Irinos IR Users Manual

This users manual has been optimized for an interactive web view.
Use the PDF version only, if no access to the web version is available.

Introduction

1 Introduction

1.1 Imprint

Title	Irinos IR Users Manual
Manufacturer	Messtechnik Sachs GmbH Siechenfeldstraße 30/1 D-73614 Schorndorf Germany Phone +49 7181 99960-0 post@messtechnik-sachs.de
For use with	Measurement modules Irinos IR
Copyright note	© 2015 - 2016 Messtechnik Sachs GmbH, Germany
Trademarks	All product names used in this manual are trademarks of their respective owners.
Material-No.	785-1018
Change not	Subject to change without notice.
Release date	10/10/2016

1.2 Revision history

Version	Datum	Changes
A	2016-05-17	First revision
B	20	Correction of pin assignment for RS422 incremental

	16-07-14	encoders 96 (picture).
C	2016-09-30	Integration of control panel IR-HMI1 61 into the documentation. Affected chapters: • Overview of components 32 • Product description IR-HMI1 61 • Bit I/O via MscDll 160
D	2016-10-10	Correction of pin assignment for M12 power connector 82.

1.3 Legal notes

1.3.1 Terms of use for software & documentation

I. Protection rights and scope of use

Messtechnik Sachs provides operating instructions, manuals, documentation, and software programs - all collectively referred to as "LICENSED OBJECT" below - either on portable data storage devices (e.g. diskettes, CD ROMs, DVDs, etc.), in written (printed) form or in electronic form, for a fee and/or free of charge. The LICENSED OBJECT is subject to proprietary safeguarding provisions among other regulations. Messtechnik Sachs or third parties have protection rights for this LICENSED OBJECT. In so far as third parties have whole or partial right of access to this LICENSED OBJECT, Messtechnik Sachs has the appropriate rights of use. Messtechnik Sachs permits the user the use of the LICENSED OBJECT under the following conditions:

1.1) Scope of use for electronic documentation

a) With the acquisition/purchase or relinquishment of a LICENSED OBJECT, you as the user acquire a simple, non-transferable right of use with regard to the respective LICENSED OBJECT. This right of use authorises the user to use the LICENSED OBJECT for the user's own, exclusively company-internal purposes on any number of machines within the user's business premises. This right of use includes exclusively the right to save the LICENSED OBJECT on the central processors (machines) used at the location.

b) Irrespective of the form in which operating instructions and/or documentation are provided, the user may furthermore print out any number of copies on a printer at the user's location, providing this printout is printed with or kept in a safe place together with these complete terms and conditions of use and other user instructions.

c) With the exception of the Messtechnik Sachs logo, the user has the right to use pictures and texts from the operating instructions/documentation for creating the user's own machine and system documentation. The use of the Messtechnik Sachs logo requires written consent from Messtechnik Sachs. The user is responsible for ensuring that the pictures and texts used match the machine/system or the product.

d) Further uses are permitted within the following framework:
Copying exclusively for use within the framework of machine and system documentation from electronic documents of all documented supplier components. Demonstrating to third parties exclusively under guarantee that no data material is stored wholly or partly in other networks or other data storage devices or can be reproduced there.
Passing on printouts to third parties not covered by the regulation in item 3, as well as any processing or other use are not permitted.

1.2) Scope of use for software products
For any type of Messtechnik Sachs software including the associated documentation, the customer shall receive a non-exclusive, non-transferable and time-unlimited right of use on a certain hardware product or on a hardware product to be determined in individual cases. Messtechnik Sachs shall remain the owner of the copyright as well as of any other industrial property rights. The customer may make copies for back-up purposes only. Any copyright notes may not be removed.

2. Copyright note
Every LICENSED OBJECT contains a copyright note. In any duplication permitted under these provisions, the corresponding copyright note of the original document concerned must be included:
Example: © 2016, Messtechnik Sachs GmbH,
D-73614 Schorndorf

3. Transferring the authorisation of use
The user can transfer the authorisation of use re. the respective LICENSED OBJECT as per these provisions in the scope and with the limitations of the conditions in accordance with items 1 and 2 completely to a third party. The third party must be made explicitly aware of these terms and conditions of use.

II. Exporting the LICENSED OBJECT
When exporting the LICENSED OBJECT or parts thereof, the user must observe the export regulations of the exporting country and those of the acquiring country.

III. Warranty
1. Messtechnik Sachs products are being further developed with

regard to hardware and software. If the LICENSED OBJECT, in whatever form, is not supplied with the product, i.e. is not supplied on a data storage device as a delivery unit with the relevant product, Messtechnik Sachs does not guarantee that the electronic documentation corresponds to every hardware and software status of the product. In this case, the printed documentation from Messtechnik Sachs accompanying the product is alone decisive for ensuring that the hardware and software status of the product matches that of the electronic documentation.
2. The information contained in an item of electronic documentation can be amended by Messtechnik Sachs without prior notice and does not commit Messtechnik Sachs in any way.
3. Messtechnik Sachs guarantees that the software program it created agrees with the change description and program specification but not that the functions included in the software run entirely without interruptions and errors or that the functions included in the software can run or meet the requirements in all combinations selected by and in all conditions of use designated by the acquirer.

IV. Liability/limitations on liability
1. Messtechnik Sachs provides LICENSED OBJECTS to allow the user to use - in conformity with the contract - Messtechnik Sachs products which require software for proper operation, or to assist the user in creating the user's machine and system documentation. In the case of electronic documentation which in the form of data storage devices does not accompany a product, i.e. which is not supplied together with that product, Messtechnik Sachs does not guarantee that the electronic documentation separately available/supplied matches the product actually used by the user.
The latter applies particularly to extracts of the documents for the user's own documentation. The guarantee and liability for separately available/supplied portable data storage devices, i.e. with the exception of electronic documentation provided on the Internet/Intranet, are limited exclusively to proper duplication of the software, whereby Messtechnik Sachs guarantees that in each case the relevant portable data storage device or software contains the latest status of the documentation. In respect of the electronic documentation on the Internet/Intranet, it is not guaranteed that this has the same version status as the last printed edition.
2. Furthermore, Messtechnik Sachs cannot be held liable for the lack of economic success or for damage or claims by third parties resulting from use of the LICENSED OBJECTS by the user, with the exception of claims arising from infringement of protection rights of third parties concerning the use of the LICENSED OBJECTS.
3. The limitations on liability as per paragraphs 1 and 2 do not apply if, in cases of intent or wanton negligence or lack of warranted quality, liability is absolutely necessary. In such a case, the liability of Messtechnik Sachs is limited to the damage recognisable by Messtechnik Sachs when the specific circumstances are made known.

V. Safety guidelines/documentation
Guarantee and liability claims in conformity with the regulations mentioned above (items III and IV) can only be made if the user has observed the safety guidelines of the documentation in conjunction with use of the machine and its safety guidelines or the terms and conditions of use of the software. The user is responsible for ensuring that the electronic documentation, which is not supplied with the product, matches the product actually used by the user.

1.3.2 Warning notice system

This users manual contains notes, which you must observe to ensure your personal safety, as well as to protect the product and connected equipment. Notes related to your personal safety are highlighted by a yellow exclamation mark. Notes for property / material damage are without an exclamation mark. These notices are graded according to the degree of danger.

	Danger
	indicates the immediate threat of danger. If it is not avoided, it will result in death or serious injury.

	Warning
	indicates a possibly dangerous situation. If it is not avoided, it may result in death or serious injury.

	Caution
	indicates a possibly dangerous situation. If it is not avoided, it may result in injury.

Attention
indicates a possibly harmful situation. If it is not avoided, the product or related / surrounding equipment may be damaged.

1.3.3 Qualified personnel

The product system described in this documentation must only be handled by qualified personal according to the given scope of work. All documentation relevant for the scope of work must be observed, especially the safety and warning notes. Due to its education and experience, qualified personal is able to identify risks and possible dangers when using this products / systems.

1.3.4 Disclaimer

The content of this documentation has been carefully reviewed to comply with the documented hard- and software. We can, however, not exclude discrepancies and do therefore not accept any liability for the exact compliance. This documentation is reviewed regularly. Corrections may be contained in newer versions.

1.4 Preface

	Warning
	Carefully read this complete users manual and all related documentation before setup and use of the Irinos-System. This applies especially to the safety instruction. Misuse may lead do death, serious injury, injury or damage of man, equipment or machine.

1.4.1 Purpose

This users manual contains all relevant information for setup, use and maintenance of the Irinos-System. Target groups are users and service technicians, who setup the product or who perform system diagnostics.

1.4.2 Scope of this users manual

This users manual is valid for the industrial measurement system Irinos and the options, which are descibed in the component overview32.

1.4.3 Intended use

Irinos is a flexible High-Speed measurement system for the industrial production measurement technology.

The measurement device is not appropriate for use in medical fields or in explosive areas, for aerospace and for home- or office use. Other fields of application, which are not mentioned but similar, are also excluded from use.

In safety critical areas, the safety in operation must be ensured by external equipment (e.g. external emergency stop).

Please note:

	Warning
	Products from Messtechnik Sachs GmbH must only be used for applications, which are mentioned in the dataheet or in the related documentation. If third party products are used, these must be recommended or permitted by Messtechnik Sachs GmbH. Proper and safe operation of the products require appropriate transportation, storage, mounting, usage and maintenance. Environmental conditions stated in the specification must be observed as well as notes in the related documentation.

1.4.4 Required knowledge

For the mechanical integration and mounting, solid knowledge and skills in mechanics and machinery are required.

For the electrical installation and the setup, solid knowledge and skills in electrics and electrical safety are required.

For the setup of the measurement application, profund knowledge in industrial measurement technology is required as well as PC skills.

1.4.5 Further documentation

Please note the short booklet, which is delivered with each Irinos module. This applies especially to the safety warnings, which are mentioned in it. The specifications of the Irinos-Boxes can be found in the respective datasheet.

For developers using the communication library (MscDll.dll), a reference guide is available.

1.4.6 Firmware version

This users manual is related to Firmware version V1.1.

Safety instructions

2 Safety instructions

Attention
Damage by opening the device Do not open the Irinos components. They are designed for use without the necessity to open them. The measurement box and/or the measurement system may be damaged. Malfunction or destruction are possible results. Opening the Irinos components will void the warranty.

Attention
Unintended operating situation High frequency radiation, e.g. from a mobile phone, can interrupt the device operation and may lead to malfunction of the Irinos-System. People or material can be injured or damaged. Avoid high frequency radiation: o Do not place sources of radiation next to the Irinos measurement system. o Turn off devices, which are a source of radiation. o Reduce the radio performance of radiation emitting devices. Ensure the compliance regarding electromagnetic compatibility.

Warning

Electric shock

An insufficient earth grounding and/or electrical separation from the mains can lead to injury or damage of people or machine.

Please note:

- Only use PELV supply circuits according to IEC60204-1 (Protective Extra-Low Voltage, PELV).
- Observe the additional requirements for PELV supply circuits according to IEC60204-1.

Only use power supplies, which allow a safe separation of the operating voltage and the load voltage according to IEC60204-1.

Warning

Danger at unprotected machinery

For the operation of machinery, the following has to be observed:

At unprotected machinery danger may exist according to the results of a risk analysis. This danger may lead to personal injury.

Personal injury can be avoided according to a risk analysis by the following actions:

- Additional protection equipment at the machinery. Thereby especially programming, parameterization and wiring of the peripherals must comply with the safety performance (SIL, PL or Cat.), which has been assessed in a risk analysis.
- Appropriate use of the Irinos-System, which is verified by a functional test at the machinery. This allows identifying programming, parameterization or wiring mistakes.
- Documentation of the test results in the relevant safety documentation.

Attention

Electrostatic sensitive devices

An Irinos-Box contains electrostatic sensitive devices. It is possible that electrostatically sensitive equipment is destroyed by energies and voltages that are far less than the human threshold of perception.

Do not open the Irinos-Box. Thereby you avoid touching the sensitive devices.

Attention
Damage of the Irinos-System by transport and storage If the Irinos-System is transported or stored without packaging, shocks, vibrations, pressure or moisture may harm the Irinos components. Damaged packaging signals, that environmental conditions have already affected the Irinos components. The Irinos components and/or the Irinos-System may be damaged. Do not dispose the original packaging. Proper packaging is required during transport and storage.

Attention
Damage due to condensation If the Irinos components or the Irinos-System are exposed to low temperatures or high temperature changes, moisture may cover the Irinos-System or the components. Moisture leads to short circuit and damages the Irinos-System. To avoid damage, please observe the following advices: o Wait before use, until the temperature of the Irinos-System has adjusted to the surrounding temperature. o Avoid direct heat radiation next to the Irinos-System. o If moisture is present, wait until the Irinos-System has completely dried (ca. 8 hours).

Attention
Environmental conditions and chemical resistance Environments, which are not appropriate for the Irinos-System, may lead to malfunction. Chemical substances (e.g. cleaning agent) can change the colour, form or structure of the device. The Irinos-components may be damaged. This may result in malfunction. Please note: ○ Only use the Irinos-System in closed rooms. ○ Only use the Irinos-System according to the environmental conditions given in the specifications. ○ Protect the Irinos-System against dust, moisture and heat. ○ Do not place the Irinos-System into direct sunlight or other strong sources of light. ○ Without additional actions, the Irinos-System must not be used in surroundings where caustic vapours or gases are used. ○ Only use appropriate cleaning agents. Any and all warranty or liability claims are excluded if these regulations are violated. Inappropriate cleaning agents may damage the device. Only use washing-up liquid for cleaning. Do not use: ○ Aggressive solvents and abrasive cleaner ○ Steam jet ○ Compressed air ○ Vacuum cleaner

	Caution
	Unexpected / unintended reaction while cleaning the Irinos-System If the Irinos-System is in operation while cleaning, this may result in unintended actions. This may lead to personal injury or damage at machinery. Always turn off the Irinos-System before cleaning.

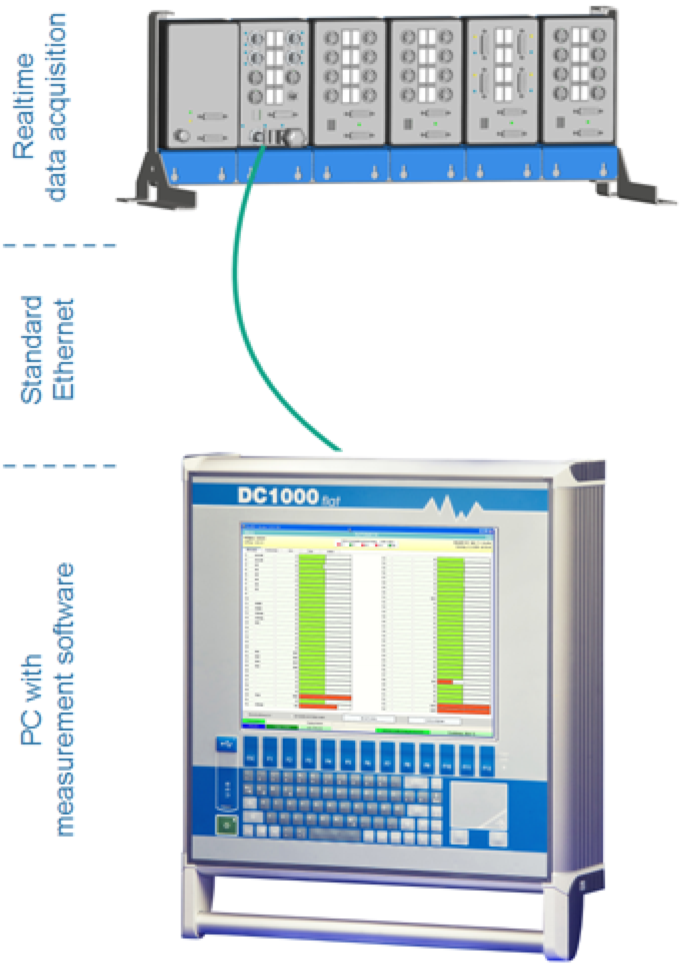
System overview

3 System overview

3.1 System concept

As part of an industrial measurement system, the Irinos-System covers all the functionality, which is time-critical and hardware dependant.

Measurement data of various probe and sensor types is acquired and stored with high speed and high accuracy synchronously in realtime. It is then transferred via a standard ethernet connection to the PC.



Irinos System Concept

An important characteristic of the system is its independency of the time-behaviour of the PC. Concentricity and form measurement are possible with almost all standard PCs, even Laptops. No special hardware or realtime functionality is required. Reading measurement data from the Irinos-System and exchanging control data is done via a Windows DLL 130, which can be accessed by the Windows 32 Bit API. This DLL runs with all current Windows versions (Windows XP, Vista, 7, 8, 10).

Measurement-PC and measurement-software are available from Messtechnik Sachs. Together with the Irinos-System, they build a complete measurement system. Because of the open concept, the Irinos-System can also be combined with measurement PCs of various other vendors. Alternatively the DLL can be integrated into customer specific software.

3.2 Modularity

Due to its flexible design, the Irinos-System can be used for a wide range of measurement applications. This reaches from small SPC systems up to complex measurement machines with a lot of measurement probes.

The Irinos-System consits of at least one and a maximum of 32 Irinos-Boxes. Each Irinos-Box has a fixed number of measurement channels (e.g. 4 or 8). These Irinos-Boxes are connected to each other in line-topology via the ILink interface, i.e. two Irinos-Boxes are always connected by an ILink cable. The maximum distance allowed between the first and the last Irinos-Box is 20m.

The ILink interface combines three jobs at once:

a) Data exchange between the Irinos-Boxes.

b) Time-synchronisation 28 of all measurement channels.

c) Power supply of the Irinos-Boxes.

Irinos-Boxes with different types of measurement inputs can be combined without restrictions.

For the PC respectively the measurement software, the number of Irinos-Boxes is irrelevant. It always "sees" one Irinos-System, whose number of measurement channels is determined by the Irinos-Boxes available.
The measurement values are transferred automatically between the Irinos-Boxes via the ILink interface. Via the Ethernet interface they are transferred as a coherent block to the PC.

3.3 Synchronisation

All Irinos-Boxes have the same system time, called "ILink-Time" (Unit: µs). Time-deviations between different Irinos-Boxes, which cannot be completely avoided, are constantly eliminated. The end user does neither have to parametrize nor to check this. In practice these deviations are less than 200 ns.

The data acquisition of all Irinos-Boxes is started simultaneously based on the ILink-Time. The measurement values are stored in the internal memory of the Irinos-Box and then transferred to the Master-Box. Here they are merged to a single block of measurement values. This allows a fast simultaneous data acquisition of all channels.

The realtime capability is independent of the number of measurement channels, because each Irinos-Box has its own buffer for measurement values.

3.4 Master vs. integrated Master vs. Slave

Each Irinos-System has exactly one "Master-Box". Ethernet connection to the PC is made via this box. All other (optional) Irinos-Boxes are called "Slave-Box".

Each Irinos-System has exactly one Master-Box. Multiple Master-Boxes are not allowed.

A Master-Box has an Ethernet interface (connector M12, waterblue) and an ILink interface (2x DSUB 15-pin).

A Slave-Box only has an ILink interface.

Regarding the Master-Box, a difference has to be made between a separate Master-Box an a so-called "integrated Master".

The purpose of a separate Master-Box (IR-MASTER) mainly is the communication to the PC. Depending on the variant, it has digital in- and outputs and analogue inputs. Typically it is used together with at least one Slave-Box.

The integrated Master is a measurement Box with integrated

Ethernet interface. If only a few measurement channels are used, it can be the only box used in an Irinos-System.

For the PC it does not matter, which type is used.

No general advice can be given, which type to use. This depends on the measurement application and perhaps also on the requirements made by the end customer. The following table lists the most important advantages of each type:

Master	**Integrated Master**
o Better maintenance: all measurement boxes of the same input type are Slave-Boxes. Thereby they are identical -> less stock of replacement boxes due to less variants. o Variant available with 4 digital in- and outputs and 3 analogue inputs ±10V. o Connection of additional digital in- and outputs via IO-Bus. o Higher data throughput to the PC with fast measurements and many measurement channels. o 7-digit display for easier diagnostics	o No additional costs for separate Master. o 2 digital inputs, e.g. for push button or foot pedal.

3.5 Digital in-/outputs (Bit-I/O)

The Irinos-System can be enhanced by digital in- and outputs. These are intended to connect push buttons, foot pedals, proximity switches, lamps, valves, etc. to the measurement software. Further they can be used for data exchange between a PLC and the measurement software (e.g. start measurement, stop measurement, result ok, ...). Alternatively a ProfiNet module is available.

The speed of Bit I/O is limited in two ways:

- It depends on the speed of the measurement PC. Usually it is not limited by the PC-hardware, but by the realtime capabilities of the operating system.
- It depends on the communication speed between the Irinos-Boxes and between the Irinos-System and the PC.

Please note:

> The digital in- and outputs of the Irinos-System are not intended to replace a large PC and they have no realtime-capabilities.
>
> Although it is possible to control a measurement device of even a small machine, this must be tested for each application.

Two systems are available for digital in- and outputs:

a) The digital in- and outputs are directly integrated into the Irinos-Boxes. This is for example the case with the Irinos-Boxes IR-MASTER-KB1 [47] or IR-DIO [58].

b) Separate I/O-Boxes [68] are connected via the IO-Bus. Connection of the IO-Bus is made via the Irinos-Box IR-MASTER [47] .

Both systems can be combined.

3.6 Power supply

All Irinos-Boxes have a common 24 V power supply. This supply is connected to the Irinos-System via the ILink interface. The supply is forwarded to all other Irinos-Boxes via the ILink cabling.

Exceptions are:

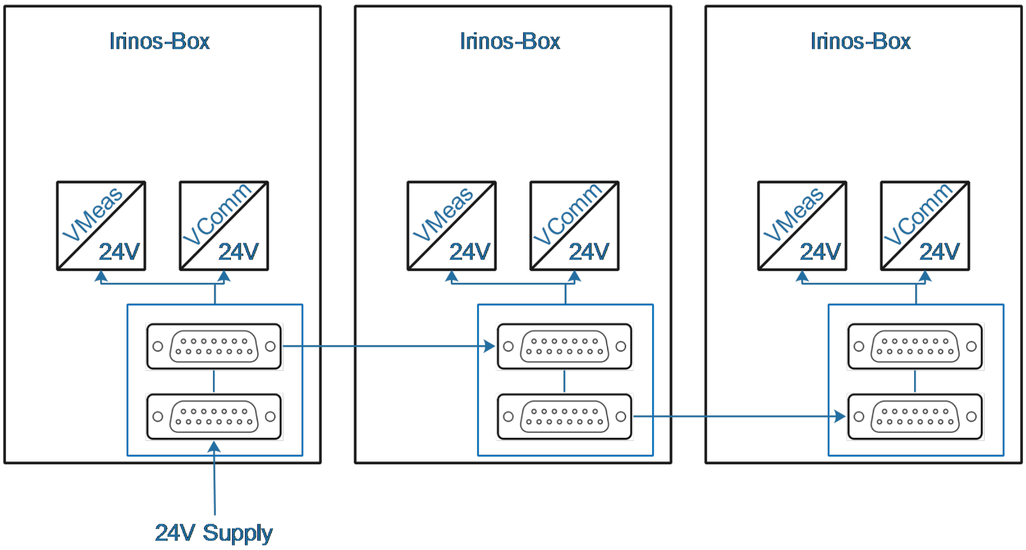
- The Irinos-Boxes for digital in- and outputs (IR-DIO [58]-...-EXTP-...): an external 24 V power supply is required for the in- and outputs.
- I/O-Boxes connected via the IO-Bus. Depending in their type, some have an integrated power supply, some require a separate

one.

An Irinos power supply IR-PU 67 is used to supply the Irinos-Boxes.

All internal supply voltages are generated via DC/DC convertes with galvanic isolation. For analogue voltages additional linear regulators are used (see figure). As a result, all internal power supplies of different Irinos-Boxes are separated. This increases the noise immunity, which is a precondition for precise measurement results. In addition, ground loops are avoided.

Please note that in any case a stable 24 V power supply with low noise in required for a trouble-free operation of the Irinos-System.



Irinos power supply

3.7 Mounting options

In order to provide the best mounting option for different installation locations, various mounting options are available:

- Via 2 tapped bushed M4 on the rear side, each Irinos-Box can be directly mounted 104 .
  For this mounting option, no additional mounting material is necessary. The dimensions can be found in the specifications section.
  It should be possible to access the rear side in order to allow a quick exchange of an Irinos-Box.
- Via the frontside mounting 106 kit IR-MFFM-1, each Irinos-Box can be accessed from the front side, e.g. for placement into a control cabinet.

- Via the hat rail / din rail mounting[105] kit IR-MHRM-1, each Irinos-Box can be mounted on a standard hat rail / din rail.
- Via the mounting kit IR-MITEM-40, each Irinos-Box can be directly mounted onto a 40x40mm aluminium profile[107] of the manufacturer Item (or compatible).

All mounting options can be mixed up. I. e. it is for example possible to fix an Irinos-Box IR-MASTER[47] into a control cabinet via the frontside mounting kit IR-MFFM-1, while an Irinos-Box of the type IR-TFV[50] is mounted next to the measurement device via the rear side tapped bushes.

Via the stand IR-MWIP-40 together with a 40x40mm Item profile and a mounting kit IR-MITEM-40, a mounting stand[109] can be build. It allows for example an easy mounting of the Irinos-System into a table.

The width of the stand depends on the length of the Item profile, which allows for an easy adaption on the number of Irinos-Boxes used.

3.8 Overview of components

The following tables provide an overview about all components available for the Irinos-System.

The Irinos-System is continuously expanded. All available components are listed in the newest version of this users manual.

Overview Irinos-Boxes

Order-No.	**Description**	**Type**	**In- / outputs**
IR-MASTER[47] for the ethernet connection to the PC			
828-5000	**IR-MASTER-**KB1-68-68-3-SYSP-ETHIL	Master	○ IO-Bus via M8 ○ 4 digital inputs via 2x M12 ○ 4 digital outputs via 2x M12

Order-No.	Description	Type	In- / outputs
			○ 3 analogue inputs ±10V via 3x M16 7-pin
828-5001	**IR-MASTER**-IOB-64-64-0-SYSP-ETHIL	Master	IO-Bus via M8
IR-TFV 50 **for inductive probes**			
828-5002	**IR-TFV**-8-TESA-M16-ETHIL	Integrated Master	○ 8 inputs for probes **Tesa halfbridge** ○ 2 digital inputs via 1x M12, e.g. for push buttons
828-5003	**IR-TFV**-8-TESA-M16-IL	Slave	8 inputs for probes **Tesa halfbridge**
828-5004	**IR-TFV**-8-TESA-KF27-IL	Slave	8 inputs for probes **Tesa halfbridge**, indirectly connected via connection box
828-5005	**IR-TFV**-8-TESA-M16IP-IL	Slave	8 inputs for probes **Tesa halfbridge** with connectors **IP65**
828-5006	**IR-TFV**-8-IET-M16-ETHIL	Integrated Master	○ 8 inputs for probes **Knäbel IET** ○ 2 digital inputs via 1x M12, e.g. for push buttons
828-5007	**IR-TFV**-8-IET-M16-IL	Slave	8 inputs for probes **Knäbel IET**
828-5008	**IR-TFV**-8-IET-KF27-IL	Slave	8 inputs for probes **Knäbel IET**, indirectly connected

Order-No.	Description	Type	In- / outputs
			via connection box
828-5009	**IR-TFV**-8-IET-M16IP-IL	Slave	8 inputs for probes **Knäbel IET** with connectors **IP65**
IR-AIN 54 **with analogue inputs ±10V**			
828-5010	**IR-AIN**-8-D10-M16-ETHIL	Integrated Master	○ 8 analogue inputs ±10V, each via M16 7-pin ○ 2 digital inputs via 1x M12, e.g. for push buttons
828-5011	**IR-AIN**-8-D10-M16-IL	Slave	8 analogue inputs ±10V, each via M16 7-pin
828-5012	**IR-AIN**-8-D10-M16IP-IL	Slave	○ 8 analogue inputs ±10V, each via M16 7-pin IP65
IR-INC 56 **for incremental encoders 1Vss or TTL/RS422**			
828-5013	**IR-INC**-4-SEL1VSS-DSUB15F-ETHIL	Integrated Master	○ 4 inputs for incremental encoders 1Vss or TTL/ RS422 via DSUB 15-pin: **pre-configured for 1Vss** ○ 2 digital inputs via 1x M12, e.g. for push bottons
828-5014	**IR-INC**-4-SEL1VSS-DSUB15F-IL	Slave	4 inputs for incremental encoders 1Vss or TTL/ RS422 via DSUB 15-pin: **pre-configured for 1Vss**
828-5015	**IR-INC**-4-SELTTL-DSUB15F-ETHIL	Integrated Master	○ 4 inputs for incremental encoders 1Vss or TTL/ RS422 via DSUB 15-pin: **pre-configured for TTL/**

Order-No.	Description	Type	In- / outputs
			RS422 ○ 2 digital inputs via 1x M12, e.g. for push bottons
828-5016	**IR-INC**-4-SELTTL-DSUB15F-IL	Slave	4 inputs for incremental encoders 1Vss or TTL/ RS422 via DSUB 15-pin: **pre-configured for TTL/ RS422**
IR-DIO 58 **with digital in- and outputs**			
828-5019	**IR-DIO**-16-16-M23-EXTP-IL	Slave	○ 16 digital inputs via 2 connectors M23 ○ 16 digital outputs via 2 connectors M23 ○ **Separate power supply** (for high output power of digital I/Os)
828-5022	**IR-DIO**-16-16-M23-SYSP-IL	Slave	○ 16 digital inputs via 2 connectors M23 ○ 16 digital outputs via 2 connectors M23 ○ **Power supply via ILink** (for low output power)
828-5020	**IR-DIO**-16-16-D37-EXTP-IL	Slave	16 digital inputs and 16 digital outputs via DSUB 37-pin Terminal block connection via 828-5021
828-	**Terminal block**		Terminal block module for

Order-No.	Description	Type	In- / outputs
5021	for IR-DIO-16-16-D37-IL		828-5020
IR-HMI1[61] **control panel**			
828-5029	**IR-HMI1**-6-NA-MPA-0-0-IL	Slave	o 6 pcs. 22.5mm push buttons o Selection of Machine-No. (1-99) o Selection of Testplan-No. (1-99) o 7 membrane keys: 4 arrows + OK + # + ESC
IR-PU[67] **power supply**			
828-5017	**IR-PU**-50-HWS-F		Industrial power supply 50 watt for Irinos-System, European connector (type F)
828-5018	**IR-PU**-12-STK-C		Wall power supply, 12 watt for Irinos-System, European connector (type C)

Overview accessories mounting and labelling

Order-No.	Description
Mounting	
828-5041	IR-MHRM-1 Mounting kit for hat rail / DIN rail

Order-No.	Description
828-5042	IR-MFFM-1 Frontside mounting kit
828-5043	IR-MITEM-40 Mounting kit for 40mm aluminium profile Item or similar
828-5044	IR-MWIP-40 Stand for 40mm aluminium profile Item or similar (Does not contain Item profile and mounting kit IR-MITEM-40).
Labelling	
828-5040	IR-MIPL-8-ABB179 Labelling carrier for 8 plastic labels type Murrplastik ABB 17x9 (Order-No. Murrplastik: 86421020).

Overview Irinos cabling

Order-No.	Description	Length [m]
ILink-cable IP40 for the connection between two Irinos-Boxes		
828-5055	IR-ILINK-002-IP40 ILink cable	0,2
828-5056	IR-ILINK-010-IP40 ILink cable	1
828-5057	IR-ILINK-020-IP40 ILink cable	2
828-5058	IR-ILINK-030-IP40 ILink cable	3

Order-No.	Description	Length [m]
828-5059	IR-ILINK-050-IP40 ILink cable	5
828-5060	IR-ILINK-100-IP40 ILink cable	10
ILink-Kabel IP65 for the connection between two Irinos-Boxes		
828-5061	IR-ILINK-002-IP65 ILink cable	0,2
828-5062	IR-ILINK-010-IP65 ILink cable	1
828-5063	IR-ILINK-020-IP65 ILink cable	2
828-5064	IR-ILINK-030-IP65 ILink cable	3
828-5065	IR-ILINK-050-IP65 ILink cable	5
828-5066	IR-ILINK-100-IP65 ILink cable	10
Ethernet cable M12 RJ45 for the connection of the Irinos-System to a standard RJ45 network port		
828-5050	IR-CETH-RJ45-M12-010 Ethernet cable	1
828-5051	IR-CETH-RJ45-M12-020 Ethernet cable	2
828-5052	IR-CETH-RJ45-M12-050 Ethernet cable	5

Order-No.	Description	Length [m]
828-5053	IR-CETH-RJ45-M12-100 Ethernet cable	10
828-5054	IR-CETH-RJ45-M12-150 Ethernet cable	15
Connection cable for quick-change system for IR-TFV 50		
828-5067	K8F27-D25-030	3
828-5068	K8F27-D25-050	5
828-5069	K8F27-D25-070	7
828-5070	K8F27-D25-100	10
Connection cable for Irinos IO-Bus, 2xM8 circular connector		
828-8013	Connection cable for Irinos IO-Bus with 2xM8 circular connector	1
828-8010	Connection cable for Irinos IO-Bus with 2xM8 circular connector	3
828-8011	Connection cable for Irinos IO-Bus with 2xM8 circular connector	5
Connection cable for Irinos IO-Bus, 1xM8 circular connector, 1xDSUB 9-pin		
828-8012	Connection cable for Irinos IO-Bus with 1xM8 circular connector and 1x DSUB 9-pin	3

Overview of connection boxes for quick-change system for IR-TFV

Order-No.	Description
Connection boxes for Tesa halfbridge (to be used with IR-TFV-8-TESA-KF27-IL)	
820-2212	Connection box AB8F27 Tesa for indirect connection of 8 inductive probes Tesa halfbridge and compatible
820-2210	Connection box AB4F27 Tesa for indirect connection of 4 inductive probes Tesa halfbridge and compatible
Connection boxes for Knäbel IET (to be used with IR-TFV-8-IET-KF27-IL)	
820-2232	Connection box AB8F27 IET for indirect connection of 8 inductive probes Knäbel IET
820-2230	Connection box AB4F27 IET for indirect connection of 4 inductive probes Knäbel IET

Overview I/O-Boxes for Irinos IO-Bus

Order-No.	Description
I/O-Boxes, desktop 68 **version with integrated power supply, circular connector M8**	
828-1832	I/O-Box with 8 digitale in- and outputs, connection via 2 pcs M23 connector 16-pin, 50W power supply 115 / 230 V AC integrated.
828-1834	I/O-Box with 16 digitale in- and outputs, connection via42 pcs M23 connector 16-pin, 50W power supply 115 / 230 V AC integrated.
I/O-Boxen for 70 **hat rail / din rail mounting, DSUB 9-pin**	
828-1812	I/O-Module with 8 digital in- and outputs for hat rail / din rail mounting

Order-No.	Description
828-1814	I/O-Module with 16 digital in- and outputs for hat rail / din rail mounting

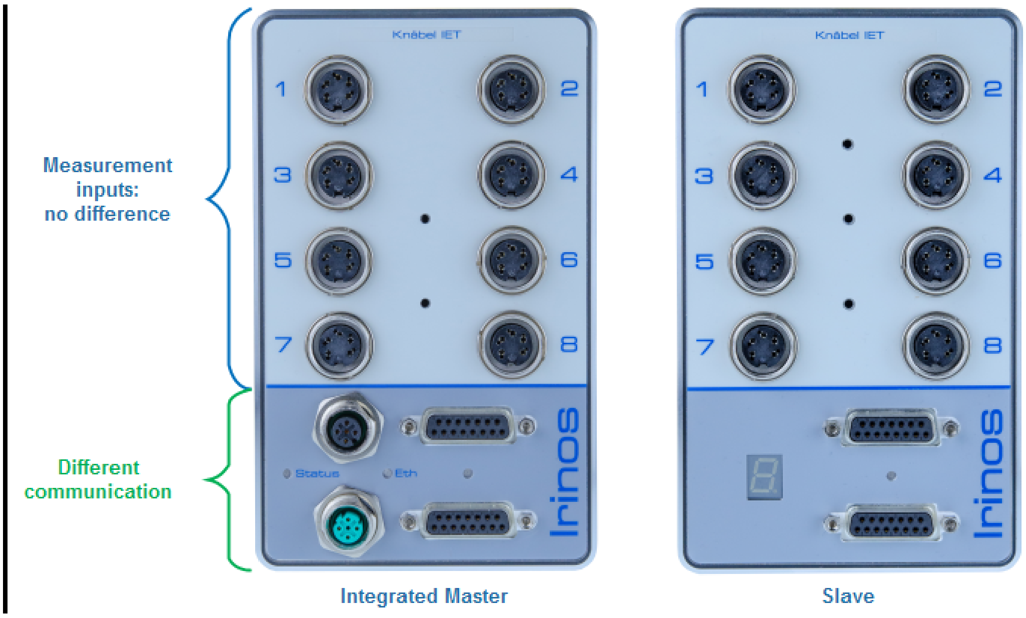
Product descriptions

4 Product descriptions

These product descriptions give an overview about the different Irinos-Boxes and various accessories. Pinning information and technical details to the connectors are available in the chapter "Connectors".
All article numbers can be found in the component overview.

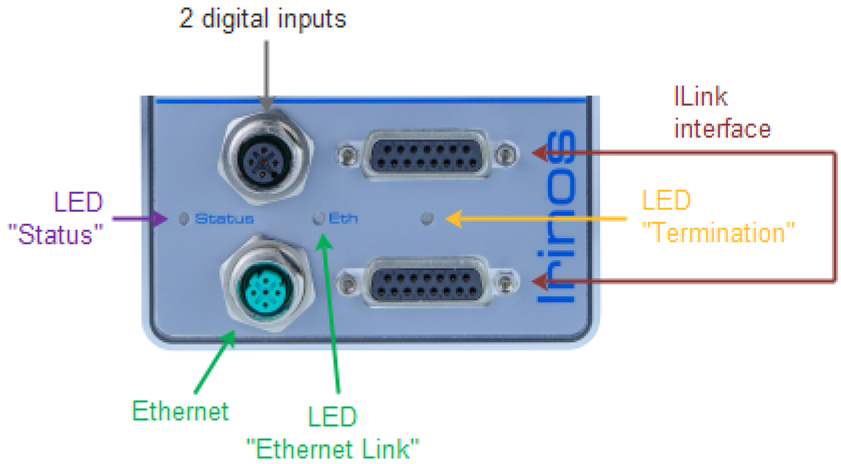
4.1 Common

Some Irinos-Boxes are available in two different types, "integrated Master" and "Slave". Their only difference is the communicaton section. The measurement sections is always identical. The follwing figure shows this with the Irinos-Box IR-TFV 50:



Comparison "integrated Master" vs. "Slave"

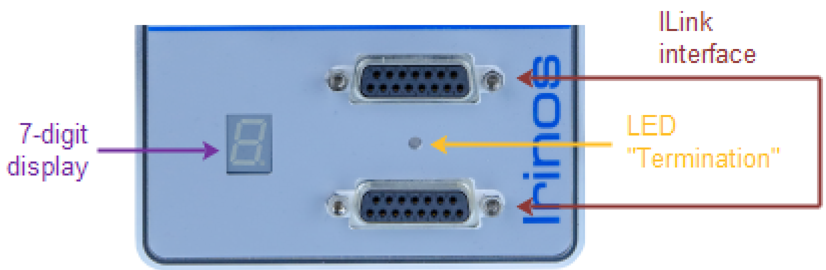
The ***integrated Master*** has a status LED 45 for signalling the system status, the ILink interface 74 for the enhancement by additional Slave-Boxes, an Ethernet interface 76 for the connection to the PC and two digital inputs 77 , e.g. for the connection of a foot pedal or push button:



Connections and signalling elements of the "integrated Master"

The Ethernet status is displayed next to the Ethernet connector: the LED "Eth" is turned on as soon as an Ethernet connection is active. While data is transferred, this LED flashes.

A ***Slave-Box*** has a 7-digit status display [46] and the ILink interface [74] for the connection to the Master-Box and for further enhancement [27] of the Irinos-System by additional Slave-Boxes:



Connections and signalling elements of a "Slave-Box"

4.1.1 Status via LED (integrated Master)

Using an "integrated Master [28] ", the status of the Irinos-Box is displayed via a status LED:

LED „Status"	Status
Flashing slowly (< 1 Hz)	Everything ok
Flashing fast (4 Hz)	Sine oscillator short circuit (only IR-TFV[50])
Always on	Event[180] active (e.g. error)

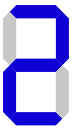
The status is related only to the Master-Box and not to the whole Irinos-System.

It can be read out by the PC via the MscDll.

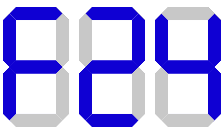
4.1.2 Status via 7-digit display (Master & Slave)

The current status of the Irinos-Box is displayed via the 7-digit display.

Typically the 7-digit display shows the Box-number of the Irinos-Box. It is assigned at startup[118] by the Master-Box[28]. In case the Box-number has only one character (which is most times the case), the dot of the 7-digit display flashes:



As soon as an event[180] occures, the character "F" followed by the event number is displayed. E.g. F24 for overload of the incremental encoder power supply of the Irinos-Box IR-INC[56]. This status can also be read out by the PC via the MscDll.



After startup, the current IP network information[120] of the Irinos system is displayed once on the first available 7-digit display. If a Master IR-MASTER[47] is used, it will be displayed on this box. If an integrated Master is used, it will be displayed on the first Slave-Box.

If the DHCP-Server is active, only "dHCP" will be displayed. Otherwise the IP address followed by the subnet mask will be displayed.

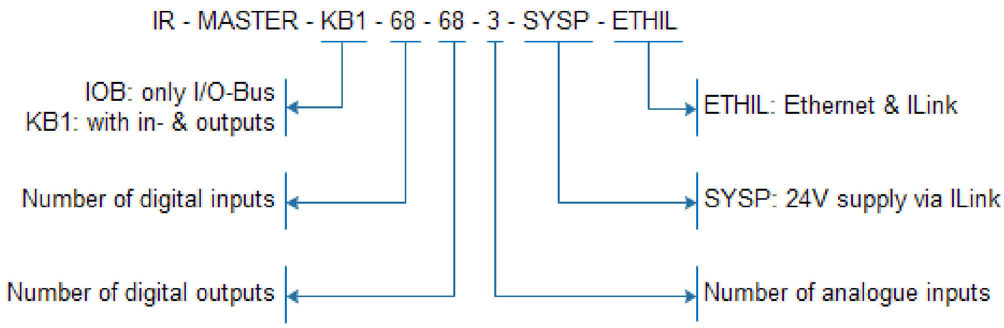
[Note: If an integrated Master is used without a Slave-Box, the IP information cannot be displayed.]

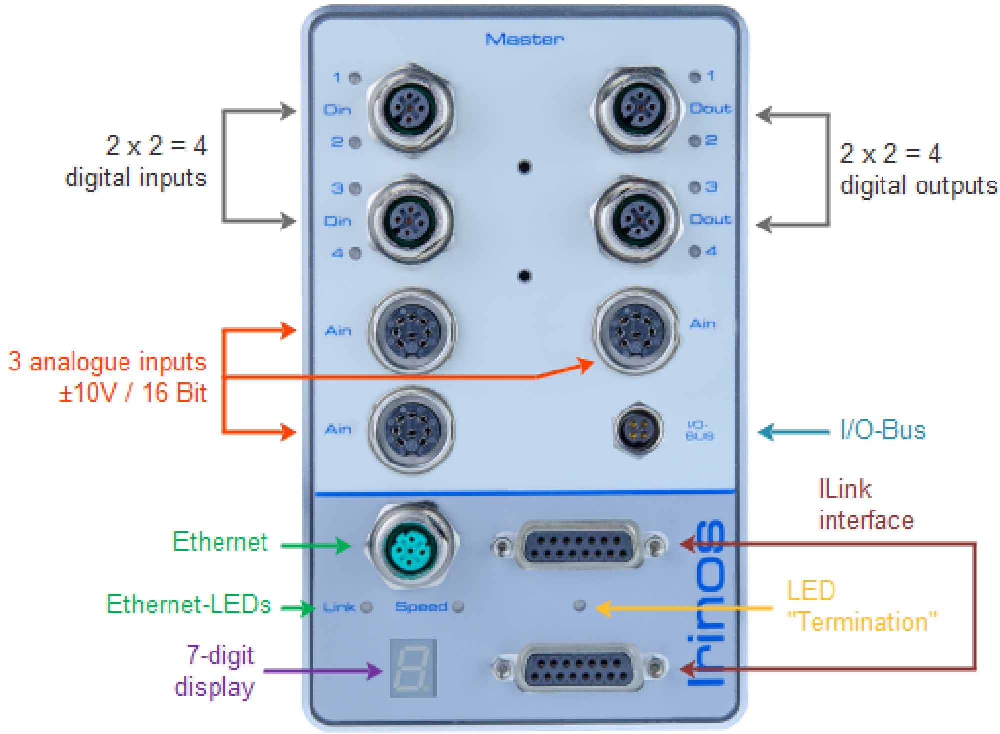
4.2 IR-MASTER for the communication with the PC

The main task of the Irinos-Box IR-MASTER is the communication with the PC. It is available in two types:

- Type **IR-MASTER-IOB**-64-64-0-SYSP-ETHIL has a connector for the IO-Bus. It allows the connection of external I/O-Boxes with digital in- and outputs. These are available as a desktop version or for hat rail / DIN rail mounting.
- Type **IR-MASTER-KB1**-68-68-3-SYSP-ETHIL is additionally equipped with 4 digital inputs, 4 digital outputs and 3 analogue inputs ±10V (see figure).

The definition of the type string is as follows:





Connector overview IR-MASTER-KB1

Connectors / Elements

The Irinos-Box IR-MASTER has the following elements (depending on type):

○ 7-digit display 46

○ ILink interface 74 for the connection of additional Slave-Boxes

○ Ethernet interface 76 for the connection to the PC

○ 2x2 digital inputs 78 and 2x2 digital outputs 79

○ 3 analogue inputs ±10V 86

○ IO-Bus 90 - for the connection of I/O-Boxes

Ethernet LEDs

LED „Speed"	Status
Off	Connection speed 10 MBit/s
On	Connection speed 100 MBit/s

LED „Link"	**Status**
Off	No ethernet connection established
On	Ethernet connection established, no data transfer active
Flashing	Ethernet connection established, data transfer active

IO-Bus

The IO-Bus is internally always handled as 64 digital in- and outputs. I.e. if an Irinos-Box IR-MASTER is used, the first 64 digital inputs are assigned to the IO-Bus. The same applies to the digital outputs. This is also the case, if no I/O-Box is connected to the IO-Bus.

The 4 integrated digital in- and outputs are counted as input / output 65..68.

Each integrated in- and output has an additional Status-LED. It is turned on, if the respective in- or output is at high level.

Analogue inputs

The analogue inputs are ***measured synchronously*** (no multiplexing).

In order to allow a quick exchange of the Irinos-Box by an identical type, all input channels are digitally pre-calibrated to deliver -32.000 digits at -10V and +32.000 digits at +10V.

Due to their high input impedance, the measurement value may float if no input signal is connected. This is not a defect. It is a consequence of the high-class input specificatons.

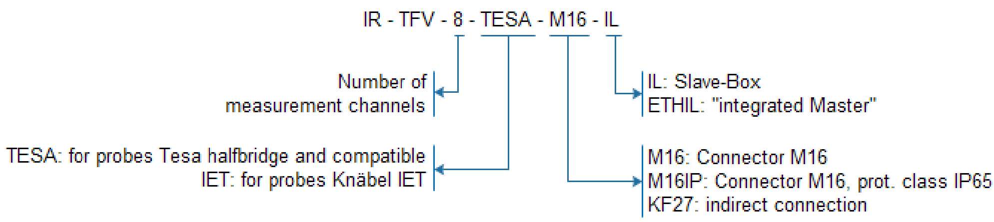
4.3 IR-TFV for inductive probes

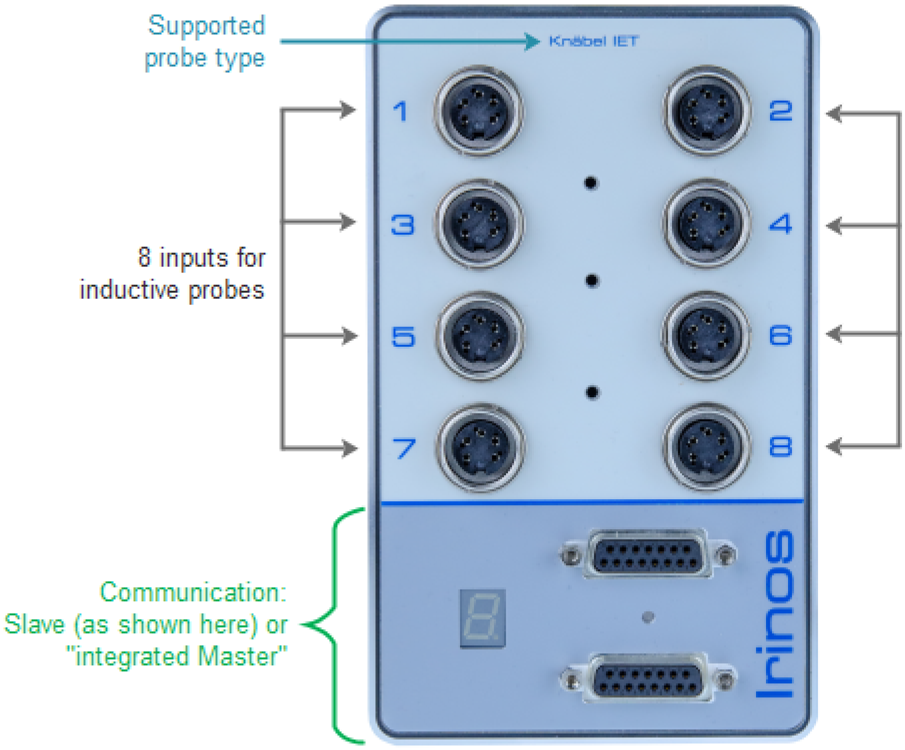
The Irinos-Box IR-TFV is appropriate for the connection of 8 inductive probes. Each Irinos-Box is optimized for a specific probe type. It is written on the front side of the Irinos-Box.

Additionally, the following types are available:

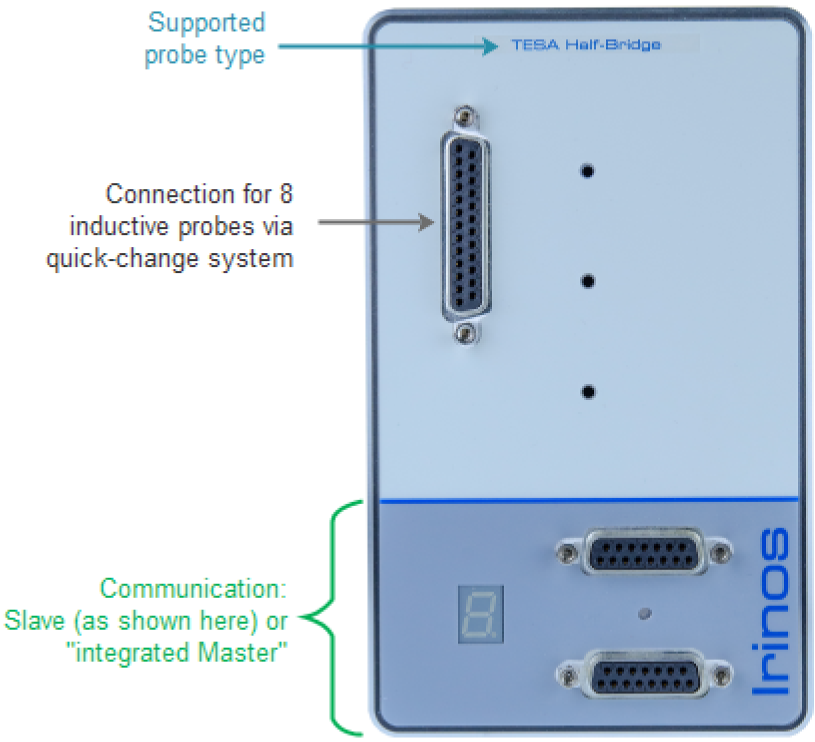
- With the types M16 and M16IP, the probes are directly connected to the Irinos-Box.
  For M16IP, connectors with protection class IP65 are used. Please note that these type only is appropriate, if the probe connectors also support this protection class. This is rarely the case.
- Using the type KF27, the measurement probes are connected via a separate connection box.

The definition of the type string is as follows:





Connectors for "IR-TFV-8-x-M16-x" und "IR-TFV-8-x-M16IP-x"



Connectors for "IR-TFV-8-x-KF27-x"

Irinos-Box IR-TFV with separate quick-change box

Connectors / elements

- 8 inputs for inductive probes, directly connected (M16 [92]) or indirectly connected (KF27 / DSUB25 [93]).
- An explanation of the communication section [44] can be found in the general overview.

Sine oscillator / XSync oscillator synchronisation

Inductive probes require a sine signal. This is generated by the sine oscillator of the Irinos-Box. All measurement channels of an Irinos-Box have a ***common sine-oscillator***. In order to get an accurate measurement value, the sine signal has no DC signal component.

In case of an overload of the sine-oscillator, e.g. when a probe is broken, the sine oscillator is automatically turned off. This event is signalled as followes:

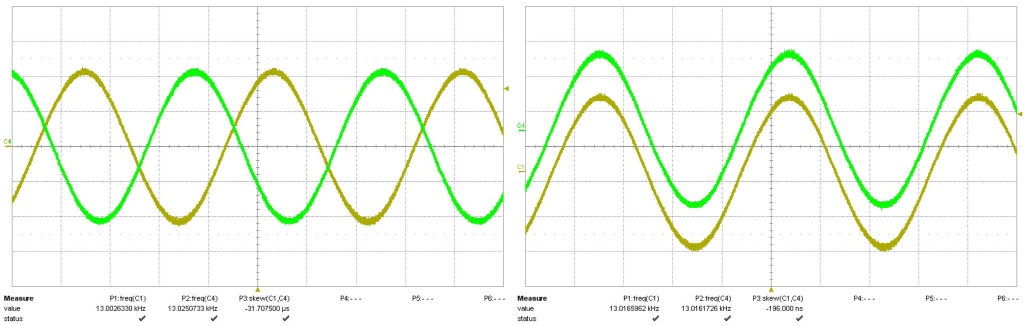
- The Status-LED [45] flashes very fast (integrated Master).
- The text "OSC" is displayed on the 7-digit display [46] (Slave-Box).

○ A default measurement value 166 is used and the hardware status 166 is set.

The Irinos-Box checks cyclically, if the short circuit is removed. If yes, the sine oscillator automatically restarts.

If the cabling situation is difficult, like for example in plug gauges, a phase shift between two sine oscillators could lead to signal interference, which may result in unstable measurement values.

Within an Irinos-System, all sine-oscillators are synchronized 28 over all Irinos-Boxes. This is called the "XSync oscillator synchronisation" (patent pending). Thereby the Irinos system is also appropriate for difficult cabling situations.



IR-TFV ohne / mit XSync Oszillator-Synchronisation

Data acquisition

All measurement channels are ***sampled synchronously*** (no multiplexing).
The whole measurement signal is used (integrating measurement). This method has a much higher noise immunity than the widely used 1-point or 2-point sampling method. The filter, which is used for this method, is adjusted to the mechanical specification of the probe type.

In order to allow a quick exchange of the Irinos-Box by an identical type, all input channels are digitally pre-calibrated. The following table shows the value range:

Digital value	Corresponding stroke for probe type Tesa halfbridge	Corresponding stroke for probe type Knäbel IET
- 32.000	- 2000 µm	- 200 µm
0	0 µm	0 µm
+32.000	+ 2000 µm	+ 200 µm

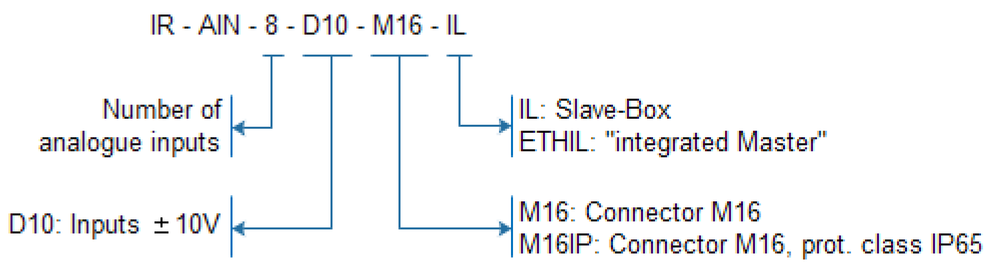
Connector type KF27 93 **for quick-change system**

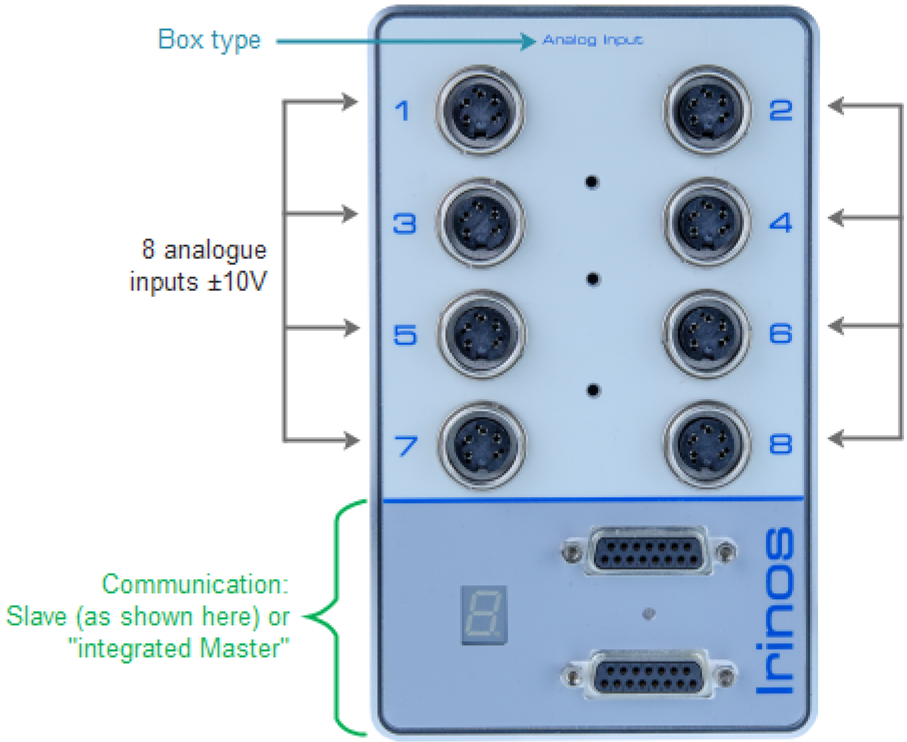
If the quick-change system is used, it is possible to disconnect and connect all 8 channels via a single connector during operation of the system. This allows for example changing the measurement device without restarting the Irinos system.

4.4 IR-AIN with analogue inputs ±10V

The Irinos-Box IR-AIN is appropriate for the connection of 8 analogue signals ±10V. It is available with standard connectors or with IP65 connectors.

The definition of the type string is as follows:





Connectors for "IR-AIN"

Connectors / elements

○ 8 analogue inputs [86] ±10V

○ An explanation of the communication section [44] can be found in the general overview.

Analogue inputs

The analogue inputs are ***measured synchronously*** (no multiplexing).

In order to allow a quick exchange of the Irinos-Box by an identical type, all input channels are digitally pre-calibrated to deliver -32.000 digits at -10V and +32.000 digits at +10V.

Due to their high input impedance, the measurement value may float if no input signal is connected. This is not a defect. It is a consequence of the high-class input specificatons.

4.5 IR-INC for incremental encoders 1Vpp or TTL / RS422

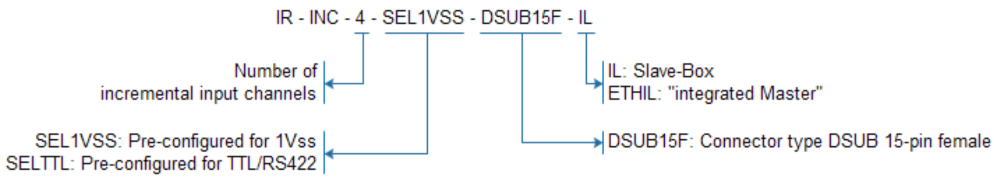
--> Please read the application notes [200] ***for incremental encoders.***

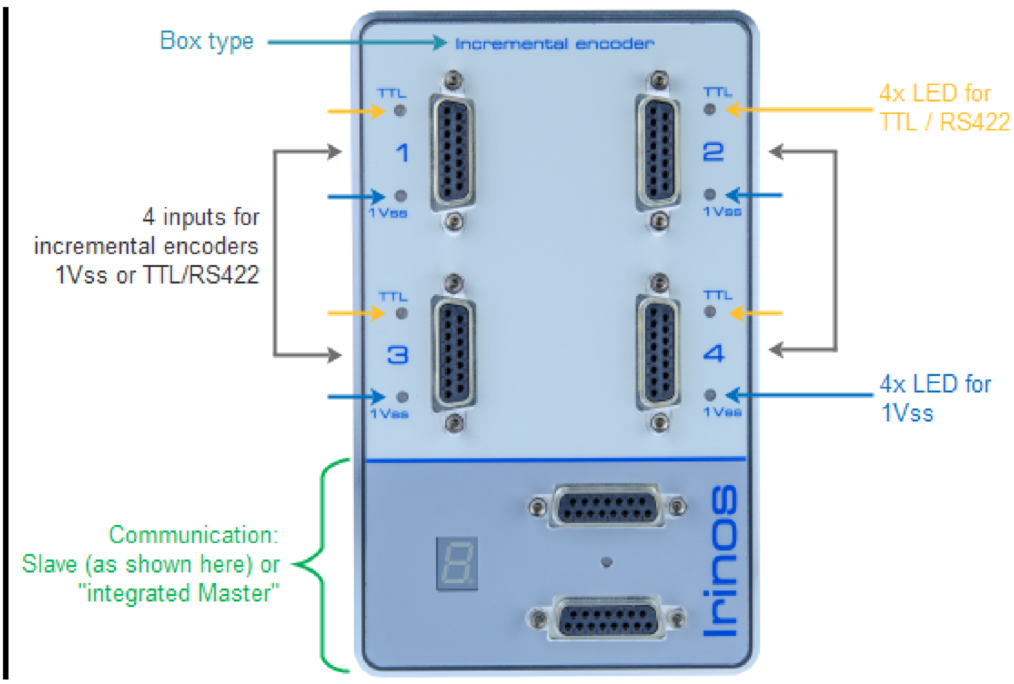
4 incremental encoders can be connected to the Irinos-Box IR-INC.

Each input channel can be set either to 1Vpp or to TTL / RS422. The current input configuration is displayed via LEDs next to the connector. Changing the input type can be done via the Irinos-Tool or via a configuration string by the measurement software.

Depending on the order number, all channels are either set to 1Vpp or TTL/RS422 as factory defaults.

The definition of the type string is as follows:





Connectors for "IR-INC"

Connectors / elements

○ 4 connectors [96] for incremental encoders

○ An explanation of the communication section 44 can be found in the general overview.

> The same connector type is used for the ILink interface 74 and the incremental encoders.
>
> The pinning of the ILink connector has been chosen to avoid any damage of incremental encoders with standard pinning, if they are plugged into the ILink interface. For encoders with special pinning, this may be different.
>
> Only use the appropriate connectors for the incremental encoders.

Leds for incremental encoders

LED	Status
Blue on	Configure for 1Vpp. No error detected.
Blue, flashing	Configured for 1Vpp. An error has occured.
Yellow on	Configured for TTL / RS422.

Power supply

The incremental encoders have a shared 5V power supply. It is designed to provide enough power reserves for all typical applications. Details can be found in the datasheet.

Each channel has its own overload and short circuit protection. If an overload occurs, the power supply of the respective channel is immediately turned off and an error is signalled. As soon as the short circuit has been removed, the power supply is automatically turned on.

An additional protection ensures that the total load is not exceeded. If an overload of the total output power occurs, the power supply for all channels is turned off and an error is signalled. After removing this error, the Irinos system must be restarted.

A power supply error is signalled as follows:

- The Event „F24“ [184] becomes active. Irinos-Boxes with 7-digit display [46] (Slave) show the event "F24".
- For Irinos-Boxes with a status LED [45] (integrated Master), the LED is turned on permanently.
- The corresponding hardware status bit of the respective measurement channel is set. It can be read out by the measurement software.

Data acquisition

All channels are ***sampled synchronously***. It does not matter, whether the input channel is configured for 1Vpp or TTL / RS422.

Error detection

Channels, which are configured for the input type 1Vpp, have a channel-wise error detection of the input signals. If everything is ok, the blue LED is turned on, otherwise it flashes. In parallel, errors can be detected via the following ways:

- The corresponding bit in the hardware status of the measurement channel are set. It can be read out by the measurement software.
- The event "F25" [185] becomes active. Using the standard settings, it is not displayed on the 7-digit display [46]. However, it will be recorded in the diagnostic memory [190].

If an error occurs, the position value still can be read out. It may be valid, but this is not necessarily the case. For a reliable operation of the measurement application, it is required to read the hardware status [166] by the measurement software. In case it contains an error information, appropriate action regarding the application has to be taken.

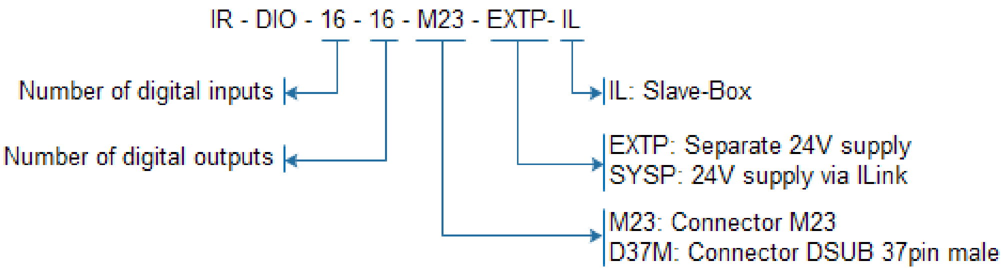
4.6 IR-DIO with digital in- and outputs

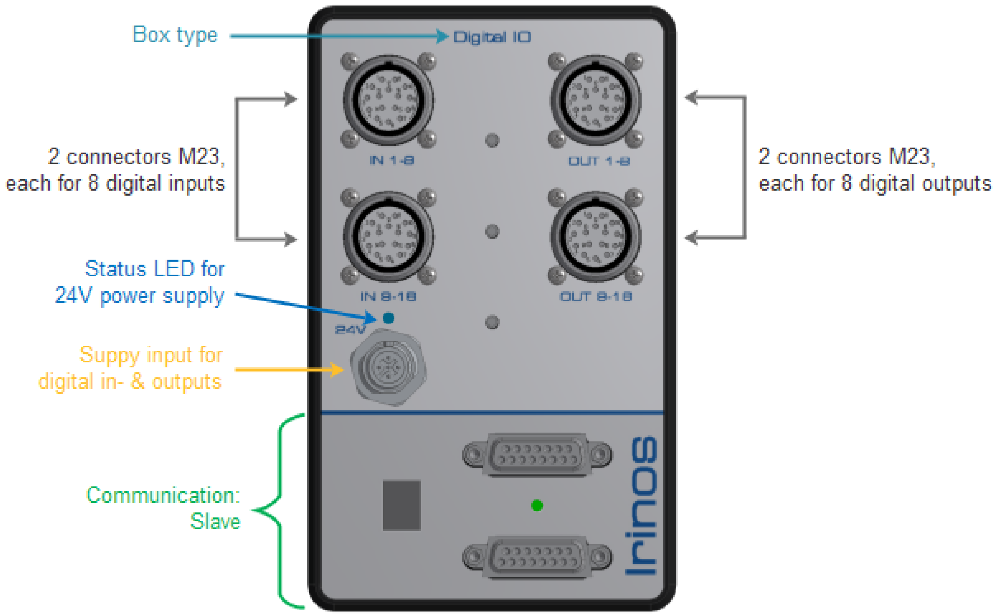
The Irinos-Box IR-DIO provides 16 digital inputs and 16 digital outputs. Three types are available:

- With 4 connectors M23 for the connection of external connection modules. The supply for the in- and outputs must be supplied via a M12 connector.
- With 4 connectors M23 for the connection of external connection modules. The supply for the in- and outputs is taken from the ILink supply power (for low power applications).

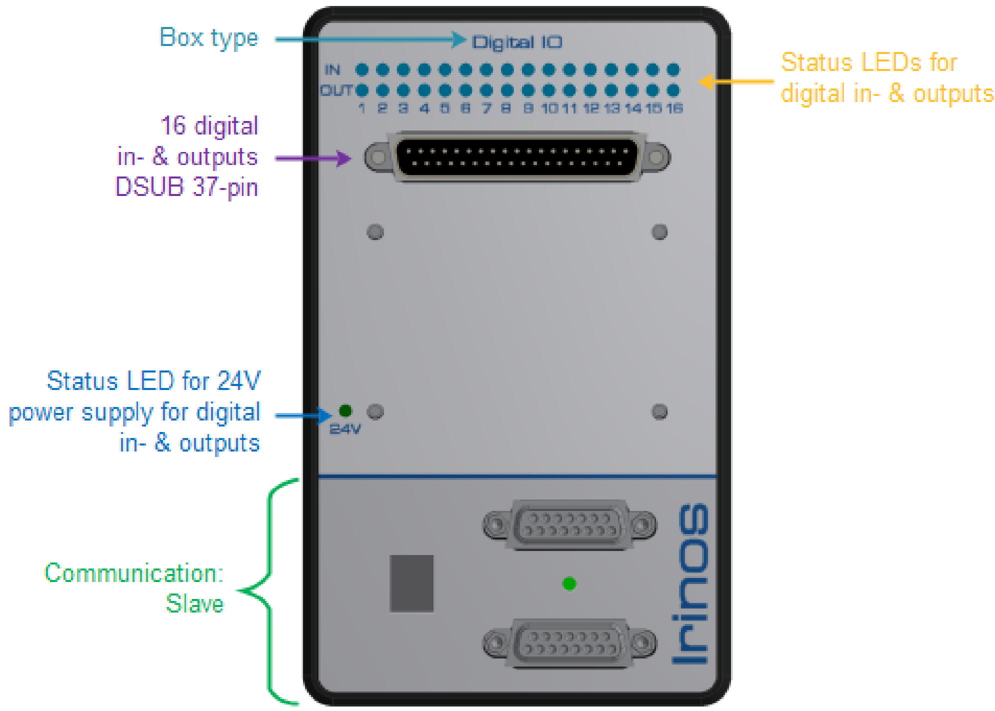
○ With one connector DSUB 37-pin for direct connection.
Via an adapter module (828-5021), the in- and outputs can be accessed via terminal blocks.

The definition of the type string is as follows:

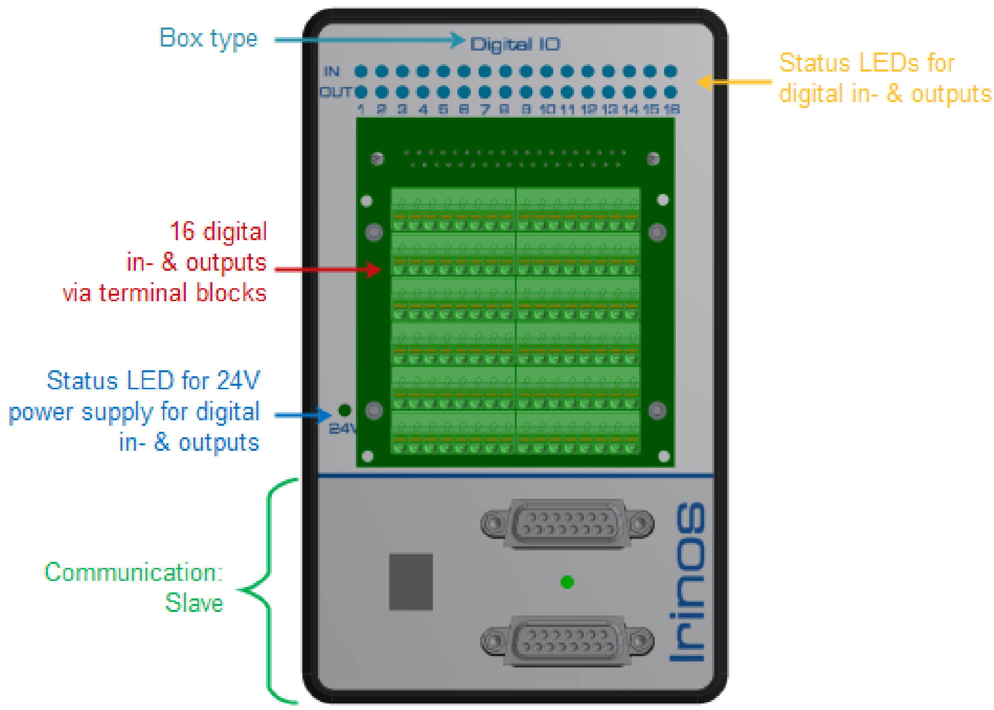




Connectors for "IR-DIO-16-16-M23-EXTP-IL"



Connectors for "IR-DIO-16-16-D37M-EXTP-IL"



Connectors for "IR-DIO-16-16-D37M-EXTP-IL" with terminal block adapter

Connectors / elements

○ 16 digital in- and outputs via M23 (80) or DSUB37 (83).

○ 24V power supply for the digital in- and outputs via M12 82.

○ An explanation of the communication section 44 can be found in the general overview.

Power supply

Depending on the type of the Irinos-Box IR-DIO, digital in- and outputs must be supplied by an external power supply. Using the M23 type, this is done via an M12 82 connector. Using the DSUB type, the supply is directly connected to the DSUB connector.

A blue LED is turned on, if the 24V power supply is available.

A power supply of the protection class PELV must be used.

Note: It is possible to use 12V DC for the power supply. In this case the status LED is turned off. All in- and output-LEDs have a low brightness. The switching characteristics remain unaffected.

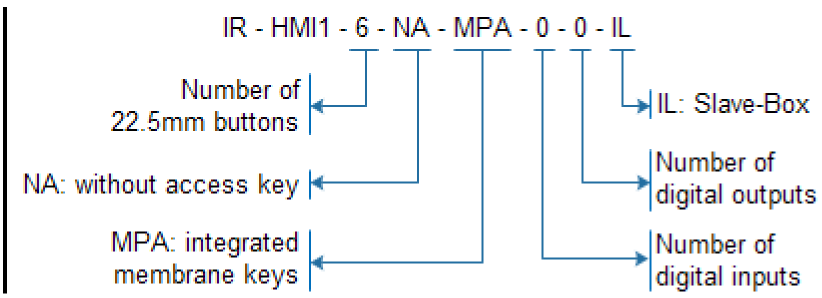
Protection

All outputs are individually protected against short circuit. Additionally the internal output circuit is protected against thermal overload. The available output power is specified in the data sheet.

4.7 IR-HMI1 control panel

The Irinos IR-HMI1 is a universal control panel for the Irinos measurement systems. All elements are directly accessible via Bit I/O 160, similar to an I/O Box with 40 in- and outputs.

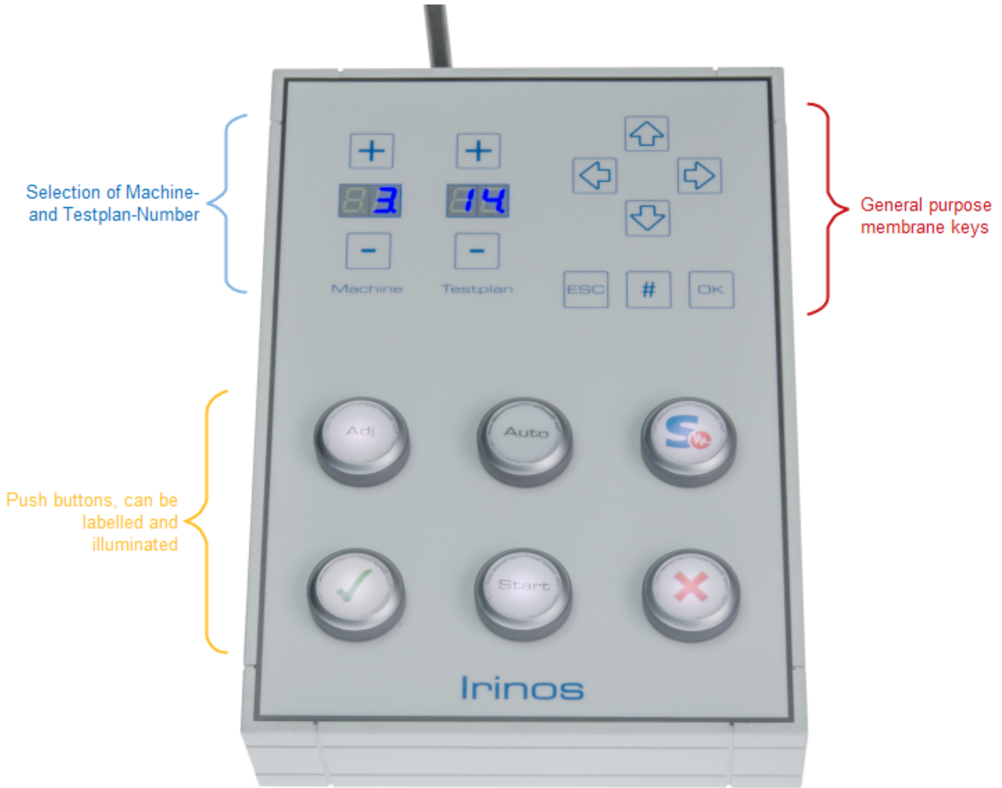
The definition of the type string is as follows:



The control panel provides the following control elements:

- 6 pcs. push buttons 22.5mm with user-definable functionality, each equipped with a LED for individual illumination, printable lables available.
- Selection of machine number (1-99) via numeric display and +- keys.
- Selection of testplan number (1-99) via numeric display and +- keys.
- 7 membrane keys with user-definable functionality: 4 arrows + ESC + # + OK

The control panel is connected to the Irinos system via the ILink113 interface. Power is also supplied via ILink.



Übersicht der Bedienelemente

The bit assignment is as follows:

Bit-No.	**Input** **(Irinos -> PC)**	**Output** **(PC -> Irinos**
0	Push button 1 (upper left)	LED for push button 1
1	Push button 2 (upper middle)	LED for push button 2
2	Push button 3 (upper right)	LED for push button 3
3	Push button 4 (lower left)	LED for push button 4
4	Push button 5 (lower middle)	LED for push button 5
5	Push button 6 (lower right)	LED for push button 6
6	-	Reserved
7	-	Reserved
8	Key "arrow left"	Preset value for Testplan No.
9	Key "arrow right"	Preset value for Testplan No.
10	Key "arrow up"	Preset value for Testplan No.
11	Key "arrow down"	Preset value for Testplan No.

Bit-No.	**Input** **(Irinos -> PC)**	**Output** **(PC -> Irinos**
12	Key "ESC"	Preset value for Testplan No.
13	Key "#"	Preset value for Testplan No.
14	Key "OK"	Preset value for Testplan No.
15	-	Latch preset value for Testplan No.
16	Reserved	Preset value for Machine No.
17	Reserved	Preset value for Machine No.
18	Reserved	Preset value for Machine No.
19	Reserved	Preset value for Machine No.
20	Reserved	Preset value for Machine No.
21	Reserved	Preset value for Machine No.
22	-	Preset value for Machine No.
23	-	Latch preset value

Bit-No.	**Input** **(Irinos -> PC)**	**Output** **(PC -> Irinos**
		for Machine No.
24	Current Testplan No.	Max Testplan No.
25	Current Testplan No.	Max Testplan No.
26	Current Testplan No.	Max Testplan No.
27	Current Testplan No.	Max Testplan No.
28	Current Testplan No.	Max Testplan No.
29	Current Testplan No.	Max Testplan No.
30	Current Testplan No.	Max Testplan No.
31	-	Lock Testplan selection
32	Current Machine No.	Max Machine No.
33	Current Machine No.	Max Machine No.
34	Current Machine No.	Max Machine No.
35	Current Machine No.	Max Machine No.
36	Current Machine No.	Max Machine No.
37	Current Machine No.	Max Machine No.

Bit-No.	**Input** **(Irinos -> PC)**	**Output** **(PC -> Irinos**
38	Current Machine No.	Max Machine No.
39	-	Lock Machine selection

Presetting the Testplan- / Machine-No.

The Testplan-No. can be preset by the measurement software. First the number must be set by the bits "Preset value for Testplan-No.". A low-to-high transition of the Bit "Latch Preset value for Testplan-No." latches this value as the new Testplan No.

The same applies to the Machine No.

Limiting the Testplan- / Machine-No.

The highest Testplan-No. can be defined by the Bits "Max. Testplan-No.".
If the current Testplan-No. is higher, it will automatically be set to the Max-value.
If the maximum value is 0, limiting the Testplan-No. is inactive.

The same applies to the Machine No.

Lock selection of Testplan- / Machine-No.

The selection of the Testplan-No. can be locked by setting the Bit "Lock Testplan selection".

The same applies to the Machine No.

Storing the Testplan- / Machine-No.

After a Testplan- or Machine-No. has been changed, the new value is stored permanently in the control panel (approx. 5s after the last change). After a restart of the Irinos system, the previous Testplan- and Machine-No. are displayed.

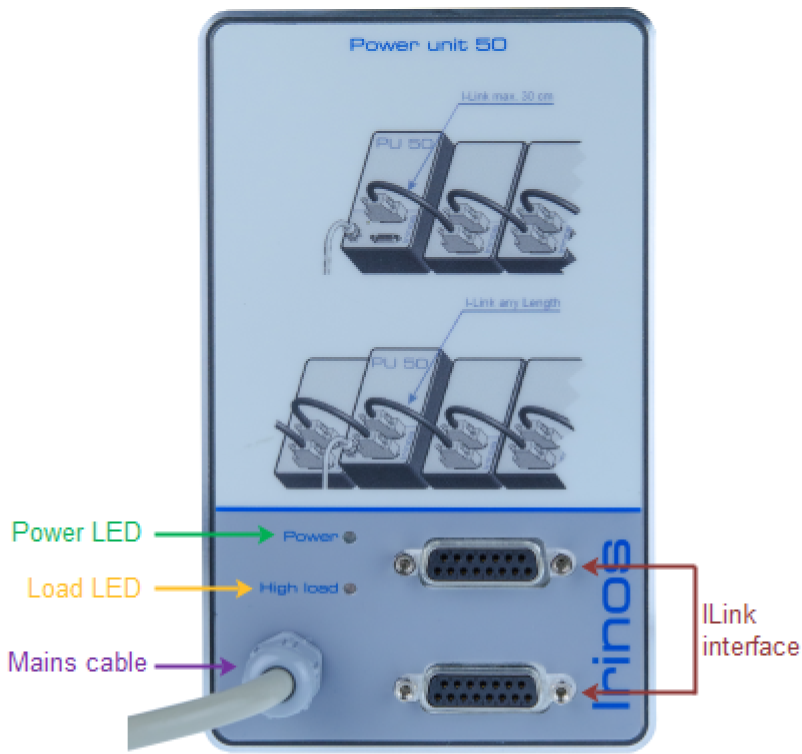
4.8 IR-PU power supply (industrial version)

The Irinos-Box IR-PU-50-HWS-x is a 24V power supply for the Irinos system with 50 watts rated output power. It has been designed for rough industrial environments with a long life expectancy.

It is the only Irinos-Box, which does not have a real ILink[27] interface. The ILink signals are connected to both connectors, but there is no internal circuit connected to the communication signals.

If the Irinos power supply IR-PU-50-HWS-x is used as the first or as the last Irinos-Box, the cable length[113] to the next Irinos-Box must not exceed 30cm.

The mains plug is equipped with a protective earth conductor. Hence no separate earth connection is required. However, it is important that the earth pin of the mains cable is connected to earth. The 24V output power will then fulfill the requirements of the protecion class PELV.



Connectors for "IR-PU-50-HWS-x"

LEDs / output power

The power supply has two LEDs:

- The green power LED signals that the 24V output power is available.
- The yellow load LED ("High load") is turned on, if the power consumption of the Irinos-System exceeds about 95% of the rated power. This means the LED is turned on if about 45 watts or more are consumed.

Even though the power supply is designed for continuous operation at full rated power, it is advised ***not to exceed 95% of the rated output power***. Use a more powerful supply, if the load LED is active continuously.

Most power consumers have no constant consumption[203]. Therefore it is advised to keep some margin.

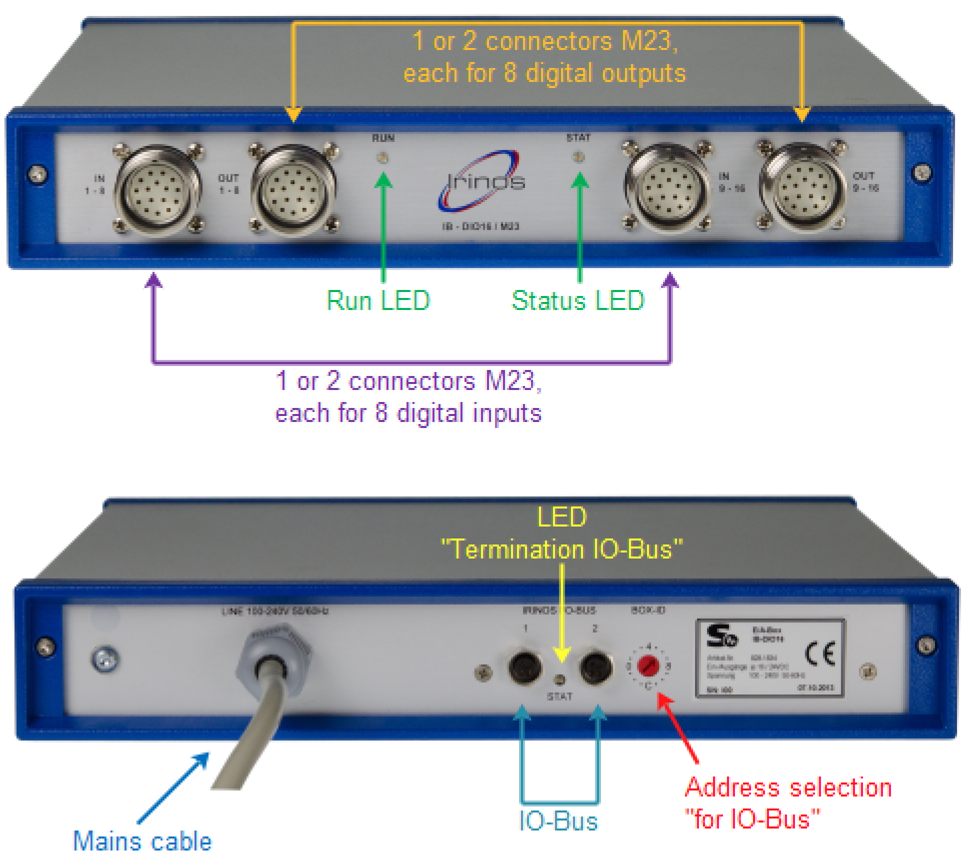
A short flickering of the load LED while the Irinos-System is in operation may indicate, that the power supply was overloaded for a short moment. This can lead to an unstable operation of the Irinos-System. A short flickering at startup is normal.

4.9 I/O-Boxes for the IO-Bus (desktop version)

The I/O-Boxes provide additional in- and outputs for the Irinos-Box IR-MASTER[47]. They are connected via the IO-Bus.

Types with 8 or 16 digital in- and outputs are available.

An important characteristic of the desktop I/O-Boxes is the integrated 50 watts power supply. It supplies the internal electronics as well as the digital in- and outputs.



Connectors for desktop I/O-Boxes

Connectors

○ 8 or 16 digital in- and outputs via connectors M23 80.

○ Connectors for IO-Bus 90.

Mains plug

The mains plug is equipped with a protective earth conductor. Hence no separate earth connection is required. However, it is important that the earth pin of the mains cable is connected to earth. The 24V output power will then fulfill the requirements of the protecion class PELV.

Protection

All outputs are individually protected against short circuit. Additionally the internal output circuit is protected against thermal overload. The available output power is specified in the data sheet.

IO-Bus - Connector

The I/O-Box is equipped with two IO-Bus connectors for further expansion by additional I/O-Boxes. Both connectors are equal. It does not matter, which of the connectors is used.

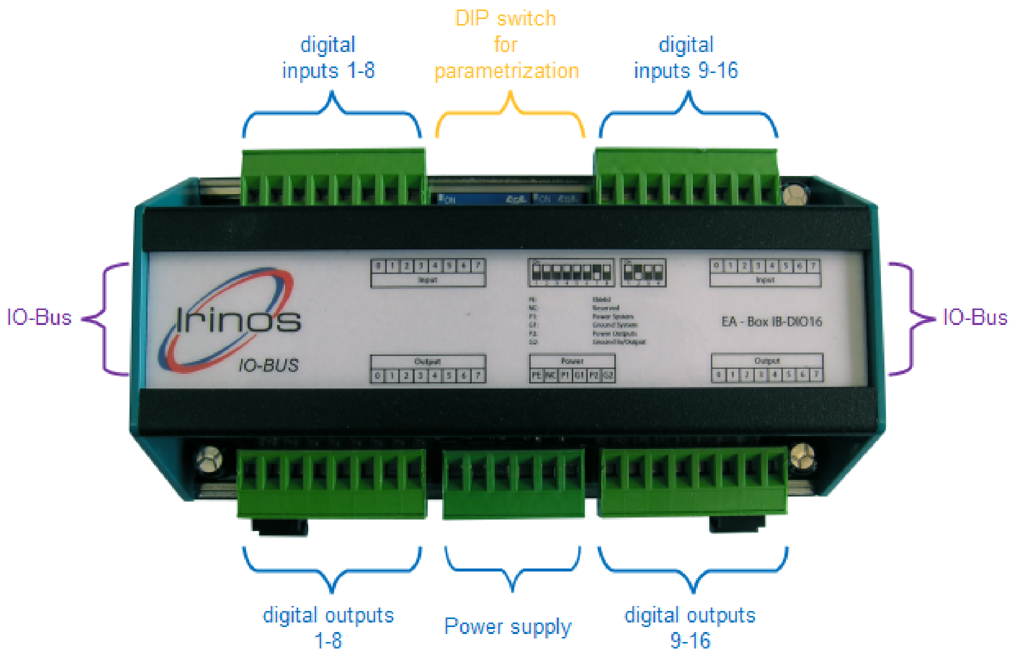
If only one cable is connected, this will be detected automatically. The IO-Bus will then be terminated internally. This is signalled by the termination LED.

The selection of the IO-Bus address (2, 4, 6 or 8) is made via the rotary switch at the rear side.

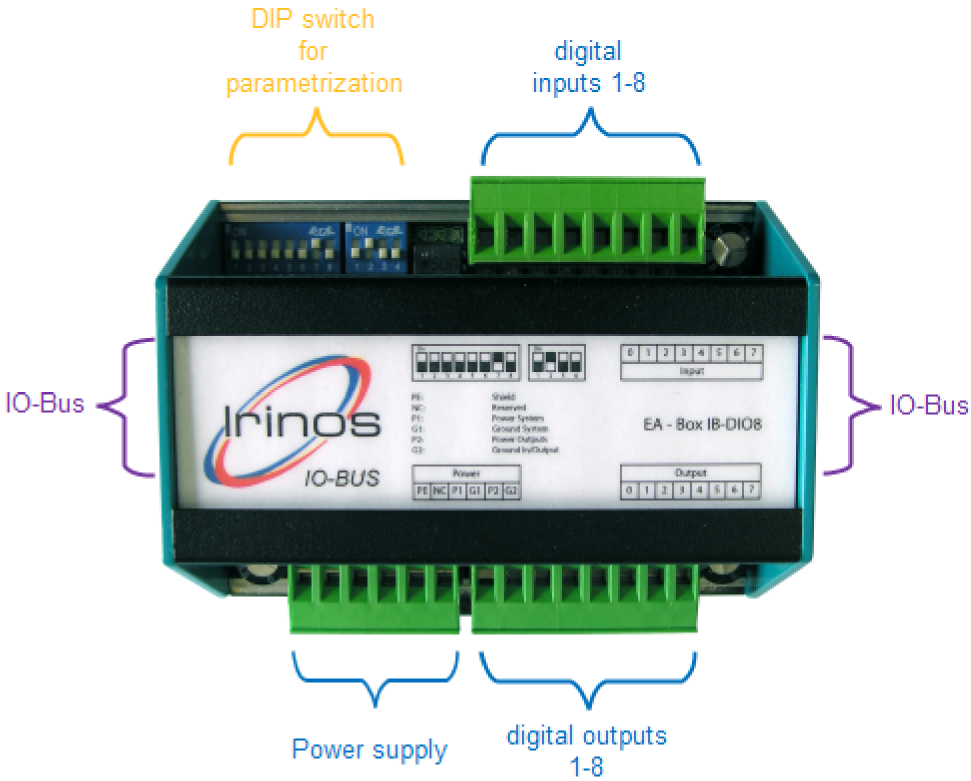
4.10 I/O-Boxes for the IO-Bus (hat / din rail version)

The I/O-Boxes provide additional in- and outputs for the Irinos-Box IR-MASTER 47. They are connected via the IO-Bus 91.

Types with 8 or 16 digital in- and outputs are available, both appropriate for hat / din rain mounting.



Connectors for I/O-Box hat/din rail with 16 I/Os



Connectors for I/O-Box hat/din rail with 8 I/Os

The ***power supply must be connected to terminal block*** PL5:

Pin	Description	Note
1	GND_IO	Ground potential for in- and outputs
2	24V_OUTPUTS	24V power supply for digital outputs
3	GND	Ground potential for the module
4	24V	24V power supply for the module
5	-	
6	PE	Shielding

The selection of the ***IO-Bus – Address*** (2, 4, 6 or 8) is made via the DIP-switch SW1. If it is the last I/O-Box connected to the IO-Bus, the termination114 must be enabled via this DIP-switch:

Switch								Function
1	2	3	4	5	6	7	8	
On								Termination enabled
Off								Termination disabled
	Off	Off	Off	Off	Off	**On**	Off	IO-Bus - Address 2
	Off	Off	Off	Off	**On**	Off	Off	IO-Bus - Address 4
	Off	Off	Off	Off	**On**	**On**	Off	IO-Bus - Address 6
	Off	Off	Off	**On**	Off	Off	Off	IO-Bus - Address 8

The DIP-switch SW2 must always have the following switch settings:

Switch				Note
1	2	3	4	
Off	Off	**On**	Off	Use these switch settings

Pin assignment

5 Pin assignment

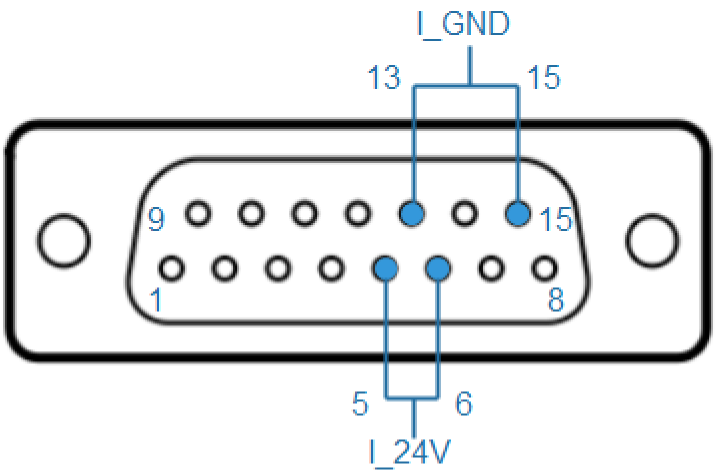
5.1 ILink connector (Master, integrated Master & Slave)

➔ ***Connector type: DSUB 15-pin female***

The ILink interface 27 combines wires for the power supply and the communication between the Irinos-Boxes in a single cable / connector. For proper operation of the Irinos-System, it is important to use high quality ILink cables with specific electrical specifications. Therefore only original cables 37 of the type IR-ILINK must be used.

By using an appropriate housing for the cable connectors, protection class IP65 can be reached. Therefore the ILink cables are available as a standard type and as a IP65 type.

Power 30 to the Irinos-System is supplied via the ILink interface. All other pins must remain unconnected, since this could lead to reflections, which can interrupt the data transfer between the Irinos-Boxes.



Power supply via ILink interface

Pin	Description	Note
1	IL_Data1	Data 1
2	IL_GND	Ground potential for data lines (not connected to the power supply or the internal supply of the Irinos-Box)

Pin	Description	Note
3	IL_Data2	Data 2
4	-	
5	**I_24V**	**24 V – power supply for the Irinos-System. Must be connected to Pin 6.**
6	**I_24V**	**24 V – power supply for the Irinos-System. Must be connected to Pin 5.**
7	IL_Data3	Data 3
8	IStat	Detection signal
9	IL_Data4	Data 4
10	-	
11	IL_Data5	Data 5
12	-	
13	**I_GND**	**Ground potential for the power supply of the Irinos-System. Must be connected to Pin 15.**
14	IL_Data6	Data 6
15	**I_GND**	**Ground potential for the power supply of the Irinos-System. Must be connected to Pin 13.**

The pinning of the ILink connector has been chosen to avoid any damage of incremental encoders with standard pinning96, if they are plugged into the ILink interface. For encoders with special pinning, this may be different.

The communication of the ILink interface is based on a bus system, which must be terminated by a termination resistor at both ends. This termination is switched on automatically113 : the Irinos-System detects the first and the last Irinos-Box and enables their termination resistors. This is signalled via the termination LED between the ILink connectors.

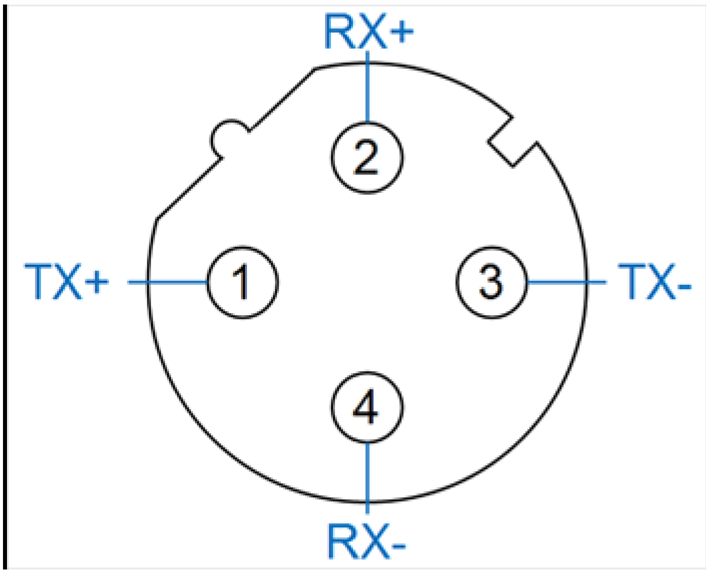
An exception are the Irinos power supplies IR-PU67. If it is placed at the beginning or the end of the ILink cabling113, the cable length to the next Irinos-Box must not exceed 0,3m.

5.2 Ethernet (Master & integrated Master)

➔ ***Connector type: M12 4-pin female, D-coded***

The Ethernet interface of the Irinos-System is a standard type, like it is for example available in standard PCs. Data rates of 10 Mbit/s and 100 Mbit/s are supported. Hence the Irinos-System can be connected to any typical network interface.

It has an integrated "crossover-detection". Therefore it does not matter, whether a standard cable or a crossed cable is used.



Pinning of the ethernet interface

Pin	Description	Note
1	TX+	Transmission Data+
2	TX-	Transmission Data-
3	RX+	Receive Data+
4	RX-	Receive Data-

Connection cable

Connection cables[38] are available from different manufacturers (e.g. PhönixContact, Lapp, Murr Elektronik). The connection cable must be of cat 5e or better. Cables from Phönix Contact are supporting the "SpeedConn"-locking for faster (un)plugging of the cable.

The ethernet interface is electrically identical to realtime interfaces like ProfiNet, EtherCat or Sercos. However, it does neither support the extensions of these nor the respective protocols.

Ethernet-Status / LEDs

The ethernet status is signalled via the LED "Ethernet Link".

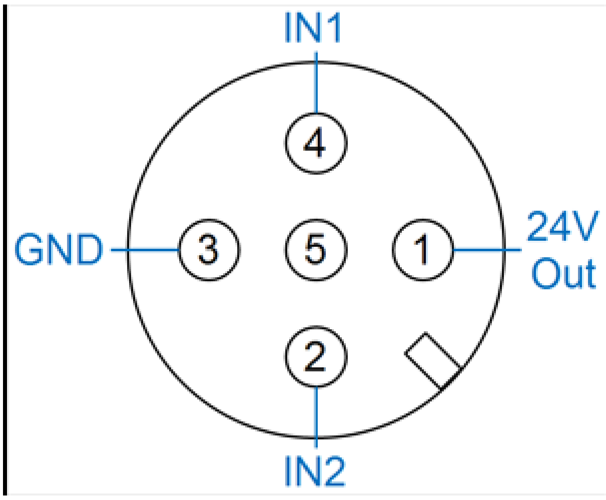
5.3 2 digital inputs (integrated Master)

➔ *This description is valid only for the digital inputs of the* ***integrated Master***[28] *. The digital inputs of the Irinos-Box IR-MASTER[47] are almost identical: they provide a higher output power.*

➔ ***Connector type: M12 5-pin female, A-coded***

Each "integrated Master" provides two digital inputs with M12 standard pinning. It also has a 24V output, which is internally connected via an 1 kOhm resistor to the power supply[30] of the Irinos system. It is appropriate for the supply of buttons or switches, which are connected to the digital inputs (e.g. foot pedals or push buttons).

It is not appropriate for the supply of sensors or similar, like for example proximity switches. A separate power supply must be used for this purpose.



Connector for digital inputs M12

Pin	Description	Note
1	24V Out	24V Output Please observe the notes above!
2	IN2	Digital input 2
3	GND	Ground for digital inputs
4	IN1	Digital input 1
5	-	

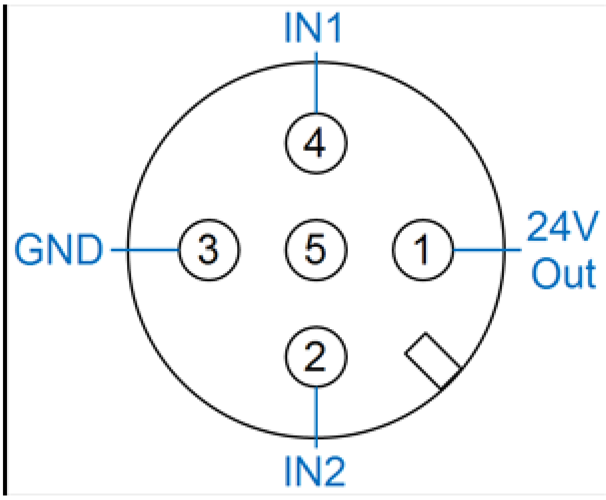
5.4 Digital inputs M12 (IR-MASTER)

➔ *This description is valid only for the digital inputs of the Irinos-Box **IR-MASTER** 47. The digital inputs of the integrated Master 28 are almost identical: they provide a lower 24V output power.*

➔ ***Connector type: M12 5-pin female, A-coded***

2 digital inputs are available in a single M12 connector with standard pinning. Via the Pin "24V Out", external components like push buttons, foot pedals, limit switches etc. can be supplied directly. Therefore the 24V supply of the ILink-Interface is used.

Please note that the output power is limited (see specifications in datasheet). Further the power supply 30 used for the Irinos-System must be able to provide the required load power.



Connector for digital inputs M12

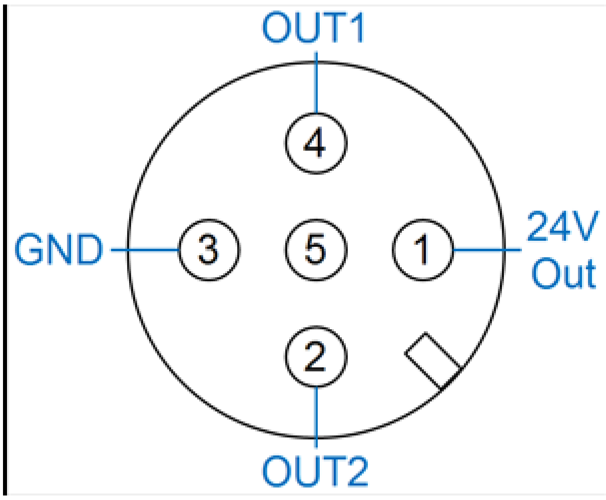
Pin	Description	Note
1	24V Out	24V Output
2	IN2	Digital input 2
3	GND	Ground for digital inputs
4	IN1	Digital input 1
5	-	

5.5 Digital outputs M12 (IR-MASTER)

➔ ***Connector type: M12 5-pin female, A-coded***

2 digital inputs are available in a single M12 connector with standard pinning. Via the Pin "24V Out", external components can be supplied directly.

For the digital outputs, the 24V supply 30 from the ILink interface is used. Please note, that the output power is limited (see specifications in datasheet). Further the power supply used for the Irinos-System must be able to provide the required load power.



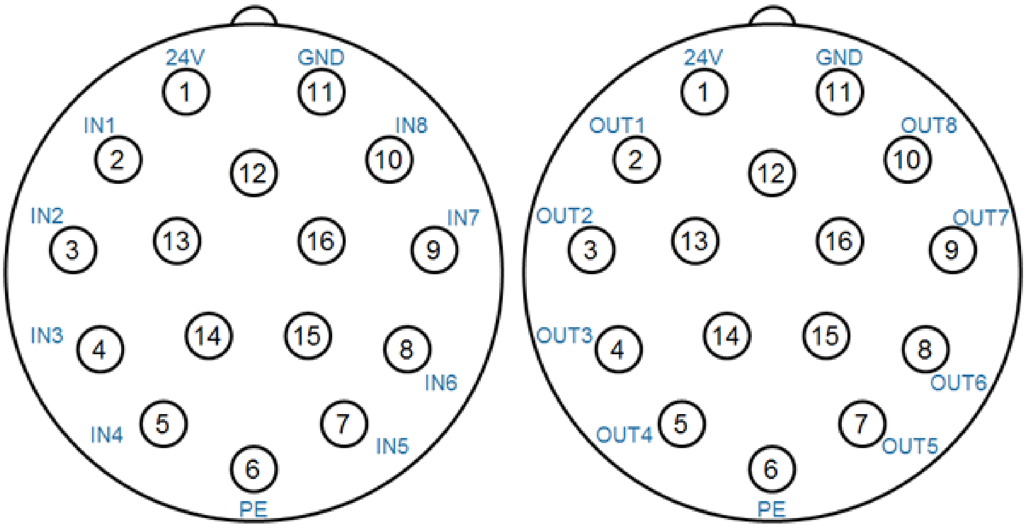
Connector for digital outputs M12

Pin	Description	Note
1	24V Out	24V Output
2	OUT2	Digital output 2
3	GND	Ground for digital outputs
4	OUT1	Digital output 1
5	-	

5.6 Digital in-/outputs M23 (IR-DIO-16-16-M23-xx-IL) and I/O-Boxes for IO-Bus

➔ ***Connector type: M23 16-pin female***

8 digital in- or outputs are provided per M23 connector. It is typically used to connect "quick connect elements" with M12 or M8 connectors. These are available from different suppliers (e.g. MurrElektronik, Turck, Weidmüller, PhönixContact, Erni, Escha).



Connectors for digital in-/outputs M23

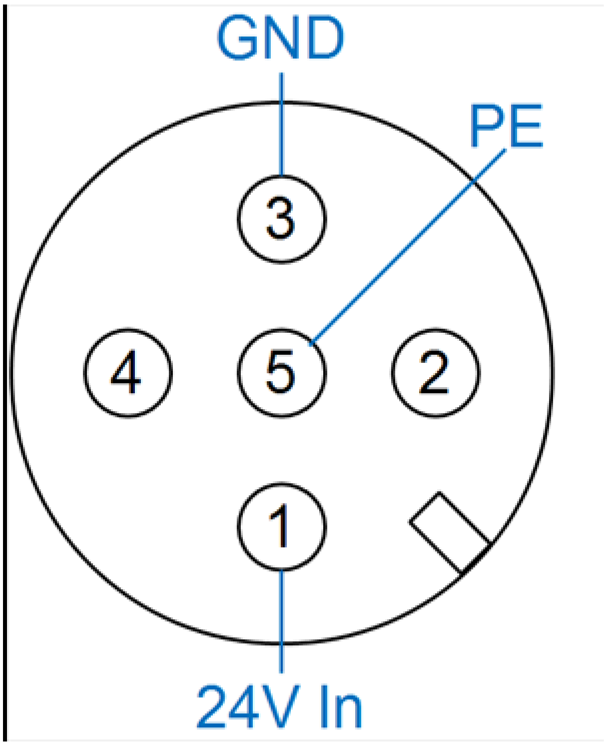
Pin	Description	Note
1	24V	24V Output
2	IN1 / OUT1	Digital in- or output 1 / 9
3	IN2 / OUT2	Digital in- or output 2 / 10
4	IN3 / OUT3	Digital in- or output 3 / 11
5	IN4 / OUT4	Digital in- or output 4 / 12
6	PE	Protective earth (**only** for type IR-DIO-16-16-D37-**EXTP**-IL with separate power supply)
7	IN5 / OUT5	Digital in- or output 5 / 13
8	IN6 / OUT6	Digital in- or output 6 / 14
9	IN7 / OUT7	Digital in- or output 7 / 15
10	IN8 / OUT8	Digital in- or output 8 / 16
11	GND	Ground for digital in- and outputs
12 .. 16	-	

5.7 Power supply for digital in- and outputs M12 (IR-DIO-16-16-M23-EXTP-IL)

➔ ***Connector type: M12 5-pin male, A-coded***

Power for the digital in- and outputs must be supplied via this connector.

Make sure to use a power supply, which fulfills the requirements of the protecion class PELV!



Connector for I/O power supply via M12

Pin	Description	Note
1	24V In	24V Supply input
2	-	
3	GND	Ground
4	-	
5	PE	Protective earth

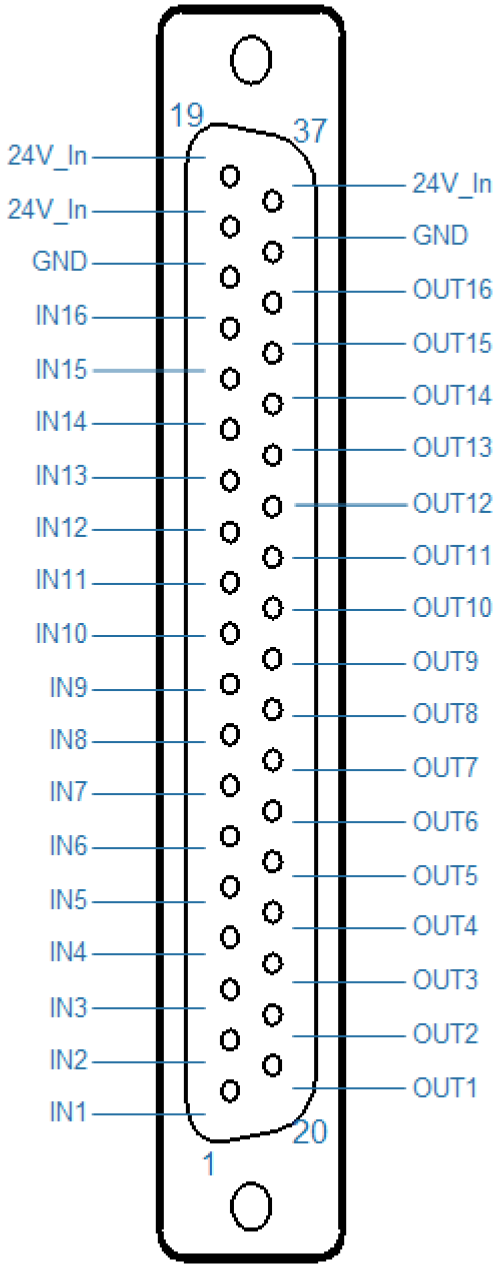
5.8 Digital in-/outputs DSUB 37 (IR-DIO-16-16-D37-EXTP-IL)

➔ ***Connector type DSUB 37-pin, male***

All digital in- and outputs are available via a single DSUB connector.

If only digital inputs are used, there is no need for a 24V supply. Otherwise an external 24 supply must be connected to the pins 18, 19 & 37. In any case a ground wiring is required (pins 17 & 36).

In order to fulfill the requirements of the protection class IP65, an IP65 DSUB connector housing from FCT / Molex is required.



Connector DSUB37 for digital in- and outputs

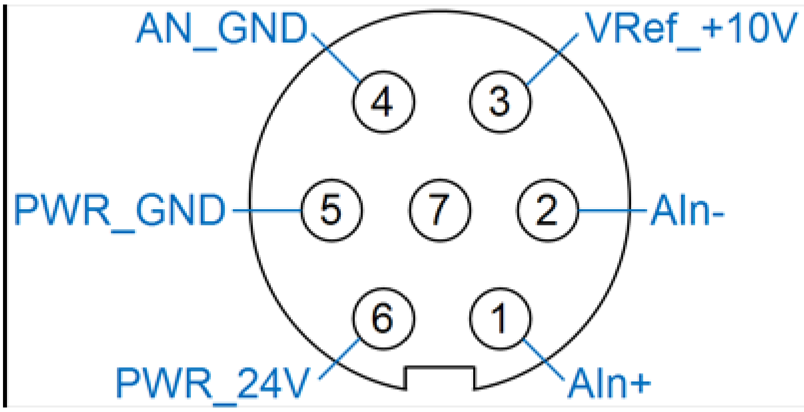
Pin	Description	Note	Pin	Description	Note
1	IN1	Input 1	20	OUT1	Output 1
2	IN2	Input 2	21	OUT2	Output 2
3	IN3	Input 3	22	OUT3	Output 3
4	IN4	Input 4	23	OUT4	Output 4
5	IN5	Input 5	24	OUT5	Output 5
6	IN6	Input 6	25	OUT6	Output 6
7	IN7	Input 7	26	OUT7	Output 7
8	IN8	Input 8	27	OUT8	Output 8
9	IN9	Input 9	28	OUT9	Output 9
10	IN10	Input 10	29	OUT10	Output 10
11	IN11	Input 11	30	OUT11	Output 11
12	IN12	Input 12	31	OUT12	Output 12
13	IN13	Input 13	32	OUT13	Output 13
14	IN14	Input 14	33	OUT14	Output 14
15	IN15	Input 15	34	OUT15	Output 15
16	IN16	Input 16	35	OUT16	Output 16
17	GND	Ground for digital in- & outputs	36	GND	Ground for digital in- & outputs

Pin	Description	Note	Pin	Description	Note
18	24V_In	24V power supply (input)	37	24V_In	24V power supply (input)
19	24V_In				

5.9 Analogue inputs ±10V (IR-MASTER & IR-AIN)

➔ *Connector type M16 7-pin, female*

Each connector provides one analogue input ±10V:



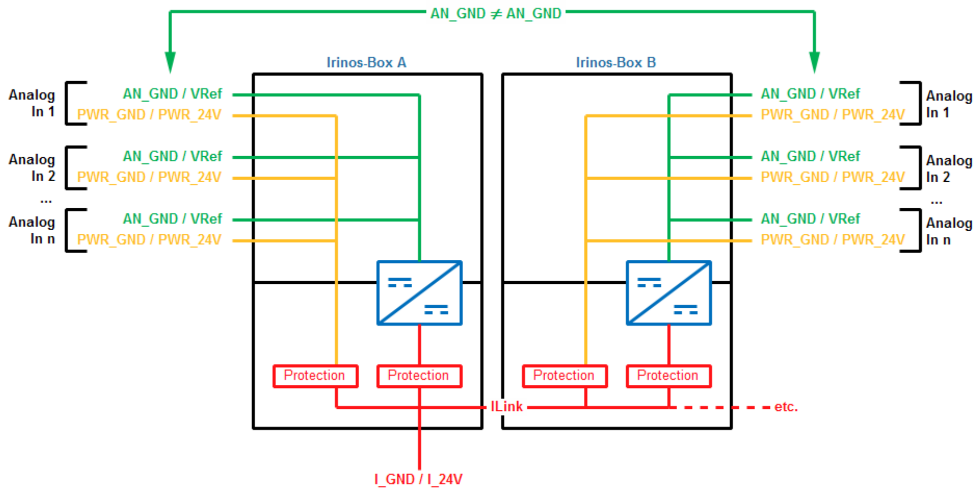
Connector for analogue inputs ±10V

Pin	Description	Note
1	Ain+	Positive signal of the differential analogue input.
2	Ain-	Negative signal of the differential analogue input.
3	VRef_+10V	+10 V Reference voltage (output)
4	AN_GND	Ground for analogue input (Ain+ / Ain-) and reference input VRef_+10V.
5	PWR_GND	Ground for 24V sensor supply output
6	PWR_24V	24V sensor supply output
7	-	

Isolation between ground potentials

The whole internal power supply of each Irinos-Box is galvanically isolated from the ILink power supply 30. This also covers the analogue ground AN_GND. Hence the analogue ground potentials of two Irinos-Boxes are also galvanically isolated. All analogue inputs of the same Irinos-Box have the same ground potential.

This separation of ground potentials prevents ground loops, which typically lead to measurement problems. Nonetheless it is important, that all signal sources are properly connected to the Irinos-System.

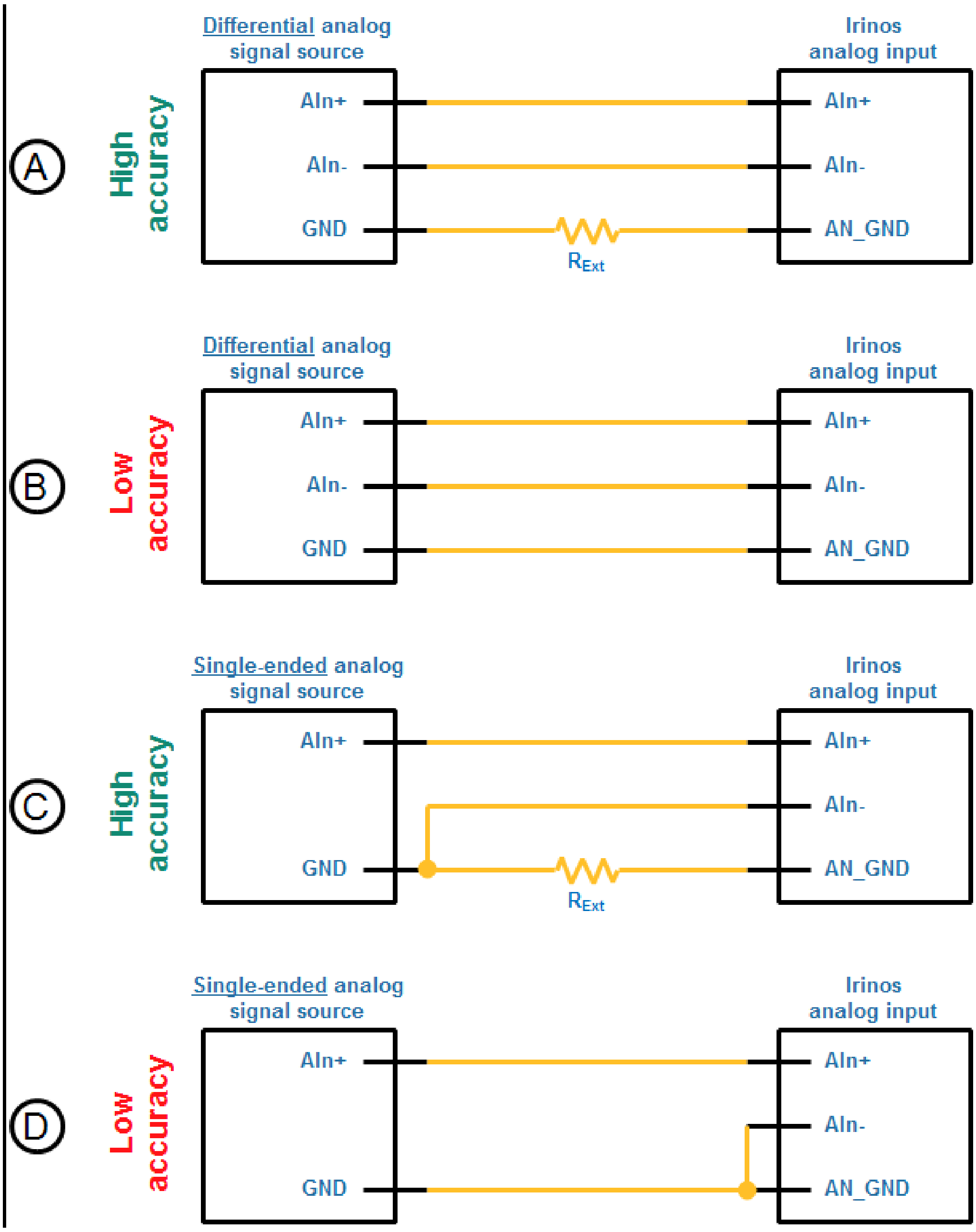


Ground potentials for analogue inputs

Connecting analogue signal sources

The following figure shows several examples for the connection of analogue signal sources. As shown, the analogue ground AN_GND should not be connected directly to any other ground potential. In order to achieve a high measurement stability, it is advised to use a resistor RExt (see A and C). The resistance value depends on the application. Typically a 1 kOhm resistor is appropriate.

For single-ended measurements, the input AIn- must be connected to the ground potential GND of the signal source. This connection should be made next to the signal source, if the ground potentials are connected via a resistor as depicted above (see B). Without this resistor, the connection should be made next to the Irinos-Box, e.g. inside the connector (see D).

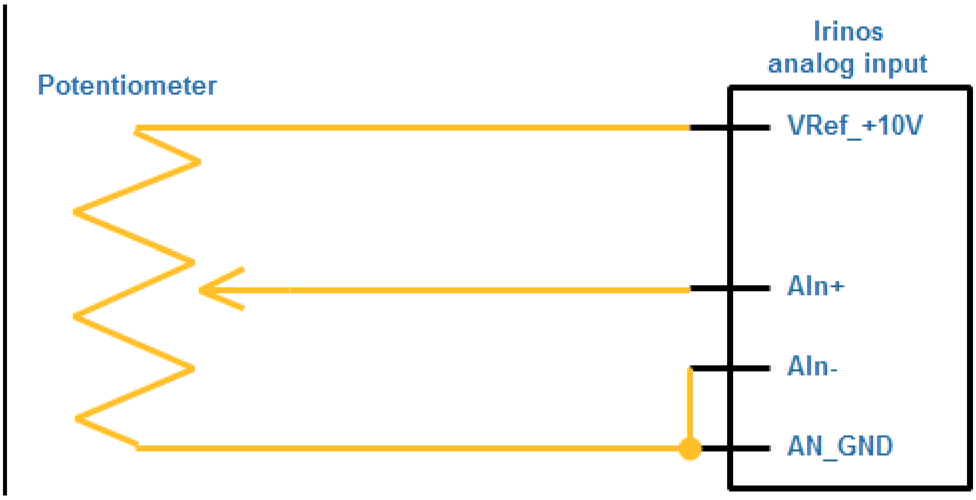


Connecting analogue signal sources

Output VRef_+10V

The output VRef_+10V can be used as a reference voltage for a high-impedance measurement circuit. This is for example required for the connection of measurement potentiometers.

Please observe the maximum output power for the reference voltage. It can be found in the datasheet.

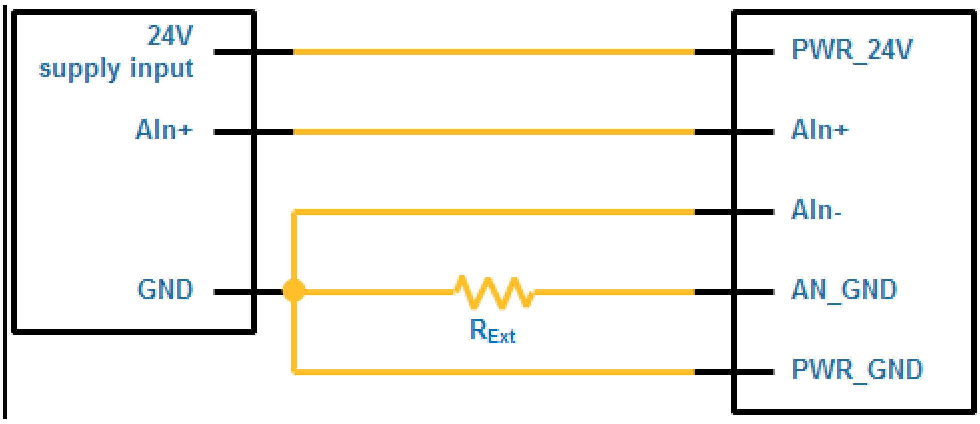


Example: connecting a potentiometer to an analogue input

24 supply for analogue sensors

For the supply of analogue sensors, e.g. temperature transducers, 24V are available at each output connector. These are taken from the ILink power supply. The maximum load is defined in the datasheet.

The following figure shows an example for the supply of an analogue sensor via the Irinos-System. Since the analogue ground AN_GND and the ILink supply ground PWR_GND are isolated, these must be connected via an external resistor.



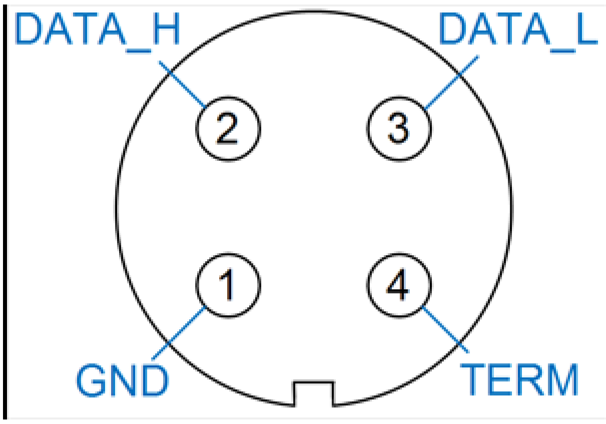
Example for the connection of an analogue sensor

5.10 IO-Bus M9 connector for IR-MASTER and I/O-Boxes

➔ Connector type M9 4-pin, female

For a reliable operation of the IO-Bus[114], it is recommended to use ready-made cables (different types available[39]).

The IO-Bus must be terminated at both ends. It is integrated into the Irinos-Box IR-MASTER 47. Thus pin 4 (Term) is not used. However, for compatibility reasons, it should be connected to pin 1 (GND). The TERM signal must not be connected between both ends of the cable.



Connector M9 for M9

Pin	Description	Note
1	GND	Ground
2	DATA_H	Data high
3	DATA_L	Data low
4	TERM	Termination detection (input; not used at IR-MASTER 47)

5.11 IO-Bus DSUB connector for I/O-Boxes

➔ ***Connector type DSUB male or female***

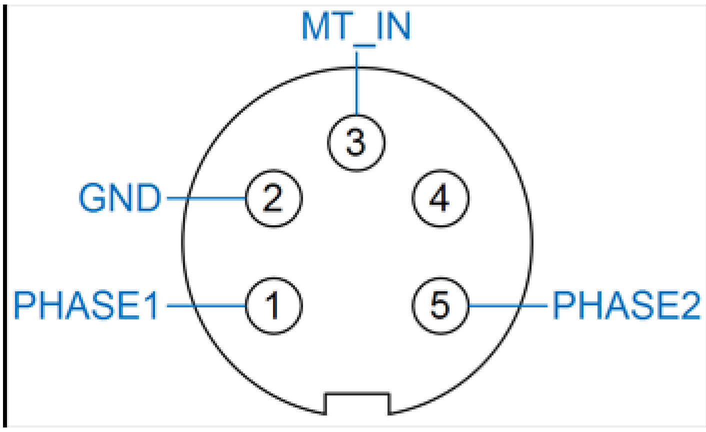
For a reliable operation of the IO-Bus, it is recommended to use ready-made cables (different types available 39).

Pin	Description	Note
1	-	
2	DATA_L	Data low
3	GND	Ground
4	-	
5	-	
6	GND	Ground
7	DATA_H	Data high
8	-	
9	-	(Do never connect this pin, since it may be connected to 24V at some box types.)

5.12 Inputs for inductive probes (IR-TFV)

➔ **Connector type M16 5-pin 270°, female**

One inductive probe can be connected to each connector. The pinning corresponds to the standard pinning for the respective probe.



Connector M16 for inductive probes

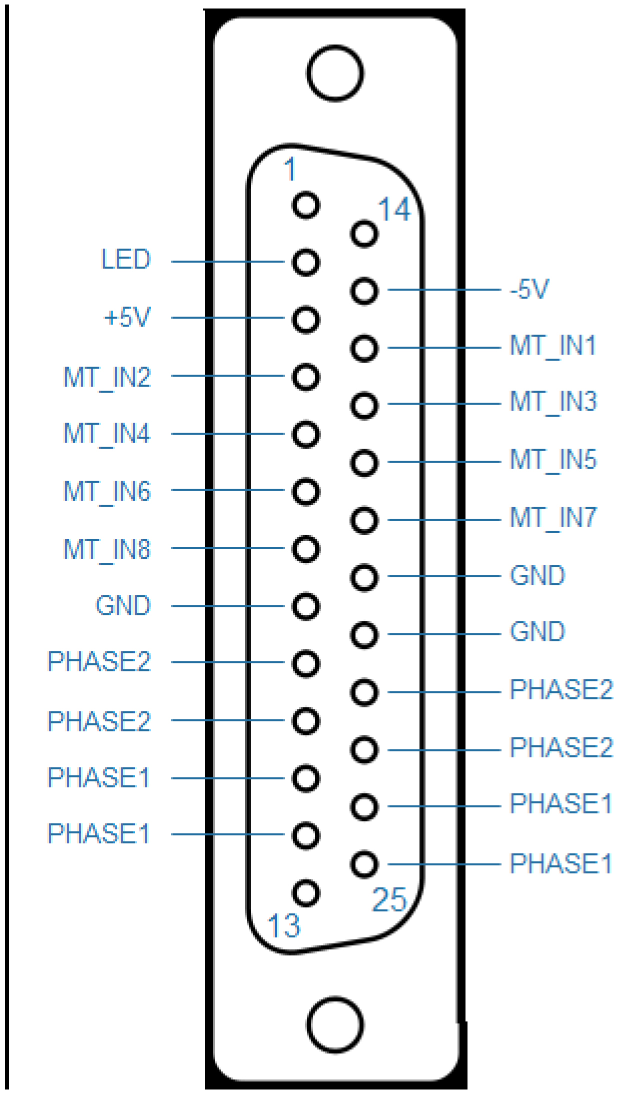
Pin	Description	Note
1	PHASE1	Sine-oscillator (together with PHASE2)
2	GND	Ground
3	MT_IN	Measurement signal input (measurement signal of the inductive probe)
4	-	
5	PHASE2	Sine-oscillator (together with PHASE1)

5.13 Connector KF27 for quick-change box for inductive probes (IR-TFV)

➔ *Connector type DSUB 25-pin, female*

Please note: cabling has a high influence on measurement accuracy and stability. It is recommended to use ready-made cables39.

In order to fulfill the requirements of the protection class IP65, an IP65 DSUB connector housing from FCT / Molex is required.



Connector for quick-change box for inductive probes

Pin	Description	Note
1	-	
2	LED	Connection of an external LED (currently not used).
3	+5V	Positive analogue supply

Pin	Description	Note
4	MT_IN2	Input 2 (Measurement signal of inductive probe 2)
5	MT_IN4	Input 4 (Measurement signal of inductive probe 4)
6	MT_IN6	Input 6 (Measurement signal of inductive probe 6)
7	MT_IN8	Input 8 (Measurement signal of inductive probe 8)
8	GND	Ground
9	PHASE2	Sine-oscillator (together with PHASE1)
10	PHASE2	
11	PHASE1	Sine-oscillator (together with PHASE2)
12	PHASE1	
13	-	
14	-	
15	-5V	
16	MT_IN1	Input 1 (Measurement signal of inductive probe 1)
17	MT_IN3	Input 3 (Measurement signal

Pin	Description	Note
		of inductive probe 3)
18	MT_IN5	Input 5 (Measurement signal of inductive probe 5)
19	MT_IN7	Input 7 (Measurement signal of inductive probe 7)
20	GND	Ground
21	GND	
22	PHASE2	Sine-oscillator (together with PHASE1)
23	PHASE2	
24	PHASE1	Sine-oscillator (together with PHASE2)
25	PHASE1	

5.14 Incremental encoder 1Vpp or TTL/RS422 (IR-INC)

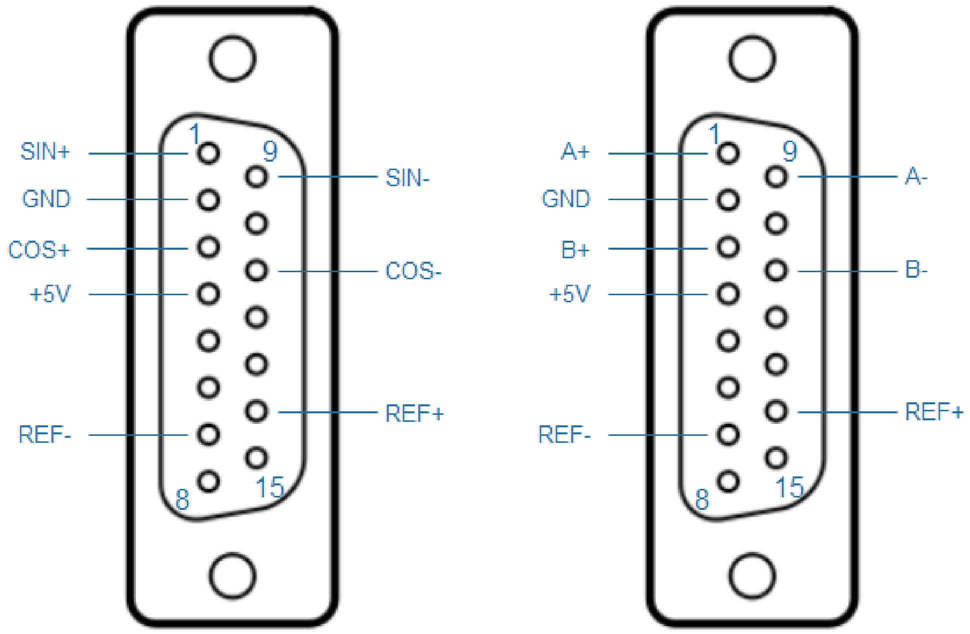
➔ ***Connector type DSUB 15-pin, female***

In order to fulfill the requirements of the protection class IP65, an IP65 DSUB connector housing from FCT / Molex is required (type FWA2GA).

The same connector type is used for the ILink interface 74 and the incremental encoders.

The pinning of the ILink connector has been chosen to avoid any damage of incremental encoders with standard pinning, if they are plugged into the ILink interface. For encoders with special pinning, this may be different.

Only use the appropriate connectors for the incremental encoders.



Connector for incremental encoder 1Vpp or TTL / RS422

Pin	Description 1Vpp	Description TTL	Note
1	SIN+	A+	1Vpp: Sine positive TTL/RS422: A signal positive
2	GND	GND	Ground
3	COS+	B+	1Vpp: Cosine positive TTL/RS422: B signal positive

Pin	Description 1Vpp	Description TTL	Note
4	+5V	+5V	+5V supply for incremental encoder
5	-	-	Not used. (At the ILink-Interface this pin is connected to +24V.)
6	-	-	Not used. (At the ILink-Interface this pin is connected to +24V.)
7	REF-	REF-	Reference signal, negative
8	-	-	
9	SIN-	A-	1Vpp: Sine negative TTL/RS422: A signal negative
10	-	-	
11	COS-	B-	1Vpp: Cosine negative TTL/RS422: B signal negative
12	-	-	
13	-	-	Not used. (At the ILink-

Pin	Description 1Vpp	Description TTL	Note
			Interface this pin is connected to power GND.)
14	REF+	REF+	Reference signal, positive
15	-	-	Not used. (At the ILink-Interface this pin is connected to power GND.)

Mounting & Cabling

6 Mounting & Cabling

➔ Carefully read the safety instructions 18 before mounting.

6.1 Checking the delivery

Procedure:

- Please check the packing and complain partial damage immediately at the carrier.
- Unpack the Irinos components at destination.
- Keep the original packaging in case the device should be stored or transported at some future time.
- Check the delivery regarding completeness and possible damage. If anything is missing, incomplete or damaged, please immediately inform your supplier.
- Keep all included documents. They are part of the Irinos-System.

The delivery of each Irinos-Box contains:

- Irinos-Box
- Booklet with safety notes
- Warning note DHCP server (only for IR-MASTER 47 and „integrierted Master 28")
- 2 protection caps for ILink DSUB connector (only for IR-MASTER 47 and „integrierted Master 28")

Cables are delivered without further accessories.

The following table shows the contents of Irinos accessories:

Description	Content of delivery
IR-MHRM-1 [36] Hat / Din rail mounting kit	o Hat / Din rail mounting adapter o 2 countersunk head screws hex key M4 for mounting the adapter to the Irinos-Box
IR-MFFM-1 [36] Front side mounting kit	o Front side mounting adapter o 2 countersunk head screws hex key M4 for mounting the adapter to the Irinos-Box
IR-MITEM-40 [36] Mounting bracket for aluminium profile	o Mounting bracket o 2 head screws hex key M4 for mounting the bracket to the Irinos-Box
IR-MWIP-40 [36] Stand for aluminium profile	o 2 Mounting brackets (1 left side, 1 right side)
IR-MIPL-8-ABB179 [36] Labelling carrier	o Labelling carrier o 3 head screws hex key M3 for mounting the labelling carrier to the Irinos-Box

6.2 Mounting location

The Irinos-Boxes are designed as field devices. This allows mounting in closed cases, e.g. in a switching cabinet, as well as placement next to the measurement device.

Especially for larger measurement applications, placement next to the measurement device is preferred. This provides two important benefits:

- Cabling of the probes, encoders or any other sensors can be kept short. This results in a higher quality of the analogue signals. Further it helps to keep cables away from possible sources of noise.
- Changing a probe is easier, e.g. in case it is broken.

For a troublefree operation of the Irinos-System, please note:

Always keep a distance to possible sources of noise, e.g. frequency changers or motor cables.

An important factor regarding the location selection, is the IP protection class. Even though most of the Irinos components are available with a high protection class, it is in reality difficult to achieve it for the whole system. In order to comply with high IP protection classes, it is important to use appropriate cables and connectors. Most of the standard probes and incremental encoders are equipped with a connector providing a low protection (e.g. IP00 or IP40). These connectors would need to be replaced, which typically is not a favourite choice for the end user.

Hence it is adviced to place the Irinos-Boxes at places where a low or no IP protection is required.

The Irinos-Boxes have a low internal heat generation and are designed for typical industrial ambient temperatures. Further the internal measurement electronics has a high temperature stability.

Nevertheless a location with a moderate ambient temperature should be chosen (see also the specifications in the datasheets). It should particularly be avoided to place the Irinos components next to sources of heat, e.g. heat sinks of other devices or heating elements.

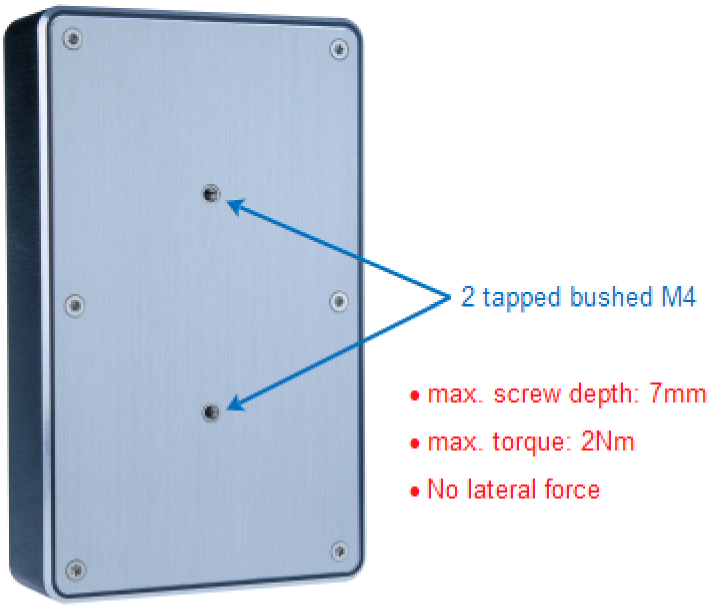
6.3 Mounting the Irinos-Boxes

6.3.1 Direct mounting via tapped bushes on rear side

Each Irinos-Box has 2 tapped busches M4 on the rear side. These allow direct mounting.

Please observe the following limitations:

- The maximum screw depth is 7 mm. Please chose screws with appropriate length.
- The maximum torque is 2 Nm.
- No lateral force is allowed. Otherwise the tapped bushes may become loose.



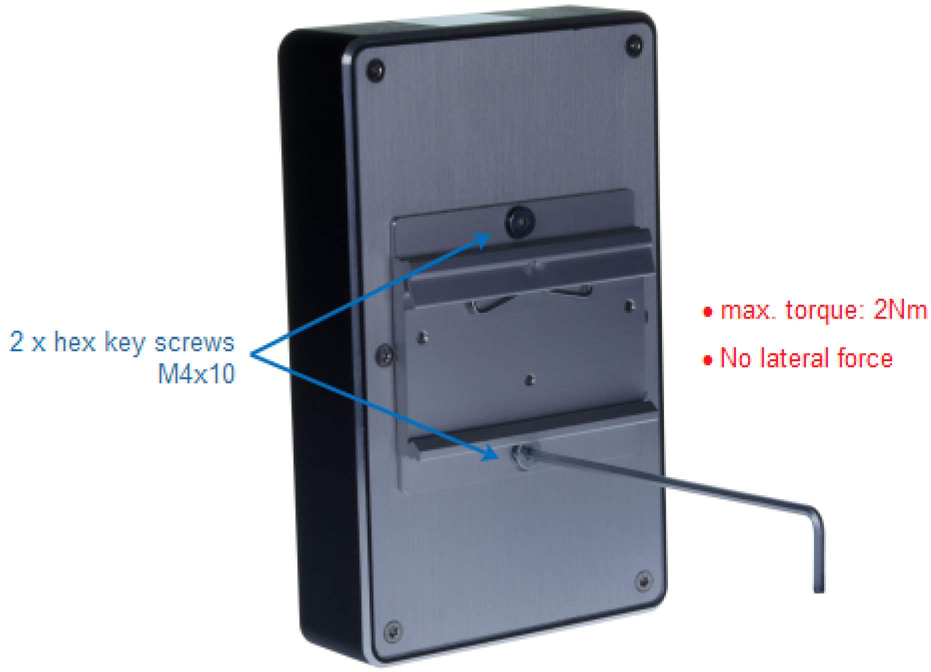
Mounting via tapped bushes on rear side

6.3.2 Hat / Din rail mounting

➔ ***For hat / Din rail mounting, the mounting kit IR-MHRM-1*** 36 ***is required.***

The mounting kit IR-MHRM-1 is delivered pre-assembled (see figure). It is fixed at the Irinos-Box with 2 hex key screws.

Hat / Din rail mounting kit IR-MHRM-1



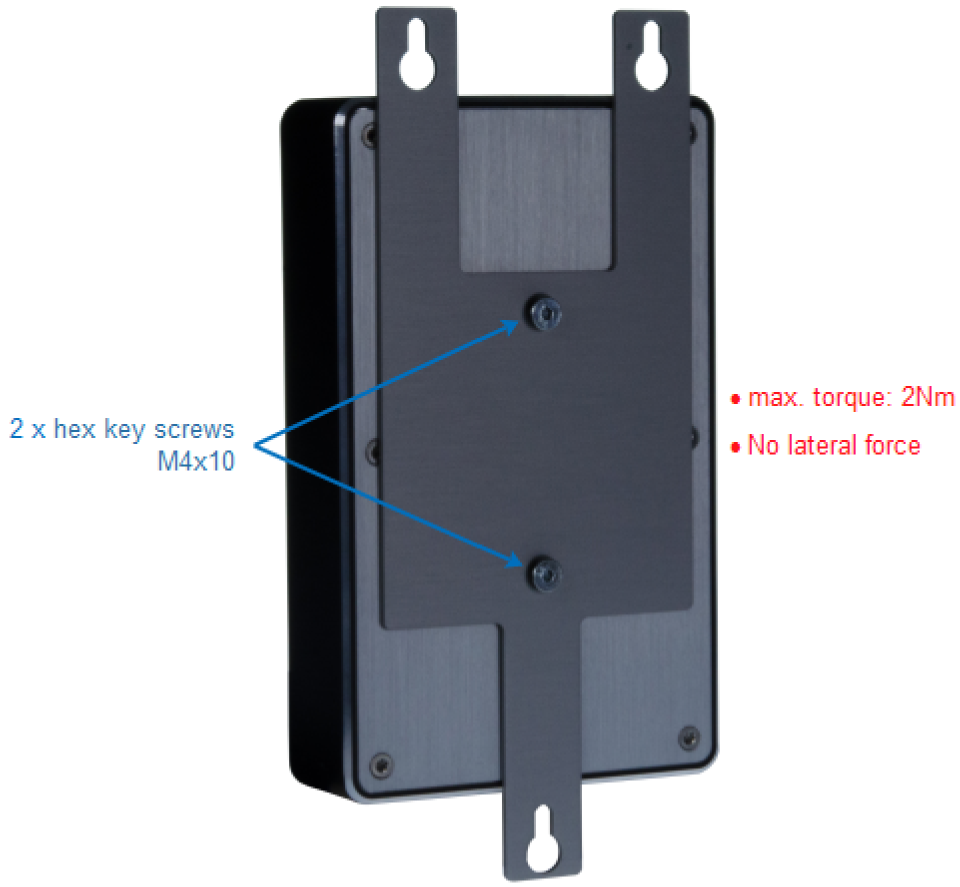
Fixing the hat / din rail mounting kit IR-MHRM-1

6.3.3 Front-side mounting

➔ ***For front-side mounting, the mounting kit IR-MFFM-1*** 36 ***is required.***

The front-side mounting adapter is fixed to the rear side of the Irinos-Box using two hex screws.

Afterwards the Irinos-Box can be mounted from the front side using the three "keyholes".

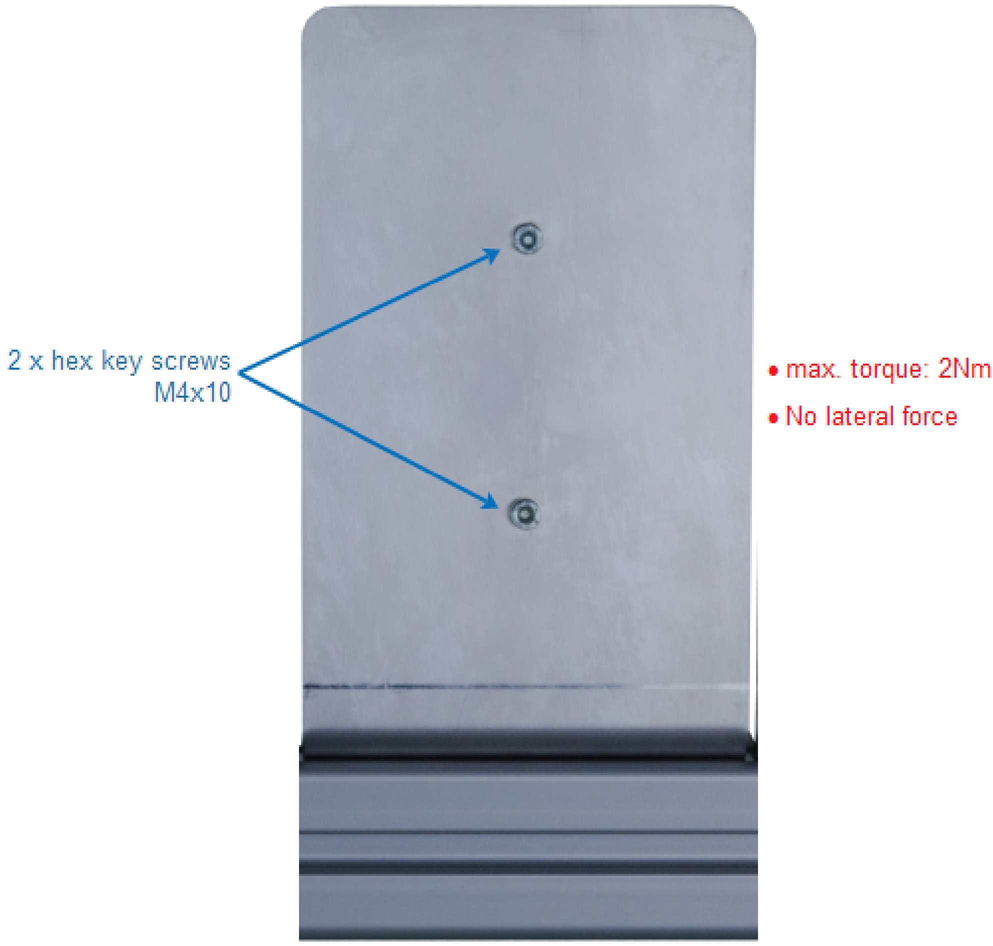


Front-side mounting with IR-MFFM-1

6.3.4 Mounting at 40mm aluminium profile

➔ ***Requires mounting bracket [IR-MITEM-40](#) 36.***

The mounting bracket is fixed at the rear side of the Irinos-Box with 2 hex key screws:



Fixing the mounting bracket at the Irinos-Box

Afterwards the bracket can be fixed at an aluminium profile 40mm using the two "keyholes". Appropriate slot-nuts and screws are required. These are not included in the delivery.

Please note that the bracket has been designed for aluminium profiles of the manufacturer Item. In most cases it can also be used for aluminium profiles of other manufacturers. However, this must be checked before use.

Fixing the mounting bracket at the aluminium profile

6.3.5 Stand for 40mm aluminium profile

A complete mounting stand requires

- the stand IR-MWIP-40 36 and
- a 40mm aluminium profile (Item or compatible).

The aluminium profile ist not contained in the delivery. Its length determines the maximum number of Irinos-Boxes, which can be mounted.

A tap is required to cut a thread into the aluminium profile (typically M8; depending on profile).

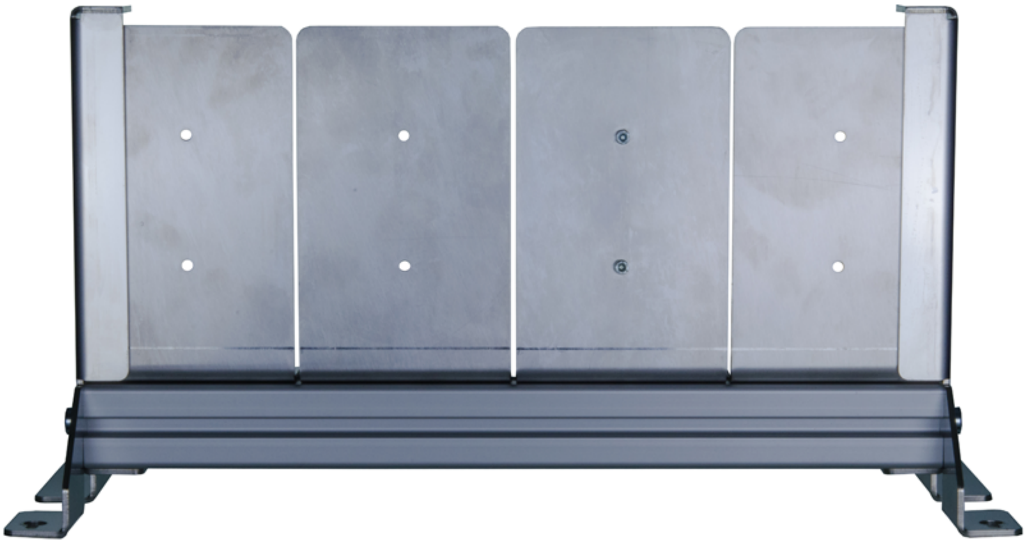
Procedure

1. Cut a thread into the aluminium profile at both ends.
2. Fix the stand IR-MWIP-40 at both ends of the aluminium profile. Please note that one of them is appropriate for the left side and the other one for the right side.
3. Optionally a protection cover can be fixed at the upper side of the stand. This allows protecting the Irinos-System against dripping fluids.

4. *Side view*

Example

Front side



Rear side

6.4 Cabling

Proper cabling is necessary for a trouple-free operation of the Irinos-System. Please observe the following instructions:

- All cables must be laid in distance to possible sources of noise, e.g. frequency changers or motor cables.
- Avoid cables which are longer than required. Especially avoid cable loops.
- All measurement and ILink cables must be shielded properly.

o Use the fixing elements of the connectors.

o Avoid mechanical stress regarding the cables and connectors.

If you are using drag chains, make sure to use appropriate cables.

Cover all unused connectors with a protection cap.

During cabling, the Irinos-System and all other devices must be turned off.

The following shows a typical connection sequence:

a) Connect Ethernet cable[112].

b) Connect ILink cables[113].

c) Connect measurement equipment and digital in- and outputs.

6.4.1 Ethernet cabling

The Ethernet interface is of standard type (no special enhancements), like it is used for example in IT networking. This allows connecting the Irinos-System to any standard PC or laptop Ethernet port. It can also be used with standard Ethernet switches.

The communication between the Irinos-System and the PC has been designed for fault tolerance. In case a data packet is lost, it will be retransferred automatically. This retransmission results in a delay of the availability of measurement values. Hence it should be an exceptional case. (It has no influence on the realtime capability of the system, since the values are internally buffered.)

In order to minimize the number of retransmissions, ***it is strongly advised to use a direct connection between the Irinos-System and the PC***. Therefore the Ethernet interface of the Irinos-System is directly connected to an unused Ethernet interface of the PC. Experience has shown, that retransmissions almost never occur.

The Irinos-System has not been designed for use with routers, VPN connections, wireless connections and similar.

An integrated "cross-over detection" ensures that it does not matter, whether 1:1 or crossed Ethernet cables are used.

By factory defaults, the DHCP-Server of the Irinos-System is activated. This is a good choice for a direct connection to a PC.

Before using the Irinos-System in an IT network, the DHCP-Server must be deactivated. Therefore the Irinos-System must be connected directly to a PC. Changing the IP settings is done via the Irinos-Tool.

6.4.2 ILink cabling

Changing the ILink cabling during operation is not allowed. Very short interruptions will lead to a communication error. Afterwards the system must be restarted.

Introduction

- ILink is a bus system in line topology. Each Irinos-Box has two ILink connectors.

- Two Irinos-Boxes are connected by one ILink cable 36. At the first and the last Irinos-Box, one of the ILink connectors will remain unused. Two protection caps are delivered with each Master-Box. Use these to protect the unused connectors.

- The Irinos-System enables the required termination at both ends of the ILink cabling automatically.
  An active termination is signalled by a blue LED between both ILink connectors. At all other boxes, this LED must be off.

- In general, it does not matter whether the upper or the lower ILink connector is used. Both are technically identical.
  The only exception is, if the Master-Box 28 is not the first or the last Box of the Irinos-System. In this case the connector affects the addressing 118 of the Slave-Boxes.
  Therefore it is advised to use the Master-Box as the first or the last Box of the Irinos-System, since then no care must be taken when this Box is replaced.
  (The Irinos-Power supply can be placed at the the other side without affecting the addressing, since it is not part of the ILink communication system.)

- The maximum allowed length of the ILink cabling is 20m (between first and last Irinos-Box).

- If the power supply IR-PU50 67 is used as the first or the last Irinos-Box, the maximum cable length to the next Irinos-Box is 0,3m.

- Please note that only one Irinos-Box containing an Ethernet-

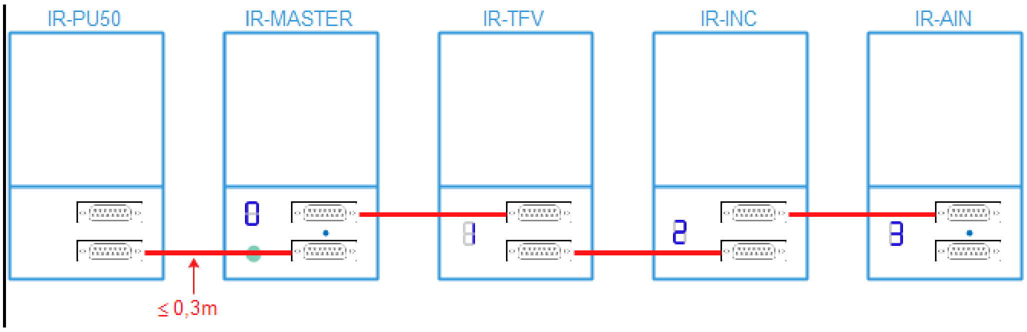
Interface is allowed per Irinos-System (i.e. only one Master-Box IR-MASTER 47 or "integrated Master 44"). Using multiple Master-Boxes, the ILink communication will not work.

- The maximum number of Irinos-Boxes allowed is 32 (including Master-Box).

Procedure

Connect neighbouring Irinos-Boxes using an ILink connection cable. Make sure that the DSUB connector is fixed using both threaded bolds.

The following figure shows an example for the ILink cabling:



Example for the ILink cabling

Protect all unused DSUB connectors using the protection caps delivered with the Master-Box.

6.4.3 IO-Bus cabling

➔ *The IO-Bus cabling is only required, if I/O-Boxes 68 need to be connected to the Master-Box IR-MASTER 47 via the IO-Bus. If the Irinos-Boxes IR-DIO 58 are used, no IO-Bus connection is required. These are connected via ILink 113.*

Introduction

- The IO-Bus is a bus system in line topology.

- The Irinos-Box IR-MASTER 47 is the start of the IO-Bus. A maximum of 4 I/O-Boxes, each with up to 16 digital in- and outputs, can be connected.

- The last I/O-Box requires termination.
  If the desktop version 68 is used, this is done automatically, signalled by its termination LED.
  If the hat / din rail version 70 is used, the termination must be activated via one of its DIP switches.

- Each I/O-Box must be addressed manually. Therefore the address must be selected via decode switch or dip switch.
  Valid addresses are 2, 4, 6, 8. Each address must only be unique.
- The maximum length of the IO-Bus cabling is 20m.

Procedure

- Select the addresses at the I/O-Boxes (-> desktop version 68 or -> hat-/din-rail version 70).
- If I/O-Boxes of the hat-/din-rail version are used: activate the termination via the DIP switch at the last box of the IO-Bus cabling.
- Connecting neighbouring I/O-Boxes using the IO-Bus cables 39.

6.4.4 Connecting inductive probes

Connect the inductive probes to the M16 connectors. Please note:

- The inductive probe type must fit the type of the Irinos-Box IR-TFV 50 .
  Connect measurement probes "Tesa halfbridge or compatible" only to an Irinos-Box IR-TFV-8-**TESA**-...
  Connect measurement probes "Knäbel IET" only to an Irinos-Box IR-TFV-8-**IET**-...
- Make sure to lock all connectors. This ensures longer connector lifetime.
- Use protection caps for unused input channels. These are available separately.

6.4.5 Connecting incremental encoders

Connect the incremental encoders (1Vpp or TTL/RS422) to the DSUB receptables. Please note:

The same connector type is used for the ILink interface 74 and the incremental encoders 96.

The pinning of the ILink connector has been chosen to avoid any damage of incremental encoders with standard pinning, if they are plugged into the ILink interface. For encoders with special pinning, this may be different.

Only use the appropriate connectors for the incremental encoders.

All measurement inputs are pre-configured for 1Vpp or TTL/RS422. If the other type is used, the measurement will not work properly. However, damage to the input is not possible.

Use a protection cap for all unused input channels. These are available separately.

6.4.6 Connecting analogue sensors

Connect the analogue sensors to the dedicated measurement inputs.

Use protection caps for all unused measurement inputs. These are available separately.

6.4.7 Connecting digital in- and outputs

Connect all digital in- and outputs to the dedicated connectors or terminal blocks.

Use protection caps for unused connectors. These are available separately.

Setup

7 Setup

7.1 First steps

The Irinos-System has been designed to get started quickly without the need for configuration. Exceptions are:

- Configuration[120] of the network settings, if the Default settings cannot be used.
- Configuration of the incremental input type, if the pre-configured type is not appropriate (can be done via the Irinos-Tool or the MscDll).

After mounting and cabling, the Irinos-System can be switched on immediately.

7.2 Box addressing

All Irinos-Boxes are automatically enumerated at startup. The Master-Box[28] always has the address 0. The addresses for the Slave-Boxes[28] are incremented according to the order in the ILink cabling. The order of the measurement inputs and digital in- and outputs depends on the order of the boxes.

The following figures provide some examples for Box addressing:

Examples for Box addressing

Position of the Master-Box

If the Master-Box is not the first or the last Box of the Irinos-System (i.e. Slave-Boxes are connected to both [ILink connectors](44) of the Master-Box), the Boxes are enumerated as follows:

First all Boxes, which are connected to the upper ILink connector, are enumerated.

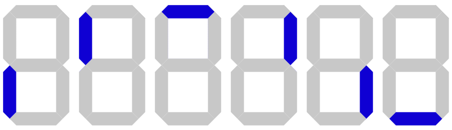
Afterwards all Boxes, which are connected to the lower ILink connector, are enumerated.

In case the cables are interchanged, the box addressing and the order of the measurement channels changes. Therefore it is advised to use the Master-Box as the first or the last Box of the Irinos-System, since then no care must be taken when this Box is replaced.
(The Irinos-Power supply can be placed at the the other side without affecting the addressing, since it is not part of the ILink communication system.)

Enumeration process

The duration of the enumeration process depends on the number of Irinos-Boxes connected. Typically it takes only a few seconds. If many Irinos-Boxes are used, it may last up to 15s.

Prior to being enumerated, the elements of the 7-digit display46 turn clockwise:



Afterwards the Box number is shown in the 7-digit display. An exception is the first Box having a 7-digit display: it first shows the IP configuration. Depending on the configuration, this lasts a few seconds.

Termination

During the enumeration process, the first and the last Irinos-Box of the ILink cabling are automatically detected and terminated. This is signalled by a blue LED between both ILink connectors at these Boxes. The LED must be off at all other Boxes.

Checks

The following checks should be done after startup:

- All slave Boxes must display a valid address.
- The termination LED must be turned on at the first and the last Irinos-Box of the ILink cabling.

7.3 Network configuration

The Master-Box of the Irinos-System provides an integrated DHCP-Server, which is activated at delivery state. The ***IP address of the Irinos-Systems is 192.168.3.99***, the subnet mask is 255.255.255.0. If the Ethernet-Interface of the PC is configured as "DHCP-Client", a valid IP address is assigned to it (range 192.168.3.100 to 192.168.3.254). No further network configuration is required.

In delivery condition the DHCP-Server of the Irinos-System is activated. This is a good choice for a direct connection to a PC.

> Before using the Irinos-System in an IT network, the DHCP-Server must be deactivated. Therefore the Irinos-System must be connected directly to a PC. Changing the IP settings is done via the Irinos-Tool.

In case the DHCP functionality is not preferred, there are two options:

a) The DHCP-Server at the Master-Box remains activated. Fixed IP settings are assigned to the network port of the PC, for example:
IP address: 192.168.3.98
Subnet mask: 255.255.255.0

b) The DHCP-Server of the Master-Box is deactivated via the Irinos-Tool. Almost any valid IP settings can be set. Fixed settings are assigned to the network port of the PC. For further information, take a look at the documentation of the Irinos-Tool.

For checking the IP connection, open a webbrowser at the PC and enter the IP address of the Irinos-Box into the address line. If the connection works properly, the website of the Irinos-System is shown (displaying measurement values). In case any problems appear, take a look at the first aid section.

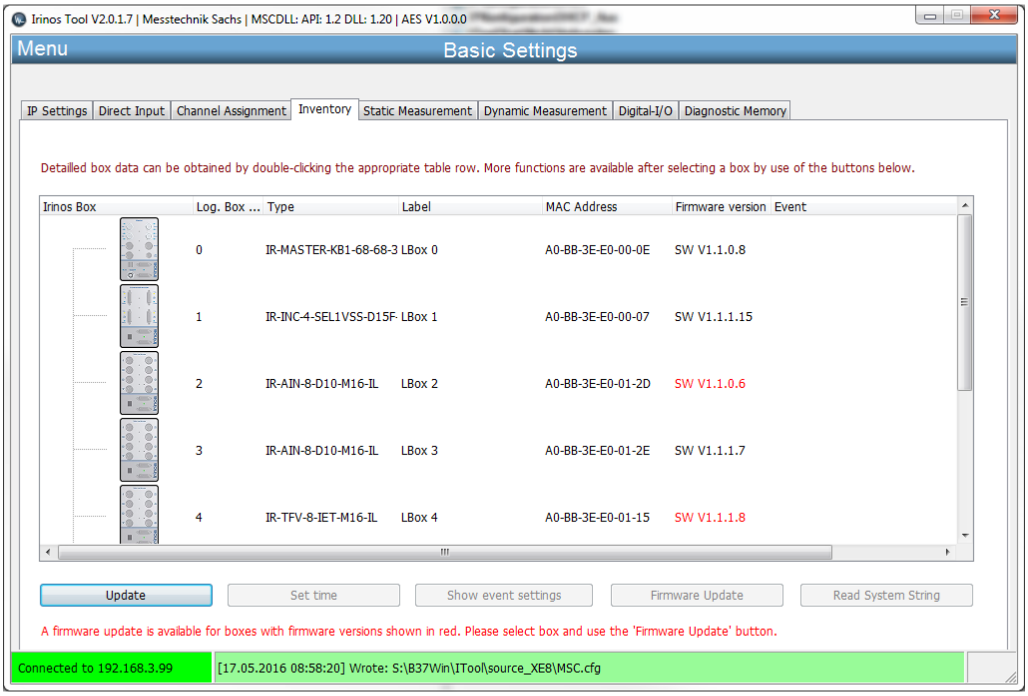
7.4 Irinos-Tool

The Irinos-Tool has been designed as a multifunctional configuration and test tool for the Irinos-System. The scope of functions includes:

- Changing the network settings of the Irinos-System.
- Generating the configuration file Msc.cfg for the MscDll.dll.
- Listing the available Irinos-Boxes and measurement channels.
- Configuration of the measurement channels for incremental encoders (1Vpp or TTL/RS422).
- Diagnostic functions for incremental encoder signals (1Vpp).
- Displaying the static measurement values.
- Performing firmware updates.
- Reading the content of the diagnostic memory 190 (and saving it to a file).

Further information is available in the documentation of the Irinos-Tool.

It is advised to put a copy of the Irinos-Tool on the measurement PC. No license fees apply, as long as it is only used together with the Irinos-System.



Irinos-Tool

7.5 Web-Server

The Irinos-System is equipped with an integrated Web-Server to be used for setup and diagnostics. It can be accessed from almost any standard web-browser, e.g. InternetExplorer or Firefox. Therefore the IP address of the Irinos-System must be entered into the address line of the browser (at delivery: 192.168.3.99).

The web-server has been tested with various web-browsers. Because of the different interpretation of standards, it cannot be guaranteed with all browsers.

4 websites are available:

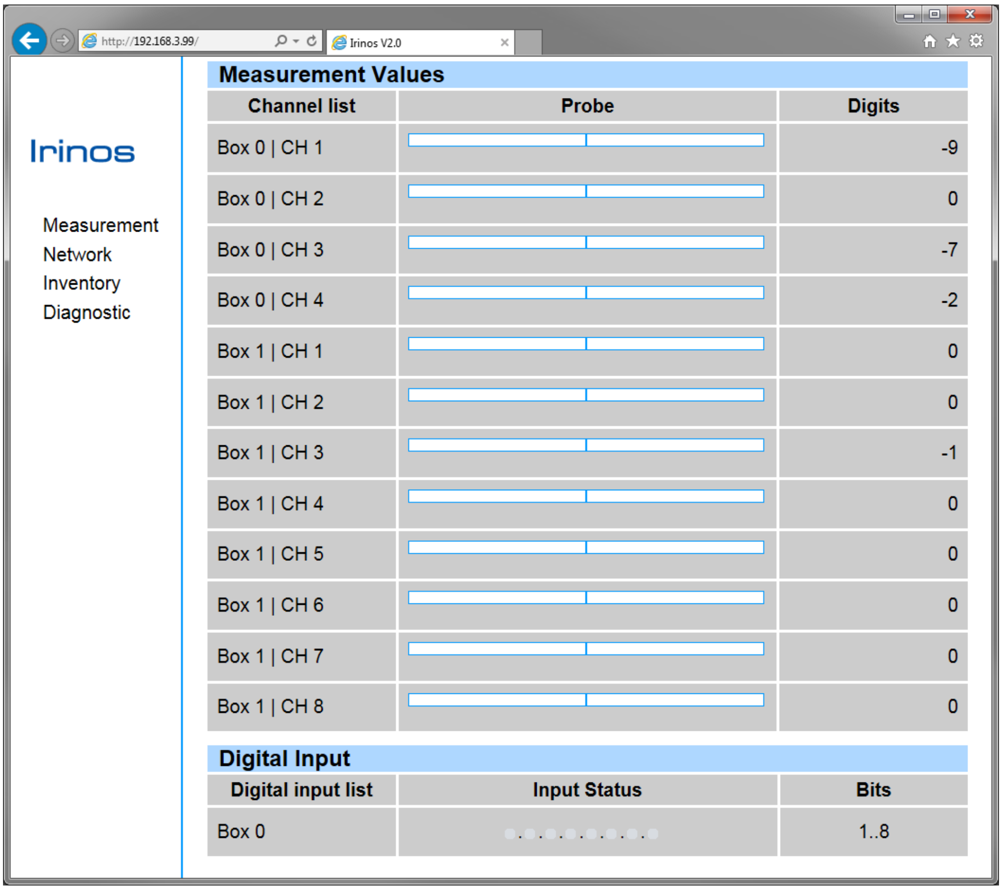
Measurement (measurement values)

The website "Measurement" shows the current measurement values of all measurement channels and the state of the digital inputs (live view, update rate about 4 Hz).

This allows:

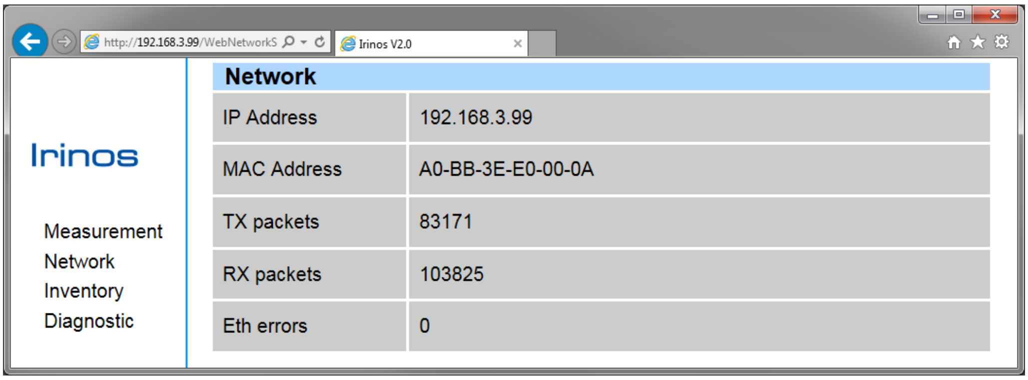
○ Adjusting the measurement probes without a measurement software.

○ Comparing the measurement values delivered by the Irinos-System to those shown in the measurement software.

○ 

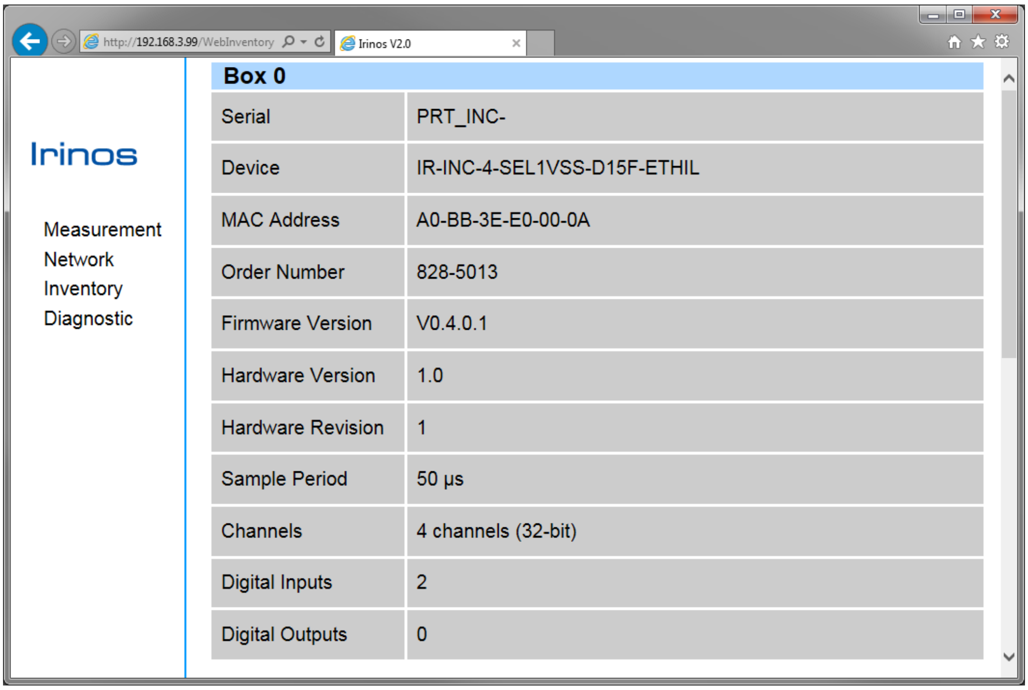
Network

The website "Network" shows the configuration and status of the network connection to the PC:



Inventory

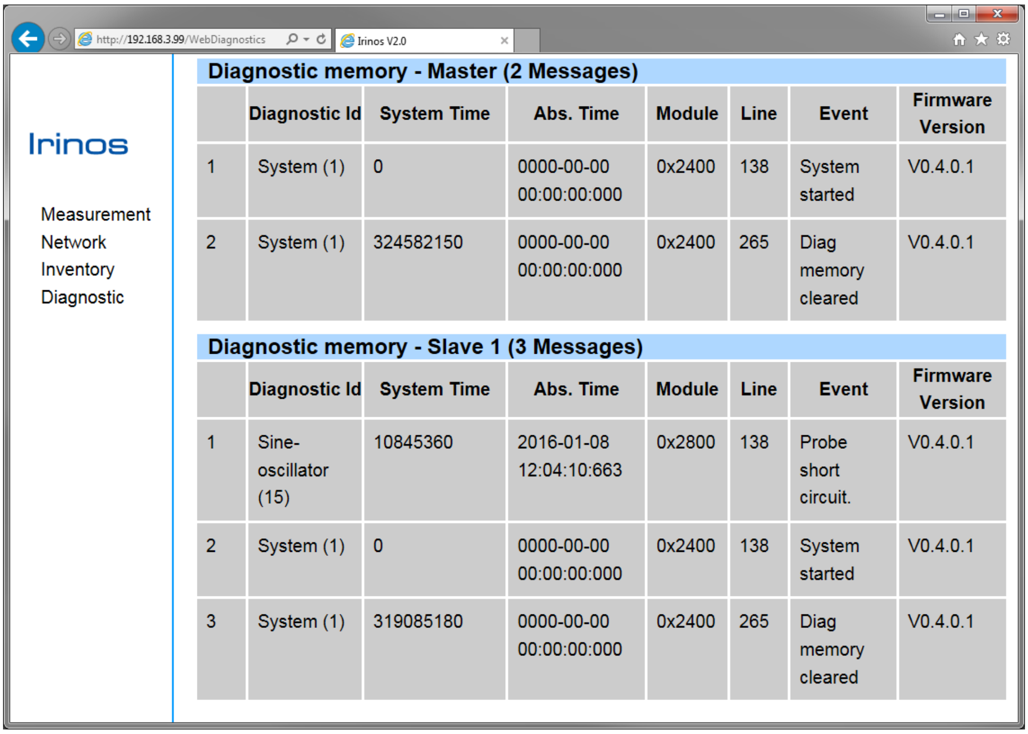
The website "Inventory" provides an overview of all Irinos-Boxes available in the Irinos-System. Further some detailed information is available about the Irinos-Boxes:



Information	Example	Description
Serial	I001234	Serial number of the Irinos-Box
Device	IR-INC-4-SEL1VSS-D15F-ETHIL	Description string of the Irinos-Box
MAC Address	A0-BB-3E-E0-00-0A	Unique MAC-ID of the Irinos-Box.
Order Number	828-5013	Order number. 32 . This number always starts with 8.
Firmware Version	V1.0.0.11	Firmware version.
Hardware Version	V1.1	Hardware version.
Hardware Revision	1	Compatibility code between hardware and firmware.
Sample Period	50 µs	Internal sample rate of the Irinos-Box.
Channels	4 channels (32-bit)	Number of measurement channels and internal data type.
Digital Inputs	2	Number of available digital inputs.
Digital Outputs	0	Number of available digital outputs.

Diagnostic (Diagnostic memory)

The website "Diagnostic" shows the contents of the diagnostic memories of all Irinos-Boxes:



Diagnostic memory - Master (2 Messages)

	Diagnostic Id	System Time	Abs. Time	Module	Line	Event	Firmware Version
1	System (1)	0	0000-00-00 00:00:00:000	0x2400	138	System started	V0.4.0.1
2	System (1)	324582150	0000-00-00 00:00:00:000	0x2400	265	Diag memory cleared	V0.4.0.1

Diagnostic memory - Slave 1 (3 Messages)

	Diagnostic Id	System Time	Abs. Time	Module	Line	Event	Firmware Version
1	Sine-oscillator (15)	10845360	2016-01-08 12:04:10:663	0x2800	138	Probe short circuit.	V0.4.0.1
2	System (1)	0	0000-00-00 00:00:00:000	0x2400	138	System started	V0.4.0.1
3	System (1)	319085180	0000-00-00 00:00:00:000	0x2400	265	Diag memory cleared	V0.4.0.1

Column	Example	Description
Diagnostic-Id	Sine-oscillator (15)	Event type 180 .
System Time	15364470	ILink-Time 28 (unit: µs) since startup of the Irinos-System. This internal time is identical at all boxes of an Irinos-System.
Abs. Time	2016-01-08 12:04:10:663	Date and time of the event (only available if set by measurement software 170 before). Year-Month-Day Hour:Minute:Second: Millisecond
Module	0x2800	Additional information for manufacturer support.
Line	138	
Event	Probe short circuit.	Additional information for the event.
Firmware-Version	V0.4.0.1	Firmware-Version at the time the event occured.

Measurement / Control via MscDll

8 Measurement / Control via MscDll

➔For further information, see the MscDll reference manual.

➔Demo applications in various programming languages are available.

8.1 Introduction

The MscDll is the link between the application software (measurement software) and the Irinos-System. It provides a common interface for reading the measurement values and status information and for system parameterization.

The MscDll directly uses the Windows API functions for the IP-based communication and for the thread-management and timing. It can be used with different Windows versions. It has been successfully tested with Windows XP (32 Bit) and Windows 7, 8 and 10 (32 & 64 Bit).
Inside the DLL a separate thread is running, controlling all the communication to the Irinos-System. The DLL interface functions provide data to this thread and vice versa.

Communication to the Irinos-System is based on UDP/IP. The DLL automatically retransmits a data packet, if it has been lost. A direct ethernet connection between the Irinos-System and the PC is adviced. Complex network structures, e.g. routing, tunneling, VPN, etc. are not supported due to timing efficiency.

In this chapter, the procedures and specialities of the application of the MscDll are described. A detailled specification of the functions and opcodes is available in the reference manual.

Even though it is technically possible to connect multiple Irinos-Systems via the same DLL, in reality this makes no sense because of the modular Irinos concept. Thus in the following chapters only systems with one connection are discussed.

8.2 Basics

According to the basic idea of the Irinos-System, all realtime functionality is implemented inside the Irinos-System. The PC or the communication channels do not need to fill any realtime requirements. Therefore the Irinos-System can be used with standard windows installations. Special extensions, like for example realtime kernel extensions, are not required.

In order to achieve this, all realtime data is buffered inside the Irinos-

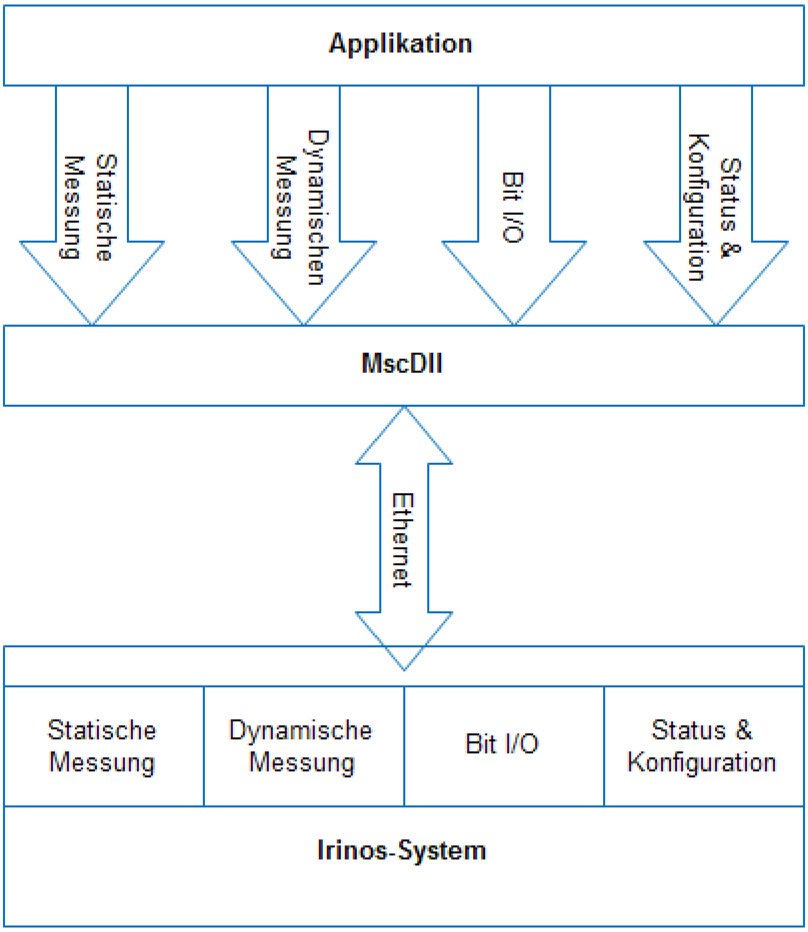
Boxes. In reality, there are no long interruptions between the Irinos-System and the PC. However, depending on the parametrization, the PC can even pause for a few hundred milliseconds without having any data loss.

The data exchange between the PC and the Irinos-System is controlled by the MscDll automatically. The only thing the application needs to do, is a one-time parametrization before connecting to the Irinos-System.

Data exchange is always started by the PC. A request message frame is send to the Irinos-System by the MscDll. Afterwards it waits until a response message frame has been received. The Irinos-System never sends data without a request. This cycle is started in cyclic intervals, defined by the "send-period".

Timeouts are used to supervise any packet loss. If a packet gets lost, the data transfer is restarted after that timeout. Only if multiple retransmissions fail, the communication is regarded as broken.

A message frame can contain different types of data, e.g. measurement values, Bit I/Os and hardware status (see figure below). Packing and unpacking that data is done by the MscDll. The application is neither able nor required to influence this.



Parallel data exchange via the MscDll

In traditional systems, data must be read from one or multiple measurement cards by using their APIs. Since the measurement cards are often produced by different manufacturers, different APIs with different strategies must be implemented into the measurement software. Using the Irinos-System, all data is available via a single interface: the MscDll. The only thing which varies is the number of measurement channels and of the in- and output bits.

The most important difference to traditional systems is reading the measurement values:
The MscDll transfers all measurement values in background. They are not directly read from the hardware. Instead the MscDll updates them in a predefined interval. The application software is informed each time new data is available.

8.3 Static vs. dynamic measurement

The MscDll distinguishes between two types of measurement: Static measurement and dynamic measurement. Both can be used in parallel. Following a short introduction shall be given. More information is provided in the respective chapters (-> static measurement 136 / -> dynamic measurement 141).

Using static measurement, measurement values are updated continuously. The measurement values are neither synchronized nor transferred in realtime. The typical update rate depends on the number of boxes and the configuration. The range is 30-100 Hz per channel. For many measurement applications, this is the perfect choice.

The implementation of the static measurement is very easy. Once it has been started via the MscDll, measurement values are periodically requested from the Irinos-System. These are copied into a buffer, which has been provided by the application before. If new measurement values have "arrived", the application is informed. These can be read by the application at any time.

Often the static measurement is also used to achieve an online-view of the current measurement values.

The dynamic measurement is used for synchronized realtime data acquisition with a maximum sample rate of 10.000 samples/s per channel.

A dynamic measurement is always limited in time. Typically its duration is equal to or less than one measurement cycle, which correlates to a few seconds. Each measurement value is either related to a specific time or position (e.g. angle). While the measurement is active, all measurement values are stored in the internal memory of the Irinos-System. During and after the dynamic measurement, the values are transferred to the application software via the MscDll.

Dynamic measurement is for example used to measure the form of an object (e.g. roundness). Usually it only uses a limited number of measurement channels. A simple roundness measurement may require even only one measurement channel. The maximum number of measurement channels, which can be used for a dynamic measurement simultaneously, is limited to 32.

Before the start of a dynamic measurement, various parameters must be defined, like the measurement channels used, the start time, the number of values measured, etc.

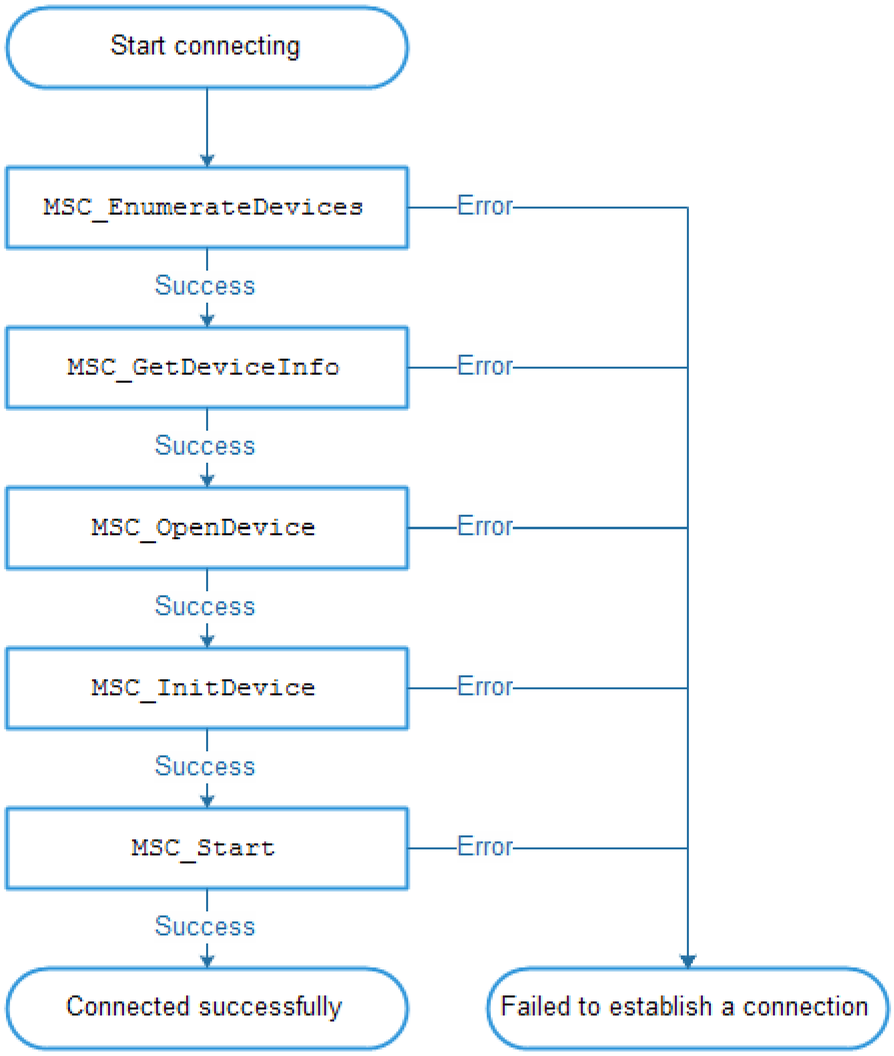
8.4 Integrating the DLL / Configuration

The steps necessary to integrate the MscDll into the measurement software depend on the programming language and the development environment. For many programming languages, ready-to-use example applications are available. These can and should be used as a starting point.

The MscDll always requires the configuration file Msc.cfg, which contains the network settings of the Irinos-System. It must be located in the same directory as the MscDll.dll.
This configuration file is a text-file, which can be edited with any text editor. Typically only the IP address of the Irinos-System must be exchanged. Alternatively the configuration file can be generated using the Irinos-Tool.

8.5 Connecting to the Irinos-System

Establishing the connection to the Irinos-System is typically done at the start of the measurement application. The connection remains active, until the application is closed. The following figure shows necessary steps for establishing a connection:



Steps for "Establishing a connection with the MscDll"

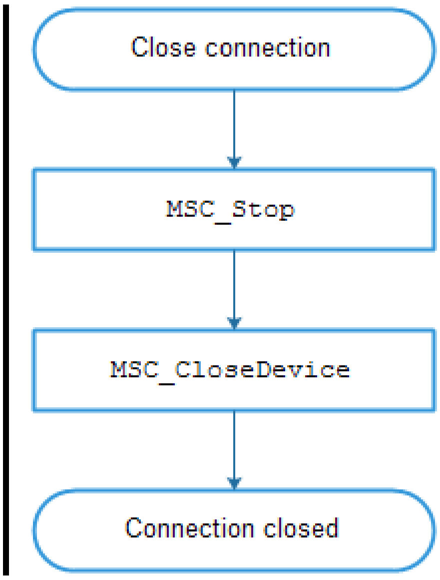
- The function `MSC_EnumerateDevices` lists all the Irinos-Systems available. This takes about 2-3 seconds. The number of systems found will be returned. Usually only one Irinos-System is connected, which results in a return value of 1. If no Irinos-System can be found, 0 is returned.
- The function `MSC_GetDeviceInfo` returns a string, which identifies the Irinos-System.
- The function `MSC_OpenDevice` opens the connection to the Irinos-System.
- The connection is initialized using the function `MSC_InitDevice`.
- The function `MSC_Start` starts the data exchange between the MscDll and the Irinos-System. Various timeouts are defined within this function call. Additionally the send-period is defined, which determines how often data is updated. The reference manual shows some examples.
  The send-period determines, how often static measurement values are updated. For most applications a value of 1 (= 1ms) is appropriate.

In case establishing the connection fails, typically either an invalid network configuration or an invalid IP address in the configuration file MSC.cfg is the reason.

It is adviced to check the system structure after the connection has been established.

8.6 Closing the connection

A connection to the Irinos-System must always be closed before the application (measurement software) is closed. Otherwise an exception can occur. The following figure shows the steps necessary for closing the connection:

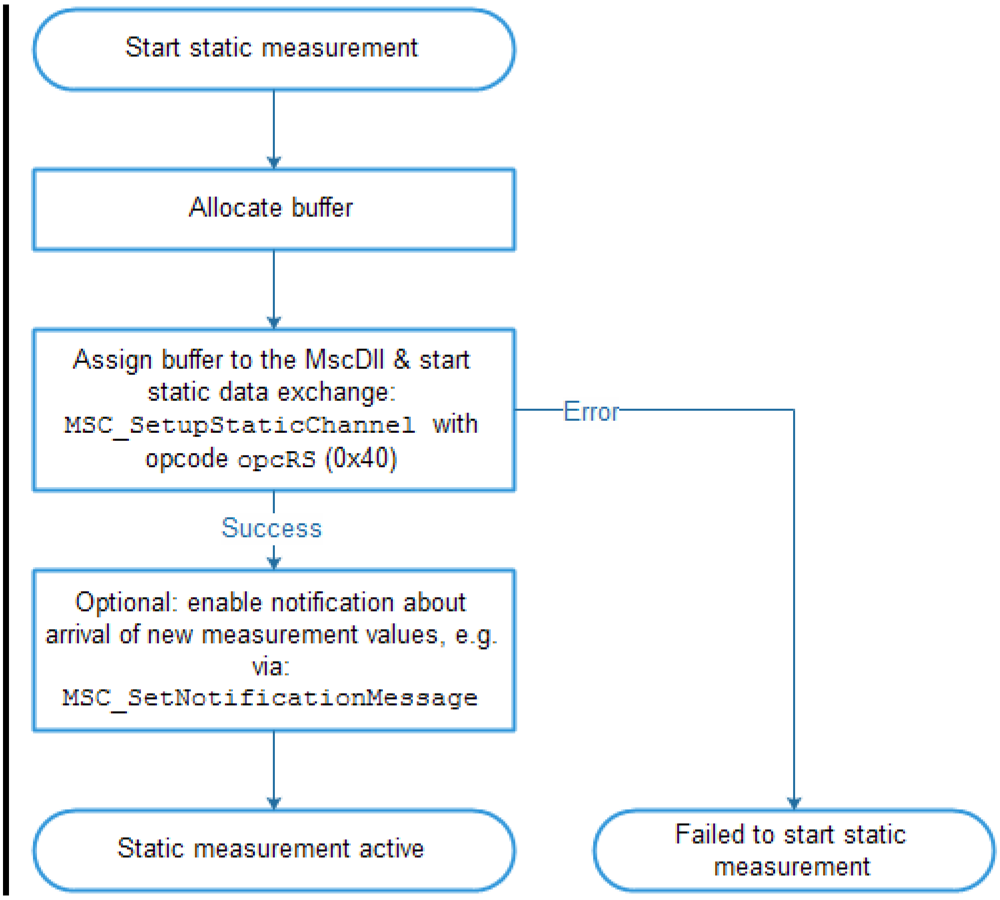


Steps for "closing the connection to the MscDll"

- The function `MSC_Stop` stops the data exchange to the Irinos-System.
- The function `MSC_CloseDevice` closes the connection to the Irinos-System.

8.7 Static measurement

If only dynamic measurement data is required, there is theoretically no need to start static measurement. However, in reality static measurement is always started, e.g. for a live view of the measurement values while a dynamic measurement 141 is active. The following figure shows the required steps for starting a static measurement:



Steps for "starting static measurement"

- The function `MSC_SetupStaticChannel` together with the Opcode `opcRS` (0x40) starts the cyclic transfer of static measurement values through the MscDll. A buffer must be provided to this function. The MscDll will place the static measurement values into this buffer. All measurement values are stored as „32 Bit signed integer" – values (little ended), independent of the data type of the respective measurement channel.

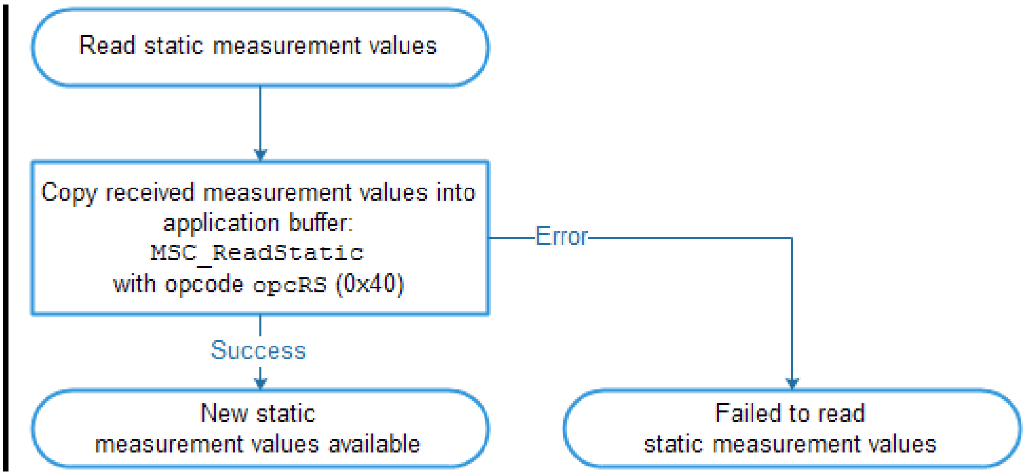
  The required buffer size depends on the maximum number of measurement channels. Each channel required 4 Bytes, e.g.:
  8 measurement channels -> 32 Bytes
  64 measurement channels -> 256 Bytes
  Since plenty of memory is available on a PC, it is good practice to allocate 1024 bytes of memory for a maximum of 256 measurement channels.

The following table shows the buffer layout:

Bytes (Hex)	Size	Content
0x00 .. 0x03	4 Bytes	Measurement value of channel 1
0x04 .. 0x07	4 Bytes	Measurement value of channel 2
0x08 .. 0x0B	4 Bytes	Measurement value of channel 3
0x0C .. 0x0F	4 Bytes	Measurement value of channel 4
0x10 .. 0x13	4 Bytes	Measurement value of channel 5
0x14 .. 0x17	4 Bytes	Measurement value of channel 6
0x18 .. 0x1B	4 Bytes	Measurement value of channel 7
0x1C .. 0x1F	4 Bytes	Measurement value of channel 8
0x20 .. 0x23	4 Bytes	Measurement value of channel 9
etc.		
	4 Bytes	Measurement value of channel n

- After calling the function MSC_SetNotificationMessage (alternatively MSC_SetNotificationEvent or MSC_SetNotificationCallback) together with the opcode opcRS (0x40), the application is informed every time new static measurement values are available. It is up to the application whether it uses or ignores them.
  Using notifications is recommended, but not required.

- The measurement values are first stored into an internal buffer of the MscDll. In order to copy them into the buffer of the application, the function MSC_ReadStatic together with the opcode opcRS (0x40) must be called (see following figure). This procedure must be repeated every time new measurement values have been received.



Steps for "copying static measurement values into the buffer of the application"

Update-Rate

The update rate for measurement values depends on two factors:

○ The update-rate of the MscDll.

○ The internal update-rate of the Irinos-System.

The update-rate of the MscDll is determined by the send-period, which has been defined during connection establishment with MSC_Start. If the send-period is 30ms, the update-rate is about 30 updates/s; 20ms send-period result in 50 updates/s. The highest achievable update-rate with a send-period of 1ms is a little bit more than 100 updates/s.

The internal update-rate of the Irinos-System depends on whether a dynamic measurement is active or not.
If a dynamic measurement is active, the update-rate is always about 30 updates/s.
If dynamic measurement is inactive, the update-rate is about 100 updates/s using up to 8 Irinos-Boxes. If 32 Irinos-Boxes are used, the internal update-rate is about 30 updates/s.

The following table shows typical update rates for various combinations of send-period and "number of Irinos-Boxes", if no dynamic measurement is active:

Send-period MscDll	Update-rate MscDll	Number of Irinos-Boxes	Internal Update-rate	Approximate number of achievable updates/s per measurement channel
[ms]	**Updates/s**		**Updates/s**	**Updates/s**
30	30	≤ 8	100	**30**
30	30	12	80	**30**
30	30	16	60	**30**
30	30	24	45	**30**
30	30	32	30	**30**
20	50	≤ 8	100	**50**
20	50	12	80	**50**
20	50	16	60	**50**
20	50	24	45	**45**
20	50	32	30	**30**
1	100	≤ 8	100	**100**
1	100	12	80	**80**
1	100	16	60	**60**
1	100	24	45	**45**
1	100	32	30	**30**

➔ ***For most applications a send-period of 1ms is recommended.***

The update-rate is independent of the number of channels. It does not matter, whether 7, 23 or 41 static measurement channels are used.

Traditional systems often provide an update-rate, which depends on the number of measurement channels. Thus it is far lower than it appears at first glance. This limitation does not apply to the Irinos-System.

Please not that all update-rates above are estimated values but not guaranteed. In case a data frame gets lost, a longer break between two updates may occur. If a guaranteed sample-rate is required, dynamic measurement 141 must be used.

8.8 Dynamic measurement

➔ Tip: various examples 151 are available for using dynamic measurement.

8.8.1 General

A dynamic measurement is always limited in time. It is usually started before the beginning of a dynamic measurement process (e.g. before starting a motor). The maximum time allowed is 1 hour, which is just a theoretical value. Usually it lasts less than a minute.

There are **two types of dynamic measurement**:

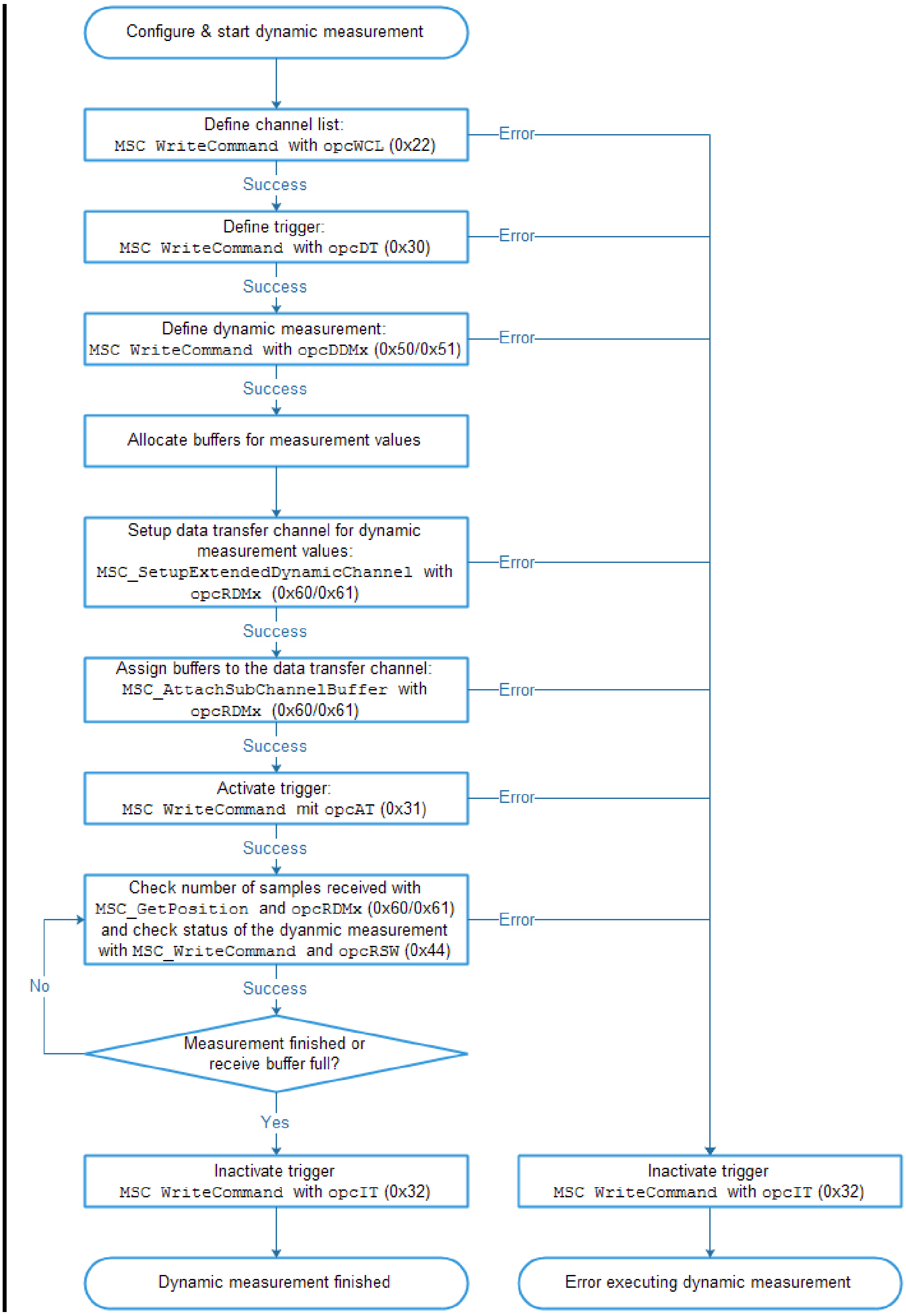
- Using **time-triggered** dynamic measurement, all selected measurement channels are sampled simultanesouly in equidistant time-spans. It is possible to sample 10.000 values/s per channel.
- Using **position-triggered** dynamic measurement, all measurement values are sampled simultaneously in pre-defined position-distances. This requires a measurement channel, which provides the position information. For example an incremental

encoder can be used for concentricity measurement with 360 measurement values / rotation.
The measurement channel, which provides the position information, is ***internally sampled at 1.000 Hz***.

Position-triggered measurement is a good choice, if the speed of the work piece varies during measurement, e.g. if a rotary disc is rotated by hand or with an uncontrolled motor.

Modern servo drives usually have a very accurate speed control. Thus constant position- or angle-distances can be achieved. Therefore often time-triggered measurement can be used in such applications.

Before starting a dynamic measurement, it must be defined. The following figure shows the typical steps for configuring, starting, running and finishing a dynamic measurement:



Steps for performing a dynamic measurement

a) Using the function `MSC_WriteCommand` together with the opcode `opcWCL` (0x22), a channel list is transferred to the Irinos-System. The channel list contains a list of all measurement channels, which shall be used for the dynamic measurement. 32 channels can be used at maximum.

b) A trigger is defined using the function `MSC_WriteCommand` together with the opcode `opcDT` (0x30) The trigger definition contains:

- Whether it is a time- or position-triggered dynamic measurement.
- The name of the measurement channel, which provides the position information (only used for position-triggered dynamic measurement).
- The start time/position of the dynamic measurement.
- The time-period respectively position-distance between two measurement values.
- The end time/position of the dynamic measurement (optionally).

c) The dynamic measurement itself is defined using the function `MSC_WriteCommand` together with the opcode `opcDDMx` (0x50/0x51). The definition contains:

- The channel list, which shall be used for the dynamic measurement.
- The trigger configuration to be used for the dynamic measurement (trigger number).
- The maximum number of sampled measurement values per channel.

d) A data transfer channel for reading the dynamic measurement values is set up using the function `MSC_SetupExtendedDynamicChannel` together with the opcode `opcRDMx` (0x60/0x61). This data transfer channel is required to transfer the measurement values from the Irinos-System to the PC / application.

e) A buffer for storing the dynamic measurement values must be provided to the MscDll using the function `MSC_AttachSubChannelBuffer` together with the opcode `opcRDMx` (0x60/0x61).
For each measurement channel a buffer must be allocated and provided to the MscDll via this function. If for example 5 measurement channels are used, then 5 buffers must be allocated and assigned to the MscDll via this function.
The size of all buffers is identical. It depends on the maximum number of samples, which can be expected for the dynamic measurement. Similar to the static measurement, all measurement values are stored as 32 bit integer values.

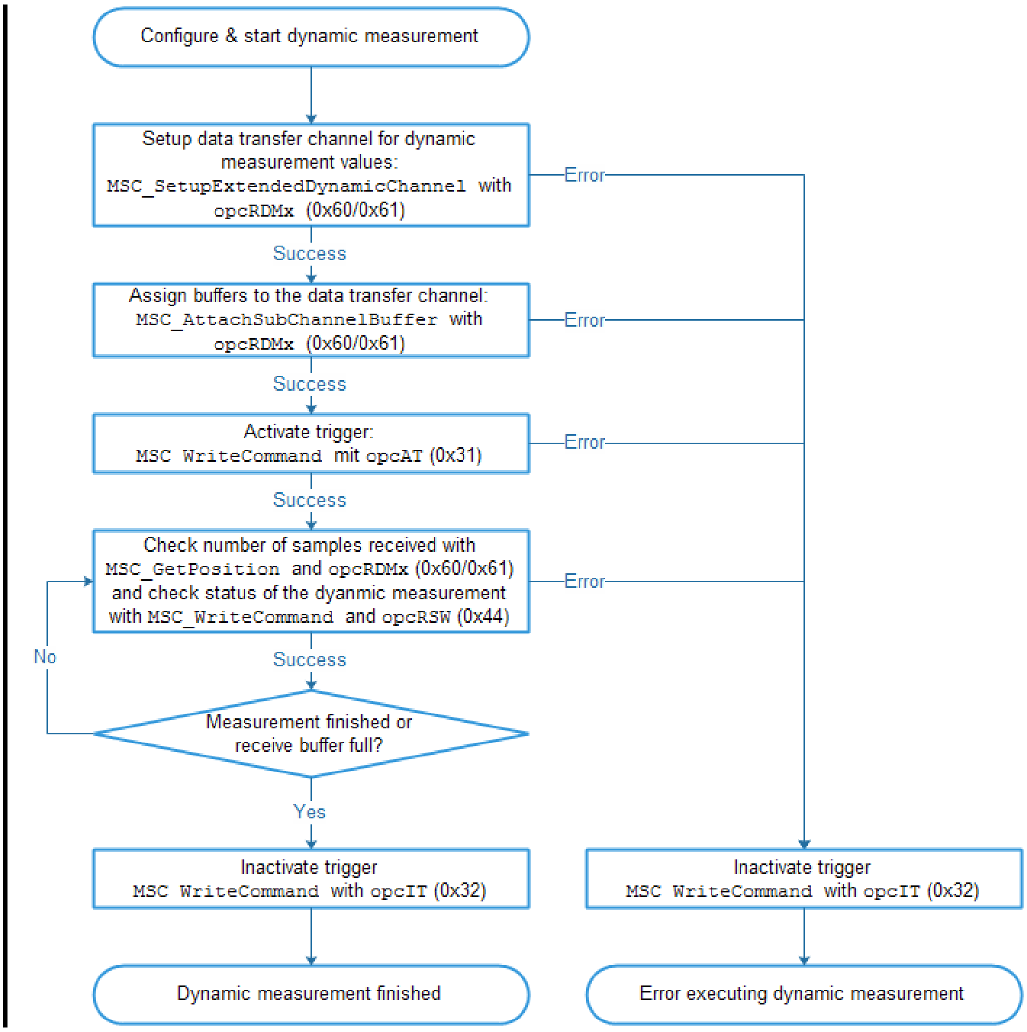
f) The dynamic measurement is started using the function `MSC_WriteCommand` together with the opcode `opcAT` (0x31). This will start the trigger, which is the same as starting the dynamic measurement.

g) The function `MSC_GetPosition` together with the opcode `opcRDMx` (0x60/0x61) returns the number of measurement values already received.
Using the function `MSC_WriteCommand` together with the opcode `opcRSW` (0x44) provides a statusword with status information about the dynamic measurement. One of the bits for example is set, if the dynamic measurement is finished.

Typically one of these functions is called cyclically, while a dynamic measurement is active.
It depends on the application, when a dynamic measurement is defined as finished. The safest way is using the Bits of the statusword.

h) It is recommended do inactivate the trigger after the dynamic measurement using the function `MSC_WriteCommand` together with the opcode `opcIT` (0x32). This ensures that a newly defined dynamic measurement does not start prior to the defined start time.

If an identical dynamic measurement is repeated, it is possible to omit the steps a) to d). The follwing figure shows the steps for repeating a dynamic measurement:



Steps for repeating a dynamic measurement (with identical configuration)

8.8.2 Further notes

Two dynamic measurements simultaneously

Two dynamic measurements can be executed simultaneously. These are distinquished by the opcodes:

- Dynamic measurement 1: opcDDM1 (0x50) and opcRDM1 (0x60)
- Dynamic measurement 2: opcDDM2 (0x51) und opcRDM2 (0x61)

If two time-triggered dynamic measurements are executed simultaneously, the sum of the sample-rates is limited to 5.000 measurement values / s. Examples for valid sample-rates are:

- Time-triggered dynamic measurement 1 with a sample period of 0,3ms (=3.333,33 values/s) and time-triggered dynamic measurement 2 with a sample period of 0,75ms (=1.333,33 values/s).
  Sum: 4.666,66 updates/s.
- Time-triggered dynamic measurement 1 and 2, each with a sample period of 0,5ms (=2.000 values/s).
  Sum: 4.000 updates/s

If a position-triggered and a time-triggered dynamic measurement are executed simultaneously, the shortest sample period allowed for the time-triggered measurement is 1ms (=1.000 values/s).

In case both dynamic measurements are not running longer than 60 seconds, the shortest sample period allowed for the time-triggered measurement is 0,25ms (=4.000 values/s).

Two position-triggered dynamic measurements can be executed simultaneously.

Sample period of the time-triggered dynamic measurement

The sample period of the time-triggered dynamic measurement must be a multiple of the sample-period of all Irinos-Boxes involved. All Irinos-Boxes currently available support a sample period of 0,05ms = 50µs. Examples for valid sample-periods are:

sample period	Samples/s
0,1 ms	10.000
0,15 ms	6.666,666
0,2 ms	5.000
0,25 ms	4.000
0.4 ms	2.500
0.5 ms	2.000
1 ms	1.000
2 ms	500
4 ms	250
10 ms	100

Number of measurement channels vs. sample rate

All sample periods are independent of the number of channels. It does not matter whether 1, 5 or 17 measurement channels are used for a dynamic measurement. The only limitation is the maximum number of measurement channels (32).

Memory size / maximum number of measurement values

Each dynamic measurement has an internal buffer, which allows storing 100.000 measurement values / channel. If this buffer is full, the dynamic measurement is automatically stopped.

If measurement values are read by the MscDll while a dynamic measurement is active, these are removed from the buffer and more than 100.000 measurement values per channel can be stored. Since the speed of reading the values depends on various factors, no maximum value can be guaranteed.

10 channel lists

A maximum of 10 different channel lists are available inside the Irinos-System. Each dynamic measurement can use any of these channel lists.

2 Trigger configurations

A maximum of two triggers can be configured. Each dynamic measurement can use any of these triggers. It is also possible to use the same trigger-configuration for both dynamic measurements. In this case, they are started and stopped together using the opcodes `opcAT` (0x31) and `opcIT` (0x32).

Configuration during dynamic measurement

A copy of the assigned channel list and trigger configuration is made at the beginning of a dynamic measurement. If a channel list or a trigger configuration is changed while a dynamic measurement is active, this has no effect on the dynamic measurement.

8.8.3 Buffer (array) with dynamic measurement values

The MscDll stores all dynamic measurement values in buffers, which must be provided by the application. For each measurement channel used in a dynamic measurement, a separate buffer is required. The channel list determines the number of measurement channels.

Example:

For the dynamic measurement 1, the channel list 1 is used. It is defined as follows:
`#1;T2;T4;T5;T6;T17#`

-> The channel list contains 5 measurement channels. Therefore 5 buffers for measurement values are required.

The size of the buffers depends on the maximum number of measurement values (-> maximum number of samples), which are required for the dynamic measurement. All measurement values are stored as "32 Bit integer" values (little endian), independent of the data type of the respective measurement channel. Thus follows:

o All buffers have the same size.

- Each measurement value required 4 bytes.

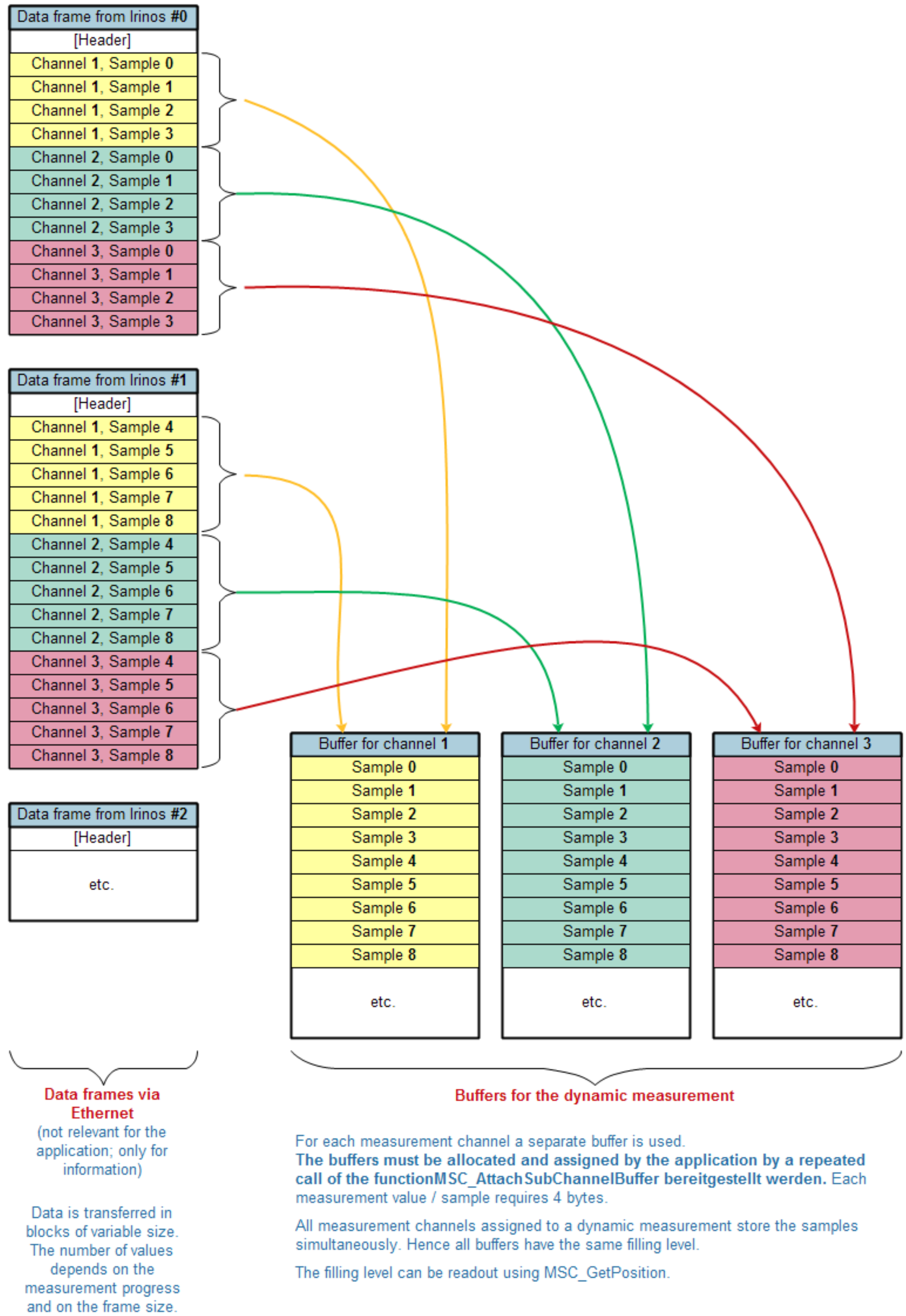
Example:

A time-controlled dynamic measurement will take 2,5 seconds. The sample rate is 4.000 samples/s. Thus the required buffer size is:

4000 samples/s * 2,5s * 4 bytes/sample = 40000 bytes.

The measurement values are always transferred in related blocks to the MscDll. The MscDll stores them in the buffers, which have been allocated by the application (see following figure). This means the demultiplexing is done by the MscDll. The application can access the measurement values in the order of samples.

All buffers have the same filling level, i.e. the filling level returned by the function `MSC_GetPosition` applies to all buffers of a dynamic measurement.



Buffers with dynamic measurement values

8.9 Examples for dynamic measurement

The following examples show the use of the dynamic measurement. Together with the demo applications, these are a starting point for the implementation of the dynamic measurement into the measurement application.

For better readability, no error handling is done in the code examples. Nevertheless it should be implemented into the application.
A pseudo-code is used for the examples. It can be transferred easily into various programming languages.

All examples use the function WriteCommandStr, which sends and receives an ASCII-String via the Dll function `MSC_WriteCommand`. It can also be found in the demo applications.

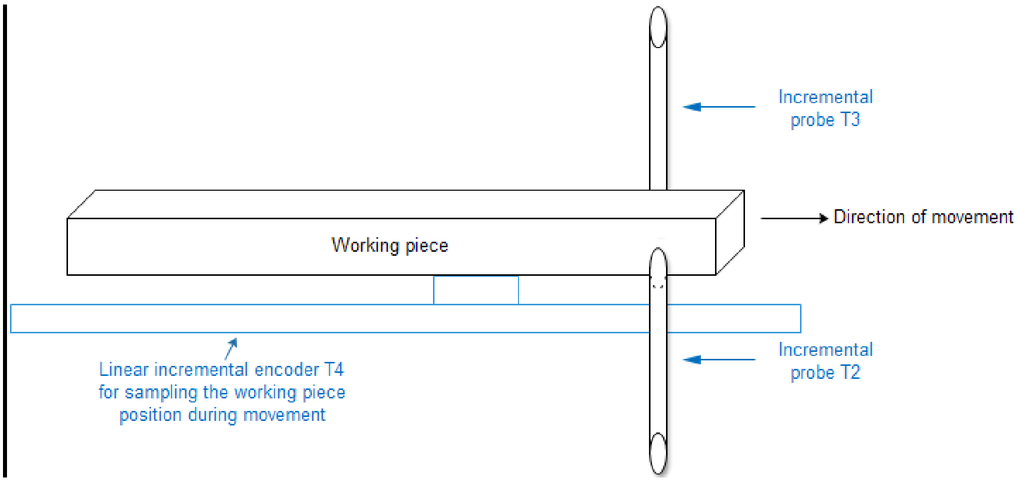
8.9.1 Example 1: Timer-triggered dynamic measurement

Task

For a working piece with a length of 12cm, the surface planarity of two sides shall be measured. Therefore an incremental probe is used for each side (T2 and T3). The working piece is moved constantly via a servo-drive.

It takes about 2 seconds to move the working piece along the probes. In order to measure very small non-planarity, a sample rate of 5.000 samples/s is required.

The position of the working piece is measured via a linear incremental encoder with 32000 increments/cm (T4). It will be used for determining the surface planarity.



Example "time-triggered dynamic measurement"

Preliminary considerations

It is known that the measurement takes about 2 seconds; an exact duration is unknown. Therefore the dynamic measurement must be started with a longer duration. 4 seconds are chosen. While the dynamic measurement is active, the position of the working piece (T4) is used to check, whether all samples have been taken. If yes, the dynamic measurement will be stopped.

The length of the working piece in increments of the incremental encoder T4 is $32000 \frac{increments}{cm} * 12cm = 384000\ increments$. As soon as this value has been reached, the measurement is stopped.

Example-Code

```
// Write channel list 1 using the measurement channels T2, T3
and T4
ansiString = "#1;T2;T3;T4#";
WriteCommandStr(opcWCL, ansiString);
if (ansiString != "#0#") return -1; // An error occured: cancel
starting measurement

// Define trigger:
// TriggerNo 1; time-triggered; * = no input channel required;
divisor = 1;
// interval = 0.2ms (-> 5000 values/s); start = 0ms; end =
4000ms
ansiString = "#1;T;*;1;0.2;0;4000#";
WriteCommandStr(opcDT, ansiString);
if (ansiString != "#0#") return -2; // An error occured: cancel
starting measurement

// Define dynamic measurement 1:
// TriggerNo 1; channel list 1; dyn. measurement active;
// 5s * 4000 samples/s = 20000 samples
ansiString = "#1;1;1;20000#";
WriteCommandStr(opcDDM1, ansiString);
if (ansiString != "#0#") return -3; // An error occured: cancel
starting measurement

// Setup data transfer channel for dynamic measurement values
result = MSC_SetupExtendedDynamicChannel(pDevice, opcRDM1, 3,
1, NULL);
if (result != MSC_STATUS_SUCCESS) return -4;

// Allocate 3 buffers, each having a size of 20000 * 4 Bytes =
80000 Bytes
// for the measurement values. Assign these buffers to the DLL.
for (i = 0; i < 3; i++) {
  buffer[i] = malloc(20000*4);
  result = MSC_AttachSubChannelBuffer(pDevice, opcRDM1, i,
20000*4, &buffer[i]);
  if (result != MSC_STATUS_SUCCESS) return -5;
}

// Activate trigger
ansiString = "#1#";
WriteCommandStr(opcAT, ansiString);
if (ansiString != "#0#") return -6; // An error occured: cancel
```

```
starting measurement

// Wait until the dynamic measurement is finished
do {
   result = MSC_GetPosition(pDevice, opcRDM1, &nSamples);
   if (result != MSC_STATUS_SUCCESS) return -7;
   Sleep(50); // Take a break for 50ms
} while ((nSamples < 20000) && (staticValueT4 < 384000));

// Inactivate trigger
ansiString = "#1#";
WriteCommandStr(opcIT, ansiString);

// Wait until all measurement values are transferred to the PC.
// Note that if the loop above has been stoped by the second
condition
// (staticValueT4 < 384000), not all values may have been
transferred to
// the PC yet.
```

It is assumed that the static measurement is running in parallel. The variable `staticValueT4` contains the measurement value T4, which is cyclically updated via static measurement.

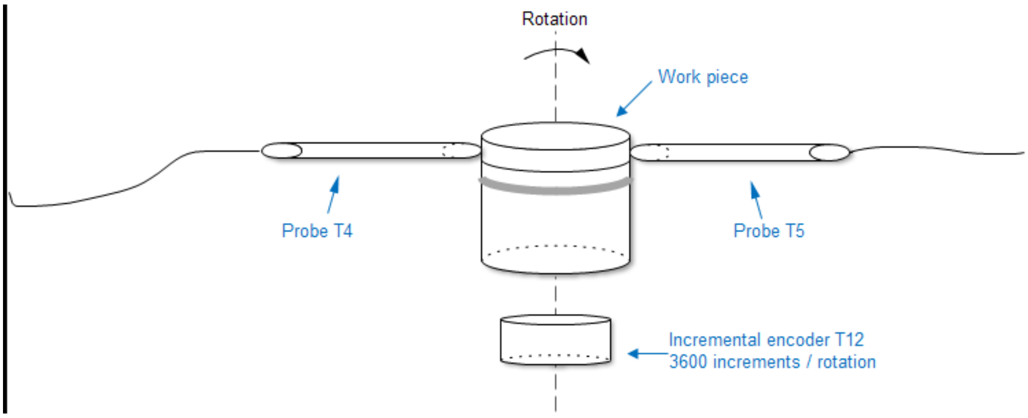
8.9.2 Example 2: Position-triggered dynamic measurement

Task

A roundness measurement shall be performed for a working piece. The working piece is placed on a rotary disk, which is driven by an uncontrolled motor with a speed of 5 s/rotation. During this rotation 720 measurement values in 0,5°-steps shall be sampled.

An incremental encoder providing 3600 increments is placed in between the motor and the disk. The working piece is measured using two inductive probes.

The incremental encoder is connected to measurement input T12, the inductive probes are connected to T4 and T5.



Example "position-triggered dynamic measurement"

Preliminary considerations

Based on the resolution of the incremental encoder of 3600 increments / rotation and the step-size of 0,5°, the step-size between two samples is 5 increments.

Example-Code

```
// Write channel list 1 using the measurement channels T4 and
T5
ansiString = "#1;T4;T5#";
WriteCommandStr(opcWCL, ansiString);
if (ansiString != "#0#") return -1; // An error occured: cancel
starting measurement

// Define trigger:
// TriggerNo 1; position-triggered; trigger-channel =
measurement channel T12; divisor = 1;
// step-size = 5; start = 0; end: * = without end
ansiString = "#1;P;T12;1;5;0;*#";
WriteCommandStr(opcDT, ansiString);
if (ansiString != "#0#") return -2; // An error occured: cancel
starting measurement

// Define dynamic measurement 1:
// TriggerNo 1; channel list 1; dyn. measurement active; 720
measurement values
ansiString = "#1;1;1;720#";
WriteCommandStr(opcDDM1, ansiString);
if (ansiString != "#0#") return -3; // An error occured: cancel
starting measurement

// Setup data transfer channel for dynamic measurement values
result = MSC_SetupExtendedDynamicChannel(pDevice, opcRDM1, 2,
1, NULL);
if (result != MSC_STATUS_SUCCESS) return -4;

// Allocate 2 buffers, each having a size of 720 * 4 Bytes =
```



```
2880 Bytes for the
// measurement values; assign these buffers to the MscDll
for (i = 0; i < 2; i++) {
  buffer[i] = malloc(720*4);
  result = MSC_AttachSubChannelBuffer(pDevice, opcRDM1, i,
720*4, &buffer[i]);
  if (result != MSC_STATUS_SUCCESS) return -5;
}

// Activate trigger
ansiString = "#1#";
WriteCommandStr(opcAT, ansiString);
if (ansiString != "#0#") return -6; // An error occured: cancel
starting measurement

// An error occured: cancel starting measurement
do {
  result = MSC_GetPosition(pDevice, opcRDM1, &nSamples);
  if (result != MSC_STATUS_SUCCESS) return -7;
  Sleep(50); // Take a break for 50ms
} while (nSamples < 720);

// Inactivate trigger
ansiString = "#1#";
WriteCommandStr(opcIT, ansiString);

// The measurement values are available now and can be used for
further processing
```

Starting the dynamic measurement

A common problem with such measurements is the start of the dynamic measurement. In the example above, the start position 0 is expected. This works only, if the rotational disk is turned backwards, until the incremental encoder T12 has a negative value. This neither always possible, nor does it make sense.

In order to solve this problem, the position of the incremental encoder T12 can be set to a negative value by software. The function `MSC_WriteCommand` together with the opcode `opcSP` (0x35) must be used for this purpose. If the incremental encoder has an index signal, the following solution is a good choice:

a) Before starting the measurement, the position of the incremental encoder is set to a negative value, e.g. -100.000. Simultaneously the index evaluation is activated.

b) Measurement is started. The drive starts turning.

c) If the incremental encoder crosses the index signal, the position of the incremental encoder is set to 0. The dynamic measurement now starts sampling the measurement values.

Using the code example from above, the following code must be executed before activating the trigger:

```
// Set position of incremental encoder ("-100000") and activate
index pulse ("REFON")
ansiString = "#T12;-100000;REFON#";
WriteCommandStr(opcSP, ansiString);
if (ansiString != "#0#") return -8; // An error occured: cancel
starting measurement
```

For some applications this solution has the disadvantage, that the position of the incremental encoder is set to 0 again after 1 rotation (in the example 3600 incremenets). In case this is a problem, the index pulse can be deactivated during the measurement without affecting the position of the incremental encoder. In order to do this, the do-while-loop of the example must be changed as follows:

```
// An error occured: cancel starting measurement
flagOnce = true;
do {
   result = MSC_GetPosition(pDevice, opcRDM1, &nSamples);
   if (result != MSC_STATUS_SUCCESS) return -7;

   if ((flagOnce == true) && (staticValueT12 > 0)) {
       flagOnce = false;
       ansiString = "#T12;*;REFOFF#";
       WriteCommandStr(opcSP, ansiString);
       if (ansiString != "#0#") return -9; // An error occured:
cancel measurement
   }

   Sleep(50); // Take a break for 50ms
} while (nSamples < 720);
```

It is assumed that the static measurement is running in parallel. The variable `staticValueT12` contains the measurement value T12, which is cyclically updated via static measurement.

8.9.3 Example 3: 2 dynamic measurements simultaneously

Task

Two working pieces shall be measured in parallel using time-triggered dynamic measurement. Because of the different characteristics of the workiing pieces, the first one must be sampled at 4.000 samples/s, for the second one a sample rate of 1.000 samples/s is sufficient.

The measurement channels T1, T2, T3 and T9 are required for the first working piece.

The measurement channles T2, T6, T7, T8, T10, T11 and T12 are required for the second working piece.

Both dynamic measurements shall be started simultaneously. The first measurement takes 2,5 seconds. The second measurement shall be stopped after 5000 measurement values / channel have been sampled.

Example-Code

```
// Write channel list 1 using the channels T1, T2, T3 and T9.
// It will be used for the first dynamic measurement.
ansiString = "#1;T1;T2;T3;T9#";
WriteCommandStr(opcWCL, ansiString);
if (ansiString != "#0#") return -101; // An error occured:
cancel starting measurement

// Write channel list 2 using the channels T5, T6, T7, T8, T10,
T11 and T12.
// It will be used for the second dynamic measurement.
ansiString = "#2;T2;T6;T7;T8;T10;T11;T12#";
WriteCommandStr(opcWCL, ansiString);
if (ansiString != "#0#") return -201; // An error occured:
cancel starting measurement

// Define trigger 1, which will be used for the first dynamic
measurement.
// TriggerNo 1; time-triggered; * = no input channel required;
divisor = 1;
// interval = 0.25ms (-> 4000 values/s); start = 0ms; end =
2500ms
ansiString = "#1;T;*;1;0.25;0;2500#";
WriteCommandStr(opcDT, ansiString);
if (ansiString != "#0#") return -102; // An error occured:
cancel starting measurement

// Define trigger 2, which will be used for the second dynamic
measurement.
// TriggerNo 2; time-triggered; * = no input channel used;
divisor = 1;
// interval = 1ms (-> 1000 values/s); start = 0ms; end: * = no
end defined
ansiString = "#2;T;*;1;1;0;*#";
WriteCommandStr(opcDT, ansiString);
if (ansiString != "#0#") return -202; // An error occured:
cancel starting measurement

// Define dynamic measurement 1
// TriggerNo 1; channel list 1; dyn. measurement active;
// number of samples: * = unlimited
ansiString = "#1;1;1;*#";
WriteCommandStr(opcDDM1, ansiString);
if (ansiString != "#0#") return -103; // An error occured:
cancel starting measurement
```

```
// Define dynamic measurement 2:
// TriggerNo 2; channel list 2; dyn. measurement active;
// number of samples: 5000
ansiString = "#2;2;1;5000#";
WriteCommandStr(opcDDM2, ansiString);
if (ansiString != "#0#") return -203; // An error occured:
cancel starting measurement

// Setup data transfer channel for dynamic measurement values
result = MSC_SetupExtendedDynamicChannel(pDevice, opcRDM1, 4,
1, NULL);
if (result != MSC_STATUS_SUCCESS) return -104;

// Allocate 4 buffers, each with 2,5s * 4000 samples/s * 4
Bytes/sample = 40000 Bytes
// for the measurement values of the first dynamic measurement.
// Assign these buffers to the MscDll.
for (i = 0; i < 4; i++) {
  bufDyn1[i] = malloc(40000);
  result = MSC_AttachSubChannelBuffer(pDevice, opcRDM1, i,
40000, &bufDyn1[i]);
  if (result != MSC_STATUS_SUCCESS) return -105;
}

// Setup data transfer channel for dynamic measurement values
result = MSC_SetupExtendedDynamicChannel(pDevice, opcRDM2, 7,
1, NULL);
if (result != MSC_STATUS_SUCCESS) return -204;

// Allocate 7 buffers, each with 5000 * 4 Bytes = 20000 Bytes
// for the measurement values of the second dynamic
measurement.
// Assign these buffers to the MscDll.
for (i = 0; i < 7; i++) {
  bufDyn2[i] = malloc(20000);
  result = MSC_AttachSubChannelBuffer(pDevice, opcRDM2, i,
20000, &bufDyn2[i]);
  if (result != MSC_STATUS_SUCCESS) return -205;
}

// Activate trigger 1. This will start the first dynamic
measurement.
ansiString = "#1#";
WriteCommandStr(opcAT, ansiString);
if (ansiString != "#0#") return -106; // An error occured:
cancel starting measurement

// Activate trigger 2. This will start the second dynamic
measurement.
ansiString = "#2#";
WriteCommandStr(opcAT, ansiString);
if (ansiString != "#0#") return -206; // An error occured:
cancel starting measurement

// Wait until the dynamic measurements are finished.
do {
  result = MSC_GetPosition(pDevice, opcRDM1, &nSamplesDyn1);
```

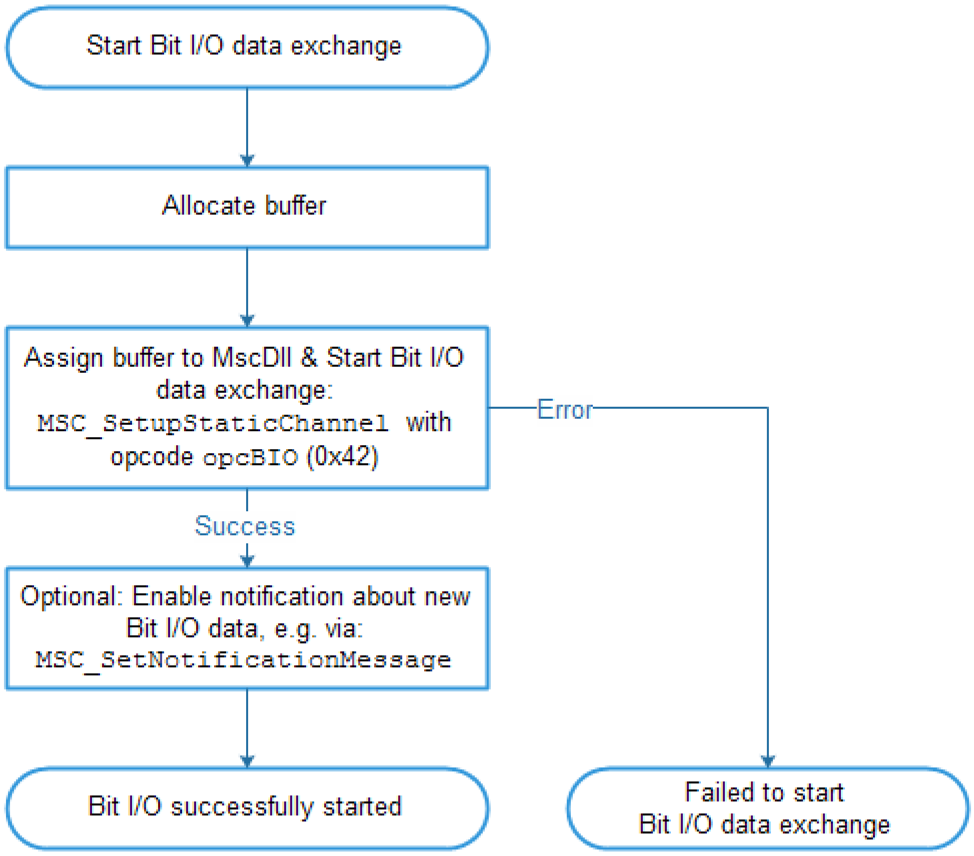
```
    if (result != MSC_STATUS_SUCCESS) return -107;
    result = MSC_GetPosition(pDevice, opcRDM2, &nSymplesDyn2);
    if (result != MSC_STATUS_SUCCESS) return -207;
    Sleep(50); // Take a break for 50 ms
} while ( (nSamplesDyn1 < 10000) && (nSamplesDyn2 < 5000) );

// Inactivate trigger 1
ansiString = "#1#";
WriteCommandStr(opcIT, ansiString);

// Inactivate trigger 2
ansiString = "#2#";
WriteCommandStr(opcIT, ansiString);

// All measurement values are available now and can be
evaluated.
```

Notes

- In this example, both dynamic measurements are started simultaneously. They can also be started independently.
- The time-triggered measurements are stopped in different ways: measurement 1 is stopped after a certain time, measurement 2 is stopped after a certain number of samples have been taken. Using the time-triggered dynamic measurement, both methods are equal.
  In reality, one would use either of these methods for both measurements.
- The do-while-loop is finished, if all dynamic measurements are finished. Since the first dynamic measurement stops earlier, it would be possible to evaluate its measurement values earlier.
- Measurement channel T2 is used in both dynamic measurements. This is not a problem, since both dynamic measurements use an independent recording of the measurement values.

8.10 Bit I/O

The exchange of Bit I/O data is similar to the static measurement. The following figure shows the steps for the start of the Bit I/O data exchange:

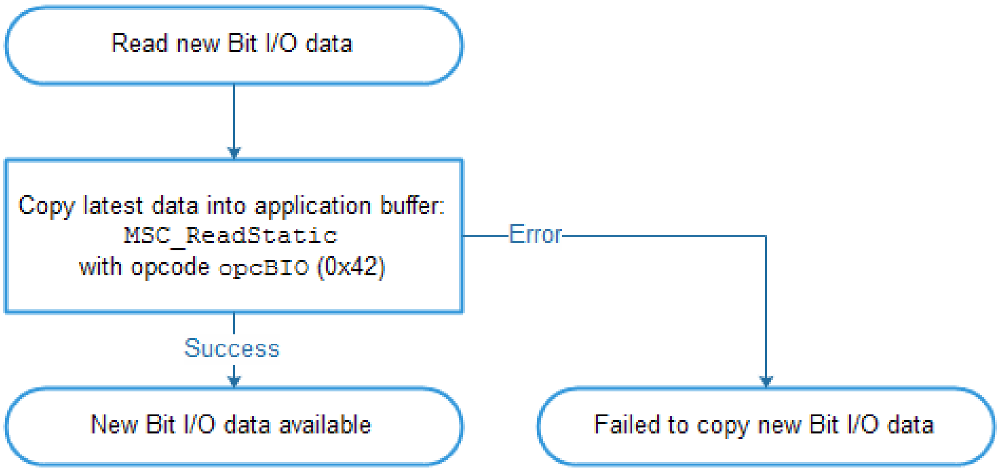


Steps for "starting Bit I/O data exchange"

- The function `MSC_SetupStaticChannel` together with the opcode `opcBIO` (0x42) starts the continuous exchange of Bit I/O data through the MscDll.
  Two buffers must be provided:
  - A receive buffer, which contains the current output data, followed by the input data received.
  - A send buffer, which contains the output data, that shall be send to the Irinos-System.

- After calling the function `MSC_SetNotificationMessage` (alternativ `MSC_SetNotificationEvent` or `MSC_SetNotificationCallback`) together with the opcode `opcBIO` (0x42), the application is informed every time new Bit-I/O data is available. It is up to the application whether it uses or ignores this data.
  Using notifications is recommended, but not required.

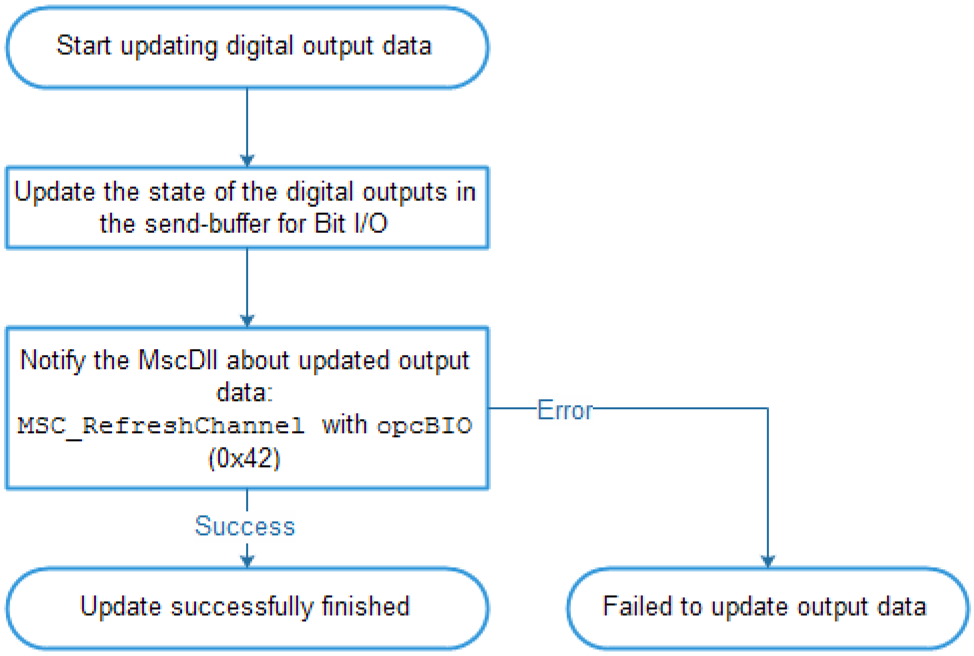
- The received Bit-I/O data is first stored in an internal buffer of the MscDll. In order to copy them into the buffer of the application, the function `MSC_ReadStatic` together with the opcode opcBIO (0x42) must be called (see following figure). This function must be called every time new Bit-I/O data has been received.



Steps for "copying bit I/O data into the buffer of the application"

Writing output data to the MscDll

Compared to reading static measurement values, the Bit-I/O data exchange is enhanced by writing data to the Irinos-System. New output data can be written into the send buffer anytime. After each update, the function `MSC_RefreshChannel` together with the opcode `opcBIO` (0x42) must be called in order to inform the MscDll about the update:



Steps for "updating Bit-I/O data"

Bit assignment

The number of digital in- and outputs is rounded up to a multiple of 8 per Irinos-Box. Examples:

- An Irinos-Box has 2 digital inputs. 6 "virtual" inputs are added, the sum is 8 bits.
  All virtual inputs are always low.
- An Irinos-Box has 68 digital in- and outputs. 4 „virtual" digital in and outputs are added. The sum is 72 in- and outputs.

The bit order depends on the order of the Irinos-Boxes (according to the Box addresses). Two examples are provided in the following tables:

Box type	Number of digital inputs	Input bits	Number of digital outputs	Output bits
IR-MASTER 47	68	1 .. 68 [69 .. 72 -> "virtual"]	68	1 .. 68 [69 .. 72 -> "virtual"]
IR-TFV 50	0	-	0	-
IR-INC 56	0	-	0	-
IR-DIO 58-16-16	16	73 .. 80	16	73 .. 80
IR-DIO 58-16-16	16	81 .. 96	16	81 .. 96

Box type	Number of digital inputs	Input bits	Number of digital outputs	Output bits
IR-TFV[50] (integrated Master)	2	1 .. 2 [3 .. 8 -> "virtual"]	0	-
IR-TFV[50]	0	-	0	-
IR-DIO[58]-16-16	16	9 .. 24	16	1 .. 16
IR-INC[56]	0	-	0	-
IR-HMI1[61]	40	25 .. 64	40	17 .. 56

Update-Rate

The update-rate is identical to the update-rate for static measurement values.

An exception are the digital in- and outputs, which are connected via the IO-Bus to the Irinos-Box IR-MASTER[47]. These are updated about 30 times/s.

Please note:

Bit-I/O data exchange is not done in realtime. Always use a handshake-method for the exchange of status information via Bit-I/O.

Send- / receive- buffer

The size of the send- and receive buffers depends on the maximum number of in- and output bits, which shall be used. A basic rule is, that always the same amount of input bits is read as output bits are written.
If for example 64 output bits are written, then 64 input bits will be read.
If for example 128 output bits are written, then 128 input bits will be read.

For Irinos-Systems without fieldbus-interfaces like ProfiNet, 128 bits = 16 bytes are sufficient. In this case the size of the send buffer must be 16 bytes.
The receive buffer first contains a copy of the digital output bits. These are followed by the digital input bits. Thus the receive buffer must always have twice the size of the send buffer. If for example 128 bits are used, the size of the receive buffer must be 32. The following figure provides an example:

Send buffer
Byte 0: Output bits 1-8
Byte 1: Output bits 9-16
Byte 2: Output bits 17-24
Byte 3: Output bits 25-32
Byte 4: Output bits 33-40
Byte 5: Output bits 41-48
Byte 6: Output bits 49-56
Byte 7: Output bits 57-64
Byte 8: Output bits 65-72
Byte 9: Output bits 73-80
Byte 10: Output bits 81-88
Byte 11: Output bits 89-96
Byte 12: Output bits 97-104
Byte 13: Output bits 105-112
Byte 14: Output bits 113-120
Byte 15: Output bits 121-128

Output data is mirrored by the Irinos-System to the input data.

Receive buffer
Byte 0: Output bits 1-8
Byte 1: Output bits 9-16
Byte 2: Output bits 17-24
Byte 3: Output bits 25-32
Byte 4: Output bits 33-40
Byte 5: Output bits 41-48
Byte 6: Output bits 49-56
Byte 7: Output bits 57-64
Byte 8: Output bits 65-72
Byte 9: Output bits 73-80
Byte 10: Output bits 81-88
Byte 11: Output bits 89-96
Byte 12: Output bits 97-104
Byte 13: Output bits 105-112
Byte 14: Output bits 113-120
Byte 15: Output bits 121-128
Byte 16: Input bits 1-8
Byte 17: Input bits 9-16
Byte 18: Input bits 17-24
Byte 19: Input bits 25-32
Byte 20: Input bits 33-40
Byte 21: Input bits 41-48
Byte 22: Input bits 49-56
Byte 23: Input bits 57-64
Byte 24: Input bits 65-72
Byte 25: Input bits 73-80
Byte 26: Input bits 81-88
Byte 27: Input bits 89-96
Byte 28: Input bits 97-104
Byte 29: Input bits 105-112
Byte 30: Input bits 113-120
Byte 31: Input bits 121-128

Bit I/O send- and receive- buffer (example for 128 Bits)

8.11 Error- and diagnostics-management

The Irinos-System has various intgerated diagnostic mechanisms. The MscDll allows reading the operation status. Two categories are available:

○ The status of the measurement channels 166.

○ The status of the Irinos-Boxes 169.

Both categories can be read via the MscDll.

8.11.1 Status of measurement channels ("hardware status")

Manipulation of measurement values

If an error at a measurement channel is detected, which leads to an unusable measurement value, the Irinos-System automatically manipulates the measurement value: a default value is delivered instead of a measurement value, as long as the error is active. This functionality depends on the input type.

It is especially useful, if the measurement software has an integrated probe supervision. A typical implementation of a probe supervision is a check, whether the probe is moved at least the distance x between two measurements.

The values are also manipulated, if the Master-Box is not receiving measurement values from the Slave-Boxes because of a communication problem. Typically the reason for this is a broken ILink cabling 113 (broken cable, connector not fixed properly, ...).

The manipulated value and the behaviour depend on the measurement channel type:

Channel type	Default value in case of an error	Condition for manipulation of the measurement value	No manipulation in case ...
IR-TFV[50] Inductive probe	0x00007FFF = 32767	○ Sine oscillator short circuit ○ ILink communication error	
IR-AIN[54] / IR-MASTER[47] Analogue input ± 10V	0x00007FFF = 32767	○ Overload of the reference voltage ○ ILink communication error	○ Overload of the 24V supply output
IR-INC[56] Incremental encoder	0x7FFFFFFF = 2147483647	○ ILink communication error	○ Incremental encoder error

Hardware status

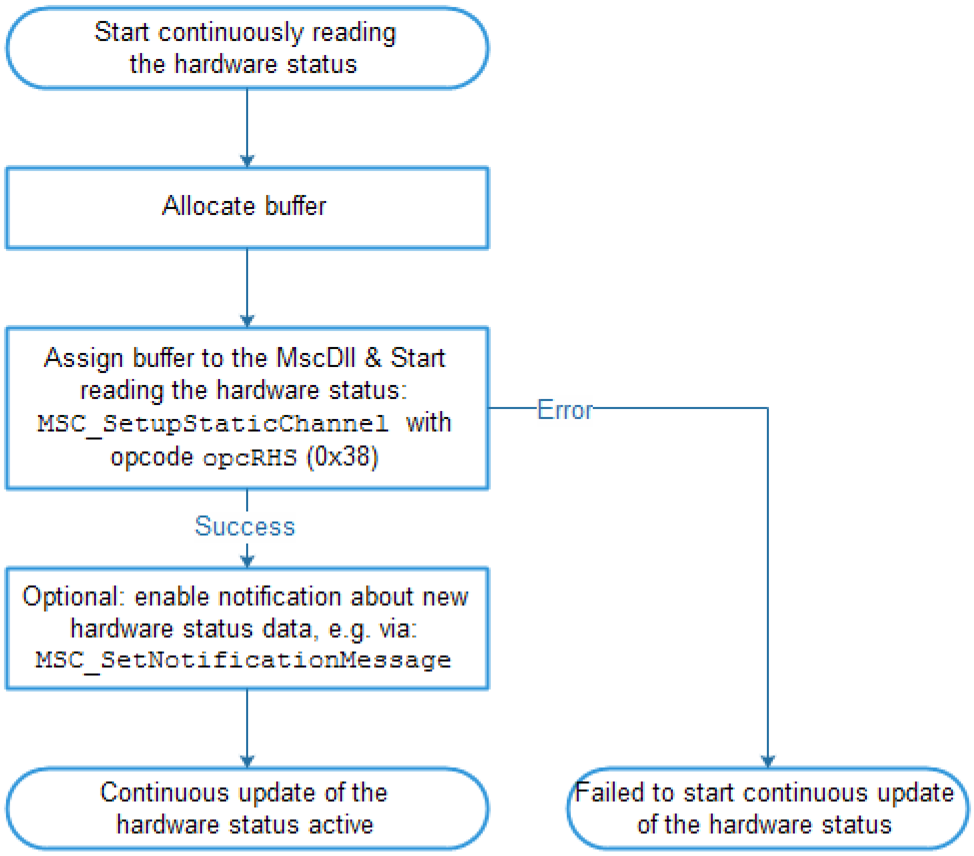
In addition to the value manipulation, the "hardware status" of the measurement channels can be readout separately. The opcode `opcRHS` (0x38) delivers one byte per measurement channel, containing various error bits. The bit definition depends on the type of measurement channel.

Evaluating the hardware status is strongly recommended for incremental inputs[56], since they are not subject to the measurement value manipulation in case of an error. Additionally the hardware status byte of an incremental channel also contains the information, whether an index pulse has been detected. This information is required for various measurements.

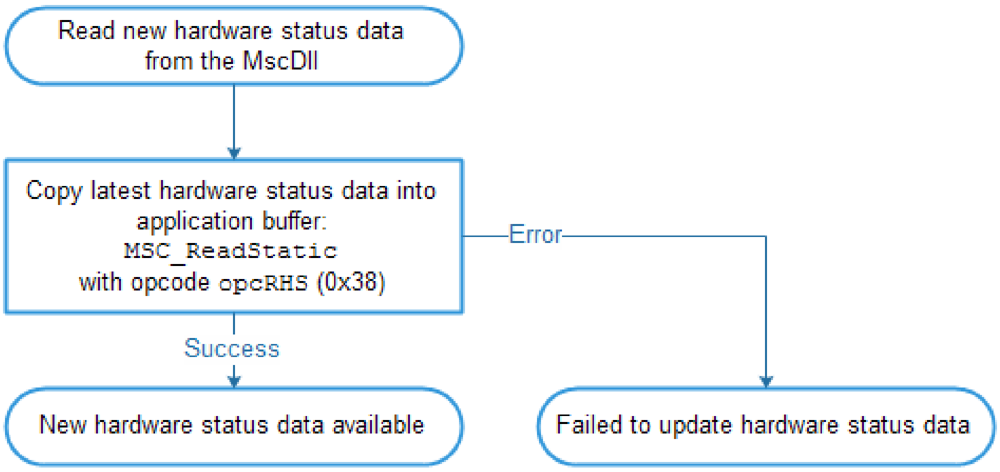
Two possibilities are available for accessing the hardware status:

1. It can be updated continuously via a static data channel by the MscDll, like it is used for the static measurement values or the Bit

I/Os.

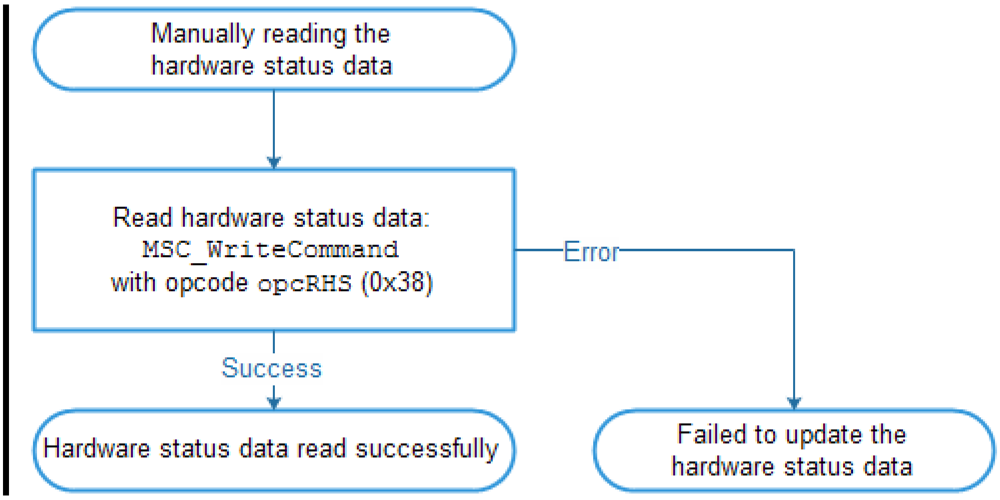


Steps for "continuous update of the hardware status"



Steps for "reading the hardware status from a static data channel"

2. It can be readout manually by the application via the function `MSC_WriteCommand` e.g. at the end of a measurement sequence.



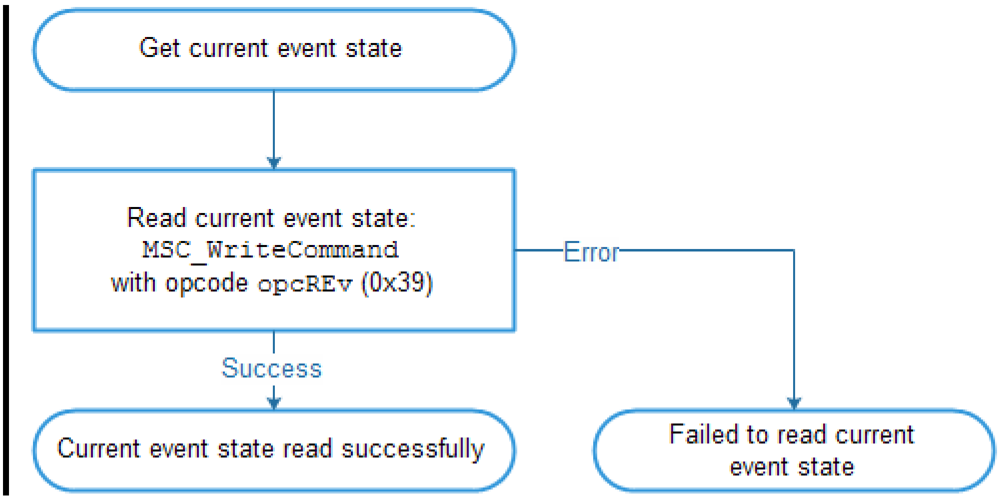
Steps for "manually updating the hardware status"

8.11.2 Reading / resetting the event status of Irinos-Boxes

Reading the event status

Each Irinos-Box has its own event handler. If a special <u>event</u> 180 occurs in the Irinos-Box, it is reported to the event handler. Under normal operation, no event should occur.

The last event reported is set as "current Event". It can be readout via the opcode `opcREv` (0x39) together with the function `MSC_WriteCommand`:



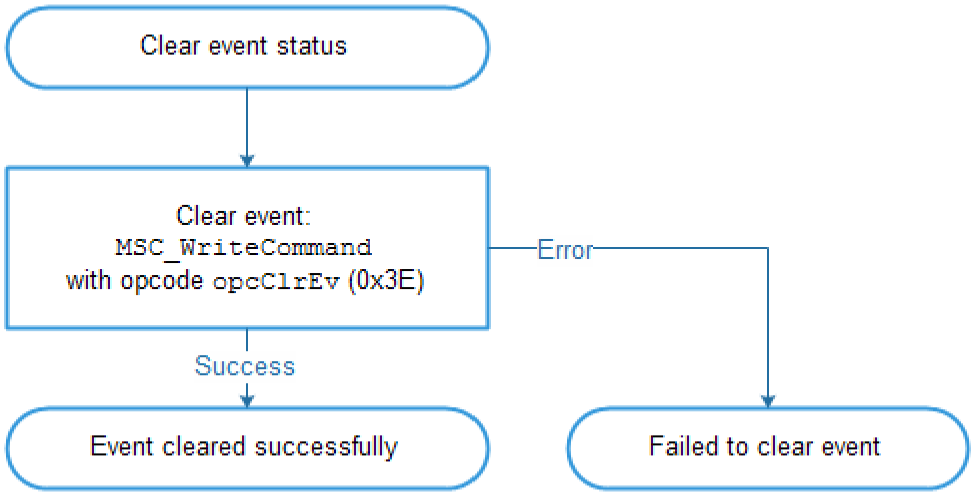
Steps for "reading the event status"

Clear the event status

By clearing (deleting) an event, it is no longer set as "current Event" and it is no longer displayed via the <u>status LED</u> 45 respectively the <u>7-digit display</u> 46. Please note:

Clearing the event does not solve the problem! Depending on the event type, it may reoccur immediately.

Clearing the event[180] is done via the opcode `opcClrEv` (0x3E) together with the function `MSC_WriteCommand`:



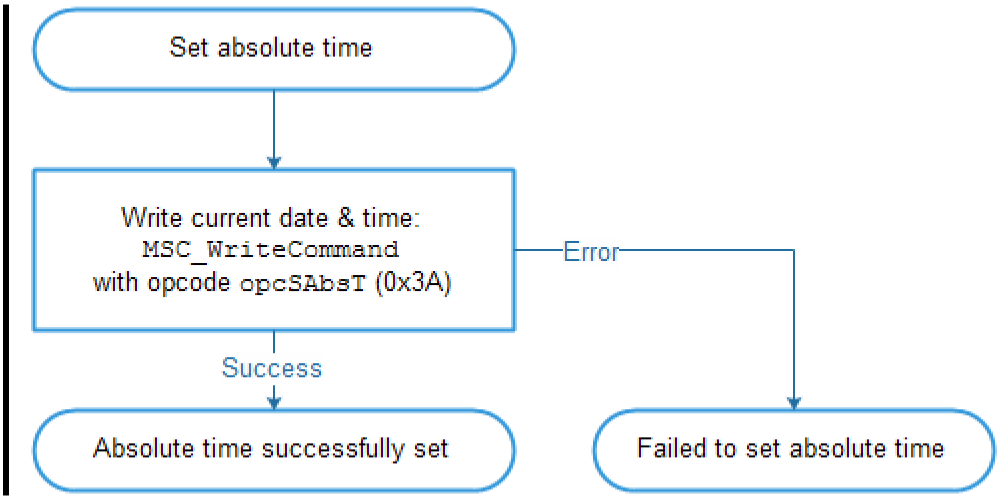
Steps for "clearing the event status"

8.11.3 Setting the absolute time for the diagnostics memory

Each Irinos-Box has an integrated diagnostics memory[190]. It is a tool which can support problem analysis. Especially if problems occur sporadically, it is very useful to get detailed information about the time and the frequency an event has occured. Therefore the ILink-Time[28] is added to each entry in the diagnostics memory.

Additionally the current date and time can be added to each entry. Since the Irinos-System does not have an internal realtime clock (RTC), the application needs to send the current date and time after the Irinos-System has been started. This is done via the function `MSC_WriteCommand` together with the opcode `opcSAbsT` (0x3A).

For longterm operation it is recommended to update the date and time once per day in order to adjust any inaccuracies of the internal clock.



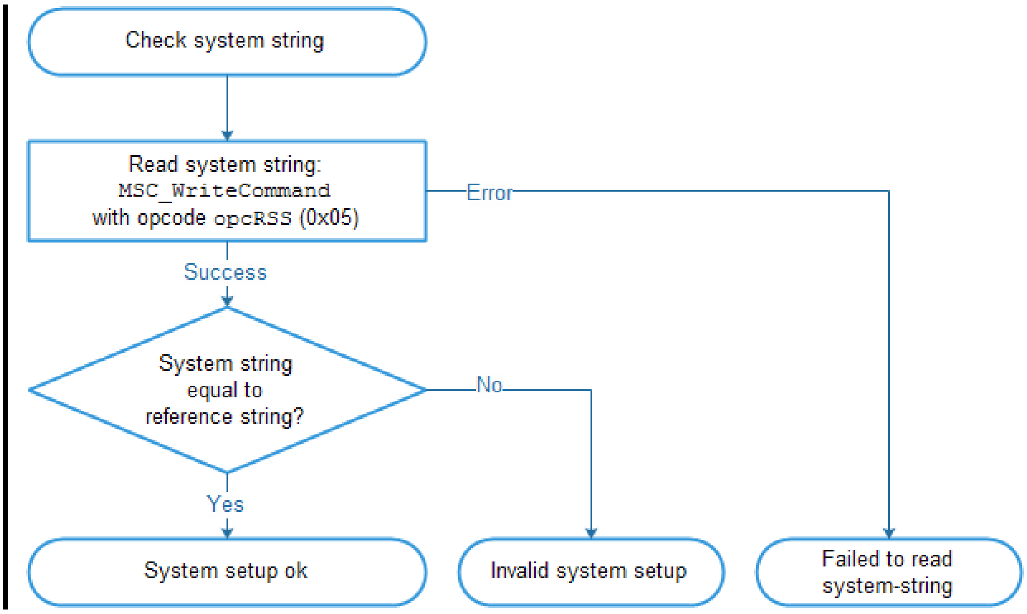
Steps for "set absolute time"

8.12 Verifying the system structure

The Irinos-System provides a string, which reflects its structure. It contains the number of boxes and their order numbers in the order of their addresses.

It is recommended to request this string after the start of the measurement application in order to compare it with a reference string. Therefore the reference string must be stored in the application once, e.g. in its ini-file.
This ensures that the Irinos-System has the structure, which is expected by the application. If for example an Irinos-Box has been removed, the application recognizes this through the string comparison and can take appropriate action (e.g. showing a warning message).

The following figure shows the required steps for reading and comparing the system-string:



Steps for "comparing the system-string"

The system-string is requested using the function `MSC_WriteCommand` together with the opcode `opcRSS` (0x05).

It can also be requested manually using the inventory section of the Irinos-Tool.

Alternatively or additionally it is possible to request a detailed overview 172 about the Irinos-System. However, this approach is far more complex.

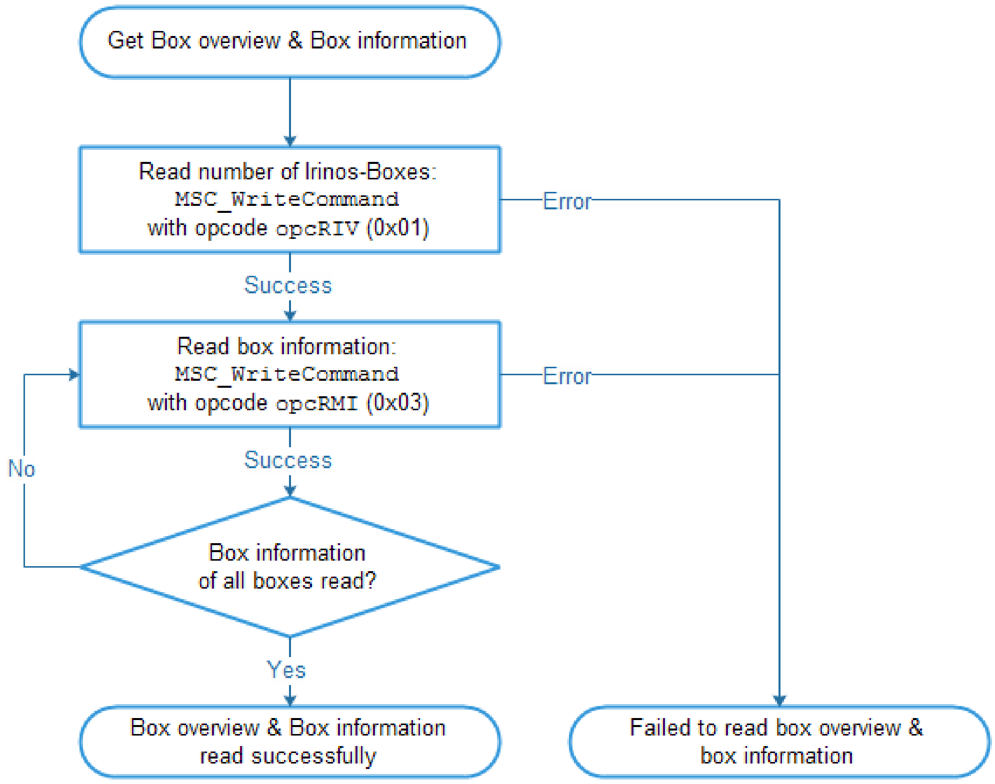
8.13 Tips und Tricks

8.13.1 Reading system information

➔*Typically it is sufficient to read the system-string* 171*. Reading the system information is only necessary, if further details are required.*

In case the application requires detailed information about the Irinos-System, e.g. for providing a graphical system overview, it is possible to read this information directly.

First it is required to read the number of Boxes using the opcode `opcRIV` (0x01). Afterwards further information can be requested for each of the available Irinos-Boxes using the opcode `opcRMI` (0x03). Both opcodes are used with the function `MSC_WriteCommand`:



Steps for "reading detailed system information"

8.13.2 Limiting the number of static channels

➔ *Limiting the number of static channels makes only sense if many measurement channels are available (about 64 or more) AND if a lot of dynamic measurement data needs to be transferred. If dynamic measurement is not used, there is no need to limit the number of static channels.*

If the static measurement is active, the measurement values of all channels are transferred within each data frame from the Irinos-System to the MscDll. For systems with many measurement channels, this can lead to a low bandwidth for the transfer of dynamic measurement values. This can result in a long transfer time. However, typically not all static measurement channels are used simultaneously. Therefore it may be advantageous to limit the number of static measurement channels in systems with about 64 or more measurement channels, in case a lot of dynamic measurement data needs to be transferred.

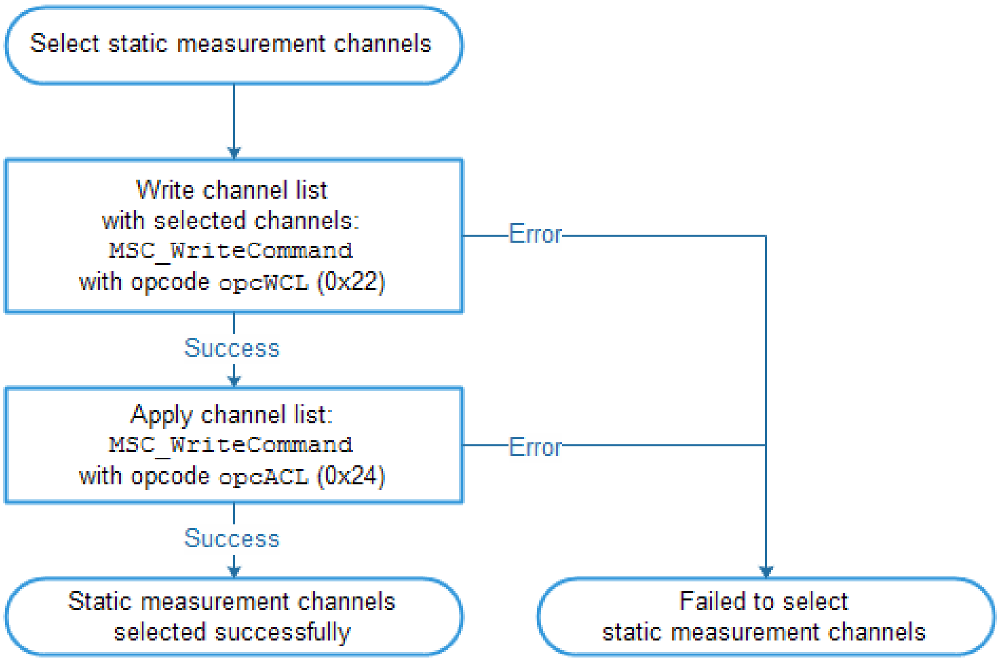
Two steps are required:

a) All channels used for the static measurement must be defined via a channel list. Therefore the function `MSC_WriteCommand` together with the opcode `opcWCL` (0x22) is used.

Make sure to use a channel list, which is not used for dynamic measurement, e.g. channel list 10.

b) This channel list must be activated for static measurement using the function `MSC_WriteCommand` together with the opcode `opcACL` (0x24).

Please note that regarding the hardware status (opcode `opcRHS`) only information for measurement channels, which are contained in the channel list, is transferred (--> hardware status channels are equal to the static measurement channels).



Steps for "limiting the number of static channels"

If necessary, multiple groups of measurement channels can be defined. Each group is placed into a separate channel list. Switching between the groups is done by selecting the respective channel list. By selecting the channel list 0, all measurement channels are activated for the dynamic measurement.

8.14 License

If used together with the Irinos-System, there are no license costs for the MscDll. It can be copied on any PC and it is allowed to deliver it together with the measurement application.

If the MscDll is used with another hardware than the Irinos-System, license fees must be paid. Licenses are solely sold directly by Messtechnik Sachs GmbH.

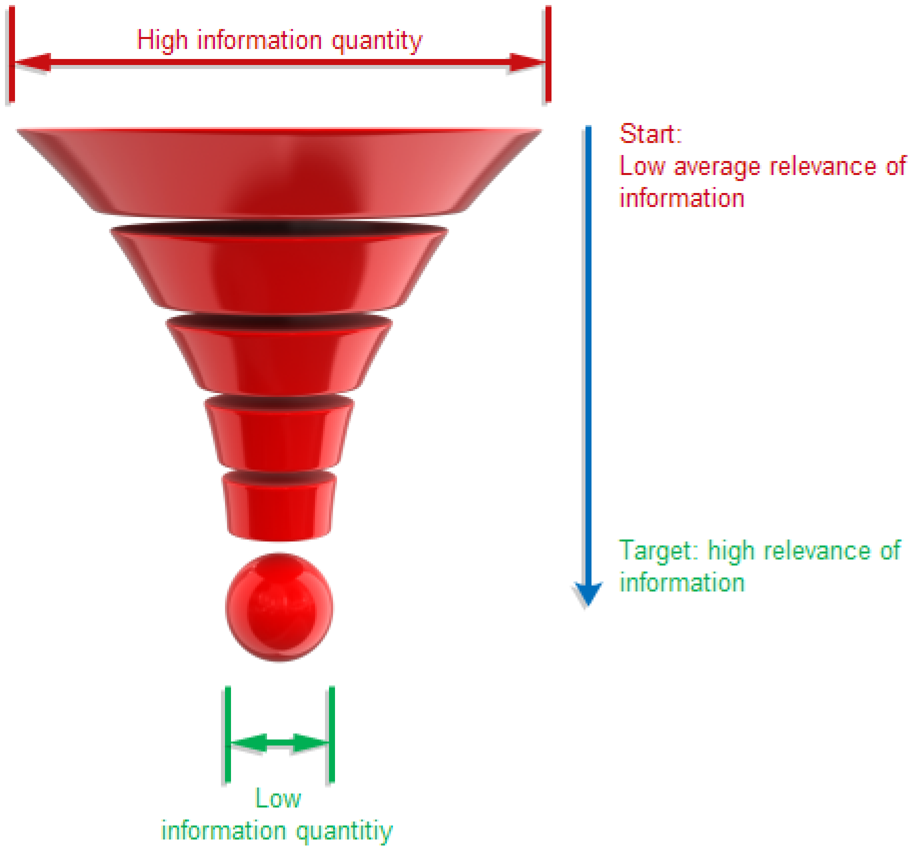
Detailed licensing information [9] is provided in the legal notes section of this manual.

Troubleshooting and "first aid"

9 Troubleshooting and "first aid"

9.1 General procedure

As with every technical system, efficient problem solving requires a systematic approach. A large amount of information with low average information relevance must be filtered in order to get high relevance information, which helps solving the problem:



Information filtering for solving problems

Follow this steps:

I. Problem description
Describe the problem as exactly as possible.
It can be very useful, to write it down on paper. This is especially important, if there are possibly multiple problems and it is unclear, whether they are related to each other. It may be necessary to formulate multiple problem descriptions.

II. Gathering information
Describe the situation before and after the problem occurred. The following questions may help:

- Was there any special situation or action before the problem occurred?
- Has the system been modified before the problem occurred?
- Are there any conspicuous symptoms after the problem occurred? Check the state of all status LEDs / displays: do they signal any errors?

III. Read the available diagnostic information. Try to find out, if the error occurred once or multiple times.

IV. **Evaluating the information**
Sort the information gathered in steps II and III by relevance using the following categories:

- Not relevant
- Possibly relevant
- Probably relevant

V. **Verifying the information**
Try to deliberately cause the problem by using the information of the category "probably relevant". Use a step-wise approach by concentrating on only one possible issue at once.
If this is not successful: try the same procedure for the other two categories.

Solving the problem is only possible, if it can be reproduced and if the reason for it is known.

Tips

- Differentiation
  If multiple problems occur, often no distinction is made between them. Always analyse only one problem. A simultaneous analysis of multiple problems is too difficult.
- Keep calm
  Problems almost always need to be solved quickly. Thus there is always a lot of time pressure. However, it does lead to faster results. Keep calm and locate the problem in systematic steps.
- Watch for little details
  Especially when it is difficult to locate the problem, it is important to watch for little details. Sometimes a LED or a suspicious noise provides more and better information than the best measurement

device.

- Simplify
  Especially if systems are complex, it is useful to reduce it to a minimized system, which runs without problems. Afterwards add complexity step by step until the problem occurs again.

9.2 Diagnostic events

Each Irinos-Box has a common event handler. If an event occurs, it will be reported to this event handler. Depending on the event type, it

- is reported to the user and the application and/or
- it will be added to the diagnostic memory.

In normal operation, no event should occur.

In order to distinguish between different events, several event types are available. Each event type has a specific event number.

Depending on the event configuration, it will be displayed via the 7-digit display 46 or the status LED 45 of the Irinos-Box. Further it can be readout via the MscDll.

The Irinos-System has very sophisticated internal error detection mechanisms to ensure a reliable operation. Most events have a hypothetical character. Therefore they are not documented. Please contact the support, if such an event occurs.

Following those events are listed, which are relevant for practice:

Event 1: System	
Description	Common system event
Type	Information
Reason	○ System started ○ Diagnostics 190 memory has been

	cleared
Reported to the user / application	No, cannot be enabled
Stored in diagnostic memory	Yes, cannot be disabled

Event 4: MscDll communication error	
Description	An error has been detected regarding the communication between the Irinos-System and the MscDll.
Type	Error
Reason / Solution	o an invalid opcode has been used („Invalid opcode in RX packet") -> Only use valid opcodes o The send- or receive- buffer size is too small („Too much TX data") -> Use the port number and buffers sizes for the Msc.cfg, which are given in the reference manual.
Reported to the user / application	Yes, cannot be disabled
Stored in diagnostic memory	Yes, can be disabled

Event 12: ILink module detection error	
Description	This error can occur only during the start-up phase of the Irinos-

	System. One or more Slave-Boxes cannot be detected or enumerated properly or the ILink termination cannot be activated properly.
Type	Error
Reason / Solution	o Invalid ILink cabling or broken ILink cable. -> Check the ILink cabling. o Multiple Master-Boxes in one Irinos-System. -> Only one Master-Box is allowed per Irinos-System. o Broken Irinos-Box -> Replace the Irinos-Box
Note	For Irinos-Systems with multiple Slave-Boxes: First test the system using only 1, then 2, then 3, ... Slave-Boxes in order to locate the problem.
Reported to the user / application	Yes, can be disabled.
Stored in diagnostic memory	Yes, can be disabled.

Event 13: ILink communication error	
Description	The ILink communication between the Irinos-Boxes has been disrupted.
Type	Error

Reason / Solution	o Invalid ILink cabling or broken ILink cable. -> Check the ILink cabling. o Insufficient power supply (e.g. short voltage drops) -> Use a sufficiently dimensioned power supply.
Note	The ILink communication has an integrated data integrity check and a data retransmission in case of an error. If the retransmission fails multiple times, this event occurs.
Reported to the user / application	Yes, can be disabled
Stored in diagnostic memory	Yes, can be disabled

Event 15: Sine-oscillator	
Description	The sine oscillator for the inductive probes has been overloaded (e.g. short circuit).
Type	Error
Reason / Solution	o Defective probe -> Replace the defective probe. o Probe connected incorrectly, e.g. if an extension cable is used. -> Check the pinning
Note	If the problem is removed, the event is automatically cleared. ➔In order to locate the defective

	probe, remove them one after the other. Wait for 10s after each probe. If the event has been cleared (can bee seen at the 7-digit display or the status LED), the defective probe has been removed.
Reported to the user / application	Yes, can be disabled
Stored in diagnostic memory	Yes, can be disabled

Event 21: Invalid dynamic measurement config	
Description	A configuration mismatch has been detected at the start of a dynamic measurement.
Type	Error
Reason / Solution	o Invalid time-period in trigger for time-triggered dynamic measurement. -> The time-period must be a multiple of the sample-times of the involved Irinos-Boxes.
Reported to the user / application	Yes, can be disabled
Stored in diagnostic memory	Yes, can be disabled

Event 24: Inc. encoder power error	
Description	The power supply of one or multiple channels for incremental

	encoders has been overloaded.
Type	Error
Reason / Solution	○ Broken incremental encoder or defective cable -> Replace incremental encoder. ○ Invalid conncetion of the incremental encoder. -> Check pinning. ○ Power consumption of the incremental encoder is too high -> See specifications.
Note	○ If an overload or short circuit occurs at a single input channel for incremental encoders, only this channel will be disabled. All other input channels remain active. As soon as the overload / short circuit condition has been removed, the event will be cleared automatically. ○ If the total load is too high, all input channels will be turned off. Restarting the Irinos-System is required.
Reported to the user / application	Yes, can be disabled.
Stored in diagnostic memory	Yes, can be disabled.

Event 25: Inc. encoder application error	
Description	Encoder signals were / are out of specification.

Type	Error
Reason / Solution	o Connector of incremental encoder has been removed. o Connector of incremental encoder is not properly fixed (loose connection). -> Fix connector properly. o Input frequency of the incremental encoder too high. -> Slow down the speed of the incremental encoder / Avoid mechanical tension or shock o Cable of incremental encoder too long. -> Use a short cable o Invalid pin assignment of the incremental encoder -> Check pin assignment o Output signals of the incremental encoder are outside of the specification -> Check signals using the Irinos-Tool.
Note	o The incremental signals can be verified using the Irinos-Tool (only 1Vpp). o Please observer the application notes regarding incremental encoders. o An input channel can be reset using the opcode opcSP (0x35). See reference manual for further details.
Reported to the user / application	Yes, can be disabled.

Stored in diagnostic memory	Yes, can be disabled.

Event 27: Firmware update error	
Description	An error occurred during the firmware update process.
Type	Error
Reason / Solution	o Invalid firmware file. -> Use a valid firmware file. o Transmission error -> Repeat the firmware update.
Note	If a firmware update fails, the old firmware version remains active.
Reported to the user / application	Yes, can be disabled.
Stored in diagnostic memory	Yes, cannot be disabled.

Event 28: Firmware update successful	
Description	Firmware successfully updated
Type	Information
Reported to the user / application	No, cannot be enabled.
Stored in diagnostic memory	Yes, cannot be disabled.

Event 33: Reference voltage error	
Description	Overload / short circuit of the reference value for the analogue inputs.
Type	Error
Reason / Solution	o Overload of the reference voltage -> Reduce load (see specifications) o Sensor broken or defective cable -> Check sensor and cable o Invalid pin assignment of the sensor connector -> Check pin assignment
Note	After the problem has been fixed, the Irinos-System must be restarted.
Reported to the user / application	Yes, can be disabled.
Stored in diagnostic memory	Yes, can be disabled.

Event 34: 24V output error	
Description	The 24V supply for the digital in-/outputs and the analogue inputs has been overloaded.
Type	Error
Reason / Solution	o Load too high

	-> Reduce maximum load (see specifications) o Broken sensor / actuator or defective cable -> Check sensor / actuator and cabling o Invalid pin assignment of the connector -> Check pin assignment
Note	After the problem has been fixed, the 24V supply output will be re-enabled and the event will be cleared automatically.
Reported to the user / application	Yes, can be disabled.
Stored in diagnostic memory	Yes, can be disabled.

Event 35: IoBus communication error	
Description	The communication between the IR-MASTER and an I/O-Box has been / is disturbed.
Type	Error
Reason / Solution	o I/O-Box has been switched off. o Cable to the I/O-Box has been removed. o Defective IO-Bus cable -> Replace cable o Improper termination of the IO-Bus. -> Check termination

Reported to the user / application	Yes, can be disabled.
Stored in diagnostic memory	Yes, can be disabled.

Event 36: Digital I/O error	
Description	The output driver for the digital outputs has been overloaded (thermal overload).
Type	Error
Reason / Solution	○ Continuous load too high. -> Reduce load according to specifications.
Note	As soon as the output driver has cooled down, the outputs will be re-enabled and the event will be cleared automatically.
Reported to the user / application	Yes, can be disabled.
Stored in diagnostic memory	Yes, can be disabled.

9.3 Diagnostic memory

Each Irinos-Box has an integrated non-volatile diagnostic memory. If an event occurs, it will be stored in the diagnostics memory (if activated in the event configuration). The diagnostic memory can be accessed via the integrated webserver or the Irinos-Tool.

The diagnostic memory is an important tool for detailed problem analysis, especially if events occur sporadically.

Each Irinos-Box is able to store a minimum of 32 entries in the diagnostic memory. If it is full, the oldest entries will be cleared in order to provide space for new entries.

Besides the event type itself, a diagnostic entry contains additional information, which can be useful for problem-solving (e.g. ILink time 28 and absolute time 170).

The ILink time is restarted together with the Irinos-System. Its unit is µs. It is identical on all Irinos-Boxes.

The absolute time contains the date and the time of the diagnostic entry. Since the Irinos-System is not equipped with a realtime clock, the absolute time is always 0 after start up of the Irinos-System. It must be written once by the PC application via the MscDll.

After start up, the diagnostic entry "System (1)" together with the text "System started" is added to the diagnostic memory. This allows determining, whether an event has occurred before or after the last system restart.

9.4 First aid "network connection"

➔ This chapter provides help for typical network connection problems. More information is available in the users manual of the Irinos-Tool.

Usually one or more of the following reasons lead to connection problems:

- Network cabling is invalid.
- The network configuration of the PC differs from the network configuration of the Irinos-System.
- The communication settings for the MscDll are wrong.

Checking network cabling

a) Check, if the network interface of the Irinos-System is connected to the network interface of the PC.
A working electrical connection is signalled by the "Link-LED" (IR-MASTER 47) respecively the "Ethernet-LED" (integrated Master 44). This LED must either be turned on or flashing.

Proceed with the next step, if this is the case.

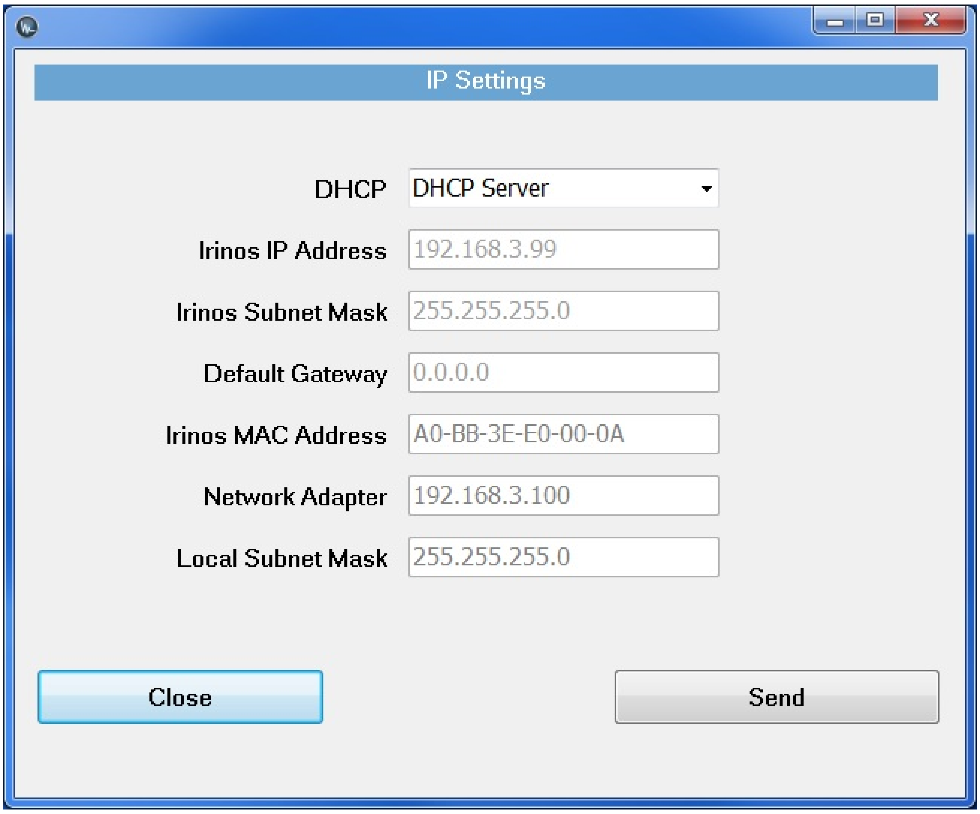
Verifying the network configuration

b) Start the Irinos-Tool. It searches for all available Irinos-Systems in the network and lists them in a table. Your Irinos-System should be listed in this table. You can use the MAC address to verify this (the MAC address is printed on the type-plate of the Master Irinos-Box).
The IP settings will also be displayed in this table.

c) Try to connect to the Irinos-System via the Irinos-Tool.

Proceed with the next step, if the connection cannot be established.
Otherwise proceed with step f).

d) Open the network configuration of the Irinos-System by douple-clicking in the table row. The following window opens:



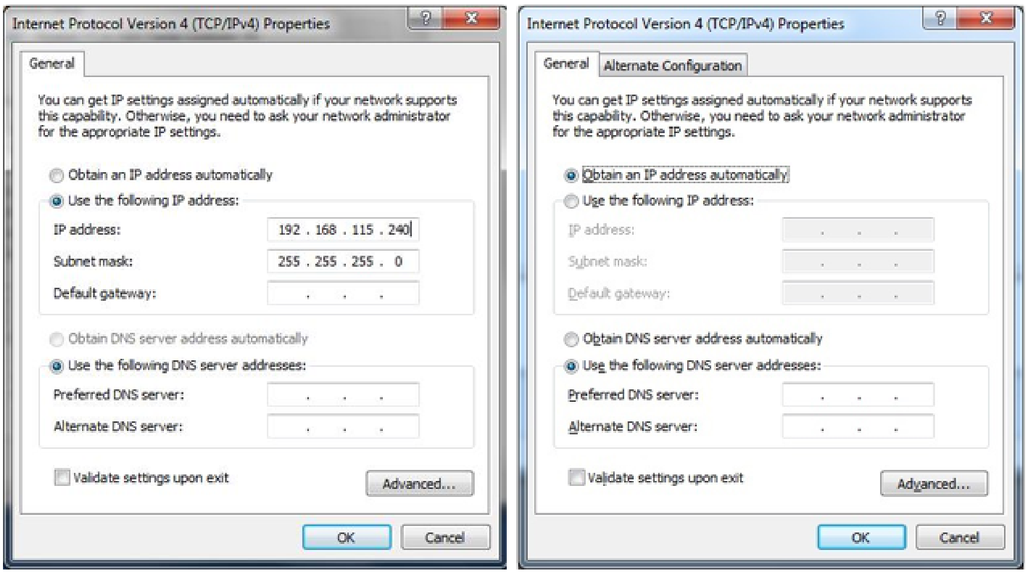
Network configuration of the Irinos-Tool

e) Open the Windows configuration settings of the network adapter, which is connected to the Irinos-System. Open the settings for "internet protocol version 4 (TCP/IPv4)".

In case the DHCP-Server of the Irinos-System is enabled, the option "Optain an IP address automatically" must be enabled (see left figure below).

In case the Irinos-System has a fixed IP address, the PC must also have a fixed IP address. Both IP addresses must be within the same subnet. In most cases the subnet mask 255.255.255.0 is used. In this case the first three elements of the IP addresses must be the same. If the Irinos-System for example has the IP address 192.168.178.1, the PC must have an IP address of the range 192.168.178.2 to 192.168.178.254 (see right figure below).

If necessary, change the Windows IP settings.



Windows IP configuration

Adopting the communication settings for the measurement application

f) If a connection is established via the Irinos-Tool, is generates the configuration file Msc.cfg. It is located in the same directory as the Irinos-Tool.

The same configuration file is required by the measurement application. It must be located in the same directory as the MscDll.dll.
Copy the file Msc.cfg from the directory of the Irinos-Tool into the directory of the measurement application. In case this file already exists, it must be replaced.

g) Start the application. The connection should now be established.

Maintenance, cleaning and disposal

10 Maintenance, cleaning and disposal

10.1 Maintenance

The Irinos-System has been designed for maintenance-free continuous operation.

It is recommenced to check regularly (e.g. monthly), if all connectors are fixed properly.

10.2 Cleaning

	Attention
	Unexpected / unintended reaction while cleaning the Irinos-System If the Irinos-System is in operation while cleaning, this may result in unintended actions. This may lead to personal injury or damage at machinery. Always turn off the Irinos-System before cleaning.

The cleaning work listed in the table below needs to be done regularly.

In case the Irinos-System is used in an environment with extensive dirt, the intervals may need to be reduced.

Interval	Cleaning work
3 Months	Remove oil and dust from all connector surfaces. Use a cleaning paper, which has been moistened with washing-up water. Wait until the connectors are completely dry before restarting the Irinos-System.
Monthly	Clean the case with a cleaning paper, which has been moistened with washing-up water. Use a scratch-less paper.

10.3 Disposal

Dispose the Irinos-System and the accessories at appropriate recycling- or disposal stations. Contact your local authorities for further information.
Do not dispose it in household waste.

Application notes

11 Application notes

11.1 Incremental encoders 1Vpp or TTL/RS422

11.1.1 Referencing for absolute measurement

Incremental encoders are no absolute measuring devices. In order to get absolute measurement values after start-up and after a signal error, referencing is required. The Irinos-Box IR-INC provides the following possibilities for referencing:

- Referencing via index

  The counter value is set to 0, if the index signal is passed.

- Referencing via software:

  The counter value can be set by software any time. It is possible to set the value 0 as well as to any other value (in the valid value range).

Both actions can be done via the opcode `opcSP` (0x35) of the MscDll.

Please note that the Irinos-Box IR-INC56 can only provide the technical possibility for referencing. Details depend on the respective measurement procedure. Hence referencing must already be taken into account in the planning phase. It is particularly important to have a plan for referencing after an incremental encoder error.

11.1.2 Input frequency

The input frequency of the incremental signals (TTL / RS422) or the signal period (1 Vpp) is limited. Details can be found in the specifications for the respective Irinos-Box.

In most measurement applications the theoretical input frequency is far below the limits. However, in reality it can be exceeded quickly through jerky movements. Examples are:

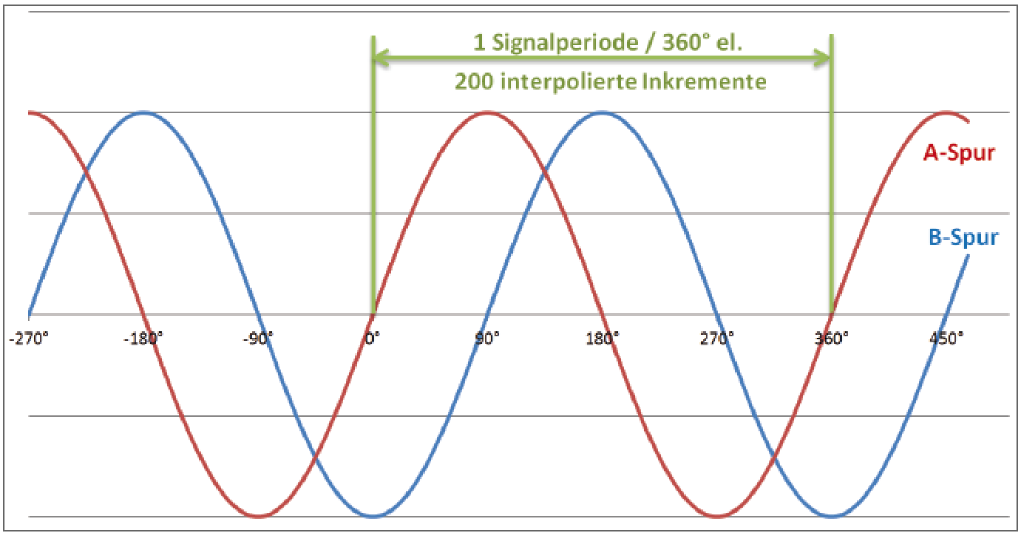
- “Cutting loose” at the beginning of a movement (crossing static friction)
- Mechanical stroke
- Jerky movement due to mechanical tensions

It is recommended to take this into account during the planning phase. If a jerky movement cannot be avoided, this must be considered in the measurement process (e.g. by referencing while moving).

11.1.3 Interpolation (only 1Vpp)

An incremental encoder with 1 Vpp - Interface provides two sine waves, each as a differential output signal. The phase shift between these is 90°. One signal period (i.e. 360°) relates to one incremental division of the encoder. The Irinos-Box IR-INC 56 separates a division into 200 incremental steps via analogue interpolation. Thereby the usable resolution of the encoder increases by 200.

Example: An incremental encoder has a resolution of 1.800 divisions/revolution. Via the interpolation, this results in 1.800 divisions/revolution * 200 increments/division = 360.000 increments/revolution.



1Vpp signal period

The accuracy and reliability of the interpolations depends on the quality of the differential sine signals. An ideal signal has the following characteristics:

- The differential voltage is 1 Vpp.
- The signal offset is 0, i.e. at 0° the signal always provides the same value.
- The phase shift between both sine signals is exactly 90°.

In reality, such a signal is rarely available. Because of this, the Irinos-Box IR-INC 56 has an integrated gain- and offset control (patent applies). It corrects these deviations within the allowed value range (see limiting values in the specification section of the datasheet).

Below or above the threshold values, a reliable interpolation is not possible. An integrated signal examination detects such errors. The error status can be read by software in parallel to the measurement. In case of an error, the counter channel should be reset. The referencing procedure should be redone.

Signal quality

The signal quality depends on various factors. Important ones are:

- Speed of the incremental encoder

  The higher the speed of the incremental encoder, the smaller the actual differential voltage. Some incremental encoders have a very good signal quality while standing still or at low speed. As soon as they are moved / turned, the signal quality decreases rapidly.

- Mechanical stability of the incremental encoder and the mechanics

  An incremental encoder or a measurement device, which does not move smoothly, leads to variations in the measurement signal.

- Adjustment of the incremental encoder

  Some incremental encoders (e.g. glass scales) need to be adjusted. An improper adjustment can lead to an insufficient sensor signal (especially for dynamic movements).

- Cable length and cable quality

  The longer the cable, the worse the signal quality.

  The more connectors are used, the worse the signal quality.

  A cable with insufficient shielding or wrong line impedance deteriorates the signal quality.

Often the reason for a signal problem is a combination of these factors.

Suggestions

- Please observe the cut-off frequency of the incremental encoder. It can be found in the encoder datasheet.

Attention: The cut-off frequency depends on the cable length.

- Check the signal quality during system setup. The signal levels should have enough margin to the limits.
  The Irinos-Tool provides a live-view of the signal levels.

- Make sure that no strong jerk can disturb the signal quality.

- Be prepared to enhance the measurement procedure by the functionalities "reset encoder error" and "restart referencing".

- Use short cables with sufficient shielding (this also applies to the connectors). Avoid extension cables. The Irinos-System allows placing the Irinos-Box next to the incremental encoder.

- Keep distance between the incremental encoder cable and possible sources of noise, like for example frequency changers or motor cables.

11.2 Power consumption

The power consumption of the Irinos-System depends on the number of Irinos-Boxes and the loads connected. Loads are for example probes or sensors.

The following table helps estimating the total power consumption. Please note that all values are guidelines. The actual power consumption may differ. Detailed information regarding the power consumption can be found in the respective datasheets.

Irinos-Box	Typical power consumption without external loads	Suggested value for estimating the power consumption with external loads
IR-MASTER 47	4,5 W	7 W (100mA / 24V for digital I/Os) 29 W (1A / 24V for digital I/Os)
IR-TFV 50	4 W	5 W
IR-AIN 54	4 W	4 W (24V supply output not used) 10 W (250mA / 24V for 24V supply output)
IR-INC 56	4 W	5,5 W (4 incremental encoders, each requiring 150mA)
IR-DIO 58	3 W	3W (Without I/O supply)

The power consumption of the Irinos-Boxes is constant. Critical factors for the calculation of the total power consumption are:

- External sensors / loads, which are connected to the 24V supply outputs of the Irinos-Boxes IR-MASTER, IR-AIN or IR-DIO.
- External actuators / loads, which are connected to the digital outputs of the Irinos-Box IR-MASTER or IR-DIO.

For most applications the additional power consumption for external loads is in the range of a few watts. Special care must be taken regarding loads switched via digital outputs, e.g. valves or lamps.

It is suggested to measure the actual power consumption after setup of the Irinos-System to ensure that there is enough margin.

11.3 Storing data in the non-volatile memory

The non-volatile memory has a limited amount of write operations. By the design of the Irinos-System, this limit is typically not reached. The following table lists the maximum number of write operations:

System function	Maximum number of write operations	Note
Diagnostic memory 190	4,8 Millions	
Measurement channel configuration	200.000	Executed via the opcode opcWCC.
Network configuration	200.000	Changing the IP settings.
Firmware update	100.000	

Specifications & dimensions

12 Specifications & dimensions

Detailed specifications can be found in the datasheets of the Irinos-Boxes.

12.1 Common specifications

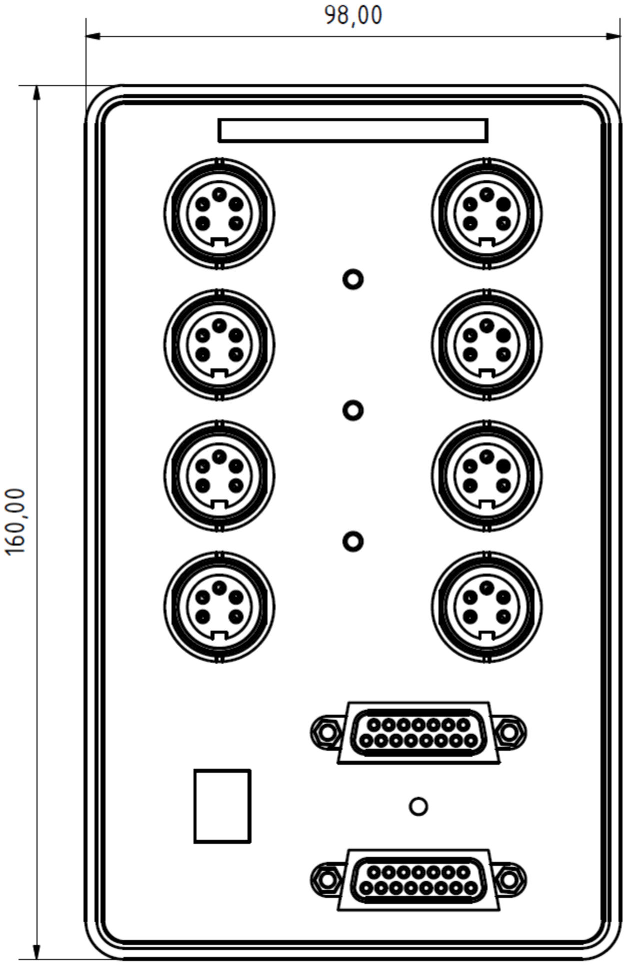
Sampling / measurement speed	
Static / continuous measurement	Sample rate min. 30 Hz, appropriate for online-view
Dynamic measurement	Up to 10.000 samples/s on all channels simultaneously, e.g.: 1 channel -> total: 10.000 samples/s 17 channels -> total: 170.000 samples/s 32 channels -> total 320.000 samples/s
Synchronisation	Simultaneous sampling of all measurement channels (also via multiple Irinos-Boxes)

System enhancement / ILink interface	
Maximum number of Irinos-Boxes	32
Maximum number of measurement channels	Depending on the Irinos-Boxes used. For example with IR-TFV max. 256 measurement channels.
Maximum ILink cable length	20 m (total length from first to last Irinos-Box)
ILink termination	automatic
Box addressing	automatic

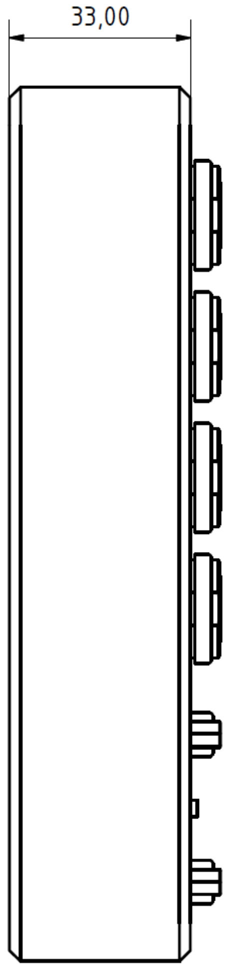
Case	
	Aluminium designer housing, black anodized, rear plate clear anodized, front plate with cover foil
Dimensions	Standard: 160 x 98 x 33 mm (H x W x D) Power supply IR-PU: 160 x 98 x 57 mm (H x W x D)
Protection class	Up to IP65 with appropriate connectors.
Mounting standard	2 tapped brushed M4 on rear side
Mounting accessories	o Mounting kit for hat / din rail o Mounting kit for front side mounting o Mounting bracket for aluminium profile 40mm o Stand for aluminium profile 40mm
Labelling	Labelling carrier for labelling the measurement channels and digital in-/outputs, appropriate for labels type Murrplastik ABB 17x9 (Order-No. Murrplastik: 86421020).

12.2 Dimensions Irinos-Box

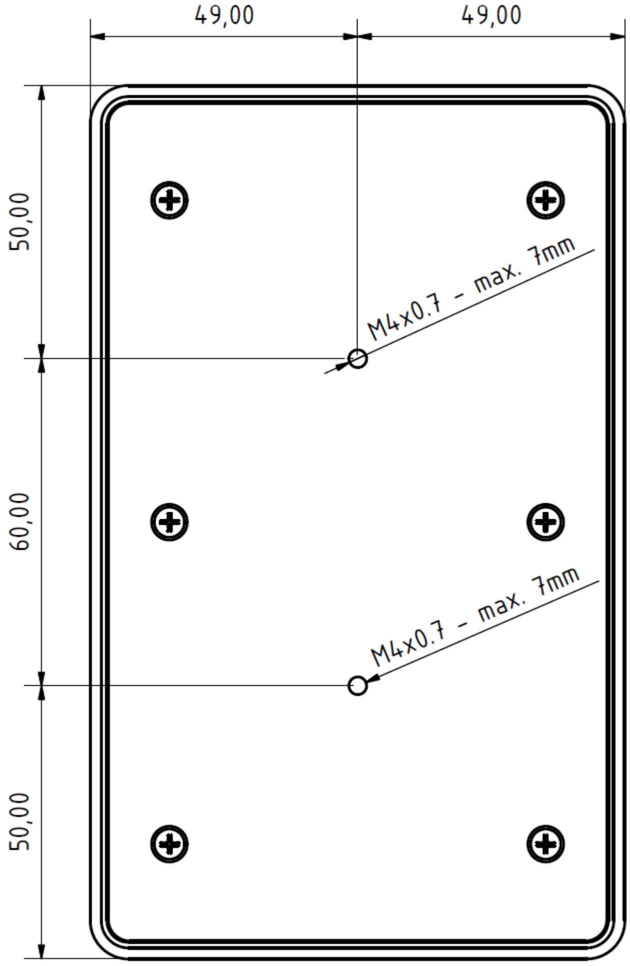
All dimensions are in mm.



Dimensions Irinos-Box (front side)



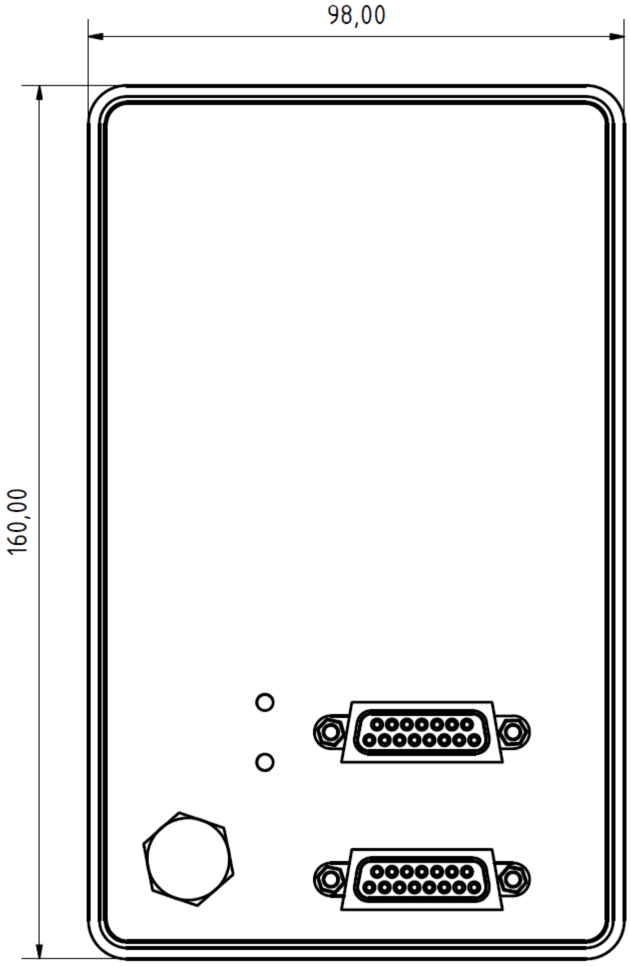
Dimensions Irinos-Box (side view)



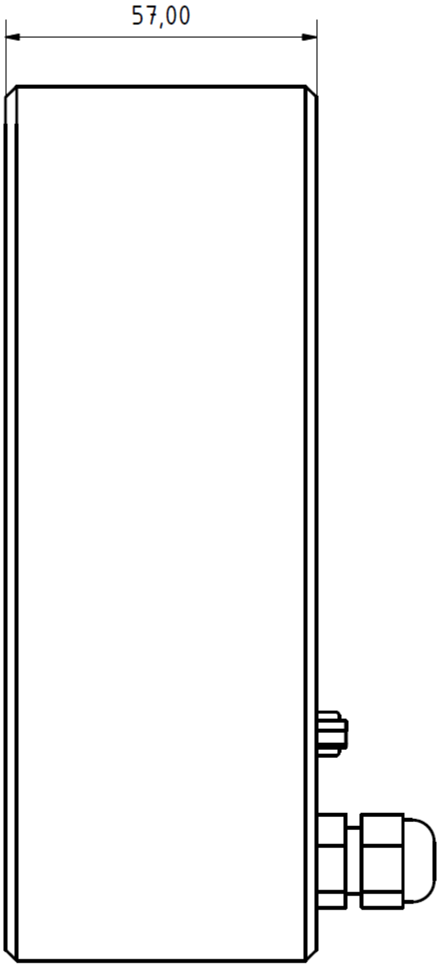
Dimensions Irinos-Box (rear view)

12.3 Dimensions power supply IR-PU50

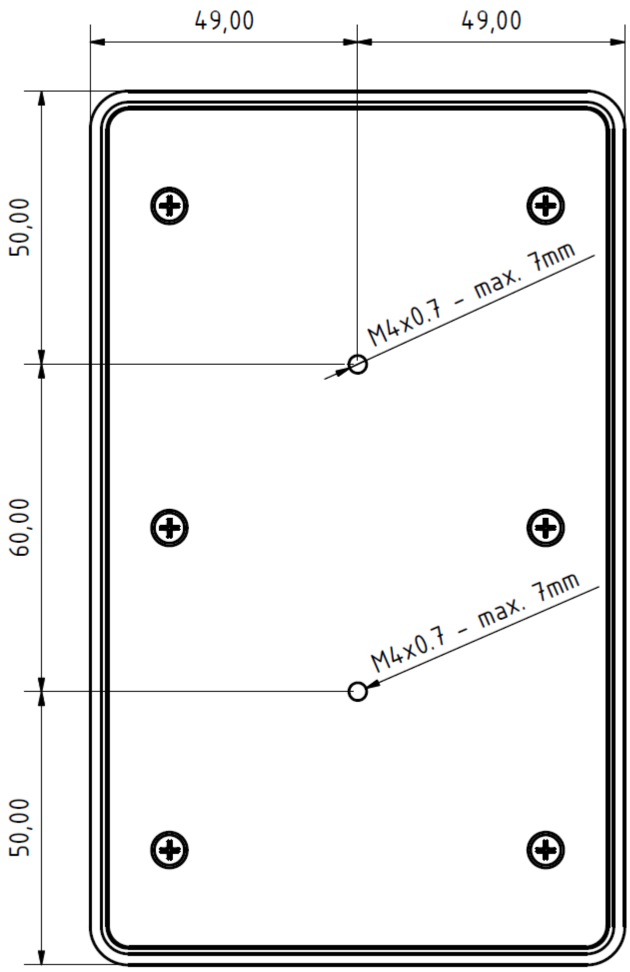
Alle dimensions are in mm.



Dimensions power supply (front view)



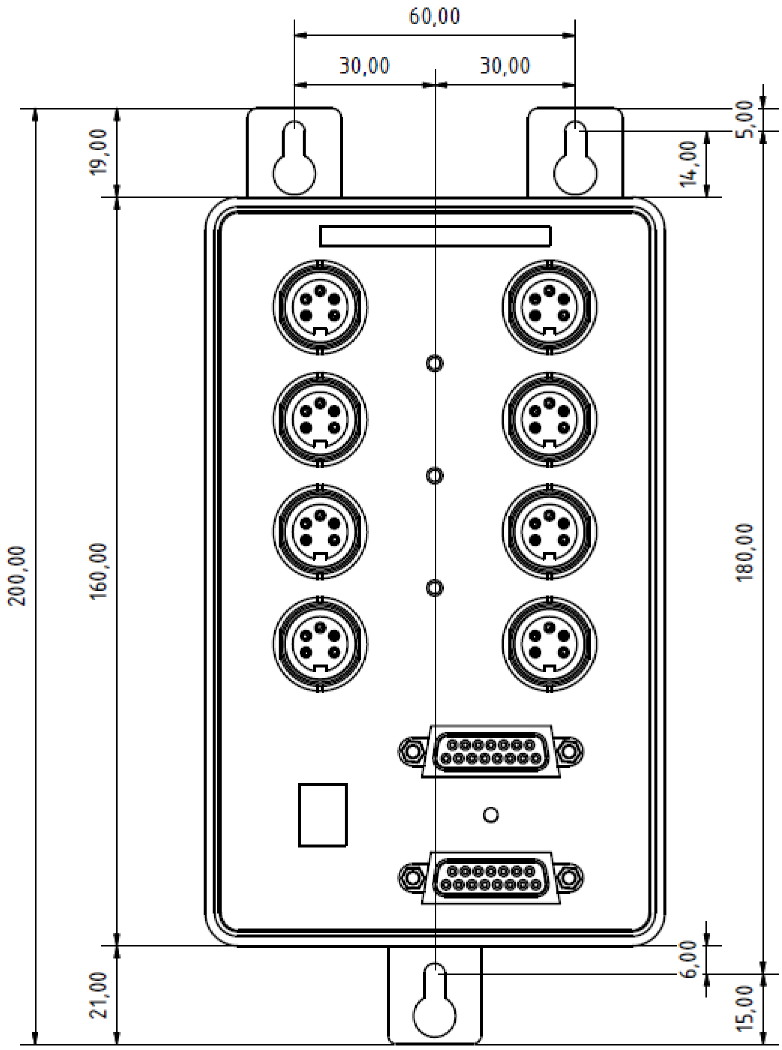
Dimensions power supply (side view)

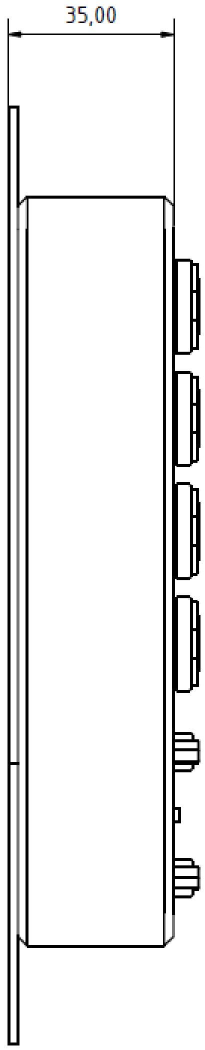


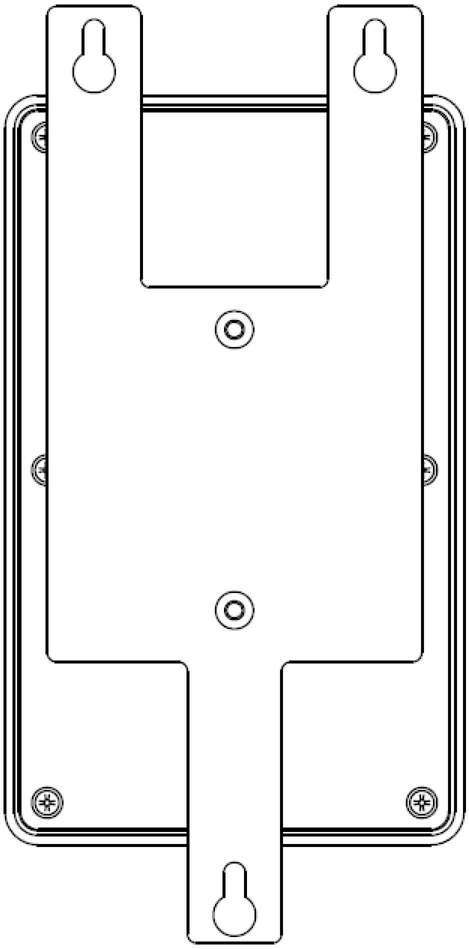
Dimensions power supply (rear view)

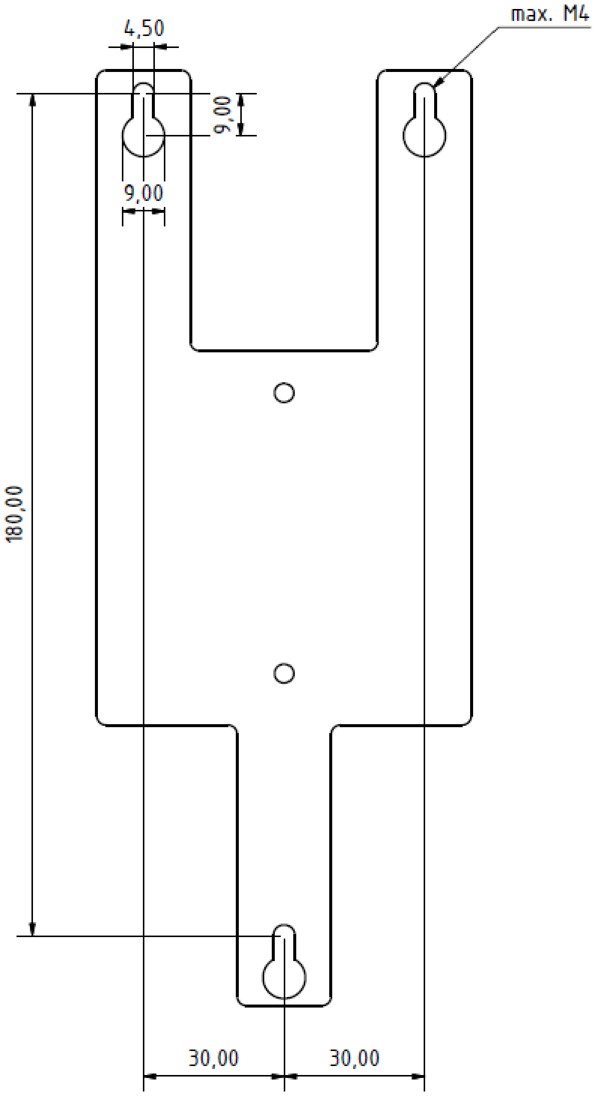
12.4 Dimensions front-side mounting adapter IR-MFFM-1

Alle dimensions are in mm.


60,00
30,00
30,00
19,00
200,00
160,00
21,00
5,00
14,00
180,00
6,00
15,00


35,00





4,50
max. M4
9,00
9,00
180,00
30,00
30,00