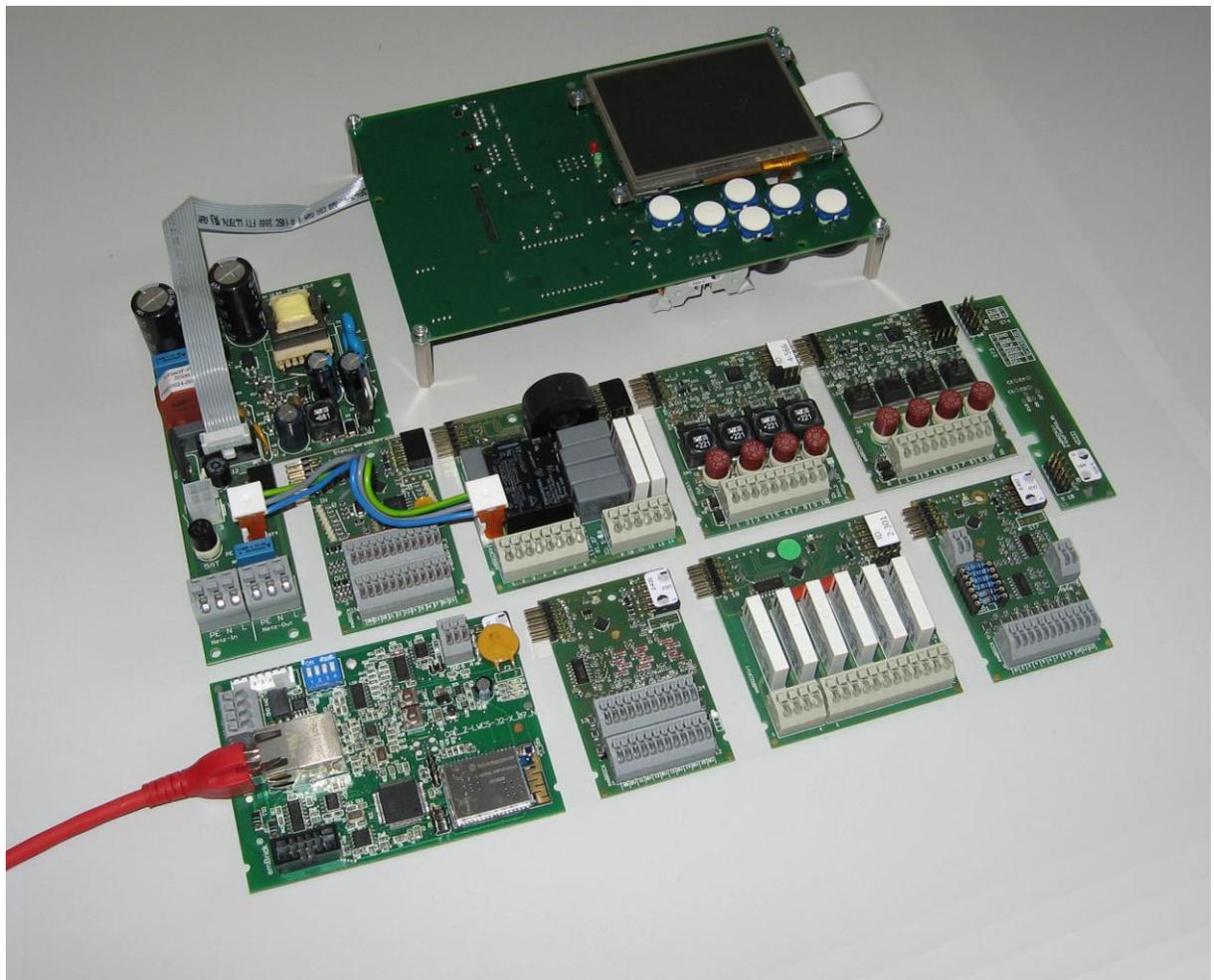


emBRICK[®] - EPC

Embedded Patch-board Controller



FAQ Manual

Rev. 9p

emBRICK® is developed and supported by



IMACS GmbH

Alfred-Nobel-Straße 2

D – 55411 Bingen am Rhein

www.imacs-gmbh.com

www.embrick.de

support@embrick.de

Hotline: +49 (0) 7154 80 83 - 15

IMACS GmbH reserves the right to make changes without further notice to any products herein. IMACS GmbH makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does IMACS GmbH assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in IMACS GmbH data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. IMACS GmbH does not convey any license under its patent rights nor the rights of others.

copyright © IMACS GmbH 2016. All rights reserved.

Reproduction, in part or whole, without the prior written consent of IMACS GmbH is prohibited.

Content

1. The emBRICK® Mission	3
1.1 Typical Applications.....	3
1.2 Basic Characteristics.....	4
1.3 Available Hardware Products	4
1.4 Available Host Platforms, Connectivity.....	4
1.5 Available Programming Platforms	4
2. Introduction.....	5
2.1 About this Manual.....	5
2.2 References / Manual Overview.....	5
3. Basic System FAQs	6
3.1 Topology.....	6
3.2 Performance.....	6
3.3 Protocol.....	7
3.4 Immunity	7
3.5 Norm.....	8
4. Hardware FAQs.....	9
4.1 Power Consumption	9
4.2 Electrical coupling.....	9
4.3 Requirements to Master CPU	9
5. Programing FAQs.....	10
5.1 RemoteMaster adaptions	10
5.2 LocalMaster	10
5.3 Own Slave.....	10
5.4 Programming Languages	10
6. Licensing FAQs.....	11
7. Error FAQs	12
7.1 LAN.....	12

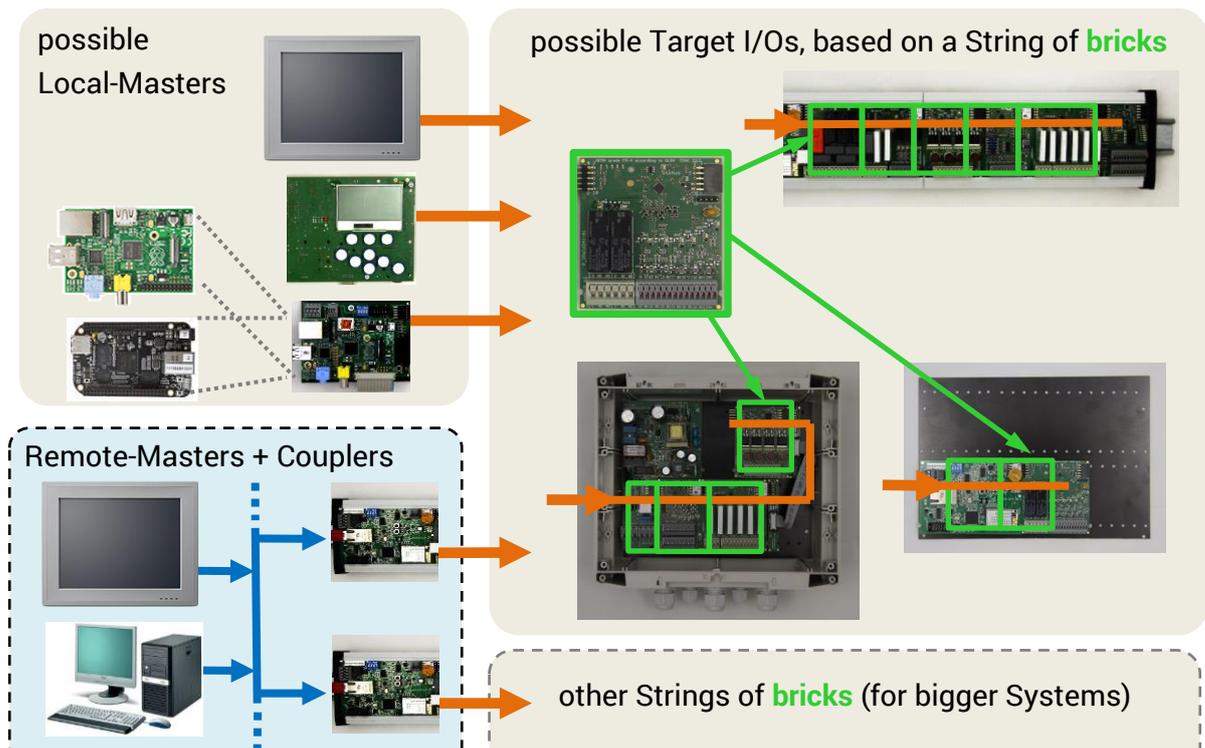
1. The emBRICK® Mission

The mission of *emBRICK®* is an **open** and **free** I/O system to ...

build **compact** and **industrial suited** electronic **control systems**
by **assembling** small **existing/own** embedded **boards** (**bricks**) ...

... via a SPI-based **local interface** and optional **remote buses** (LAN, WLAN, CAN, RSxxx, ...).

We call this new class of controllers simple **EPC** (= **E**mbedded **P**atch-board **C**ontroller).



emBRICK® combines in a perfect way the **cost-efficient** and **tailored** characteristics of a dedicated embedded system with the **ready to use** and **flexibility** of a PLC system.

To ensure a high acceptance, it is an open and free system. I.e. besides buying existing devices, everyone can develop his own components to realize easily his individually tailored, cost-efficient and industrial-suited measure and control system.

1.1 Typical Applications

- Small, medium and large size **measure and control systems**
- **Sectoral purpose**, with direct sensor/actor interface
- **Autonomous single box** control solutions i.e. with HMI and communication interfaces
- **Rapid hardware prototyping** system for control and measuring applications
- **PLC replacement** (i.e. with a Soft-PLC, IPC or an embedded controller)
- Medium and large size **distributed IO-systems** (i.e. building automation)
- Physical front-end for **IoT** (Internet of Things)

For more details see *Product_Catalogue* and *Application_Manual*.

1.2 Basic Characteristics

- **free** - also for commercial use in own appliances (for pure EMS with a license fee)
- **open** - supplying reference schematics, protocol source code, samples and starter kits
- **adaptable to all systems**, using common, low cost standard μ Cs/components
- **half ... third price** compared to common control systems (complete system view)
- scalable local and remote topologies, **1 ... >1000 I/Os**, up to **1ms update**, deterministic
- **low own power** consumption, average 50mW/slave module in operation (outputs inactive)
- global and sector specific modules for **direct connection** of various **sensors and actors**
- **easy installation**, no configuration necessary, simple plug modules together and use
- works with / programmable by **various established**, well known **platforms / languages**

1.3 Available Hardware Products

Beside own developments, currently the following components are available from IMACS:

Slave-Modules > 50 different modules for the sectors: General Purpose, Building Automation, Process Control (Safety, Medical/Analytics planed)

Master boards Core: Cortex-M3/4, ARM9/11, PIC24/32; HMI: 128x64 ... WVGA

Adaption boards for LAN, WLAN, CAN, RSxxx, Raspberry Pi, Beaglebone Black

Appliances / Enclosures ready Single Box Controller for and top-het rail and wall mounting

Starterkits for MSVC, CODESYS, Raspberry Pi, Beaglebone Black

1.4 Available Host Platforms, Connectivity

emBRICK® can be adapted to all platforms with almost every footprint/performance. For master units, currently the following system implementations are available (others planed):

Computer platforms PC, Embedded-PC, Module-PC, Raspberry Pi, Beaglebone Black

μ Controller platforms ARM-Ax, ARM-Cortex-Mx, Microchip PIC24 / PIC32

Host Interfaces Ethernet, CAN, RS232, RS485

Wireless Interfaces WLAN

1.5 Available Programming Platforms

emBRICK® can be programmed by various systems, languages and IDEs (integrated development interface). Currently for master units the following systems are available (others planed):

OS / RTOS Windows, Linux, FreeRTOS, proprietary

Programming languages C, C++, IEC61131, Model-based (by implementing UML)

Model-based / Soft-PLC CODESYS, radCASE, Enterprise Architect

C/C++ IDEs MSVC, Cocox (GCC), MPLab (Microchip), Geany (Raspberry Pi), every other C/C++ IDE

2. Introduction

2.1 About this Manual

This manual contains answers to the most frequently asked questions about *emBRICK®* and its usage.

2.2 References / Manual Overview

For *emBRICK®* and *brickBUS®* the following documents are available. Before reading this document it is recommended to read them in the given order:

- [System Manual](#)(*embrick_System-Manual_#.pdf*) ... the basic system manual that contains the idea, the intention and the basic technical concept of *emBRICK®/ brickBUS®* like mechanics, electronics and communication protocol. It includes the glossary for all other documents.
- [Application Examples](#)..... (*emBRICK_Application-Examples_#.pdf*) ... overview of typical *emBRICK®* device configurations and sample constellations for different industrial applications. It gives an idea how to use *emBRICK®* as an alternative to a normal PLC or an individual PCB / embedded system.
- [Product Catalogue](#) (*emBRICK_Product-Catalogue_#.pdf*) ... contains the overviews and detailed datasheets of all IMACS-available *emBRICK®* components and products. This includes electrical and mechanical characteristics, terminal assignment and notes about their usage.
- [Programmers Manual](#) (*emBRICK_Programmers-Manual_#.pdf*) ... is the manual for application software programmers when using established programming systems like Embedded-IDEs, Soft-PLCs, CASE-Tools but also native C/C++-coding.
- [FAQ Manual](#)..... (*emBRICK_FAQ-Manual_#.pdf*) ... contains answers to the most frequently asked questions about *emBRICK®* and its usage.
- Developers Manual..... is the manual for system developers, who like to create their own slave modules or master adaptations. It includes all technical details specifications of *brickBUS®* and also sample schematics and code samples of the software stacks. This document is only available on request from IMACS GmbH and needs the agreement on the *emBRICK®* free license conditions. Please contact support@embrick.de.

3. Basic System FAQs

3.1 Topology

Q B1.1: [2015-03-15] What topologies are possible for the remotebus an localbus?

A: The remotebus can run with star and tree topology, maybe a switch is needed. For the localbus only a daisy chain is useable.

Q B1.2: [2015-03-15] How many nodes are possible?

A: With the actual CAE_Z-LWCS-M32-0# as localmaster up to 10 different nodes are possible. The number of nodes is limited by the amount of adjustable IP addresses.

Q B1.3: [2015-03-15] How many slaves are allowed in each node?

A: All *Local-Masters* can support up to 32 slaves.

Q B1.4: [2015-03-15] Is it possible that a master can act as a *Remote-Master* and *Local-Master*, for more than one node?

A: Yes. The Master can control node A directly and node B over LAN or CAN.

Q B1.5: [2015-04-02] Which Select Line is the one facing the Master?

A: On startup, the Slave has both sides set up as inputs. The first side that gets pulled from GROUND to HIGH is considered the side facing the Master, the other side is considered the side opposing the Master. The side opposing the Master has then to be configured as an Output. It is pulled to HIGH once the Slave receives the command "Select Next Slave" (Command 0x03).

3.2 Performance

Q B2.1: How often does the Local-Master update the *emBRICK®*-Modules?

A: There is no recommended value or range but typically, every module in the *brickBUS®* gets updated every 10ms. For very long busses with large modules (I/O-size), the *Local-Master* might not be able to update the modules that frequently. On the contrary, very short busses with small modules could be updated way more often.

When using LAN-Communication, the *Local-Master* will start a new update-cycle on the arrival of new data via TCP/IP. This eliminates heterodyne beats. When using CAN-Communication, the frequency of the update-cycles can be adjusted and are running independently from the CAN-Communication.

Q B2.2: Can I control the date for an *emBRICK®*-update-cycle while using LAN?

A: When using LAN-Communication, the *Local-Master* will start a new update-cycle on the arrival of new data via TCP/IP. This eliminates heterodyne beats.

Q B2.3: Can I control the date for an emBRICK®-update-cycle while using CAN?

A: When using CAN-Communication, the frequency of the update-cycles can be adjusted and are running independently from the CAN-Communication.

Q B2.4: How long does it take till an Output is set after sending the corresponding Data via LAN?

A: As soon as the TCP/IP-Packet reaches the *Local-Master*, the incoming Data will be sent to the *brickBUS*. Thus, it takes about as long as the *Local-Master* needs to update all Modules. The time needed for this action will typically not exceed 10ms.

On the Module itself, the Data will need additional time, depending on the type of Output, until a change in output levels takes place.

Q B2.5: How long does it take till an Input reaches the *Remote-Master*, when using LAN-Communication?

After receiving Data from the *Remote-Master*, an update-cycle will be started. When the update-cycle has been finished, the *Local-Master* sends the received Data to the *Remote-Master*. The time needed for this action will typically not exceed 10ms.

3.3 Protocol

Q B3.1: Does the protocol posses an error detection mechanism?

A: Yes the protocol includes a checksum in each communication block.

Q B3.2: What happens with a corrupted message?

A: The message will be ignored. The Master will try to contact the Slave again. After five attempts, the Master will continue to the next Slave.

Q B3.3: Is the communication encrypted ?

A: No encryption has been implemented yet.

3.4 Immunity

Q B4.2: [2015-04-02] When is the brickBUS considered malfunctioning?

A: The master periodically sends out messages to which the addressed Slave will respond. If the addressed slave does not respond or responds with a wrong checksum, then this transmission is considered faulty. The master will try to get a proper response from the slave up to a maximum of 5 tries in total. If the slave was unable to respond properly, the master will carry on with the next slave. The brickbus is considered malfunctioning if this behavior appears frequently. The threshold until the Master considers the brickBUS to be malfunctioning depends on implementation. Under no circumstances is it acceptable to develop Slave modules with the philosophy that one doesn't have to respond to every message. A Slave module has to have the ability to receive the header of every message and respond if addressed.

Q B4.3: [2015-04-02] What happens if the Master considers the brickBUS to be malfunctioning?

A: The Master will pull his Select-Pin to GROUND. The next Slave Module in line will recognize this on the Select-Pin that faces the Master and subsequently pull its Select-Pin opposite to the Master to Ground. This process will continue until the last Slave Module in the emBRICK-string. To prevent this process from happening due to electromagnetic disturbances, this Pin has to be secured by Hardware (with a Low-Pass for $f_c = 30..100kHz$) and by Software (make sure that that the Select-Pin stays GROUND for atleast $30\mu s$). The Slave receiving this signal will then pull its Select-Pin to GROUND for atleast $100\mu s$ before it resets itself.

Q B4.4: [2015-04-02] Can the Slave pull its Select-Pin to Ground when it notices that a lot of messages are faulty?

A: In protocol-version 11 the Slave will reset itself if it receives zero messaged addressed to him in the timeframe of one second. This behaviour will be configurable in protocol-version 12. When resetting, naturally all outputs will be GROUND. Application outputs can be set to any default value to prevent equipment damage.

3.5 Norm

Q B5.1: [2015-04-13] What norms have to be observed, when creating a new slave?

A: this depends on the usecase of the product. We internally used a list with many relevant items. They have to be customised from the developer for the product.

4. Hardware FAQs

4.1 Power Consumption

Q H1.1: How much power does a Slave require ?

A: Power Consumption for every Slave is listed in the Product-Catalogue.

Q H1.2: How much power is allowed to use for sensor/actors on a Slave ?

A: It is recommended to use for the supply of external devices only the 24V. In need of other voltage levels e.g. 12V, 5V or 3,3V they must be created out of 24V on each slave. If more power is needed a separate clamp for supply input is required.

Q H1.3: How thick have the 24V and 3,3V track to be on my own pcb?

A: On our boards the two tracks are realized with 0,6mm so they are ready up to 2A. But this is not recommended because of the contact resistance through all moduls.

4.2 Electrical coupling

4.3 Requirements to Master CPU

Q H3.1: Is a real SPI unit needed?

A: An application with a real SPI is much faster. If your application allows a slower communication, you can also emulate a SPI via GPIO. However, every Module has to be updated every 50ms. Each Byte has to be sent out at a stretch. Between two Bytes, small gaps may exist. None of those gaps may exceed the Sync-Gap. See Developer Documentary for further details.

Q H3.2: How much RAM is needed to implement a full emBRICK stack?

A:

5. Programming FAQs

5.1 RemoteMaster adaptations

Q P1.1: Welche Programmiersysteme kann ich verwenden?

A:

Q P1.2: I want to program my own remote master, what is to do?A: In the *Programmers Manual* is the specification for the communication via LAN and CAN.

5.2 LocalMaster

Q P2.1: Which requirements a μ C must meet?

A: a μ C should have a SPI-module. However, if your application allows a slower communication, you can also emulate a SPI via GPIO. Every Module has to be updated every 50ms. Each Byte has to be sent out at a stretch. Between two Bytes, small gaps may exist. None of those gaps may exceed the Sync-Gap. See Developer Documentary for further details.

A clockspeed above 64 MHz is recommended although not necessary, depending on your timing requirements.

5.3 Own Slave

Q P3.1: [2015-04-02] When can I send a broadcast-message?

A: Always. However, only Slaves that have their Select-Line facing the Master on HIGH may receive and respond to messages.

5.4 Programming Languages

6. Licensing FAQs

Q L1.1 We want to use emBRICK in our products, what are the conditions for commercial use?

A: You can buy the emBRICK modules from us and mount them into your products or your system. In this usecase you can freely use it.

But when you want to produced the modules on your own please contact us support@embrick.de.

7. Error FAQs

7.1 LAN

Q E2.1: What is wrong if nodes are not detected?

A: At first please try it with the tool NetBRICK it lists all found *local-master* with their IP addresses. Is the missed node here found?

Then the eB_LAN.conf file is wrong other wise the DIP switches are in a wrong constellation or your Ethernet port is wrong configurated.

