User Guide

Drives DataHub

The Drives DataHub is a software package that creates a gateway between Lenze inverters and an IIoT interface. The existing process control runs independently of the Drives DataHub and thus remains unaffected.

All common Lenze inverters can be connected without adapting the device firmware. Using a standardized, manufacturer-independent interface, the inverter data is made available in the required quality to a higher-level cloud system via MQTT.

The Drives DataHub thus implements the connection to all inverters in the Lenze portfolio and makes the data available to higher-level services such as Asset Management, Condition Monitoring, or Predictive Maintenance via an IIoT interface.

Purpose

Document history

Document history

Version	Date	Description
1.0	10.05.2023	First edition
1.1	07.08.2023	
1.2	23.07.2024	
1.3	16.12.2024	
1.4	01.04.2025	
1.5	07.11.2025	New Version

Overview of innovations:

- Chapter and command line updated

System requirements

Hardware requirements

The minimum system requirements are derived from the system requirements for Docker.

On Windows systems, the requirements are based on the Docker Desktop system requirements.

On Linux systems, they are either based on the Docker Desktop or the Docker Engine system requirements.

Depending on the planned number of device connections, additional hardware resources are required:

Device classes	Hardware requirements
10	CPU: Intel ® compatible 2-core processor | RAM: 4GB | HDD: 10 GB
50	CPU: Intel ® compatible 2-core processor | RAM: 4GB | HDD: 10 GB
250	CPU: Intel ® compatible 4-core processor | RAM: 8GB | HDD: 20 GB
1000	CPU: Intel ® compatible 8-core processor | RAM: 16GB | HDD: 40 GB

System requirements

Software requirements

Drives DataHub requires a current Docker environment version. On Linux-based systems, Docker Engine is sufficient. Windows-based systems require Docker Desktop.

Recommendation

To simplify the container management with a graphical user interface, Portainer can be used.

As an option, Portainer Agent can be added in addition to Portainer to extend its functionalities. Please refer to the Portainer documentation for details.

Installation

Instalation

The following sections describe the initial setup for Drives DataHub containers. Depending on the system used, an internet connection may be required.

Docker commands on Linux require root access. Only the root user can access Docker. Other users can only access Docker using “sudo”. To manage Docker as a non-root user, please refer to the Docker documentation.

Alternatively, Docker can be installed in rootless mode. For details, please refer to the Docker documentation.

Connect the Container Registry

All Docker images required to run Drives DataHub are provided via an Azure Container Registry. The login to the Container Registry can either be done via Portainer or via the Docker CLI.

Registry URL:
lenze.azurecr.io

User with read access to the Lenze Container Registry:

Username:
Please contact the sales department.

Password:
Please contact the sales department.

Add Lenze Container Registry in Portainer

To add the Lenze Container Registry to your local Portainer instance, follow the Portainer documentation part “Add an Azure registry”.

Login with Docker CLI

Login command:
Please contact the sales department.

Installation

Start containers using Docker Compose

To start up all Drives DataHub containers at once, a Docker Compose file can be used.

Either via Portainer or Docker CLI, the file provides the latest versions of all required containers as well as the Docker network and Docker volume setup.

The Docker Compose file specifies certain versions or tags for individual images for which compatibility is guaranteed.

Using other versions or tags than specified can lead to incompatibilities and thus malfunctions.

Compose with Portainer: Create a stack

A Docker Compose can be used in Portainer by creating a stack.

For details about adding a new stack in Portainer, please refer to the Portainer documentation part “Add a new stack”.

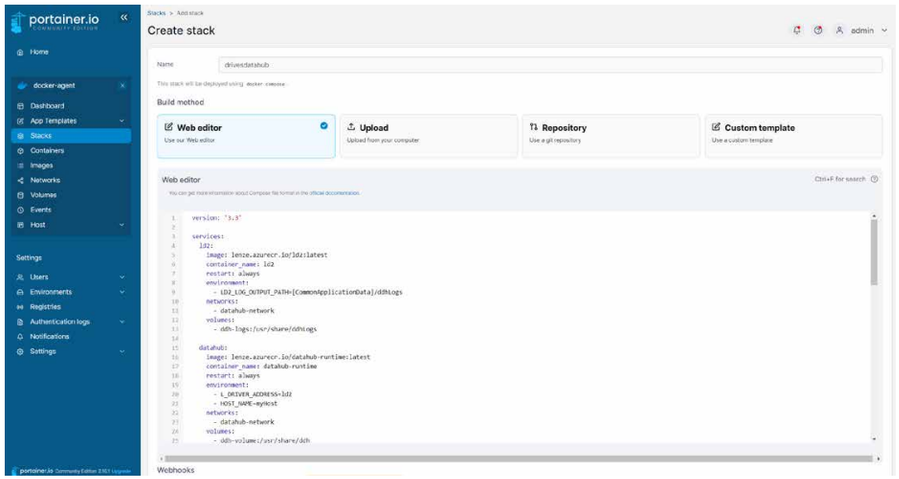
Using Portainer, a new stack can be created with the following Compose file.

In this example, the declared environment variable KEYCLOAK_PKS_PASSWORD contains the default password "LenzeAdmin123". This should be replaced by your own password for security reasons. Please note that you should only use the characters a...z and A...Z and the numbers 0...9 when defining your password. Otherwise, processing within the Docker container will fail. The password length must be at least 6 characters. The password in KEYCLOAK_PKS_PASSWORD is only used when the keystore file is recreated. This is done during the composing process if the file is not found in the declared file path KEYCLOAK_PKS_PATH.

This Compose file must be customized depending on the host environment of the Docker containers. The environment variable “KC_HOSTNAME” of the user management container must be set to the host name of the environment used.

Docker Compose file Download

To create the stack in Portainer, either the web editor can be used by copying the content of the Compose file or the upload process:



Compose with Docker CLI

To build and run multiple containers with a Docker Compose file, use the following command in the folder containing the Compose file:

$ docker compose up -d

This will load all the required images, create the containers, volumes, and network. Using “-d”, makes the stack run in the background (detached mode).

Environment variables

Variable	Container	Description	Values	Mandatory
POSTGRES_PASSWORD	db	Password for database user 'postgres'	lenze	yes
LD2_LOG_OUTPUT_PATH	ld2	Log folder	[CommonApplicationData]/ddhLogs	yes
L_DRIVER_ADDRESS	datahub	Ld2 container name	Ld2 container name	no
HOST_NAME	datahub	Used for global settings (log, settings and more)	myHost	no
DATABASE_HOST	datahub	DB container name	db	no
DATABASE_USER_NAME	datahub	DB user name	postgres	no
DATABASE_PASSWORD	datahub	DB user password. Must be same as in db container	lenze	yes
BROKER_HOST	datahub	Broker container name	broker	no
BROKER_PORT	datahub	Broker port	1883	no
RUNTIME_SERVER_PROTOCOL	api	Grpc server protocol	http	no
RUNTIME_SERVER_PROTOCOL	api	Grpc server protocol	http	no
RUNTIME_SERVER_PORT	api	Grpc server port	5001	no
RUNTIME_SERVER_NAME	api	Grpc server name	datahub	no
AUTH_SERVER_URL	api	Address of usermanagement container	http://usermanagement:8080/auth	no
SSL_PORT	ui	Used for return code 301. Needs to be set to mapped (to host) ui port	1443	yes
API_URL	ui	API Container name: Used for proxy passthrough to api container	api	no
AUTH_URL	ui	UserManagement Container name: Used for proxy passthrough to usermanagement container	usermanagement	no
KC_BOOTSTRAP_ADMIN_USERNAME	user-management	Keycloak admin for accessing keycloak admin ui. Additionally mapped port is needed. No default admin due to security reasons. This admin is optional and independent from the DataHub admin role.	kcAdmin	no
KC_BOOTSTRAP_ADMIN_PASSWORD	user-management	Keycloak admin password for accessing keycloak admin ui. Additionally mapped port is needed. No default admin due to security reasons.	LenzeAdmin123	no
KEYCLOAK_PKS_PASSWORD	usermanagement	Password for pks	LenzeAdmin123	yes
KEYCLOAK_PKS_PATH	usermanagement	Path to save the private key safes	/opt/keycloak/conf/keycloak.Keystore.pkcs12	yes
KC_HOSTNAME	user-management	Frontend URL	https://localhost:1443/auth	yes
KC_HOSTNAME_ADMIN	user-management	Admin Console URL	https://localhost:1443/auth	no
KC_HOSTNAME_BACKCHANNEL_DYNAMIC	user-management	By default backchannel URLs are dynamically resolved from request headers to allow internal and external applications. If all applications use the public URL this option should be enabled.	**true**, false	yes
KC_HOSTNAME_STRICT	user-management	Disables dynamically resolving the hostname from request headers. Should always be set to true in production, unless proxy verifies the Host header.	true, **false**	yes
KC_HTTP_ENABLED	user-management	Enables the HTTP listener.	**true**, false	yes
KC_LOG_LEVEL	user-management	Keycloak log level	DEBUG,ERROR,WARN,INFO,FATAL	no
KEYCLOAK_DB_PATH	user-management	Database path	/opt/keycloak/data/h2/keycloakdb.mv.db	yes

Installation

Change container names

The container names can be changed within the compose file by changing the property of the 'container_name' field. The container name is used as the container hostname within the Docker network.

Please refer to the internet standards (Request for Comments RFC 952) for protocols which specify that hostnames may contain only the ASCII letters a through z (in a case-insensitive manner), the digits 0 through 9, and the hyphenminus character ('-'). The original specification of hostnames required that labels start with an alpha character and not end with a hyphen.

Installation

Start containers individually

Not necessary if Compose/stack is used.

If required, containers can be started and stopped via Portainer in the “Containers” section.

Installation

Set up a system without internet connection

Docker containers are based on Docker images which can be downloaded from Container Registries hosted on the internet. An initial internet connection is required to download the images. The system that downloads the images needs a running Docker environment to connect to the Container Registries.

After pulling the images on a device connected to the internet, they can be exported to an archive via the command line to transfer them to the offline device.

Portainer

Export and import can be performed via Portainer:
To export an image, please refer to the Portainer documentation section “Export an image”.

To import an image, please refer to the Portainer documentation section “Import an image”.

Docker CLI

1. Export image
Command to export existing images from the local image store:

Save an image to a .tar archive:
$ docker save --output datahub.tar {your image name or ID}

Example:
$ docker save --output datahub-runtime.tar datahub-runtime

Refer to the Docker documentation for details.

2. Import image
Command to import Docker images from an archive:

Load the .tar archive to Docker:
$ docker load --input datahub.tar

Example:
$ docker load --input datahub-runtime.tar

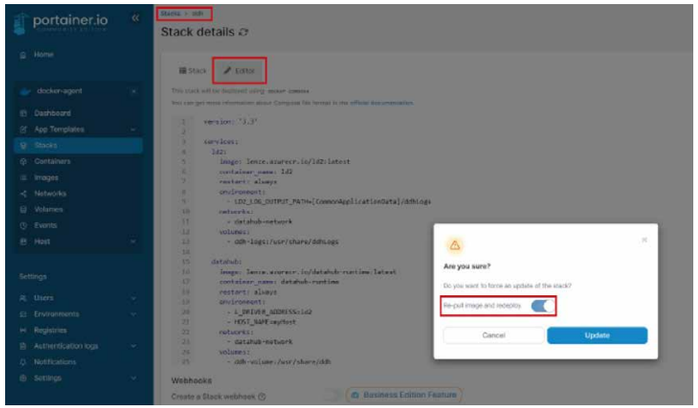
Refer to the Docker documentation for details.

Installation

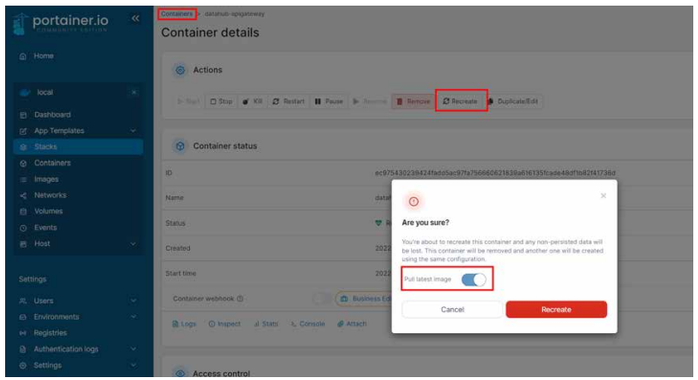
Update running containers

Portainer

To update a complete stack of running containers, including a re-pull of all images, open the stack details page using the editor view and press “Update”:



To update individual running containers using Portainer, the container can be recreated with all its settings while loading the latest image from the Container Registry. Therefore, Portainer provides the recreation on the containers details page:



Docker CLI

Stop the running application via:
$ docker compose down

This will remove the complete stack. Any changed data within the containers will be lost and the networks will be removed. The volumes will not be removed. Refer to the Docker documentation for details.

Update the Compose file (if necessary, e.g. for changing image tags) and recreate the stack with:
$ docker compose up -d

Installation

Remove Drives DataHub containers

Attention: Removing “ddh-volume” will also remove any active licenses.

Portainer

Via Portainer, the individual containers or the entire stack can be removed. Refer to the Portainer documentation for details, sections “Remove a container” and “Remove a stack”.

To fully remove all data, the images, volumes, and networks need to be removed separately.

Docker CLI

To remove the complete stack created via docker compose run 'docker compose down --rmi "all" -v'.

This stops and removes containers, networks, volumes and images which were created by 'docker compose up' initially.

$ docker compose down --rmi "all" -v

Configuration

Start Drives DataHub user interface

The Drives DataHub user interface container exposes the user interface via port 443. This access is encrypted and uses https. In the default Compose file, this port is mapped to port 1443 and can be changed by the user. If the mapped port of the UI container is changed, this port must also be changed in the environment variable “SSL_PORT” of the UI container, and in the environment variable “KC_HOSTNAME_URL” of the user management container.

Drives DataHub provides a self-signed certificate. This is not trusted by the web browser by default. Therefore, the user has to trust this certificate to access the user interface via https.

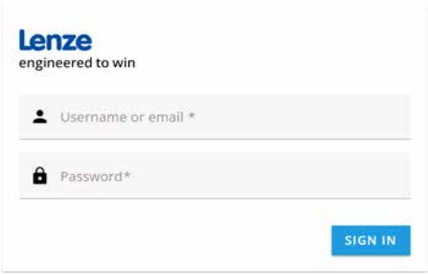
The recommended web browser is the latest Microsoft Edge version.

User sign in

If the UI container is running, the user interface can be opened via the URL “https://Hostname:1443”. Please insert the host name according to the environment used.
A login screen appears asking for username and password. The initial credentials for the admin user account are:

Username:
Please contact the sales department.

Password:
Please contact the sales department.



These credentials must be changed after the first login. A new password must follow these rules:

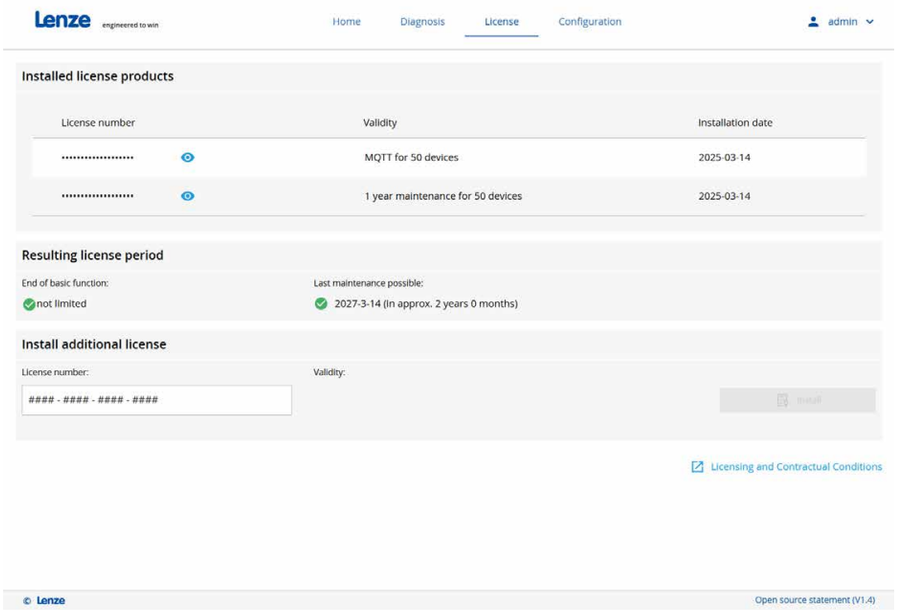
- Minimum length: 8 characters
- Maximum length: 128 characters
- The last 10 passwords must be different (in case of password change)
- The password must not be the same as the username
- The password must not be the same as the email entered in the user account
- The password must contain 1 upper case letter, 1 lower case letter, and 1 digit

The license terms will be shown after initial login and need to be accepted to use Drives DataHub.

Configuration

Add a perpetual or maintenance license

Open the License view via the main menu. This view provides an overview of all installed licenses including their product validity and the installation date. The installed licenses result in a license period for the basic function and a maintenance period. During the maintenance period, software updates are possible.



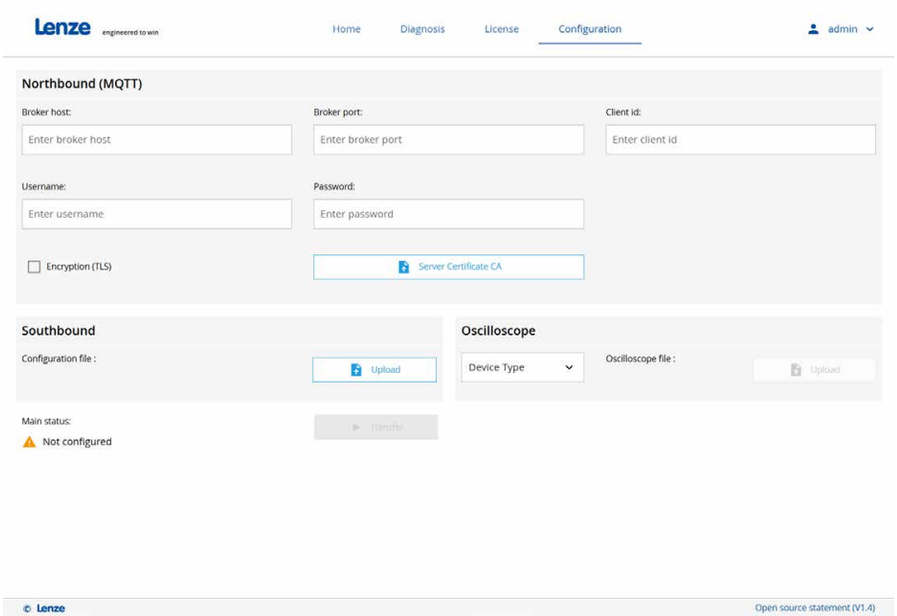
To install an initial or additional license at the bottom of the view, the license number can be pasted into the text field. An instant check indicates the validity of the inserted license, and clicking “Install” adds the license to the running Drives DataHub instance and will be shown in the list of installed license products.

Configuration

Initial configuration of Drives DataHub

After the initial start of Drives DataHub and after a version update, the configuration needs to be done via the web user interface.

Open the Configuration view via the main menu. This view provides an overview of the northbound and southbound configuration.

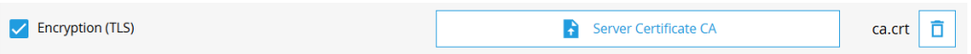


The northbound configuration for MQTT is used to specify the MQTT broker to which the data from the field devices is to be sent. The configuration can be done via the following text fields:

- Broker host: IP address or DNS name of the MQTT broker *
- Broker port: Port of the MQTT broker *
- Client ID: MQTT client ID which is used by Drives DataHub when connecting the MQTT broker *
- Username/Password: Username and password for authentication (if requested by the MQTT broker)
- Encryption (TLS): Activate TLS encryption. Minimal supported version is TLS 1.2
- Server Certificate CA: Upload a CA server certificate if required by the broker. Accepted formats are .cer and .crt format

When using Drives Datahub with a TLS-encrypted connection, it is necessary to upload a CA server certificate to verify the server's identity and ensure secure communication. The CA server certificate is required to validate the server's certificate. Without this certificate, the client cannot verify the server's authenticity, which can lead to security risks.

An uploaded certificate can be removed from the configuration by clicking the bin button.



To configure which devices should be connected on the southbound, a csv-file* is required. This file has four columns. In the first column the device IP address must be given. In the second column the device type must be given. The third column is optional and can be used to give a file name of a oscilloscope configuration.

Default oscilloscope configuration files are included in the container. For details about the included oscilloscope configuration and their file names, see chapter 'oscilloscope configuration'.

In the fourth column an oscilloscope recording interval can be given. This changes the time interval between two oscilloscope recordings. The interval is optional and the default value is 3 minutes if no interval is written to the csv file. The interval in whole numbers can be specified in the following time units: s (seconds - default), m (minutes), h (hours), d (days), a(years).

* Fields are mandatory

Configuration

Example csv-file

192.168.17.2;i550motec;i550motec.los;5m

192.168.17.3;i950;i950.los;10h

192.168.174;8400TL;;90s

The first and the second device are using an oscilloscope interval of five minutes and ten hours. The third devices does not use an oscilloscope configuration, therefore the interval has no effect.

Available device types for the second column:
i950 cabinet (i950), i550 cabinet (i550), i550 motec (i550motec), 8400 StateLine (8400SL), 8400 HighLine (8400HL), 8400 TopLine (8400TL), 9400 HighLine (9400HL), 8400 protec (8400protec) and 8400 motec (8400motec)

Basic ontologies

Drives DataHub includes a set of basic ontologies for various device types, which are used to continuously send data from the connected field devices. These ontologies are sent via MQTT using different MQTT topics. The data of the field devices are read out at a fixed interval.
Supported device types: i950 cabinet, i550 cabinet, i550 motec, 8400 StateLine, 8400 HighLine, 8400 TopLine, 9400 HighLine, 8400 protec and 8400 motec

Topic	Interval
LifeDiagnosis	At startup and every hour
DriveInfo	At startup and every 24 hours
Diagnosis	Every 15 seconds
Process	Depending on the oscilloscope configuration. If a forced trigger is used, the recording is sent to the device every 3 minutes. If a variable or channel trigger is used, the interval depends on the frequency of the trigger.

In addition to the configurations of device connections and oscilloscope recordings described above, the ontologies and, within defined limits, the topic structures and the payload format of the published data can be customized.
For enquiries, please contact the sales department.

Configuration

Topic and payload format

Topic structure:

*Lenze/*IpAddress*/*DeviceType*/*Ontology**

Example:

Lenze/192.168.12.2/I550MOTEC/Diagnosis

Telemetry payload format

Json-based example for i950 cabinet

```
{
    "Diagnosis": {
        "drive.state": 34512,
        "drive.utilization": 0,
        "drive.heatsinktemperature": 33.1,
        ...
        "TimeStamp": "2022-12-01T08:32:45.4647272+00:00"
    }
}
```

Oscilloscope payload format

Json-based example for i950 cabinet

```
{
    "PROCESS": {
        "timestamp": "2023-01-26T14:04:18.7749873+00:00",
        "timebase": 0.5,
        "samplerate": 0.005,
        "channelcount": 4,
        "channel1": {
            "name": "0x606C:000 Actual speed",
            "unit": "rpm",
            "data": [
                -0.205,
                ...
            ]
        },
        "channel2": {
            "name": "0x6077:000 Actual torque",
            "unit": "%",
            "data": [
                0.155,
                ...
            ]
        },
        "channel3": {
            "name": "0x6078:000 Actual current",
            "unit": "%",
            "data": [
                0.1,
                ...
            ]
        },
        "channel4": {
            "name": "0x2DDD:000 Output frequency",
            "unit": "Hz",
            "data": [
                0.1,
                ...
            ]
        }
    }
}
```

Configuration

Oscilloscope configuration

Multiple default oscilloscope configurations are included in the Drives DataHub container.
They are available for a forced trigger and a channel trigger.

Forced trigger

File names: i950.los, i550motec.los, i550.los, i550protec.los; 9400HL.los; 8400SL.los; 8400HL.los; 8400TL.los; 8400protec.los; i750.los.

The following device types and settings are included:

i950 cabinet servo inverter:
- Time base: 500 ms
- Sample rate: 5 ms
- Trigger: Forced trigger
- Delay: 0%
- Channels:

Channel	Parameter
1	0x606C:000 Actual speed
2	0x6077:000 Actual torque
3	0x6078:000 Actual current
4	0x2DDD:000 Output frequency

i750 cabinet servo inverter:
- Time base: 500 ms
- Sample rate: 5 ms
- Trigger: Forced trigger
- Delay: 0%
- Channels:

Channel	Parameter
1	0x606c:000 Actual speed
2	0x6077:000 Actual torque
3	0x6078:000 Actual current
4	0x2DDD:000 Output frequency

9400 HighLine servo inverter:
- Time base: 500 ms
- Sample rate: 5 ms
- Trigger: Forced trigger
- Delay: 0%
- Channels:

Channel	Parameter
1	.MI_dnActualMotorSpeed_s
2	.MI_dnActualMotorTorque_n
3	.MI_dnActualMotorCurrent_n
4	.MI_dnActualMotorFreq_s

8400 protec frequency inverter*:
- Time base: 500 ms
- Sample rate: 5 ms
- Trigger: Forced trigger
- Delay: 0%
- Channels:

Channel	Parameter
1	LS_MotorInterface.nMotorSpeedAct_a
2	LS_MotorInterface.nDCVoltage_a
3	LS_MotorInterface.nStatorCurrentIS_a
4	LS_MotorInterface.nMotorFreqAct_a

i550 protec:
- Time base: 139.912 ms
- Sample rate: 0.875 ms
- Trigger: Forced trigger
- Delay: 0%
- Channels:

Channel	Parameter
1	0x6044:000 (P783:000) Actual speed
2	0x6077:000 (P107:000) Actual torque
3	0x6078:000 (P103:000) Actual current
4	0x2DDD:000 (P100:000) Output frequency

i550 cabinet, i550 motec frequency inverter:
- Time base: 500 ms
- Sample rate: 5 ms
- Trigger: Forced trigger
- Delay: 0%
- Channels:

Channel	Parameter
1	0x6044:000 (P783:000) Actual speed
2	0x6077:000 (P107:000) Actual torque
3	0x6078:000 (P103:000) Actual current
4	0x2DDD:000 (P100:000) Output frequency

8400 HighLine/TopLine frequency inverter*:
- Time base: 500 ms, 255 ms (SL)
- Sample rate: 5 ms
- Trigger: Forced trigger
- Delay: 0%
- Channels:

Channel	Parameter
1	LS_MotorInterface.nMotorSpeedAct_a
2	LS_MotorInterface.nDCVoltage_a
3	LS_MotorInterface.nStatorCurrentIS_a
4	LS_MotorInterface.nMotorFreqAct_a

8400 StateLine frequency inverter*:
- Time base: 255 ms
- Sample rate: 5 ms
- Trigger: Forced trigger
- Delay: 0%
- Channels:

Channel	Parameter
1	LS_MotorInterface.nMotorSpeedAct_a
2	LS_MotorInterface.nDCVoltage_a
3	LS_MotorInterface.nStatorCurrentIS_a
4	LS_MotorInterface.nMotorFreqAct_a

* Requires firmware from version 12.00.00

Configuration

Channel trigger

All settings are identical to the forced trigger. Only the trigger is set to channel 1.

The edge is set to rising with a value of 50 rpm.

File names:

- i950_SpeedTrigger.los,
- i550motec_SpeedTrigger.los and i550_SpeedTrigger.los
- 9400_SpeedTrigger.los
- 8400SL_SpeedTrigger.los; 8400HL_SpeedTrigger.los; 8400TL_SpeedTrigger.los; 8400protec_SpeedTrigger.los
- i750_SpeedTrigger.los

Configuration

Upload oscilloscope configuration file

Reusing a saved configuration only makes sense for inverters of the same type (e.g. i950 cabinet servo inverters), as otherwise incorrect values will be displayed due to unadapted scaling of the oscilloscope channels!

Oscilloscope configuration files can be created using a Lenze engineering tool such as EASY Starter. After saving the oscilloscope configuration to a file of type .los, this file can be uploaded to Drives DataHub from the configuration view.

Once the file is opened, the upload starts immediately. During the upload, the file is checked for syntax and size. If the upload is successful, the file name is displayed. If an error occurs, an error message is displayed.

The configuration files can be several megabytes in size if they contain recordings. These are not required for configuration. In this case it is recommended to import the file with the EASY Starter and save it again without the recorded values.

If a file is uploaded with an existing file name, the existing file will be overwritten without warning. After the upload, the oscilloscope configuration .los-files can be assigned to a device address in a new configuration file (*.csv) by the file name.

Restrict oscilloscope file Download

The file name of the configuration file may contain the following characters:

- The letters A to Z in upper and lower case
- The digits 0 to 9
- The umlauts ä ö ü ß
- The characters - and _

Limitations

Drives DataHub detects when an oscilloscope command is sent to a device from outside the Drives DataHub (e.g. initiated by EASY Starter). In this case, the device is set to a blocked list for a fixed period of 24 hours. During this period, no oscilloscope commands are sent from Drives Datahub to the device to not interrupt the recordings of other tools.

Configuration

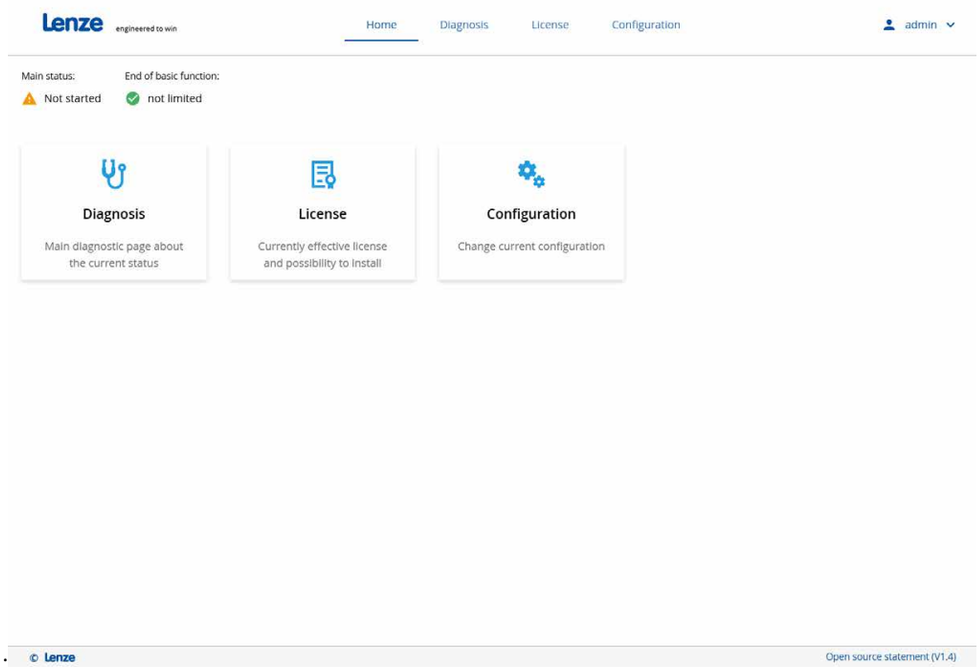
Update running configuration

To update a running configuration, open the configuration view, edit the MQTT connection settings, and upload a new csv-file for the southbound configuration. Starting this new configuration will stop and replace the old configuration.

Operation

Main menu

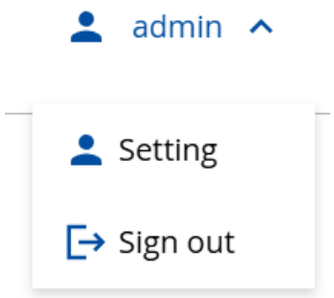
The main menu provides an overview of the main status of the software and the remaining time for the basic function depending on the remaining license period. In addition, all available views are displayed



Operation

User menu

The currently signed-in admin user can sign out via the user menu. In addition, the password can be changed in the user setting.



Operation

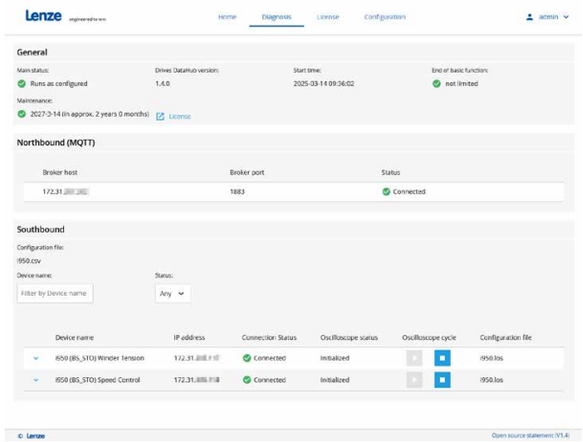
Diagnosis

The diagnosis view provides an overview of the software and the running configuration. The general information shows the main status, the installed version, and the start time of the running instance. The main status displayed in the main menu and in the diagnosis view indicates whether the loaded configuration is valid.

In addition, information about the installed license is provided, including the date on which the basic function and maintenance period expire.

The “Northbound” section provides information about the configured MQTT broker and the connection status.

The “Southbound” section provides information about the loaded configuration and the configured devices with the individual connection status.



Operation

Oscilloscope Control

The Oscilloscope status shows the current status of the cyclic oscilloscope processing for each device. When starting a new configuration, the oscilloscope processing is in state: Initialized

If an oscilloscope configuration is configured for the device, this status is stored until the oscilloscope configuration is transferred to the device.

If no oscilloscope configuration is configured for the device, the oscilloscope control remains in this state.

For devices that do not have an oscilloscope function, the status: Unavailable is displayedDuring an oscilloscope measurement, the following states are possible: Waiting, Triggered and Completed.

In addition the status can change to Error, Blocked or Stopped. In case of an error, recording is retried after 10 minutes.

The Blocked status indicates that the device is on the blocked list according to the limitations described in the configuration section. The Stopped status is initiated by controlling the oscilloscope cycle.

The oscilloscope cycle column includes two buttons. After loading a configuration file containing an oscilloscope configuration, the oscilloscope cycle is in running mode. Clicking the stop button stops the automatic oscilloscope cycle and no further oscilloscope recordings are started on the device. The oscilloscope processing status changes to 'Stopped'.

The automatic oscilloscope cycle can be continued by clicking the start button. The oscilloscope recordings are then continued at the set interval with the given configuration file.

Use Cases

1) The cyclic request of oscilloscope data can be stopped explicitly by the Stop button for individual devices, so that conflict-free parallel oscilloscope recordings are possible with other engineering tools (e.g. EASY Starter).

2) If there is a conflict during cyclic oscilloscope recording with an ongoing parallel oscillogram recording (e.g. initiated by the EASY Starter), the cyclic recording is blocked and only retried after 24 hours. If the parallel recording is completed earlier than this period, the blocked cyclical processing can be restarted directly via the Start button.

Oscilloscope status	Oscilloscope cycle	Configuration file
Stopped	▶ ■	i950.los
Completed	▶ ■	i950.los

Operation

Open source statement

Lenze is aware of the importance of using open source software and wants to give something back to the open source community. The software we develop may include certain software from the open source community that is subject to the respective open source licenses. We believe that using open source software allows us to develop better products for our customers. In the course of our work with open source software, we make an ongoing contribution to give back to the open source community.

A list of open source software used in Drives DataHub and the corresponding licenses is available via a link in the footer of the user interface.

The licensing terms and conditions of the open source software components used in this product can be found in the “License” directory. To copy all licensing terms and conditions from the container to the host system, the following commands can be used:

$ docker create --name datahub-dummy lenzeautomation.azurecr.io/datahub-apigateway:latest

$ docker cp datahub-dummy:/app/SBOM /dest/to/file

$ docker rm -f datahub-dummy

Operation

Error handling

Error	Help
Error while downloading the Drives DataHub Docker images	Please review the credentials to access the Container Registry.
Containers cannot be started	Please review the Compose file and make sure that all required images are downloaded and all required containers are created.
User interface cannot be opened	The user interface runs internally on port 443 via https. Please check in the Docker Compose file which port of the host system is mapped to port 443 on the datahub UI container. Make sure that this port is available and not blocked by any firewall. Additionally, the self-signed certificate must be trusted in the web browser. The recommended web browser is Microsoft Edge. If the mapped port of the UI container is changed, this port must also be changed in the environment variable “SSL_PORT” of the UI container and in the environment variable “KC_HOSTNAME_URL” of the user management container.
No data published via MQTT	Please check the broker configuration and the status for northbound (MQTT) on the Diagnosis view.
No connected devices	Please check the list of connected devices on the Diagnosis view in the “Southbound” section. Make sure that the devices can be accessed from the computer running the Drives DataHub.

Security aspects

"Usermanagement" container: user data and private keys

The “usermanagement“ container and its “ddh-usermanagement“ volume contain security relevant data. Therefore, the DataHub administrator has to take care of them. The “ddh-usermanagement“ volume contains all user data and also contains the certificates and keys being used by the “user management“. The user data is stored in the “/data/“ subdirectory and all keys being used by the user management are stored in the “/conf/“ subdirectory.

Security aspects

Configuration features

Define own keys and certificates for the user management.

The user management of the DataHub needs a predefined set of keys for its work. These keys are used for encryption and signing. The keys are stored in Java Key Store file which is encrypted with password protection. The Key Store file path and the password are declared by environment variables KEYCLOAK_PKS_PASSWORD and KEYCLOAK_PKS_PATH within the Docker compose file. See the example in the chapter above. In this example the “ddh-usermanagement” volume of the “usermanagement” container is configured to match the path which is defined in KEYCLOAK_PKS_PATH and the Java Key Store file is saved to the “/conf/“ directory within the “ddh-usermanagement” volume.

The “usermanagement” container has created its set of default keys by executing the following Linux command shell based commands:

```
keytool -genkeypair -storepass $KEYCLOAK_PKS_PASSWORD -storetype PKCS12
-keyalg RSA -keysize 4096 -dname "CN=CA-DrivesDataHub" -alias kc.key -ext
"SAN:c=DNS:localhost,IP:127.0.0.1" -validity 9125 -keystore $KEYCLOAK_PKS_PATH
```

```
keytool -genkeypair -storepass $KEYCLOAK_PKS_PASSWORD -storetype PKCS12
-keyalg RSA -keysize 4096 -dname "CN=CA-DrivesDataHub" -alias kc.sig -ext
"SAN:c=DNS:localhost,IP:127.0.0.1" -validity 9125 -keystore $KEYCLOAK_PKS_PATH
```

```
keytool -genkeypair -storepass $KEYCLOAK_PKS_PASSWORD -storetype PKCS12
-keyalg RSA -keysize 4096 -dname "CN=CA-DrivesDataHub" -alias kc.sig.256 -ext
"SAN:c=DNS:localhost,IP:127.0.0.1" -validity 9125 -keystore $KEYCLOAK_PKS_PATH
```

```
keytool -genkeypair -storepass $KEYCLOAK_PKS_PASSWORD -storetype PKCS12
-keyalg RSA -keysize 4096 -dname "CN=CA-DrivesDataHub" -alias kc.enc -ext
"SAN:c=DNS:localhost,IP:127.0.0.1" -validity 9125 -keystore $KEYCLOAK_PKS_PATH
```

With the same keytool command line tool you can also replace the default keys by adding your own keys or generating new keys if necessary. For this, you have to shut down the “usermanagement” container and its dependent containers. Then you can access the Java Keystore file in the given access path KEYCLOAK_PKS_PATH and replace existing keys by according keytool commands. See the detailed keytool documentation for this. The keytool used here is part of any Java installation in Linux or Windows.

Appendix

Data ontologies i950 cabinet servo inverter

Life diagnosis		Address	
		Code	Subcode
Power-on time	drive.powerontime	2D81	2
Operating time	drive.operatingtime	2D81	1
Mains switching cycles	drive.mainswitchingcycles	2D81	4
Short circuit counter	drive.shortcircuitcounter	2D81	6
Earth fault counter	drive.earthfaultcounter	2D81	7

Drive Info		Address	
		Code	Subcode
Device name	drive.name	2001	0
Product code	drive.type	2000	1
Serial number	drive.serialnumber	2000	2
Firmware version	drive.firmware.version	2000	4
Communication module – type	drive.communication.module1.type	2002	10
Communication module – serial number	drive.communication.module1.serialnumber	2002	11
IP address	drive.communicaton.diagnose.ipaddress	2452	1
MAC address	drive.communicaton.diagnose.macaddress	2452	5
PROFINET: IP address	drive.communicaton.profinet.ipaddress	2382	1
PROFINET: MAC address	drive.communicaton.profinet.macaddress	2382	5
PROFINET: station name	drive.communicaton.profinet.stationname	2382	4
Ethernet/IP: IP address	drive.communicaton.ethernetip.ipaddress	23A2	1
Ethernet/IP: MAC address	drive.communicaton.ethernetip.macaddress	23A2	5
Motor name	drive.motor.name	2C01	10
Rated motor power	drive.motor.ratedpower	2C01	6
Rated motor speed	drive.motor.ratedspeed	2C01	4
cos φ of the motor	drive.motor.cosphi	2C01	8
Maximum current	drive.maxcurrent	2DDF	2
Rated current	drive.ratedcurrent	2DDF	1

Diagnosis		Address	
		Code	Subcode
DC-bus voltage	drive.dcbusvoltage	2D87	0
External supply voltage	drive.24vsupplyvoltage	2539	1
Application state	drive.applicationstate	5810	1
CiA status word	drive.state	6041	0
Cause of disable	drive.causeofdisable	282A	1
Error code	drive.errorcode	603F	0
Heatsink temperature	drive.heatsinktemperature	2D84	1
Actual motor temperature	drive.motor.temperature	2D49	5
Control board temperature	drive.controlboardtemperature	2539	2
Device utilization	drive.utilization	2D40	7
Motor utilization	drive.motor.utilization	2D4F	0

Appendix

Data ontologies i750 cabinet servo inverter

Life diagnosis		Address	
		Code	Subcode
Power-on time	drive.powerontime	2D81	2
Operating time	drive.operatingtime	2D81	1
Mains switching cycles	drive.mainswitchingcycles	2D81	4
Short circuit counter	drive.shortcircuitcounter	2D81	6
Earth fault counter	drive.earthfaultcounter	2D81	7

Drive Info		Address	
		Code	Subcode
Device name	drive.name	2001	0
Product code	drive.type	2000	1
Serial number	drive.serialnumber	2000	2
Firmware version	drive.firmware.version	2000	4
Communication module – type	drive.communication.module1.type	2002	10
Communication module – serial number	drive.communication.module1.serialnumber	2002	11
IP address	drive.communicaton.diagnose.ipaddress	2452	1
MAC address	drive.communicaton.diagnose.macaddress	2452	5
PROFINET: IP address	drive.communicaton.profinet.ipaddress	2382	1
PROFINET: MAC address	drive.communicaton.profinet.macaddress	2382	5
PROFINET: station name	drive.communicaton.profinet.stationname	2382	4
Ethernet/IP: IP address	drive.communicaton.ethernetip.ipaddress	23A2	1
Ethernet/IP: MAC address	drive.communicaton.ethernetip.macaddress	23A2	5
Motor name	drive.motor.name	2C01	10
Rated motor power	drive.motor.ratedpower	2C01	6
Rated motor speed	drive.motor.ratedspeed	2C01	4
cos φ of the motor	drive.motor.cosphi	2C01	8
Maximum current	drive.maxcurrent	2DDF	2
Rated current	drive.ratedcurrent	2DDF	1

Diagnosis		Address	
		Code	Subcode
DC-bus voltage	drive.dcbusvoltage	2D87	0
External supply voltage	drive.24vsupplyvoltage	2539	1
Application state	drive.applicationstate	5810	1
CiA status word	drive.state	6041	0
Cause of disable	drive.causeofdisable	282A	1
Error code	drive.errorcode	603F	0
Heatsink temperature	drive.heatsinktemperature	2D84	1
Actual motor temperature	drive.motor.temperature	2D49	5
Control board temperature	drive.controlboardtemperature	2539	2
Device utilization	drive.utilization	2D40	7
Motor utilization	drive.motor.utilization	2D4F	0

Appendix

Data ontologies 9400 HighLine servo inverter

Life diagnosis		Address	
		Code	Subcode
Power-on time	drive.powerontime	179	0
Operating time	drive.operatingtime	178	0
Mains switching cycles	drive.mainswitchingcycles	177	1

Drive Info		Address	
		Code	Subcode
Device name	drive.name	199	0
Product code	drive.type	200	0
Complete device	drive.completedevice	203	8
Serial number	drive.serialnumber	204	8
Firmware version	drive.firmware.version	99	0
Module in MXI1	drive.communication.module1.type	203	3
Module in MXI1 – serial number	drive.communication.module1.serialnumber	204	3
Module in MXI2	drive.communication.module2.type	203	4
Module in MXI2 – serial number	drive.communication.module2.serialnumber	204	4
Rated motor power	drive.motor.ratedpower	81	0
Rated motor speed	drive.motor.ratedspeed	87	0
cos φ of the motor	drive.motor.cosphi	91	0
Maximum current	drive.maxcurrent	22	0
Rated current	drive.ratedcurrent	88	0

Diagnosis		Address	
		Code	Subcode
DC-bus voltage	drive.dcbusvoltage	53	0
External supply voltage	drive.24vsupplyvoltage	65	0
Device state	drive.state	183	0
Error number	drive.errorcode	168	
Heatsink temperature	drive.heatsinktemperature	61	0
Actual motor temperature	drive.motor.temperature	63	0
Control board temperature	drive.controlboardtemperature	69	0
Device utilization	drive.utilization	64	0
Motor utilization	drive.motor.utilization	66	0

Appendix

Data ontologies i550 cabinet frequency inverter

Life diagnosis		Address	
		Code	Subcode
Power-on time	drive.powerontime	2D81	2
Operating time	drive.operatingtime	2D81	1
Mains switching cycles	drive.mainswitchingcycles	2D81	4
Short circuit counter	drive.shortcircuitcounter	2D81	6
Earth fault counter	drive.earthfaultcounter	2D81	7

Drive Info		Address	
		Code	Subcode
Device name	drive.name	2001	0
Product code	drive.type	2000	1
Serial number	drive.serialnumber	2000	2
Firmware version	drive.firmware.version	2000	4
PROFINET: IP address	drive.communicaton.profinet.ipaddress	2382	1
PROFINET: MAC address	drive.communicaton.profinet.macaddress	2382	5
PROFINET: station name	drive.communicaton.profinet.stationname	2382	4
Ethernet/IP: IP address	drive.communicaton.ethernetip.ipaddress	23A2	1
Ethernet/IP: MAC address	drive.communicaton.ethernetip.macaddress	23A2	5
Motor name	drive.motor.name	2C01	10
Rated motor power	drive.motor.ratedpower	2C01	6
Rated motor speed	drive.motor.ratedspeed	2C01	4
cos φ of the motor	drive.motor.cosphi	2C01	8
Maximum current	drive.maxcurrent	2DDF	2
Rated current	drive.ratedcurrent	2DDF	1

Diagnosis		Address	
		Code	Subcode
DC-bus voltage	drive.dcbusvoltage	2D87	0
CiA status word	drive.state	6041	0
Cause of disable	drive.causeofdisable	282A	1
Error code	drive.errorcode	603F	0
Heatsink temperature	drive.heatsinktemperature	2D84	1
Device utilization	drive.utilization	2D40	4
Motor utilization	drive.motor.utilization	2D4F	0

Appendix

Data ontologies i550 motec frequency inverter

Life diagnosis		Address	
		Code	Subcode
Power-on time	drive.powerontime	2D81	2
Operating time	drive.operatingtime	2D81	1
Mains switching cycles	drive.mainswitchingcycles	2D81	4
Short circuit counter	drive.shortcircuitcounter	2D81	6
Earth fault counter	drive.earthfaultcounter	2D81	7

Drive Info		Address	
		Code	Subcode
Device name	drive.name	2001	0
Product code	drive.type	2000	1
Serial number	drive.serialnumber	2000	2
Firmware version	drive.firmware.version	2000	4
PROFINET: IP address	drive.communicaton.profinet.ipaddress	2382	1
PROFINET: MAC address	drive.communicaton.profinet.macaddress	2382	5
PROFINET: station name	drive.communicaton.profinet.stationname	2382	4
Ethernet/IP: IP address	drive.communicaton.ethernetip.ipaddress	23A2	1
Ethernet/IP: MAC address	drive.communicaton.ethernetip.macaddress	23A2	5
Motor name	drive.motor.name	2C01	10
Rated motor power	drive.motor.ratedpower	2C01	6
Rated motor speed	drive.motor.ratedspeed	2C01	4
cos φ of the motor	drive.motor.cosphi	2C01	8
Maximum current	drive.maxcurrent	2DDF	2
Rated current	drive.ratedcurrent	2DDF	1

Diagnosis		Address	
		Code	Subcode
DC-bus voltage	drive.dcbusvoltage	2D87	0
CiA status word	drive.state	6041	0
Cause of disable	drive.causeofdisable	282A	1
Error code	drive.errorcode	603F	0
Heatsink temperature	drive.heatsinktemperature	2D84	1
Device utilization	drive.utilization	2D40	4
Motor utilization	drive.motor.utilization	2D4F	0

Appendix

Data ontologies i550 protec frequency inverter

Life diagnosis		Address	
		Code	Subcode
Power-on time	drive.powerontime	2D81	2
Operating time	drive.operatingtime	2D81	1
Mains switching cycles	drive.mainswitchingcycles	2D81	4
Short circuit counter	drive.shortcircuitcounter	2D81	6
Earth fault counter	drive.earthfaultcounter	2D81	7

Drive Info		Address	
		Code	Subcode
Device name	drive.name	2001	0
Product code	drive.type	2000	1
Serial number	drive.serialnumber	2000	2
Firmware version	drive.firmware.version	2000	4
PROFINET: IP address	drive.communicaton.profinet.ipaddress	2382	1
PROFINET: MAC address	drive.communicaton.profinet.macaddress	2382	5
PROFINET: station name	drive.communicaton.profinet.stationname	2382	4
Ethernet/IP: IP address	drive.communicaton.ethernetip.ipaddress	23A2	1
Ethernet/IP: MAC address	drive.communicaton.ethernetip.macaddress	23A2	5
Motor name	drive.motor.name	2C01	10
Rated motor power	drive.motor.ratedpower	2C01	6
Rated motor speed	drive.motor.ratedspeed	2C01	4
cos φ of the motor	drive.motor.cosphi	2C01	8
Maximum current	drive.maxcurrent	2DDF	2
Rated current	drive.ratedcurrent	2DDF	1

Diagnosis		Address	
		Code	Subcode
DC-bus voltage	drive.dcbusvoltage	2D87	0
CiA status word	drive.state	6041	0
Cause of disable	drive.causeofdisable	282A	1
Error code	drive.errorcode	603F	0
Heatsink temperature	drive.heatsinktemperature	2D84	1
Device utilization	drive.utilization	2D40	4
Motor utilization	drive.motor.utilization	2D4F	0

Appendix

Data ontologies 8400 TopLine frequency inverter

Life diagnosis		Address	
		Code	Subcode
Power-on time	drive.powerontime	179	0
Operating time	drive.operatingtime	178	0
Mains switching cycles	drive.mainswitchingcycles	177	1
Short circuit counter	drive.shortcircuitcounter	177	3
Earth fault counter	drive.earthfaultcounter	177	4

Drive Info		Address	
		Code	Subcode
Device name	drive.name	199	1
Product code	drive.type	200	0
Complete device	drive.completedevice	203	8
Serial number	drive.serialnumber	204	8
Firmware version	drive.firmware.version	99	0
MCI module	drive.communication.module1.type	203	3
MCI module – serial number	drive.communication.module1.serialnumber	204	3
Rated motor power	drive.motor.ratedpower	81	0
Rated motor speed	drive.motor.ratedspeed	87	0
cos φ of the motor	drive.motor.cosphi	91	0
Maximum current	drive.maxcurrent	22	0
Rated current	drive.ratedcurrent	88	0

Diagnosis		Address	
		Code	Subcode
DC-bus voltage	drive.dcbusvoltage	53	0
DC-bus voltage	drive.24vsupplyvoltage	65	0
Device state	drive.state	137	0
Error number	drive.errorcode	165	1
Heatsink temperature	drive.heatsinktemperature	61	0
Motor temperature	drive.motor.temperature	63	1
Temperature inside the controller	drive.controlboardtemperature	62	1
Device utilization	drive.utilization	64	1
Motor utilization	drive.motor.utilization	66	0

Appendix

Data ontologies 8400 HighLine frequency inverter

Life diagnosis		Address	
		Code	Subcode
Power-on time	drive.powerontime	179	0
Operating time	drive.operatingtime	178	0
Mains switching cycles	drive.mainswitchingcycles	177	1
Short circuit counter	drive.shortcircuitcounter	177	3
Earth fault counter	drive.earthfaultcounter	177	4

Drive Info		Address	
		Code	Subcode
Device name	drive.name	199	1
Product code	drive.type	200	0
Complete device	drive.completedevice	203	8
Serial number	drive.serialnumber	204	8
Firmware version	drive.firmware.version	99	0
MCI module	drive.communication.module1.type	203	3
MCI module – serial number	drive.communication.module1.serialnumber	204	3
Rated motor power	drive.motor.ratedpower	81	0
Rated motor speed	drive.motor.ratedspeed	87	0
cos φ of the motor	drive.motor.cosphi	91	0
Maximum current	drive.maxcurrent	22	0
Rated current	drive.ratedcurrent	88	0

Diagnosis		Address	
		Code	Subcode
DC-bus voltage	drive.dcbusvoltage	53	0
DC-bus voltage	drive.24vsupplyvoltage	65	0
Device state	drive.state	137	0
Error number	drive.errorcode	165	1
Heatsink temperature	drive.heatsinktemperature	61	0
Temperature inside the controller	drive.controlboardtemperature	62	1
Device utilization	drive.utilization	64	1
Motor utilization	drive.motor.utilization	66	0

Appendix

Data ontologies 8400 StateLine frequency inverter

Life diagnosis		Address	
		Code	Subcode
Power-on time	drive.powerontime	179	0
Operating time	drive.operatingtime	178	0
Mains switching cycles	drive.mainswitchingcycles	177	1
Short circuit counter	drive.shortcircuitcounter	177	3
Earth fault counter	drive.earthfaultcounter	177	4

Drive Info		Address	
		Code	Subcode
Device name	drive.name	199	1
Product code	drive.type	200	0
Complete device	drive.completedevice	203	8
Serial number	drive.serialnumber	204	8
Firmware version	drive.firmware.version	99	0
MCI module	drive.communication.module1.type	203	3
MCI module – serial number	drive.communication.module1.serialnumber	204	3
Rated motor power	drive.motor.ratedpower	81	0
Rated motor speed	drive.motor.ratedspeed	87	0
cos φ of the motor	drive.motor.cosphi	91	0
Maximum current	drive.maxcurrent	22	0
Rated current	drive.ratedcurrent	88	0

Diagnosis		Address	
		Code	Subcode
DC-bus voltage	drive.dcbusvoltage	53	0
DC-bus voltage	drive.24vsupplyvoltage	65	0
Device state	drive.state	137	0
Error number	drive.errorcode	165	1
Heatsink temperature	drive.heatsinktemperature	61	0
Temperature inside the controller	drive.controlboardtemperature	62	1
Device utilization	drive.utilization	64	1
Motor utilization	drive.motor.utilization	66	0

Appendix

Data ontologies 8400 protec frequency inverter

Life diagnosis		Address	
		Code	Subcode
Power-on time	drive.powerontime	179	0
Operating time	drive.operatingtime	178	0
Mains switching cycles	drive.mainswitchingcycles	177	1

Drive Info		Address	
		Code	Subcode
Device name	drive.name	199	1
Product code	drive.type	200	0
Complete device	drive.completedevice	203	8
Serial number	drive.serialnumber	204	8
Firmware version	drive.firmware.version	99	0
MCI module	drive.communication.module1.type	203	3
MCI module – serial number	drive.communication.module1.serialnumber	204	3
Rated motor power	drive.motor.ratedpower	81	0
Rated motor speed	drive.motor.ratedspeed	87	0
cos φ of the motor	drive.motor.cosphi	91	0
Maximum current	drive.maxcurrent	22	0
Rated current	drive.ratedcurrent	88	0

Diagnosis		Address	
		Code	Subcode
DC-bus voltage	drive.dcbusvoltage	53	0
Device state	drive.state	137	0
Error number	drive.errorcode	165	1
Heatsink temperature	drive.heatsinktemperature	61	0
Device utilization	drive.utilization	64	1
Motor utilization	drive.motor.utilization	66	0

Appendix

Data ontologies 8400 motec frequency inverter

Life diagnosis		Address	
		Code	Subcode
Power-on time	drive.powerontime	179	0
Operating time	drive.operatingtime	178	0
Mains switching cycles	drive.mainswitchingcycles	177	1

Drive Info		Address	
		Code	Subcode
Device name	drive.name	199	1
Product code	drive.type	200	0
Complete device	drive.completedevice	203	8
Serial number	drive.serialnumber	204	7
Firmware version	drive.firmware.version	99	0
MCI module	drive.communication.module1.type	203	3
MCI module – serial number	drive.communication.module1.serialnumber	204	3
Rated motor power	drive.motor.ratedpower	81	0
Rated motor speed	drive.motor.ratedspeed	87	0
cos φ of the motor	drive.motor.cosphi	91	0
Maximum current	drive.maxcurrent	22	0
Rated current	drive.ratedcurrent	88	0

Diagnosis		Address	
		Code	Subcode
DC-bus voltage	drive.dcbusvoltage	53	0
Device state	drive.state	137	0
Error number	drive.errorcode	165	1
Heatsink temperature	drive.heatsinktemperature	61	0
Device utilization	drive.utilization	64	1
Motor utilization	drive.motor.utilization	66	0