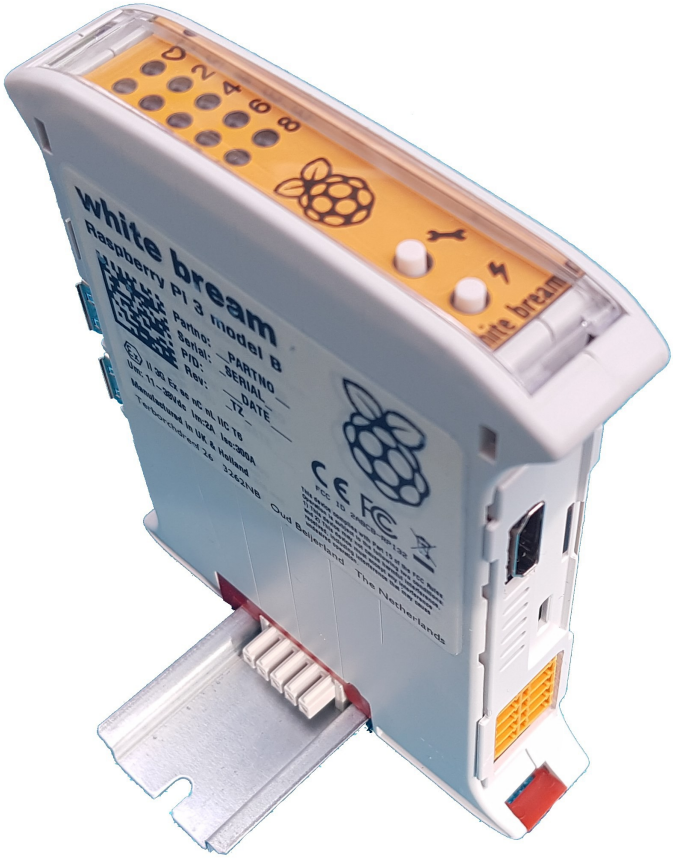
white bream

CAN•net

DIN-rail
Raspberry Pi



White Bream Oud-Beijerland The Netherlands https://whitebream.com			
Description:	Reference manual		E109RP001 Manual Raspberry DIN.odt
Project:	E109		*E109RP001*
Status:	Draft	Pages: 23	

Table of Contents

1 Preface

1.1 Disclaimer

White Bream products are not authorized for use in, or in connection with surgical implants, or as critical components in any medical or nuclear, or aircraft, or other transportation devices or systems where failure to perform can reasonably be expected to cause significant injury to the user, without the express written approval of an executive officer of White Bream. Such use is at buyer's sole risk, and buyer is responsible for verification and validation of the suitability of products incorporated in any such devices or systems. Buyer agrees that White Bream is not liable, in whole or in part, for any claim or damage arising from such use and shall have no obligation to warranty such products. Buyer agrees to indemnify, defend and hold White Bream harmless from and against any and all claims, damages losses, costs, expenses and liabilities arising out of or in connection with buyer's use of White Bream products in such applications to the extent buyer has not obtained the express written approval of an executive officer of White Bream.

1.2 Trademarks & copyrights

Throughout this manual, the trade names and trademarks of various companies and products may have been used, and no such uses are intended to convey endorsement of or other affiliations with this manual or product. Any brand names or product names used within this manual are trademarks or registered trademarks of their respective holders.

1.3 Warranty

This product is warranted to be in good working order for a period of two years from the date of purchase. Should this product fail to be in good working order at any time during this period, we will, at our option, replace or repair it at no additional charge except as set forth in the following terms. This warranty does not apply to products damaged by misuse, modifications, accident or disaster.

1.4 Liability

White Bream assumes no liability for any damages, lost profits, lost savings or any other incidental or consequential damage resulting from the use, misuse of, or inability to use this product. White Bream will not be liable for any claim made by any other related party.

2 Description

This module integrates the Raspberry Pi 3 with a rugged power supply, various real-world inputs and outputs and a shutdown + watchdog circuit. All ready to use in a DIN-rail mounted enclosure with convenient T-Bus connection. It is ideally suited for industrial applications involving CANopen, DeviceNet or generic CAN.

2.1 Specifications:

- Raspberry Pi 3 model B+ (included with 69-109-30X)
- 12-36V power supply 5V/4A
- Active 'eFuse' + 2.5A traditional backup fuse
- CAN-bus and RS232 interfaces, 36V ruggedized
- 4 Analog inputs with 61.44V analog range
- 4 Low-side protected outputs, IDC=1.2A, VDS=60V, RDS=500mΩ
- Soft-power control and system watchdog
- Bi-color LED indicators for power, watchdog and 8× general purpose, the latter controlled via an 16-bit I²C I/O expander
- 2× Front panel buttons, one of which doubles as soft power on
- 2× 10-way 3.81mm spring terminals block for GPIO
- 5 terminal pins wired to internal 0.1″ headers
- DIN rail enclosure 101x119.5mm, width 22.5mm
- T-Bus power & CAN connection (DIN socket connector included)

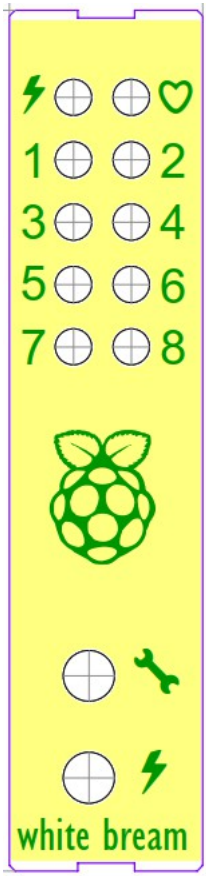
The T-Bus system provides a neat and convenient method of connecting multiple DIN-rail mounted device to power and a CAN-bus data network. This Raspberry Pi module has provisions for both.

The interface board contains an 8-channel 12-bit MAX11615 analog to digital converter from Maxim Integrated. This ADC is connected to the I²C bus and connects with the 4 inputs via 29:1 resistive dividers, giving about 60V full-scale input range. It is also connected to VIN, VDD (5V), VCC (3.3V) and input current monitor.

Power input is diode OR'ed from the T-bus connection and two regular power input terminal contacts. That power is then converted by a high efficiency 5V / 20W regulator to power the PI computer. Those 20W give plenty of margin for powering external USB peripherals. This power supply is managed by the remote control input line and by the watchdog circuit.

The HDMI port from the Raspberry is rerouted to the side of the case.

3 Operation



3.1 Indicators

#	Function	Description
ϟ	Power	Green: 3.3V from R-Pi Red: Powerfail from 5V PSU
♡	Watchdog	Green: toggle with each refresh
1	User 1	Please refer to section 6.3, Program front panel LEDs
2	User 2	"
3	User 3	"
4	User 4	"
5	User 5	"
6	User 6	"
7	User 7	"
8	User 8	"

3.2 Controls

The two buttons on the front-panel are connected to GPIO05 and GPIO06.

#	Function	I/O	Description
ϟ	Power	GPIO05	Button activates power when the device is off. When on, it depends on software functionality.
🔧	User	GPIO06	

3.3 Remote control

Pin 2 of the bottom I/O block provides a remote control input (RMT). When the RMT input received input signal, the power supply of the device will switch on. To ensure that the device switches on whenever power is applied, tie this input to the power rail. For power on functionality, it depends on th eimplementation of the software as described in 6.7, Watchdog.

4 Specifications

4.1 Electronic

Parameter	Min	Max	Unit
Nominal input voltage	6	36	V
Input voltage surge	-60	60	V
Input current limiter		2	A
Input fuse		2.5	A-F
Power dissipation		tbd	W
CAN transceiver signaling level		3.3	V
CAN transceiver signaling rate	10	1000	kbps
RS232 receive input level high		12	V
RS232 receive input level low		-12	V
RS232 transmit output level high			V
RS232 transmit output level low			V
RS232 signaling rate		250	kbps

4.2 Connections

Input power from the T-Bus and the two terminal block connections are OR-ed via 60V/3A schottky diodes.

4.2.1 DIN-rail T-Bus

Pin numbers are counted starting from the bottom side of the DIN rail.

#	Name	Description	Signal
1	CAN-L	CAN L-signal	
2	CAN-H	CAN H-signal	
3	-V	Bus ground	0V
4	+V	Bus power supply	0-36VDC
5	SHIELD	Cable shield	0V

4.2.2 CAN-Bus

Pin # 1 is next to the HDMI connector.

#	Name	Description	Signal
1	CAN-L	CAN L-signal	
2	CAN-H	CAN H-signal	
3	-V	Bus ground	0V
4	+V	Bus power supply	0-36VDC
5	SHIELD	Cable shield	0V

4.2.3 I/O 1 (bottom-side)

With the serial number label facing down, pin # 1 is bottom at the side of the HDMI connector. Pin # 6 is on the top row, same side as #1.

#	Name	Description	Signal
1	VBB	Power supply in	0-36VDC
2	RMT	Remote control in	0-36VDC
3	GND	Ground	0V
4	TXD	RS232 Transmit	±4~6V
5	RXD	RS232 Receive	±3-12V
6	NC1	No connect to J6 pin 4	
7	NC2	No connect to J6 pin 5	
8	NC3	No connect to J6 pin 6	
9	NC4	No connect to J6 pin 7	
10	NC5	No connect to J6 pin 8	

4.2.4 I/O 2 (top-side)

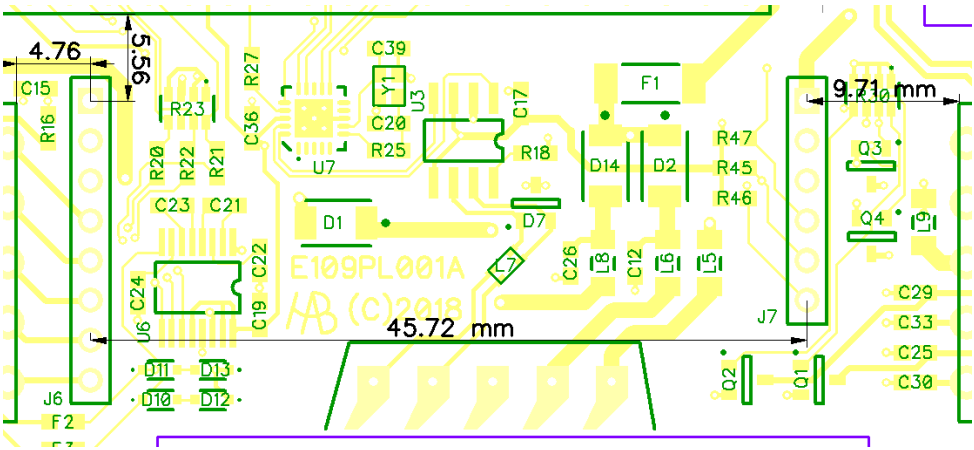
With the serial number label facing down, pin # 1 is at the bottom twoards the foot of the enclosure. Pin # 6 is on the top row, same side as #1.

#	Name	Description	Signal
1	OUT1	Open drain output	60V 1.3A
2	OUT2	Open drain output	60V 1.3A
3	GND	Ground	0V
4	AIN1	Analog input	Max 61V
5	AIN2	Analog input	Max 61V
6	OUT3	Open drain output	60V 1.3A
7	OUT4	Open drain output	60V 1.3A
8	GND	Ground	0V
9	AIN3	Analog input	Max 61V
10	AIN4	Analog input	Max 61V

4.2.5 USB, HDMI & Ethernet

Please refer to Raspberry Pi specifications.

4.3 Internal connections



Refer to drawing above for the maximum dimensions of the internal PCB and pin spacings. The terminal strip spacings are standard 2.54mm/0.1". The carrier board contains receptable headers so the expansion board needs to have pin headers.

4.3.1 8-Pin header (J6, left)

#	Name	Description	Signal
1	VBB		0V
2	GND		0-38DC
3	nc		0V
4	NC1	No connect to I/O1 pin 6	
5	NC2	No connect to I/O2 pin 7	
6	NC3	No connect to I/O3 pin 8	
7	NC4	No connect to I/O4 pin 9	
8	NC5	No connect to I/O5 pin 10	

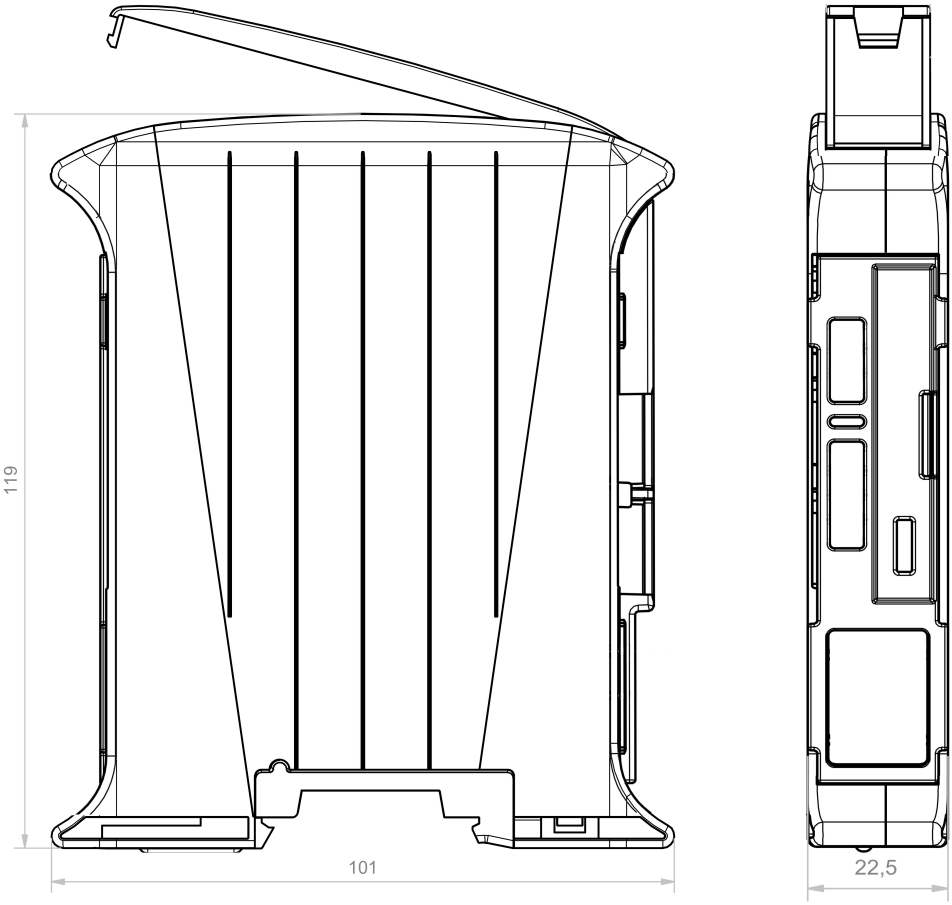
4.3.2 6-Pin header (J7, right)

#	Name	Description	Signal
1	VDD	Carrier power supply	5.0V 4A
2	GND	Ground	0V
3	VCC	Raspberry power supply	3.3V
4	IRQ	I2C Interrupt / SMB Alert	LVTTL OD
5	SCL	I2C Clock	LVTTL OD
6	SDA	I2C Data	LVTTL OD

4.4 Environmental

Parameter	Min	Max	Unit
Operating temperature range	-20	+50	°C
Non-operating temperature range	-40	+120	°C
Humidity	0	100	%
Ingress Protection	IP20		

4.5 Dimensions



5 Installation

<TODO>

5.1 Wiring

<TODO>

5.2 Fuse

The device contains a two-stage over-current protection. Level 1 consists of a semiconductor based current limiting and overload protection device, set at 2A. Level 2 is a 2.5A fast fuse, which is soldered onto the PCB..

5.3 Maintenance and re-use

WARNING: Do not open the enclosure, connect or disconnect cables when the circuit is energized and explosive atmosphere is present!

6 Programming

6.1 Activate I2C and SPI interfaces

Before the CAN interface and I2C devices can be used, those serial interfaces need to be activated in the Raspberry Pi configuration system, that is SPI and I2C:

```
sudo raspi-config
```

After this configuration, 'sudo i2cdetect -y 1' should list two devices;
the MCP23017 at address 20 and the MAX11615 ADC at address 33:

```
sudo i2cdetect -y 1
```

6.1.1 Enable I2C clock stretching

Edit /boot/config.txt, find the line that says “dtparam=i2c_arm=on”.
Somewhere below that line, add the following line:

```
# Clock stretching:
dtoverlay=i2c-gpio,i2c_gpio_delay_us=10
```

More details can be found at https://learn.adafruit.com/circuitpython-on-raspberrypi-linux/i2c-clock-stretching

6.1.2 Install I2C libraries

```
apt-get install i2c-tools python-smbus
```

6.2 Activate CAN interface

More details at https://www.raspberrypi.org/forums/viewtopic.php?t=141052

Edit /boot/config.txt and enable mpc2515 & bcm2835-spi overlays.

```
dtoverlay=mcp2515-can0,oscillator=16000000,interrupt=25
dtoverlay=spi-bcm2835
```

After reboot ls /sys/bus/spi/devices/spi0.0/net should list 'can0'

```
ls /sys/bus/spi/devices/spi0.0/net
```

6.2.1 Install CAN libaries

```
sudo apt-get install can-utils
```

6.2.2 Initialize CAN interface and listen to traffic

```
;Setup CAN interface, here we use 1Mbps baudrate:
sudo ip link set can0 up type can bitrate 1000000

;Show incoming traffic:
candump can0
```

6.3 Program front panel LEDs

Below is an example to program the front panel LEDs using the i2cset command line function:

```
i2cset -y 1 0x20 0x14 0xFF b ;initialize LEDs 1-4 (off)
i2cset -y 1 0x20 0x15 0xFF b ;initialize LEDs 5-8 (off)
i2cset -y 1 0x20 0x00 0x00 b ;activate LED outputs 1-4
i2cset -y 1 0x20 0x01 0x00 b ;activate LED outputs 5-8

i2cset -y 1 0x20 0x14 0x55 b     ;LEDs 1-4 green
i2cset -y 1 0x20 0x15 0xAA b     ;LEDs 5-8 red

i2cset -y 1 0x20 0x14 0xEF b     ;LED 3 red
i2cset -y 1 0x20 0x14 0xDF b     ;LED 3 green

i2cset -y 1 0x20 0x15 0xFE b     ;LED 5 red
i2cset -y 1 0x20 0x15 0xFD b     ;LED 5 green
```

6.4 Read front panel buttons

<TODO>

6.5 Reading analog inputs

The python example reads all 8 analog input channels from the ADC and converts them into their real values in volts (except for last channel, which is in mA).

```
import smbus
import time

bus = smbus.SMBus(1)
address = 0x33
val = [0,0,0,0,0,0,0,0]
ref = 4095 / 2.048

bus.write_byte(address. 0xDA)
```

```
while True:

 dat = bus.read_i2c_block_data(address, 0xF, 16)

 val[0] = (30 / ref) * ((dat[0]&0xF)*256 + data[1])
 val[1] = (30 / ref) * ((dat[2]&0xF)*256 + data[3])
 val[2] = (30 / ref) * ((dat[4]&0xF)*256 + data[5])
 val[3] = (30 / ref) * ((dat[6]&0xF)*256 + data[7])
 val[4] = (5 / ref) * ((dat[8]&0xF)*256 + dat[9])
 val[5] = (30 / ref) * ((dat[10]&0xF)*256 + data[11])
 val[6] = (2 / ref) * ((dat[12]&0xF)*256 + data[13])
 val[7] = (78.25*12.100 / ref) * ((dat[14]&0xF)*256 +
                                                  dat[15])

 print ", ".join(str(float(int(100 * v)) / 100)
                                        for v in val)
 time.sleep(2)
```

6.6 Program open-drain outputs

<TODO>

6.7 Watchdog

The soft-power and watchdog are enabled by a small Python script that is run from rc.local. This script interfaces with the circuit using GPIO4. Most of the time, this signal is configured as input, so the script can monitor the status of the remote-control signal. Every 250 milliseconds, the GPIO is momentarily changed to an output having opposite logic polarity as the actual input signal. This overrides the fairly high impedance input signal, and results in the pulse for the watchdog controller (a tiny PIC10F322) to reset the guard time. After power-on the watchdog timer sleeps for 4 seconds to prevent signal bounce from starting this shutdown watchdog. After these 4 seconds the watchdog waits for the first pulse before the actual watchdog is enabled. This enables an arbitrary long boottime, or no watchdog script at all. Actually, without the script, the power supply works almost as any other power supply, which runs until the input power is cut.

Install the script by downloading it to your etc directory and adding it to rc.local:

```
cd /etc
sudo wget https://whitebream.com/files/project/e109.py
sudo nano /etc/rc.local
```

Just above the “exit 0” line, add the following line:

```
python /etc/e109.py &
```

It is also possible to start the system with a short pulse, and initiate shutdown with a second pulse. This way, the Pi works just like a regular PC with a soft power switch. The

start pulse should last no longer than until the start of the watchdog script. The stop pulse should last no longer than the actual shutdown of the system, or the system will simply restart.

When the Raspberry Pi shuts down, the script is stopped, hence the pulses stop as well. After 16 seconds, this results in a reset pulse from the watchdog monitor. That reset pulse disables the 5V regulator, which shuts the Pi down. After the 5V has dropped enough to disable the watchdog IC, the power control depends on the Remote input signal. If still active, the Pi will restart, otherwise it stays off until the next activation of the button or the Remote input signal.

Note that this power control method does not support the reboot command. The hardware will hard reset the Pi after 16 seconds of missing watchdog reset pulses, and that is not enough for a reboot of the Raspberry Pi. As a substitute, an ordinary shutdown will result in a reboot if the RMT input signal is still active.

CPU load of the script is less that measurable; both ps -p <pid> -o %cpu and top -p <pid> display a CPU usage of 0.0%, with top showing 0.3% incidentally.

Scipt code:

```
import RPi.GPIO as GPIO
import time import sleep
import os import system

# Using the GPIOxx numbering scheme
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(4, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(5, GPIO.IN)
prev_input = 0
prev_button = 0

while True:
  # Read the remote signal from the screw terminal
  input = GPIO.input(4)
  button = GPIO.input(5)
  if ((prev_input and not input) or (not prev_button
              and input)):
    os.system("sudo shutdown -h now")
  prev_input = input
  prev_button = input

  # Output a pulse (~100usec) for the watchdog.
  # This pulse is opposite level of the remote
  # signal so that the watchodg IC will detect
  # a toggle.
  GPIO.setup(4, GPIO.OUT)
  GPIO.output(4, (not input))
```

```
GPIO.setup(4, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
# Sleep a while
time.sleep(0.25)
```

6.8 Miscellaneous

Stock settings of the HDMI port tend to make displays loose sync periodically. This causes the screen to go blank momentarily. This can be fixed by adjusting the HDMI signal levels 'config_hdmi_boost=9' in /boot/config.txt:

```
sudo vi /boot/config.txt
```

6.9 Serial port

Using the serial port on the Raspberry Pi 3 comes with some issues because it's hardware serial port is redirected to the Bluetooth interface.
https://spellfoundry.com/2016/05/29/configuring-gpio-serial-port-raspbian-jessie-including-pi-3/

<TODO>

7 Ordering information

Partno	Description	Revision
69-109-301	DIN-rail Raspberry Pi 3 model B+ module, complete with Pi & connectors	A, Sep 14, 2018
80-942-922	DIN-rail T-bus connector 22.5mm (10pcs)	

7.1 Hardware revision info

Rev	Date	Changes
0	Jun 28, 2018	Prototype
A	Sep 14, 2018	Initial version

8 Document revisions

8.1 Rev 0.1 (October 31, 2018)

Ref	Description
-	Initial version

Annex A: Declaration of Conformity for CE / ATEX

The manufacturer hereby declares that this product is according to the requirements of ~~annex II of the EEC directive 2014/34/EC regarding ATEX and~~ directive 2014/30/EU regarding electromagnetic compatibility (EMC), directive 2014/35/EU regarding low voltage equipment (LVD) and directive 2011/65/EU regarding RoHS.

Manufacturer, facility:	White Bream L.J. Costerstraat 13d 3261LH, Oud-Beijerland The Netherlands
Product:	DIN-rail Raspberry Pi
Model:	69-109-301
~~**ATEX Marking:**~~	~~II 3GD~~ ~~Ex nA IIB T5 Gc X~~ ~~Ex tc IIIC T100°C Dc X IP20~~ Tamb -30° to +50°C WARNING: DO NOT CONNECT OR DISCONNECT WHEN ENERGIZED
~~**ATEX Certificate:**~~	~~N/A. Comformity verified according annex VIII of directive 2014/34/EU (internal production control) Our IecEx checklists are available on request.~~
CE & RoHS Marking:	

continued on next page

~~This product has been found in comformity with directive 2014/34/EU (ATEX) by testing and verification with the following standards:~~

- ~~**EN 60079-0:2012/A11:2013** Explosive atmospheres – Part 0: General requirements~~
- ~~**EN 60079-15:2010** Explosive atmospheres - Part 15: Equipment protection by type of protection "n"~~
- ~~**EN 60079-31:2014** Explosive atmospheres - Part 31: Equipment dust ignition protection by enclosure "t"~~

This product has been found in comformity with directive 2014/30/EU (EMC) by testing and verification with the following standards:

- **EN 61000-6-1:2007** Electromagnetic compatibility (EMC) - Part 6-1: Generic standards - Immunity for residential, commercial and light-industrial environments
- **EN 61000-6-3:2007/A1:2011/AC:2012** Electromagnetic compatibility (EMC) - Part 6-3: Generic standards – Emission standard for residential, commercial and light-industrial environments

This product has been found in comformity with directive 2014/35/EU (LVD) by testing and verification with the following standards:

- **EN 61010-1:2010/C1:2011** Safety requirements for electrical equipment for measurement, control, and laboratory use - Part 1: General requirements
- **EN 62368-1:2014/A11:2017** Audio/video, information and communication technology equipment - Part 1: Safety requirements

This product has been found in comformity with directive 2011/65/EU (RoHS) by testing and verification with the following standards:

- **EN 50581:2012** Technical documentation for the assessment of electrical and electronic products with respect to the restriction of hazardous substances

October 31, 2018
Henk Bliek, White Bream, Owner

Annex B: Declaration of Conformity for FCC

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna
- Increase the separation between the equipment and receiver
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected
- Consult the dealer or an experienced radio/TV technician for help

CAN•net

white bream
L.J. Costerstraat 13d
3261LH Oud-Beijerland
The Netherlands

https://whitebream.com
https://whitebream.nl

