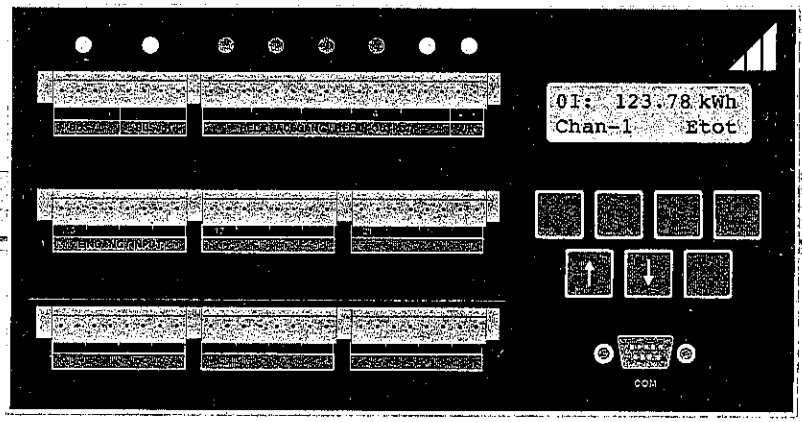
U1600

Energy Control System



Operating Instructions

3rd Edition 12th of August, 1994

GOSSEN METRAWATT CAMILLE BAUER

Published by: **GOSSEN-METRAWATT GmbH**
Thomas Mann Straße 16-20
D-90471 Nürnberg
Telefon (0911) 8602-0
Telefax (0911) 8602-669
BTX * 22228 #

Version of ECS Operating Software: **V1.31m / 06-06-94**

Table of contents

System Description

Introduction

The Energy Control System (ECS), for the transparent evaluation of cost centres, is made up of summation stations of type U1600.

Energy Meters

Each summation station is designed to be connected to up to 24 energy meters with pulse outputs. The data from the meter inputs is integrated, and stored in battery supported semiconductor memory.

Virtual Channels

Eight virtual channels are provided in order to generate totals and/or differences from the values supplied by the meters. Values from other summation stations, connected to the ECS LAN, may also be used to create virtual channels.

The keyboard can be used to show any of the stored readings on the display unit of the summation station.

ECS LAN

In the case of applications requiring more than 24 meters, a number of stations may be linked via the ECS LAN, a multimaster RS485 bus. Networking can be performed using two wire connections in bus or linear structure. The maximum length of a segment is 1200 m. Optical fibre transitions, using 4 wire technology, can be used to span distances of several kilometres between 2 summation stations.

Multimaster

The most significant advantage of the multimaster system is that each of the elements connected to the bus has unlimited access to the data and functionality of the whole system. A maximum of 255 summation stations can be connected to the ECS LAN. This means that a distance of around 300 km can be spanned without the need for additional signal amplifiers.

RS232

A PC connected to the RS232 serial interface has full access to all the stored system data. The parameter setting software **ECSoft 2**, which runs on a PC, manages the entire data of the ECS, and allows it to be evaluated in tabular form.

Programmable Relays

Four user programmable relays (change over contacts) allow messages to be sent or operations to be performed if certain criteria are met. For example, a relay output may be activated if the average power over the synchronisation period exceeds a defined value.

Summary of Available Readings

The following readings may be called for each meter input and each virtual channel:

Energy

Total Energy from Defined Starting Point

Etot	total energy regardless of tariff
EtotT1	total energy Tariff 1
EtotT2	total energy Tariff 2
EtotT1T2	total energy Tariff 1 plus Tariff 2

Total Energy for Defined Periods

EDay	for the current day and each of the last 10 days
EMon	for the current month and each of the last 12 months
EYear	for the current and the previous year and the year before that
EInt	synchronisation period readings for all intervals (list of readings)

Maximum Value of Synchronisation Period Readings with Date and Time
(only for physical channels)

Emax	the 10 highest values from all periods
EmaxDay	daily maximum for the current and each of the last 10 days
EmaxMon	monthly maximum for the current and last 12 months
EmaxYear	the maximum values of the current, the last and the year before last

Costs

Total Costs from Defined Starting Point

CostT1	Costs for Tariff 1
CostT2	Costs for Tariff 2
CostT1T2	Costs for Tariff 1 plus Tariff 2

Power

Instantaneous Value

Pmom	calculated from the time between the last two meter pulses

Average Power for Defined Periods

PDay	for the current day and each of the last 10 days
PMon	for the current month and each of the last 12 months
PYear	for the current and the previous year and the year before that
PInt	synchronisation period readings for all intervals (list of readings)

Maximum Value of Synchronisation Period Readings with Date and Time
(only for physical channels)

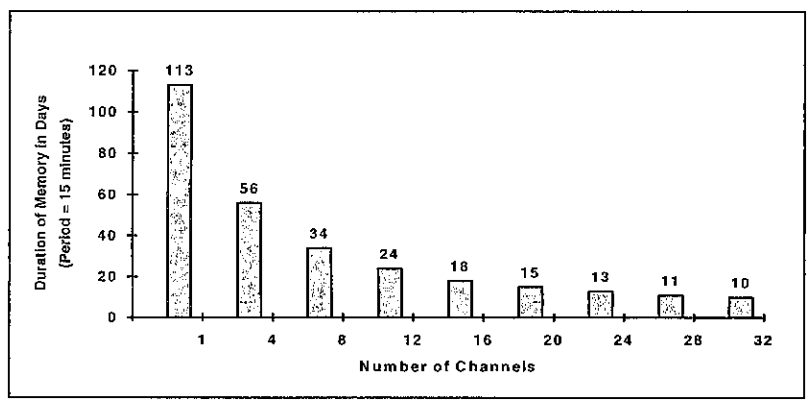
Pmax	the 10 highest values from all periods
PmaxDay	daily maximum for the current and each of the last 10 days
PmaxMon	monthly maximum for the current and last 12 months
PmaxYear	the maximum values of the current, the last and the year before last

Table of Available Readings

Energy							
Etot	EtotT1	EtotT2	EtotT1T2	EInt	EDay	EMon	EYear
Power							
Pmom				PInt	PDay	PMon	PYear
Maximum Energy *							
				EmaxInt	EmaxDay	EmaxMon	EmaxYear
Maximum Power *							
				PmaxInt	PmaxDay	PmaxMon	PmaxYear
Costs							
	CostT1	CostT2	CostT1T2				

* : only available for physical channels

Summary of Relationship between Duration of Memory and Number of Channels used for Synchronisation Period Readings (List of Readings):



No. of Channels	No. of Entries	Memory in Days with period = 15 mins
1	10922	113
4	5461	56
8	3276	34
9	2978	31
12	2340	24
16	1820	18
20	1489	15
24	1260	13
28	1092	11
32	963	10

Meter Inputs

24 optically separated meter inputs conforming to the S0 standard are provided. A physical meter channel is allocated to each input. So called Virtual Channels are generated from summarisations of the physical channels (as well as various summation stations connected to the ECS LAN). In addition, each summation station has eight of these **virtual channels**. Unused physical channels may take on the role of virtual channels.

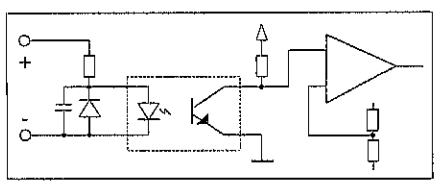
Binary Inputs

Regardless of the metering functions, the 24 input channels may be also used as **binary inputs**. A binary input recognises two voltage levels, logical '1' and '0' (the max. '0'-level is adjustable with the ECL command LEVEL). For example, input 24 can be used to perform switching between Tariff 1 and Tariff 2. The timing signal (synchronising pulse) from the energy company could also be connected, say, to input 23. The ECL command **INPUT** allows the status of the program to be queried.

Characteristics of the Input Circuits:

- ❑ Input voltage range: 0V .. 24V
- ❑ Input resistance: 5K Ohm
- ❑ Voltage level logical '1': > 4,5V
- ❑ Voltage level logical '0': < 2V
- ❑ Test voltage in/output: 500V

Meter Input



Rebound Period and Trigger Edges

The rebound period [pulse recognition time] and trigger edges can be adjusted separately for each of the 24 inputs. The scanning time is 5 ms in all 24 channels.

Rebound Period

- ❑ Rebound period adjustable between 10ms to 2.55s, in 10ms steps (ECL command: PULSE)

Trigger Edge

- ❑ Trigger '+' (1): Triggers on change from 0 to 1 (ECL command: EDGE)
- ❑ Trigger '-' (0): Triggers on change from 1 to 0

The rebound period also applies if an input is being used as a binary input. Therefore, if a rebound period of 1 s is set, only those signals remaining constant at the 1 or 0 level are processed. The input status display on the operating panel also only shows signals lasting longer than the rebound period.

Channel Start/Stop

Start/Stop Function

The Start/Stop function is used to control acceptance of the meter impulses of an input (ECL command: STARTSTOP).

- ❑ A virtual channel created using a 'differential connection' can be influenced analogously using the Start/Stop function.
- ❑ Because the binary input status of a channel is not affected, the function can be used to screen unwanted data from the binary information.

Channel On/Off

On/Off Function

The On/Off function is used simply to determine whether details of a channel are available when browsing channel data at the operating panel, or when outputting data using '*' (ECL command: ONOFF). The function of the channel is not influenced in any other way.

Energy Calculation

The meter pulses filtered in this way are integrated into a temporary meter. Every 1 2 seconds, the meter values are converted in rotation into the corresponding energy values, and added to the total energy registers. The pulses are converted to energy values using the following formula:

Energy Calculation

$$\text{Energie [kWh]} = \frac{\text{Impulse [Imp]}}{\text{Zählerkonstante } \left[\frac{\text{Imp}}{\text{kWh}}\right]} \cdot \text{Uratio} \cdot \text{Iratio}$$

$$\text{Uratio} = \frac{\text{Uprimär}}{\text{Usekundär}} \quad , \quad \text{Iratio} = \frac{\text{Iprimär}}{\text{Isekundär}}$$

Calculation of Power

The 'instantaneous power' PMOM is calculated from the time difference between incoming pulses. All other power data is calculated from the relevant energy data, on the basis of the particular time period.

The default calculation of power is based on an energy unit per hour (kWh). In the case of other units of measurement (eg. litres/minute), the P factor must be adjusted accordingly (ECL command: PFACTOR). The default value is 3600 for 'litres/minute' it would be 60.

The formula used to calculate power from energy E and period dt:

P = E * P factor / dt

Power Calculation

Channel Numbers

Channel Indication

- ❑ Physical channels: **1 .. 24**
- ❑ Virtual channels: **25 .. 32** oder **V1 .. V8**

Channel Parameters

Parameters	*ECL Command*
❑ Meter constants:	MCONST
❑ U ratio, I ratio:	URAT / IRAT
❑ P factor:	PFACTOR
❑ Rebound time:	PULSE
❑ Trigger edge:	EDGE
❑ Channel Start/Stop:	STARTSTOP
❑ Channel On/Off:	ONOFF

Floating point numbers

Precision

In order to obtain the highest precision, all internal calculations are performed on 64 bit floating point numbers. This provides 15 (!) decimal places.

Access to Channel Names throughout the System

Channel Names

A channel name, up to 8 characters long, can be assigned to each physical / virtual channel. This name is used to make identification of the channel easier when processing and showing results. The name may also be used to access a particular channel throughout the system:

```
<A> Etot Motor5
```

From station A: a system wide search is conducted for a channel with the name 'Motor5', and the total energy of the channel is displayed (cf also ECL command: **FINDER**).

Relay Outputs

4 programmable Relays

Four user programmable relays (change over contacts) allows messages to be sent or events to be performed if certain criteria are met. For example, a relay output may be activated if the average power over the synchronisation period exceeds a defined value. Please note the technical characteristics in the Annex.

One LED per relay output indicates the current relay status. If the LED is illuminated, the pin of the relay contact is touching the make contact, otherwise it is touching the break contact.

Status Relay

Status Relay

A fifth relay output (status relay) is used to generate operating status messages. While the electronic circuits are operational and no system errors are recognised, this output is activated and the green LED is lit. If the circuit are faulty, the relay drops out and the green LED is extinguished. A horn connected to the break contact of the status relay could then be used to signal the error status.

STATCHECK Function

Functional monitoring can be extended by appropriate programming (ECL command STATCHECK, cf also Online Help: ? STATCHECK). For example, output can be deactivated (error status) when the 24V supply is lost or if the lithium battery deteriorates to the extent that secure data storage is no longer possible (the battery test is only performed when the machine is switched on).

Access to Relay Names throughout the System

Relay Names

Each relay can be assigned a channel name up to 8 characters long. This name is used to make identification of the relay easier when processing and showing results. The name may also be used to access a particular channel throughout the system:

```
<A> REL Hupe=1
```

From station A: a system wide search is conducted for a relay with the name 'Hupe', and the relay is switched on
(cf also ECL command: **FINDER**).

Serial RS232 Interface

Communication between a host computer (PC), a terminal, a modem or a printer is performed by means of the RS232 interface. The physical interface is electrically separated from the ECS system.

Access to all System Data

A PC connected to the serial RS232 interface has full access to all the system readings. The parameter setting software **ECSoft**, which runs on a PC, manages the entire ECS data and allows it to be evaluated in tabular form.

Transmission Parameters of the RS232 Interface
(Default Values in Bold):
Transmission Rate: 1200, 2400, 4800, **9600** oder 19200 Baud
Parity: **None**, even or odd
Data bits: 8
Handshake: Xon/Xoff software handshake

DTE

Connector Configuration
The RS232 connector is configured as a DTE (Data Terminal Equipment) connector, with the signals present in a 9 pin subminiature D plug. This DTE configuration is the same as that normally found in PCs and terminals. The Annex contains details of the wiring of the connecting cable. In general, the following types of cable are required:

Connection of PC or terminal:
Null modem cable GTZ5232000R0001

Modem connection:
1:1 cable GTZ5233000R0001

RS232 printer connection:
Null modem cable GTZ5234000R0001

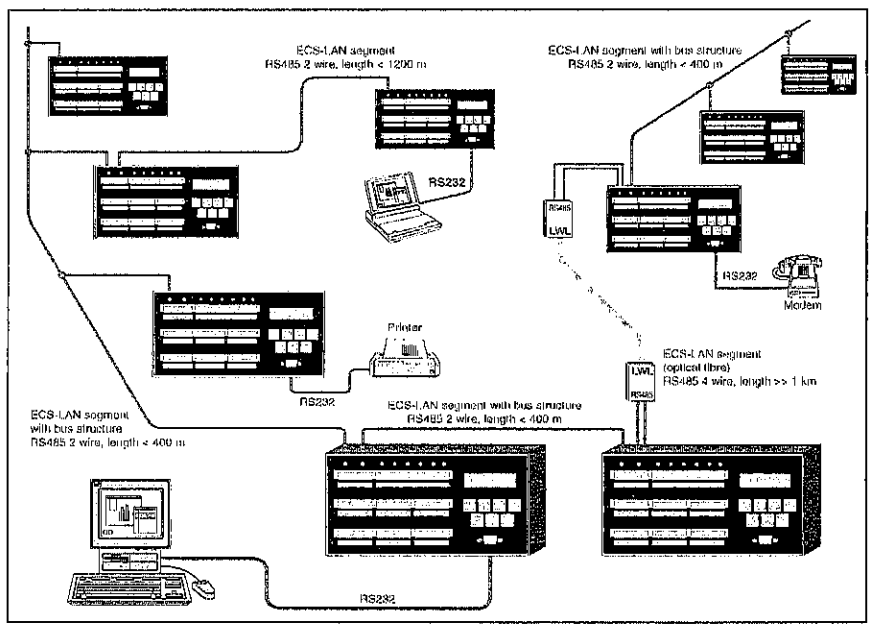
Pin assignment of Sub Min D9 connector

Pin#	Funktion:
1	DCD
2	RXD
3	TXD
4	DTR
5	Signal-Ground
6	DSR
7	RTS
8	CTS
9	RI

ECS-LAN

In the case of applications which require the use of more than 24 meters, a number of stations can be connected via a multimaster RS485 field bus (**ECS LAN**). Networking can be performed using two wire connections in bus or linear structure. The maximum length of a segment is 1200 m. Optical fibre transitions, using 4 wire technology, can be used to span distances of several kilometres between 2 summation stations.

Multimaster-system
The most significant advantage of the multimaster system is that each of the elements connected to the bus has unlimited access to the data and functionality of the whole system. A maximum of 255 summation stations can be connected to the ECS LAN. This means that a distance of around 300 km can be spanned without the need for additional signal amplifiers.



ECS-LAN Example Network

Two fully functional ECS LAN interfaces

Bus Left, Bus Right

Each summation station provides **2 fully functional ECS LAN interfaces**. They are identified with the terms **Bus Left** (BL) and **BUS Right** (BR). Each of the interfaces will work with 2 wire or 4 wire transmission methods (RS485).

2 Wire

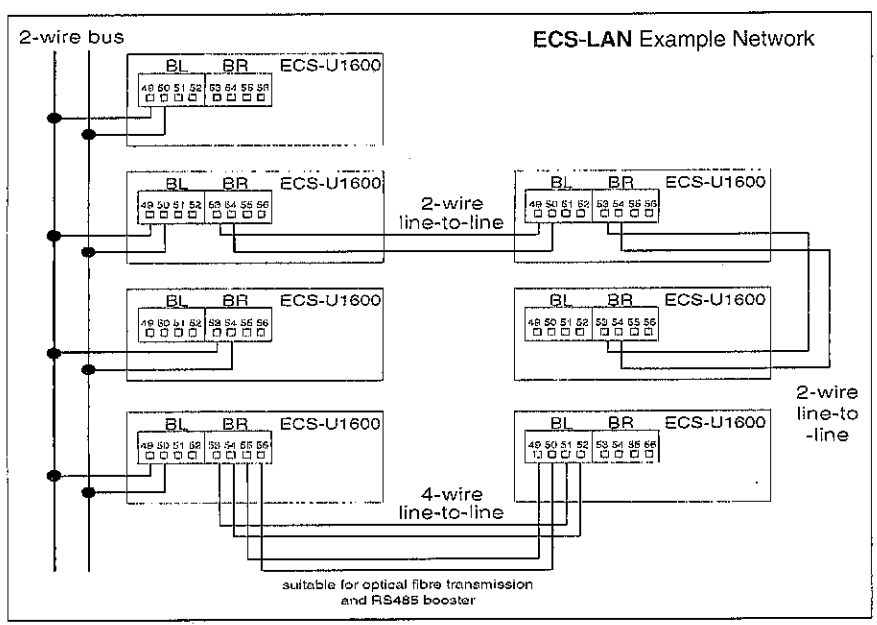
A bus system in which a number of summation stations are connected to a shared bus must use **2 wire** connections.

4 Wire

4 wire technology is used to span particularly long distances, or if a booster is used in the line; in this case only a line to line connection is possible.

Connection Length

If more than one station (up to 16) are physically located near to each other (total bus length up to 400m), it is recommended that the bus connection is performed using twisted pair. When the distance between two station exceeds 400m, however, the line to line connection should be performed using twisted pair 4 wire connections.



BL Contact	BR Contact	Function	2-wire	4-wire
49	53	EA+	Input and output '+'	Output '+'
50	54	EA-	Input and output '-'	Output '-'
51	55	E+	no function	Input '+'
52	56	E-	no function	Input '-'

Meaning of Contacts

General Notes

- ❑ The **network technologies** of the individual LAN segments can be chosen and mixed freely.
- ❑ The **transmission rate** determines the maximum line length, in accordance with RS485. The ECS LAN operates normally at a rate of 62.5 K baud; the maximum line length in this case is 1.2km.

62.5 K Baud

- ❑ The transmission line must be terminated at each end (but nowhere else) by a **terminating resistor**. The terminating resistor is integrated and turned on and off from the operating panel. The operation of the 2 wire connection can only be reliably secured by the use of the integrated terminating resistors. No external resistors may be connected!

Terminating Resistor

- ❑ The loop resistance of the 2 wire transmission line must not exceed 100 Ohm.
- ❑ Up to **16** stations can be connected to a bus segment. If the terminating resistor is connected properly (cf. overleaf), the stub line is kept to a minimum, and the total loop resistance of the transmission line is no more than 100 Ohm, then up to 32 stations may be connected to a segment.

16 Stations in a Bus Segment

- ❑ Statistics of the connected devices can be called up from the operating panel (device status).

Multimaster structure

Multimaster

Because the ECS LAN possesses a multimaster structure, it is not necessary to declare a station to be the bus master. Each summation station is completely independent, and has access to all the data of the other stations.

Device Directory

Directory of ECS LAN Devices

Each station automatically generates an internal directory (ECL command: DIR) of all devices connected to the ECS LAN (provided unique IDs are provided throughout the structure). Each station announces its presence throughout the system every 3 seconds by means of a broadcast message, allowing all stations which receive the message to update their internal directories accordingly. If the message does not arrive for more than 10 seconds, the station in question is removed from the internal directories.

Unique IDs

Station ID

Each station in the ECS LAN system must be allocated a unique ID. There are 255 different possible IDs.
An ID has the form: **A, A1 .. A9, B, B1 .. B9, .. , Z, Z1 .. Z4**

Eg.: From station A, you wish to call the total energy of channel 1 of station D1:

```
<A> D1: Etot 1
```

Access to Readings throughout the System

ECS LAN Status Display

Bus Status LED

The correct operation of the ECS LAN is indicated by two green LEDs, one for Bus Left and one for Bus Right:

- If no ECS LAN devices are connected, the LED is not lit.
- If one or more devices are connected to the bus segment concerned, the LED is illuminated.
- If two or more devices have the same ID, the corresponding bus LEDs of the devices with the same ID start flashing. Exception: If the station with the same ID is connected to the same LAN segment, no unambiguous error message is sent. On installing devices, therefore, always compare the number of stations with the total number as shown in the device statistics (operating panel: device status).
- If the internal terminating resistor is not turned on, the corresponding bus LED starts flashing.

Virtual Channels

Any Summations or Differences

Each summation station has eight virtual channels, the purpose of which is to generate totals or differences from the physical channels or other virtual channels. This is performed regardless of which summation unit in the ECS LAN system the channels are actually connected to!

- Virtual channels **V1 .. V8** (corresponding to channels 25 .. 32) are free to use all the registers with the exception of the maximum registers.

32 virtual Channels

- Unused physical channels may be used as virtual channels, in which case the maximum registers of the corresponding physical channel are available.
- The synchronising period readings list is able to use data from virtual channels.

Formation

There are two possibilities in defining a virtual channel:

Fixed Link with VSUM and VIRT

1. Fixed Link (ECL commands **VSUM** and **VIRT**)

All the registers of the source channels are added in the background, and the total is written into the appropriate registers of the virtual channel. The virtual channel is therefore fully dependent on the source channels, and the data from the virtual channel cannot be deleted or altered.

Uses: The meter reading of the virtual channels is designed to correspond to the sum or difference in meter readings from the source channels.

Differential Link with dVSUM and dVIRT

2. Differential Link (ECL commands **dVSUM** and **dVIRT**)

From the moment the channel is defined, the incoming quantums of energy (~ meter pulses) of the source channels are added, and the total quantums of energy are sent to the virtual channel, "as if they had actually just been measured". The virtual channel is disconnected from the source channel, and data can be altered as required.

Uses: Logical linking of input signals (as if the input signals had been connected together on a single meter channel).

Fixed Link Example

Example of Fixed Links with VSUM and VIRT:

Virtual channel V1 is occupied with the totals of physical channels 1 .. 8, weighted with 0.8, and the totals of physical channels 1 .. 12 at station **B1**, weighted with 0.2. The following background program (eg. H10) is installed:

```
<A> H10 = 'VSUM 1..8 0.8, B1: VSUM 1..12 0.2, VIRT V1='
```

Here, channel V1 of station D is occupied with a balancing of channels 9..16 with channel 17 (sum of the individual losses less the total meter):

```
<A>H 11 = 'VSUM 9..16, VSUM 17 -1, D: VIRT V5='
```

Cyclic Calculation of all Channel Registers

Since a large number of different energy data registers are available to each channel (31 registers excluding maximums: **Etot**, **EtotT1**, **EtotT2**, **Pmom**, 10+1 **EDay**, 12+1 **EMon**, 2+1 **EYear**), only the most important summations are performed at all times, with the less important being performed on a rotating basis. For example, **Etot** is provided constantly, while **EMon-12** is provided infrequently (approximately every 30 .. 90 seconds). Power values except **Pmom** are always calculated from energy values, and so they do not need to be included in the total.

All data registers of a fixed virtual channel always correspond to the summation of the channels used to create the virtual channel.

Programming

ECL - Energy Control Language

The flexibility of the ECS is based on the ease of programming the individual summation stations using a programming language specially developed for the ECS: **ECL - E**nergy **C**ontrol **L**anguage.
Cf. **Interpreter** Chapter for details of this high level language, which is similar to FORTH, but which can be learned easily, like BASIC.

The programming capability provides the following additional system characteristics:

Realisation of virtual Channels

Virtual Channels
Formulas, even complex ones, governing the creation of virtual channels can be produced quickly and easily in the form of background channels. No predetermined models exist to impede flexibility.

Programmable Relays

Programmable Relays
The change in state of a relay can be made dependent on a wide variety of conditions. ECL can be used to formula any required conditions, even where they relate to more than one station. For example, one type of relay programming could form the basis of a simple energy management system...

Simple Energy Management System
For example: If the average power of a consumer device exceeds a specified value, then the device can be switched off using one of the four relays. The decision to switch off can be made on the basis of the system time (eg. only at night), by querying the binary input (eg. switch only if a logical '1' is present at input 17), or else by evaluating the power values of other consumer devices.

Tariffs

Tariff Switching
Time comparison functions can be used in background programs to deal with highly specialised tariff switching requirements.

ASCII Database

Flexible Adaptation to Database Specific Transfer Formats
Repeated data queries can be stored as a normal P program. Full ASCII based database formats can be put together in this kind of program. Calling the program causes the data transfer to start according to the required format.

Cyclic Execution of Background Programs

Background Programs H0 .. H19
20 background programs are available. Each program can contain up to 127 characters. The background programs are run in the background in rotation. Thanks to the multitasking operating system, these background programs do not affect normal operation in any way.

Printer Output

H 19 has a special place. It is only run when "Print" is pressed on the operating panel. It is used to print information to a connected printer, but can also be used for other tasks if required.

Common Programs

Programs P0..P19
20 programs P0..P19 are provided to contain frequent command sequences, which can be called simply by calling the appropriate program. The programme can be linked, and a normal P program can be called by a background program as a sub routine.

Access by Names

A name can be assigned to each P program. This name may be used to call the corresponding P program throughout the system.

Synchronising Period Readings List

Period Energy Values are stored in a List

The energy values measured over a certain period can be stored in a readings list. The period in questions (duration: 10s .. 999h, default: 15 minutes) is either based on the system time, or else the boundary of the period is provided by the edge of a meter pulse (default: Channel 23). The energy measured during the period (**EINT**) is entered in the list together with the date and time. The average power of the period (**PINT**) is calculated on the basis of **EINT** and the length of the period in question.

Formatting

Format the List

The number of entries in the readings list depends on the number of channels. The readings list can therefore be formatted according to the required number of channels. The formatting can only be changed using the interpreter (ECL command: FORMAT).

The Number of Entries will be fixed

The **Format** instruction reserves space in system memory for the selected channels. This fixes the maximum number of entries, but not the duration of memory, which depends exclusively on the duration of the period.
For example: Only channels 1..4, channel 17 and virtual channel V1..V3 are to be included in the readings list:

```
<A> FORMAT=1..4+17+V1..V3
```

If the Format command is called without parameters, then the status information of the readings list is displayed. In particular, this shows the maximum number of entries which the readings list can accept.

```
<A> FORMAT = 1..4+17+V1..V3
<A> format
Format: 8 channels, 3276 entries(=34 days @period=
15min), 0 used
Channels: 1;2;3;4;17;25;26;27
```

Relationship between Duration of Memory and Number of Channels

No. of Channels	No. of Entries	Memory in Days with period = 15 mins
1	10922	113
4	5461	56
8	3276	34
9	2978	31
12	2340	24
16	1820	18
20	1489	15
24	1260	13
28	1092	11
32	963	10

Output of Readings List

The data stored in the readings list, together with date and time, can be output direct to the device using the operating panel, or else via the serial interface using the ECL command **EINT**.
This example shows the output of all readings of channels 1 to 5 in ASCII format:

ASCII Database Output

```
<A> Eint/## 1..5 * **
16.04.93;17:45:00;1;0.5;0.75;0.99;1.36
16.04.93;18:00:00;1.01;0.1;0.76;0.80;0.83
16.04.93;18:15:00;0.99;0.48;0.75;1.02;1.28
...
```

The ECL command **INDEX** can be used to read out data at a specified date. This example (in text) shows the value of the readings list of channel 1 on 16.04.93 at 18:15:

Access to one particular Record

```
<A> Index 16.04.93 18h15, Eint/ 1 .
16.04.93 18:15:00 : Eint-863 (01:Motor7) = 0.99 kWh
```

Structure of Readings List Memory

The readings list is a ring memory of a fixed size, which is appropriately formatted. Access is performed using an index number. Index 0 always points to the current period, index 1 to the last, and index 2 to the period before that, etc. The ECL command **INDEX** transforms a time into the corresponding index.

Rotating Memory

When memory space is used up, the oldest entry (with the highest index) is deleted to make space for the latest. The period which has just come to an end is given the index 1, and the indexes of the previous entries are incremented by one.

Value Range in the Readings List

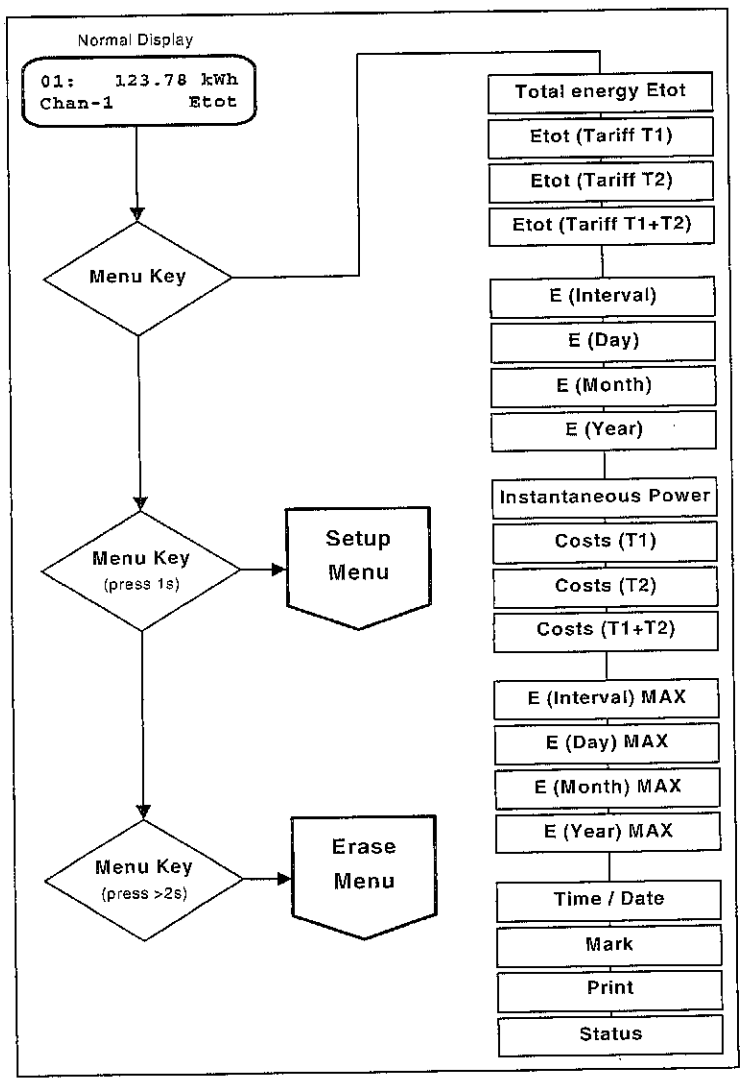
Limitation in Resolution because of Data Reduction

All data registers in the summation stations are 8 bytes wide internally (64 bit floating point). However, for the data list (from Eint 1), where the total duration of memory is dependent upon the amount of available memory, the data format has to be reduced to 2 bytes (with a corresponding loss of accuracy). As only energy values are stored (and not power values since these are calculated), the range is limited as follows:

Value range	Resolution
-0,8191 ... 0 ... +0,8191	Resolution: 0,0001
-81,91 ... -0,82 , +0,82 ... +81,91	Resolution: 0,01
-8191 ... -82 , +82 ... +8191	Resolution: 1,0
-819100 ... -8200 , +8200 ... +819100	Resolution: 100
-819100 > Range > +819100	N/A

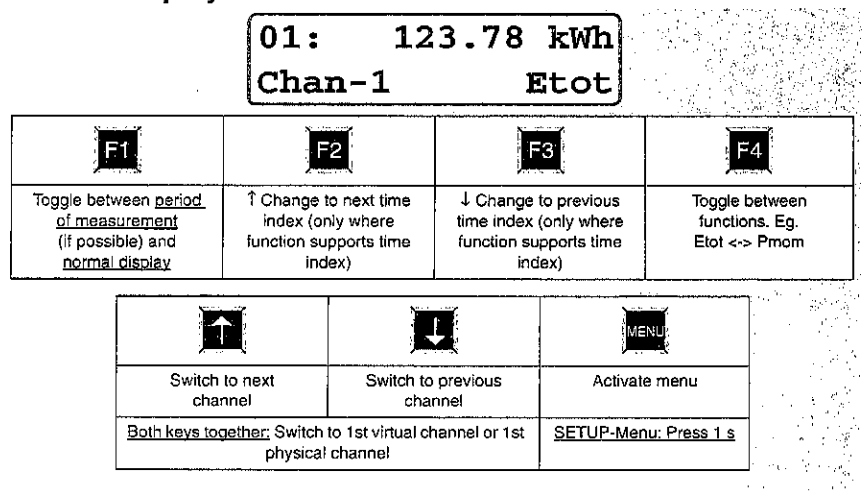
Device Panel

The integrated display / operating panel can be used at each location to obtain nearly all the information relating to the summation station, and to set operating parameters.



Menu Structure of the Operating Panel

Normal Display



When the device is switched on, or if no keyboard input is received for 45 seconds, the display shows the following information.
The active channel number is shown at the top left of the display. After the colon, there are 9 spaces for the reading to be displayed (the decimal point occupies one space).

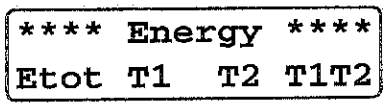
The position of the decimal point can be set separately for each channel (none, 1 .. 3 decimal places). If the reading cannot be contained in the display, the device first attempts to show fewer decimal places. If the reading is still too large, overflowing numbers are suppressed and displayed with leading zeros.
This operation does not adversely affect the internal calculations (64 bit floating point notation, 15 significant decimal places).
At the bottom left, the user programmable channel name is shown (max. 8 characters).
The currentlyactive function is shown at the bottom right (in this case Etot, total energy).

Note: There are two exceptions to the normal display described above: The time/date display and input status display. In cases where the power supply is interrupted, however, the display changes to the above state.

Etot	Ptot
EtotT1, EtotT2, EtotT1T2	KostT1, KostT2, KostT1T2
Other **Exxx**	Other **Pxxx**

Displaying the Period of Measurement
F1 toggles between the normal display and the period of measurement. With the period of measurement shown, F1 can be used to alternate time/date format (years are shown instead of seconds). Selecting the channel or index can be done during the display of the period of measurement. 45 seconds after the last key stroke, the display reverts to the normal display.

1st Menu: Energy Totals



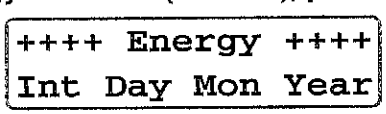
F1	F2	F3	F4
Etot Total energy	**EtotT1** Total energy for tariff 1	**EtotT2** Total energy for tariff 2	**EtotT1T2** Total energy for tariff 1+2

↑	↓	MENU
Cancel Menu Selection	Return to last menu item	Move to next menu
		SETUP-Menu: Press 1 s

The menu allows the user to select from 4 displays:

- ❑ **Etot** Total energy
- ❑ **EtotT1** Total energy tariff **T1**
- ❑ **EtotT2** Total energy tariff **T2**
- ❑ **EtotT1T2** Total energy tariff **T1** +**T2**

2nd Menu: Energy in Period (Interval), per Day, Month +Yearr



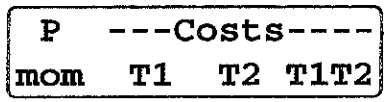
F1	F2	F3	F4
EInt Energy in period of measurement	**EDay** Energy per day	**EMon** Energy per month	**EYear** Energy per year

↑	↓	MENU
Cancel Menu Selection	Return to last menu item	Move to next menu
		SETUP-Menu: Press 1 s

The menu allows the user to select from 4 displays.
All four functions support a time index:

- ❑ **EInt:** Energy in period of measurement
- ❑ **EDay:** Energy per day
- ❑ **EMon:** Energy per month
- ❑ **EYear:** Energy per year

3rd Menu: Costs of Energy Totals, Instantaneous Power



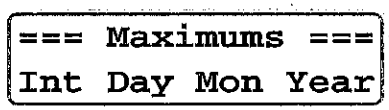
F1	F2	F3	F4
Pmom Instantaneous power	**CostT1** Costs of total energy for tariff 1	**CostT2** Costs of total energy for tariff 2	**CostT1T2** Costs of total energy for tariff 1+2

↑	↓	MENU
Cancel Menu Selection	Return to last menu item	Move to next menu
		SETUP-Menu: Press 1 s

The menu allows the user to select from 4 displays:

- ❑ **Pmom** Instantaneous power
- ❑ **CostT1** Costs of total energy for tariff 1
- ❑ **CostT2** Costs of total energy for tariff 2
- ❑ **CostT1T2** Costs of total energy for tariff 1+2

4th Menu: Energy Maximum in Period, per Day, Month +Year



F1	F2	F3	F4
EmaxInt Energy maximum in period of measurement	**EmaxDay** Energy maximum in period of measurement per day	**EmaxMon** Energy maximum in period of measurement per month	**EmaxYear** Energy maximum in period of measurement per year

↑	↓	MENU
Cancel Menu Selection	Return to last menu item	Move to next menu
		SETUP-Menu: Press 1 s

The menu allows the user to select from 4 displays.
All four functions support a time index:

- ❑ **EmaxInt** Energy maximum in period of measurement
- ❑ **EmaxDay** Energy maximum in period of measurement per day
- ❑ **EmaxMon** Energy maximum in period of measurement per month
- ❑ **EmaxYear** Energy maximum in period of measurement per year

5th Menu: Time Display Chart, Print and Status

The menu allows the user to select from 4 operations. The option "mark" is slightly different from the others, in that it is a simple on/off switch. The word "MARK" appears in uppercase letters if the function is activated (on). "mark" in lowercase letters, therefore, indicates that the function is switched off.
All four functions support a time index:

```
Time     Print
     mark  Status
```

F1	F2	F3	F4
Time display Time and date. Cancelled by pressing: ↑, ↓, MENU	**mark** On/off switch for mark-function (cf. below)	**Print** Activate printer output (using the output program defined in H 19)	**Status** Device status is shown in more than one display.

↑	↓	MENU
Cancel Menu Selection	Return to last menu item	Zum nächsten Menü gehen (wieder 1. Menü)
		SETUP-Menu: Press 1 s

Mark allows the disk of a classic Ferraris meter to be simulated. The position of the display cursor is determined by the spacing of the meter pulses. The cursor appears as an underline character in the upper part of the normal display.
Note: Depending on the spacing between meter pulses, it is sometimes inevitable that the chart may run off the display; therefore, the cursor may be invisible despite the function being activated.

Print activates printer output using the background program H19. H19 is started when F3 is pressed. If a serial printer (or parallel printer with RS232/Centronics interface) is connected to the RS232 interface, all the output from the H19 program is sent to this printer (important: H19 must be installed at the summation station to which the printer is connected). If printer output is not required, H19 may be used for other purposes.

The factory setting of background program H19 provides the following printouts:

- ❑ Printout of energy totals **Etot** for all activated channels and
- ❑ Printout of instantaneous power **Pmom** for all activated channels.

Status shows the device status in more than one displays. The MENU key moves on to the next status item, the ↓ key to the previous item, the ↑ key goes to the normal display. F1 .. F4 keys provide no function. The following 9 displays are provided:

1. **ID** and **Station Name**:

```
SummationStation
A:
```

2. Software Version and date:

```
ECS-U1600
V1.31m, 06.06.94
```

3. Current **Tariff** (T1 or T2) and **Tariff Source**, and **Period of Measurement** and **Interval Source**::

```
Tariff T1:Chan24
Interval : 15 m
```

4. **Status of the 24 meter inputs**, taking into account the preset pulse duration ('+' : input active, '_' : input not active, ':' : frequency >2.5Hz applied):

```
13____ ____ ____
01____ ____ ____
```

5. **Status of relays** ('+': ON, '_': OFF, 'p': OFF by program, 'P': ON by program):

```
 Relay Status
1:p 2:P 3:_ 4:+
```

6. **RS232** COM1 port settings:

```
COM 1:   9600 Bd
Pari:off HS: Xon
```

7. **ECS-LAN** BL (bus left) and BR (bus right) settings ('**2W+**': 2-wire technology, terminating resistor activated, '**2W**': .... without terminating resistor, '**4W**': 4-wire technology)

```
BL: ECS-LAN  BR:
62K5 2W+62K5 2W+
```

8. ECS-LAN **device statistics** (total number of connected devices, number of BLs (with no. of immediate neighbours) and number of BRs (with no. of immediate neighbours)

```
BL total=003  BR
002(01)  000(00)
```

9. **Status** of **24V** meter supply and **lithium** memory backup battery:

```
24Volt:       OK
Lithium bat.: OK
```

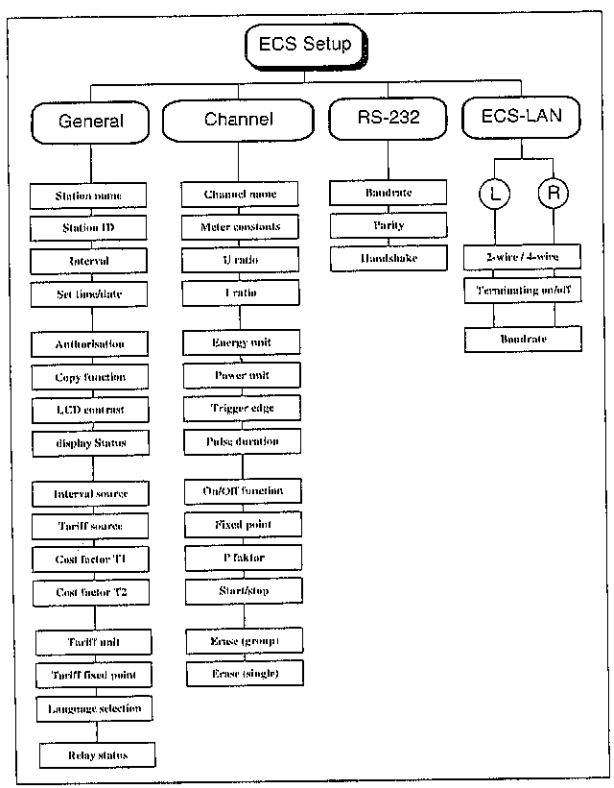
Setup Menu

Keeping the MENU key pressed (for around 1 second) displays the Setup Menu list:

```
---- SETUP -----
Gen Chan Ser Bus
```

F1	F2	F3	F4
Gen General station parameters	**Chan** Channel specific parameters	**Ser** Serial interface COM1 settings	**Bus** Adjust bus parameters (ECS LAN)

↑	↓	MENU
Cancel Menu Selection		
		Press 1.5 seconds: Activate ERASE



Gen. Setup: Station Name, ID, Synchronising Interval, Time

```
Name      Interval
ID        Time
```

F1	F2	F3	F4
Name Enter station name	**ID** Enter station ID	**Interval** Enter synchronising period	**Time** Set time and date

↑	↓	MENU
Back to normal display	Return to last menu item	Move to next menu
		SETUP-Menu: Press 1s

The menu allows the user to select from 4 displays:

- ❑ **Name** Enter station name
- ❑ **ID** Enter station ID
- ❑ **Interval** Enter synchronising period
- ❑ **Time** Set time and date

Station Name

```
St.Name:U1600
     <<   >> Enter
```

F1	F2	F3	F4
	<< Moves the cursor (underline character) one character to the left	**>>** Moves the cursor (underline character) one character to the right	**Enter** Confirms input (return)

↑	↓	MENU
Move through alphabet, figures and special characters	Move through alphabet, figures and special characters	**Cancel** (back to last menu)
Fast shift to character groups _ 0 A a		SETUP-Menu: Press 1s

- ❑ The maximum string length is 8 characters. If a space appears in the string, then the characters from the space are truncated after the enter key is pressed (**F4**). The station name must be at least one character long, otherwise the station name automatically becomes the '-' character.
- ❑ The following characters can be used: **_ + - 0 .. 9 A .. Z a .. z**
- ❑ ECL command: **STATION**

ID

```
Station ID   :A
     <<  >> Enter
```

F1	F2	F3	F4
	<< Moves the cursor (underline character) one character to the left	>> Moves the cursor (underline character) one character to the right	**Enter** Confirms input (return)

↑	↓	MENU
Move through alphabet, figures and special characters	Move through alphabet, figures and special characters	**Cancel** (back to last menu)
Fast shift to character groups _ 0 A a		SETUP-Menu: Press 1s

- ❑ The maximum string length is 2 characters. If a space appears in the string, then the characters from the space are truncated after the enter key is pressed (**F4**).
- ❑ An ID has the following form: **A, A1 .. A9, B, B1 .. B9 ... Z, Z1 .. Z4.**
- ❑ ECL command: **SETID**

Unique IDs must be given within a single ECS LAN; no two devices must have the same IDs. A maximum of 255 IDs are possible.

Unique IDs must be given within a single ECS LAN; no two devices must have the same IDs. A maximum of 255 IDs are possible.

Synchronising-Period

```
Interval [m]:_15
hms <<   >> Enter
```

F1	F2	F3	F4
hms Toggles between input in seconds, minutes and hours	<< Moves the cursor (underline character) one character to the left	>> Moves the cursor (underline character) one character to the right	**Enter** Confirms input (return)

↑	↓	MENU
Move through figures	Move through figures	**Cancel** (back to last menu)
Reset to zero		SETUP-Menu: Press 1s

- ❑ Input range for synchronising period: 10 seconds .. 999 hours.
- ❑ ECL command: **INTERVAL** (ITV)

Set Time and Date

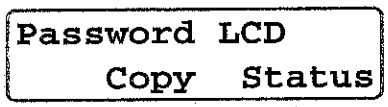
```
12:34:56  Time
10.07.92 <<  >>
```

F1	F2	F3	F4
		<< Moves the cursor (underline character) one character to the left	>> Moves the cursor (underline character) one character to the right

↑	↓	MENU
Move through figures	Move through figures	**Cancel** (back to last menu)
		SETUP-Menu: Press 1s

- ❑ After setting the time in the first line the cursor jumps to the date line. The word "Time" is then replaced with "Date".
- ❑ Any input has an immediate effect on the built in real time clock, which means that it is not possible to cancel the entry.
- ❑ ECL command: **TIME / DATE**

General Setup: Password, Copy, LCD Contrast, Status



F1	F2	F3	F4
Password Enter access authorisation	**Copy** Copy channel parameters	**LCD** Adjust contrast of LCD display	**Status** The device status is shown in various displays

↑	↓	MENU
Back to normal display	Return to last menu item	Move to next menu
		SETUP-Menu: Press 1s

- The device status is displayed in exactly the same way as the equivalent status function in the 5th menu, and is not therefore described in detail here.

Password (Access Authorisation)

Pressing the **F1** key in the second general setup menu displays the sub menu for setting the password..

The password consists of a 6 figure number, made up only of numbers between 1 and 4 (corresponding to **F1** .. **F4**). The special password "**111111**" removes password protection, ie. the password is not requested during normal operation.
If a password is defined as different to "**111111**", the existing password cannot be altered without first entering it.

The new password is then entered. The password is repeated to confirm.

Make sure you make a note of the password and keep it in a safe place. If password protection is activated it can only be removed if the current password is entered!

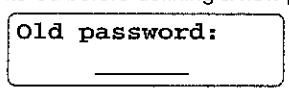
Method Used in Access Authorisation:
Access authorisation is activated when a password other than "**111111**" is defined. A timer is started once the correct password is given after a request, or immediately after defining a password. If NO actions take place at the instrument panel for more than 15 minutes (no keys pressed), or if the power supply is interrupted, the system recognises that the password must be requested from that point. If a key is pressed within those 15 minutes, the timer is reset to zero. If less than 15 minutes shows on the timer, the password is not requested.
If authorisation is required, the password is requested if:

- the setup menu is requested
- the Erase function is activated.

Note !
No more than 3 incorrect passwords may be entered. After the fourth incorrect attempt, the device cannot be used via the operating panel for a certain period.

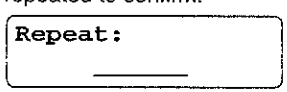
Checking the password:

The old password must be entered before defining a new password::



Input request for new password:

Input of the new password is repeated to confirm:

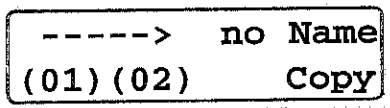


F1	F2	F3	F4
1	2	3	4

↑	↓	MENU
		Cancel (back to last menu)
		SETUP-Menu: Press 1s

When a number is entered, an asterisk '*' appears in the corresponding position. The **MENU** key can be used throughout to cancel an incorrect entry.

Copy

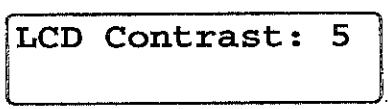


F1	F2	F3	F4
Source channel selection. Each press increments the channel number.	**Destination channel selection.** Each press increments the channel number.		**Copy** Copies channel parameters from source --> destination.

↑	↓	MENU
Back to normal display	Return to last menu item	**Cancel** (back to last menu)
		SETUP-Menu: Press 1s

All channel parameters (meter constants, U ratio, I ratio, E unit, P unit, P-factor, edge, pulse, OnOff, StartStop and channel fixed point) except the channel name are copied from a source channel to a destination channel.
At the bottom left, the source and destination channels are entered next to each other, with the arrow above showing the direction of the copy. "no name" indicates that the channel name is NOT copied.
Pressing **F4** copies the channel parameters, and a short message appears in the display. Because the source and destination channels cannot be the same, the destination channel is automatically incremented if they are the same.

LCD Contrast



F1	F2	F3	F4

↑	↓	MENU
Increment Contrast level	**Decrement** Contrast level	**Cancel** (back to last menu)
		SETUP-Menu: Press 1s

The menu is used to adjust the contrast of the LCD display. 8 levels are possible (0 .. 7). You should adjust the contrast to make the display as clear as possible. The default value is **5** and gives good results in most circumstances.

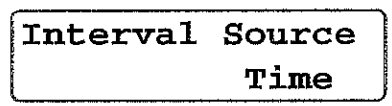
General Setup: Period and Tariff Source, Cost Factors

F1	F2	F3	F4
Interval Source Select source of synchronising period generation	**Tariff Source** Select source of tariff transition	**Cost Factor T1** Input cost factor for tariff 1	**Cost Factor T2** Input cost factor for tariff 2

↑	↓	MENU
Back to normal display	Return to last menu item	Move to next menu
		SETUP-Menu: Press 1s

- **Interval Source** Select source of synchronising period generation
- **Tariff Source** Select source of tariff transition
- **Cost Factor T1** Input cost factor for tariff 1
- **Cost Factor T2** Input cost factor for tariff 2

Interval Source



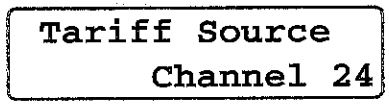
F1	F2	F3	F4
(+) increments channel number	(-) decrements channel number	Next possible source	Next possible source

↑	↓	MENU
Back to normal display	Return to last menu item	**Cancel** (back to last menu)
		SETUP-Menu: Press 1s

The synchronising period may be determined in one of the following three ways:
1. Time: periods are generated on the basis of the preset period duration.
2. Program: a period is only generated using the ECL command **SYNC=**
3. Channel 23: A meter input works as a binary input for the synchronising signal. The preset pulse duration and edge are noted. The F1 and F2 keys can be used to set the required channel number.
The interval source is selected by pressing **F3** or **F4**.

ECL command: **INTERVALSOURCE** (IQ)

Tariff Source



F1	F2	F3	F4
(+) increments channel number	(-) decrements channel number	Next possible source	Next possible source

↑	↓	MENU
Back to normal display	Return to last menu item	**Cancel** (back to last menu)
		SETUP-Menu: Press 1s

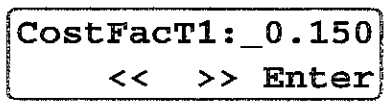
The current tariff T1 or T2 is selected in one of the following three ways:

- ❑ **1. Channel 24:** A meter input works as a binary input for determining the current tariff (the channel number can be set using **F1** and **F2**). The preset pulse duration is taken into account. The edge parameter is used to determine the signal level to which tariffs **T1** and **T2** are allocated. With the edge set to '+' (1), logical '0' (0 volt) corresponds to Tariff **T1**, while logical '1' (24 volt) corresponds to tariff **T2**. With the edge set to '-' (0), the situation is reversed.
- ❑ **2. Program:** The current tariff is determined using the ECL commands **Tariff=1** or **Tariff=2**. These allocations can only be used if the source is set to "Program".

The tariff source is selected by pressing **F3** or **F4**.

ECL command: **TARIFFSOURCE** (TQ)

Cost Factors for Tariffs T1 and T2



F1	F2	F3	F4
	<< Moves the cursor (underline character) one character to the left	>> Moves the cursor (underline character) one character to the right	**Enter** Confirms input (return)

↑	↓	MENU
Move through figures	Move through figures	**Cancel** (back to last menu)
Reset to zero		SETUP-Menu: Press 1s

The procedures for entering the cost factors for tariffs 1 and 2 are identical. The following describes the entry of cost factors for tariff 1.
The cost factor is used to convert energy data into costs. The conversion can be performed on the energy totals registers for the tariffs: **EtotT1**, **EtotT2**, **EtotT1T2**.

- ❑ Cost factor input range: 0.000 .. 99.999
- ❑ ECL command: **COSTFAC1**, **COSTFAC2**

General Setup: Tariff Unit, Tariff Fixed Point and Language

F1	F2	F3	F4
Tariff Unit Enter tariff unit for cost display.	**Tariff Fixed Point** Enter number of decimal places for cost display.		**Language** Enter Language Selection menu.

↑	↓	MENU
Back to normal display	Return to last menu item	Move to next menu
		SETUP-Menu: Press 1s

- ❑ **Tariff Unit** Enter tariff unit for cost display.
- ❑ **Tariff Fixed Point** Enter number of decimal places for cost display.
- ❑ **Language** Enter Language Selection menu.

Tariff Unit

```
Tariff Unit:DM
     <<   >> Enter
```

F1	F2	F3	F4
	<< Moves the cursor (underline character) one character to the left	>> Moves the cursor (underline character) one character to the right	**Enter** Confirms input (return)

↑	↓	MENU
Move through alphabet, figures and special characters	Move through alphabet, figures and special characters	**Cancel** (back to last menu)
Fast shift to character groups _ 0 A a		SETUP-Menu: Press 1s

- The maximum length of the string is 4 characters. If a space appears in the string, then the characters from the space are truncated after the enter key is pressed (**F4**). The tariff unit must be at least one character long, otherwise the tariff unit automatically becomes the '-' character.
- The following characters can be used: **_ + - 0 .. 9 A .. Z a .. z**
- ECL command: **TUNIT**

Tariff Fixed Point

```
TarifFIX=2: 0.00
```

F1	F2	F3	F4

↑	↓	MENU
Increment fixed point number	**Decrement** fixed point number	**Cancel** (back to last menu)
		SETUP-Menu: Press 1s

The tariff fixed point describes the number of decimal places used for displaying costs.

0: no decimal places "0"
1: one decimal place "0.0"
2: two decimal places "0.00"
3: three decimal places "0.000"

- ECL-Befehl: **TFIX**

Language Selection

```
2: English
****mm/dd/yy****
```

F1	F2	F3	F4
	date format selection: **dd.mm.yy** or **mm/dd/yy** or **mm-dd-yy**		

↑	↓	MENU
Increment language ID	**Decrement** language ID	**Cancel** (back to last menu)
		SETUP-Menu: Press 1s

All **menus**, **ECL messages** and **online helptexts** are displayed in the choosen language (devices with Multi Language Option only).

ECL command	LANGUAGE	DATEFORMAT
Select English	LANGUAGE = English	DATEFORMAT=mm/dd/yy
Select German	LANGUAGE = German	DATEFORMAT =dd.mm.yy
Query the setting	LANGUAGE	DATEFORMAT

General Setup: Relay Status

```
 Relay Status
1:p 2:_ 3:+ 4:P
```

F1	F2	F3	F4
Relay 1 Set status	**Relay 2** Set status	**Relay 3** Set status	**Relay 4** Set status

↑	↓	MENU
Back to normal display	Return to last menu item	Move to next menu (back to 1st general setup menu)
		SETUP-Menu: Press 1s

The status of each of the four relays is set as follows:

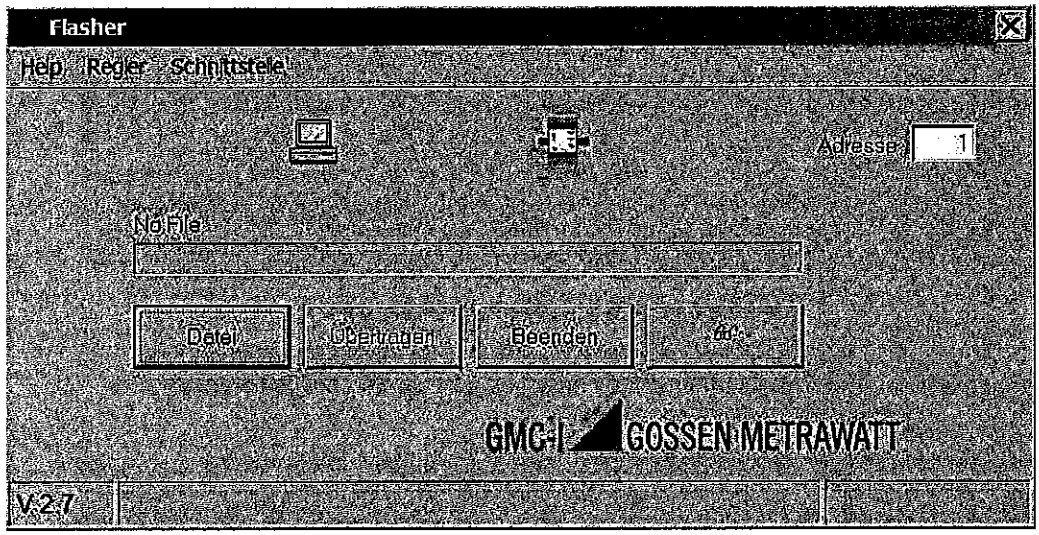
- "**p**": The relay can be switched on and off from the program [ECL interpreter] (Eg. REL 1=1). "**P**" indicates that the relay is on, "**p**" that it is off.
- "**+**": The relay is switched **on** permanently.
- "**_**": The relay is switched **off** permanently.

Anleitung zur Flash-Programmierung der Regelbaugruppe R355

Bitte lesen Sie vor Beginn der Flash-Programmierung aufmerksam die Hinweise unter „Help"

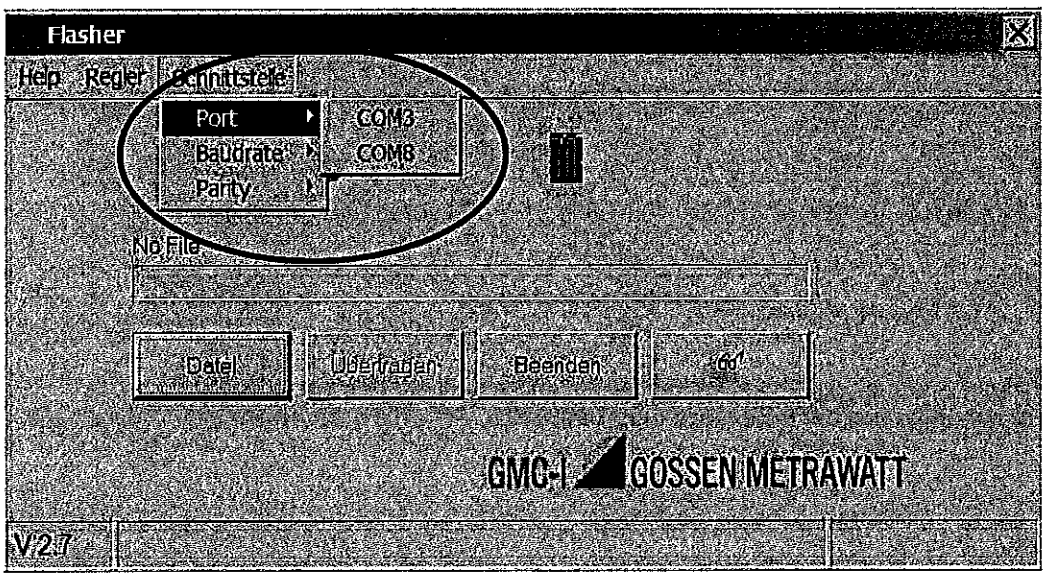
1. Flasher installieren (Icon auf Desktop) und MHX.File in ein Verzeichnis Ihrer Wahl kopieren

2. Flasher öffnen mit Doppelklick auf das Flasher Symbol

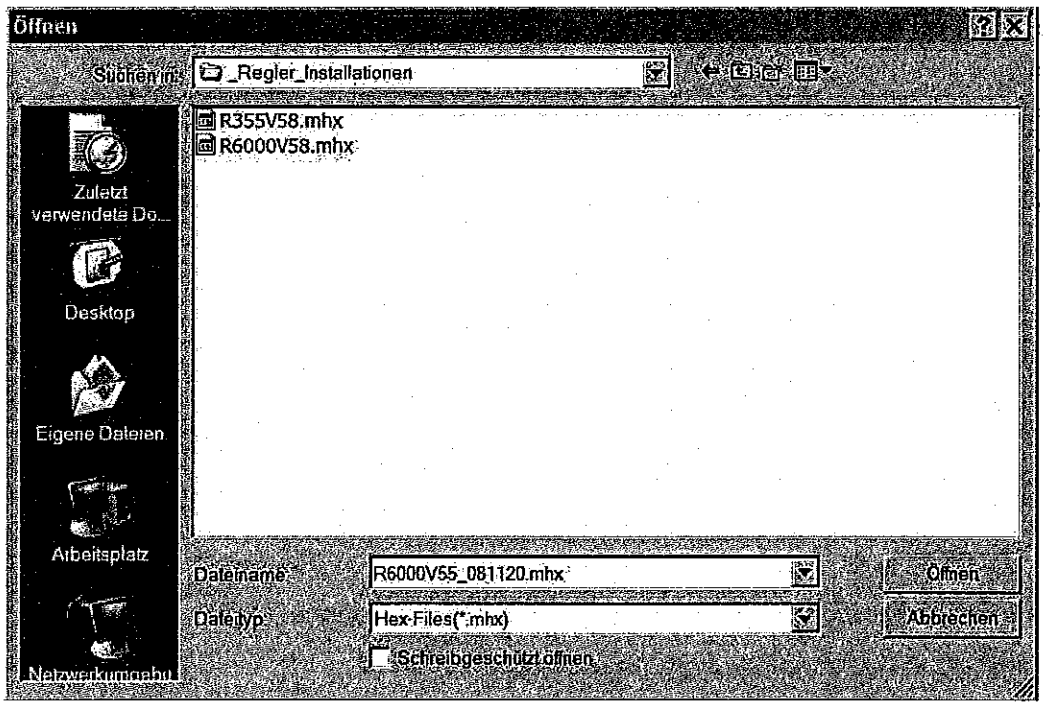


3. Konfiguration der Schnittstelle

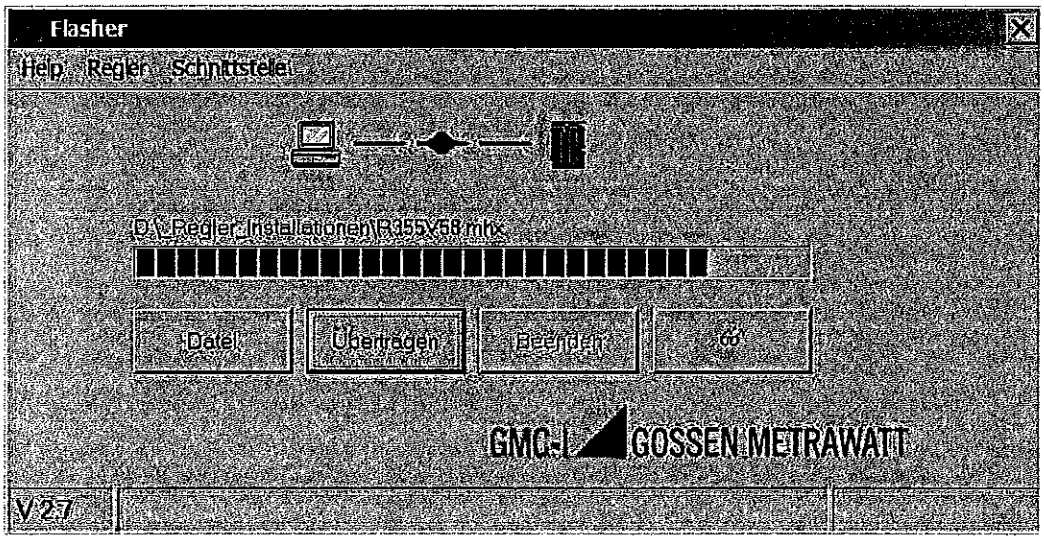
   Bemerkung: Vorhandene (auch virtuelle) Ports werden automatisch angezeigt und der richtige Anschluss ist auszuwählen.



6. Mit Button „Datei" das MHX.File aus dem vorgegebenen Verzeichnis laden (Öffnen)



7. Daten übertragen



Channel Setup: Chan. Name, Meter Constants, U ratio, I ratio

```
Chan-Name Urat
   MConst   Irat
```

F1	F2	F3	F4
Channel name Enter channel name	**MConst** Enter meter constants	**Urat** Enter U-ratio factor	**Irat** Enter I-ratio factor

↑	↓	MENU
Back to normal display	Return to last menu item	Move to next menu
		SETUP-Menu: Press 1s

❑ Pressing the **F2** key in the setup menu list displays the 1st channel specific setup menu, named channel setup menu:

Channel Names

```
Ch-Name:Chan-1
(01) << >> Enter
```

F1	F2	F3	F4
Select Channel The channel number is increased each time the key is pressed.	**<<** Moves the cursor (underline character) one character to the left	**>>** Moves the cursor (underline character) one character to the right	**Enter** Confirms input (return)

↑	↓	MENU
Move through alphabet, figures and special characters	Move through alphabet, figures and special characters	**Cancel** (back to last menu)
Fast shift to character groups _ **0 A a**		SETUP-Menu: Press 1s

A name may be assigned to each physical and virtual channel to make identification easier. The name does not have to be unique unless it is to be used as a field name in a database.

❑ The maximum string length is 8 characters. If a space appears in the string, then the characters from the space are truncated after the enter key is pressed (**F4**). The channel name must be at least one character long, otherwise the channel name automatically becomes the '-' character.

❑ The following characters can be used: **_ + - 0 .. 9 A .. Z a .. z**

❑ ECL command: **CHANNEL** (CHAN)

Meter Constants

```
MConst:__100.000
(01) << >> Enter
```

F1	F2	F3	F4
Select Channel The channel number is increased each time the key is pressed.	**<<** Moves the cursor (underline character) one character to the left	**>>** Moves the cursor (underline character) one character to the right	**Enter** Confirms input (return)

↑	↓	MENU
Move through alphabet, figures and special characters	Move through alphabet, figures and special characters	**Cancel** (back to last menu)
Reset to zero		SETUP-Menu: Press 1s

The meter constants may be defined separately for each physical channel (the formula used in the calculations is shown in the section System Description, Meter Inputs).

❑ **Changing the sign**: Press '<<' (**F2**) when the cursor is at the far left.

❑ Input range for meter constants: 0.001 .. 99999.999 [pulses/kWh].

❑ ECL command: **MCONST**

Formula Used for Calculation of Energy

$$\text{Energy [kWh]} = \frac{\text{Pulses [Puls]}}{\text{Meter constants } \left[\frac{\text{Puls}}{\text{kWh}}\right]} \cdot \text{U-ratio} \cdot \text{I-ratio}$$

$$\text{U-ratio} = \frac{\text{Primary voltage}}{\text{Secondary voltage}} \quad \text{I-ratio} = \frac{\text{Primary current}}{\text{Secondary current}}$$

U ratio and I ratio

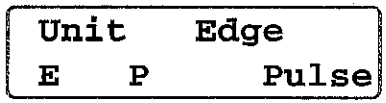
F1	F2	F3	F4
Select Channel The channel number is increased each time the key is pressed.	<< Moves the cursor (underline character) one character to the left	>> Moves the cursor (underline character) one character to the right	**Enter** Confirms input (return)

↑	↓	MENU
Move through figures	Move through figures	**Cancel** (back to last menu)
Reset to zero		SETUP-Menu: Press 1s

The factors for U ratio and I ratio can be defined separately for each physical channel (the formula used in the calculations are shown in the section System Description, Meter Inputs).

- **Changing the sign**: Press '<<' (**F2**) when the cursor is at the far left.
- Input range for U ratio and I ratio: 0.000 .. 99999.999
- A channel can be blocked to meter pulses by setting the U ratio or I ratio factor to zero (but better: use the Start/Stop function).
- ECL command: **URAT**, **IRAT**

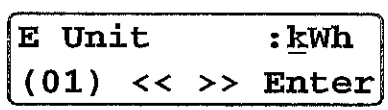
Channel Setup: Energy + Power Unit, Trigger Edge, Pulse Duration



F1	F2	F3	F4
Unit E Enter energy unit	**Unit P** Enter power unit	**Edge** Define trigger edge	**Pulse** Define pulse duration

↑	↓	MENU
Back to normal display	Return to last menu item	Move to next menu
		SETUP-Menu: Press 1s

Energy and Power Unit



F1	F2	F3	F4
Select Channel The channel number is increased each time the key is pressed.	<< Moves the cursor (underline character) one character to the left	>> Moves the cursor (underline character) one character to the right	**Enter** Confirms input (return)

↑	↓	MENU
Move through alphabet, figures and special characters	Move through alphabet, figures and special characters	**Cancel** (back to last menu)
Fast shift to character groups _ **0 A a**		SETUP-Menu: Press 1s

Any unit of energy (eg. kWh) and unit of power (eg. kW) may be assigned to each physical and virtual channel individually.

- The maximum string length is 4 characters. If a space appears in the string, then the characters from the space are truncated after the enter key is pressed (**F4**). The unit must be at least one character long, otherwise the unit automatically becomes the '-' character.
- The following characters can be used: **_ + - 0 .. 9 A .. Z a .. z**
- ECL command: **EUNIT**, **PUNIT**

Trigger Edge

F1	F2	F3	F4
Select Channel '+' The channel number is increased each time the key is pressed.	**Select Channel '-'** The channel number is decreased each time the key is pressed.		

↑	↓	MENU
Toggle between rising or falling edge	**Toggle** between rising or falling edge	**Cancel** (back to last menu)
		SETUP-Menu: Press 1s

Each channel may be defined separately to recognise a meter pulse either on a rising edge (labelled '+' or 1) or on a falling edge (labelled '-' or 0). The setting is shown symbolically in the display.If the input is used for tariff selection, the allocation of tariffs can be governed using the edge: If the edge is set to + (1), 0 volts at the input corresponds to tariff 1, 24V to tariff 2. If it is set to (0), the situation is reversed (0V corresponds to T2 ...).

- ❑ The input takes effect immediately, therefore does not need to be confirmed.
- ❑ ECL command: **EDGE**

Pulse Duration

F1	F2	F3	F4
Select Channel The channel number is increased each time the key is pressed.	**<<** Moves the cursor (underline character) one character to the left	**>>** Moves the cursor (underline character) one character to the right	**Enter** Confirms input (return)

↑	↓	MENU
Move through figures	Move through figures	**Cancel** (back to last menu)
Reset to zero		SETUP-Menu: Press 1s

The pulse duration ("rebound time") can be defined separately for each of the physical channels (the formula used in the calculations are shown in the section System Description, Meter Inputs).

- ❑ Input range for pulse duration: 10 ... 2550 milliseconds (0.01 .. 2.55 s)
- ❑ ECL command: **PULSE**

Channel Setup: On/Off, Fixed Point, P Factor, Start/Stop

F1	F2	F3	F4
OnOff On/off function of channel	**Fix** Number of decimal places for figure display	**PFact** Enter P factor	**StaSto** Start/stop function of channel

↑	↓	MENU
Back to normal display	Return to last menu item	Move to next menu
		SETUP-Menu: Press 1s

- ❑ **OnOff** On/off function of channel
- ❑ **Fix** Number of decimal places for figure display
- ❑ **PFact** Enter P factor
- ❑ **StaSto** Start/stop function of channel

On/Off Function

```
ChannelFunct: ON
(01) -
```

F1	F2	F3	F4
Select Channel '+' The channel number is increased each time the key is pressed.	**Select Channel '-'** The channel number is decreased each time the key is pressed.		

↑	↓	MENU
Toggle between ON and OFF	**Toggle** between ON and OFF	**Cancel** (back to last menu)
		SETUP-Menu: Press 1s

If all channels are switched off, the normal display shows time and date.
Each channel can be adjusted separately with regard to whether or not the channel is made visible in the display during queries from the operating panel, or during '*' enumerations in interpreter mode. The operation of the channel is not affected! For example, if channels 1 .. 3 are switched on, only values from these 3 channels can be displayed the device presents itself as having 3 channels.'**Etot ***', called from interpreter mode, only displays the energy totals from these three channels.

- ❑ The input takes effect immediately, therefore does not need to be confirmed.
- ❑ ECL command: **ONOFF**

Channel Fixed Point

```
DeciFIX=2:  0.00
(01) -
```

F1	F2	F3	F4
Select Channel '+' The channel number is increased each time the key is pressed.	**Select Channel '-'** The channel number is decreased each time the key is pressed.		

↑	↓	MENU
Increment fixed point number	Decrement fixed point number	**Cancel** (back to last menu)
		SETUP-Menu: Press 1s

The number of decimal places used in the display of energy or power can be defined separately for each physical or virtual channel.

0:	no decimal places	"0"	2:	two decimal places	"0.00"
1:	one decimal place	"0.0"	3:	three decimal places	"0.000"

- ❑ The input takes effect immediately, therefore does not need to be confirmed.
- ❑ ECL command: **CFIX**

P Factor

```
PFact :_3600.000
(01) << >> Enter
```

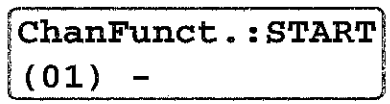
F1	F2	F3	F4
Select Channel The channel number is increased each time the key is pressed.	**<<** Moves the cursor (underline character) one character to the left	**>>** Moves the cursor (underline character) one character to the right	**Enter** Confirms input (return)

↑	↓	MENU
Move through figures	Move through figures	**Cancel** (back to last menu)
Reset to zero		SETUP-Menu: Press 1s

The P factor can be defined separately for each physical channel (You will find the calculation formula under: Sustem Description, Meter Inputs).

- ❑ **Changing the sign**: Press '<<' (**F2**) when the cursor is at the far left.
- ❑ Input range for the P factor: 0.001 .. 99999.999.
- ❑ ECL command: **PFACTOR**

Start/Stop Function



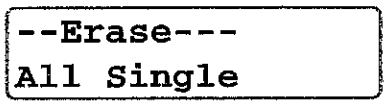
F1	F2	F3	F4
Select Channel '+' The channel number is increased each time the key is pressed.	**Select Channel '-'** The channel number is decreased each time the key is pressed.		

↑	↓	MENU
Toggle between START and STOP	**Toggle** between START and STOP	**Cancel** (back to last menu)
		SETUP-Menu: Press 1s

The Start/Stop function is used to control acceptance of the meter pulses at an input.

- A virtual channel created using a 'differential connection' can be influenced analogously using the Start/Stop function.
- Because the binary input status of a channel is not affected, the function can be used to screen unwanted data from the binary information.
- The input takes effect immediately, therefore does not need to be confirmed.
- ECL command: **STARTSTOP** (STSP)

Channel Setup: Erase All and Single



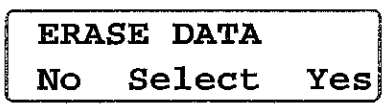
F1	F2	F3	F4
Erase All Calls erase menu, default = erase all	**Erase Single** Calls erase menu, default = erase single		

↑	↓	MENU
Back to normal display	Return to last menu item	Move to next menu
		SETUP-Menu: Press 1s

- **Erase All** Calls erase menu, default = erase all
- **Erase Single** Calls erase menu, default = erase single

Erase All

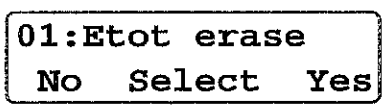
Pressing the **F1** function key (**Erase All**) in the 4th channel setup menu displays the following sub menu:



The erase menu described below is displayed with the default setting to erase all data from readings (except the list of period reading).

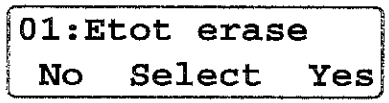
Erase Single

Pressing the F2 function key (Erase Single) in the 4th channel setup menu displays the following sub menu:



The erase menu described below is displayed with the default setting to erase a particular value (the readings function is the function which was last active, the channel number is selected using the arrow keys).

Erase Menu



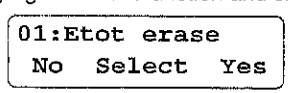
F1	F2	F3	F4
No Cancel	↑ **Select erase types**	↓ **Select erase types**	**Yes** Perform erase after asking for confirmation

↑	↓	MENU
Change to next channel	**Change** to previous channel	**Cancel** (back to last menu or normal display)
both keys together: Change to 1st physical channel or 1st virtual channel		SETUP-Menu: Press 1s

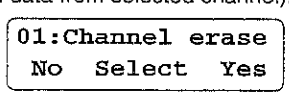
Keeping the **Menu** key pressed for an extended period (2 seconds) in the normal display causes the central erase menu to appear.
The readings functions (here: Etot) is the function which was last active, and the channel number is the number of the last used channel.

F2 and F3 select from 4 erase types:

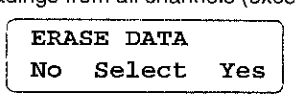
Display for single erase applying to current function and channel number:



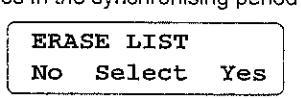
Display for channel erase (all data from selected channel):



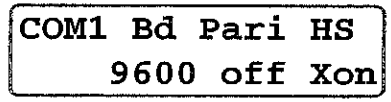
Display for data erase, all readings from all channels (except readings list) erased:



Display for list erase, all entries in the synchronising period readings list are erased:



RS232 Setup: Serial Interface COM1



F1	F2	F3	F4
	Transmission Rate 1200, 2400, 4800, **9600**, or 19200 baud	**Parity** Even (Evn), Odd or **OFF**	

↑	↓	MENU
Back to normal display	Return to last menu item	
		SETUP-Menu: Press 1s

Pressing the function keys **F2** .. **F3** changes the interface settings for COM1 shown above the keys.
The handshake (HS) is permanently set to Xon/Xoff software handshake.

- **Transmission Rate** 1200, 2400, 4800, **9600**, or 19200 baud
- **Parity** Even (Evn), Odd or **OFF**

Bus Setup: ECS LAN Settings

```
BL: ECS-LAN  BR:
2Wr Trm 2Wr Trm
```

F1	F2	F3	F4
Bus Left: 2Wr / 4Wr Select 2 wire or 4 wire connection	**Bus Left:** Trm / - In 2 wire operation, the internal resistor may be on or off.	**Bus Right:** 2Wr / 4Wr Select of 2 wire or 4 wire connection	**Bus Right:** Trm / - In 2 wire operation, the internal resistor may be on or off.

↑	↓	MENU
Back to normal display	Return to last menu item	change to menu **ECS-LAN Baudrate**
		SETUP-Menu: Press 1s

Pressing the function keys **F1 .. F4** changes the ECS LAN settings shown above the keys.

BL: Bus Left settings (contacts 49..52)
BR: Bus Right settings (contacts 53..56)

A 2 wire line (contacts 49+50 or 53+54) is used normally (this is the only possible way to allow a bus configuration with more than one device on the same bus line). The integrated terminating resistors (Trm) of the first and last device on the bus line must also be switched on. The bus will not work properly without terminating resistors (Bus LED flashes).
In the case of longer transmission distances, or if boosters need to be used, 4 wire technology can be used (only line to line connections are possible). The necessary terminating resistors are activated automatically (selecting Trm is not required).

Bus Setup: ECS-LAN Baudrate

```
BL: Baudrate BR:
62K5     62K5
```

F1	F2	F3	F4
Bus Left: Baudrate 7K8, 15K6, 31K2, **62K5** or 125K baud		**Bus Right: Baudrate** 7K8, 15K6, 31K2, **62K5** or 125K baud	

↑	↓	MENU
Back to normal display	Return to last menu item	
		SETUP-Menu: Press 1s

The standard baudrate is 62,5 K baud and should be changed in special cases only.

- ❑ **Bus Left:** **Baudrate** 7K8, 15K6, 31K2, **62K5** or 125K baud
- ❑ **Bus Right:** **Baudrate** 7K8, 15K6, 31K2, **62K5** or 125K baud

Master Reset

Whenever the power supply is interrupted or the internal watchdog is invoked, a normal reset is carried out, which means that all readings and parameters are erased.

If, in other circumstances, all the readings and parameters are required to be reset to their factory state, a **Master Reset** can be carried out via the U1600 input keys.

The following three keys are pressed simultaneously for at least 5 seconds:

F1 + F4 + MENU

The master reset is performed, and the device returns to the normal display.
Note: The master reset only operates within 60 seconds of turning on the summation station.

The following are not affected by the master reset:

- the ID
- the baud rate of the RS232 interface
- the parity of the RS232 interface
- the BUS/L and BUS/R parameters
- the coupling of the Status-Relay (**STATCHECK**)
- the dialog language

ECL Interpreter

Introduction

The ECL command interpreter (ECL = **E**nergy **C**ontrol **L**anguage) is the logical interface between a summation station and the PC (host) or terminal. In physical terms, the signals are transmitted via a serial RS232 interface.

Data may be exchanged with additional summation stations connected on the ECS LAN such that the addressed stations present themselves as being connected directly to the PC or terminal.

Communication occurs by means of plain language commands, and the output format can be suited to any database or user specific requirements. The individual commands can be arranged in sequence, and their order of processing can be subject to certain conditions, which means that it represents a fully functional programming language. The programming language is called **ECL** - **E**nergy **C**ontrol **L**anguage.

ECL is a mixture of **FORTH** and **BASIC**. If you are familiar with reverse Polish notation (**RPN**) used in HP calculators and have some experience of programming in BASIC, you will be able to learn the language without difficulty.

You are perhaps wondering why the ECS requires a high level language at all? Firstly, the summation stations make use of virtual channels, and a clear notation is required to define them (especially where the application relates to a number of devices), and secondly, effective programming of relay output and other procedures can only be carried out with a well developed programming language. Imagine you are declaring summation station 'B':

Relay 1 in summation station '**B**' is turned on if the sum of instantaneous power values in channels 1 to 5 in device '**A**', plus channels 8 and 17 in device '**G5**' is greater than 125kW.

We write the following (logged into station B) :

<B> **A**:Pmom- 1..5, **G5**:Pmom- 8+17, +,125,>, IF,Rel 1=1, ELSE,Rel 1=0

Detailed Analysis of This Sequence of Commands:
As mentioned above, commands are placed in sequence, creating a sequence of commands. Unlike BASIC, however, the sequences do not bring about extra semantic problems (semantics: meaning), because the relationships between the individual commands is governed by a well defined **parameter stack**. The stack is arranged as a push down store, ie. the order in which elements are removed from the stack is opposite to that in which they were inserted.

Example: If we add elements 1, 5, 8 and 17 to the stack in that order, the first element removed is 17, followed by 8, then 5 and finally 1.

Each command pushes its results on to the stack, or pops elements from the stack. The addition operator '+', for example, pops 2 elements from the stack, adds them and pushes the result on to the stack. The output command '!' removes an element from the stack and "prints" it. So:

`2,5,+,!`	outputs '7' (sum of 2+5)
`17.5;-4;3;*;+;!`	outputs '5.5' (multiplication -4 * 3 = -12, then addition 17.5 + -12 = 5.5)

Within a command, the form used is well known from BASIC:

FunctionsName (Argument1, Argument2, ...)

We have made a few syntactic changes. The brackets around the argument, and the commas between them may (and must) be omitted.

- The separator used between the function name (ECL command) and the argument (parameter) and between two arguments is the space character.
- The comma or the semicolon is used as the separator between two commands.

ECL-Command Parameter1 Parameter2 ... = Assignment1 Assignment2 ...

The above example now seems more clear. The first command is as follows:

```
A:Pmom- 1..5
```

A: indicates to the command interpreter that the current command should be diverted to device **A** (in our example we assume that we are communicating with device **B**). The result (in this case the sum of the instantaneous power values of channel 1 to 5) is transmitted back to device **B** and pushed on to the stack.
The second command,

```
G5:Pmom- 8+17
```

pushes the sum of the instantaneous power values of channels 8 + 17 of device '**G5**' on to the stack.

```
+, 125, >
```

The third command '+' adds the two sums of instantaneous power values, and pushes the fourth command '**125**', simply the number 125, on to the stack.
The fifth command '>' compares the total with 125 (..Total > 125). If the comparison is true, ie, the total is greater than 125, a 1 is pushed on to the stack, otherwise a 0 (zero).

```
IF, Rel 1=1
```

The sixth command 'IF' then determines whether the 1st relay is switched on (**REL 1=1**) or

```
ELSE, Rel 1=0
```

off (**REL 1=0**).

Ranges, Numbers and Strings

The numerical data occurring in a summation unit can cover a very large range. More important than the size of the range of number, however, is the accuracy, which is here expressed in terms of significant decimal places. 15 significant decimal places are available, and the range of numbers which can be represented in the interpreter covers 27 places before the decimal point, and 9 places after it.
Note: If 15 places are not sufficient to represent a number, the system changes internally to exponential notation (64 bit floating point numbers).

For example, the following value can be processed without sacrificing accuracy:

1.234.567.890,12345 kWh

All operations provided by the interpreter work to the quoted accuracy.

It is necessary to introduce the data type **REAL**, even if the comparison with real numbers is somewhat clumsy. We should bear in mind that the interpreter recognises only this data type for numbers integers are regarded as a sub type of **REAL** (there is one exception: enumeration, eg. 1..4+7).

Note: The parameter stack only accepts elements of type **REAL**.

The following mathematical functions are provided:

Basic Arithmetic	+ - * / **MOD**
Boolean Operators	& | ^
Comparisons	< <= == != >= >
Conversions	**INT INTR FRAC ABS**
Square Root	**SQRT**
Trigonometric Functions (Basis = radian measure)	**SIN COS** **ASIN ACOS**
Exponential Functions	**EXP LOG** **

Strings

Strings are provided as well as numbers. Strings can be made up of letters, numbers and special characters in any order. The name of a channel is a string, and assignment is carried out as follows:

```
Channel 4=ROOM5b
```

The programs themselves are also strings. The sequence of commands used as an example above can be brought "to life" by turning it into background program H 10, constantly controlling the status of the relay:

```
<B> H 10= 'A:Pmom- 1..5, G5:Pmom- 8+17, +, 125, >, IF, Rel 1=1, ELSE, Rel 1=0'
```

Strings may also contain spaces or special characters such as comas. For this reason, strings must be enclosed in quotation marks if spaces or other syntactically relevant special characters occur in the string.

For example: The output function '!' prints the string passed to it as a parameter:

`! "the 'printout'!"`	the 'printout'!

Note: The string delimiters used must not occur within the string itself! If quotation marks are used as delimiters, then single inverted commas may be used within the string, but not quotation marks (and vice versa).

There is no stack for strings a buffer is used instead to contain the last string used. This makes it possible to copy an existing program into another:

```
A:P! 1,b:P 11=$
```

The program P1 in device A: is listed and copied into program P11 in device B:. The $ symbol is used to indicate that the string buffer is to be used.

Arguments, Extensions, Assignment and Error Messages

Each command may be called with up to 3 arguments, where appropriate. In ECL, arguments are also treated as parameters to an ECL command.

In addition, the assignment operator '=' allows extra arguments to be provided for an assignment operation. Command extensions are also possible. The extension is used to control the behaviour of the command.

The **argument type** depends on the command, and a number of types are possible for each argument. The following types are defined:

REAL	12 / 27.3 / -36.3E-2
ENUMERATION	2..7+V1..V7 / * / ** / #
.	Pops a real element from the parameter stack
STRING	"An 'Example' ..." / Chan-5
$	Uses the contents of the string buffer

The following notation is used for **enumeration**:

2..7	Channel 2 to channel 7
2+7	Channel 2 and channel 7
V1..V3	Virtual channels V1 (==channel 25) to V3 (==channel 27)
2..7+V1	Channels 2 to 7 and V1
1..8+17+20..V3	...
*	All channels switches **ON** (cf. On/Off Function)
**	All possible channels
#	All channels for which the readings list was formatted
##	who **

The **extension** influences the behaviour of most commands. Where appropriate, extensions can be combined without restriction. For details → ECL Command Reference.

-	Suppress output (where provided)
+	Concatenate output (no new lines)
.	Database output, separator ';', terminator <CR> <LF>
#	Output number only, no extra information. Terminator <CR> <LF>
##	As #, but with terminator ';'
/	Output with time details "to"
//	Time details "from" and "to"
^	Output with time details "to" with seconds count from 1.1.90
^^	Output with time details "from" and "to" with seconds count from 1.1.90
&	The ID (eg. A1:) is also printed at the start of the line
&&	The ID as a number (eg. 2:) is also printed at the start of the line
!	Forces output (eg. P!3 prints a listing of program P3)

Example using energy total Etot from channel 2 (channel name = heating room):

Etot 2	Etot (02: heatroom) = 21.31 kWh
Etot& 2	A:Etot (02: heatroom) = 21.31 kWh
Etot. 2	Etot;2;heatroom;21.30527;kWh;
Etot# 2	21.30527
Etot/ 2	15.08.92 23:10:11 : Etot (02) = 21.31 kWh
Etot#// 2	10.08.92;14:00:04;15.08.92;23:11:21;21.30527
Etot/## 1..4	15.08.92;23:11:21;0;21.30527;0;0
Etot^## 1..4	82768281;0;21.30527;0;0

The **assignment operator** indicates to a command that an assignment is required to be performed instead of output:

```
Etot 1=123.23 $
```

Here, the register of the total energy of channel 1 is assigned the value 123.23.

Example:An example should make the above information clear. We assume that we are communicating with device A:, that B:Channel 17 is called "Chan-17", and that EtotT1 2 is 222.22 kWh:

```
<A> b:Chan- 17, c:Chan V1..v4+V8=$, ! "Name <<" $ ">>, Value = ", EtotT1#+ 2, d:Etot 5..8=.
```

- Name of 17th channel of B; in string buffer, no output
- Assignment of $ (string buffer) to channel names V1 to V4 and V8
- Output: **Name <<Chan-17>>, Value = 222.22**
- Push EtotT1 of channel 2 on to the parameter stack
- Assign the topmost parameter stack element (=EtotT1 2) to the Etot channels 5 to 8 in D:

Note: As shown in this example, command names can be written in either uppercase or lowercase letters.

Error Messages

ECL provides plain text error messages to help locate errors. When an error occurs, the program execution is halted and an error message is displayed.

Errors in background programs are only displayed on request. The command used for this purpose is called "**ERR**", and outputs a list of error messages relating to all background programs.

Please bear in mind that **ECL** is an interpreted programming language. Program commands are only evaluated when they are next in turn. For example, if an error exists between **IF .. ELSE**, the associated error message may only occur when this section is actually executed, ie. when the IF condition is met (==1).

Interrupting Programs

The following key combination can be used to interrupt a sequence of commands in the ECL interpreter:

<STRG> + X

Background programs cannot be interrupted in this way. The command **HBREAK** must be entered instead. This causes the currently running background program to be interrupted and all processing to be halted for 16 seconds. After this, the cycle starts again with H 0.

The System Prompt and the Help System

After the <ENTER KEY> ↵ is pressed, the station responds with its prompt:

<A>

The prompt tells us that we are currently communicating with the summation station with the ID A:. The command or sequence of commands can be entered after the prompt. A maximum of 128 characters can be entered per line. The <ENTER KEY> signals the end of input, and processing is started. The prompt appears again as soon as processing is completed.

Logging On

Because of networking in the ECS LAN system, it is possible to log on to any device in the network. The interpreter then behaves as if the terminal were connected directly to the RS232 interface of the device in question. The prompt is the only way to determine which summation station is involved.
For example, to log on to device **B1**, type the following:

```
B1: : ↵
```

IIf device B1 is available, a new prompt appears: **<B1>**. From this point, all communication occurs direct with device B1, ie. all commands without ID details apply to device B1.

List of Available Commands - Help System

A list of available **ECL** interpreter commands can be displayed by entering the following:

HELP ↵ *or* **?** ↵

All commands are listed according to function groups. Keywords relating to general topics are also displayed. Further detailed help on the commands and keywords listed can be obtained by entering the following command:

HELP <search term> ↵ *or* **?** <search term> ↵
(Note the space required between **HELP** or **?** and <search term>)

For example: You require general information on the use of **parameters**. There is no need to enter all the words of a search term (provided it is unambiguous):

```
? Para
```

The following command provides a display of all help texts:

```
? BOOK
```

Using the PC parameter setting software **ECSoft 2**, the entire output can be diverted to a file or printer.

> The help system provides information on all commands of the **ECL** interpreter. The information provided always relates to the current version of the operating software.

Summary of ECL Command Groups

Arithmetic, Boolean Operators, and Stack Manipulation:
+ - * / & | ^ < <= > >= == != DUP DROP SWAP PICK

Conditional Branches and Loops:
IF IFF ELSE ENDIF FORI I NEXTI FORJ J NEXTJ ALL NEXTA EXIT RETURN

Mathematical Functions and Number Manipulation:
SQRT SIN COS ASIN ACOS DEG RAD EXP LOG ** ABS FRAC FIX INT INTR MAX MIN MOD

Energy Totals, Costs and Instantaneous Power:
Etot EtotT1 EtotT2 EtotT1T2 CostT1 CostT2 CostT1T2 Pmom

Energy in Period, per Day, Month, Year, maximums:
EInt EDay EMon EYear Emax EmDay EmMon EmYear

Power:
PInt PDay PMon PYear Pmax PmDay PmMon PmYear

Creation of Virtual Channels, Time and Calendar Functions:
VSUM VIRT DAY WDAY MON YEAR HH MM SS FROM TO DURATION TIME DATE

Period Readings List:
Setting Periods, Formatting, Index and Erasing List, Erasing Channel.
INTERVAL INTERVALSOURCE SYNC FORMAT INDEX ERALIS ERACHAN

Station Parameters:
Station and Group Name, Current tariff with Tariff Parameters.
STATION GROUP TARIFF TFIX TUnit COSTFAC1 COSTFAC2 TARIFFSOURCE LEVEL

Channel Parameters:
Channel Name, Meter Constant, U Ratio, I Ratio, Channel Fixed Point, ON/OFF, PFaktor ...
CHANNEL MCONST URAT IRAT EUnit PUnit EDGE PULSE ONOFF CHANNELFIX PFACTOR

Input Query, Relay Control, Additional Tools, List of Interruptions to Power Supply:
IN RELAY RELAYMODE RELAYNAME STATREL STATCHECK DISPLAY KEY PAUSE POWERFAIL PASSWORD

P and H Programs, Error Handling, Variables, Print Commands:
P PLIST H HLIST HBREAK ERR ERRNR LERR LBERR A ! PRINT DELIMITER LANGUAGE

Directory of ECS LAN Devices, other Tools:
DIR DIRN DIRS INDIR ID SETID ENUM FINDER MELD REM STATUS VER DATEFORMAT

Tool Box

The following useful example programs are designed to make you more familiar with the ECL interpreter.
Please note that H programs are executed cyclical one after the other and so the calculation time for each H program effects the whole H program period time.

Hallo!

All stations show "Hallo !" in their display if Pmom (1) > 30kW. An example of the practical use of background programs:

```
H 10='pmom- 1,30,>,if,all,meld "Hallo !" 2'
```

Setting the Time and Date of All Stations in the Network

This command line sets all the clocks in the ECS LAN.

```
all, time=12h34.56; date=16.08.93
```

Synchronising All Clocks in the Network

All clocks are synchronised at 0h0:15 every day from station A: (for example). The 'x' in the time and date entries is replaced by the current values of the station on which the program is run. The ALL loop command with extension ' ' causes the ALL loop to be run for all stations except the 'home' station (in this case A).

```
<A>   H 10= 'if 0h0.15, ALL-, Time=x:x:x, Date=x.x.x'
```

Tariff Switching

T1 (NT) applies from 21h to 6h, otherwise T2 (HT) applies. The tariff source must be set to 'Program'!

```
<A>   H 11= 'hh,6,>=,hh,21,<,&,if, Tariff=2, else, Tariff=1'
```

Tariff Synchronisation in Network

System wide tariff synchronisation from station A: (for example). The tariff source must be set to 'Program' at all stations!

```
<A>   H 12= ' tariff-,all-,dup,tariff=. '
```

Alternatively: The update of the current tariff does not take place continually, but only at a change of tariff (using **IFF**, the sequence of commands between IFF and ELSE or ELSE and ENDIF is only carried out once after the condition changes):

```
<A>   H 12= ' tariff-,1,-,iff,all-,tariff=2,nexta,else,all-,tariff=1'
```

Interval Synchronisation in Network

At station A: (for example), the external synchronising pulse is fed in from channel 24 (A:Intervallquelle=24). This station then performs period synchronisation throughout the ECS LAN.
The period sources of the slave stations must be set to 'Program' (Intervallquelle=P).

```
<A>   H 13= ' Sync,iff,all-,sync= ',tariff=1'
```

Printout of H19 Print Program Every Day at 6h30

The following are printed:

- The current total energy of all meter channels switched on (throughout the system)
- The energy consumed the day before by all meter channels switched on (throughout the system, with details of the period)

```
<A>   H 14= 'if 6h30, h 19'
<A>   H 19= 'all, etot& *, na, !!, all, eDay//& * 1, na, !!'
```

Printout of H19 Print Program at 12h on the First of Each Month

The following are printed:

- The current total energy of all meter channels switched on (throughout the system)
- The energy consumed the month before by all meter channels switched on (throughout the system, with details of the period in the header: "Consumption in April 1993")

```
<A>  H 14= 'if 1.x.x 12h, h 19'
<A>  H 19= 'EMon% "Consumption in %/dM 19%/dj"11,!!,all,EMon&*1,na,!!'
```

Copying P and H Programs

P10 copies all P programs, P11 copies all H programs to station B:

```
P 10='! "Copy all P-Programme to B:",0,19,fori,i,p- .,i,B:p .=$'
P 11='! "Copy all H-Programme to B:",0,19,fori,i,h- .,i,B:h .=$'
```

Operating Time

24V is present at input 4 when the device is switched on, otherwise it is 0V.
The operating time is read from Etot in channel 3 (in seconds), Etot in channel 4 indicates how often the device has been switched on. P 18 is called to initialise the counter, and evaluation is performed by H 6:

```
P 18='! "Initialise Operating Time Meas.",zkonst 4=1,chan 4=Count,p 19'
P 19='   chan 3=OpTime,EUnit 3=Sec,EUnit 4="",kfix 3..4=0,Etot 3..4=0'
H 6='in- 4,if,time-,dup,a 6,-,Etot- 3,+,Etot 3=., else,time-, endif, a 6=.'
```


Meter Control

Meter 1 counts only while input 8 is at logical '1'. The command STARTSTOP is used the control the meter channel in question.

```
H 7='in- 8,iff,startstop 1=1,else,startstop 1=0'
```

Activation of Relay Depending on PMOM

Relay 1 of station B: is activated, as soon as the instantaneous power of virtual channel V2 at station A exceeds 55 kW.
This background program runs in station B and monitors Pmom in station A.

```
<B>  H 10= 'A:Pmom- V2, 55, >, IF, Rel 1=1, ELSE, Rel 1=0'
```

Checking number of Devices in ECS LAN

If the number of ECS LAN devices deviates from the normal number (4 in this example), a warning is indicated on all LCD displays of the stations, and relay 4 of station X1 is activated.
This background program runs in station A: The precise number of devices must be known and included as a constant in the program.

```
<A>  H 18= 'Bus-,4,!=,dup,X1:Rel 4=.,IF, All,meld "BUS Inconsist." 2'
```

Resetting to Summer/Winter Time

An H program is required in a specific station (eg. the one performing system wide time synchronisation) for each changeover time.
As soon as the changeover time arrives, the H program resets all the clocks in the system correspondingly, and finally erases itself.

```
<A> H 15='rem SUMMER/WINTER,if 26.9.93 3h,h?,h .="",all,time,3600,,time=.'
<A> H 16='rem WINTER/SUMMER,if 24.4.94 2h,h?,h .="",all,time,3600,,time=.'
<A> H 17='rem SUMMER/WINTER,if 25.9.94 3h,h?,h .="",all,time,3600,,time=.'
```

Bridging a Missing Synchronising Pulse

If the synchronising pulse does not arrive for more than 10 seconds after the preset period, an "artificial" interval is generated. If a station is acting as "period synchronising master", the program only needs to be installed in that device.

```
<A>  H 14= 'rem SYNC-BRIDGING,  sync/, intervall-, -,10,>,if, sync+='
```

Generating Pulses from the Energy of a Virtual Channel

A pulse is provided at relay 1 for every 10 kWh (division factor 1/10) of energy from virtual channel V1. A background program (H 0) and a P program (P 0) are both required, as well as a variable (A 0).

P 0 is called by H 0 because there is insufficient memory to implement all the commands in H 0. Variable A 0 is initialised when H 0 is programmed in. Pulse output starts at this moment the pulse duration and the length between pulses can be set (cf. bold parts of P 0). 'PAUSE 0' causes a pulse duration / gap of around 80 ms, but otherwise the duration / gap can be adjusted in steps of 200 ms. An example of a 400 ms pulse duration / gap: 'PAUSE 400'.

If V1 is altered by resetting or re assigning the physical channels, an attempt is made to restore the count in the event of an increase by generating successive pulses (if more than 50 pulses are required, no attempt is made to restore the count). If the value is reduced, pulse generation starts again automatically from the reduced value.

In H 0, the division factor of pulse generation is printed in bold type (pulse number=energy/division factor).

```
H 0='1,iff,Etot- v1,10,/,int,a=.,endif,Etot- v1,10,/,int,dup,a,-,dup,dup,p,a=.'
P 0='0,>,swap,51,<,&,if,2,*,1,fori,i,2,mod,rel i=.,PAUSE 0,nexti,else,drop'
```

Querying the Readings List

All **Pint** entries (to Pint 1) in the readings list for channel 1 at station A: are output from 17.08.92 18h45. The time and date "from" and "to" are included in the printout:

```
<A>index 17.8.92 18h45, pint// 1 . *
17.08.92 18:30:00 -- 17.08.92 18:45:00 : Pint-215 (01) = 1.23 kW
17.08.92 18:45:00 -- 17.08.92 19:00:00 : Pint-214 (01) = 1.80 kW
17.08.92 19:00:00 -- 17.08.92 19:15:00 : Pint-213 (01) = 1.12 kW
17.08.92 19:15:00 -- 17.08.92 19:30:00 : Pint-212 (01) = 2.10 kW
17.08.92 19:30:00 -- 17.08.92 19:45:00 : Pint-211 (01) = 2.05 kW
17.08.92 19:45:00 -- 17.08.92 20:00:00 : Pint-210 (01) = 2.07 kW
...
```

All **Eint** are printed out in database format (up to Eint-0) with time and date "to". The sequence of commands is assigned to P 2:

```
<A>P 2='Eint/## # * **
<A>p 2
16.08.92;17:45:00;1;0.5;0.75;0.99
16.08.92;18:00:00;1.01;0.1;0.76;0.80
16.08.92;18:15:00;0.99;0.48;0.75;1.02
16.08.92;18:30:00;0.89;0.5;0.76;0.99
16.08.92;18:45:00;1;0.52;0.77;1
16.08.92;19:00:00;1.01;0.51;0.75;0.98
...
```


Creating a Database in ASCII Format; Columns = Channels

A database is created in ASCII format (separator ;), containing the following selection of readings from all summation stations connected to the ECS LAN:

Energy totals ETOT, ETOTT1, ETOTT2 and instantaneous power PMOM.
Column descriptions: **Station, Function, Value Channel 1,..., Value Channel V8**
The first line contains the column headings.

Station	Function	1	2	3	...	32
A	Channel	HeatRoom	Motor015	Chan-3	...	TotCost8
A	Etot	12.7	6.956	0	...	147.9734
A	ETOTT1	12.7	6.956	0	...	147.9734
A	ETOTT2	0	0	0	...	0
A	Pmom	0	0	0	...	0.37
...						
C1	Channel	Motor001	Motor002	Motor003	...	TotMot01
C1	Etot	0	17.22	158	...	1379.5554
C1	ETOTT1	55.3	0.12	0	...	147.9734
C1	ETOTT2	0	0.93	0	...	192.11
C1	Pmom	0.54	1.17	0	...	5.557
....						

The ASCII database is created as follows:

```
Station;Function;1;2;3;  ...  ;32
A;Channel; HeatRoom; Motor015; Chan-3;  ...  ;TotCost8
A;Etot;12.7;6.956;0;  ...  ;147.9734
A;ETOTT1;12.7;6.956;0;  ...  ;147.9734
A;ETOTT2;0;0;0;  ...  ;0
A;Pmom;0.37;0;0;  ...  ;0.37
...
C1;Channel;Motor001;Motor002;Motor003;  ...  ;TotMot01
C1;Etot;0;17.22;158;  ...  ;1379.5554
C1;ETOTT1;55.3;0.12;0;  ...  ;147.9734
C1;ETOTT2;0;0.93;0;  ...  ;192.11
C1;Pmom;0.54;1.17;0;  ...  ;5.557
...
```

Calling program P 10 from the station connected to the PC via RS232 generates the required output, P 11 to P 13 are auxiliary programs of P 10.
Program P 10 (together with the auxiliary programs) can only be called from the station connected to the PC (logged in under its ID): P 10
Note: An ID in the form AA: always addresses the station to which the PC is connected.

```
AA:P 10='! "Station;Function;",aufz##+ **,all,AA:p 11,AA:p 12,AA:p 13'
AA:P 11='kenn.,!+ ";Channel;", channel##+ **'
AA:P 12='kenn.,!+ ";Etot;", etot##+ **, kenn.,!+;EtotT1;",etotT1##+ **'
AA:P 13='kenn.,!+ ";EtotT2;",etotT2##+ **,kenn.,!+ ";Pmom;  ",PMOM##+ **'
```

Using **ECSoft 2**, it is possible to divert output direct to a file - a script metalanguage allows the operation of ECSoft to be automated.

Creating a Database in ASCII Format; Columns = Functions

A database is created in ASCII format (separator ;), containing the following selection of readings from all summation stations connected to the ECS LAN:

Energy totals ETOT, ETOTT1, ETOTT2 and instantaneous power PMOM.
Column descriptions: **Station, Channel number, Channel, ETOT, ETOTT1, ETOTT2, PMOM**
The first line contains the column headings.

Station	ChanNo	Channel	Etot	EtotT1	EtotT2	Pmom
A	1	HeatRoom	12.7	12.7	0	0.37
A	2	Motor015	6.956	6.956	0	0
A	3	Channel-3	0	0	0	0
A	...					
A	32	TotCost8	147.9734	147.9734	0	0.37
...						
C1	1	Motor001	0	55.3	0	0.54
C1	2	Motor002	17.22	0.12	0.93	1.17
C1	3	Motor003	158	0	0	0
C1	...					
C1	32	TotMot01	1379.5554	147.9734	192.11	5.557
...						

The ASCII database is created as follows:

```
Station; ChanNo;Channel;Etot;EtotT1;EtotT2;Pmom
A;1;HeatRoom;12.7;12.7;0;0.37
A;2;Motor015;6.956;6.956;0;0
A;3;Channel-3;0;0;0;0
A;...
A;32;TotCost8;147.9734;147.9734;0;0.37
...
C1;1;Motor001;0;55.3;0;0.54
C1;2;Motor002;17.22;0.12;0.93;1.17
C1;3;Motor003;158;0;0;0
C1;...
C1;32;TotMot01;1379.5554;147.9734;192.11;5.557
...
```

Calling program P 15 from the station connected to the PC via RS232 generates the required output - P 16 ... P 18 are auxiliary programs of P 15.
Program P 15 (together with the auxiliary programs) can only be called from the station connected to the PC (logged in under its ID): P 15
Note: An ID in the form AA: always addresses the station to which the PC is connected.

```
AA:P 15='! "Station;ChanNo;Channel;Etot;EtotT1;EtotT2;Pmom",AA:p 16'
AA:P 16='all,fori **,i,AA:p 17,AA:p 18,nexti,nexta'
AA:P 17='dup,channel.& .,!+ ";",dup,etot+# .,!+ ";",drop,dup,etott1+# .'
AA:P 18='!+ ";",drop,dup,etott2+# .,!+ ";",drop,pmom+# .,drop'
```

Using **ECSoft 2**, it is possible to divert output direct to a file - a script metalanguage allows the operation of ECSoft to be automated.

ECL Command Reference

Contents

General Information

Online Command List

```
-------------------------------------------------------------------------------
ECL Command List created with "Help"
-------------------------------------------------------------------------------
+ - * / & | ^ < <= > >= == != DUP DROP SWAP PICK
 IF IFF ELSE ENDIF FORI I NEXTI FORJ J NEXTJ ALL NEXTA EXIT RETURN
  SQRT SIN COS ASIN ACOS DEG RAD EXP LOG ** ABS FRAC FIX INT INTR MAX MIN MOD
   Etot EtotT1 EtotT2 EtotT1T2 CostT1 CostT2 CostT1T2 Pmom
    EInt EDay EMon EYear Emax EmDay EmMon EmYear
     PInt PDay PMon PYear Pmax PmDay PmMon PmYear
      VSUM VIRT DAY WDAY MON YEAR HH MM SS FROM TO DURATION TIME DATE
     INTERVAL INTERVALSOURCE SYNC FORMAT INDEX ERALIS ERACHAN PASSWORD
    STATION GROUP TARIFF TFIX TUnit COSTFAC1 COSTFAC2 TARIFFSOURCE LEVEL
   CHANNEL MCONST URAT IRAT EUnit PUnit EDGE PULSE ONOFF CHANNELFIX PFACTOR
  IN RELAY RELAYMODE RELAYNAME STATREL STATCHECK DISPLAY KEY PAUSE POWERFAIL
 P PLIST H HLIST HBREAK ERR ERRNR LERR LBERR A ! PRINT DELIMITER LANGUAGE
DIR DIRN DIRS INDIR ID SETID ENUM FINDER MELD REM STATUS VER DATEFORMAT
-------------------------------------------------------------------------------
General: INFO SYNTAX EXTENSION PARAMETER STRINGS STACK IDs RS232
For help on a topic: HELP <CommandName> or ? <CommandName>
```

General Information

```
General information on using the ECL interpreter commands:
Interrupt current output: ^X (CTRL and X at the same time)
Interrupt output of background programs:
- Input possible despite output
- Interrupt command: cf. HBREAK (16s pause of H programs)
- Control character ^B suspends the simultaneous output of H programs for
  10 seconds (H program output is ignored for this period)
Obtaining help on specific topics:
- Not all the letters of a search term need to be entered if the abbreviated
  form is unambiguous.
- Short forms exist for certain commands (given in brackets); the command
  search also works with the shortened commands.
- Complete output of all help topics: ? BOOK
```

ECL SYNTAX

```
ECL SYNTAX - Metalanguage for definitions
<abcd> :=          : Definition
[ ]                : Optional details
<ab> | <cd>        : Alternatives
{ .. } : List,     [..]^ or {..}^ : Repeat,      __ : Space
<command_sequence> := <ID><command> [ , | ; <command_sequence>]
<command> := <text><ext> [__<par>[__<par>[__..]]] [=[<par>[__<par>[__..]]]]
<ID>       := {A AA A1 .. A9 AN B B1 .. B9 C .. Z4 ZZ} : | :: [__]
<ext>      := [{  ! + - # . * / ^ $ _ ? | @  }]^
<par>      := <real> | <string> | <enumeration> | . | $
<real>     := [-]<integer>[ E <integer>]
<integer>  := [-]{0 .. 9}^
<string>   := [" | ']  <text> [__<text>]^  [' | "]
<text>     := {a .. z A .. Z 0 .. 9 _ - +}^
<enumeration> := { * ** # ## <channel> } [ .. | + | - | ^ { <kanal>]]^
<channel> := <integer> | {V1 .. V8}
```


Extension

```
The extension <ext> influences the behaviour of the commands. [Examples cf.]
=============                                                 [EXTEXAMPLES ]
The following rules normally apply:
-    :  Suppress output (where provided)
--   :  Divert output to string buffer (command must recognise ext %)
+    :  Concatenate output (NO new line at the beginning)
!    :  Forces output (eg. P!3 prints a listing of program P3)
%    :  The first parameter formats output (cf. PRINTFORMAT)
&    :  The ID is also printed at the start of the line (cf. ID)
* @  :  Command modifications, eg. pulse instead of energy output (cf. Etot)
_    :  Harmonised (rewritable) output of energy commands
|    :  Additional output format options (cf. Etot)
.    :  Database output, separator ';', terminator <CR><LF>
..   :  As .  but separator ';' between output blocks
...  :  As .. but separation of continuous lines with ';', not <CR><LF>
#    :  Output main value only. ## and ### similar to .. and ...
$    :  Used with . or #, output of names in ""  ($$ : time also)
/    :  Output with time details (other details cf. FROM or TO)
^    :  Output with time details with seconds count from 1.1.90
```

Extension Examples

```
Examples of the use of extensions
Etot     1+V2       :     Etot (01:Room501 ) = 874.01 kWh
                          Etot (V2:Room777 ) = 12.74 kWh
Etot.    1+V2       :     Etot;1;Room501;874.0124;kWh
                          Etot;26;Room777;12.739;kWh
Etot..   1+V2       :    Etot;1;Room501;874.0124;kWh;Etot;26;Room777;12.739;kWh
Etot#    1+V2       :     874.0124
                          12.739
Etot##   1+25+V2    :     874.0124;100;12.739
EMON_    1 2        :     EMon 01 2 = 500.00 kWh          [rewritable]
EMON*    1 2        :     EMon* (01:Room501 ) = 50000.00  [number of pulses]
EMON//   1 2        :     01.09.92 00:00:00--01.10.92 00:00:00  EMon-2 (01:...
EMON/##  1..4 2     :     01.10.92;00:00:00;500;1.1234;7555;0.0001
EMON^##  1..4 2     :     86745600;500;1.1234;7555;0.0001
INTERVALL&          :     A:INTERVALL = 15 minutes
```

Command Parameters

```
Commands take the following PARAMETERS (examples):
Enumerations:    [cf. FINDER for searching by name]
=============    <enumeration> range 1..32 or (where possible) 0..31
  2..9+15+17     : 2 up to 9 and 15 and 17
  5+V1..V4       : 5 and V1(=25) up to V4(=28)
  2..16-5-8+24   : 2 up to 16 and 24, without 5 and 8
  *              : all ON channels (cf.ONOFF). If no reference possible: as **
  #              : all formatted channels (cf. FORMAT)
  **             : 1 up to 32
  *-3..6         : all '*' without (3 up to 6)
  *+7^+10..12    : complement of (all '*' and 7) plus 10 up to 12
  .              : single item from Stack, decimals places removed.
General number input:   <real>          <integer>       from Stack
=====================   -12.34E3        8               .
General string input:   <string>                        last string (buffered)
====(cf STRINGS)=====   Hallo   |   "Hallo 'World'"     $
```

FINDER: System Wide Search

```
System wide search for channel names, relay names or station names:
- Instead of <enumeration> the name of the channel required can be given in
  channel related functions (also relay names with REL... and station names
  with KENN). A search is conducted throughout the ECS LAN system, starting
  with the prompt station.
- The search term must begin with a letter, otherwise a $ must be placed in
  front of it.
- "name*" searches for the first occurrence of "name....", case insensitive.
- "name**" provides all matching channels from starting with the first
  possible (!) station.
- <search_term>& searches only the current station.
- <search_term)@ suppresses '.. not found' messages.

FINDER (FI)  :  Function to provide station details from the search term.

Call:           FINDER <search_term>
Stack:          - >>> <channel_number> <IDnumber> <1:found/0:not_found>
ext:            * (search for relay name)       @ (search for station name)
```

STRINGS

```
Including special / control characters in strings:
\#      :  "                    \"      :  "
\|      :  '                    \'      :  '
\\      :  \
\b      :  0x08 Backspace
\l      :  0x12 ^L
\n      :  0x0A (LF)
\r      :  0x0D (CR)
\t      :  0x09 (TAB)
\nnn    :  nnn= '3-digit decimal number' corresponding to the character code
- Conversion of the metacode into the corresponding character
  does NOT take place during the ASSIGNMENT of a string (Except \" und \'):
    p='! "Letter \#\065\#"' , p!        : P 0='! "Letter \#\065\#"'
    p                                   : Letter "A"
```

Parameter Stack

```
Depth:      15
Data type:  <real> : 64 bit floating point number (15 significant dec.places)
An intermediate buffer exists for strings, containing the last string result.
Referenced sing $ in <par>.
Note: The stack and the buffer are only valid during the processing of a line
(interlacing of programs possible). The stack is erased when the prompt
returns. This creates a consistent programming environment.
Variables
=========
Twenty registers A0 .. A19 are available for long term storage of
<real> numbers (cf. A).
```


IDs

```
Each summation station possesses a unique ID (KENNUNG)
There are 255 IDs:  A, A1..A9, B, B1..B9 ... Z4   and 3 special IDs:
AA :    ID of the station connected to the RS232 interface
ZZ :    ID of the station indicated by the interpreter prompt
AN :    Fetches an ID as a number from the stack (A==1 .. Z4==255)
A1::<command>  or   A1::       : complete ID context switch
A1: <command>  or   A1:, ...   : switches ID for current line only
A1:<command>                   : ID applies to current command only
Example of special ID ZZ:
  A P program is started while logged on to a remote station. P contains an
  ALL loop from which a further P program is called which relates to the
  same logical station. Because reference to the fixed station is not
  provided within the ALL loop, ZZ: makes it possible to reinstate the
  fixed reference, without having to name the station explicitly.
  <C> .. All, .. , P 15, ...      calls P 15 from ALL stations
  <C> .. All, .. , ZZ:P 15, ...   calls P 15 only from station C:
```

RS232 Interface Protocol

```
^M  (RETURN)        13d   : sends the input line, the system prompt appears
                            after the response ( eg. <A1> )  [no prompt: ^W^M]
^J  (CTRL-RETURN)   10d   : as RETURN, but ^Z (SUB,26d) appears instead of the
                            prompt (intended for host connections)
^X  (CAN)           24d   : Cancel output, erase all buffers and flags
^Y  (EM)            25d   : Erase line, no output
ESC                 27d   : Erase line, cursor moves to new line on screen
---- Protocol Flags ----
^B  (STX)           2d    : suppresses input echo for current line.
^A  (SOH)           2d    : Prefix for coded error message: ^A nnn
^A^A                      : as ^A, but instead of nnn error code: ^A <ErrText>
^V  (SYN)           22d   : Erase internal checksum and prefix for output
                            with "Checksum after SUB"
^V^V                      : Checksum must follow ^M or ^J, response: ACK/NAK
-  Protocol Flags ^A, ^V and ^B always only apply to the next command.
-  ^V and ^B suppress inclusion of H program output for 10s+command duration.
-  Checksum: addition to 16-bit INT, fixed 4-digit representation in HEX.
```

Reference

Arithmetic Operators

```
+ - * / ** & | ^ < <= > >= == != !
=====================================
+  :    a  b >>> (a+b)
-  :    a  b >>> (a-b)
*  :    a  b >>> (a*b)
/  :    a  b >>> (a\b)
** :    a  b >>> (a to the power of b)
&  :    a  b >>> (a logical AND b)
|  :    a  b >>> (a logical OR b)
^  :       a >>> (NOT a)
<  :    a  b >>> (Comparison a <  B) yes= 1, no= 0
<= :    a  b >>> (Comparison a <= B)
> , >= , == , !=    :   a>b, a>=b, a equal_to b, a not_equal_to b
!  :    stack output function, takes a number from the stack and outputs it.
       Cf. PRINT for other uses of the ! command
```

General Number Manipulations

```
General Number Manipulations:  ABS FRAC INT INTR MIN MAX MOD   FIX
                  Stack:
ABS:    <value> >>> absolute value function (<value>)
FRAC:   <value> >>> fraction (<value>)
INT:    <value> >>> integral part (<value>)
INTR:   <value> >>> 5/4 rounded integral part (<value>)
MIN:     a  b   >>> the smaller of a and b
MAX:     a  b   >>> the greater of a and b
MOD:     a  b   >>> (a modulo b)
FIX:    <value> >>> -   <value> provides the decimal place applying to the
                        rest of the line. FIX 9 switches to floating point
                        representation with up to 9 decimal places (default).
                        FIX <value>   ==   <value>, FIX
                        Example:        12.345, FIX 0,!   -->  12
                                        12.345, 2, FIX,!  -->  12.35
                                        12.345, FIX 5,!   -->  12.34500
                                        12.345, 9, FIX,!  -->  12.345
```

A Registers

```
A 0 .. A 19     : Registers A 0 .. A 19 for <real> numbers
Call:   A <enumeration> [= <newValue>]
Stack:    -   >>> Contents(Ai)  : Query (total created for enumerations)
        value >>> -             : Assignment
ext:    + - . # ! ++ --
- without <enumeration> == A 0
- A1 .. A19 correspond to A 1 .. A 19,  A5! == A! 5
- Increment (+1):  A++ <enumeration>
- Decrement (-1):  A-- <enumeration>
- Addition of <value> to register:      A++ <enumeration>=<value>
- Subtraction of <value> from register: A-- <enumeration>=<value>
- <newValue>=={t|z} assigns the current time in seconds (with 1/100s).
ALIST   or   ALIST <enumeration> : Lists A Registers (corresponds to A! *)
```


ALL NEXTA

```
ALL  NEXTA (NA)  :  Program loop "all IDs"
The loop command allows a command to address all stations. There is
no control variable, and the current ID of the line is incremented
(only IDs from stations connected to the ECS LAN).
Call:   ALL [<from_IDNumber> [<to_IDNumber>]]          NEXTA
Stack:  - >>> -                                        - >>> -
ext:    -       : omits the station with the current ID
Example:
  ALL , Etot 1, nexta     Output: Etot Channel1 of all devices
- When calling P sub programs from within the ALL loop, please note the
  example of the special ID ZZ: (cf. IDs)
- Cf. example in INDIR for a demonstration of the internal sequence of ALL...
```

BUS BUSL BUSR

```
BUS  BUSL  BUSR : ECS-LAN status
Outputs the number of devices connected to the ECS LAN:
   Total number,  Number BUS left  (of which direct neighbours)
                  Number BUS right (...)

Call:   BUS
Stack:  -  >>>  <anzahl_Bus_Teilnehmer>
                BUS:  Total number
                BUSL: Number left, or  1 if BUS L error
                BUSR: Number right, or  1 if BUS R error
ext:    + - . #
Examples:
  BUS    :  Bus devices total = 8, BL= 3(1), BR= 4(4)
  BUS.   :  8;3;1;4;4;0;0   [last two values: L ;R errors (1:error)]
  BUS#   :  8
  BUSL#  :  3
```

CHANNEL

```
CHANNEL (CHAN)  :  The name (max. 8 characters) of the channel
Call:   CHANNEL <enumeration> [=<string>]
Stack:  - >>> -
ext:    + - . # $ %
- special characters allowed:   !$%/.:-+#_<>|
- $ and . must not be used as the first character.
- In order to allow the channel name to be used as a database field name,
  only '_' should be used apart from letters and numbers.
```

COSTFAC1 + 2, TFIX

```
COSTFAC1 COSTFAC2 : Cost factors for tariffs 1 and 2:
Call:   COSTFAC1 [=<factor>]
Stack:  - >>> <factor>
ext:    + - . # %
TFIX : Fixed point costs
Call:   TFIX [=<fix>]
Stack:  - >>> <fix>
ext:    + - . #
```

DELIMITER

```
DELIMITER (DELI) (DL) : Sets the delimiter for the rest of the command line
Call:           DELI  [[<fieldseparator>][__<recordseparator>]]
Stack:          - >>> -
ext:            *
- DELI without parameters resets to default values:
  <fieldseparator> = ';'  <recordseparator> = '\r\n'   [<CR><LF>]
- Both delimiters may contain up to 8 ASCII characters.
- In order to maintain a consistent programming environment, the new
  delimiters apply only to the rest of the command line, after which
  the default values apply again.
- Ext * reverses the order of parameters: DELI* <recordsep>[__<fieldsep>]
- If only one parameter is stated, the other remains unchanged.
* Instead of the normal ';' separator, output destined for the dBase command
  APPEND FROM ... DELIMITED should be delimited with ','. The "" delimiters
  are not necessary for dBase (Extension $):
       DELI ","; Etot.$ 1      -->     "Etot",1,"Chan-1",127.34,"kWh"
```

DIR DIRN DIRS

```
DIR DIRN DIRS : Directory of all ECS LAN devices
DIR     : IDs only of all devices (A:)
DIRN    : IDs and station names of all devices (A:U1600)
DIRN_   : ID and station name of selected station
Stack:  - >>> -
ext:    + - # . %
DIRS    : Like DIR, but includes information for each ID
          L=Left, R=Right, + = direct neighbour, * = "Me"
Stack:  - >>> <number_of_BUS_devices>
ext:    + - # .
```

DISPLAY

```
DISPLAY (DD) :  Fetch LCD display
Call:           DISPLAY [<keyString>]   <keyString> : cf. TASTE
Stack:          - >>> -
ext:            + - # . $
Output:         3 lines with 1st and 2nd LCD lines and LED status of 8 LEDs
Example:  "01:  1234.56 kWh"
          "Chan-1      Etot"
          "*_**__**"          : '_' = off, '*' = on;  BUS/L BUS/R REL1 .. 24V
- the cursor in the display is encoded with a preceding '&'
- Ext. # causes output without ""
```


DUP DROP SWAP PICK

```
DUP (DU)  DROP (DR)  SWAP (SW)  PICK : Stack manipulation
DUP     : n1 n2     >>> n1 n2 n2
DROP    : n1 n2 n3 >>> n1 n2
SWAP    : n1 n2     >>> n2 n1
DUP  <n>: performs DUP n times
DROP <n>: performs DROP n times
PICK <i>: copies the ith stack element to the top. 'PICK 1' == 'DUP'
```

DVSUM DVIRT

```
dVSUM dVIRT  :  Defining function to create virtual channels (in background
                H programs) with "differential link".
dVSUM:  adds channels in the enumeration in the intermediate register
Call:   dVSUM <enumeration> [<factor>]
Stack:  - >>> -
dVIRT:  Assigns the intermediate register-diff-sums to the virtual channels
Call:   dVIRT <enumeration> [<factor>] =
Stack:  - >>> -
ext:    +     : if an assignment is made to a physical channel, the measured
                energy is added to the calculated energy. Without the +, the
                measured energy has no effect.
- dVIRT can be used for all channels.
- The sums in the virtual channels are entirely independent of the base
  channels. Sums generated can be directly erased.
- STARTSTOP can therefore be used regardless of the base channels.
- One off addition of an energy quantum to a channel: cf. DELTA
```

DELTA

```
DELTA   :       One-off addition of an energy quantum to a channel. The
                procedure corresponds to the single use of DVIRT with a
                specific energy amount.
Call:   DELTA <enumeration> = <value>
Stack:  - >>> -
- STARTSTOP permits assignment (STSP==1) or ignores it (STSP==0).
```

EUNIT PUNIT TUNIT

```
EUNIT PUNIT TUNIT :     Units of energy E, power P and tariff
                        (max. 4 characters)
Call:           EUNIT <enumeration> [=<string>]
Stack:          - >>> -
ext:            + - . # $ %
- For usable characters, cf. CHANNEL.
```

EINT PINT

```
EINT PINT : Energy and power in period (Intervall)
Call:   Eint <enumeration> [<startindex>] [<how_many>]    [=<newValue>]
Stack:  -  >>>  <value>  (with enumerations: <value>=total(individual values))
ext:    + - . # / * _ | $ %
Output starts with <startindex> and is carried out in chronological order
<startindex> no details  --> <startindex>=0
<startindex>=='*'        --> <startindex>=first entry in time
<how_many> no details    --> <how_many>=1
<how_many>=='**' : all entries from <startindex>  (with Eint-0)
<how_many>=='*'  : all entries from <startindex>  (WITHOUT Eint-0)
<how_many> greater than <startindex> : "      "      "       "
- cf. also: INDEX
- EINT## is suitable for fast data transfer
```

ETOT ETOTT1.. COSTT1.. PMOM

```
Etot:           Total energy
EtotT1:         Total energy tariff 1
COSTT1:         Costs tariff 1
EtotT2:         Total energy tariff 2
COSTT2:         Costs tariff 2
EtotT1T2:       Total energy tariffs 1 + 2
COSTT1T2:       Costs tariffs 1 + 2
PMOM:           Instantaneous power
Call:   Etot <enumeration> [=<newValue>]
Stack:  -  >>>  <value>  (with enumerations: <value>=total(individual values))
ext:    + - . # / * _ | $ %
- The ext. * can be used to output the (calculated) number of pulses, instead
  of energy. Example: 'Etot* 1'
  Mconst, Urat and Irat of the corresponding channels are used in the
  calculations (even with virtual channels)
- Ext. | outputs only <value> and <unit>: Etot| 1  -->  123.34 kWh
- <newValue>==0  -->  Time data also erased (FROM=0,TO=0)
```

EDAY EMON EYEAR EMAX
PDAY PMON PYEAR PMAX

	in period	per day	per month	per year
energy:	(cf. Eint)	EDay	EMon	EYear
ave. power:	(cf. Pint)	PDay	PMon	PYear
Energy maximums:	Emax (10 reg.)	EmDay	EmMon	EmYear
ave. power:	Pmax (10 reg.)	PmDay	PmMon	PmYear
List length:	variable	10+curr.day	12+curr.month	2+curr.year

```
Call:   EDay <enumeration> [<enumeration_index>]  [=<newValue>]
ext:    + - . # / * _ | $ %
Stack:  -  >>>  <value>  (with enumerations: <value>=total(individual values))
- <enumeration_index>=0 or no details: continuous cycle
- <newValue>=0          -->  Time data also erased (FROM=0,TO=0)
- EMAX <..> <index>=0   -->  Erase from <index> to <maxIndex> (only EMAX)
```


ENUM - Enumeartion

```
ENUM (AUFZ) : Returns list with the channel numbers included in <enumeration>
Call:   ENUM <enumeration>
ext:    + - ##
Stack:  - >>> number of elements
- Range of <enumeration> 1..32 oder 0..31
- <enumeration> examples cf. PARAMETER
```

ERACHAN ERALIST

```
ERACHAN : Erase all readings of a channel
          (except the values in the readings list).
Call:   ERACHAN = <channelEnumeration>
ERALIST : Erase period readings list, starting from index.
Call:   ERALIST = <fromIndex>    : erase including <fromIndex> .. end
        ERALIST = *              : whole list
```

ERR ERRNR LERR LBERR

```
ERR   : Output error state of background H programs
LERR  : - >>> n  in the event of error in last program execution
LBERR : - >>> n  in the event of bus error in last program execution
ERRNR : Error number --> description
Aufruf: ERRNR <enumeration>              ERRNR
Stack:  -  >>>  -                        <number>  >>>  -
Ext:    + - . #  $ %
```

EXIT RETURN

```
EXIT    : premature termination of running program.
Call:   EXIT
RETURN  : premature termination of running sub program.
RET     : short form of RETURN
Call:   RETURN
```

FORI NEXTI FORJ NEXTJ

```
FORI I NEXTI (NI)   FORJ J NEXTJ (NJ)  :  Program loops
FORI has the control variable I, FORJ has J. Each sub program has its own
FORI/FORJ set. <from> and <to> are fetched from the stack; alternatively,
an enumeration may be specified.
Call:   FORI                      FORI <enumeration>      I          NEXTI
Stack:  <from> <to> >>> -         - >>> -                 - >>> 1    - >>> -
Note:
- The loop is executed at least once. I or J are incremented by one, or set
  to the next possible enumeration element, on each pass.
- If there are no commands after the NEXTI/J, NEXTI/J may be omitted.
- <enumeration> may cover the range 0..31 or 1..32
- FORI-- oder FORI*  : Reverses the order of processing
Example:        ! Test:, 2,5, FORI, i, !+ " " ., nexti
Output:         Test 2 3 4 5
Equivalent:     ! Test:, FORI 2..5, i, !+ . " "
```

FORMAT

```
FORMAT  :  Formats the readings list (calls data with EINT)
- The data list is reformatted (using Erase) only if an <enumeration> is
  assigned to the FORMAT command. Without the assignment, FORMAT outputs
  the current formatting, and the stack receives the number of channels
  which can be stored.
- The formatting process organises the memory for a certain number of
  channels - however, the total depth of memory is dynamically related to
  the length of the synchronising period. If the memory is full, the data
  is rotated: the oldest entry is erased to make room for the newest.
- Any enumeration (even with virtual channels) is possible.
Call:           FORMAT
Stack:          - >>> number_of_channels
ext:            + - . # ##
Output:         Formatting info (not with . and # ##), and list of channels
Call:           FORMAT = <enumeration>     Formatting according to
Stack:          - >>> -                    channel enumeration
```

FROM TO DURATION

```
FROM (VON)  TO (BIS)     : Query time number of last output "with time".
DURATION (DUR)           : Duration of time FROM .. TO in seconds.
  The two variables FROM and TO are set when an appropriate command is used
  with the extension '/' or '^'. DURATION is used to provide the length of
  the period between FROM and TO in seconds.
Call    :  FROM
Stack   :  - >>> <timeNumber>   <timeNumber> : seconds count from 1.1.1990
date/time <timeNumber> ------ Notes for the use of extensions / and ^ --------
 /      :   ^       : output with date/time "to"
 //     :   ^^      : output with date/time "from--to"
 ///    :   ^^^     : output with date/time "from"
 ////   :   ^^^^    : suppress output (FROM and TO are set nevertheless)
Modifikation of output defined by / or // or /// (/ allways preceeds ^):
 /^     : output time/date instead of date/time            17:33:56;12/31/93
 /^^    : output date/time, date in DBase format yyyymmdd  19931231;17:33:56
 /^^^   : output time/date, date in DBase format yyyymmdd  17:33:56;19931231
 /^^^^  : date/time sep.';' --> ' ' (valuable for MS-EXCEL) 12/31/93 17:33:56
 /^^^^^ : time/date sep.';' --> ' ' (  "       "       "  ) 17:33:56 12/31/93
```


H Programs HLIST HBREAK

```
H 0 .. H 19  :  Background H programs
H 0 .. H 18 are executed in the background in order. Run time errors may be
flagged with ERR. Output from the background programs is always sent to COM1
at the station on which the background program is running.
Call:           H <enumeration>  [=<string>]    max. line length: 128
ext:            + - . # $ ? !                   No execution during output
                H! *  lists all H's
- H without <index> == H 0,   H1..H19 == H 1..H 19,   H5! == H! 5
- Maximum number of linked P sub programs = 3  (H='.. P[P[P]]..')
- H? pushes the number (0..19, -1:Pause) of current H program on to the stack.
  H?? pushes on to the stack: 1= Focus H program, 0= Focus command line
- H 19 is the "print program" activated from the operating panel. H 19 only
  runs in the background when activated and not at other time.
HLIST  or  HLIST <enumeration> : lists H Programs (corresponds to H! *)
HBREAK : Interruption and 16s (HBREAK+  32s) pause in execution of H programs
```

HH MM SS DAY WDAY MON YEAR

```
Query of internal real time clock, result to the stack:
HH              : Hour
MM              : Minute
SS              : Second
DAY   (TAG)     : Day
WDAY  (WTAG)    : Weekday (1: Monday  .. 7: Sunday)
MON             : Month
YEAR  (JAHR)    : Year
Call:   HH              : Basis = System time
Stack:  - >>> <hourNumber>
Call:   HH .            : Basis = Second number from stack
Stack:  <secNumber> >>> <hourNumber>
ext:    !               : Output of number or with WTAG: name of day
                          or with MON: name of month
        -               : Divert output to string buffer
```

ID

```
ID (KENN) :  station ID as number on the stack   (set ID with SETID)
===========
Call      :  ID [<IDnumber>]            without <IDnumber>: current ID
Stack     :  -  >>>  <IDnumber>        A: 1, A1: 2,.. B: 11, C: 21,.. Z4: 255
ext       :  ! + . # %
Examples  :  z4:ID. !     ID! 21        ID. 21         ID# 21
             255          C:            C              21
Influence of Extension '&' on General Command Output:
=========================================================
 &       : output ID at start of line.          [Etot& 1    -->  A:Etot ....]
 &#      :   <ID>;<val1>
 &#.     :   <ID>;<val1>
 &.      :   <ID>;<vala>;<valb>;...
 &&      : output ID at start of line and before each output block.
 &&##    :   <ID>:<val1>;<ID>:<val2>;...
 &&#.    :   <ID>;<val1>;<ID>;<val2>;...
 &&..    :   <ID>;<vala>;<valb>;..;<ID>;<val2a>;<val2b>;..
 &&& &&&& : as & or && but with ID number.      [Etot&&& 1  -->  1:Etot ....]
```

IF ELSE ENDIF

```
IF ELSE ENDIF  :  Program branching commands
The IF command takes an element from the stack and rounds it (5/4). If it is
not equal to zero, the commands between IF and ELSE / ENDIF are processed. If
it is equal to zero, the optional section between ELSE and ENDIF is executed.
Call:   IF                      ELSE            ENDIF
Stack:  <condition> >>> -       - >>> -         - >>> -
Example of "condition met":
    1, IF, REM "This section is executed", ELSE , REM "but this is not"
Example of "condition NOT met":
    0, IF, REM 'not execute', ELSE , REM "execute", endif; rem Both
- Each sub program may have up to four levels.
- If there are no more commands after ELSE or ENDIF, these commands
  may be omitted.
- If the section between IF and ELSE is executed only once following the
  occurrence of a true condition, then use IFF (cf. IFF).
- For the use of IF with time comparison, cf. TIMECOMPARISON.
- EIF is the short from of ENDIF
```

IFF

```
IFF  .. ELSE ENDIF  :  Program branching command for executing once
The command IFF is used in cases where, unlike IF/ELSE/ENDIF, the section
between IF and ELSE or ELSE and ENDIF is only carried out ONCE following
the occurrence of a true condition.
An internal permanent flag (IFF) is kept automatically for each background
H program, as well as a temporary flag (IFF+), which is always initialised
during power on. In H programming, both flags are initialised. The IFF and
IFF+ commands can be used once in an H program (with its P sub programs).
Call:   IFF                     ELSE            ENDIF
Stack:  <condition> >>> -       - >>> -         - >>> -
ext:    +  (IFF+ flag is always initialised after power on)
        For the use of IFF with time comparison, cf. TIMECOMPARISON
Example:
  If +24V is present at input 8, the Etot of channels 1..4 are printed out
  once, and the time is printed out once if the voltage returns to 0V.
        H 10 = 'IN- 8, IFF, Etot 1..4, ELSE, time'
```

INDEX

```
INDEX   :  Generates an index for the readings list
Call:   INDEX *
Stack:  - >>> 'Index of first entry in chronological order'
Call:   INDEX **
Stack:  - >>> 'maximum possible number of entries'
Call:   INDEX <fromDate/fromTime> [<fromTime>]
Stack:  - >>> 'Index for search time'
ext:    +    : Index for search time - 1 (avoids overlap)
        //   : the search is carried out "from" the time instead of "to" it
Example:       INDEX 29.03 12:15, eint##/ 1..4 . *
- The validity of the INDEX number can be guaranteed for the command
  following the INDEX command (in the same line), regardless of whether a
  period transition has occurred in the meantime (validity up to 0.3 s).
- Period transitions during an EINT command do not affect data output.
```


INDIR

```
INDIR   : checks if an ID is present in the directory (DIR)
Call:   INDIR
Stack:  <IDno>  >>>  (1: <IDno> in DIR / 0: not available)
Call:   INDIR <IDno>
Stack:  -  >>>  (1: <IDno> in DIR / 0: not available)
Call:   INDIR*
Stack:  -  >>>  (1: "current" ID in DIR / 0: not available)
Example:
* Once station G3 is in the ECS LAN, "G3 in Network" is displayed for 10s
  on all LCDs:
    <A> H=' G3:indir*, iff, all, meld "G3 in Network" 10
* The ALL loop command is included for demonstration purposes:
    'ALL, ..... , NEXTA, ...'   corresponds to:
    '1,255,fori,i,indir .,if,i,an:, ..... , endif,nexti, zz:, ...'
```

INPUT

```
INPUT (IN)  :  Reads in input status
Call:   IN <enumeration>
Stack:  - >>> (1: 24V present, 0: 0V present)
ext:    + - . # %
```

INTERVAL (Period)

```
INTERVAL (ITV)  : Synchronising period
Call:   INTERVAL [=<duration>]                  <duration> : 10s .. 999h
or:     INTERVAL [=<number>__<unit>]            <unit> : s | m | h
                                    (without <unit> : seconds)
Stack:  - >>> <durationinSeconds>
ext:    + - . # | %
Example: INTERVAL =15 M   oder   INTERVAL = 35s
- Parts of hours or minutes must be converted: 1h30 --> 90m or 5400 s. For
  output, the largest unit of time is always used.
- Cf. also INTERVALSOURCE and SYNC
```

INTERVALSOURCE TARIFFSOURCE

```
INTERVALSOURCE  (IQ)    : Source for generating synchronising period
TARIFFSOURCE    (TQ)    : Source for generating current tariff T1 or T2
Call:   IntervalSource [=<source>]
Stack:  - >>> <sourceNumber>
ext:    + - . # $ %
                <source>                <sourceNumber>
INTERVAL  :     1 .. 24                 1 .. 24
                C  |  Channel           23          (preset for channel 23)
                T  |  Time              99
                P  |  Program           100
TARIFF    :     1 .. 24                 1 .. 24
                C  |  Channel           24          (preset for channel 24)
                P  |  Program           100
Example : Tariff of channel 17  :       TARIFFsource = 17
          Time-dependent period :       IQ = T
```

KEY

```
KEY (TT)  :  Transfer keystrokes to the operating panel
Call:   KEY <keySequence>
Stack:  - >>> -
Elements of <keySequence>, maximum length, 20 elements:
        1 .. 4  :  F1 .. F4
        + : up arrow
        - : down arrow
        m : Menu
        s : Setup  (like "press Menu key for 1s")
        u : switch (like "press up and down arrow keys together")
        l : enter Erase menu or similiar function
        x : set basic status (normal display with Etot, channel 1)
Example   : KEY x++++4
  x sets the operating panel to normal display, 4 x '+' moves to channel 5.
  4 stands for F4, ie. switch to Pmom.
```

LANGUAGE

```
LANGUAGE :  sets the dialog language
==========
Call     :  LANGUAGE [=<country>]       Query language setting: LANGUAGE
ext      :  + - # . $ %                  <country> := {English German Deutsch}
- set german:   LANGUAGE = German       or      LANGUAGE = 1
- set english:  LANGUAGE = english      or      LANGUAGE = 2
- only the first letter of <country> must be specified.
DATEFORMAT (DATEFOR) : sets the date format for all date outputs
======================
Call     :  DATEFORMAT [=<dformat>]     Query actual dateformat: DATEFORMAT
ext      :  + - # . $ %

                                        commands with date output: override
- possible values for <dformat>:        actual setting with extension -
    dd.mm.yy (tt.mm.jj)  -->  31.12.93  :  ~
    mm/dd/yy (mm/tt/jj)  -->  12/31/93  :  --
    mm-dd-yy (mm-tt-jj)  -->  12-31-93  :  ~~~
  only the first 2 or 3 letters of <dformat> must be specified (dd mm/ mm-).
```

LEVEL

```
LEVEL   : Sets the input sensitivity.
          The max. low input level can be set to values in the range of
          approximately 3 volts (Lo) to 5.5 volts (Hi).
Call    :  LEVEL [=<value>]
Stack   :  -  >>>  <value>      (when reading)
ext     :  + - %                <value> :  0 (Lo) ... 3 (Hi), default: 1
- 50Hz interferences on the inputs can be avoided with:
  LEVEL = 3 (Hi)  [Pulse > 30ms].
- the level setting is for all inputs together.
- 74HCxxx logic level adaption with LEVEL = 0 or LEVEL = 1.
```


MELD MELD2

```
MELD  :  Displays a message in the LCD for a defined period.
         The display is interrupted as soon as a key is pressed on the panel.
Call:          MELD <string>  [<length_in_secs>]  (max. 60s, default=5s)
Stack:         - >>> -
Example:       MELD "! Motor faulty !"          --->    "! Motor faulty !"
                                                         "****************"

MELD2 :  Displays a two line message in the LCD.
Call:          MELD2 <string> [<string> <length_in_secs>]
Stack:         - >>> -
Example:       2,1,MELD2 "%!. Line""%!. Line"  --->    "1. Line         "
                                                       "2. Line         "
- <length_in_secs> == 99  :  wait for unlimited period.
- If an empty string "" is given in MELD2, the corresponding line in the
  display is left unchanged (except if the extension is *).
- Ext. - copies message to the buffer.
```

P Programs, PLIST

```
P 0 .. P 19     : Programs P0 .. P19
Call:           P <enumeration>  [=<string>]    max. line length: 128
ext:            + - . # $ ! ? @                 No execution during output
- P without <index> == P 0,   P1 .. P19 == P 1 .. P 19,   P5! == P! 5
- Maximum number of linked P sub programs = 3  ( P[P[P]] )
- P? pushes the number (0..19),-1: none) of current P program on to the stack.
- Unused P programs can be used as string memory.
- P@ executes the strings stored in the buffer as a program.
- Use RETURN and EXIT for premature termination of the sub program.
* Calling a P program by user defineable name: cf. REM
* List all P's:                         P! *          or      PLIST
* Copy P 7 to P 13:                     p- 7,P 13=$   or      p7-,p13=$
* Copy all P's to station B:            fori 0..19, i,p- ., i,B:p .=$
PLIST  or  PLIST <enumeration>  :  lists P programs (corresponds to P! *)
```

PASSWORD

```
PASSWORD : Secure the access via RS232 to the station / ECS-LAN.
==========
 A master user (user 1) and four users (2..5) can use different passwords
 (numbers in the range 1..999999999). The master user sets the access rights
 of the individual users, the users themselves may change their password (as
 long as the old password is known). The master's access right is used, when
 no user is logged in or if the timeout has elapsed. The master always got all
 rights (=5). Initial all passwords are 0, only the master can change 0-pass-
 words. To clear all passwords: Master password = 0.
       !! ATTENTION : Wrong Inputs may result in time barriers !!
LOGIN.....log in with a password
  call  : LOGIN <user> <password>        <user>   :     1=Master, 2..5=user
  stack : - >>> -                        <password> :   1..999999999
LOGOUT.....log out
WHOAMI.....print out user number and rights
PASSWORD.....Setup of the station passwords and rights (access levels)
               <pw_old>  : old password or Master-pw as access grant
               <pw_new>  : new password (must be repeated)
               <timeout> : in minutes, 0=no timeout
  call:   PASSWORD  <user> <pw_old> = <pw_new> <pw_new> <rights> <timeout>
          PASSWORD  <user> <pw_old> = <pw_new> <pw_new>
          PASSWORD* <user> <pw_old> = <rights> <timeout>
  stack:  - >>> -
          PASSWORD              --> output of actual user
          PASSWORD  <user>      --> output of the rights from <user>
  stack:  - >>> <timeout> <rights> <user>
-----Example----               note the order!
Master enters:
  PASSWORD 1 0=123 123 0 5    : Master-pw=123, timeout=5m, 0-user-rights=0
  PASSWORD 2 123=222 222 3 10 : user-2-pw=222, timeout=10m, rights=3
  PASSWORD* 2 123=2 5         : change of rights=2 and timeout=5m
User changes password:
  PASSWORD 2 222=2121 2121    : password change
user loggs in:
  LOGIN 2 2121
Master clears all passwords:
  PASSWORD 1 123=0 0
Rights:         ------local------       -----ECS-LAN-----       notation
==============  read    write          read    write
        0       -       -              -       -              [-- L:--]
        1       yes     -              -       -              [r- L:--]
        2       yes     -              yes     -              [r- L:r-]
        3       yes     yes            -       -              [rw L:--]
        4       yes     yes            yes     -              [rw L:r-]
        5       yes     yes            yes     yes            [rw L:rw]
```


PAUSE

```
PAUSE (PP) : Pause in seconds
The execution of the program is suspended for n seconds - fractions of
seconds may also be specified. The effective waiting time is always a
multiple of 100ms.
Call:   PAUSE                    or      PAUSE <value>
Stack:  <value>  >>>  -                  - >>> -
Notes:
- maximum length of pause = 20s
- Numbers for n > 20 are regarded as milliseconds.
  Example: 'Pause 2.2' corresponds to Pause 2200', program execution is
  suspended for 2.2 seconds.
```

PFACTOR

```
PFACTOR : Factor used in calculating power from energy per period.
This factor can be used to adjust the time reference to the calculation
of power.
- Normally, the hourly reference is used (kWh to kW) : Pfactor = 3600
- Where seconds reference is required (Ws to W) : Pfactor = 1
Formula to calculate power p from energy E and period dt:
        P = E * Pfactor / dt
Call:   PFACTOR <enumeration> [=<value>]
Stack:  - >>> <value>      (when reading)
ext:    + - # . %
```

PRINT

```
! (PRINT)  : Output command
!         : Stack output function - pops a number and prints it out.
! ...     : Output function  ! [<par> [__<par> [__<par>]]]
            Example:  fix 3,5,! "Value = " . " kg"    --->    Value = 5.000 kg
!$        : Buffer output.
!?        : String comparison (Argument <--> buffer).
            Stack here:  -  >>>  {1|0}   : Equality = 1, Different = 0
!_        : Output of a line of 78 underlined '_'
!!        : Output function "one empty line"
- The string to be printed out may contain formatting instructions similar to
  those used in the printf() function in the 'C' language, cf. PRINTFORMAT.
- For examples, cf. PRINTEXAMPLES.
- Ext. * is used to copy the entire output into the buffer
  (max. 128 characters).
- Ext. - [suppress output] diverts all output to the buffer.
- Ext. ? for case insensitive, Ext. ?? for case sensitive comparison.
```

PRINT Formatting

```
Output formatting:                (similar to format commands in printf() in 'C')
==================                %[-][0][<width>][.<fix>]#
command:    Output:                 |  |     |        |   |
%!        : Number (from Stack)     |  |     |        |   command
%x   %X   : Number in HEX ( " )     |  |     |        decimal places
%$   %s   : Buffer reference        |  |     output width (max.99)
%<b>c<Z>  : <b>-times character <Z>  |  0 : leading zeros
%%        : %                       - : left aligned
----------  For other options, cf. PRINTMODI, for examples cf. PRINTEXAMPLES.
----------  Some commands called with % format output according to the
            format string (1st parameter). In this case:
%g   %G   : Device ID, %g : letter ID 'A1', %G : ID number
%f        : Function name (for Etot: 'Etot').
%k        : Channel number.
%v   %V   : Channel number/Code, fixed format: (00), 01 .. 24 , V1 .. V8.
%i        : Index (numerical), so for EDay-3 = 3.
%w        : Principle value (numerical) of comand (so for Etot = the energy).
%e        : Units string of the command.
%n        :  Name string of the command (Channel name)
```

PRINT Modifications

```
Stack Output / Manipulation in output formatting:
   %!           : n >>> - ; print n       %#           : n >>> n ; print n
   %<    %n<    : DROP [<n=width>]        %>    %n>    : DUP [<n=width>]
   %n^          : PICK                    %~           : SWAP
   %p00  %p19   : Contents (P i)          %h00  %h19   : Contents (H i)
   %a00  %a19   : Contents (A i)          %&a00 %&&a00 : push+print/only push
Time Output:      %t..     : Time number of stack, stack remains (n >>> n)
   %///t..      : Time                    %&..t %&&..t : push+print/only push
   %/t..        : FROM                    %//t..       : TO
   %tt   %TT    : Time$ / Time-Date$      %dd   %DD    : Date$ / Date-Time$
   %th          : Hour     (number)       %dt %dw %dT  : Day/Weekday(numb./$)
   %tm          : Minute   (number)       %dm   %dM    : Month(number / $)
   %ts          : Seconds  (number)       %dy   %dY    : Year (number / 19..)
- ID input for %p %h %a   :   %p<ID>:01    (%pA1:01)
- %p leaves contents of p intact, %P converts contents (cf. STRINGS)
- %@p %@h %@s %@$ : string is used as a formatstring (1 layer)
- %$ leaves buffer string intact, %s converts buffer '\'-codes (cf. STRINGS)
- %$ with fixed point input (eg: %.3$) : omit n character from start of string
```

PRINT Examples

```
* Output stack data
    12, 34, ! "n1 = %04!, n2 = %.3!"       -->  n1 = 0034, n2 = 12.000
    1,2.3,4, ! "%3^%06.3! %>%!,%!"         -->  002.000 4,4
* Formatting an output command (channel v1 = Heatroom)
    Etot% "%g:%f from %4n = %w %e" V1      -->  A:Etot from room = 1234.12 kWh
    Etot% "[%G]%f from %-4n = %.0w %e" V1 -->  [1]Etot from Heat = 1234 kWh
* Time output (%dT : lower case, %DT : upper case day/month string output)
    ! "Today is %///&DT,  %dt. %dM %dY"
    -->  Today is Wednesday, 3. november 1993
* HEX output
    %x    : 43981,! 'In Hex: %05x'         -->  In Hex: 0abcd
    %X    : 1997, ! 'In HEX: %X'           -->  In HEX: 7CD
```


POWERFAIL

```
POWERFAIL (PWR) : List of interruptions to power supply (max. 32 entries)
Call:   POWERFAIL <enumeration> [=0]
Stack:  - >>> -
ext:    + - # . / ^ * @ %
- List of all interruptions              :  PWR *
  (starting with the first power failure)
- Reverse order of output with Ext. *    :  PWR* *
- Erase from index 7                     :  PWR 7=0
- Duration of last interruption [s]      :  PWR-,DUR,!
- FROM and TO are always set (even without extension /).
Call:   POWERFAIL@      Special case: provides time between <PowerON>..<now>
Stack:  - >>> -                       [cf. also POWERON]
ext:    + - # . / ^
- provides the length of time [s] since PowerON:  PWR@- 0,DUR,!
```

POWERON

```
POWERON (PWRO) : Power On Time since last Power On or reset
Call    :  POWERON
Stack   :  -  >>>  <time_in_seconds>
ext     :  + - # . / ^ $
- FROM and TO are always set (even without extension /).
```

RELAY RELAYMODE

```
RELAY (REL)  :  Switch relay output on/off. 4 outputs (change over contacts)
                1 .. 4 are available. '1': active relay, '0': inactive relay.
Call    :  REL <enumeration> [= (1|0)]
Stack   :  -  >>>  (1|0)   (for reading)
ext     :  + - . # * $ %        .
RELAYMODE (RELM) : Set mode of operation of relay outputs.
                        0 : Relay permanently OFF
                        1 : Relay permanently ON
                        2 : Relay controlled by program (default)
Call    :  RELM <enumeration> [= <mode>]
Stack   :  -  >>>  <mode>   (for reading)
ext     :  + - . # * %
- Ext. * always suppresses the optional relay name (cf. RELAYNAME).
- STATUS prints:  'p' : OFF per program          'P' : ON per program
                  '_' : always OFF (Mode 0)     '+' : always ON (Mode 1)
```

RELAYNAME

```
RELAYNAME (RELN)  : Then name of a relay provided for better identification.
Call:   RELN <enumeration> [= <name>]
Stack:  -  >>>  -
ext:    + - . # %
- max. length of <name> = 8 characters.
- The relay name is very usefull in conjunction with the
  search function (cf. FINDER)
- Erase an unused name by assigning an empty string  :  RELN 2 = ""
```

REM

```
REM (@) : Insert remark
Example :       REM "Averages Program"
                @ give_upper_value
                REM one two three       (one string per <par>)
- If a line starts with '#' (the first NON space), the line is completely
  ignored. ECSoft 2 uses this property when it transfers script files for
  the inclusion of meta commands (commands processed by ECSoft 2 on the PC).
```

SETID

```
setID (setKENN) : Sets the ID
Call:           setID = (A, A1..A9, .. Z, Z1..Z4):
Example:        setID=U1:
```

STATION GROUP

```
STATION  :  Station name
GROUP    :  Optional group name (cannot be used from the operating panel)
Call:   STATION [=<name>]
ext:    + - # . $ %
- max. length of <name> = 8 characters.
- Cf. CHANNEL for available characters.
```

STATUS STATCHECK STATREL

```
STATUS (STAT)...:  Output of the device status, similiar to the
       Call    :   STATUS                LCD device status output
       Stack   :   -  >>>  -
       ext     :   + $ ##
STAT24V.........:  Status of the 24V power supply output
       Call    :   STAT24V               (no output)
       Stack   :   -  >>>  (1|0)         1: OK, 0: Error
STATBAT.........:  Status of the lithium battery (for Memory backup)
       Call    :   STATBAT               (no output)
       Stack   :   -  >>>  (1|0)         1: OK, 0: Error
STATREL.........:  Status of the status relay
       Call    :   STATREL               (no output)
       Stack   :   -  >>>  (1|0)         1: OK (Relay ON), 0: Error (OFF)
STATCHECK.......:  setting or reading of the extended error reporting via
                   the status relay (coupled or NOT coupled with the status
                   of the 24V power supply output or the lithium battery).
       Call:       STATCHECK [=<value>]
       Stack   :   -  >>>  <value>    (for reading)     0: NOT coupled,
       ext     :   + - %                                1: coupled
```


SYNC

```
SYNC    :  Generate period transition; query period status.
Call:   SYNC =              : the current period is ended,
Stack:  - >>> -               Requires: INTERVALSOURCE = PROG
ext:    +                   : the current period is ended,
                              regardless of interval source setting!
Call:   SYNC              : queries whether period transition has been reached:
Stack:  - >>> <SyncFlag>     <SyncFlag> == 0 : no
                             <SyncFlag> == 1 : for 5s after transition
                            <SyncFlag> == 2 : Sync command currently executing
Call:   SYNC*
Stack:  <NumberOfPeriodsSincePowerOn>          (max.255)
Call:   SYNC**
Stack:  <TotalPeriods>                         (max.65535)
Call:   SYNC/
Stack:  <CurrentPeriodLengthInSeconds>
- FROM and TO are always set in accordance with the current interval.
```

System functions

System-Funktionen

```
sysRESET........:  performs CPU Reset   (similiar to PowerOn Reset)
Call            :  sysRESET =0
sysTEST.........:  Tests some system functions and prints result
Call            :  sysTEST [<number>] [=0]
Stack           :  - >>> ESCC2(L)  ESCC2(R)
ext             :  - . &
- SYSTEST <number> checks the ECS-LAN. 64 data bytes are exchanged
  <number>-times for test purposes with the referenced station (32 bytes to
  and 32 bytes back). When there is no other ECS-LAN traffic the measured
  datarate is in some manner equal to the transmission line quality.
Example:
  <A>  B:systest 100
  6400 bytes are exchanged between station A: and B:, output is the time
  period and the datarate.
IMPORTANT: Every ECS-LAN segment between both devices reduces the datarate
  (time period * n). Standard value (n=1, 62K5 Bd): 2000..2500 byte/s.
```

TARIFF

```
TARIFF : Query or set current tariff
Call:   TARIFF [=<tariff>]      <tariff> = (1|2); '1':Tariff 1, '2':Tariff 2
Stack:  - >>> <tariff> (for reading)
ext:    + - * %
- With extension * <tariff> is as follows =(0|1); '0':Tariff 1, '1':Tariff 2
  Example (with ext *):
        T2 applied at the weekend (Saturday and Sunday):
        H 10= 'wtag,6,>=,tarif* =.'
```

TIME DATE

```
TIME (ZEIT )   DATE (DATUM)   :  Set or query system time and date.
       Set:                       Query:                    Show time number:
Call:   TIME = <timeString>       TIME                      TIME .
Stack:  - >>> -                   - >>> <timeNumber>        <timeNumber> >>> -
ext:    + - * // %
- Format of <timeString>: 12:36:00 or 2h15
- Format of <dateString>: 17.03.93 or 26.02     [date formats cf. DATEFORMAT]
- <timeNumber> is the seconds count since 1.1.1990
- TIME// always displays the time and date,
  and 'TIME//=30.11. 11h' can also be used to set time and date together.
- TIME* pushes (<timeNumber>.<secondsFraction>)
- Time comparisons cf. TIMECOMPARISON.
- Time measurements: [ Command Pair TM / TMD  (== ZM / ZMD) ]
    'tm,  <Block> , tmd,!'    displays length of <Block>, stack must conform!
    'a=t, <Block> ,a,tmd,!'   displays length the <Block>, stack unimportant.
    Resolution: seconds with 1/100s.
System Time Comparisons:
Call:   IF <date_or_timeString> [<timeString>]      (IFF also possible)
Stack:  - >>> -
Example:
  h 10= 'IF 17.3 xh10.xx, rel 1=1,else, rel 1=0'
  On 17.3, at ten minutes past each hour, relay 1 is activated for one minute.
```

Trigonometric Functions

```
SIN COS ASIN ACOS DEG RAD SQRT EXP LOG  :  Mathematical Functions
          Stack:
SIN  :  x  >>>  sin(x)          Radian measure is basis
COS  :  x  >>>  cos(x)
ASIN :  x  >>>  asin(x)         Inverse function of SIN
ACOS :  x  >>>  acos(x)
DEG  :  x  >>>  ((x/pi)*180)    Convert radian measure to degrees
RAD  :  x  >>>  ((x/180)*pi)    Convert degrees to radian measure
SQRT :  x  >>>  Square root(x)
EXP  :  x  >>>  (e to the power of x)
LOG  :  x  >>>  LOGe(x)
PI   :  -  >>>  pi              3.141592653589793
```

Urat Irat MCONST PULSE EDGE ONOFF STARTSTOP CHANNELFIX

```
ONOFF   STARTSTOP (STSP)   CHANNELFIX (CFIX)   :  channel specific parameters
MCONST:          Meter constants <real>
Urat:            Voltage transformation ratio <real>
Irat:            Current transformation ratio <real>
PULSE:           Pulse duration in ms (10..2550 == 0,01s..2.55s)
EDGE:            Current pulse edge or tariff allocation (binary input):
                 1: __-- Change 0V >>> 24V (+)  or  24V --> Tariff 2
                 0: --__ Change 24V >>> 0V (-)  or  24V --> Tariff 1
ONOFF:           Switch channel ON/OFF. 1: ON, 0: OFF. Enumerations '*'
                 only apply to channels which are ON.
STARTSTOP:       Control pulse counting of the channel:  1: START, 0: STOP
CHANNELFIX:      Decimal places for output (0: 0, 1: 0.0, 2: 0.00 or 3: 0.000)
Call:   MCONST <enumeration> [=<assignment>]
ext:    + - . # %
Stack:  - >>> <value>  (for reading)
```


VER

```
VER : Details of current software version
Call:   VER
Stack:  -  >>>  <versionNumber>
Ext:    + - . $ #
```

VSUM VIRT

```
VSUM VIRT  :  Description function used for creating virtual channels (in a
              background H program) with absolute totals ("fixed link").
              Differential totals ("differential links") are handled with the
              command dVSUM and dVIRT.
VSUM:   generates totals for enumerated channels and stores them in
Call:   VSUM <enumeration> [<factor>]            intermediate registers
Stack:  - >>> -
VIRT:   Assigns the intermediate register totals to the virtual channel
Call:   VIRT <enumeration> [<factor>] =
Stack:  - >>> -
Example:     H 1= 'b:VSUM 1..5+8 2.4, c:VSUM 4 -1, d:VIRT V6='
             (Channels from B: * 2.4 plus channels from C: * -1 to form D:V6)
- The virtual channels are direct images of the physical channels. VSUM and
  VIRT create totals of all channel registers, EXCEPT the maximum registers
  and period readings entries from Eint-2.
- VIRT is not limited to channels V1..V8 (which contain no maximum registers).
```

System Time Comparisons

```
Call:   IF <date_or_timeString> [<timeString>]      (IFF also possible)
Stack:  - >>> -
Example:
  h 10= 'IF 17.3 xh10.xx, rel 1=1,else, rel 1=0'
  On 17.3, at ten minutes past each hour, relay 1 is activated for one minute.
```

German - English Cross Reference

ECS operating system V1.15 provides the feature that almost every ECL command can be called by german or english command name - independent of the selected dialog language. The online help system also accepts search terms in both languages.
For the following ECL commands exist different names in both languages:

German	English
AUFZ	ENUM
BIS	TO
DATUM	DATE
DATUMFORMAT	DATEFORMAT
DAUER	DURATIO (DUR)
EEINH	EUNIT
EGES...	ETOT...
EINAUS	ONOFF
EMAXTAG, ..JAHR	EMAXDAY, ..YEAR
ERAKAN	ERACHAN
ERALIS	ERALIST
FLANKE	EDGE
INTERVALL	INTERVAL
INTERVALLQUELLE	INTERVALSOURCE
JAHR	YEAR
KANAL	CHANNE (CHAN)
KANALFIX	CHANNELFIX (CFIX)
KENN	ID
KOSTFAKi	COSTFACi
PASSWORT	PASSWORD

German	English
PEGEL	LEVEL
PFAKTOR	PFACTOR
PMAXTAG, ..JAHR	PMAXDAY, ..YEAR
PULS	PULSE
PEINH	PUNIT
RELAIS	RELAY
RELAISMODE	RELAYMODE
RELAISNAME	RELAYNAME
SETKENN	SETID
SPRACHE	LANGUAGE
TAG	DAY
TARIF	TARIFF
TARIFQUELLE	TARIFFSOURCE
TEINH	TUNIT
TASTE	KEY
VON	FROM
WTAG	WDAY
ZEIT	TIME
ZKONST	MCONST

Dialog Language and Date Format

LANGUAGE : sets the dialog language

```
==========
Call      :  LANGUAGE [=<country>]        Query language setting: LANGUAGE
ext       :  + - # . $ %                  <country> := {English German Deutsch}
- set german:   LANGUAGE = German        or       LANGUAGE = 1
- set english:  LANGUAGE = english       or       LANGUAGE = 2
- only the first letter of <country> must be specified.
```

DATEFORMAT (DATEFOR) : sets the date format for all date outputs

```
======================
Call     :  DATEFORMAT [=<dformat>]      Query actual dateformat: DATEFORMAT
ext      :  + - # . $ %
                                         commands with date output: override
- possible values for <dformat>:         actual setting with extension ~
    dd.mm.yy (tt.mm.jj)  -->  31.12.93  :  ~
    mm/dd/yy (mm/tt/jj)  -->  12/31/93  :  ~~
    mm-dd-yy (mm-tt-jj)  -->  12-31-93  :  ~~~
  only the first 2 or 3 letters of <dformat> must be specified {dd mm/ mm-}.
```


Annex

ECS/PC Connecting Cable

The symmetrical connecting cable ECS/PC, of the type 'null-modem-cable' is supplied in lengths of 2 metres, and is provided with a 9 pin Sub-Min-D9 plug at each end. It can be used to connect a PC, a terminal or an RS232 printer with a U1600 summation station.

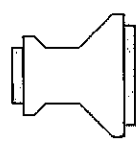
Pin allocation of ECS/PC connecting cable::

Sub-Min-D9 plug female			Sub-Min-D9 plug female	
Function	Pin#		Pin#	Function
DCD	1		4	DTR
RXD	2		3	TXD
TXD	3		2	RXD
DTR	4		1+6	DCD+DSR
Signal-Ground	5		5	Signal-Ground
DSR	6		4	DTR
RTS	7		8	CTS
CTS	8		7	RTS
RI	9		9	RI
Shield			Shield	

An adaptor is required if the PC, terminal or RS232 printer can only use a Sub Min D25 plug.

D9 - D25 Adaptor::

Sub-Min-D9 plug male			Sub-Min-D9 plug male	
Function	Pin#		Pin#	Function
DCD	1		8	DCD
RXD	2		3	RXD
TXD	3		2	TXD
DTR	4		20	DTR
Signal-Ground	5		7	Signal-Ground
DSR	6		6	DSR
RTS	7		4	RTS
CTS	8		5	CTS
RI	9		22	RI
Shield			Shield	



Contact Allocation

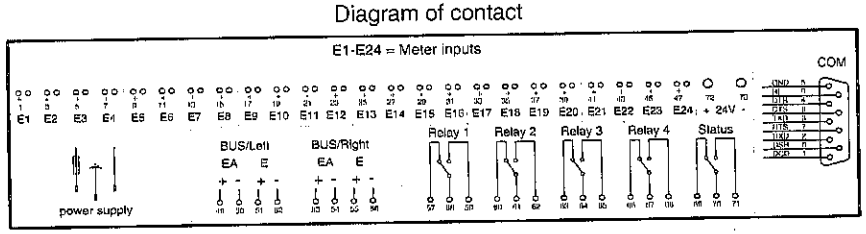
KL.#	Function	Label
1	Input 1	+
2	Input 1	-
3	Input 2	+
4	Input 2	-
5	Input 3	+
6	Input 3	-
7	Input 4	+
8	Input 4	-
9	Input 5	+
10	Input 5	-
11	Input 6	+
12	Input 6	-
13	Input 7	+
14	Input 7	-
15	Input 8	+
16	Input 8	-
17	Input 9	+
18	Input 9	-
19	Input 10	+
20	Input 10	-
21	Input 11	+
22	Input 11	-
23	Input 12	+
24	Input 12	-

KL.#	Function	Label
25	Input 13	+
26	Input 13	-
27	Input 14	+
28	Input 14	-
29	Input 15	+
30	Input 15	-
31	Input 16	+
32	Input 16	-
33	Input 17	+
34	Input 17	-
35	Input 18	+
36	Input 18	-
37	Input 19	+
38	Input 19	-
39	Input 20	+
40	Input 20	-
41	Input 21	+
42	Input 21	-
43	Input 22	+
44	Input 22	-
45	Input 23	+
46	Input 23	-
47	Input 24	+
48	Input 24	-

KL.#	Function	Label
49	BUS-Left	EA+
50	BUS-Left	EA-
51	BUS-Left	E+
52	BUS-Left	E-
53	BUS-Right	EA+
54	BUS-Right	EA-
55	BUS-Right	E+
56	BUS-Right	E-
57	Relay 1	Ö
58	Relay 1	W
59	Relay 1	Sch
60	Relay 2	Ö
61	Relay 2	W
62	Relay 2	Sch
63	Relay 3	Ö
64	Relay 3	W
65	Relay 3	Sch
66	Relay 4	Ö
67	Relay 4	W
68	Relay 4	Sch
69	Status-Relay	Ö
70	Status-Relay	W
71	Status-Relay	Sch
72	Meter power supply	+24V
73	Meter power supply	0V

Note: The meter power supply delivers 24V DC, max o.4 A (short circuit proof)

Diagram of contact

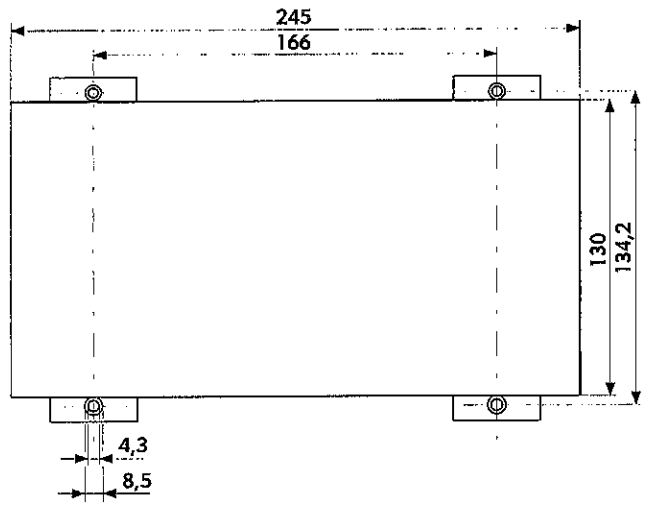


Pin#	Function:
1	DCD
2	RXD
3	TXD
4	DTR
5	Signal-Ground
6	DSR
7	RTS
8	CTS
9	RI

Pin assignment of Sub-Min-D9 connecting plug (male)

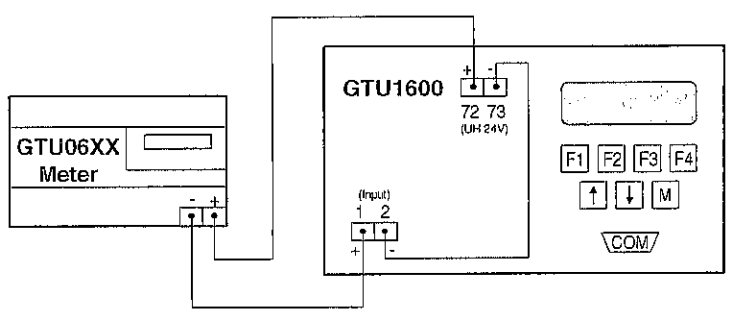
Installation Instructions

Drilling Template



Units: mm

Meter Connection Diagram (Meter GTU06XX to GTU1600)



Technical Characteristics

Dimensions:
245mm * 130mm * 80mm
Weight approx. 2.3 kg

Installation:
Assembly on tracks or mounting panels

Operating Temperature Range:
-10°C...+50°C

Power Supply:
AC/DC: 85V...250V (45..65Hz)Power consumption 19VA
DC: 20V...80V Power consumption 15W (Option)

Meter Power Supply:
The summation station supplies the meters with 24V DC, max. 0.4A (short circuit proof)

Meter Input:
24 optically-separated input, conforming to S0 standard

Alarm Outputs:
4 user-programmable relay outputs (changeover contacts)
1 relay output (changeover contact) for device status
Breaking capacity 50V, 0.5A

ECS-LAN:
Dual interface for connection of summation stations in multimaster network
- ❑ Electrical interface RS485 (optically separated) with 2 wire or 4 wire connection
- ❑ Protocol: based on SDLC/HDLC, multimaster. Reliable transmission with hamming distance 4
- ❑ Topology: open ring, bus, or mixture of ring and bus
- ❑ Max. number of stations in one ECS LAN: 255
- ❑ Max. number of stations in one BUS segment: 16 (32)
- ❑ Max. loop resistance of transmission line in BUS operation: 100 Ohm
- ❑ Terminating resistors: integrated, switched on and off via operating panel

RS232:
Serial communications interface, optically separated. Connection possible with:
- ❑ PC (host)
- ❑ Terminal
- ❑ Printer
- ❑ Modem

Memory Backup Battery:
A laser welded lithium battery and a high reliability battery management IC both guarantee that data is kept secure, and the operation of the real time clock is maintained for 10 years after interruption of the power supply (@50 C).

Supply Specification

- ❑ Summation station U1600
- ❑ Network cable 1.5m
- ❑ Operating instructions

Basic Configuration

The factory settings, and the settings after MASTER-RESET, are as follows:

Parameter	command	value
Station name	STATION	U1600
Group name	GROUP	ECS
ID*	SETID	A
Period (Interval)	INTERVAL	15 minutes
Interval source	IQ	Time
Format	FORMAT	Channel 1..32
Tariff source	TQ	Channel 24
LCD-contrast	-	5
Cost factor tariff 1	COSTFAC1	0,15
Cost factor tariff 2	COSTFAC2	0,10
Tariff unit	TUNIT	DM
Tariff fixed point	TFIX	2
Baudrate*	-	9600
Parity*	-	off
Handshake*	-	Xon/Xoff
ECS-LAN 2-wire- / 4-wire connection*	-	BL:2-wire, BR:2-wire
ECS-LAN terminating resistor*	-	BL: ON, BR: ON
ECS-LAN baudrate	-	BL: 62.5K, BR: 62.5K
Status relay extended error reporting	STATCHECK	1 (coupled)
Channel name	CHANNEL	Chan-x
Meter constants	MCONST	100
Voltage transformation ratio (U rat)	URAT	1
Current transformation ratio (I rat)	IRAT	1
P calculating factor	PFACTOR	3600
Energy unit	EUNIT	kWh
Power unit	PUNIT	kW
(Trigger) Edge	EDGE	1 == '+'
Pulse duration	PULSE	10ms
ON/OFF function	ONOFF	1
Start/Stop function	STARTSTOP	1
channel fixed point	KFIX	2
Relay mode	RELM	2 (controlled by program)
Background H programs	H 1	'vsum 1..8,virt V1='
	H 2	'vsum 9..16,virt V2='
	H 3	'vsum 17..24,virt V3='
	H 19	'Etot *,!!,Pmom *,!!'

* : these parameters are not changed when performing a MASTER-RESET.

PC Setup Software ECSoft 2

The software package **ECSoft 2** can be used to configurate and program individual summation stations. The software runs under DOS, the operating is quite similar to MS-WINDOWS.

Device Setup

All device parameters can be viewed and edited in clearly laid out lists.

Terminal

The terminal emulation allows communication with the summation station using the most direct route. A good knowledge of the commands is indispensable, however.

Panel

The front panel of each summation station connected to the ECS LAN can be shown on screen, allowing remote control of the device in question (in precisely the same way as using the device keyboard). Up to six panels can be loaded at one time.

Script Transfer, ECS-BASIC

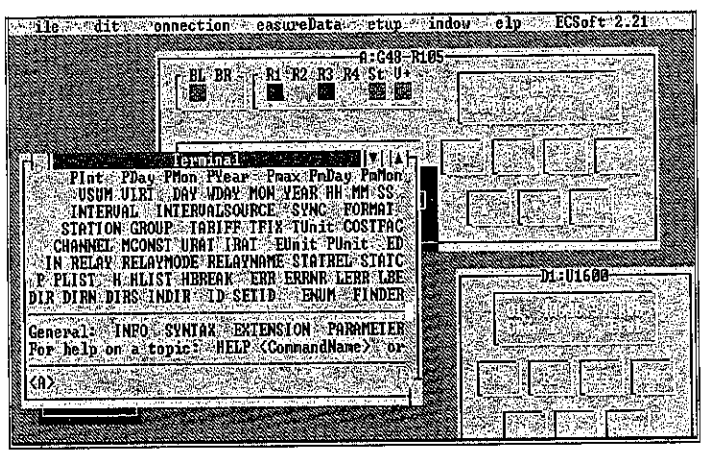
Command sequences can be stored as so called script files, and then transferred complete to the summation station. The transfer is controlled by the build-in ECS-BASIC interpreter. An integrated editor is provided to create and edit these script files.

ASCII Database

The data of all summation stations connected to the ECS LAN can be automatically stored in ASCII database files. Periodic update of these files is performed automatically.

Modem Support

Far ECS summation stations can be remote controlled via a modem connection. The connection is build up automatically or manually by selecting from a dial list.



Index

- Notes -

An example for optimal use of energies:

Energy-saving, powerful and beautiful. These are the „technical data" of the painted lady (Vanessa cardui LINNAEUS, 1758), as short and poignant as possible. In quite a unique manner, the painted lady with a weight of just some milligrams is able to generate a maximum amount of energy and to use it in such an economical way that it can perform migrations of several thousands of kilometers. „Migrations" are the yearly trips of insects which are far less known than the migrations of many of our birds. Painted ladies of the first generation hatch out in early spring from the pupa on the North-African Mediterranean shores and start their flight into the moderate climates of Europe. To the north of the Alps, they produce a summer generation whose descendents will usually not survive the continental winter. In autumn, they therefore try to cross the Alps to the south. By the way: the painted lady´s caterpillar lives on thistles, nettles and other weeds. Without these weeds and without the painted lady´s controlled use of energy, we´d miss a nice piece of natur.

Intelligent use of energy makes sense. Not only for butterflies; it makes sense always and everywhere.