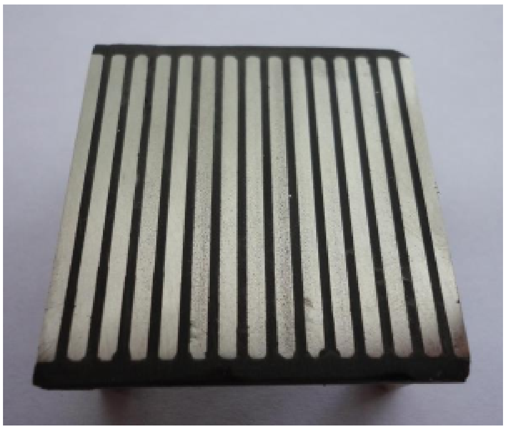
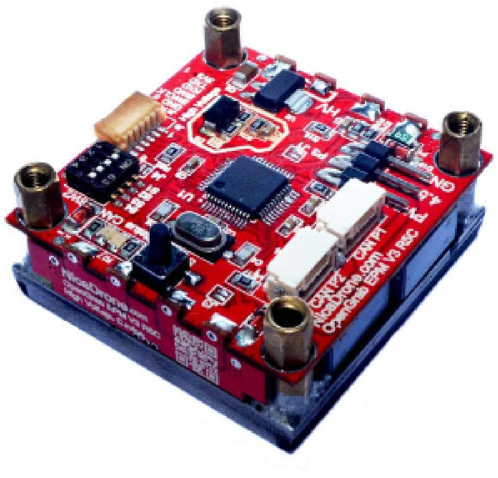
OpenGrab EPM v3

Overview

The OpenGrab EPM v3 is an electropermanent magnet, combining the advantages of electro and permanent magnets. The device creates a very strong magnetic contact with a ferrous target. It supports UAVCAN, RCPWM and push button operation. OpenGrab EPM v3 has been developed by NicaDrone in cooperation with Zubax Robotics.





Theory of operation

> **Electric shock hazard**
> The device poses an electric shock hazard. Do not touch exposed parts of the circuit while the magnet is operating.

The NXP LPC11C24 MCU drives a MOSFET connected to a transformer in a flyback configuration to charge the main PET capacitors up to 475 V. A thyristor bridge is used to discharge the capacitor in either direction through the winding inside the AlNiCo material. This results in a short, 20 s 300 A pulse creating a 100 kAm field in the AlNiCo material. This causes the magnetic domains in the AlNiCo magnets to align in a particular orientation to form a magnetic circuit with a ferrous target. More detailed explanation of the operating principle is available on Wikipedia.

An ON command results in the charging and discharging the capacitors 3 times to achieve full magnetization. An OFF command results in charging and discharging the capacitors several times with changing direction and decreasing amplitude, effectively degaussing the AlNiCo material.

Applications

- Cargo lifting in UAV and robotic applications.
- Robot workholding.
- Education, demonstration of magnetic properties.

Features

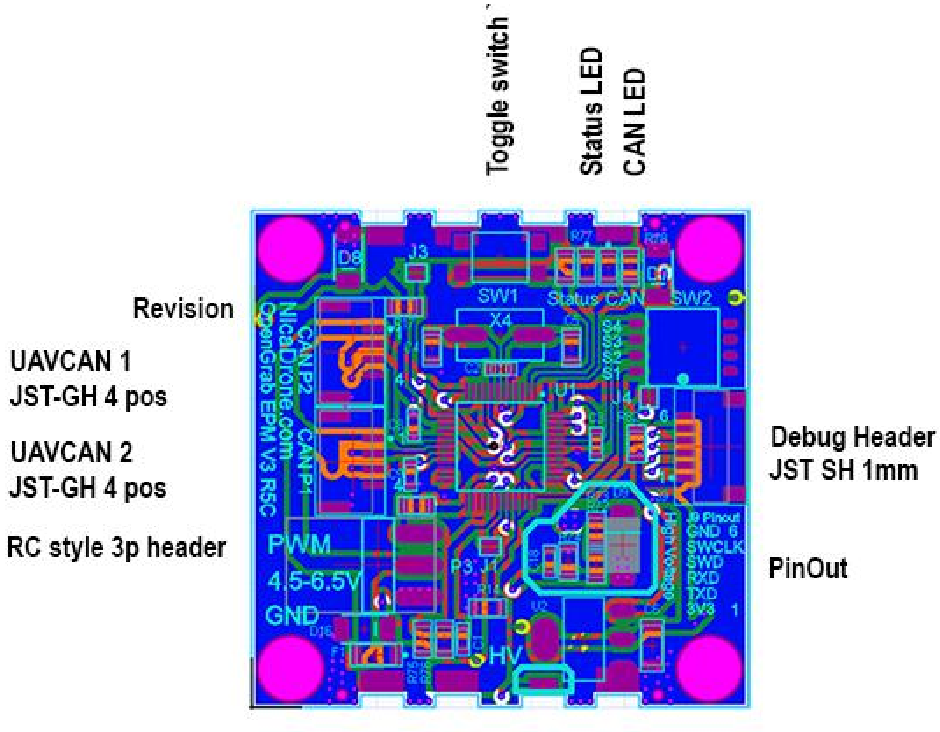
- Steady state power under 50 mW.
- Short cycle time.
- Variety of interfaces:
    - RCPWM
    - UAVCAN
    - Push button
- Open source firmware and hardware.
- 5 V supply voltage, can be powered via the RCPWM connector or via the UAVCAN port.

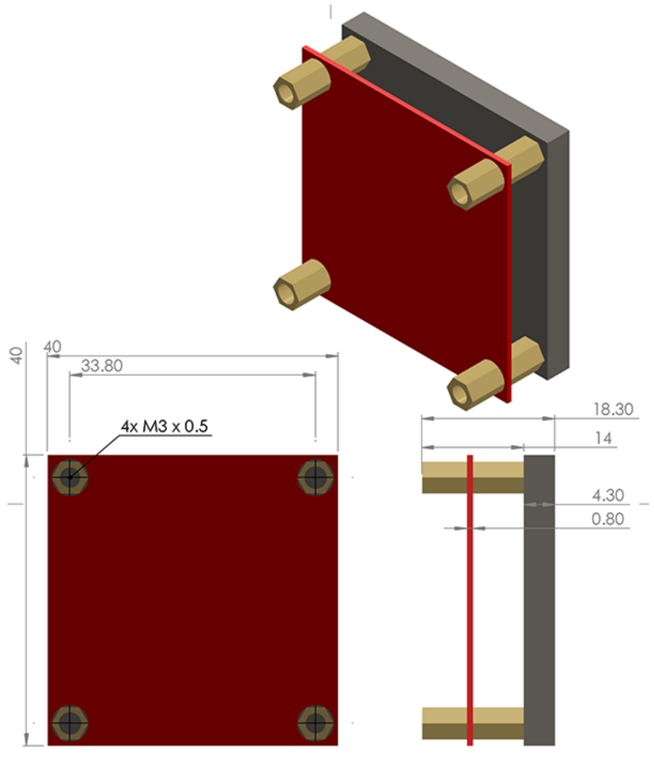
Mechanical properties

> **Safety note**
> The bottom surface of the magnet should be kept clean, because dirt or metal shavings can be crushed into the surface when the magnet is turning on, causing an insulation breakdown.

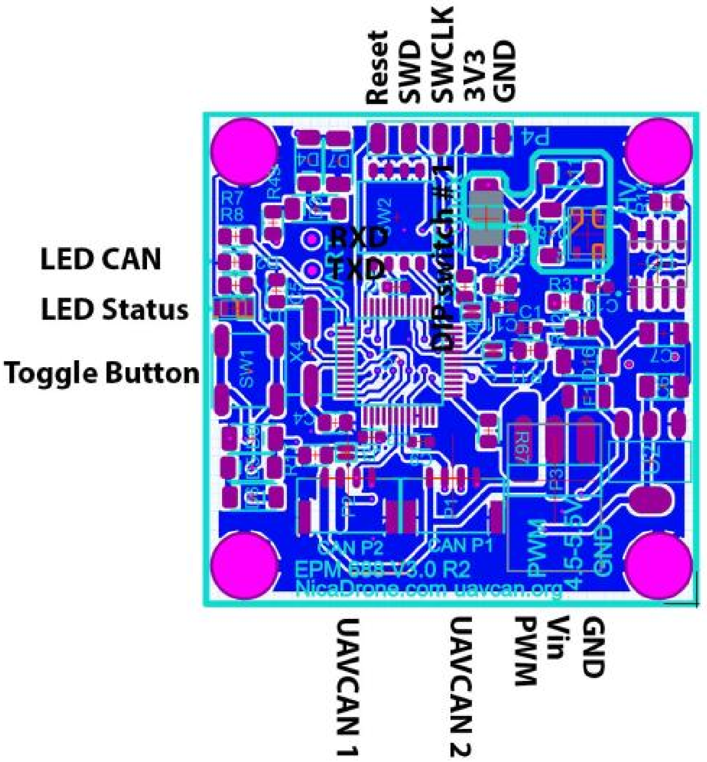
The diagrams below document the mechanical arrangement and dimensions (click to enlarge):

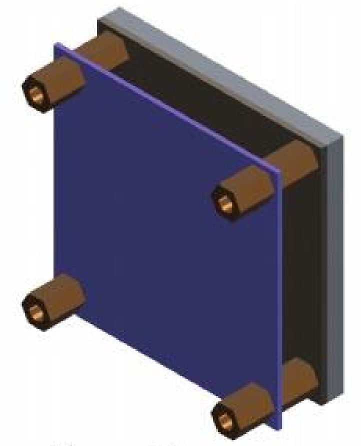
Version 3R5C and newer

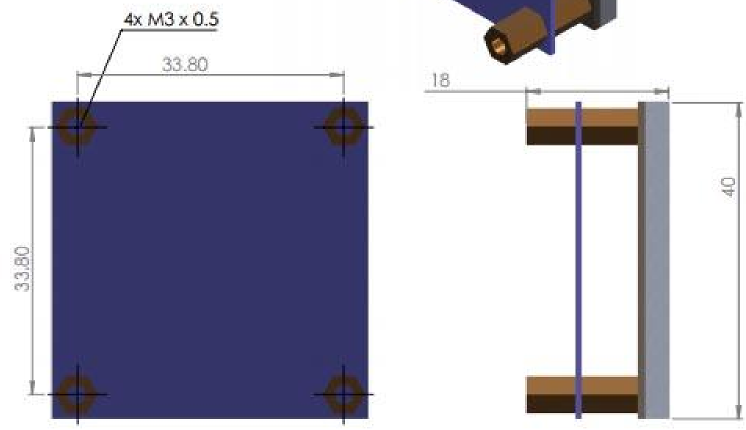




Version 3R4B and older







Characteristics

Symbol	Parameter	Minimum	Typical	Maximum	Unit
$T_{cycle(ON)}$	Time to complete one cycle		0.75		s
$T_{cycle(OFF)}$	Time to complete one cycle		1.2		s
F_{max}	Max holding force	200	300		N
V_{supply}	Operating voltage	4.75	5.0	6.5	V

I_{steady}	Steady state current draw		10		mA
I_{peak}	Peak current draw during cycle execution			1000	mA
m	Mass of the device		65		g
$t_{operating}$	Operating temperature	-40		+70	°C
$RH_{operating}$	Operating humidity (non-condensing)	0		75	%

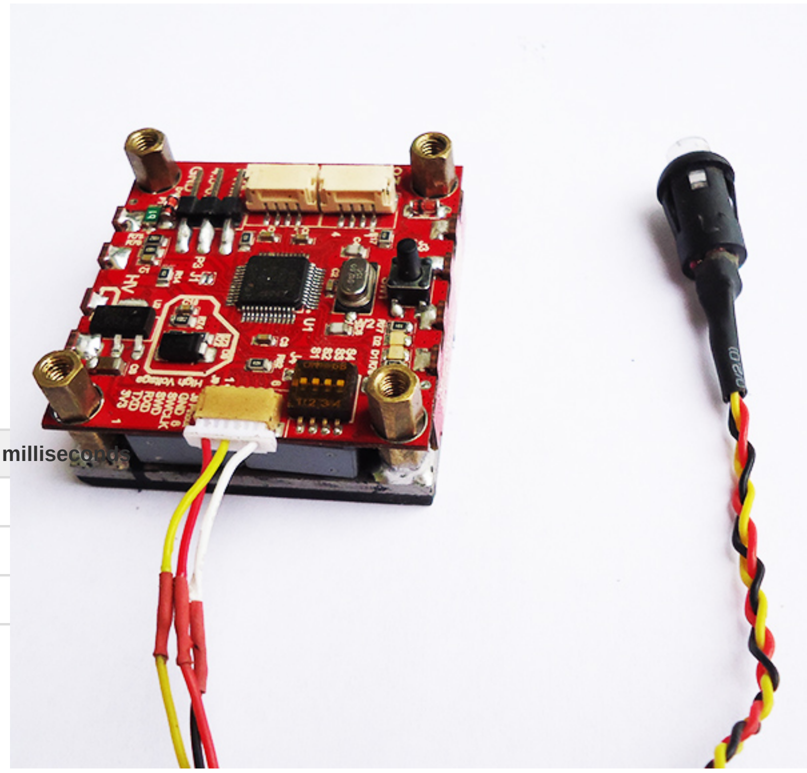
Human-machine interface

Push button

Pressing this button for at least 200 milliseconds will toggle the EPM.

External button

An external button can be connected to the pin 1 (3V3) and 3 (RXD) of the UART header (J9). Pulling pin 3 (RXD) high momentarily toggles the EPM. External button is supported in firmware builds starting from March 2017.



LED indication

Status LED

This LED indicator shows the status of the device derived from the continuous self-diagnostics, according to the UAVCAN node status code:

Health	Blinking ON/OFF duration, milliseconds
OK	50/950
WARNING	50/500
ERROR or CRITICAL	50/100

CAN LED

This LED indicates the activity on the CAN bus. Each blink indicates that there was a CAN frame that was *successfully* transmitted or *successfully* received during the last few milliseconds. Under a high bus load, this LED indicator is expected to glow constantly.

Note that CAN frames filtered out by the hardware acceptance filters will not cause the LED indicator to blink.

RCPWM interface

Connect an RC receiver or some other hardware capable of producing RCPWM signal (e.g. Pixhawk) to the RCPWM connector.

The device divides the PWM pulse duration into three ranges:

- Neutral – while the signal is in this range, the device ignores it.
- OFF – while the signal is in this range, the device will be continuously performing the turn-off sequence.
- ON – while the signal is in this range, the device will be continuously performing the turn-on sequence.

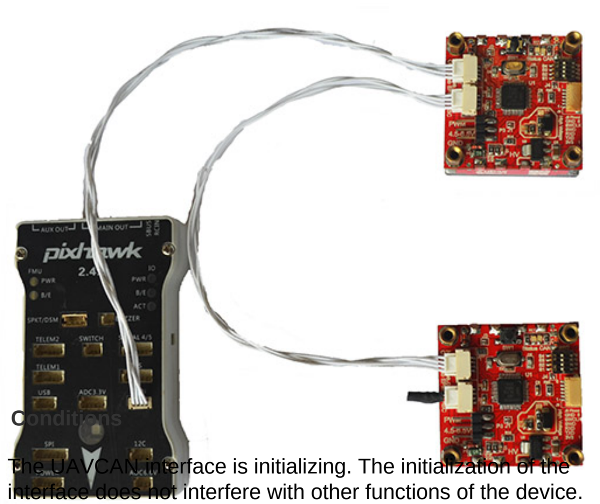
Symbol	Parameter	Minimum	Typical	Maximum	Unit
$T_{RCPWM(ON)}$	RCPWM pulse duration to turn ON	1.75		2.5	ms
$T_{RCPWM(OFF)}$	RCPWM pulse duration to turn OFF	0.5		1.25	ms
f_{RCPWM}	RCPWM input frequency	1	50	55	Hz

$V_{RCPWM(low)}$	Low-level RCPWM input voltage			$0.3 \times V_{supply}$	V
$V_{RCPWM(high)}$	High-level RCPWM input voltage	$0.7 \times V_{supply}$			V

UAVCAN interface

This section describes the properties specific for this product only. For general info about the UAVCAN interface, please refer to the UAVCAN interface documentation page.

Use the UAVCAN GUI Tool to interact with the magnet from a computer (Windows/Linux/Mac).



Mode and status codes

OpenGrab EPM v3 employs the following UAVCAN-defined operating modes:

UAVCAN operating mode	Conditions
INITIALIZING	The UAVCAN interface is initializing. The initialization of the interface does not interfere with other functions of the device.
OPERATIONAL	The UAVCAN interface and the device itself are fully operational.

The following table describes the meaning of the standard UAVCAN health codes.

UAVCAN health code	Conditions
OK	Everything is OK; the device is functioning properly.
WARNING	Not used.
ERROR	See below.
CRITICAL	Not used.

Possible reasons for the health code being `ERROR`:

- Invalid input voltage.
- The high-voltage flyback charger circuit is damaged.

Also, the device reports extended status information via the field `uavcan.protocol.NodeStatus.vendor_specific_status_code`. The higher byte is used to store the current voltage on the buffer capacitor, the units are 2 V per LSB. The lower byte is used to store implementation-specific status flags.

Services

This device does not invoke any services.

The following service servers are implemented:

Data type	Note
`uavcan.protocol.GetNodeInfo`	Name: `com.zubax.opengrab_epm_v3`

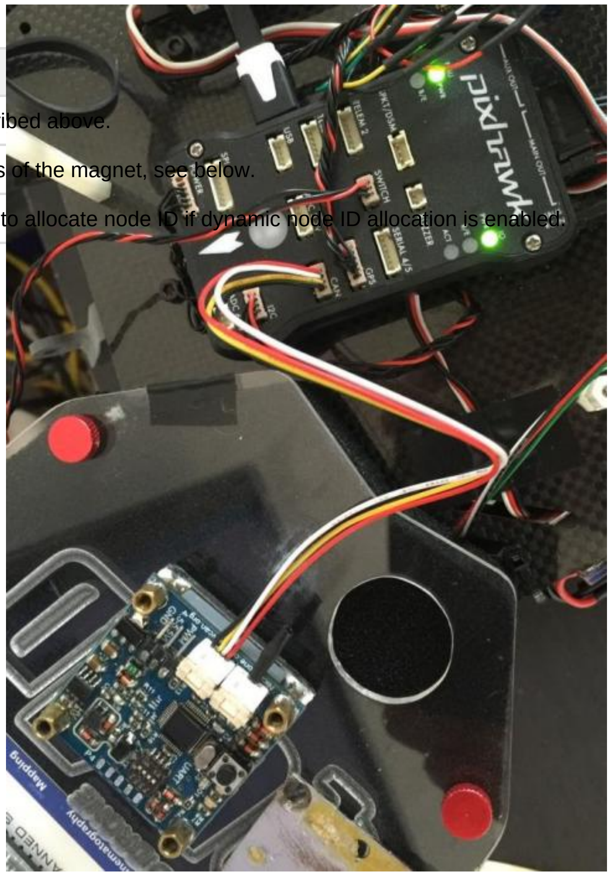
Messages

Input:

Data type	Note
`uavcan.equipment.hardpoint.Command`	Controls the magnet, see below.
`uavcan.protocol.dynamic_node_id.Allocation`	Used to allocate node ID if dynamic node ID allocation is enabled.

Output:

Data type	Note
`uavcan.protocol.NodeStatus`	Described above.
`uavcan.equipment.hardpoint.Status`	Status of the magnet, see below.
`uavcan.protocol.dynamic_node_id.Allocation`	Used to allocate node ID if dynamic node ID allocation is enabled.



`uavcan.equipment.hardpoint.Command`

This message allows to control the magnet via UAVCAN. The fields are interpreted as follows:

`hardpoint_id`

If the field does not equal the hardpoint ID of the current device, the message will be ignored.

`command`

1. If this field is **zero** and the magnet is turned **on** the magnet will turn **off**.
2. If this field is **non-zero** and the magnet is turned **off** the magnet will execute the number of turn **on** cycles specified in the field, but not less than 3 and not more than 10.
3. If this field is **non-zero**, the magnet is turned **on**, and the field has **changed** its value see #2.
4. In all other cases the command will be ignored.

`uavcan.equipment.hardpoint.Status`

This message carries the status of the magnet.

`hardpoint_id`

Hardpoint ID of the current magnet.

`payload_weight`

Always set to NaN.

`payload_weight_variance`

Always set to positive infinity.

`status`

Indicates whether the magnet is turned on or off:

- 1 - the magnet is turned on.
- 0 - the magnet is turned off.

CAN bus characteristics

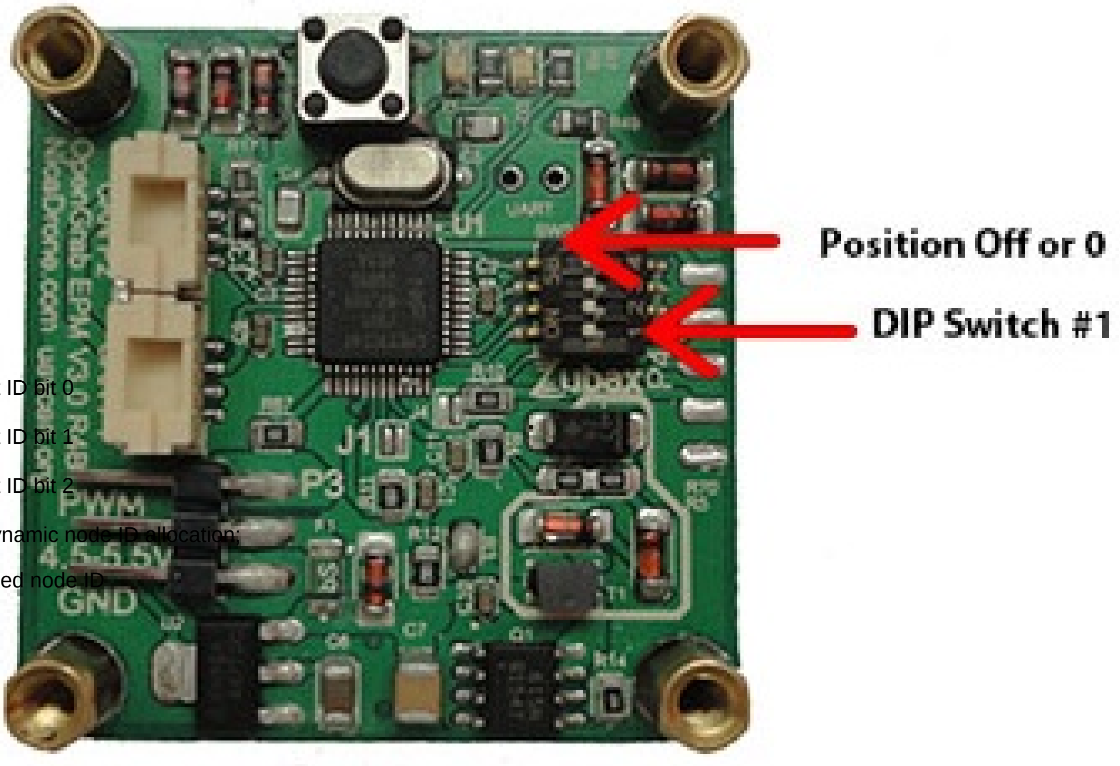
The device will detect the CAN bus bit rate automatically after powering on. The automatic detection is done by means of listening to the bus in the silent mode, alternating between the pre-defined values of supported CAN bit rates (listed in the table) until the first valid CAN frame is received. Unconfigured CAN bus does not interfere with other functions of the device.

Symbol	Parameter	Minimum	Typical	Maximum	Unit
f_{CAN}	CAN bit rate (autodetect)		100 125 250 500 1000		Kbps
$V_{CAN(out)dif-dom}$	CAN dominant differential output voltage	1.5	0	3	V
$V_{CAN(out)dif-rec}$	CAN recessive differential output voltage	-50	0	50	mV
$I_{CAN(out)dom}$	CAN dominant output current	40	70	120	mA
$I_{CAN(out)dom}$	CAN recessive output current	-5		5	mA

$R_{CAN(in)diff}$	CAN differential input resistance	19	30	52	
$t_{CAN(out)to\text{-}dom}$	CAN dominant time-out time	0.3	1	12	ms

DIP switch

The device is equipped with a 4-position DIP switch that allows the user to configure the Hardpoint ID and enable or disable UAVCAN dynamic node ID allocation:



#	Purpose
1	Hardpoint ID bit 0
2	Hardpoint ID bit 1
3	Hardpoint ID bit 2
4	0 - use dynamic node ID allocation; 1 - use fixed node ID

Hardpoint ID

Hardpoint ID is defined in binary by the lowest 3 switches. The table below clarifies the binary encoding:

Hardpoint ID	DIP #3	DIP #2	DIP #1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Node ID selection

If the DIP switch #4 is set to OFF, the device will perform dynamic node ID allocation once the CAN bus bit rate detection is done. This implies that the device will not be available via the UAVCAN interface unless the UAVCAN network contains a functioning dynamic node ID allocation server. Please refer to the UAVCAN specification for more info.

If the DIP switch #4 is set to ON, the device's node ID will be fixed at (Hardpoint ID + 100). For example, if the Hardpoint ID is set to 5, the fixed node ID will be configured as 105. In this case the device does not require an external dynamic node ID allocation server, and therefore it will be accessible via UAVCAN immediately once the CAN bus bit rate detection is done.

UART interface

The EPM reports error and status messages over this interface. This interface can also be used to update the firmware – please refer to the source repository for instructions (link below).

Parameters of the serial interface are as follows:

Parameter	Value
Baud rate	115200
Word size	8
Parity	None
Stop bits	1
New line sequence	`\r\n` (CR-LF)

Symbol	Parameter	Minimum	Typical	Maximum	Unit
$V_{UART(in-low)}$	Low-level UART input voltage			0.3 V_{supply}	V
$V_{UART(in-high)}$	High-level UART input voltage	0.7 V_{supply}			V
$V_{UART(out-low)}$	Low-level UART output voltage			0.4 V_{supply}	V
$V_{UART(out-high)}$	High-level UART output voltage	V_{supply} - 0.4			V

Links

- Source repository (firmware sources, drawings, etc)
- Purchase from NicaDrone