



# DT-M2M

Software-Description

2023 March

# INDEX

<b><u>1. Introduction</u></b>	<b>3</b>
1.1. General remarks	4
1.2. Customer satisfaction	4
1.3. Customer response	4
<b><u>2. DT-M2M Description</u></b>	<b>6</b>
2.1. General	7
2.2. Communication - From A to B	8
2.3. Communication - Between A and B	9
2.4. Communication - From A to multiple ends	10
<b><u>3. DT-M2M Webinterface</u></b>	<b>11</b>
3.1. Open the webinterface	12
3.2. Configuration - Configure the jobs	13
3.3. Destination - Configure the destinations	20
3.4. Status - Overview	22
3.5. Log - Protocol	25
<b><u>4. Appendix</u></b>	<b>26</b>
4.1. Revision	27
4.2. Copyrights and trademarks	28

# Introduction

---



# **1. Introduction**

## **1.1. General remarks**

### **Congratulations on purchasing a high quality DEDITEC product!**

Our products are developed by our engineers according to today's required quality standards. We pay attention already during the development to flexible expandability and long availability.

### **We develop modular!**

Due to a modular development we shorten the development time and - what of course benefits the customer - we sell at a fair price!

### **We ensure a long delivery availability!**

If used semiconductors are no longer available, we can react faster. With us mostly only modules have to be redesigned and not the whole product. This increases the delivery availability.

## **1.2. Customer satisfaction**

### **A satisfied customer is our first priority!**

If something is not to your satisfaction, just contact us by phone or mail.

We will take care of it!

## **1.3. Customer response**

The best products grow with our customers. We are always grateful for any suggestions or proposals.



## **DT-M2M Description**



## **2. DT-M2M Description**

### **2.1. General**

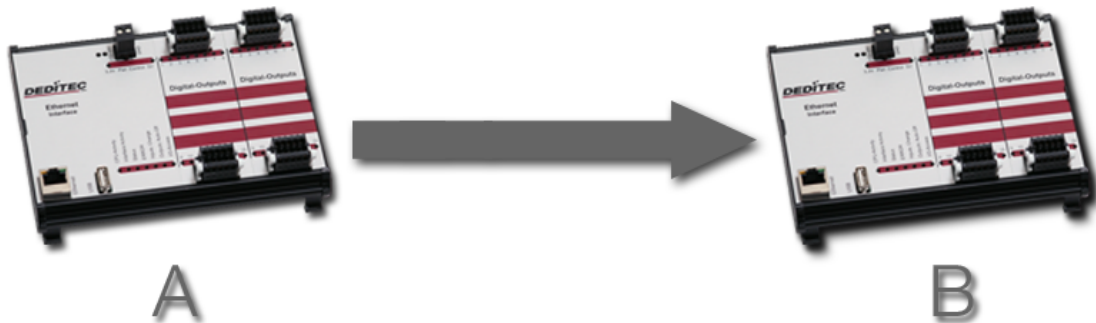
With the DT-M2M (Machine-to-Machine) software solution, several DEDITEC devices with Ethernet interface can communicate with each other.

With the help of the DT-M2M software, digital and analog signals can be transmitted over long distances.

Up to 10 jobs can be configured on a sender module which can communicate with up to 5 different receiver modules.

You can set which information will be send in a specific interval to a specific receiver module.

## 2.2. Communication - From A to B



### Description:

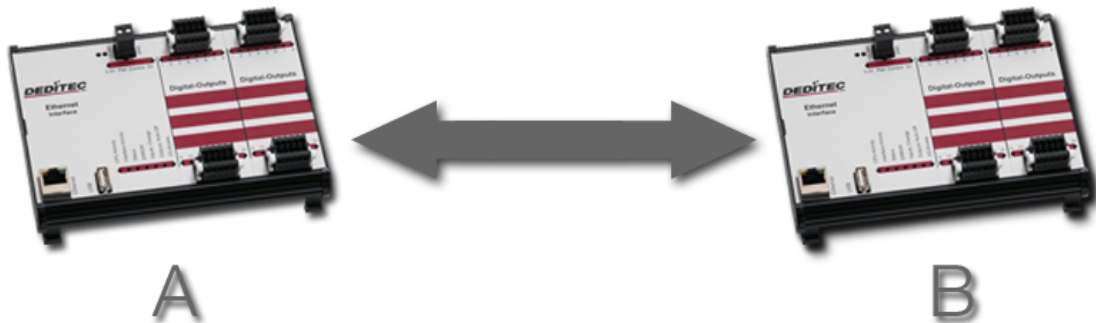
In diesem Szenario werden sowohl digitale als auch analoge Signale an den Eingängen des Sendemoduls A gemessen und an bestimmte Ausgänge des Empfängermoduls B gesendet.

### Configuration:

Die DT-M2M-Software wird nur auf Modul A benötigt und muss auch auf Modul A konfiguriert werden.



## 2.3. Communication - Between A and B



### Description:

In this scenario digital and also analog signals are measured at module A and module B and sent to the outputs of the opposite module.

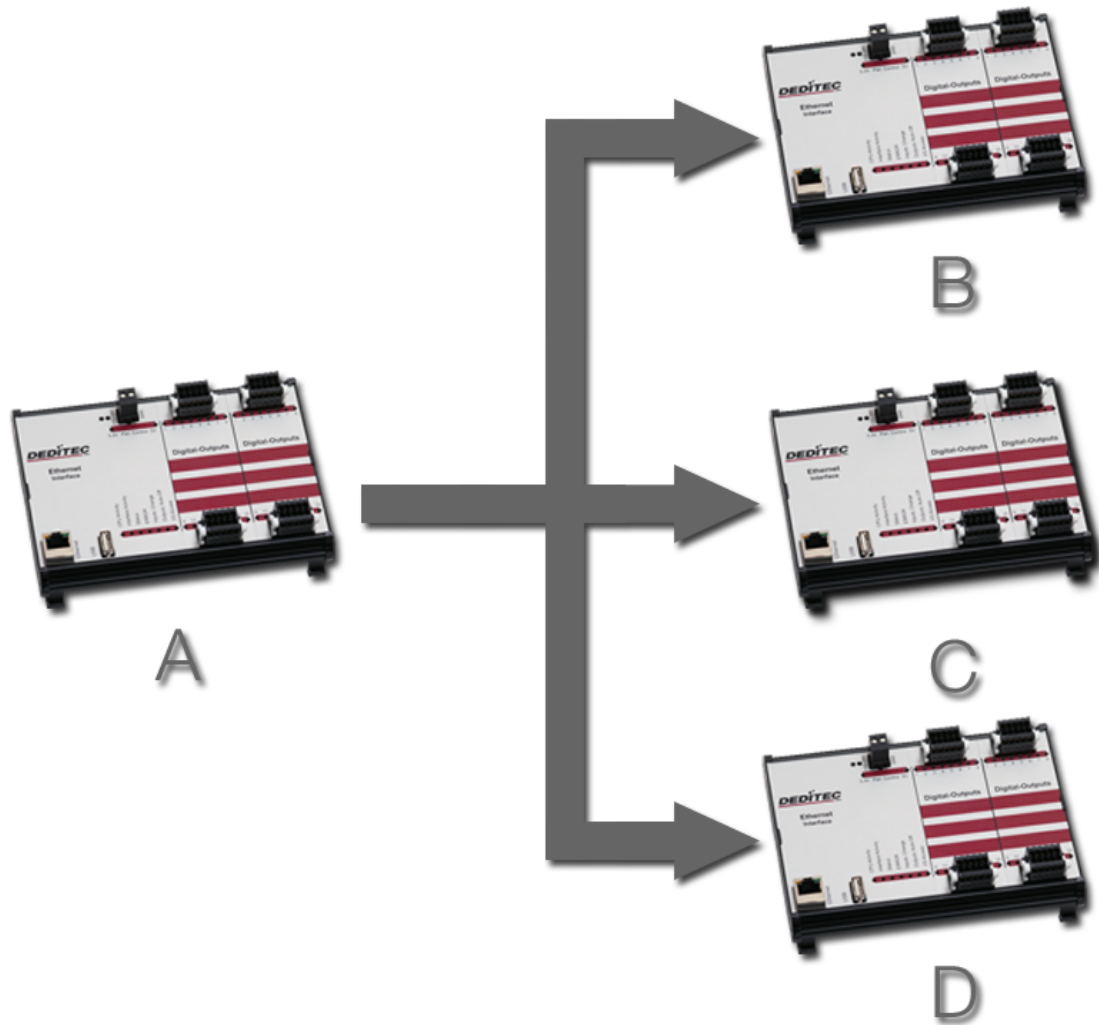
The DT-M2M software is required on both modules.

### Configuration:

On module A you need to configure which signals will be send to module B.

On module B you need to configure which signals will be send to module A.

## 2.4. Communication - From A to multiple ends



### Description:

In this scenario digital and also analog signals are measured at module A and will be sent to module B, C and D.

The DT-M2M software is only required on module A.

### Configuration:

On module A you need to configure which signals will be sent to module B, C and D.

## **DT-M2M Webinterface**

---



## **3. DT-M2M Webinterface**

### **3.1. Open the webinterface**

To open the webinterface enter the IP-address of the sender module into the address bar of a browser of your choice.

The default ip-address is "192.168.1.1".

## 3.2. Configuration - Configure the jobs

With this page of the webinterface you can configure the jobs of the DT-M2M software. Up to 10 jobs can be configured and can be active at the same time.

A job determined which signals will be send from the sender module to a receiver module.

### M2M - Configuration

1.

Selected job:

Job0 (job-name) ▼

2.

Job name:

job-name

Set name

3.

Job active:

☐

4.

Send to IP-Address:

destination-name (192.168.1.1) ▼

5.

Select data to send:

digital-inputs (16) ▼

6.

Intervall active:

☒ value: 1

\*

sec ▼

e.g. 500\*msec is the equivalent to every half second

7.

On change active:

☐

8.

Start channel:

1 ▼

9.

AD-Mode:

0-10V ▼

10.

End channel:

7 ▼

11.

Start channel:

1 ▼

12.

DA-Mode:

0-10V ▼

13.

End channel:

7

14.

Save parameter

**1.) Selected job:** With the drop-down menu you can select the job which you want to configure.

In the brackets is the name of the selected job displayed.

Up to 10 jobs can be configured.

**2.) Job name:** Here you can set a individual name for the job.

With a click on the "Set name" button the name will be saved.

**3.) Job active:** With the help of this checkbox you can activate/deactivate the selected job.

A job will only be executed then this checkbox is active.

If this checkbox is not active, all settings will be ignored for this job.

**4.) Send to IP-Address:** With the drop-down menu you can choose the receiver (destination) module for the selected job.

How you can change the settings of all receiver modules see the chapter >> "Destination - Configure the destinations"

**5.) Select data to send:**

With the drop-down menu you can choose which data will be send to the receiver module.

You can choose between "digital-inputs" oder "analog-inputs". The value in the brackets will show the amount of available inputs on the sender module.

**6.) Intervall active:** This checkbox determined if the job will be executed in a set interval.

The interval time is calculated with the Value \* and the selected time unit. You can choose between msec, sec, min and hour.

#### Examples for the interval

Value	Time unit	Interval
100	msec	10x in the second
1500	msec	each 1,5 seconds
60	sec	every minute
150	sec	each 2,5 minute
90	min	each 1,5 hours

**7.) On change active:** Not implemented at this moment!

With this function it will be possible to act immediately in a future when a input change is detected independent of the interval.

#### **Input channels of sender**

**8.) Start channel:** Sets the first channel which will be read from the sender module

**9.) AD-Mode:** Sets the voltage or current mode of the sender module. Please pay attention if the mode is supported from your sender module.

**10.) End channel:** Sets the last channel which will be read from the sender module.

#### **Outputs channels of receiver**

**11.) Start channel:** Sets the first channel of the receiver module which the data will be written.

**12.) DA-Mode:** Sets the voltage or current mode of the receiver module. Please pay attention if the mode is supported from your receiver module.

**13.) End channel:** Will be calculated. Formula (Start channel of the receiver module + (End channel of the sender module - start channel of the sender module))



**Feature AD-Mode/DA-Mode:** If the voltage or current mode is set identical for the sender and receiver module, the measured values will be written 1 on 1 from the inputs to the outputs.

Are different modes set, the measured data will be converted based on the 16Bit value.

AD-Mode	Measured	DA-Mode	Output
0-10V	5V (32767 decimal)	0-10V	5V (32767 decimal)
0-10V	5V (32767 decimal)	0-5V	2,5V (32767 decimal)
0-10V	5V (32767 decimal)	0-20mA	10mA (32767 decimal)
0-20mA	15mA (49149 decimal)	+10V	+2,5V (49149 decimal)
0-20mA	0mA (0 decimal)	+5V	-5V (0 decimal)

#### Examples for setting the start and end channels

Configure	Sender start channel	Sender end channel	Receiver start channel
A	1	1	1
B	1	4	1
C	1	1	5
D	1	2	4
E	4	8	2

A: The state of the first input of the sender module, will be send to the first output of the receiver module.

B: The states of the inputs 1 to inclusive 4 of the sender module, will be send to the outputs 1 to inclusive 4 of the receiver module.

C: The state of the first input of the sender module, will be send to output 5 of the receiver module.

D: The states of inputs 1 and 2 of the sender module, will be send to the output 4 and 5 of the receiver module.

E: The states of the inputs 4 to inclusive 8 of the sender module, will be send to the output 2 to inclusive 4 of the receiver module.

**14.) Save parameter:** Press this button to save all settings to the module. If you leave the Configuration page without clicking this button, all changed settings will be discarded.

### 3.3. Destination - Configure the destinations


Diese Seite der Weboberfläche dient zur Konfiguration der Zielmodule (Destinations). Es können bis zu 5 Zielmodule eingestellt werden.

Die hier konfigurierten Zielmodule stehen in der Konfiguration der Jobs zur Verfügung.

#### M2M - Destination

On this page you can configure the receiver of the M2M communication.

1.	Destination:	<input type="text" value="destination-name (192.168.1.1)"/>	
2.	Name:	<input type="text" value="destination-name"/>	6. <input type="button" value="Test Communication"/>
3.	IP-Address:	<input type="text" value="192.168.1.1"/>	
4.	Port:	<input type="text" value="0"/>	
5.	Timeout:	<input type="text" value="5000"/>	

  
7.

**1.) Destination:** Über das Drop-Down Menü können Sie das Zielmodul auswählen, welches Sie konfigurieren möchten.

In den Klammern steht, zur besseren Orientierung, der vergebene Name des Zielmoduls.

Es ist möglich bis zu 5 verschiedene Zielmodule zu konfigurieren.

**2.) Name:** Hier können Sie einen individuellen Namen für das Zielmodul vergeben.

**3.) IP-Address:** IP-Adresse des Zielmoduls.

**4.) Port:** Port des Zielmoduls.

**5.) Timeout:** Wert in Millisekunden.

**6.) Test communication:** Mit diesem Button wird die Verbindung zum Zielmodul überprüft.

**7.) Save parameter:** Mit diesem Button werden alle Einstellungen im Modul gespeichert. Ohne Klick auf diesen Button werden alle getroffenen Einstellungen verworfen.

### 3.4. Status - Overview

The status page will give you a overview of all jobs and receiver modules.

#### M2M - Status

1.

No.	Jobname	Activ	Time next	Time min	Time max	Count open	Count ok	Comm. errors	Delib errors	Readback errors
1.	ch 1-8	⊙	0	0	30	1	73475	0	0	0
2.	ch 9-16	⊙	0	0	37	0	73475	0	0	0
3.	job-name	●	0	0	0	0	0	0	0	0
4.	job-name	●	0	0	0	0	0	0	0	0
5.	job-name	●	0	0	0	0	0	0	0	0
6.	job-name	●	0	0	0	0	0	0	0	0
7.	job-name	●	0	0	0	0	0	0	0	0
8.	job-name	●	0	0	0	0	0	0	0	0
9.	job-name	●	0	0	0	0	0	0	0	0
10.	job-name	●	0	0	0	0	0	0	0	0

#### Destination

2.

No.	Destinationname	IP-Address	Open	Time Min	Time Max	Count open	Count ok	Comm. errors	Delib errors	Readback errors
1.	zielmodul ip 6	192.168.1.6	⊙	0	22	1	146950	0	0	0
2.	destination-name	192.168.1.1	●	0	0	0	0	0	0	0
3.	destination-name	192.168.1.1	●	0	0	0	0	0	0	0
4.	destination-name	192.168.1.1	●	0	0	0	0	0	0	0
5.	destination-name	192.168.1.1	●	0	0	0	0	0	0	0

**1.) Jobs:** Overview of all 10 jobs. Number and jobname will help to identify the job

**Active:** Checked = Job is active    Unchecked = Job is inactive

**Time next:** Time in milliseconds for the next execution of the job.

**Time min:** Time in milliseconds for the least time it took to execute this job.

**Time max:** Time in milliseconds for the most time it took to execute this job.

**Count open:** Shows how often the receiver module was opened from this job.

**Count ok:** Shows how often the job was executed without errors.

**Communication errors:** Shows how often communication errors with the receiver module occurs.

**Delib errors:** Shows internal communication errors with the delib driver library

**Readback errors:** Shows target-actual comparison between the output states of the receiver module.

**2.) Receiver modules:** Overview of all 5 receiver modules. Number and destinationname will help to identify the receiver module.

**IP-Address:** Shows the ip-address of the receiver module.

**Open:** Checked = Receiver module is open      Unchecked = Receiver module is not open

**Time min:** Shows the shortest response time in milliseconds.

**Time max:** Shows the longest response time in milliseconds.

**Count open:** Shows how often this receiver module was opened.

**Count ok:** Shows how often all jobs for this receiver module were executed without errors.

**Communication errors:** Shows how often communication errors with the receiver module occurs.

**Delib errors:** Shows internal communication errors with the delib driver library.

**Readback errors:** Shows target-actual comparison between the output states of the receiver module.



### 3.5. Log - Protocol

All events of the sender module and the DT-M2M software will be displayed here.

#### M2M - Log

Modul Status: OK

1. **ALL** DT-M2M DEDITEC DELIB others

Time	Application	Information
01.01.1970 00:52:21	RO-ETH-M2M	Reloading job/destination parameters!
01.01.1970 00:52:21	RO-ETH-M2M	*****
01.01.1970 00:52:21	RO-ETH-M2M	-   Intern   -   -   258224
01.01.1970 00:52:21	RO-ETH-M2M	4   destination-name   192.168.1.1   0   5000   0
01.01.1970 00:52:21	RO-ETH-M2M	3   destination-name   192.168.1.1   0   5000   0
01.01.1970 00:52:21	RO-ETH-M2M	2   destination-name   192.168.1.1   0   5000   0
01.01.1970 00:52:21	RO-ETH-M2M	1   destination-name   192.168.1.1   0   5000   0
01.01.1970 00:52:21	RO-ETH-M2M	0   destination-name   192.168.1.5   0   5000   0
01.01.1970 00:52:21	RO-ETH-M2M	*****
01.01.1970 00:52:21	RO-ETH-M2M	Nr   Name   IP-Addr   Port   Timeout   Handle
01.01.1970 00:52:21	RO-ETH-M2M	***DEST*****
01.01.1970 00:52:21	RO-ETH-M2M	*****
01.01.1970 00:52:21	RO-ETH-M2M	9   job-name   0   di   1000   0   7   0   0-10   0-10   1   0   0   1
01.01.1970 00:52:21	RO-ETH-M2M	8   job-name   0   di   1000   0   7   0   0-10   0-10   1   0   0   1
01.01.1970 00:52:21	RO-ETH-M2M	7   job-name   0   di   1000   0   7   0   0-10   0-10   1   0   0   1
01.01.1970 00:52:21	RO-ETH-M2M	6   job-name   0   di   1000   0   7   0   0-10   0-10   1   0   0   1

2. Get complete Log 3. Get last 1000 Log Entries 4. Clear Log

1.) **Tabs:** With the tabs on top of the log you can change the category of the log.

2.) **Get complete Log:** All events will be displayed.

3.) **Get last 1000 Log Entries:** The last 1000 events will be displayed.

4.) **Clear Log:** All logs entries will be deleted.

# Appendix



## **4. Appendix**

### **4.1. Revision**

Rev 3.00

DEDITEC Design Update

Rev 2.00

First revision

## 4.2. Copyrights and trademarks

Linux is a registered trademark of Linus Torvalds.

USB is a registered trademark of USB Implementers Forum Inc.

LabVIEW is a registered trademark of National Instruments.

Intel is a registered trademark of Intel Corporation.

AMD is a registered trademark of Advanced Micro Devices, Inc.

ProfiLab is a registered trademark of ABACOM Ingenieurbüro GbR.

ispVM System is a registered trademark of Lattice Semiconductor Corporation.

Windows, Visual-C/C++, -C#, -Basic, -Basic.NET and Visual-Studio are registered trademarks of Microsoft Corporation.

Delphi is a registered trademark of Borland Software Corporation.

Java is a registered trademark of Oracle Corporation.