec)
Enter value from 1 to 3000 (1001):
Waittime for pulse detection 1001 milliseconds

Specify delay before setting free (msec)
Enter value from 0 to 3000 (1000):
Delay before setting free 1000 milliseconds

Specify delay before setting occupied (msec)
Enter value from 0 to 1000 (100):
Delay before setting occupied 100 milliseconds

Specify action (B/D/E/I/R/S/V/?): I
```

„besetzt“ interpretiert. Wenn jedoch eine Lichtschranke, bei der der Zug den Lichtstrahl unterbricht, verwendet wird, wird für besetzt ein „high“ (5 Volt) Signal ausgegeben. Durch die Invertierung interpretiert ArLoco das „high“ Signal als besetzt. Das kann auch per Sensor durch den „I“-Befehl eingestellt werden.

- Zeit bei Pulsdetektion, in der nach Besetztmeldung der Eingang nicht abgefragt wird. (Wartezeit, siehe auch den Abschnitt über Pulserkennung)
- Wartezeit vor Freimeldung. Standard 1000 ms, kann zwischen 1 und 3000 ms gesetzt werden.
- Wartezeit vor Besetztmeldung, Standard 100 ms, kann zwischen 1 und 3000 ms gesetzt werden.

B-Kommando (Basisadresse)

Mit dem „B“-Befehl setzt man die Basisadresse des ArLoco.

Siehe auch den Abschnitt über Adressierung.

```
Specify action (A/B/M/E/N/O/S/F/D/W/V/R/?): B
Set Base address for this ArLoco
Enter value from 1 to 255 (1): 23
Base address set to 23
```

R-Kommando (Reset)

Der "R"-Befehl setzt alle Einstellungen des ArLoco zurück.

Danach startet das Programm neu.

```
Specify action (A/B/M/E/N/O/S/F/D/W/V/R/?): R
Are you sure you want to clear all settings? Y/N: █
```

E-Kommando (Exit)

Mit dem „E“-Befehl wechseln Sie in den Betriebsmodus. ArLoco meldet nun jeden erkannten Zug wie konfiguriert

zurück. Um zum Konfigurationsstatus zurückzukehren, schließen Sie das USB-Kabel wieder an und drücken Sie die Taste „C“.

```
Specify action (A/B/M/E/N/O/S/F/D/W/V/R/?): E
```

```
Do you want to start normal mode? Y/N (N=default): █
```

V-Kommando (View)

Der „V“-Befehl zeigt die Einstellungen aller Sensoren.

```
Specify action (B/D/E/I/R/S/V/?): v
```

```
Default detection method set to Continuous
```

```
Delay before setting occupied (msec): 150
```

```
Delay before setting free (msec): 800
```

```
Waittime for pulse detection (msec): 500
```

```
Base address: 25
```

```
Sensor settings:
```

```
Sensor 1 set to Pulse and Active, on address 25, with input not inverted
```

```
Sensor 2 set to Continuous and Active, on address 26, with input not inverted
```

```
Sensor 3 set to Continuous and Active, on address 27, with input not inverted
```

```
Sensor 4 set to Continuous and Active, on address 28, with input not inverted
```

```
Sensor 5 set to Continuous and Active, on address 29, with input not inverted
```

```
Sensor 6 set to Continuous and Active, on address 30, with input not inverted
```

D-Kommando (Debug)

Mit dem "D"-Befehl wird der Debugmodus ein- und ausgeschaltet. Im Debugmodus wird im Betriebsmodus jede Sensoränderung gemeldet.

```
ARLOCO, the ARduino LocoNet feedback encoder
USB version 1.0

Normal mode of ARLOCO #4

To configuration: connect USB and press 'C'

Default detection method set to Continuous

Delay before setting occupied (msec): 100
Delay before setting free (msec): 500

This Arloco uses addresses 49-64

Set address 59 from sensor 11 as occupied
Set address 60 from sensor 12 as occupied
Train detected on address 53 from sensor 5
Set address 58 from sensor 10 as free
Train detected on address 54 from sensor 6
Set address 61 from sensor 13 as free
Set address 59 from sensor 11 as free
Train detected on address 55 from sensor 7
Set address 62 from sensor 14 as occupied
Set address 61 from sensor 13 as occupied
Set address 60 from sensor 12 as free
Set address 61 from sensor 13 as free
Train detected on address 53 from sensor 5
Train detected on address 54 from sensor 6
Set address 63 from sensor 15 as occupied
Train detected on address 56 from sensor 8
Set address 62 from sensor 14 as free
Train detected on address 54 from sensor 6
Set address 58 from sensor 10 as occupied
Train detected on address 53 from sensor 5
Set address 63 from sensor 15 as free
Train detected on address 54 from sensor 6
```

Normaler Betriebsmodus

ACHTUNG:

Wenn ArLoco im Betriebsmodus mit dem PC verbunden ist und Putty gestartet wurde, kann sich auch Ihre Tastatur mit dem Arduino verbinden. Dies geschieht über Pin 0 (Tx). Ein unbeabsichtigter Tastendruck kann daher eine Besetztmeldung über Sensor 1 hervorrufen. Stellen Sie sicher, dass Putty dann nicht das aktive Fenster ist.

```
Specify action (A/M/E/N/O/S/F/D/?): D
Do you want to show all sensor changes? (Y/N, N=default) y
Debug mode set to ON
Specify action (A/M/E/N/O/S/F/D/?):
```

```
Specify action (A/M/E/N/O/S/F/D/?): d
Do you want to show all sensor changes? (Y/N, N=default) y
Debug mode set to ON

Specify action (A/M/E/N/O/S/F/D/?): e
Do you want to start normal mode? Y/N (N=default): y
ARLOCO, the ARduino LocoNet feedback encoder
USB version 1.0

Normal mode of ARLOCO #4

To configuration: connect USB and press 'C'

Default detection method set to Continuous

Delay before setting occupied (msec): 100
Delay before setting free (msec): 500

This Arloco uses addresses 49-64

365691 No change on sensor 1, still free
365735 End of wait on sensor 5
365769 End of wait on sensor 6
365802 End of wait on sensor 8
365836 Sensor 10 changed to 0
365868 Sensor 14 changed to 1
365939 Set address 58 from sensor 10 as occupied
366371 Set address 62 from sensor 14 as free
366616 Sensor 13 changed to 0
366660 Train detected on address 55 from sensor 7
366717 No change on sensor 13, still free
366732 Sensor 13 changed to 0
366814 Train detected on address 54 from sensor 6
366833 No change on sensor 13, still free
366851 Sensor 13 changed to 0
366947 Train detected on address 53 from sensor 5
366952 No change on sensor 13, still free
366978 Sensor 13 changed to 0
367081 Set address 61 from sensor 13 as occupied
367661 End of pulse on sensor 7
367733 Sensor 13 changed to 1
367815 End of pulse on sensor 6
367890 Sensor 10 changed to 1
367948 End of pulse on sensor 5
368234 No change on sensor 13, still occupied
368393 Set address 58 from sensor 10 as free
368424 Sensor 11 changed to 0
368527 Set address 59 from sensor 11 as occupied
369044 Sensor 13 changed to 1
369547 Set address 61 from sensor 13 as free
369852 Sensor 12 changed to 0
369955 Set address 60 from sensor 12 as occupied
370117 Sensor 11 changed to 1
```

Debugmodus: DieZahl gibt die Millisekunden seit Start des ArLoco-Sketches an.

Es kann auch kein OKKIE an Sensor 1 angeschlossen werden.

S-Kommando (Sensor)

Mit dem „S“-Befehl können Sensoreinstellungen für einzelne Sensoren geändert werden.
Gib einfach die Nummer des Sensors an, der geändert werden soll.

```
Specify action (B/D/E/I/R/S/V/?): S
```

```
Which sensor do you want to set?
```

```
Enter value from 1 to 58: 12
```

```
Specify action for Sensor 12 (L/A/I/M/V/?): █
```

```
Specify action for Sensor 12 (L/A/I/M/V/?): ?
```

```
L = Set Loconet address
```

```
M = Set detection Method
```

```
A = Set Active on/off
```

```
I = Set Inversion on/off
```

```
V = View settings
```

```
Specify action for Sensor 12 (L/A/I/M/V/?): █
```

Die folgenden Unterbefehle sind verfügbar:

L-Kommando (LocoNet Adresse)

Mit dem „L“-Befehl können Sie einem Sensor eine eindeutige Adresse von 1 bis 256 zuweisen. Stellen Sie sicher, dass diese Adresse auf keinem anderen ArLoco verwendet wird. Sie muss auch außerhalb des 'eigenen' Adressbereichs liegen. Wenn ein Sensor bereits eine andere Adresse hatte, kann mit Adresse 0 die Adresse auf den Wert entsprechend der Basisadresse zurückgesetzt werden.

```
Specify action for Sensor 12 (L/A/I/M/V/?): 1
```

```
Set LocoNet address for sensor 12
```

```
Enter value from 0 to 256 (36): 200
```

```
LocoNet address for sensor 12 set to 200
```

```
Specify action for Sensor 12 (L/A/I/M/V/?): 1
```

```
Set LocoNet address for sensor (0=back to base) 12
```

```
Enter value from 0 to 256 (200): 0
```

```
LocoNet address for sensor 12 set to 36
```

M-Kommando (Methode)

Mit dem „M“-Befehl kann die Erkennungs-methode für den ausgewählten Sensor eingestellt werden.

```
Specify action (A/M/E/N/O/S/F/D/?): m
```

```
Specify default detection method; 1=Continuous, 2=Pulse
```

```
Enter value from 1 to 2: 1
```

```
Detection method set to Continuous
```

```
Specify action (A/M/E/N/O/S/F/D/?): █
```

V-Kommando (View)

Der „V“-Befehl zeigt die Einstellungen des ausgewählten Sensors an.

```
Specify action for Sensor 12 (L/A/I/M/V/?): V
```

```
Sensor 12 set to Continuous and Active, on address 36, with input not inverted
```

A-Kommando (Active)

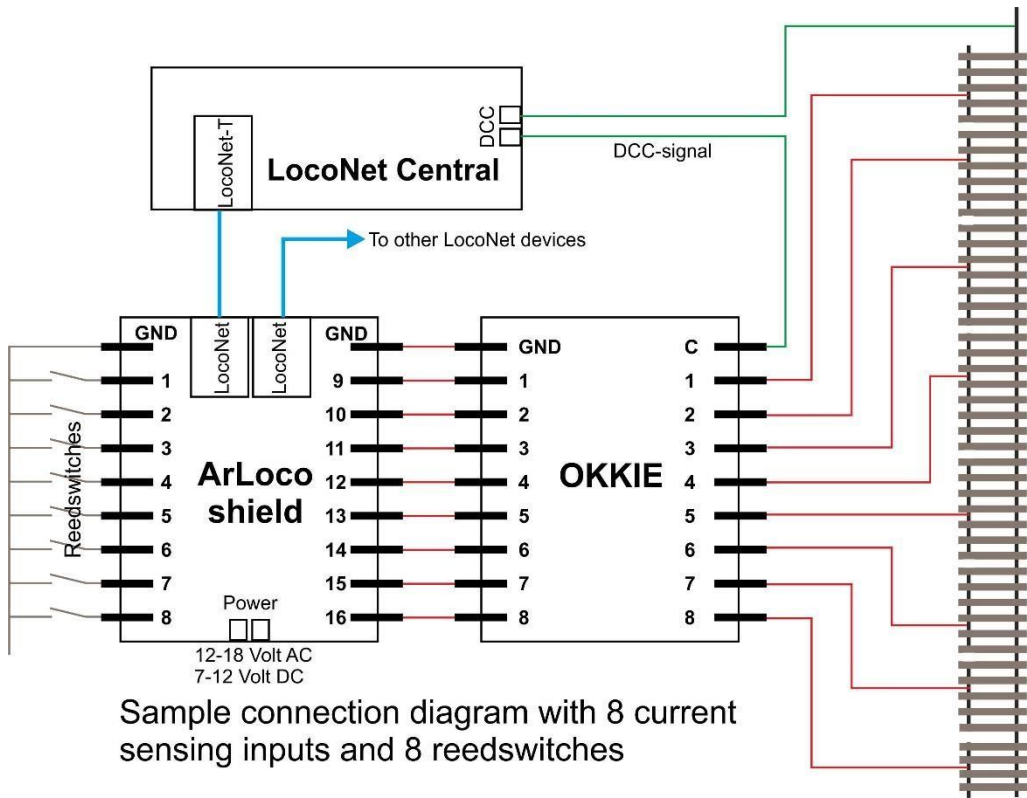
Mit dem A-Befehl kann ein Sensor aktiviert oder deaktiviert werden. Ein inaktiver Sensor wird nicht „getestet“. Indem Sie eine Reihe von Sensoren deaktivieren, werden mehr Adressen für andere ArLocos verfügbar.

Wenn ein Sensor auf inaktiv gesetzt ist, werden Sie gefragt, ob die höheren Sensornummern ebenfalls auf inaktiv gesetzt werden sollen. Dies ist nützlich, wenn Sie einen Teil der Sender nicht verwenden möchten. Wenn ein Sensor wieder aktiv gesetzt wird, erhält er die Adresse, die er zuvor hatte. Außerdem werden Sie gefragt, ob alle anderen nicht aktiven Sensoren wieder auf aktiv gesetzt werden sollen.

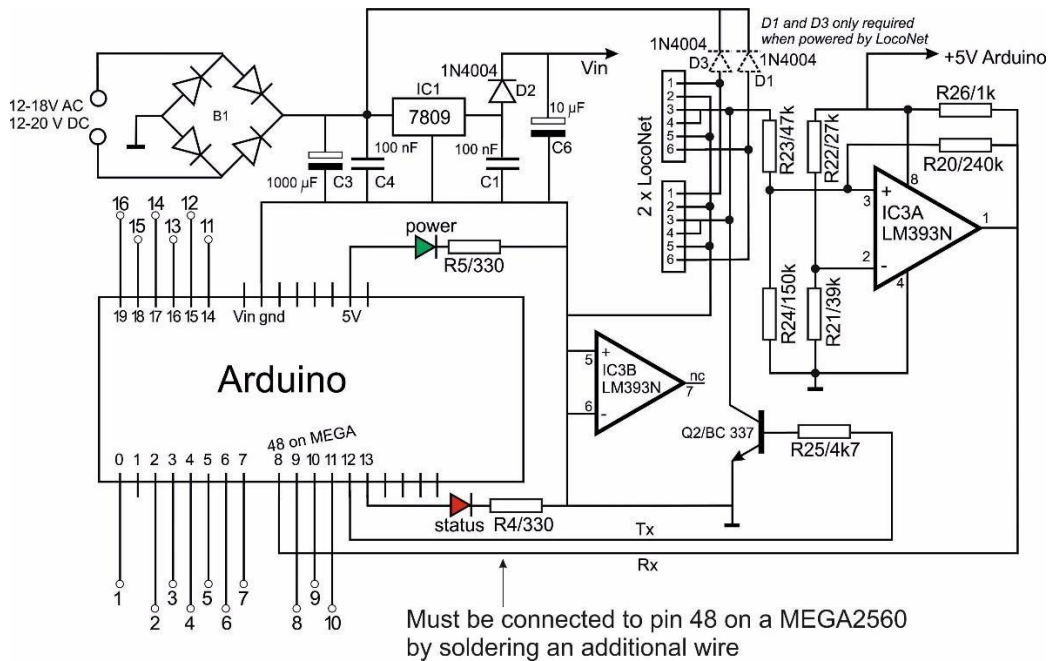
```
Which sensor do you want to set?  
Enter value from 1 to 58: 50  
  
Specify action for Sensor 50 (L/A/I/M/V/?) : A  
  
Sensor 50 set to not active  
Set sensors 51 to sensor 58 also InActive? █
```

```
Which sensor do you want to set?  
Enter value from 1 to 58: 15  
  
Specify action for Sensor 15 (L/A/I/M/V/?) : A  
  
Sensor 15 set to Active  
Set all other inactive sensors active again? (Y/N) █
```

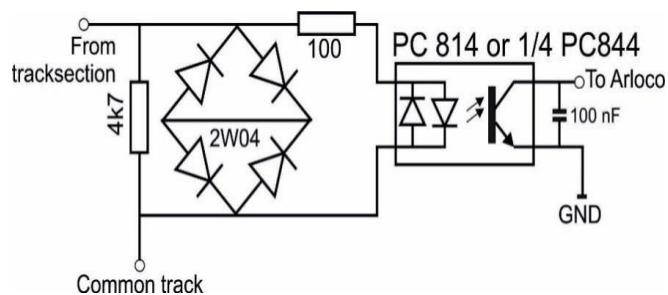
Verbindungsdiagramm



Schema ArLoco



Schema OKKIE



Schema für einen Gleisabschnitt Die Okkie Platine enthält 8 dieser Schaltungen.

Aufbauanleitung ArLoco Shield

Stückliste:

Widerstand	R4	1k	
Widerstand	R5	1k	
Widerstand	R20	240k	
Widerstand	R21	39k	
Widerstand	R22	27k	
Widerstand	R23	47k	
Widerstand	R24	150k	
Widerstand	R25	4k7	
Widerstand	R26	1k	
Kondensator	C1	100 nF	
Kondensator	C4	100 nF	
Kondensator	C6	10 uF	
Kondensator	C3	1000 uF	
Brücken- gleichrichter	B1	DB107	
Spannungs- regler	IC1	L7809CV	
OpAmp	IC3	LM393N	
Diode *	D1	1N400x	
Diode	D2	1N400x	
Diode *	D3	1N400x	
Transistor	Q2	BC337	

Schrauben und Muttern M2,5

6 x 3 Pins Schraubkonnektor

1 x 2 Pins Schraubkonnektor

2 x LocoNet RJ12 Buchse

40 Pins Stiftleiste

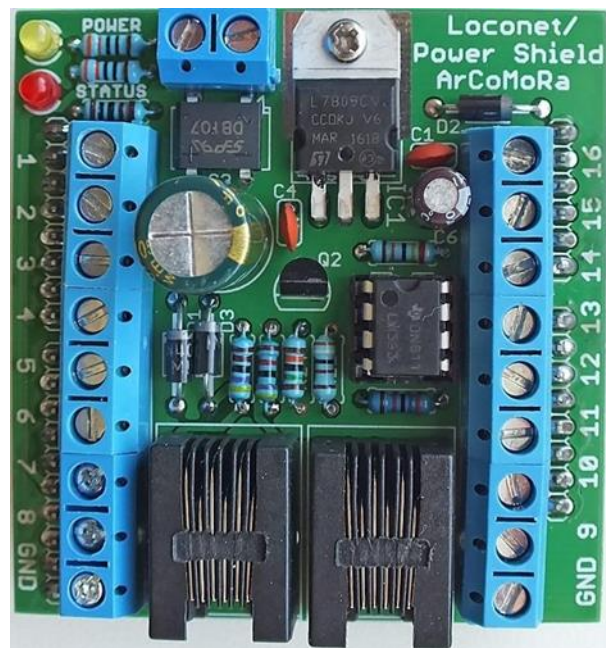
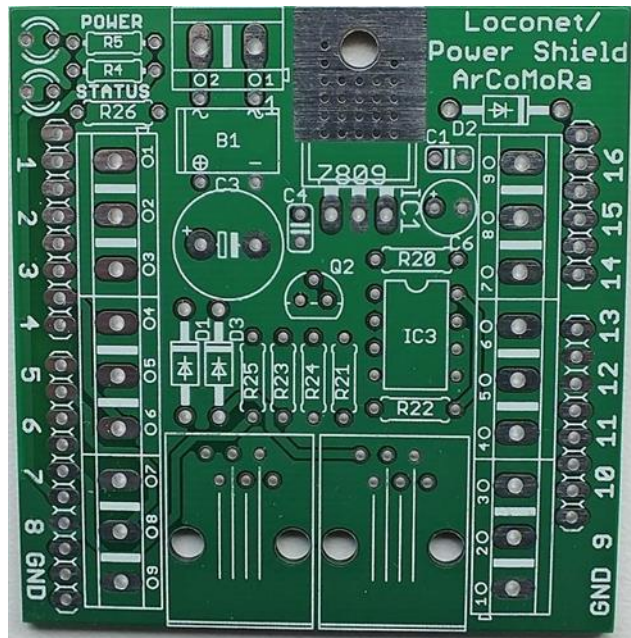
1 rote LED und 1 gelbe LED

8 Pins IC-Sockel

Der Arduino kann auf eine der folgenden Arten mit Strom versorgt werden:

- Über ein externes Netzteil
- Stromversorgung über LocoNet

* Nur erforderlich bei Stromversorgung über LocoNet, es kann auch der Klinkenstecker verwendet werden. Die Komponenten C1, C3, C4, C6, B1, D1, D3 und IC1 sind dann nicht erforderlich



Empfohlene Aufbaureihenfolge des ArLoco Shields:

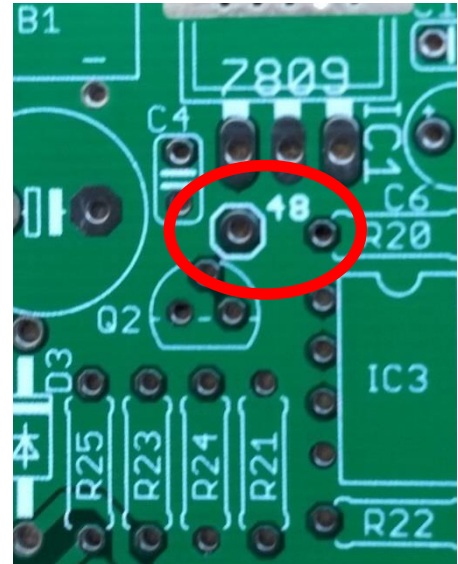
1. Alle Widerstände. **Achte auf den Farbcode!**
2. Diode D2. **Achten Sie auf die richtige Ausrichtung!** (siehe Foto)
3. Dioden D1 und D3 wenn der Arduino via LocoNet mit Strom versorgt werden soll
4. Brückengleichrichter DB107. **Achten Sie auf die richtige Ausrichtung!**
5. Montiere den IC Socket mit der Aussparung zu Widerstand R20
6. Kondensatoren C1 und C4 (100nF)
7. IC1 (7809) einfügen. Biegen Sie zuerst die drei Anschlüsse mit einer geeigneten Zange in einem Winkel von 90 Grad. Stecken Sie dann die drei Anschlüsse so in die Platine, dass das Loch mit dem Loch in der Platine übereinstimmt. Löten Sie die drei Anschlüsse und befestigen Sie den IC mit der M2,5-Schraube und Mutter
8. Löten Sie die rote und grün/gelbe LED ein. **Achten Sie auf die richtige Ausrichtung!** Das kurze Bein (Kathode) ist auf der Außenseite der Platine!
9. Transistor Q2, (BC337 oder 2N3904) **Achten Sie auf die richtige Ausrichtung!**
10. Stecken Sie IC3 (LM393) in die Fassung. **Achten Sie auf die richtige Ausrichtung!** Der Punkt an Pin 1 befindet sich auf der Seite von R20.
11. Schneiden Sie mit einem Seitenschneider aus dem Stiftleistenstreifen 4 Stück Stiftleisten mit Längen von 6, 8, 8 und 10 Stiften. Stecken Sie die Teile auf den Arduino und legen Sie die Platine auf die Stifte. Löten Sie nun die beiden äußeren Stifte jedes Stücks. Achten Sie darauf, dass die Stecker fest an der Platine anliegen. Um eine unnötige Wärmebelastung des Arduino zu verhindern, entfernen sie nun die Platine vom Arduino. Prüfen Sie nun, ob der Streifen fest an der Platine anliegt und korrigieren Sie ihn gegebenenfalls. Löten Sie dann alle anderen Stifte.
12. Je 3 Schraubverbinder mit 3 Anschlüssen zusammenschieben. Jeder Block hat auf der einen Seite einen Schlitz und auf der anderen Seite eine erhöhte Kante. Machen Sie dasselbe mit den drei anderen.
13. Stecken Sie die beiden 9er-Stecker und den Stromanschluss auf die Platine und drehen Sie sie um. Da auf der Platine (noch) keine höheren Bauteile als die Schraubverbinder sind, liegt die Platine schön flach auf den Schraubverbindern auf. Sie können den Stromanschluss weglassen, wenn Sie den Arduino über LocoNet mit Strom versorgen möchten.
14. Verlöten Sie die Schraubverbinder.
Tipp: Löten Sie zuerst die äußersten Pins der beiden Blöcke. Sie sind dann fixiert und können nicht wieder herausfallen. Überprüfen Sie, ob sie gerade sind. Korrigieren Sie ggf. und löten Sie dann die restlichen Pins.
15. Montiere und verlöte die beiden RJ12 Buchsen.
16. Kondensator C6 (10uF) einsetzen und verlöten. **Achten Sie auf die richtige Ausrichtung!** Plus (+) des Kondensators im Loch neben dem + auf dem Druck. Minus (-) ist auf dem Kondensator markiert.

17. Nur bei Verwendung des MEGA2560

Löten Sie einen Dupont-Stift in das Loch in der Nähe des mit 48 markierten Transistors. Verwenden Sie ein Dupont-Kabel (M-V), um diesen Stift mit Stift 48 des MEGA zu verbinden. **Ohne diese Verbindung funktioniert ArLoco nicht!**

18. Kondensator C3 (1000 uF) einsetzen und verlöten.

Achten Sie auf die richtige Ausrichtung! Plus (+) des Kondensators im Loch neben dem + auf dem Druck. Minus (-) ist auf dem Kondensator markiert.



Testen

Legen Sie nun den Schild auf den Arduino.

Schließen Sie zum Testen der externen Stromversorgung ein AC- oder DC-Netzteil an den Netzanschluss an.

Die Power-LED sollte jetzt leuchten. Tun Sie dies ohne LocoNet oder USB-Kabel!

Um die LocoNet-Stromversorgung zu testen, verbinden Sie ein LocoNet-Kabel mit einer LocoNet-Zentrale. Die Power-LED sollte jetzt leuchten.

Tun Sie dies ohne externe Stromversorgung oder USB-Kabel!

Schließen Sie den Arduino (mit geladenem ArLoco-Sketch) mit einem USB-Kabel an Ihren PC an und starten Sie „Configure ArLoco“. Konfigurieren Sie die ArLoco und wechseln Sie dann in den Betriebsmodus.

Testen Sie nun jeden Sensor, indem Sie den Eingang mit einem Draht an GND anschließen.

```
Set address 21 from sensor 2 as occupied
Set address 21 from sensor 2 as free
Set address 22 from sensor 3 as occupied
Set address 22 from sensor 3 as free
Set address 31 from sensor 12 as occupied
Set address 31 from sensor 12 as free
Set address 53 from sensor 34 as occupied
Set address 53 from sensor 34 as free
```


Aufbauanleitung Okkie

Stückliste:

8 x resistor 4k7



8 x Brückengleichrichter 2W10



8 x Kondensator 100 nF



2 x Optokuppler PC844 (OK1 bis OK8)
oder 2 x ILQ620 oder 2 x LTV844

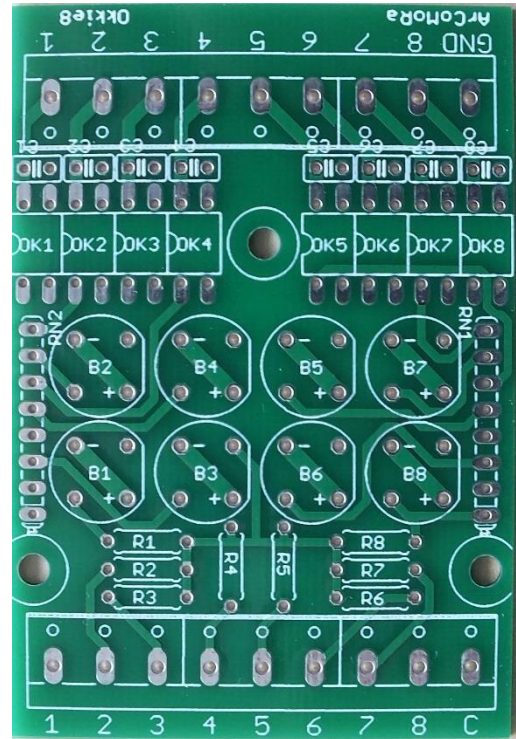


2 x 16 Pin IC Sockel.

2 x Widerstandsarray 100 Ohm

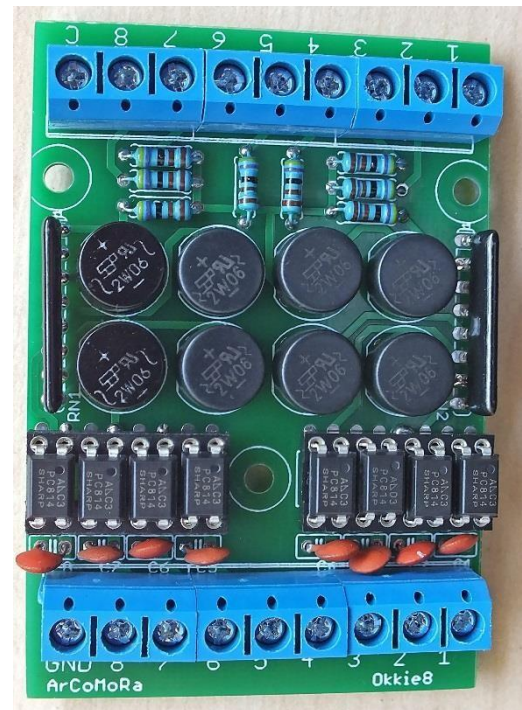


6 x 3-Pin Schraubklemmen.



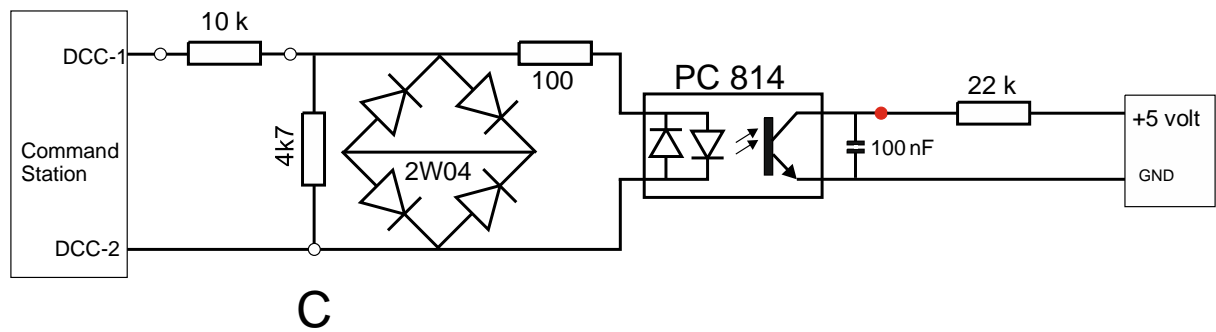
Empfohlene Aufbaureihenfolge der Okkie Platine

1. Löten Sie die Widerstände R1 bis R8 ein.
2. Löten Sie die Kondensatoren C1 bis C8 ein.
3. Verlöten sie beide IC-Sockel
4. Brückengleichrichter einstecken und verlöten. (B1 bis B8) **Achten Sie auf die richtige Ausrichtung.** Das längste Bein ist das Plus.
5. Verlöte beide Widerstandsarrays
6. Je 3 Schraubverbinder mit 3 Anschlüssen zusammenschieben. Jeder Block hat auf der einen Seite einen Schlitz und auf der anderen Seite eine erhöhte Kante. Machen Sie dasselbe mit den anderen drei.
7. Stecken Sie die beiden 9er-Stecker und den Stromanschluss auf die Platine und drehen Sie sie um. Da auf der Platine (noch) keine höheren Bauteile als die Schraubverbinder sind, liegt die Platine schön flach auf den Schraubverbindern auf. Sie können den Stromanschluss weglassen, wenn Sie den Arduino über LocoNet mit Strom versorgen möchten.
8. Verlöten Sie die Schraubverbinder.
Tipp: Löten Sie zuerst die äußersten Pins der beiden Blöcke. Sie sind dann fixiert und können nicht wieder herausfallen. Überprüfen Sie, ob sie gerade sind. Korrigieren Sie ggf. und löten Sie dann die restlichen Pins.
9. Stecken Sie die 8 (bzw. 2) Optokoppler OK1 bis OK8 in die beiden IC-Fassungen. Achten Sie auf die richtige Ausrichtung. (Siehe Foto) Der PC844 hat die gleiche Ausrichtung wie der PC814. Pin 1 (am Punkt auf IC) befindet sich also auf der Seite der Brückengleichrichter.
10. FERTIG!



[ARLOCO SOFTWARE HIER DOWNLOADEN](#)

Testen des OKKIE



Sie können eine Okkie-Sektion wie folgt testen.

- Verbinden Sie den 'C'-Anschluss mit einem der beiden DCC-Ausgänge des Bedienfelds.
- Verbinden Sie einen Okkie-Eingang über einen 10k-Widerstand mit dem anderen DCC-Ausgang. Dies simuliert eine Lokomotive.
- Verbinden Sie den entsprechenden Okkie-Ausgang über einen 22k-Widerstand mit 5 Volt. Dies simuliert den Pull-up-Widerstand eines Arduino.
- Verbinden Sie den GND-Anschluss mit dem GND des Netzteils.

Messen Sie nun die Spannung am roten Punkt.

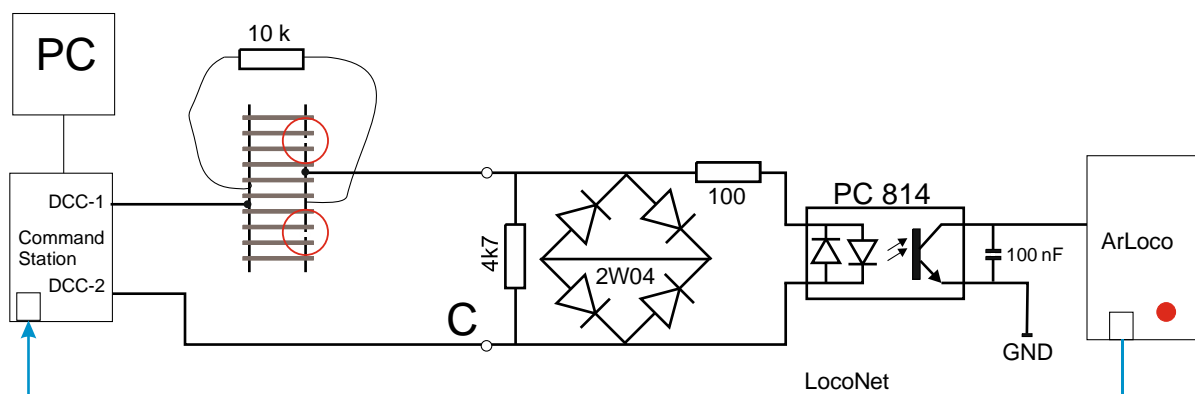
Sie sollten jetzt weniger als 0,5 Volt messen (Lok vorhanden).

Entfernen Sie nun den 10k-Widerstand.

Jetzt müssen Sie 5 Volt messen.

Wiederholen Sie dies für jeden Abschnitt des Okkie.

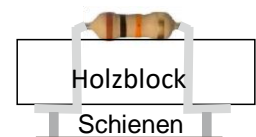
Testen der gesamten Kette: Okkie->ArLoco->PC



Indem der 10k-Widerstand auf den Schienen gehalten wird:

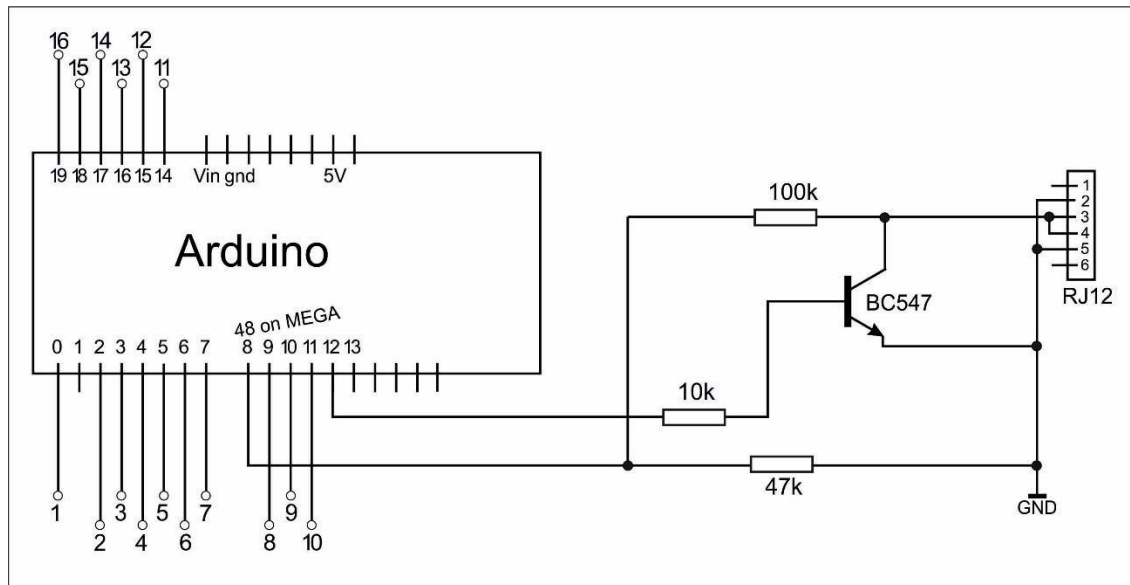
- Der Optokoppler leitet und der Eingang der ArLoco wird niedrig
- Die rote Status-LED an der ArLoco gibt einen Impuls
- ArLoco sendet eine LocoNet-Belegungsmeldung an die Zentrale. - Die Zentrale sendet es an den PC.
- Der betreffende Gleisabschnitt wird in der Zugsteuerungssoftware als belegt angezeigt.

Testblock: Bohren Sie zwei Löcher in einen Holzblock, legen Sie 10k Widerstand durch und falten Sie die Enden unten um.



Simplified schema ArLoco shield

Es ist nicht unbedingt notwendig, das ArLoco-Schild zu verwenden.
Es geht auch viel einfacher.



Sie können diese Schaltung selbst auf eine Montageplatte löten.
Der Nachteil ist, dass es keine Stromversorgung und keine richtigen Anschlussklemmen gibt.

Sonstiges

Starten

Wenn der Arduino hochfährt, blinkt die Status-LED dreimal. Sie können sehen, dass der Arduino startet. Der Arduino wird auch neu gestartet, wenn Sie im Konfigurationsmodus in Putty booten.

Abschalten

Das Bedienfeld (Putty) kann jederzeit geschlossen werden. Dafür steht kein eigener Befehl zur Verfügung.

Verwenden Sie also Alt-F4 oder klicken Sie auf das bekannte Kreuz oben rechts im Fenster. Stellen Sie sicher, dass Sie sich im allgemeinen Befehlsmodus befinden, damit die letzten Änderungen für einen Port gespeichert werden.

Logging

Der gesamte Konfigurationsdialog wird in der Datei: Eigene Dateien/Arcomora/ArLoco_<Datum>_<Uhrzeit>.log protokolliert. Außerdem ist <Datum>_<Uhrzeit> die Schließzeit.

Konfigurieren der folgenden Dekoder und MEGA2560

Nach der Installation der Software wird das Decoderprogramm beim Start von „ArLoco konfigurieren“ automatisch auf den Arduino geladen.

Für den zweiten und nachfolgende Decoder gibt es ein anderes Verfahren.

Starten Sie dazu den Shortcut 'Upload program to Arduino'.

Sie sollten dies verwenden, wenn Sie einen MEGA2560 haben.

Die Standardinstallation beinhaltet immer einen Upload auf einen UNO.

Diese finden Sie im Windows-Startmenü im Ordner Arcomora.

Dazu gibt es auch eine Anleitung.

COM-Port anpassen

Der Windows Com-Port ändert sich manchmal, wenn ein Arduino wieder angeschlossen wird.

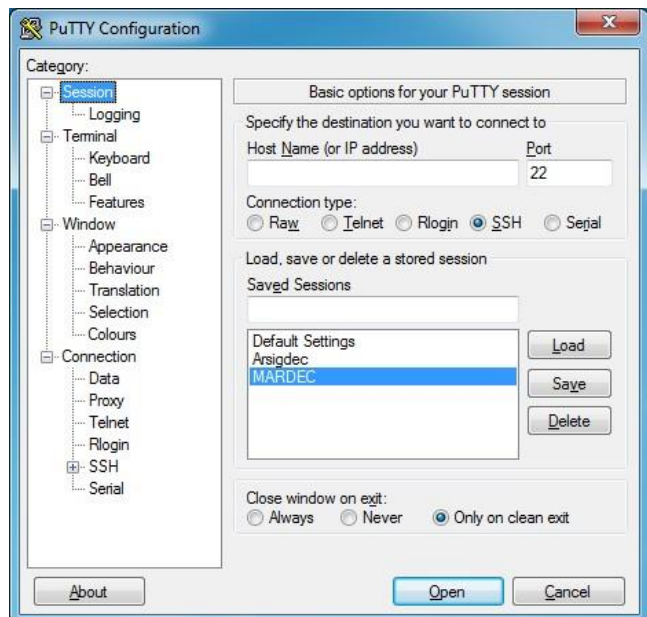
Mit dem Werkzeug COM-Port ändern können Sie den COM-Port ganz einfach ändern.

Dieses finden Sie auch im Windows-Startmenü.

Einstellungen des Konfigurationsbildschirms (Putty)

Sie können das Erscheinungsbild des Bedienfelds wie folgt anpassen.

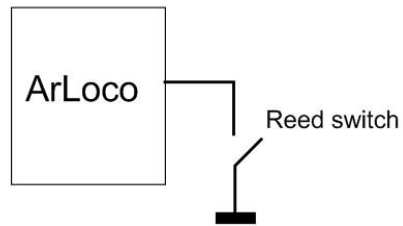
1. Klicken Sie auf die Verknüpfung Putty konfigurieren. Diese finden Sie im Windows-Startmenü im Ordner Arcomora Es erscheint nebenstehender Bildschirm:
2. Klicken Sie auf MARDEC und dann auf Laden.
3. Unter Sitzung→ Protokollierung können Sie die Protokolloptionen anpassen.
Hinweis: Wenn Sie den Dateinamen ändern, können die Konfigurationssitzungen nicht mehr gespeichert werden!
4. Unter Fenster → Aussehen/Verhalten/Farben können Sie auch das Aussehen anpassen.
5. Bei Bedarf können Sie hier auch den COM-Port ändern.
6. Ändern Sie keine anderen Einstellungen!
7. Wählen Sie Sitzung und klicken Sie auf Speichern, um die Einstellungen zu speichern.
8. Klicken Sie auf Öffnen, um das Bedienfeld erneut zu öffnen.



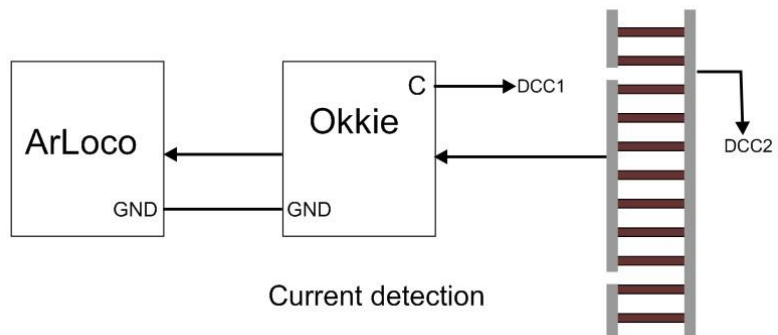
Sie können den COM-Port eines Arduino auch einfach mit dem Programm COM-Port ändern ändern. Sie finden es im Windows-Startmenü im Arcomora-Ordner.

Detektion Methoden

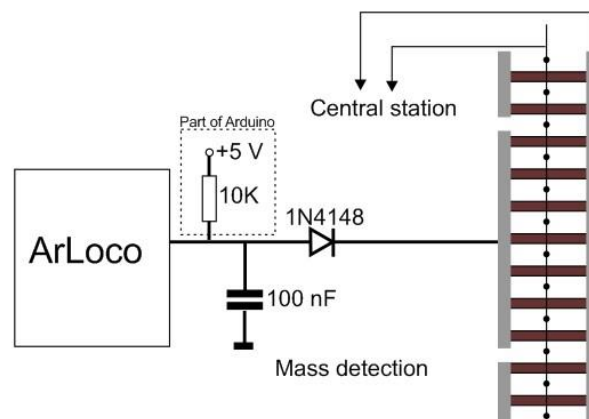
1) Pulsdetektion mit Reedkontakt



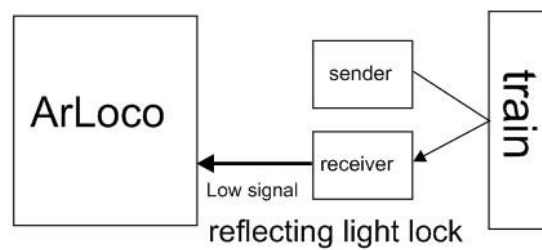
2) Strom Detektion



3) Masseerkennung



4) Lichtschranke mit Reflektion



5) Lichtschranke mit Unterbrechung

