24.1 - v2.0.0

New Product Features 🔗

Increased storage - I•C4C GO 🔗

Starting from this release we included the support for I-C4C devices with an increased flash memory storage of 16 MB.

The original I•C4C PRO and BASIC variants had 8 MB of storage.

The reason for this flash memory upgrade is to future proof our product. Larger memory will allow us to dedicate more space to the application, and thus include more logic and functionality in our future programs.

At the moment we are not running out of storage space on any of our variants.   New features that will be implemented will be supported by all I•C4C Go variants, unless stated otherwise in the release notes.

In the next releases we will work on storing logfiles in the flash memory so that they can be downloaded directly. The web-interface, which is currently embedded in the application. By removing the web-interface pages from the application we will decrease the application size by 20%. This space can be used for programming new features. The web pages will be saved outside the memory which is reserved for the application.

Logged-in user can identify the flash memory size of their device by checking the "version" and "size" of the OTA_0 and OTA_1 partition in the Firmware Tab. Size = 2048 kB and Version >= 2.0.0 indicates that the device is a 16MB model.

	8MB	8MB_MQTT	16MB
Application Size	1024 KB	1152 KB	2048 KB

OTA_0

label	ota_0
type	0
subtype	16
address	0x10000
size	2048 kb
encrypted	0
OTA State	2
magic word	2882360370
secure version	2882360370
version	v2.0.0
project name	PRLM165_v0-0-0
time	10:01:51
date	Mar 27 2024
idf_ver	v4.3.7

Rest API improvements 🔗

The Rest API is providing the information that is displayed on the web-interface. Data is is presented in a text-based format called JSON.

In the previous version the JSON text was composed without using a software library. All new API calls are implemented using a library, which improves code performance and readability. The rewriting of the REST API reduces the load on the CPU and increases the performance of the web-interface. At the moment old and new REST API functions are both used by the web-interface.

In the future all old function calls will be replaced by an improved version.

v1.0.0 - Old REST API	Waiting	Size
/input_handler	55 ms	384 B
/output_handler	65 ms	266 B
/analogOutput_handler	65 ms	115 B
	185 ms	**765 B**

v2.0.0 - New REST API	Waiting	Size
/api/v1/status/IO/	15 ms	1600 B

Ikuflex 🔗

Ikuflex is the name of the protocol that allows the I-C4C GO to present information on the Display of the radio transmitter.

The IC4C-Go sends the information to the radio receiver over CAN and the radio receivers transmits the information to the radio transmitter via a wireless signal.

The current standard for tandem applications is developed for IK3 & IK4 radio transmitter that are equipped with a TM80 radio module and a 4.3" display.

When developing the user screens we applied the "dark cockpit" concept which we borrowed from the aviation industry. This means that when everything is functioning as it should there are no icons or colors that are drawing the attention from the operator.

When certain movements are not allowed their icon is shown on the display so that the operator can see the current status of the system in the blink of an eye.

This Ikuflex integration shows the potential of our product ecosystem.

Custom screens can be created on request, the creation of these screens is a multi-disciplinary effort since it requires designing the graphics and symbols, programming the radio transmitter and the logic in the IC4C-GO.

System Overview
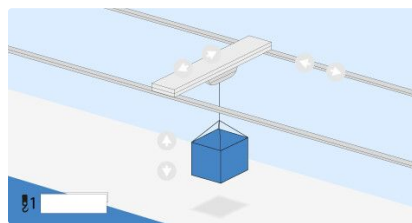
Single Hoist View

Tandem View

Partition Table 🔗

The partition table is used to define how the flash memory which is available on the I•C4C GO is used. Each piece of memory that is assigned to be used is called a partition. The label, size, type and all other information from all partitions combined is called the partition table.

It is not recommended to change the partition table on a live system, since it is stored in a protected area of the flash memory. When changing this information in a live system there is l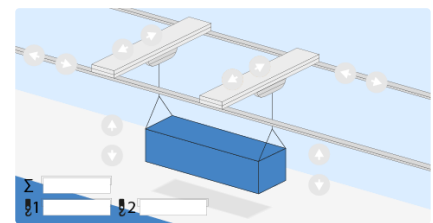arge chance of bricking the device. That is why we opted for optimizing the use of the partition table as is instead of changing it.

With the addition of new features there was a need for more storage to be able to save the configuration and data.

Since this release the parameter storage is used to store IO description and all configuration parameter which were available before.

	8MB	8MB_MQTT	16MB
Parameter storage	~~8 KB~~ 128 KB	~~8 KB~~ 64 KB	256 KB

Can bus monitoring 🔗

The status and configuration of the CAN bus are now visible for any logged in user.

When the system is configured to communicate over CAN with connected device their status is shown in a Live-view. Devices that are not available are marked as red. Active devices from which data is received are shown in green with two buttons to visualize the mapped data or raw PDO information.

This screen was added to improve the troubleshooting proces of devices connected to the CAN bus without the need of a can bus dongle.

CAN Configuration and Status

Live info from loadcell

Inactive devices shown in red

IO Naming 🔗

Any user that is connected to the IC4C-Go over Wi-Fi can open the I/O tab which shows a clear overview of the different inputs and outputs. They are shown with their pin number and corresponding state which is updated automatically.

The pin Number matches with the marking on the enclosure to avoid confusion.

When the user is logged-in it's possible to edit the text in the Description field so that the purpose of the Input/Output is clear.

The length of the description is limited to 40 characters.

I/O

Save Changes

Safety Output

Index	Pin Number	State	Description
1	23 - 24	0	safety_output_description

Analog Output

Index	Pin Number	Value (mv)	Description
1	A0	1000	analog_output_description

Outputs

Index	Pin Number	State	Description
1	1 - 2	0	output_1_description
2	3 - 4	0	output_2_description
3	5 - 6	0	output_3_description
4	7 - 8	0	output_4_description
5	9 - 10	0	output_5_description
6	11 - 12	0	output_6_description
7	13 - 14	0	output_7_description
8	15 - 16	0	output_8_description
9	17 - 18	0	output_9_description
10	19 - 20	0	output_10_description
11	21 - 22	0	output_11_description

Inputs

Index	Pin Number	State	Description
1	1	0	input_1_description
2	2	0	input_2_description
3	3	0	input_3_description
4	4	0	input_4_description
5	5	0	input_5_description
6	6	0	input_6_description
7	7	0	input_7_description
8	8	0	input_8_description
9	14	0	input_9_description
10	15	0	input_10_description
11	16	0	input_11_description
12	17	0	input_12_description
13	18	0	input_13_description
14	19	0	input_14_description
15	20	0	input_15_description
16	21	0	input_16_description
17	22	0	input_17_description
18	23	0	input_18_description
19	24	0	input_19_description
20	30	0	input_20_description
21	31	0	input_21_description
22	32	0	input_22_description