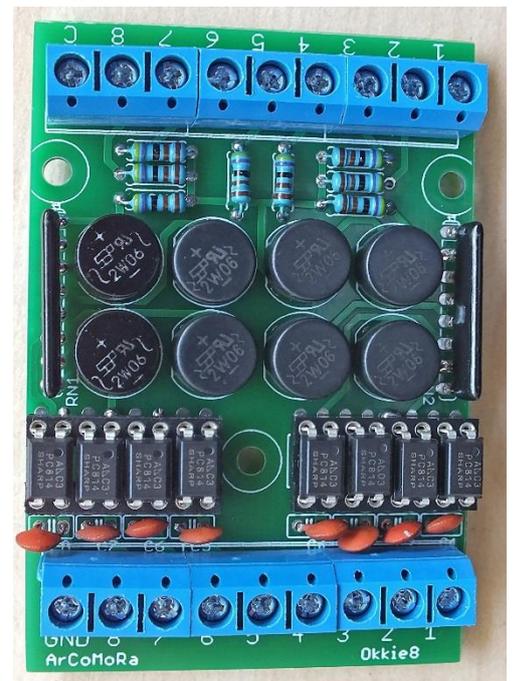
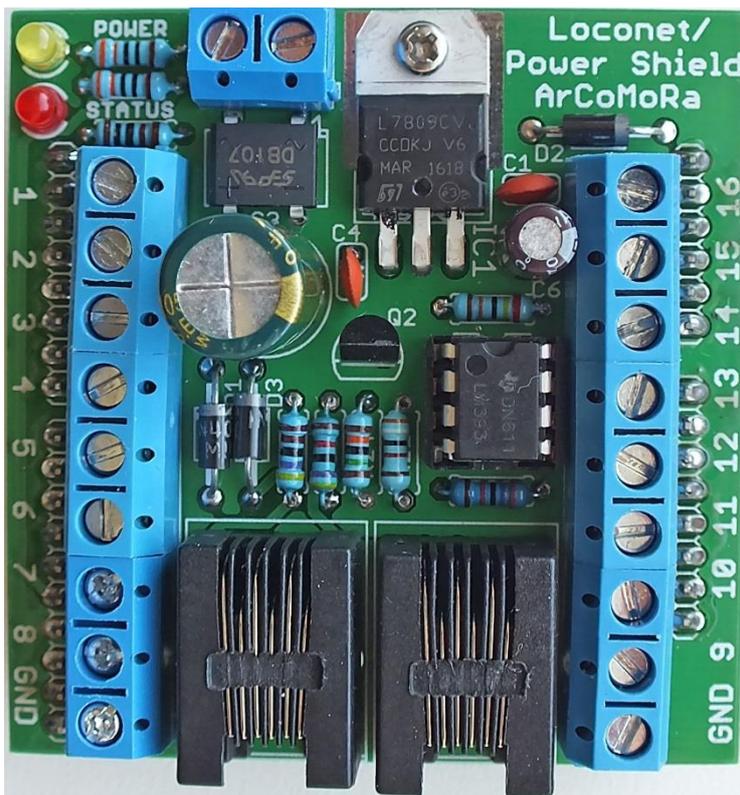


## ARduinoLOCO<sub>net</sub>

The allround feedback encoder with an Arduino and LocoNet®



Author: Nico Teering  
September 2019

Arloco version 3.0

Document version: 3.0UK

Info: [info@Arcomora.com](mailto:info@Arcomora.com)

Web: [www.Arcomora.com/Arloco](http://www.Arcomora.com/Arloco)

LocoNet® is a registered trademark of DigiTrax, Inc



[DOWNLOAD HERE THE ARLOCO SOFTWARE](#)  
[YOU CAN ORDER THE ARLOCO SHIELD AND THE OKKIE HERE](#)

## Introduction

The ArLoco is an Arduino sketch developed for sending train detection information from your railway to your Digital Command Station and from there to your computer program. It uses LocoNet as protocol. So your Command Station must be capable of accepting LocoNet signals.

A train can be detected by using current detection or reed switches. Both methods are supported by the ArLoco sketch.

## For both UNO and MEGA2560

The ArLoco sketch can be used on an Arduino UNO and on an Arduino MEGA2560.

Both Arduino's can use the ArLoco shield.

On an UNO there 16 input channels and on an MEGA 58 input channels.

## Detection methods

The ArLoco has two detection methods:

- Continuous detection
- Pulse detection

### Continuous detection

With continuous detection ArLoco sends, at a falling edge, a LocoNet pulse, with address information, to the central station. The section is now 'occupied'.

At a rising edge ArLoco send again a LocoNet pulse. The section is now 'free'.

Therefore this method should be used with current detection in combination with a current detection print. The output of that print is connected to the input of the ArLoco shield.

To prevent false detections, caused by short interruptions, ArLoco has two build in timers.

#### **Delay before occupied:**

A section is seen as occupied if the input is low for a longer period than the 'delay before occupied' time. This time is default 100ms, but can be set by the 'O'-command. This prevents false notification caused by spikes. The spike at T1 (see figure) is not detected because the duration is shorter than the 'delay before occupied' time. A LocoNet pulse is sent (T2) after the 'delay before occupied' has ended.

#### **Delay before free:**

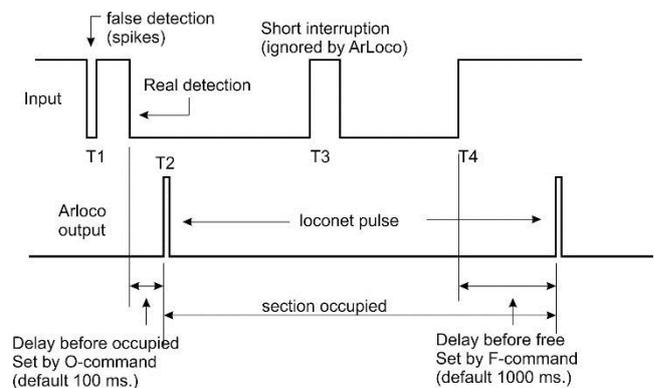
The section is set to free after the input is high again for at least the 'delay before free' time.

(Default 1000 ms.; set by the F-command). The current interruption at T3 is ignored by ArLoco because it is shorter than the 'delay before free' time.

At T4+'delay before free' time the section is set free by a LocoNet pulse.

On an ideal track both timers should be zero.

**Please Note: On sensor 1 there is no current detection possible; only pulse detection!**



### Pulse detection

This method is meant for use with reed switches. If the input goes low, ArLoco will send a LocoNet 'occupied'-pulse to the Central Station.

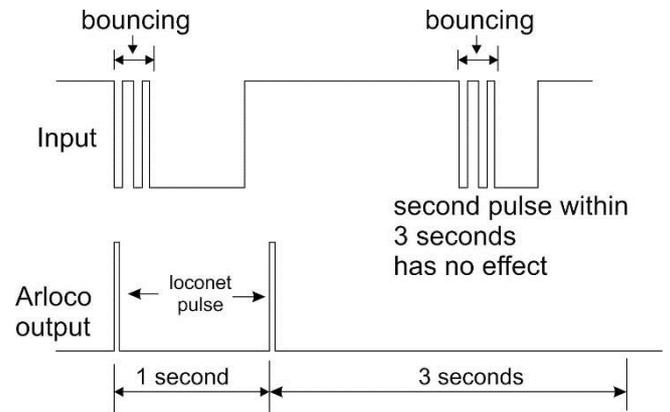
A fixed time of 1 second later ArLoco sends the LocoNet 'free'-pulse. This prevents that contact bouncing causes fake detections.

Also the speed of the train has no influence, even if a train is halted above the reed switch.

Because there are only small currents involved there is no danger for spikes. Therefore no precautions were taken against spikes.

After the second LocoNet pulse, ArLoco will wait three seconds before checking the input again. This prevents that a second LocoNet pulse is generated in case of slowly driving.

This wait time can be set with the 'W'-Command.



### Mixed detection

By default all inputs are configured as Continuous and Pulse for input 1.

However with the S- Command you can set the mode for each individual sensor.

**Only sensor 1 is always only for pulse detection.**

## Addressing

The ArLoco has 16 or 58 sensors. Each sensor has its own specific address. The total address range of ArLoco is from address 1 to address 256.

An input of the ArLoco can be deactivated (A-command). By doing so they don't use an address.

That is easy in case of a large track consisting of independent modules that require each an ArLoco with a limited number of sections that must be detected. So you can use more than 16 ArLoco's.

Each ArLoco must be assigned a Base address (B-command). The first active input will have the base address, the second the base address+1, the third the base address+2, etc.

Thanks to this method, the ArLoco can deal very flexibly with the available address space.

It is also possible to give each sensor its own address. Please note that this address is not used on any other ArLoco. Also, this address must be outside the 'own' address range.

Here is a table that indicates which Arduino pin is used for which sensor.

The pins 1,8,12 and 13 are not used on the UNO.  
The sensor numbers are listed on the shield.

The pins 1,8,12,13 and 48 are not used at the MEGA.  
The sensor numbers 1 through 16 are listed on the shield.  
The pin numbers for the sensor numbers 17 to 58 are listed on the MEGA print.

**Note: from pin 48 the pin numbers and sensor numbers are shifting!**

**Address = Sensornumber + Base address - 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

# Configuring your ArLoco

After the software is installed properly you can start to configure the ArLoco.

Click on the shortcut *ArLoco*.

At the very first time you start *ArLoco*, the drivers will be bound to the USB port to which the Arduino is connected.

### Please Note:

- Every numeric input must be closed with <enter>.
- Only one-character commands don't require an <enter>
- For most inputs the current value does not change with only <enter>. The current value is shown between brackets.
- In configuration mode the yellow led on the shield (or the built-in led on pin 13) is always on.
- In operation mode this led is off. It flashes when sensing an LocoNet signal.
- Turn on the Numlock key on the numeric keypad.
- You may use lower and upper characters for commands.
- Use ONLY backspace key to correct a numeric input.

All screenshots are based on the Arduino UNO. On the MEGA2560, however, they work identically.

You can now enter a command.

The very first time you start the ARLOCO, or after a full reset, the following will be done automatically the I command has been activated. (See with I command). Various default values are set.

An overview of the configuration will be shown automatically at each start of the configuration.

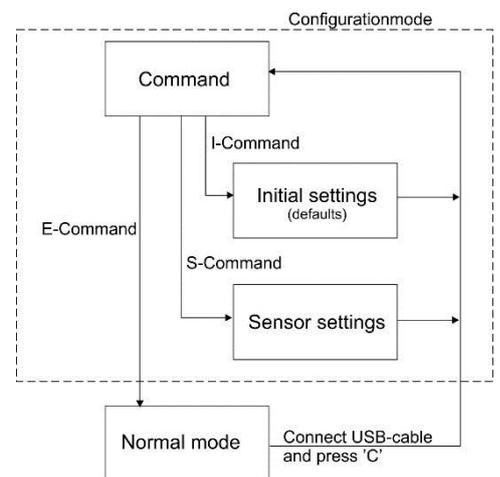
## The Commands

The ArLoco has two modes of operation. Configuration mode and Operation mode.

In the configuration mode you can configure ArLoco with one-character commands. In operation mode ArLoco 'listens' to its sensors and sends LocoNet signal to the Central Station.

Changing between modes is done as shown in the figure below.

```
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/?):
```



Each command consists of a single character and can be entered without hitting the <enter> key. The characters are based on the English description of the commands. By entering a '?' the available commands are shown.

**NOTE: ArLoco 2.x cannot be updated to version 3.0.**

**The method of data storage in EEPROM has been radically changed. This was necessary to give each sensor its own address.**

## I-Command (Initial settings)

The I command adjusts all default settings. There is no submenu but all settings are requested one after the other.

This concerns the following settings.

- Enter the administrative number of the ArLoco.
- The standard type of detection (pulse or continuous)
- ALL inputs are hereby set to this method! (with the exception of sensor 1 = always pulse) Also as M-command per sensor.
- The basic address. Also as a separate B-command.
- Standard inversion of sensors. By default, a low (=0 Volt) signal is considered 'occupied'. However, if a lock is used where the light beam is interrupted by the train, it will give a high signal (=5 Volt). Through inversion ArLoco will see a high signal when occupied. Also as I-command per sensor.
- With pulse detection the time that an input is not 'tested' after sending a busy pulse. (*Wait time*, see also paragraph on pulse detection.)
- The '*delay before free*' time. Default: 1000 ms. It can vary from 1 to 3000 msec.
- The '*delay before occupied*' time. Default: 1000 ms. It can vary from 1 to 3000 msec.

```
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
```

## B-Command (Base address)

With the B-Command you can set the base address of the ArLoco.

Also see the paragraph about addressing.

```
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-Commando (Reset)

The R-command resets all settings of the ArLoco.

After that the program will restart.

```
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-Command (Exit)

With the E-Command you can change to normal operational mode. ArLoco will now give feedback for each detected train.

```
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): █
```

To return to configure mode, connect an USB cable and press 'C' on your keyboard

## V-Commando (View)

The V-command shows the settings of all sensor

```
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-Command (Debug)

With the D-command you can turn on and off the debug-mode. In debug mode every sensor change is reported when in operation mode.

```
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/?):
```

```
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
```

*Normal operational mode*

### Take care:

When in operation mode the Arduino is connected to your PC en Putty is started you can communicate via the keyboard with the Arduino. This is done over pin 0 (Tx) which is also sensor 1.

A hit on the keyboard may cause a fake detection. Make sure Putty is not the active window.

You also can not connect an OKKIE to sensor 1.

```
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
370013 Sensor 11 changed to 1
```

*Debug mode. The number indicates the milliseconds since the ArLoco sketch is started.*

## S-Command (Sensor)

With the S-Command you can change the sensor settings for an individual sensor. Just specify the sensor number you want to change.

```
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/?): █
```

The following subcommands are available:

### L-Command (LocoNet address)

With the L command you can give a sensor a unique address from 1 to 256.

Please note that this address is not already used on another ArLoco. Also, it must remain out of its 'own' address range.

If a sensor already had a different address, you can use address 0 to reset the address to the same value as the base address.

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

With the M-Command you can set the default detection method for **ALL** sensors.

```
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/?): █
```

### I-Command (Inversion)

With the I command, the inversion of the sensor can be changed.

```
Specify action for Sensor 12 (L/A/I/M/V/?): I
Sensor 12 set to Inverted
Specify action for Sensor 12 (L/A/I/M/V/?): I
Sensor 12 set to not Inverted
```

### V-Commando (View)

The V command displays the settings of the selected sensor.

```
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-Command (Active)

The A command can be used to activate or deactivate a sensor.

A sensor that is not active will not be 'tested'. By making a number of sensors inactive, more addresses become available for other ArLoco's.

```
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? █
```

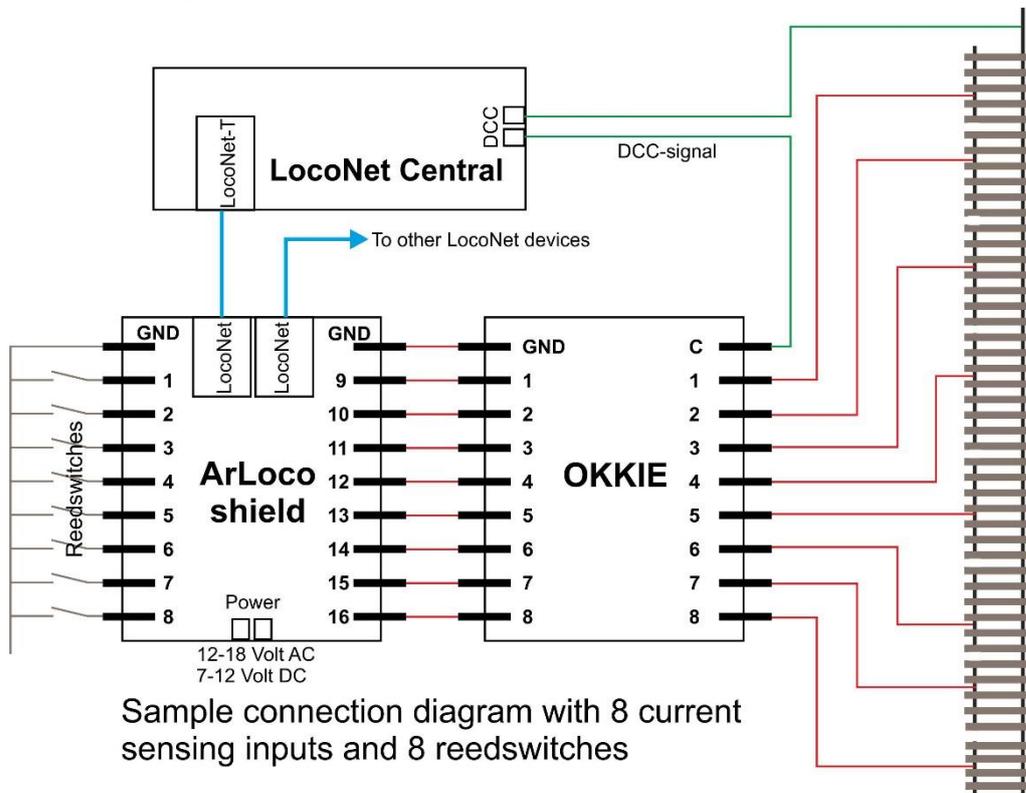
If a sensor is set to inactivated, the user is asked whether the higher sensor numbers should also be set to inactivated. This is useful if you don't want to use some of the transmitters.

When a sensor is set to active again, it gets the address it had before.

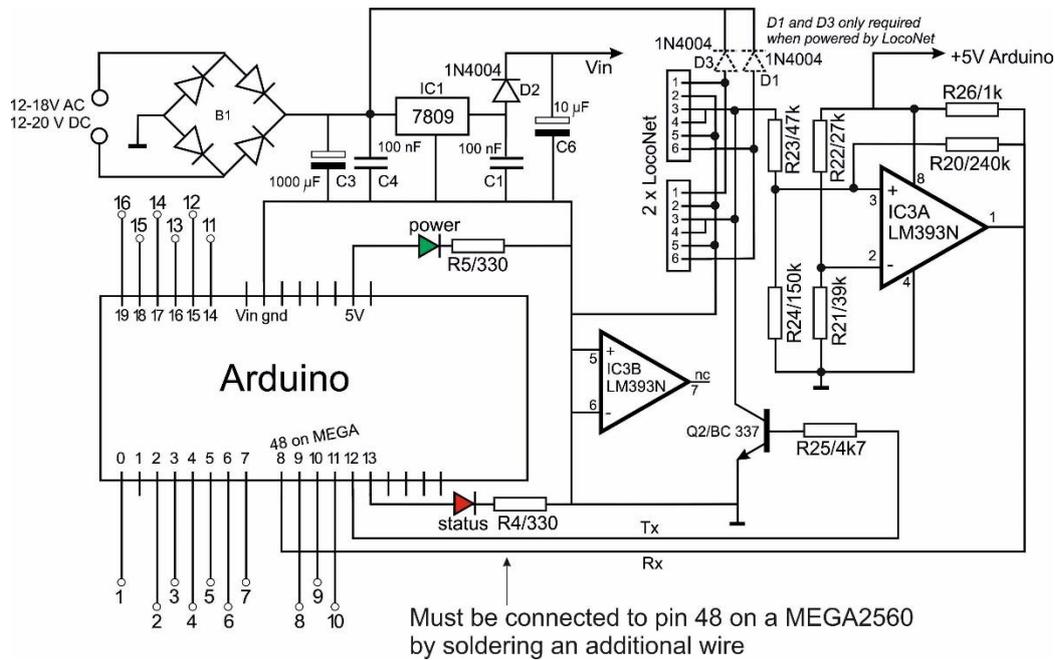
They also ask if all other inactive sensors should be switched back to active.

```
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) █
```

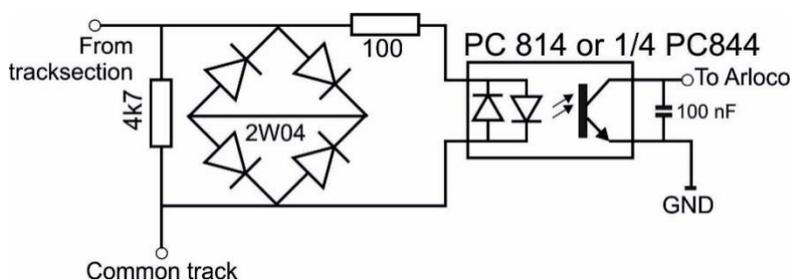
## Connection diagram



## Schema ArLoco



## Schema OKKIE



Schema for one track section.  
The Okkie print contains 8 of these circuits.

# Building the ArLoco shield

## Components:

Resistor	R4	1k	
Resistor	R5	1k	
Resistor	R20	240k	
Resistor	R21	39k	
Resistor	R22	27k	
Resistor	R23	47k	
Resistor	R24	150k	
Resistor	R25	4k7	
Resistor	R26	1k	

Capacitor	C1	100 nF	
Capacitor	C4	100 nF	

Capacitor	C6	10 uF	
-----------	----	-------	---

Capacitor	C3	1000 uF	
-----------	----	---------	---

Diode bridge	B1	DB107	
--------------	----	-------	--

Power-IC	IC1	L7809CV	
----------	-----	---------	--

Opamp	IC3	LM393N	
-------	-----	--------	---

Diode	D1	1N400x	
-------	----	--------	---

Diode	D2	1N400x	
-------	----	--------	---

Diode	D3	1N400x	
-------	----	--------	---

Transistor	Q2	BC337	
------------	----	-------	---

Bolt and nut M2,5

6 x 3pins screw terminal

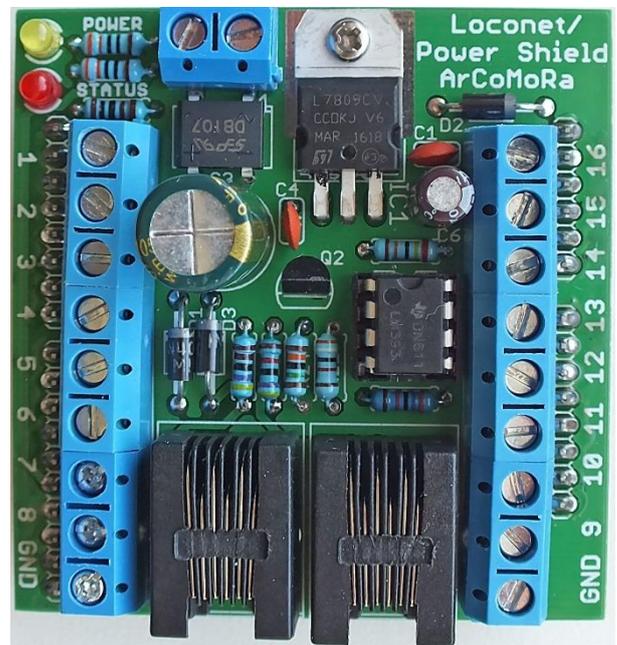
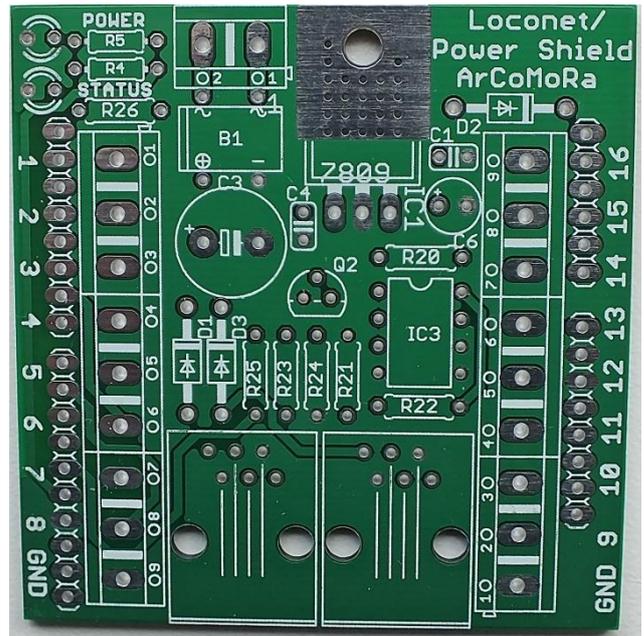
1 x 2pins screw terminal

2 x LocoNet RJ12 connector

40 pins pin header

1 red led and 1 yellow led

8 pins IC-socket



The Arduino can be powered by the shield in three different ways:

- External power (no need for D1 and D3)
- By LocoNet (no need for B1, C3 and 2 pins terminal)
- Via jack plug (no need for D1, D2, D3, B1, C1, C3, C4, C6, IC1 and 2 pins terminal)

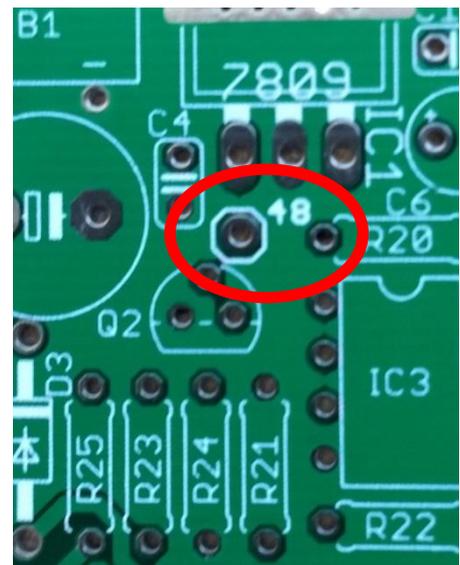
The LocoNet circuit is always powered by the 5V output of the Arduino.

## Recommended building order ArLoco shield:

1. All resistors. **Take care of the color code**
2. Diode D2. **Take care of the orientation.**
3. Diode D1 and D3 if you want to power the Arduino via LocoNet
4. Diode bridge DB107. **Take care of the orientation.**
5. Place IC socket with the gap towards resistor R20
6. Capacitors C1 and C4 of 100nF
7. IC1 (LM7809). Use bolt and nut.
8. The red and yellow led. The short wire is positioned at the edge of the print.
9. Transistor Q2, (BC337 or 2N3904)
10. Place IC LM393 in its socket. The point at pin 1 is towards R20
11. Cut the 40 pins pin header in 4 pieces of 6, 8, 8, and 10 pins and place them on an Arduino. Place the shield over the pins and solder them .
12. Slide three 3-pins connector into each other to a 9 pins screw terminal. Do the same with the other three. Place the screw terminals on the print together with the 2-pins power connector. Turn the print upside down; because there are no higher components then the screw terminals it lays flat on the table. Solder the screw terminals.
13. Capacitor C6 (10 uF). **Take care of the orientation.** The longest wire is the plus.
14. Solder both RJ12 connectors.
15. **ONLY FOR USE ON THE MEGA2560**

The latest version of the pcb has an additional hole near transistor Q2. You must solder a Dupont pin in it and connect it with a Dupont wire to pin 48.

16. Capacitor C3 (1000 uF). **Take care of the orientation.**  
The longest wire is the plus



## Testing

To test the external power connect AC or DC power source (12-18 V) to the power connector. The green or yellow led must be on. Test this without LocoNet cable.

To test the LocoNet power connect a LocoNet cable to a RJ12 plug and a LocoNet output of a Command station. The green or yellow led must be on. Test this without power cable.

Now connect the Arduino with an USB cable with your PC, start the *ArLoco* shortcut and go to the normal mode. On the Putty screen you can now see every detected train. Now test each sensor by connecting the input with a wire to GND.

```
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
```

# Building the Okkie

## Components:

8 x resistor 4k7

8 x diode bridge 2Wxx

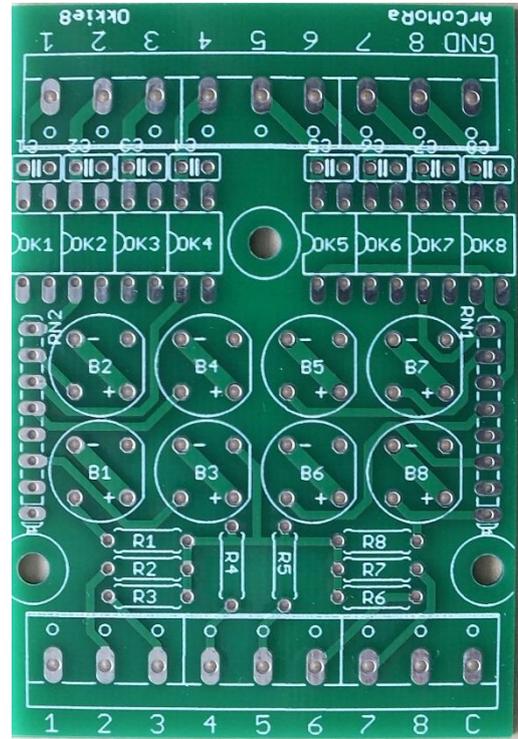
8 x Capacitor 100 nF

8 x Optocoupler PC814 (OK1 - OK8)  
or 2 x PC844

2 x 16 pins IC sockets.

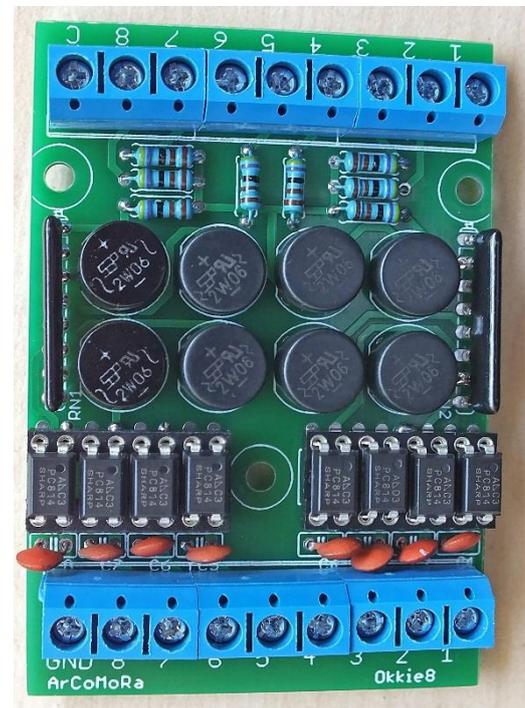
2 x resistor array 100 ohm

6 x 3-pins screw terminals.



## Recommended building order Okkie print

1. All resistors
2. All capacitors
3. Both IC-sockets
4. All diode bridges. Take care of the orientation. The longest wire is the + (plus).
5. Both resistor arrays
6. Slide three 3-pins connector into each other to a 9 pins screw terminal. Do the same with the other three. Place the screw terminals on the print and turn the print upside down. Because there are no higher components then the screw terminals it lays flat on the table. Solder the screw terminals.
7. Place the optocouplers (8x PC814 or 2x PC 844). See picture for the orientation. Both types have the same orientation.

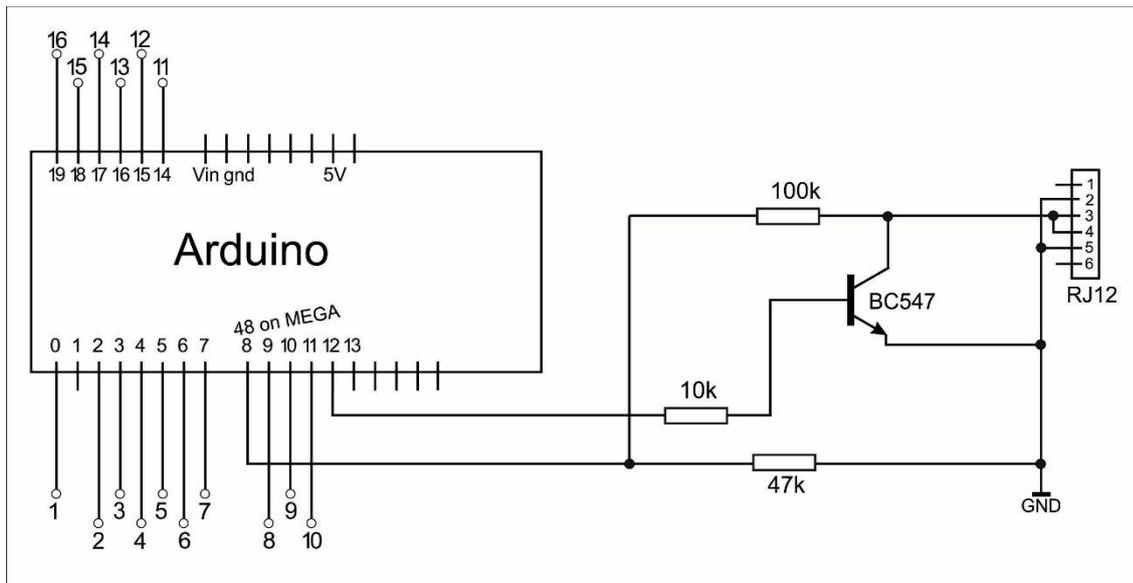


[DOWNLOAD HERE THE ARLOCO SOFTWARE](#)  
[YOU CAN ORDER THE ARLOCO SHIELD AND THE OKKIE HERE](#)

## Simplified schema ArLoco shield

It is not necessary to use the ArLoco shield.

You can also make your own shield based on the simplified schema below.



### Configure the following decoders and MEGA2560

After installing the software, the decoder program will automatically be loaded on the Arduino when you start Configure ArLoco for the first time.

For the second and next decoders there is a different way of working.

Start the shortcut '**Upload program to Arduino**'.

**This must be used if you have a MEGA2560.** With the standard installation, an upload to a UNO is always done.

This can be found in the Windows Start menu in the Arcomora folder.

A manual for this can be found there as well.

### Customize the Com-port.

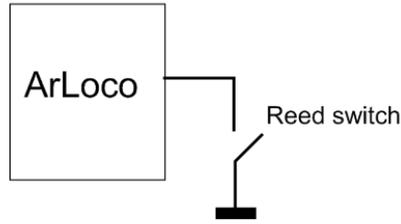
The Windows Com port may want to change when reconnecting an Arduino.

With the tool Change COM port you can easily change the Com port.

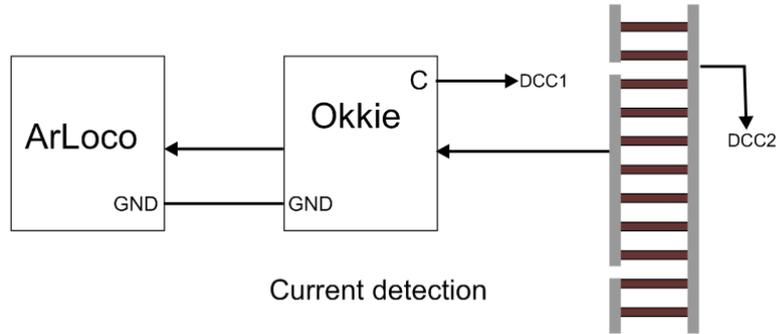
These can also be found in the Windows Start menu.

# Detection methods

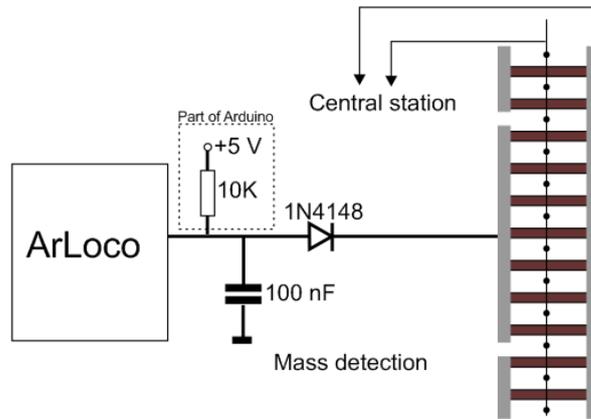
## 1) Pulsedetection with reedswitch



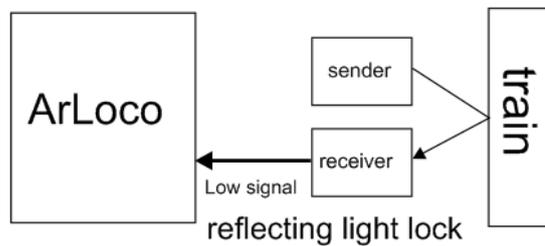
## 2) Current detection



## 3) Mass detection



## 4) Lightlock with reflection



## 5) Interrupted light lock

