Can Diplay DMA-15 lococube
Art. No. 0040-0015

MANUAL

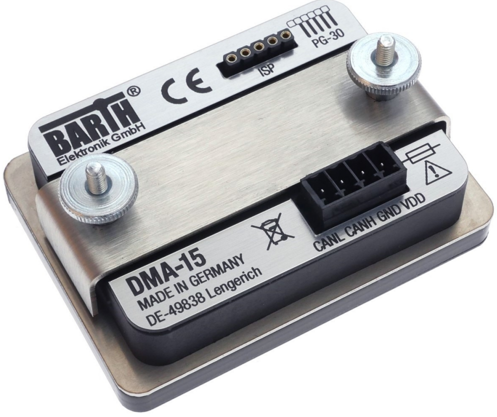


TABLE OF CONTENT

SAFETY INSTRUCTIONS

This manual contains safety instruction that should be followed to ensure your own personal safety, as well as the product's, and any connected equipment. These instructions are highlighted in the manual by a warning sign and are marked as follows according to the level of danger:

Only qualified personnel should be allowed to install and work on thisequipment. Qualified is defined as a person who is authorized to commission, to ground and to tag circuits, equipment and systems in accordance with established safety practices and standards.

Turn off the power supply before performing any wiring operations!Short circuits can be harmful, critical and can cause explosions and serious burns!

Please read this manual carefully and observe all safety instructions!

DESTINATED USE

The DMA-15 is designed as HMI for universal measuring, controlling and regulating applications.
It must not be used for life critical, medical, or fail safe appilcations.

DISCLAIMER

BARTH Elektronik GmbH assumes no liability for usage and functionality of the DMA-15 in case of disregarding this manual. The strict accordance to this manual is important since the installation methods, peripheral connections, usage and maintenance can not be controlled by BARTH Elektronik GmbH. Therefore BARTH Elektronik GmbH assumes no liability for any claim.

1 Product description

The picture below shows the CAN Touch Display **DMA-15 lococube**® (Art. No. 0044-0015).

1.1 Features

- Universal CAN Display
- Color Touch TFT 2.4" 240x320p
- LED Backlight
- Interface template selection via CAN
- No programming neccessary
- Open Source programming option
- Customized templates
- Compatible with all **lococube® mini-PLCs**
- Panel-Mount without visible screws
- Panel Cut-Out (LxW): 61 x 46 mm
- Dimensions (LxW): 69 x 50 mm
- Operating Voltage 7 to 32 VDC
- Engineered and manufactured in Germany

1.2 Applications

- Industrial process control
- Automotive and maritime technology
- Technincal education / university
- Test systems

1.3 General description

The ultra small 2.4" CAN Display **DMA-15** allows convinient connection to any **lococube® mini-PLC** via CAN interface. Its bright 240 x 320 pixel TFT display uses resisitve touch technology.
The graphical interface can be selected out of a variety of templates with a single CAN message. No additional display programming is neccessary.
The **DMA-15** can be fully integrated within the graphical miCon-L software suite supporting any **BARTH® mini-PLC** with CAN interface.

1.4 Connection and operation

The CAN display **DMA-15** can be directly connected to any **BARTH**® mini-PLC via the CAN interface.
In case you use „miCon-L“ as your graphical programming tool, you can use the templates that come pre-installed in the firmware of the device.
Please see section 5 on how to get started quickly with the free miCon-L software samples we provide. They support you in sending the correct CAN-messages if you use one of your PLCs.
Please see section 6 for a list of the CAN-commands the display will accept. These can be useful, if you want to use your own CAN-hardware.

1.5 Delivery content

- **BARTH® CAN Display DMA-15**
- 1x connector for supply and CAN

2 Installation

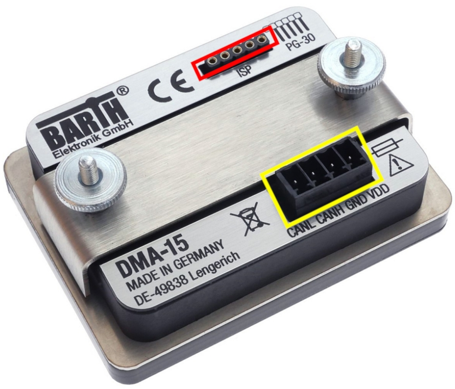
2.1 Mounting

The **DMA-15** is designed for pantel mount use only. Please use the optional gasket (SEA-24 BARTH® Art. No. 0522-0024 to meet the IP65 protection class if desired.
The measurements and dimensions are shown in the Appendix.
Please do not operate outside the specified environmental conditions!

2.2 Wiring

2.2.1 Overview

The picture below shows the backside connection layout of the **DMA-15**.



Power supply and CAN connector:

1●	+VDD	positive supply (+7 to 32 VDC)
2	GND	ground terminal (GND)
3	CANH	CAN high terminal
4	CANL	CAN low terminal

ISP connector (for Programmer PG-30):

1●	+3V3	positive power supply (Pin 1)
2	GND	ground terminal (GND)
3	SYS_SWDIO	system data IO
4	SYS_SWCLK	system clock
5	SYS_RESETN	system reset (inverted)

2.2.2 Connecting the power supply

The **DMA-15** features a wide supply voltage range from 7 to 32 VDC at very low current consumption.
The **DMA-15** can be integrated in battery supplied 12V or 24V DC systems (cars, trucks, battery powered cars, forklifts and digger, for example).

Turn off the power supply before performing any wiring operations!

⚠ **Incorrect wiring, voltage reversal or disregarding the electrical specifications may cause irreversible damage of the DMA-15!**

Connect the supply voltage of 7 to 32 VDC to the 4-pole terminal of the **DMA-15**. Wire the positive supply to the ‚+VDD‘ marked connection. The negative (ground) will be wired to the ‚GND‘ connection. All terminals may be used with a wire gauge from 0.25 to 1.5mm².

⚠ **Ensure correct power supply voltage range and polarisation! External fusing of 1A max. is mandatory! Disregarding may cause irreversible damage of the DMA-15!**

2.2.3 Connecting the CAN interface

The 4-pole connector of the **DMA-15** also contains the CAN-specific pins ‚CANH‘ and ‚CANL‘.

⚠ **The voltage at CANH or CANL must not exceed -32 or +32 VDC referred to ground (GND). Higher voltages may cause irreversible damage to the Mini-PLC!**

There is a 1k termination resistor integrated in the **DMA-15** to ensure proper operation in direct connection to a backside-mounted **BARTH® Mini-PLC**.
Please refer to the appendix for detailed electrical specification of the CAN interface.

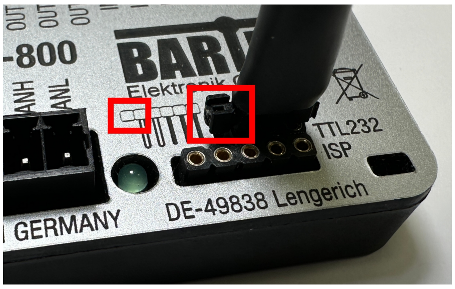
3 Requirements for Operation

To operate the **DMA-15** CAN display you will need the following items:
1. The **lococube® CAN Display DMA-15.**
2. A mini-PLC with a CAN interface (the **lococube® STG-800** is a good choice to get started and will be used in all our upcoming examples: lococube® mini-SPS STG-800
3. A **VK-16 cable** is needed to connect the mini-PLC to your computer: Verbindungskabel VK-16
4. A cable to connect the CAN-Bus of the mini-PLC to the one of the DMA-15. A **KS-85 wiring harness**, which has those connections already included can be found here: Wiring harness KS-85
5. A resistor of 60Ω needs to be placed between the CANH and the CANL connector of the DMA-15 and the mini-PLC. (You can omit this step if you‘re using a **KS-85 wiring harness** since it already includes those resistances).
6. A power adapter to power the mini-PLC and the DMA-15 that supplies a voltage between 7 - 32 VCD. (If you‘re using a **KS-85 wiring harness** you can plug in a hollow plug with these dimensions: Ø 5,5 mm / 2,1 mm).
7. A computer with a Windows operating system installed.
8. The miCon-L software which can be downloaded here: miCon-L
9. The template programs which can be downloaded here: Template Programs

4 Operation with miCon-L

4.1 Connecting the mini-PLC

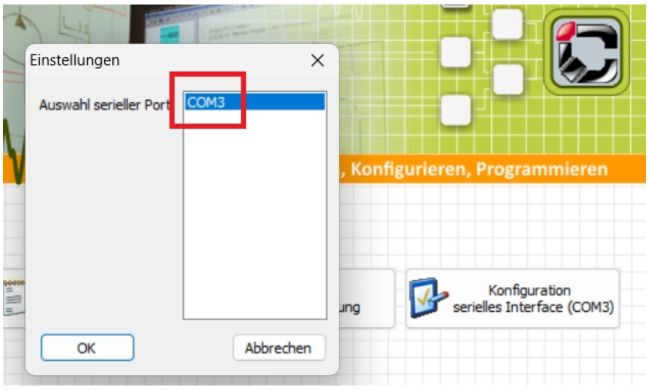
Connect the mini-PLC to your computer. We‘ll be using the **lococube® STG-800.** Plug the USB connector of the **VK-16** cable into your computer and plug the other end into the mini-PLC into the TTL232 slot. The orientation of the connector matters! Please orient the connector, with the little bump, facing **left**, as shown below.



Connect the CANH terminal of the mini-PLC to the CANH terminal of the **DMA-15** and connect the CANL terminal of the mini-PLC to the CANL terminal of the DMA-15.
If you are using our **KS-85 wiring harness**, this can be done simply by plugging the X1 connector into the mini-PLC and the X2 connector into the **DMA-15** or vice versa. If you are not using a **KS-85 wiring harness**, make sure to put a resistor of 60Ω between the CAN terminals! Plug in the power adapter and power the mini-PLC and the **DMA-15**.

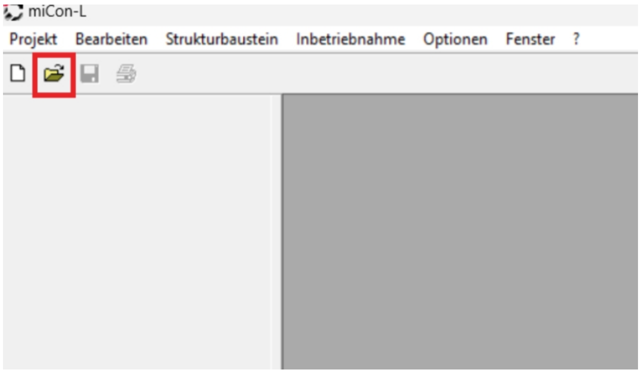
1.4 Using miCon-L

Download and install the miCon-L software on your computer. Download the template programs for the **DMA-15** here and place them at a location of your chosing: After the **STG-800** and the **DMA-15** have been properly wired up and connected, start the miCon-L software.

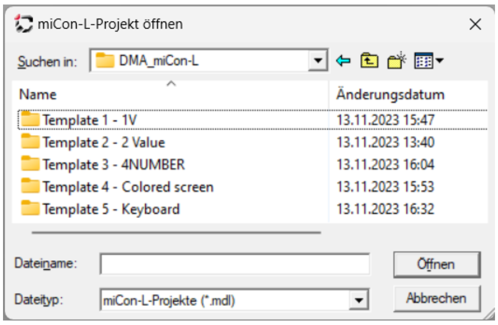

Einstellungen

Auswahl serieller Port: COM3
OK
Abbrechen
, Konfigurieren, Programmieren
Konfiguration serielles Interface (COM3)

It‘s vital that you choose the correct serial port before you continue into the actual programming environment!



Now use the „folder icon“ in the top left hand corner to navigate to the folder you have placed the template projects in.


miCon-L-Projekt öffnen

Suchen in: DMA_miCon-L

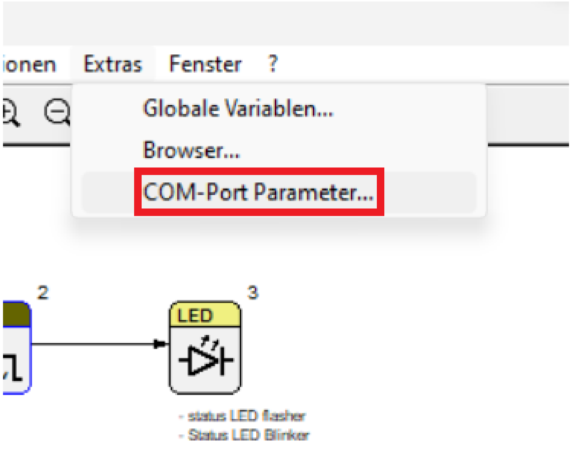
Name	Änderungsdatum
Template 1 - 1V	13.11.2023 15:47
Template 2 - 2 Value	13.11.2023 13:40
Template 3 - 4NUMBER	13.11.2023 16:04
Template 4 - Colored screen	13.11.2023 15:53
Template 5 - Keyboard	13.11.2023 16:32

Dateiname:

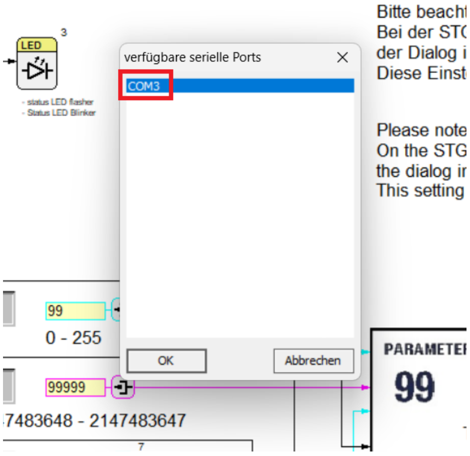
Dateityp: miCon-L-Projekte (*.mdl)

Navigate to the location you chose to put the template folder in and open one of the projects using the .MDL file.

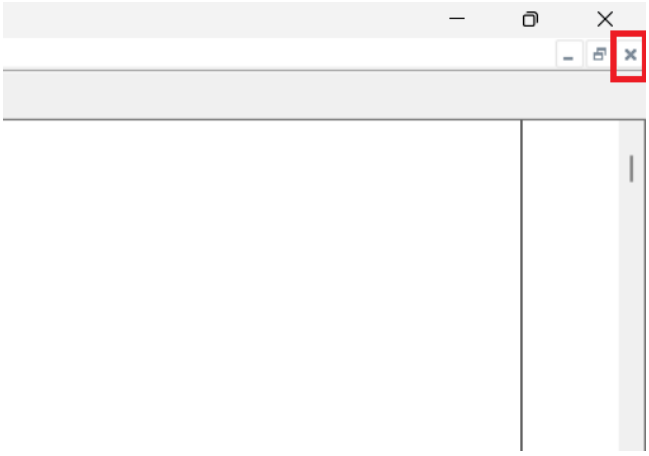
After you have succesfully opened the project of your choice, there is one additional step that you must always do when creating or opening a new project! Go to Extras -> COM-Port Parameter.



Choose the correct COM-Port again and click „OK“.

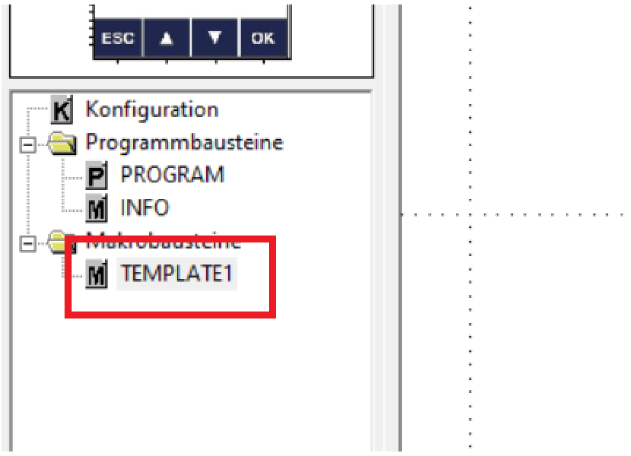


Before you can edit the program you will need to close the currently active window first. Press the ‚X‘ in the upper right hand corner to close it.

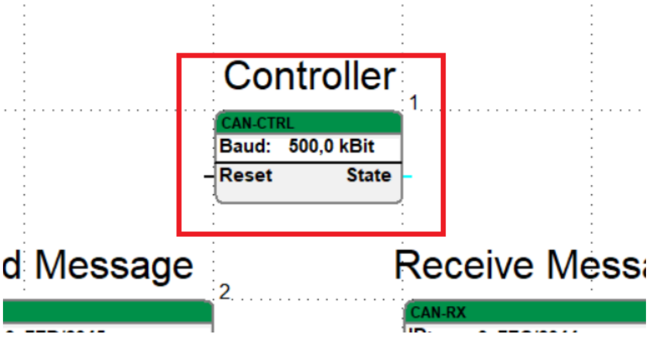


For the **DMA-15** to properly function you will have to set the correct baud rate. It‘s shown on the **DMA-15** display by default when you power it and it receives no CAN messages yet.
In our case we need to set it to a default rate of 250kBit.

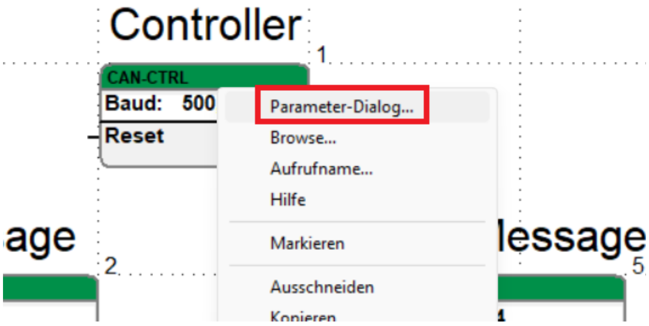
Navigate to the **macro block** on the left.



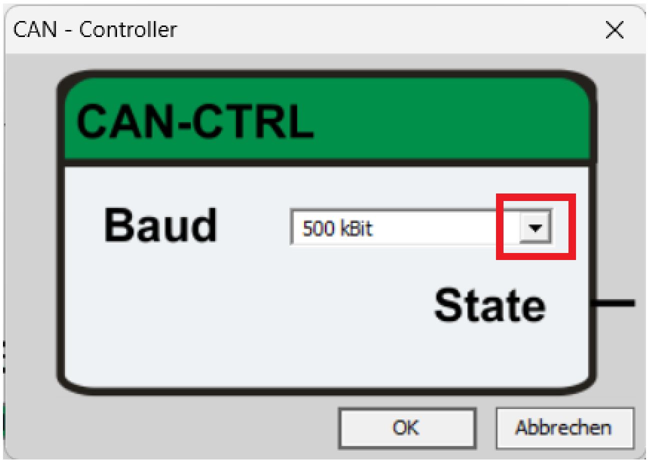
Here you can set the baud rate of the **STG-800**.



Right click on it and choose „Parameter-Dialog...“.

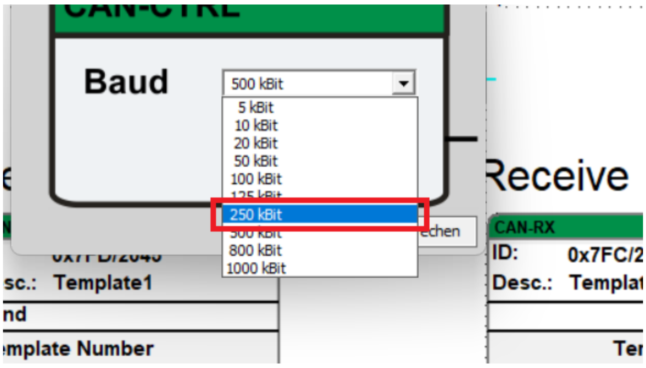


Click the little arrow pointing down to open the context .

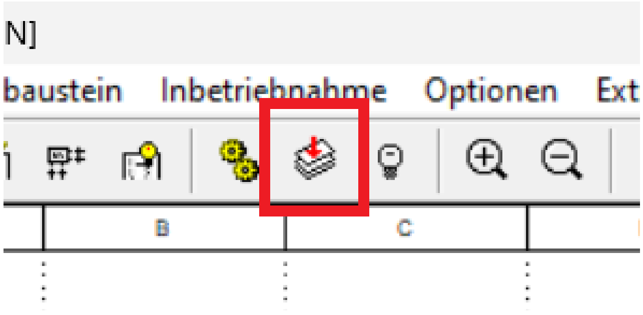


Baud 500 kBit

Now we set the baud rate to 250kBit.

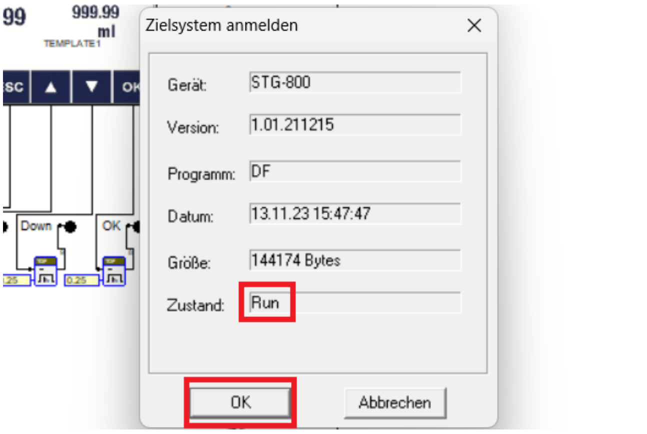


Next we need to download the program onto the **STG-800**.



Now you will see a status window pop up that has various kinds of information on it, including what type of **BARTH® mini-PLC** you're using, version, date, etc.

If the status bar at the bottom finished and the the status says „Run" at the bottom, hit „OK".


Zielsystem anmelden

Gerät: STG-800

Version: 1.01.211215

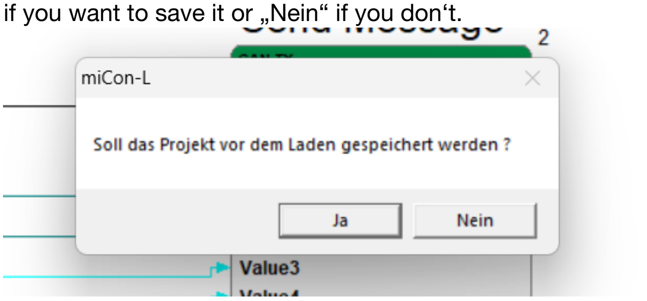
Programm: DF

Datum: 13.11.23 15:47:47

Größe: 144174 Bytes

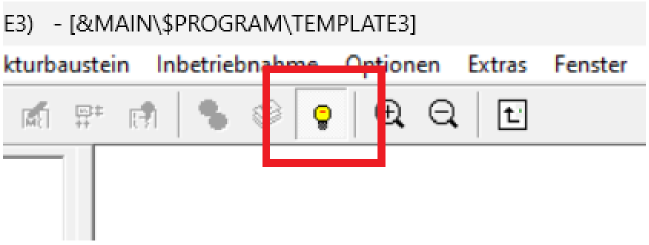
Zustand: Run

If you run the program for the first time or made any changes to it and have not saved it yet, miCon-L will ask you if you want to save the program before running. Click on „Ja" if you want to save it or „Nein" if you don't.


miCon-L

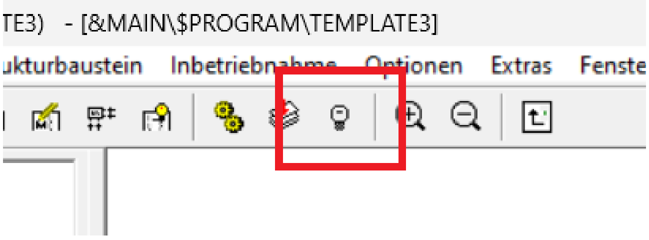
Soll das Projekt vor dem Laden gespeichert werden ?

The program will automatically be set online now, indicated by the little yellow light bulb icon at the top, among the now greyed out icons.



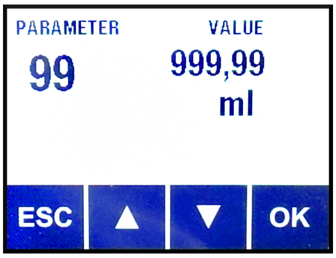
If you want to stop the program and take it offline just click on that light bulb.

If you have not made any changes to the program and just want to run it again, you can click on the light bulb to put it online again.



You only have to use the download button again if you made any changes to the program.

Now the **DMA-15** display should switch and display the information provided by the program template you‘re currently using. Here we are using „TEMPLATE1“.

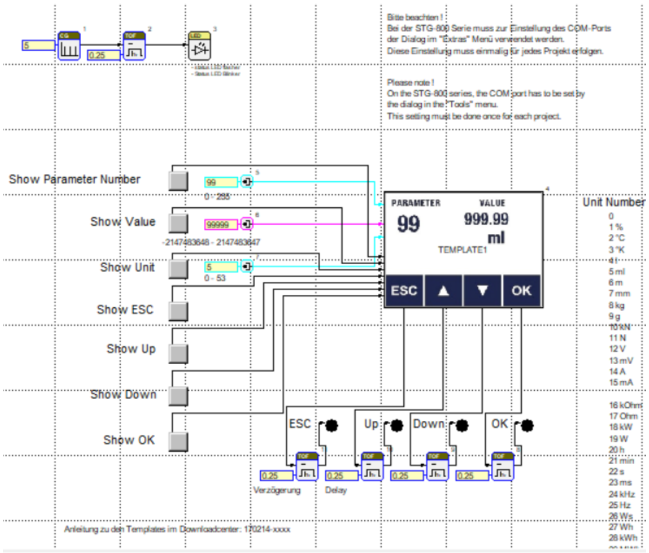


5 Programming Templates

Now we will take a closer look at the provided programming templates.

5.1 Template 1 - 1V

Here we can see an overview of the main **program**.

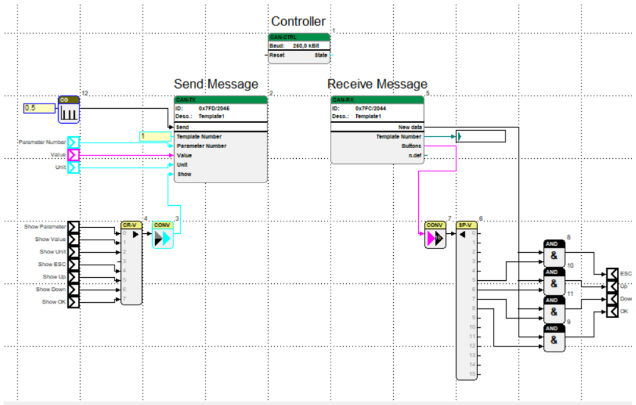


Bitte beachten !
Bei der STG-800 Serie muss zur Einstellung des COM-Ports der Dialog im "Extras" Menü verwendet werden.
Diese Einstellung muss einmalig für jedes Projekt erfolgen.

Please note !
On the STG-800 series, the COM port has to be set by the dialog in the "Tools" menu.
This setting must be done once for each project.

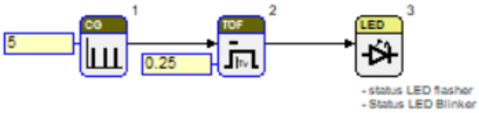
Anleitung zu den Templates im Downloadcenter: 170214-xxxx

And the **macro block** „TEMPLATE1“

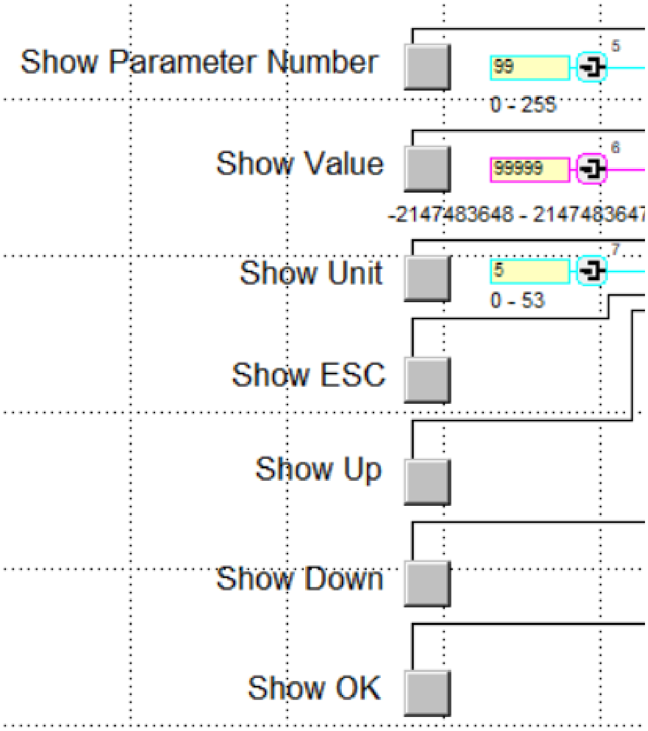


The **program** sets all the inputs and parameters, and the **macro block** is what actually controls what the buttons and those parameters do.

Let‘s take a closer look the the **program** first.

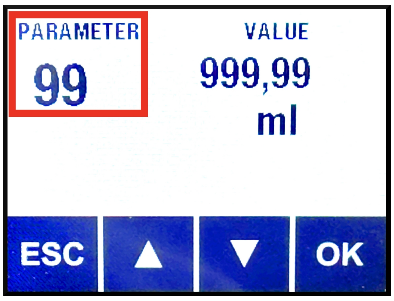


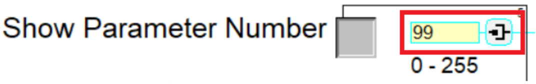
This part of the **program** makes the LED on the **STG-800** blink every 5 s with a 0.25 s long pulse. It‘s there to indicate whether or not the program has successfully downloaded and is running correctly.



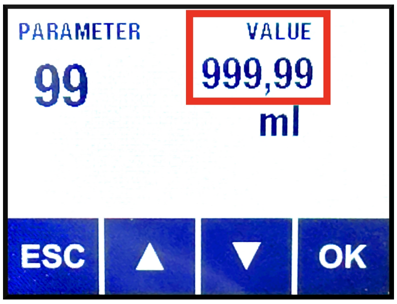
These buttons toggle every input that is displayed on the screen. Toggling them will turn the corresponding interface module off, or back on, when pressed again - even when the program is running.

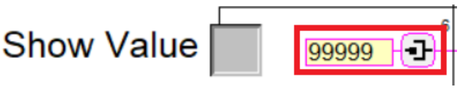
We also have three pins, each with a different global parameter assigned to them. These pins will be used as inputs in the **macro block**.



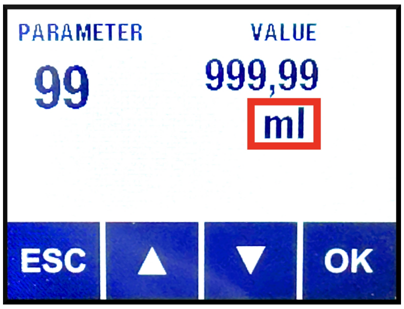


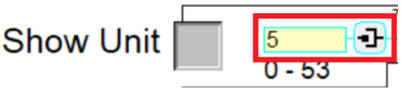
The pin next to the „Show Parameter Number“ button has a global parameter with the value of 99. It decides what numbers is displayed on the screen of the **DMA-15**. The input toggled by the button is what decides whether or not it is shown on the screen.



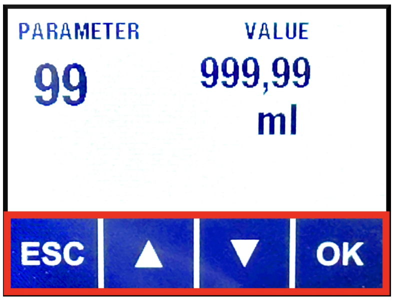


The pin next to the „Show Value“ button has a global parameter with the value of 99999. It decides what number is displayed on the screen of the **DMA-15**. The input toggled by the button is what decides whether or not it is shown on the screen.

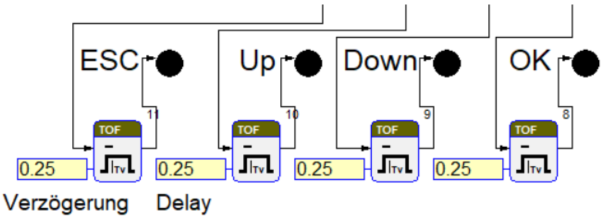




The pin next to the „Show Unit“ button has a global parameter with the value of 5. It decides what unit is displayed on the screen of the **DMA-15**. The input toggled by the button is what decides whether or not the unit is shown on the screen. The legend for the units can be found next to the template picture, in the **program** part.

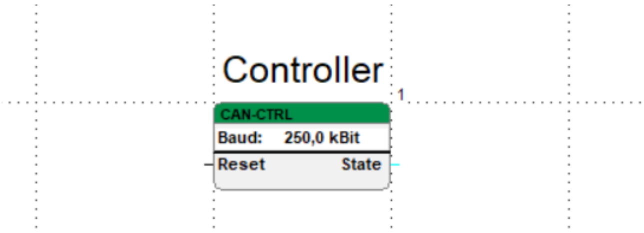


The inputs „Show ESC“, „Show Up“, „Show Down“, and „Show OK“ are used to toggle whether or not the buttons, at the bottom of the **DMA-15** screen, are displayed or not.

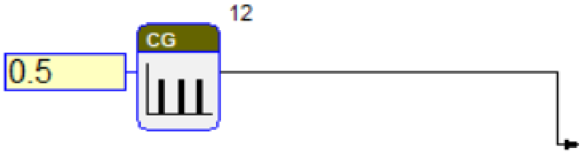


This section of the **program** updates the input buttons on the screen every 0.25 seconds.

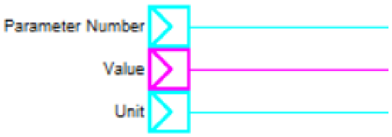
Next, let‘s take a look at the **macro block**.



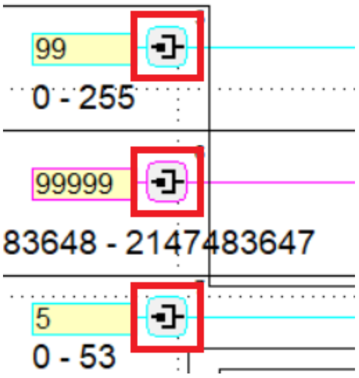
The function block „CAN-CTRL“ is the CAN-Controller and it needs to be included to set the baud rate of the **STG-800** to that of the **DMA-15**. It does not need to be connected to anything.

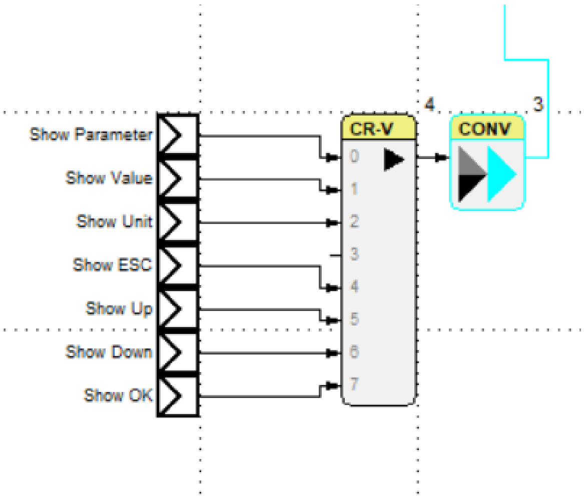


This function block sends a pulse in a 0.5 s cycle updating the „CAN-TX“ function block.



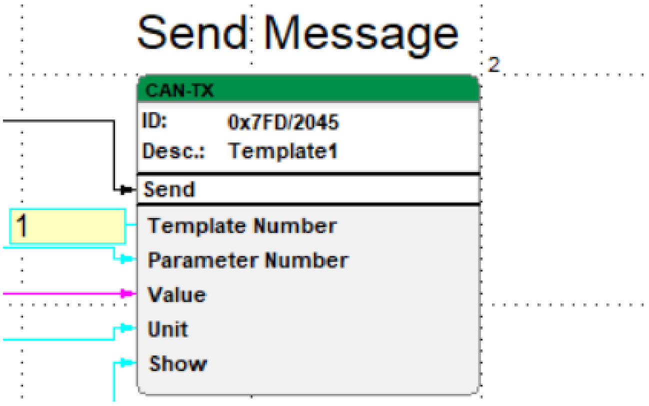
These are the inputs that are linked to the pins from the **program** (as shown below), that were mentioned earlier.





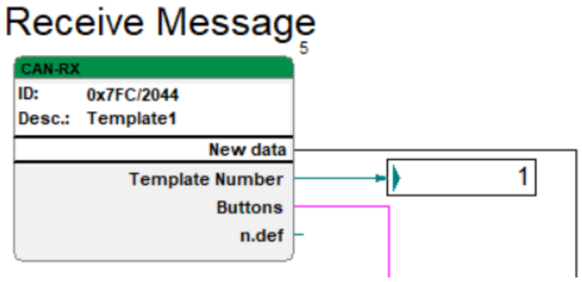
Next, we are going to look at the all the inputs from the button presses in **program** being called and added together so they can all be converted into the proper data type.

This function block takes all the individual inputs, combining the into a vector, allowing conversion with a single function block.

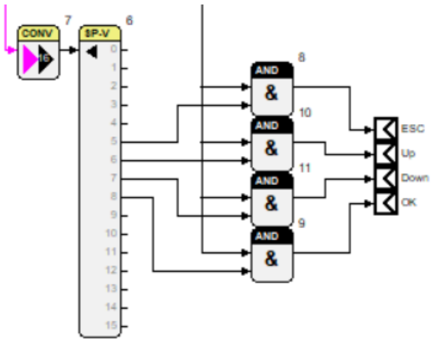


This is the „CAN-TX“ function block. It sends the data to the screen of the **DMA-15**.
The input „Template Number“ has a global parameter of 1, and sets the Template Number to 1.
Under that, we have the input „Parameter Number“ which is the number set in **program** on the first pin. By default it is set to 99 and should display that on the **DMA-15**.
Below that, we have the input „Value“ which corresponds to the second pin in **program**. By default it is set to 99999 and should display 999,99 that on the **DMA-15**.
And under that, we have the input „Unit“ which corresponds to the third pin in **program**. It‘s set to 5 by default and and displays „ml“ on the screen. A legend for the units can be found next to the template picture, in the **program** part.

And lastly, we have the input „Show“ which takes all the combined inputs from earlier and shows them on the actual screen of the **DMA-15**.



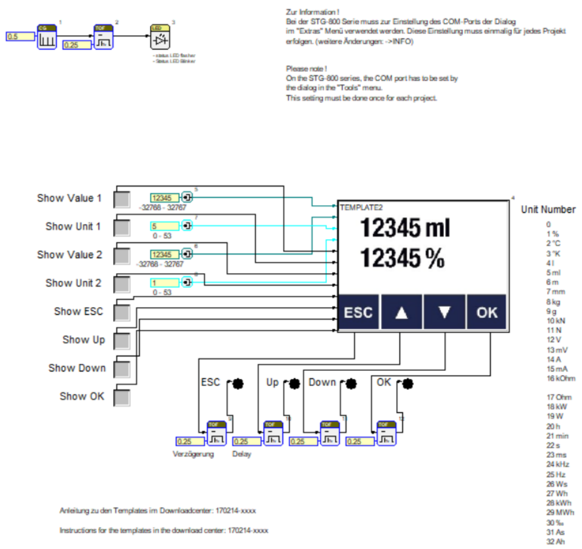
This is the „CAN-RX“ function block. It receives data (gives signal when true), shows the „Template Number“ in the box to the right, and obtains the data from button presses, via the touchscreen of the **DMA-15**. There is one undefined output. It has no value, but a data type assigned to it. It‘s needed to ensure the message has the correct length of 8 byte, required by the **DMA-15**.



This part of the **macro block** checks which button has been pressed. For that, two conditions must be met. All of the button outputs are connected to their own „AND“ function block.
Now, if any of the buttons are pressed, the „CAN-RX“ function block receives data on which of the buttons has been pressed. A button press also sends a signal that new data is being sent. The new data sets the first condition of all four „AND“ function blocks true. When a button is pressed, only the corresponding „AND“ function block gets a signal. That way, we can check which of the buttons has actually been pressed.

5.2 Template 2 - 2Value

Here we can see an overwiew of the main **program:**



Zur Information !
Bei der STG-800 Serie muss zur Einstellung des COM-Ports der Dialog im "Extras" Menü verwendet werden. Diese Einstellung muss einmalig für jedes Projekt erfolgen. (weitere Änderungen: ->INFO)

Please note !
On the STG-800 series, the COM port has to be set by the dialog in the "Tools" menu.
This setting must be done once for each project.

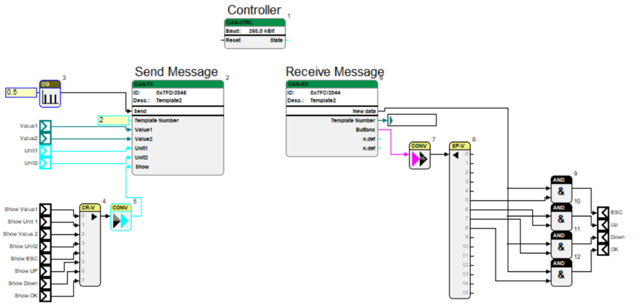
Unit Number

0
1 %
2 °C
3 °K
4 l
5 ml
6 m
7 mm
8 kg
9 g
10 kN
11 N
12 V
13 mV
14 A
15 mA
16 kOhm
17 Ohm
18 kW
19 W
20 h
21 min
22 s
23 ms
24 kHz
25 Hz
26 Ws
27 Wh
28 kWh
29 MWh
30 ‰
31 As
32 Ah

Anleitung zu den Templates im Downloadcenter: 170214-xxxx

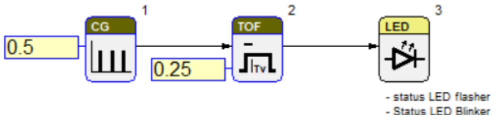
Instructions for the templates in the download center: 170214-xxxx

And the **macro block** „TEMPLATE2“.

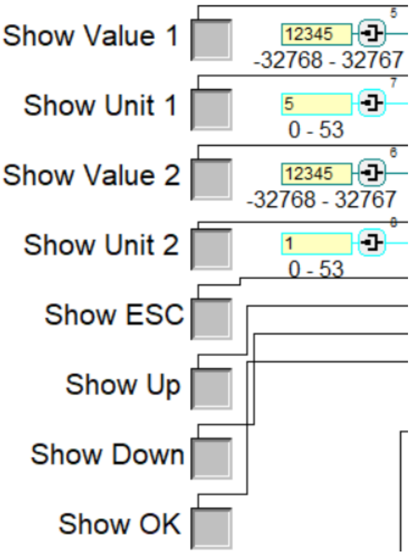


The **program** sets all the inputs and parameters, and the **macro block** is what actually controls what the buttons and those parameters do.

Lets take a closer look at the **program** first.

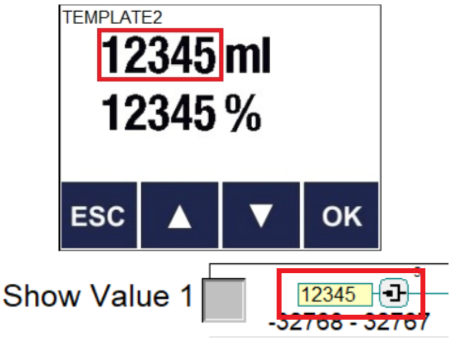


This part of the **program** makes LED on the **STG-800** blink every 0.5 s with a 0.25 s long pulse. It‘s there to indicate whether or not the program has successfully downloaded and is running correctly.

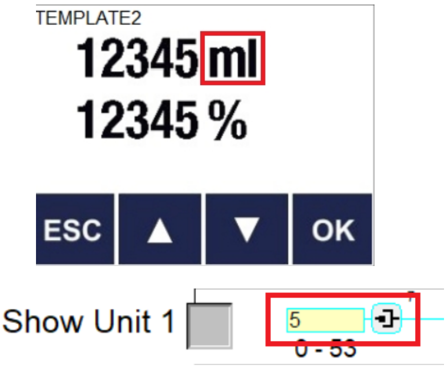


These buttons toggle every input that is displayed on the **DMA-15**. Toggling them will turn the corresponding interface module off, or back on when pressed again - even while the program is running.

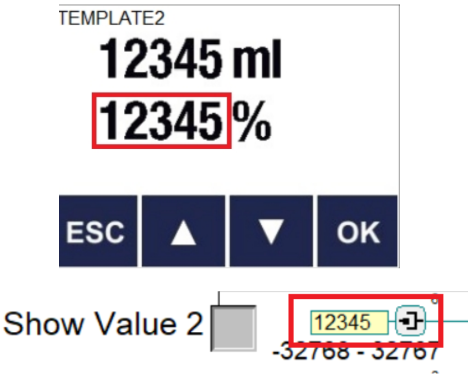
We also have four pins, each with a different global parameter assigned to them. These pins will be used as inputs in the **macro block**.



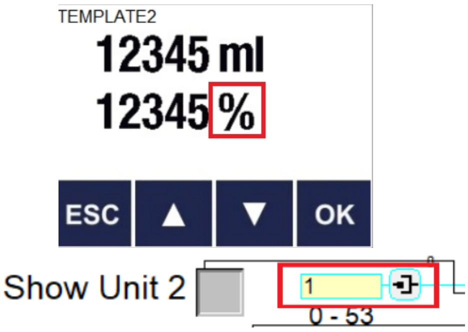
The pin next to the „Show Value 1“ button has a global parameter with the value of 12345. It decides what numbers is displayed on the screen of the **DMA-15**. The input toggled by the button is what decides whether or not it is shown on the screen:



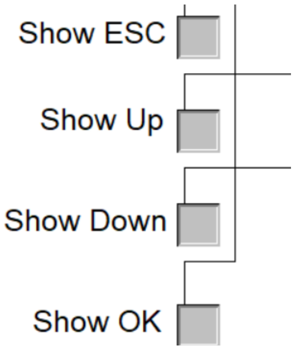
The pin next to the „Show Unit 1“ button has a global parameter with the value of 5. It decides what unit is displayed on the screen of the **DMA-15**. The input toggled by the button is what decides wether or not the unit is shown on the screen. A legend for the units can be found next to the template picture, in the **program** part.



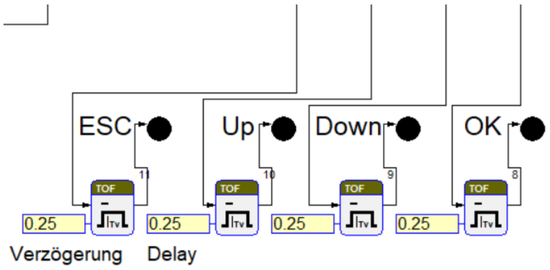
The pin next to the „Show Value 2“ button has a global parameter with the value of 12345. It decides what number is displayed on the screen of the **DMA-15**. The input toggled by the button is what decides whether or not it is shown on the screen.



The pin next to the „Show Unit 2“ button has a global parameter with the value of 1. It decides what unit is displayed on the screen of the **DMA-15**. The input toggled by the button is what decides wether or not the unit is shown on the screen. A legend for the units can be found next to the template picture, in the **program** part.

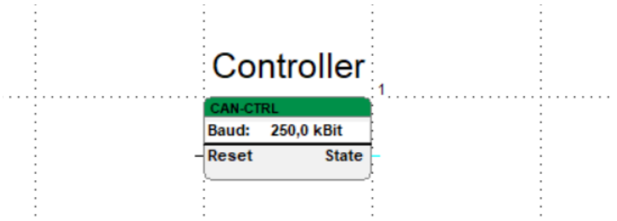


The inputs „Show ESC“, „Show Up“, „Show Down“, and „Show OK“ are used to toggle whether or not the buttons, at the bottom of the **DMA-15** screen, are displayed or not.

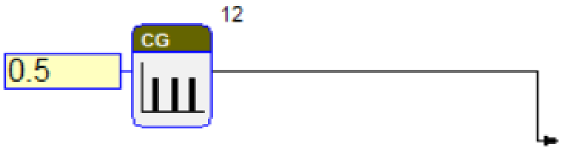


This section of the **program** updates the buttons on the screen every 0.25 seconds.

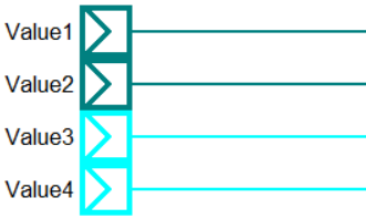
Next, let‘s take a look at the **macro block**.



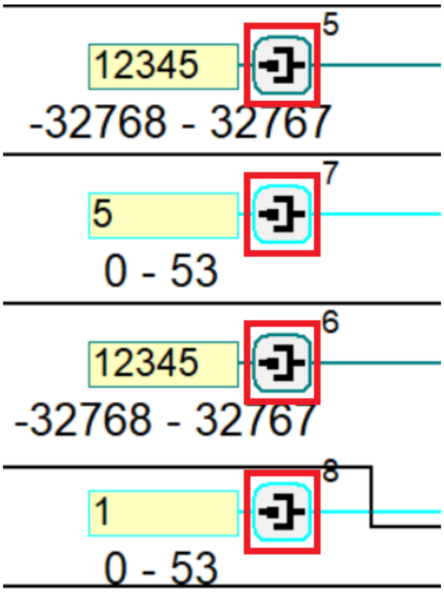
The function block „CAN-CTRL“ is the CAN-Controller and it needs to be included to set the baud rate of the **STG-800** to that of the **DMA-15**. It does not need to be connected to anything.

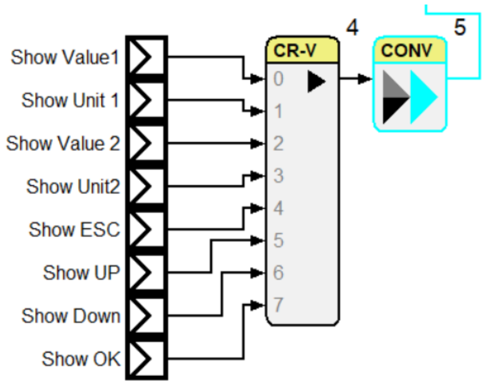


This function block sends a pulse in a 0.5 s cycle updating the „CAN-TX“ function block.

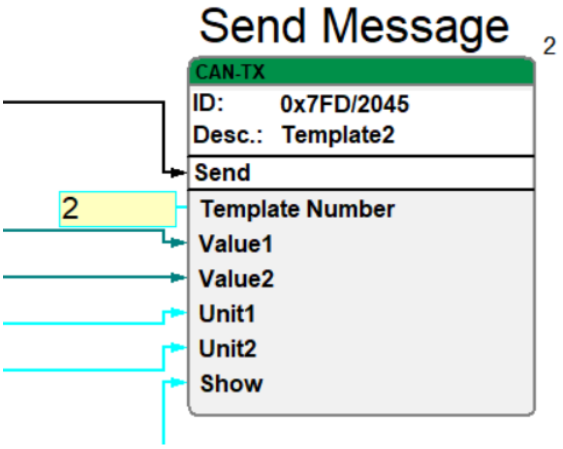


These are the inputs that are linked to the pins from the **program** (as shown below), that were mentioned earlier.



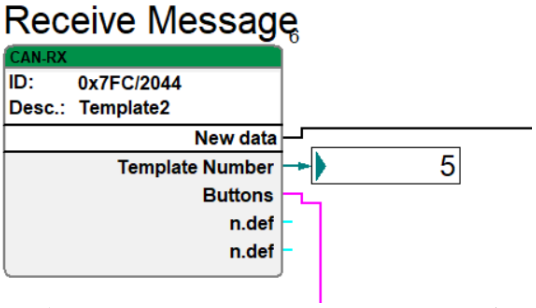


This function block takes all the individual inputs, combining them into a vector, allowing conversion with a single function block.



This is the „CAN-TX“ function block. It sends the data to the screen of the **DMA-15**.
The input „Template Number“ has a global parameter of 2, and the Template Number to 2.
Under that, we have the input „Value “ which is the number set in **program** on the first pin. By default it is set to 12345 and should display that on the **DMA-15**.
Below that, we have the input „Value2“ which corresponds to the third pin in **program**. By default it is set to 12345 and should display that on the **DMA-15**.
Under that, we have the input „Unit1“, which corresponds to the second pin in **program**. It‘s set to 5 by default and displays „ml“ on the screen.
And lastly, we have the input „Unit2“, which corresponds to the fourth pin in **program**. It‘s set to 1 by default and displays „%“ on the screen.
A legend for the units can be found next to the template picture, in the **program** part.

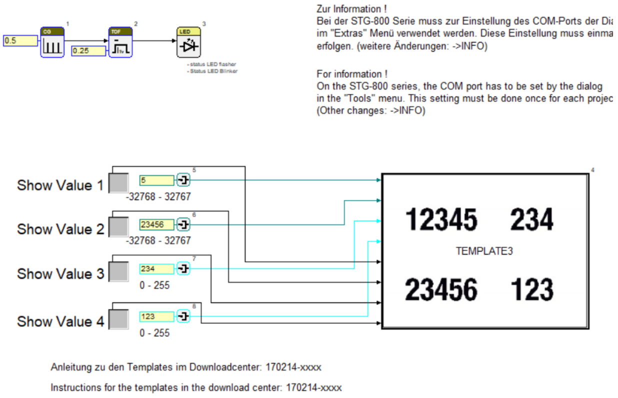
And lastly, we have the input „Show“ which takes all the combined inputs from earlier and shows them on the actual screen of the **DMA-15**.



This is the „CAN-RX“ function block. It receives data (gives signal when true), shows the „Template Number“ in the box to the right, and obtains the data from button presses on the touchscreen of the **DMA-15**. There are two undefined outputs. They have no value, but data types assigned to them. They are needed to ensure the message has the correct length of 8 byte, required by the **DMA-15**.

5.3 Template 3 - 4NUMBER

Here we can see an overview of the main **program**.



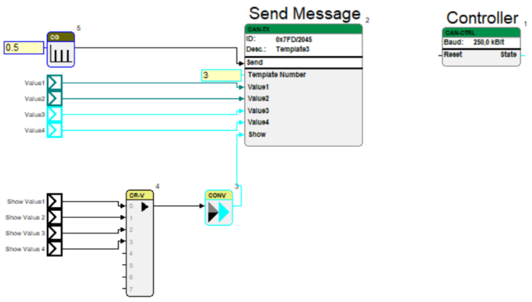
Zur Information !
Bei der STG-800 Serie muss zur Einstellung des COM-Ports der Di
im "Extras" Menü verwendet werden. Diese Einstellung muss einma
erfolgen. (weitere Änderungen: ->INFO)

For information !
On the STG-800 series, the COM port has to be set by the dialog
in the "Tools" menu. This setting must be done once for each projec
(Other changes: ->INFO)

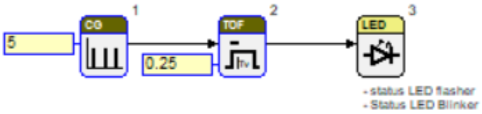
Anleitung zu den Templates im Downloadcenter: 170214-xxxx

Instructions for the templates in the download center: 170214-xxxx

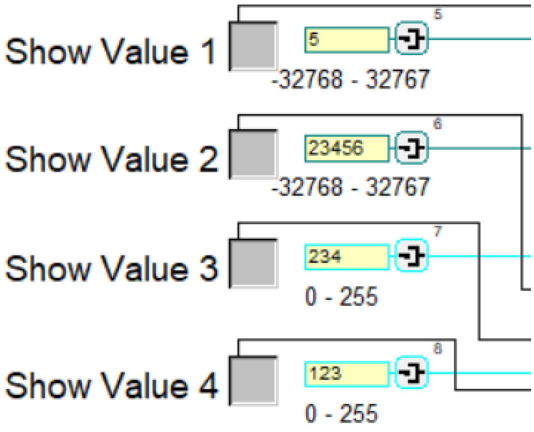
And the **macro block** „TEMPLATE3“.



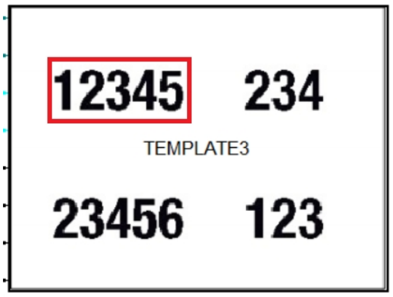
Let‘s take a closer look at the **program** first.

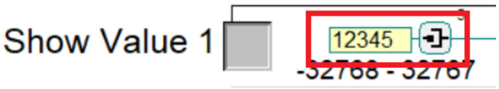


This part of the **program** makes the LED on the **STG-800** blink every 5 s with a 0.25 s long pulse. It‘s there to indicate whether or not the program has successfully downloaded and is running correctly.

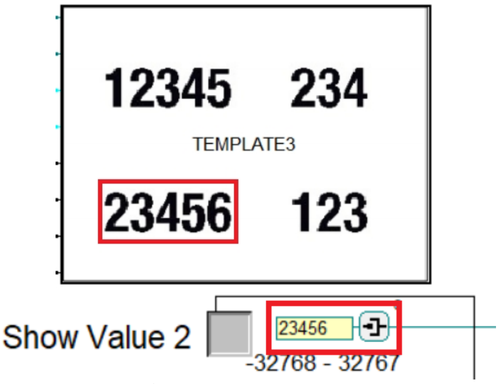


These buttons toggle every input that is displayed on the screen. Toggling them will turn on the corresponding interface module off, or back on when pressed again - even when the program is running.
We also have four pins, each with a different global parameter assigned to them. These pins will be used as inputs in the **macro block**.

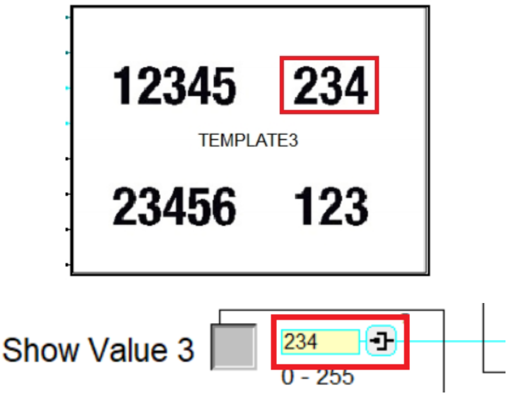




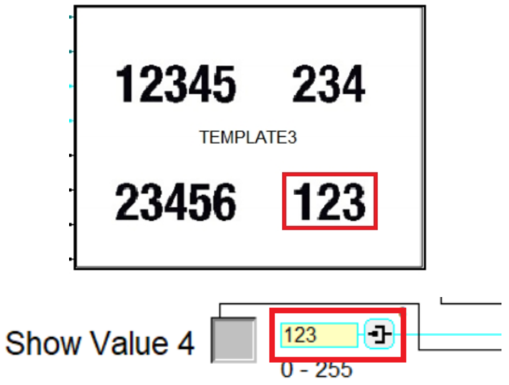
The pin next to the „Show Value 1“ button has a global parameter with the value of 12345. It decides what numbers is displayed on the screen of the **DMA-15**. The input toggled by the button is what decides whether or not it is shown on the screen.



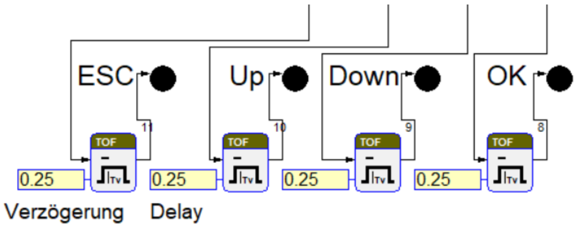
The pin next to the „Show Value 2“ button has a global parameter with the value of 23456. It decides what number is displayed on the screen of the **DMA-15**. The input toggled by the button is what decides whether or not it is shown on the screen.



The pin next to the „Show Value 3“ button has a global parameter with the value of 234. It decides what number is displayed on the screen of the **DMA-15**. The input toggled by the button is what decides whether or not it is shown on the screen.

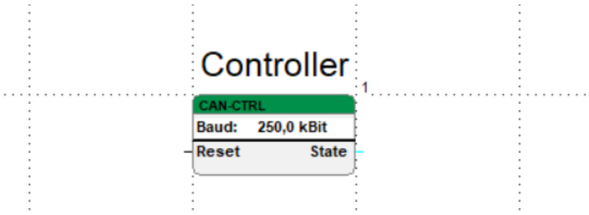


The pin next to the „Show Value 4“ button has a global parameter with the value of 123. It decides what number is displayed on the screen of the **DMA-15**. The input toggled by the button is what decides whether or not it is shown on the screen.

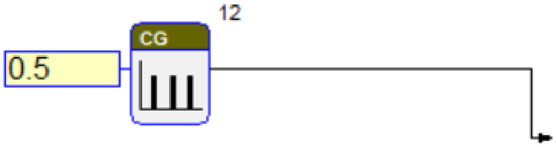


This section of the program updates the input buttons on the screen every 0.25 seconds.
This section of the **program** updates the buttons on the screen every 0.25 seconds.

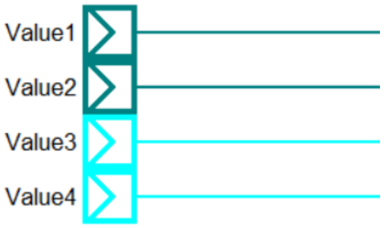
Next, we take a look at the **macro block** „TEMPLATE3“.



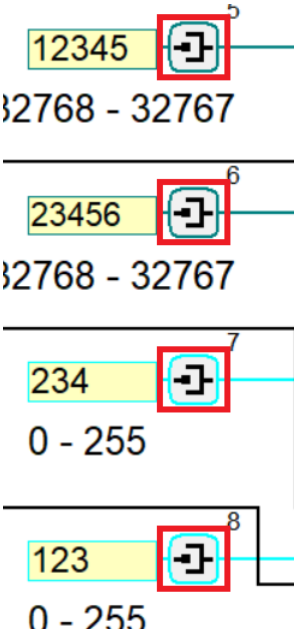
The function block „CAN-CTRL“ is the CAN-Controller and it needs to be included to set the baud rate of the **STG-800** to that of the **DMA-15**. It does not need to be connected to anything.

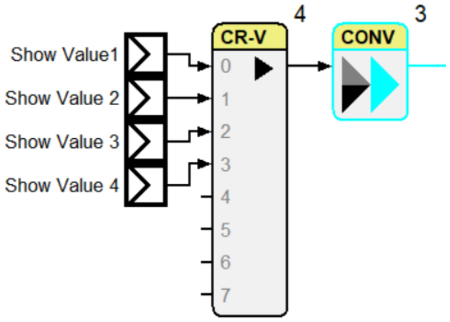


This function block sends a pulse in a 0.5 s cycle updating the „CAN-TX“ function block.

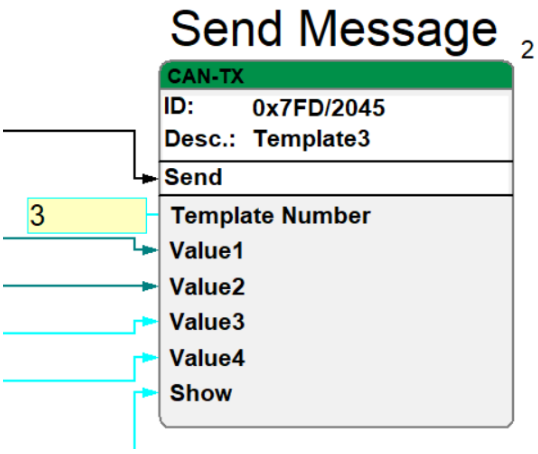


These are the inputs that are linked to the pins from the **program** (as shown below), that were mentioned earlier.





This function block takes all the individual inputs, combining the into a vector, allowing conversion with a single function block.

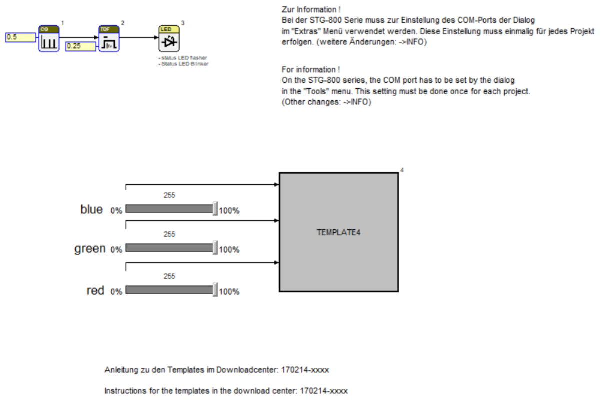


This is the „CAN-TX“ function block. It sends the data to the screen of the **DMA-15**.
The input „Template Number“ has a global parameter of 3, and sets the Template Number to 3.
Under that, we have the input „Value1“ which is the number set in **program** on the first pin. By default it is set to 12345 and should display that on the **DMA-15**.
Below that, we have the input „Value2“ which corresponds to the second pin in **program**. By default it is set to 23456 and should display that on the **DMA-15**.
After that, we have the input „Value3“ which corresponds to the second pin in **program**. By default it is set to 234 and should display that on the **DMA-15**.
Lastly, we have the input „Value4“ which corresponds to the second pin in **program**. By default it is set to 123 and should display that on the **DMA-15**.
And lastly, we have the input „Show“ which takes all the combined inputs from earlier and shows them on the actual screen of the **DMA-15**.

Since this program only displays numbers and does not receive any input, there is no need for „CAN-RX“ function block.

5.4 Template 4 - Colored screen

Here we can see an overview of the main **program**.

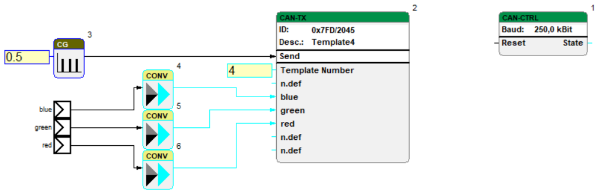


Zur Information !
Bei der STG-800 Serie muss zur Einstellung des COM-Ports der Dialog im "Extras" Menü verwendet werden. Diese Einstellung muss einmalig für jedes Projekt erfolgen. (weitere Änderungen: ->INFO)

For information !
On the STG-800 series, the COM port has to be set by the dialog in the "Tools" menu. This setting must be done once for each project.
(Other changes: ->INFO)

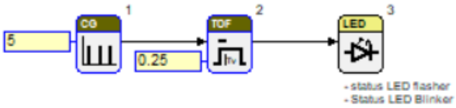
Anleitung zu den Templates im Downloadcenter: 170214-xxxx
Instructions for the templates in the download center: 170214-xxxx

And the **macro block** „TEMPLATE4“.

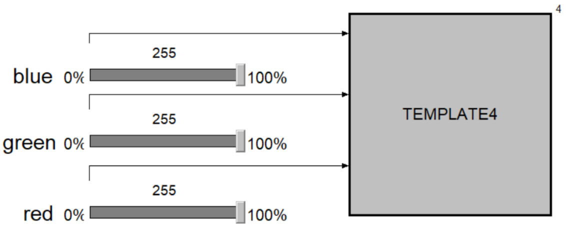


The **program** sets all the inputs and parameters, and the **macro block** is what actually controls what the buttons and those parameters do.

Let‘s take a closer look at the **program** first.

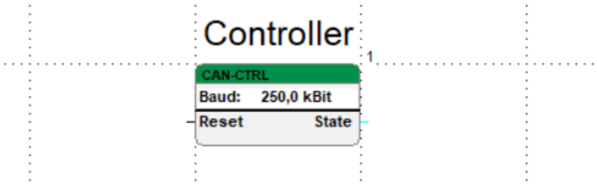


This part of the **program** makes the LED on the **STG-800** blink every 5 s with a 0.25 s long pulse. It's there to indicate whether or not the program has successfully downloaded and is running correctly.

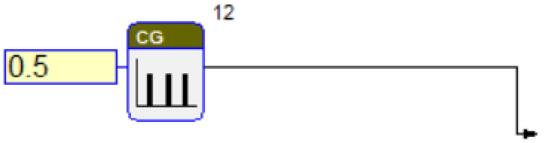


This part of the **program** has the three sliders. „blue“, „green“, and „red“, and depending on how you adjust those sliders, the color of the screen will change according to the RBG values.

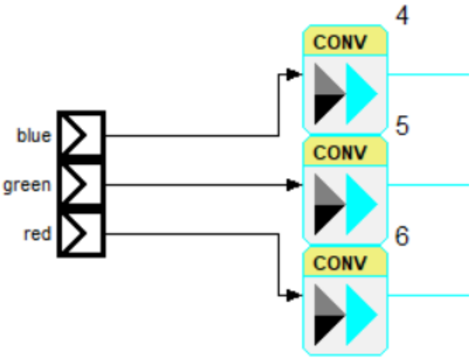
Next, let's take a look at the **macro block**.



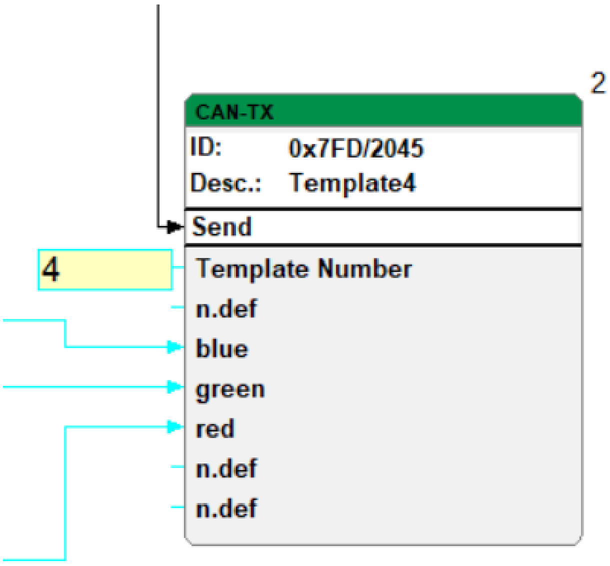
The function block „CAN-CTRL“ is the CAN-Controller and it needs to be included to set the baud rate of the **STG-800** to that of the **DMA-15**. It does not need to be connected to anything.



This function block sends a pulse in a 0.5 s cycle updating the „CAN-TX“ function block.



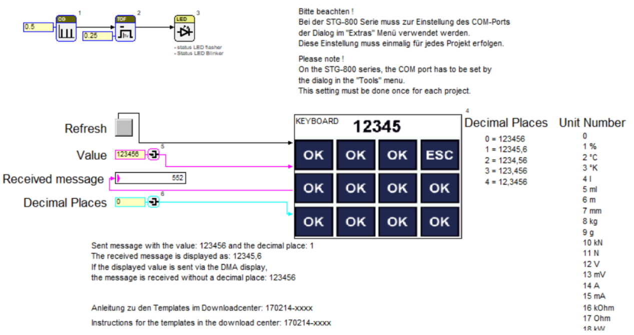
Here, the three inputs from the **program** are called and converted into the correct format.



This is the „CAN-TX“ function block. It sends the data to the screen of the **DMA-15**.
The input „Template Number“ has a global parameter of 4, and sets the Template Number to 4.
We have the inputs „blue“, „green“, „red“, which correspond to the inputs from the **program**.
The 2nd, 7th, and 8th input are not defined. but have a data type associated with them. They are needed to ensure the correct message length of 8 byte will be sent
to the **DMA-15**.

5.5 Template 5 - Keyboard

Here we can see an overview of the main **program**.



Bitte beachten !
Bei der STG-800 Serie muss zur Einstellung des COM-Ports
der Dialog im "Extras" Menü verwendet werden.
Diese Einstellung muss einmalig für jedes Projekt erfolgen.

Please note !
On the STG-800 series, the COM port has to be set by
the dialog in the "Tools" menu.
This setting must be done once for each project.

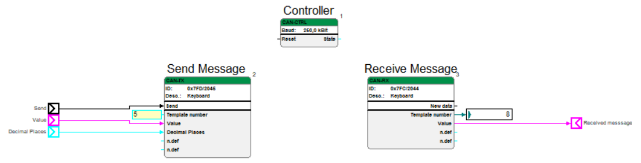
Decimal Places	Unit Number
0 = 123456	0
1 = 12345,6	1 %
2 = 1234,56	2 °C
3 = 123,456	3 °K
4 = 12,3456	4 l
	5 ml
	6 m
	7 mm
	8 kg
	9 g
	10 kN
	11 N
	12 V
	13 mV
	14 A
	15 mA
	16 kOhm
	17 Ohm
	18 kW

Sent message with the value: 123456 and the decimal place: 1
The received message is displayed as: 12345,6
If the displayed value is sent via the DMA display,
the message is received without a decimal place: 123456

Anleitung zu den Templates im Downloadcenter: 170214-xxxx

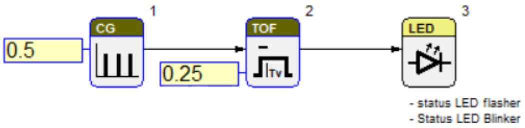
Instructions for the templates in the download center: 170214-xxxx

And the **macro block** called „KEYBOARD“.

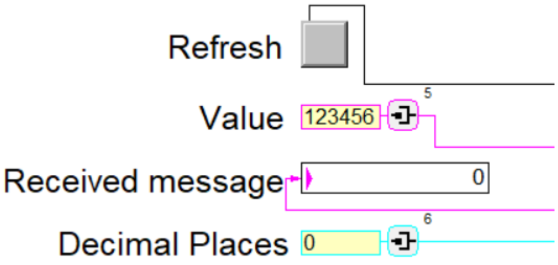


The **program** sets all the inputs and parameters, and the **macro block** is what actually controls what the buttons and those parameters do.

Let‘s take a closer look at the **program** first.

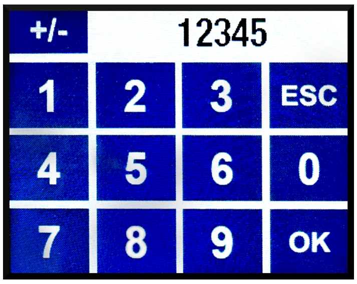


This part of the program is just to make the LED on the **STG-800** blink every 0.25 seconds, as an indicator that the program has successfully downloaded and is working properly.

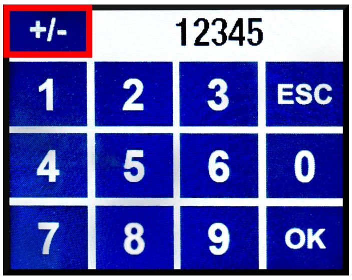


The button labeled „Refresh“ refreshes the screen in case the **DMA-15** did not update, after you‘ve started the program. There are two pins. One is called „Value“ with the global paramter 123456, and one called „Decimal Places“ with the gobal paramter 0. The first pin decides what value will be displayed on the screen of the **DMA-15** by default. The second pin decides if there will be a decimal point placed, and how many decimal places there will be.

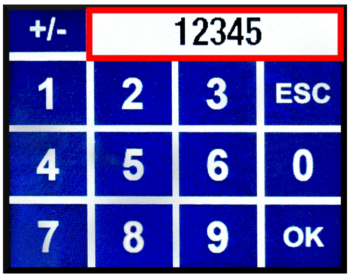
The box called „Received messages“ displays the number you‘ve entered on the touchscreen of the DMA-15, when you press „OK“.



Here we have an overview of the screen layout.

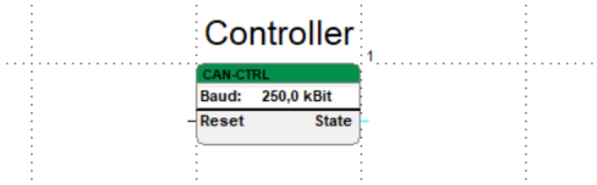


This changes the number from positive to negative, and back, when pressed again.

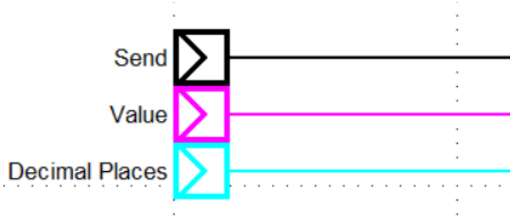


Below that, we have the keypad, with numbers ranging from 0 to 9, an „ESC“ button to reset the number back to 0, and an „OK“ button that‘s used to update the „Received message“ part of the program.

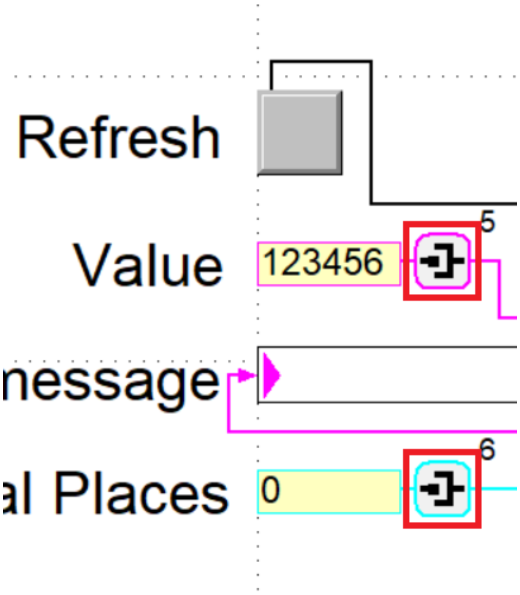
In the next step, we will take a closer look at the **macro block** called „KEYBOARD".

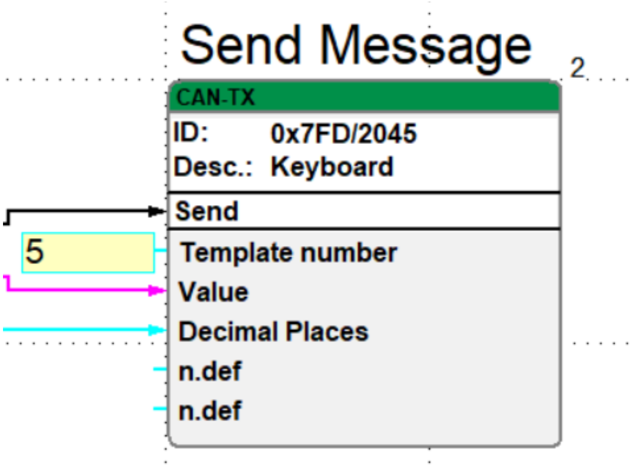


The function block „CAN-CTRL" is the CAN-Controller and it needs to be included to set the baud rate of the **STG-800** to that of the **DMA-15**. It does not need to be connected to anything.



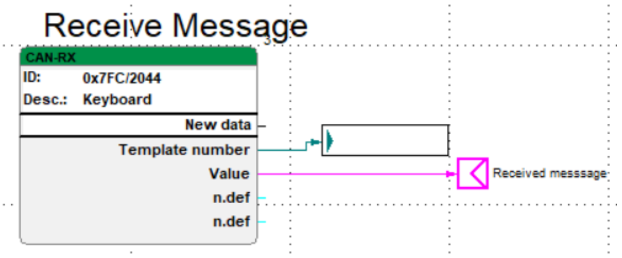
These are inputs that call the pins from the **program** (as shown below), that were mentioned earlier.





This is the „CAN-TX" function block. It sends the data to the screen of the **DMA-15**.
The input „Template Number" has a global parameter of 5, and sets the Template Number to 5.
Under that, we have the input „Value" which is the number set in **program** on the first pin. By default it is set to 123456 and should display that on the **DMA-15**.
Below that, we have the input „Decimal Places" which corresponds to the second pin in **program**. By default it is set to 0, which means there there won't be any decimal point placed at all.

The 4th and 5th input are not defined. but have a data type associated with them. They are needed to ensure the correct message length of 8 byte will be sent to the **DMA-15**.



This is the „CAN-TX" function block, responsible for receiving messages. It gives a signal when it receives new data, shows the „Template Number" in the box to the right, and fetches the number you put in on the touchscreen of the **DMA-15**. The two not defined values are there because the **DMA-15** needs a message length of 8 byte. Even though, they don't have a value set, they are assigned a data type.

6 CAN commands

In the following, we'll take a look at the CAN-commands that are used in the example templates.
A CAN-message sent to the DMA-15 has a length of 8 byte.

Next, we will take a look at the individual templates and their corresponding CAN-commands.

6.1 Template 1 – 1V

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	0x63	0x9F ← 0x86 ← 0x01 ← 0x00				0x05	0xF7
1	99	159	134	1	0	5	247
U8 Template number 0 - 255	U8 Parameter number 0 - 255	I32 Value -2147483648 - 2147483647 Value will be shown with two decimal places				U8 Unit 0 - 255	U8 Visibility of the controls 0 - 255

(Byte number, hexadecimal, decimal, data type, title, number range, comment)

1. Byte 0 is an unsigned char with a number range from 0 - 225. It has a hexadecimal value of 0x01, which is equal to the decimal number 1, effectively setting the "Template number" to 1.
2. Byte 1 is also an unsigned char with a number range from 0-255. It has a hexadecimal value of 0x63, which sets the parameter value on the display to the decimal number 99.
3. Byte 2, 3, 4, and 5 are combined to form a 4 byte int32, with a number range from -2147483648 - 2147483647. It is read from right to left, i.e. from byte 5 to byte 2, so: 0x00, 0x01, 0x86, 0x9F. This corresponds to the decimal number 99999, and since a decimal point will be placed before the last two digits it becomes 999,99.
4. Byte 6 is an unsigned char with a number range from 0 - 255. It has a hexadecimal value of 0x05, which is equal to the number 5. If you take a look at the unit number legend, to the right in the program part of the miCon-L template, you can see that the number 5 corresponds to the unit of "ml".
5. Byte 7 is another unsigned char with a number range from 0 - 255. It has a hexadecimal value of 0xF7, which is equal to the number 247, toggling visibilty, one bit per control (Bits: 1111 0111):
   Bit 0: Parameter number
   Bit 1: Value
   Bit 2: Unit
   Bit 3: -
   Bit 4: Button ESC
   Bit 5: Button UP
   Bit 6: Button DOWN
   Bit 7: Button OK

6.2 Template 2 – 2Value

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x02	0x39 ← 0x30		0x39 ← 0x30		0x05	0x01	0xFF
2	12345		12345		5	1	255
U8 Template number 0 - 255	I16 Value 1 -32768 - 32767		I16 Value 2 -32768 - 32767		U8 Unit1 0 - 255	U8 Unit2 0 - 255	U8 Visibility of the controls 0 - 255

(Byte number, hexadecimal, decimal, data type, title, number range, comment)

1. Byte 0 is an unsigned char with a number range from 0 - 225. It has a value of 0x02, which is equal to the number 2, effectively setting the "Template number" to 2.
2. Byte 1 and 2 are combined to form a 2 byte int16, with a number range from -32768 - 32767. It is read from right to left, i.e. from byte 2 to byte 1, so: 0x30, 0x39. This corresponds to the decimal number 12345.
3. Byte 3 and 4 are combined to form a 2 byte int16, with a number range from -32768 - 32767. It is read from right to left, i.e. from byte 2 to byte 1, so: 0x30, 0x39. This corresponds to the decimal number 12345.
4. Byte 5 is an unsigned char with a number range from 0 - 255. It has a value of 0x05, which is equal to the number 5. If you take a look at the unit number legend, to the right in the program part of the miCon-L template, you can see that the number 5 corresponds to the unit of "ml".
5. Byte 6 is an unsigned char with a number range from 0 - 255. It has a value of 0x01, which is equal to the number 1. If you take a look at the unit number legend, to the right in the program part of the miCon-L template, you can see that the number 5 corresponds to the unit of "%".
6. Byte 7 is another unsigned char with a number range from 0 - 255. It has a value of 0xFF, which is equal to the number 255, toggling visibilty, one bit per control (Bits: 1111 1111):
   Bit 0: Value1
   Bit 1: Unit1
   Bit 2: Value2
   Bit 3: Unit2
   Bit 4: Button ESC
   Bit 5: Button UP
   Bit 6: Button DOWN
   Bit 7: Button OK

6.3 Template 3 – 4NUMBER

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x03	0x39 ← 0x30		0xA0 ← 0x5B		0xEA	0x7B	0x0F
3	123456		123456		234	123	15
U8 Template number 0 - 255	I16 Value 1 -32768 - 32767		I16 Value 2 -32768 - 32767		U8 Value 3 0 - 255	U8 Value 4 0 - 255	U8 Visibility of the controls 0 - 255

(Byte number, hexadecimal, decimal, data type, title, number range, comment)

1. Byte 0 is an unsigned char with a number range from 0 - 225. It has a hexadecimal value of 0x03, which is equal to the number 3, effectively setting the "Template number" to 3.
2. Byte 1 and 2 are combined to form a 2 byte int16, with a number range from -32768 - 32767. It is read from right to left, i.e. from byte 2 to byte 1, so: 0x30, 0x39. This corresponds to the decimal number 12345.
3. Byte 3 and 4 are combined to form a 2 byte int16, with a number range from -32768 - 32767. It is read from right to left, i.e. from byte 4 to byte 3, so: 0x5B, 0xA0. This corresponds to the decimal number 23456.
4. Byte 5 is an unsigned char with a number range from 0 - 255. It has a hexadecimal value of 0xEA, which is equal to the decimal number 234.
5. Byte 6 is an unsigned char with a number range from 0 - 255. It has a hexadecimal value of 0x7B, which is equal to the decimal number 123.
6. Byte 7 is another unsigned char with a number range from 0 - 255. It has a value of 0x0F, which is equal to the number 15, toggling visibilty, one bit per control (Bits: 0000 1111):

Bit 0:	Value1
Bit 1:	Value2
Bit 2:	Value3
Bit 3:	Value4
Bit 4:	-
Bit 5:	-
Bit 6:	-
Bit 7:	-

6.4 Template 4 – Colored Screen

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x04	0x00	0xFF	0xFF	0xFF	0x00	0x00	0x00
4	0	255	255	255	0	0	0
U8 Template number 0 - 255	not used	U8 blue 0 - 255 0 = 0% 255 = 100%	U8 green 0 - 255 0 = 0% 255 = 100%	U8 red 0 - 255 0 = 0% 255 = 100%	not used	not used	not used

(Byte number, hexadecimal, decimal, data type, title, number range, comment)

1. Byte 0 is an unsigned char with a number range from 0 - 225. It has a hexadecimal value of 0x04, which is equal to the number 4, effectively setting the "Template number" to 4.
2. Byte 1 is not used in this template.
3. Byte 2 is an unsigned char with a number range from 0 - 255. It has a hexadecimal value of 0xFF, which is equal to the number 255. This sets the color blue to 100%
4. Byte 3 is an unsigned char with a number range from 0 - 255. It has a hexadecimal value of 0xFF, which is equal to the number 255. This sets the color green to 100%
5. Byte 4 is an unsigned char with a number range from 0 - 255. It has a hexadecimal value of 0xFF, which is equal to the number 255. This sets the color red to 100%
6. Byte 5, 6, and 7 are not used in this template.

6.5 Template 5 – Keyboard

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x05	0x40 ← 0xE2 ← 0x01 ← 0x00				0x00	0x00	0x00
5	123456				0	0	0
U8 Template number 0 - 255	I32 Value -2147483648 - 2147483648				U8 Decimal places	not used	not used

1. Byte 0 is an unsigned char with a number range from 0 - 225. It has a hexadecimal value of 0x05, which is equal to the number 5, effectively setting the "Template number" to 5.
2. Byte 1, 2, 3, and 4 are combined to form a 4 byte int32, with a number range from -2147483648 - 2147483647. It is read from right to left, i.e. from byte 5 to byte 2, so: 0x00, 0x01, 0xE2, 0x10. This corresponds to the decimal number 123456.
3. Byte 5 is an unsigned char with number range from 0 - 255. It has a dexadecimal value of 0x00, which is equal to the decimal number 0.
   Where a decimal point is placed is decided by the decimal number, corresponding to the hexadecimal number. E.g:
   0: no decimal point
   1: Tenths
   2: Hundretdhs
   3: Thousandths
   4: Ten Thousandths

- When pressing one of the number buttons the number will be added on last position of the numberfield. If the Number is out of Range of signed word the value will be zero “0”.
- If the button “+-” is pressed the value will be negated (e.g. 123 → -123 or -234 → 234).
- If the button “ESC” is pressed the field for the value will be cleared and show a zero “0”.
- If the button "OK" is pressed the message will be sent after the button is released.

6.6 Baudrate

It is possible to change the baudrate of the DMA-15 with a series of CAN messages, each corresponding to a different baudrate.
These messages need to be sent to the DMA-15 with a specific ID and message length.

Message for 125k baudrate

ID: 0x7F4
DLC: 8 Byte

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
00	00	125	00	00	00	00	00

Message for 250k baudrate

ID: 0x7F4
DLC: 8 Byte

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
00	00	250	00	00	00	00	00

Message for 500k baudrate

ID: 0x7F4
DLC: 8 Byte

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
00	00	244	01	00	00	00	00

Message for 1M baudrate

ID: 0x7F4
DLC: 8 Byte

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
00	00	232	03	00	00	00	00

7 Appendix

7.1 Specifications

7.1.1 General

Hardware design	BARTH® Mini CAN Touch Display, panel mount with plugable connector
Programming options	miCon-L Software control via CAN ,graphical (function block style), simulation, CAN programming and visualisation, free license
	Open Source, C-Programming control using the powerful KEIL µVision® Software, free license
Interfaces	CAN 2.0A/B

7.1.2 Power supply

Operating voltage	7 to 32 VDC
Current consumption	100 mA @ 12 VDC 55 mA @ 24 VDC 45 mA @ 32 VDC
Fusing	1A max. (external) mandatory for voltage reversal protection
Voltage reversal protection	yes (combined with external fuse)
ESD/TVS protection	yes, integrated
Heat dissipation air (at full load)	normally < 1 W

7.1.3 Display

Technology	TFT (Thin-Film Transistor) Liquid Crystal Display Module
Size	2.4” Diagonal
Resolution	240 x 320 Pixels / 262K colors
Touch	4-wire Resistive Touch Panel
Backlight	LED white

7.1.4 Interfaces

CAN	CAN 2.0A/B: 11/29 bit ID, base frame format supported baud rates: 50, 100, 125, **250**, 500 kbit, 1Mbit
	Internal termination resistor: 1k (for direct connection to one mini-PLC), meets or exceeds the requirements of applications ISO 11898-2, loss of ground protection from –32 V to +32 V, thermal shutdown protection

7.1.5 Security features

Security Features	System and independent watchdog Fail safe oscillator Power on/down reset Supply voltage supervisor

7.1.6 Program and data memory

Memory	5Mb Flash, 196k RAM

7.1.7 Timebase (oscillator)

Primary Oscillator	Crystal quartz MEMS unit (precise ,micro-electro-mechanical system‘)
Nominal Frequency	16.000 MHz
Frequency tolerance	±50 × 10^{-6}
Frequency aging	±5 × 10^{-6} / year max.

7.1.8 Electrical connection

Electrical Connection	plugable screw type connector 0.25 to 1.5 mm^2 Manufacturer: Phoenix Contact Series: COMBICON Type: MC1,5/4-ST-3,5(-BK)

7.1.9 Electromagnetic compatibility (EMC)

Electrostatic discharge (ESD) at supply terminals	20 kV air discharge 30 kV contact discharge (IEC/EN 61 000-4-2, level 3)
Electromagnetic fields	Field strength 10 V/m (IEC/EN 61000-4-3)
CAN bus terminals (CANH, CANL to GND)	IEC 61000-4-2: Unpowered Contact Discharge ±15000 V IEC 61000-4-2: Powered Contact Discharge ±8000 V

7.1.10 Environmental conditions

Operation temperature	-20 to +70 °C (IEC 60068-2-1/2)
Storage temperature	-30 to 70 °C (IEC 60068-2-1/2)
Relative humidity	5 to 80% non-condensing (IEC 60068-2-30)
Air pressure (in operation)	500 to 1500 hPa
Shock resistance	min. 50 m/s² (IEC 60068-2-27)
Vibration resistance	min. 10 m/s² @ 10..100 Hz (IEC 60068-2-6)
Degree of protection	IP 40 (without additional gasket) IP65 (with Gasket SEA-23) (EN 50178, IEC 60529)
Free fall (packaged)	1000 mm (IEC 60068-2-32)

7.1.11 Weight and dimensions

Weight	90 G (without connector)
Dimensions	60 x 50 x 51 mm (LxWxH)
Panel cut-out	31 x 46 mm (LxW)

7.1.12 Ordering information

Ordering information	CAN Touch Display DMA-15 Art. No. 0040 GTIN 4351929401504
Ordering information accessory	Programmer ST-Link/V2 ISOL Art. No. 0017-0066 GTIN 4251329401481
	Connection Cable VK-35 Art. No. 0091-0035 GTIN 425132940127
	Mini-PLC STG-800 Art. No. 0850-0800 GTIN 4251329401207

7.2 Documents, videos, and software

Detailed information, additional documents, application notes and videos related to this product are downloadable from Barth-Eletronik and miCon-L.

7.3 Disposal

If you wish to dispose of the product, ask your local recycling centre or dealer for details about how to do this in in accordance with the applicable disposal regulations.

7.4 Conformity declaration

For the following designated product, it is hereby confirmed, that the construction in that technical desgin provided by us corresponds to the standards specified below. In the event of any alteration which has not been approved by us being made to any device listed below, this statement shall thereby be made invalid.

Description	CAN Touch Display
Type	DMA-15
Art. No.	0044-0015
Directive 2004/108/EG relating to electromagnetic compatability (EMC) CE	Applied norms: 2004/108/EG 2004/108/EC 2014/30/EU
RoHS Directive 2011/65EU	We herby declare that our product is compliant to the RoHS directive on restrictions of the use of certain hazardous substances in electrical and electronic appliances.

Barth Elektronik GmbH

Lengerich, 19.01.2018

Dipl. -Ing. (FH) D. Barth
Managing Director