BuWizz3 API

Version:3.22

Introduction

BuWizz3 is a smart connected power brick device compatible with Lego elements. It features a built-in rechargeable battery, 4+2 power outputs (e.g. for motors) and a wireless connection using Bluetooth Low Energy (BLE, Bluetooth Smart). Long-range mode, introduced in BLE 5, is supported to achieve even longer operational range than before.

This document describes the communication protocol with the BuWizz3 device in bootloader and application mode. The document provides information on Bluetooth BLE specific device structure, interfaces and packet exchange protocol.

BLE is a wireless communication standard, introduced in Bluetooth 4. It is intended for low-power accessory devices, has lower transfer speeds and lower overhead than classic Bluetooth technology. It features a unique software model, introducing GATT (Generic Attribute Profile) as the primary communication mean.

BLE devices operate either as central (running GATT client) or device (running GATT server) and exchange data via reading and writing to characteristics values, which are exposed via service and characteristic descriptors.

Notable operations are (supported operations are shown in **bold**):

- **advertisement**: the device advertises special message so that the central can scan and find active devices

- **services list discovery** (central can discover the list of services, supported by the device)
- **write data** to the device with or without device acknowledge
- read (poll) the data from the device
- **notifications**: device notifies the central with the new data (without acknowledging)
- indications: device indicates to the central with the new data (same as notify, but uses acknowledgement)

Note

BLE supports pairing, but is not required. This allows the streamlined user experience, where the application (iOS or Android) can scan for devices and connect to one it finds (without requesting the user to pair the device in system settings first).

Operation modes

The device has two operation modes:

- Bootloader mode: enables transferring of the new firmware
- Application mode: user application running to support BuWizz3 device functions
- Application sleep mode: the device is in low-power mode, no Bluetooth function is active, motor power is disabled, device wakes up on button press or motion,
- Application off mode: the device is in low-power mode and can only be powered on by pressing and holding button for 2 seconds.

By default, the device starts in application mode if memory contents verification succeeds. Otherwise or if application requests it, the device starts in bootloader mode (bootloader has a timeout of 2 minutes - if no data is received for this amount of time, the device is automatically restarted).

Co-processor 'Tajnik'

The BuWizz3 device uses two microcontrollers – the main '*Nordic*' based on Nordic nRF52833 device and the slave '*Tajnik*'. *Nordic* is powered by the always-on power supply (unless the battery is discharged below the threshold and the device is disconnected from the battery), while *Tajnik* is powered by the motor/LED power domain. High-speed communication between *Nordic* and *Tajnik* allows real-time data transfer between devices.

Tajnik generates PWM signals for all 6 motor outputs, measures temperature and motor currents, controls the 4 RGB LEDs and UART communication on the main 4 motor connectors. Since *Tajnik* runs its own firmware, it also features a bootloader for upgrading if deemed necessary. The bootloader mode is accessed via *Nordic* in application mode via BLE connection and uses dedicated commands for firmware updating.

LED indications

There are four RGB LEDs positioned next to main four motor ports on the device. These LEDs are controlled by Tajnik and depend on the operational mode of the device.

Device mode	LED indications
Nordic in sleep/power off	All LEDs off
Nordic waking-up from power off	All green fade in
Nordic going into power off	All green, then red fade out

Tajnik in bootloader (normal)	Chasing red LED counterclockwise
Tajnik in bootloader (boot fail)	Chasing red/green LED counterclockwise
Nordic in bootloader	Red/green alternating slow
Nordic in bootloader (connected)	Red/green alternating fast
Nordic in normal mode	All blue short blinks slow
Nordic in normal mode (low battery)	All red short blinks slow
BLE connected	All LED indicating battery voltage (green -> yellow -> orange -> red)
~~BLE connected (low battery)~~	~~All red short blinks fast~~
BLE connected + LED set via API	Defined by API command
~~Motor running forward~~	~~Corresponding LED blinking green~~
~~Motor running backwards~~	~~Corresponding LED blinking red~~

By default, device will indicate BLE connected mode with notifications enabled in white and blink green/red on motor activation. However, once the command 0x36 (Set LED status) is received, the commanded values will override the default behavior until the notifications are disabled.

BuWizz3 Bluetooth BLE interface

The BuWizz3 device is a standard Bluetooth BLE device and is configured to not use the pairing with the central in order to speed up the connection process and allow multiple users to use the device (one at a time).

The BuWizz3 supports BLE 5 feature called long range / coded mode (also noted as PHY_CODED), where the range is improved on the account of lower data bandwidth and forward error correction. Once the connection with the master device has been established, the BuWizz3 device will automatically request the master to switch to PHY_CODED mode. If the master device supports this mode, both devices will switch into that mode. If the master does not support this or does not agree with the change, the communication protocol will stay in legacy 1 Mbps mode.

Discovering BuWizz3 device

When active, BuWizz3 device advertises main advertisement data and optional scan response data (shown in Figure 1). Main advertisement data contains device name ('BuWizz3' by default, but can be customized) and short manufacturer information (sequence of 8 bytes - 05:4E:'B':'W':'B':'L':00:00 in bootloader and 05:4E:'B':'W':'x':'y':<serialLSB>:<serialMSB> in the application, where x and y are replaced with firmware version and serial number contains lower 16 bits of device's serial number). Scan response data contains the 128-bit UUID of the device's main (BuWizz application) service.

The data in main advertisement packet and scan response packet can be used to identify the device. The advertising uses BLE 4.0 compatible mode (with extended mode disabled) that does not employ long range (coded mode) – the range of the device in the advertising mode is thus lower than once the devices have successfully connected and switched to coded communication mode.

The advertisement interval is 100 ms to facilitate fast device discovery and connection.

Main advertisement data (left):

Description	Value	Index
AD Data 0: <<Flags>>		
AD Data 1: <<Complete Local Name>>		
Length of this data	0x07	[3]
<<Complete Local Name>>	0x09	[4]
B	0x42	[5]
u	0x75	[6]
W	0x57	[7]
i	0x69	[8]
z	0x7A	[9]
z	0x7A	[10]
AD Data 2: <<Manufacturer Specific Data>>		
Length of this data	0x07	[11]
<<Manufacturer Specific Data>>	0xFF	[12]
0x4E	0x4E	[13]
0x05	0x05	[14]
0x42	0x42	[15]
0x6F	0x6F	[16]
0x6F	0x6F	[17]
0x74	0x74	[18]

Scan response data (right):

Description	Value	Index
AD Data 0: <<Incomplete List of 128-bit Service Class UUIDs>>		
Length of this data	0x11	[0]
<<Incomplete List of 128-bit Service Class UUIDs>>	0x06	[1]
Service:		
[0]	0x93	[2]
[1]	0x6E	[3]
[2]	0x67	[4]
[3]	0xB1	[5]
[4]	0x19	[6]
[5]	0x99	[7]
[6]	0xB3	[8]
[7]	0x88	[9]
[8]	0x81	[10]
[9]	0x44	[11]
[10]	0xFB	[12]
[11]	0x74	[13]
[12]	0x00	[14]
[13]	0x00	[15]
[14]	0x05	[16]
[15]	0x4E	[17]

Figure 1: Main advertisement data (left) and scan response data (right)

Connecting to BuWizz3 device

The device uses standard Bluetooth BLE mechanisms for establishing a connection.

Data exchange with BuWizz3 device

Commands and data are exchanged with the BuWizz3 device with the use of BuWizz communication service. The service has the following 128-bit UUID:

	Service UUID
BuWizz	93:6E:67:B1:19:99:B3:88:81:44:FB:74:D1:92:05:50

There are 6 characteristics in this service, shown in the table below. These have a data exchange characteristic with a data exchange descriptor and a Client Characteristic Configuration Descriptor (CCCD) controlling the characteristic behavior.

	Characteristic UUID	Type
Application	0x2901	Write + Notify
Bootloader	0x8000	Write + Notify
UART ch. 1	0x3901	Write + Notify
UART ch. 2	0x3902	Write + Notify
UART ch. 3	0x3903	Write + Notify
UART ch. 4	0x3904	Write + Notify

Data is sent to BuWizz3 device by writing to the data descriptor of the target characteristic, while notifications are used to receive data from the BuWizz3 device. A value of 1 must be written to CCCD descriptor to enable the notifications. Packet size that is sent to device or received from the device can be between 1 and 129 bytes long (up to 244 in bootloader mode). No error checking is included at the packet level (data integrity is guaranteed by the lower levels of the BLE protocol).

The device supports MTU size exchange to increase the MTU size in bootloader mode for improved transfer speed.

BuWizz API

Control data for the motors, device's status data and OTA (Over-The-Air) data need to be transferred via the BLE wireless link. This chapter defines the protocol that is used to construct the packets.

Application commands

The following list of commands is supported by the BuWizz3 device in the application mode with an active BLE connection.

Device status report

Once enabled by writing 1 to data CCCD, the BuWizz3 device will periodically send device status report with the approximate frequency of 20 Hz.

Byte	Function / value
0 (command)	0x01
1	Status flags - bit mapped to the following functions: <table><tr><td>**Bit**</td><td>**Function**</td></tr><tr><td>7</td><td>unused</td></tr><tr><td>6</td><td>USB connection status (1 - cable connected)</td></tr><tr><td>5</td><td>Battery charging status (1 - battery is charging, 0 - battery is full or not charging)</td></tr><tr><td>3-4</td><td>Battery level status (0 - empty, motors disabled; 1 - low; 2 - medium; 3 - full)</td></tr><tr><td>2</td><td>BLE long range PHY enabled</td></tr><tr><td>1</td><td>Motion wake-up enabled flag (1 if enabled)</td></tr><tr><td>0</td><td>error (overcurrent, overtemperature…)</td></tr></table>
2	Battery voltage (9 V + value * 0,05 V) - range 9,00 V – 15,35 V
3-8	Motor currents, 8-bit value for each motor output (value * 0,015 A) - range 0 - 3,8 A
9	Microcontroller temperature (value in °C)
10-11	Accelerometer x-axis value (left-aligned 12-bit signed value, 0.488 mg/digit)
12-13	Accelerometer y-axis value (left-aligned 12-bit signed value, 0.488 mg/digit)
14-15	Accelerometer z-axis value (left-aligned 12-bit signed value, 0.488 mg/digit)
16	Bootloader response command
17	Bootloader response code
18-20	Bootloader response data (3 bytes)
21	Battery-charge current (mA)
22-53	PoweredUp motor data structure (4x) - Motor type (unsigned 8-bit) - Velocity (signed 8-bit) - Absolute position (unsigned 16-bit) - Position (unsigned 32-bit)
54-67	Optional: PID controller state - Process value (single precision float) - Error (float) - PID output (float) - Integrator state (signed 8-bit int) - Motor output (signed 8-bit int)

0x20 Set device name

Set up a up to 12 characters long custom name for the BuWizz device.

Byte	Function / value
0 (command)	0x20 - Set device name
1-12	Device name (if less than 12 characters are used, \0 termination should be used)

0x21 Enable/disable motion-wakeup function

If motion wake-up is enabled, the device will wake-up from the shallow sleep if motion is detected by the accelerometer.

There are two timeout interval settings:

- Idle timeout: time of idle (awaken) mode, when the device can accept the BLE connection. After timeout, device transitions into shallow sleep if motion wake-up is enabled or in power-down state otherwise.
- Shallow-sleep timeout: time of device staying in shallow sleep and sensitive to motion wake-up. After timeout, device transitions into power-down state.

Byte	Function / value
0 (command)	0x21 – Enable/disable motion wake-up
1	Enable motion wake-up (1) or disable it (0)
2-5	Idle timeout in seconds (values lower than 10 are invalid), default 300 seconds
6-9	Shallow-sleep timeout value in s (0 to disable)

0x22 Set acceleration sensor calibration data

Device expects 6 32-bit floating point numbers (single precision) that are stored in the device. Avoid changing the parameters very often (e.g. on every application start), but only when needed.

Byte	Function / value
0 (command)	0x22 – Set calibration factors
1-24	Calibration factors

0x23 Read acceleration sensor calibration data

Device recalls 6 32-bit floating point numbers (single precision) that are stored in the device.

Byte	Function / value
0 (command)	0x23 – Read calibration factors

Response

Byte	Function / value
0 (command)	0x23
1-24	Calibration factors

0x30 Set motor data

Transfers motor data to the device.

Byte	Function / value
0 (command)	0x30 - Set motor data
1-6	Motor data (signed 8-bit value for each motor output) 0x81 (-127): Full backwards 0x00 (0): Stop 0x7F (127): Full forwards
7	Brake flags - bit mapped to bits 5-0 (1 bit per each motor, bit 0 for first motor, bit 5 for the last) If brake flag is set for a target motor, slow-decay control mode will be used (short circuiting the motor armature over the inactive phase). If flag is not set, the corresponding motor will be controlled in fast-decay control method (coasting the motor during the inactive phase).
8	LUT options - bit mapped to bits 5-0 (1 bit per each motor) If LUT bit is set for the target motor, look-up table (LUT) is disabled on that output port.

No response is generated

0x31 Set motor data (extended to support PU functions)

Transfers motor data to the device.

Byte	Function / value
0 (command)	0x31 - Set motor data
1-16	4x motor reference for ports 1-4 (signed 32-bit value for each motor output), function depends on the PU port state (simple PWM, speed or position servo)
17-18	2x motor reference for ports 5-6 (same as bytes 1-6 of command 0x30)
19	Brake flags - bit mapped to bits 5-0 (1 bit per each motor, bit 0 for first motor, bit 5 for the last) If brake flag is set for a target motor, slow-decay control mode will be used (short circuiting the motor armature over the inactive phase). If flag is not set, the corresponding motor will be controlled in fast-decay control method (coasting the motor during the inactive phase).
20	LUT options - bit mapped to bits 5-0 (1 bit per each motor) If LUT bit is set for the target motor, look-up table (LUT) is disabled on that output port.

No response is generated

0x32 Set data transfer period

Set the period that defines rate at which the status data is reported to the host.

Byte	Function / value
0 (command)	0x32 - Set data transfer period
1	Data transfer period in ms (valid range: 20 - 255) in steps of 5 ms

No response is sent.

0x33 Set motor ramp-up and ramp-down rates

Set the motor ramp-up and ramp-down rates in % per 128 ms (e.g. value 100 results in change from 0 to 100% in 0,13 second). Use value of 0 to disable ramp-up and down.

Use the following formula to determine the rate r from the desired time interval T (in seconds) in which the output ramps up or down for 100%:

$$r = \frac{100}{T * 7.8125}$$

Byte	Function / value
0 (command)	0x33 - Set ramp-up and ramp-down rates
1-6	Ramp-up rates
7-12	Ramp-down rates

No response is generated

0x34 Set motor timeout configuration

Set the motor timeout configuration.

Byte	Function / value
0 (command)	0x34 - Set motor timeout configuration
1	Configuration 0: stop immediately, brakes active 1: stop immediately, coast to stop 2 - 254: coast to stop in N-1 seconds

No response is generated

0x35 Activate connection watchdog

Activate the connection watchdog for the current connection. Each call to this command resets the watchdog timer. If timer expires, the device drops the connection and uses the parameters of the 0x34 command for the stop.

Byte	Function / value
0 (command)	0x35 – Activate watchdog
1	Set timeout time in seconds (0 disables the watchdog)

No response is generated

0x36 Set LED status

Set the state of the LED indicators. Send an empty request (only command byte 0x36) to switch LEDs back to default state.

Byte	Function / value
0 (command)	0x36 – Set LED status
1-3	Motor 1 LED (R, G, B components)
4-6	Motor 2 LED (R, G, B components)
7-9	Motor 3 LED (R, G, B components)
10-12	Motor 4 LED (R, G, B components)
13-16 (optional)	LED blinking configuration (1 byte per LED) bits 7…4: blinking frequency (0-15 Hz), LED is off if value is 0 bits 3...0: duty cycle (0 = 1/16, 15 = 100%)

	Default on connection established: 0xFF (solid ON in white)

No response is generated

0x37 Set acceleration data low pass filter constant
Set the period that defines rate at which the status data is reported to the host.

Byte	Function / value
0 (command)	0x37 - Set sensor LPF setting
1	Sensor sampling setting (default startup value is 2 – shown in green)

1	0	0	0	0.781Hz
1	0	0	1	1.563Hz
1	0	1	0	3.125Hz
1	0	1	1	6.25Hz
0	0	0	0	12.5Hz
0	0	0	1	25Hz
0	0	1	0	50Hz
0	0	1	1	100Hz

No response is sent.

0x38 Set current limits
Setups current limits for the motors. By default, the current limits are configured to default values of 1.5 A (for PU ports 1-4) and 3.0 A (for PF ports 5-6) on every BLE connection start.

If the motor current is above the current limit, the PWM duty cycle of the affected motor is reduced until the current falls below the limit. PWM duty cycle will increase if the motor load is reduced.

Byte	Function / value
0 (command)	0x38 - Set current limit data
1-6	Current limit in steps of 30 mA

No response is generated

0x40 UART baudrate setup
Set the baudrate on the selected channel.

Byte	Function / value
0 (command)	0x40 – Send command
1	0x10 – Set baudrate
2	Channel ID
3-6	Baudrate (unsigned 32-bit number)

No response is generated

0x50 Set PU port function
Configure the function on the target PU port.

Byte	Function / value
0 (command)	0x50 - Set PU function
1-4	PU port functions - 0x00 Generic PWM output - 0x10 PU simple PWM (default) - 0x14 PU speed servo - 0x15 PU position servo - 0x16 PU absolute position servo

Default parameters are selected for known PU motors when switching to or between PU servo modes.

0x51 Enable PID controller state reporting
This enables the PID controller state to be reported in the status message.

Byte	Function / value
0 (command)	0x51 – Enable PID state reporting
1	Set to port index (1..4) to enable, set to 0 to disable

0x52 Set servo reference input
Sets the reference input for the servo controller (speed or position, depending on the port mode)

Byte	Function / value
0 (command)	0x52 – Set PID reference
1-16	Reference values (32-bit signed int) for each channel

0x53 Set servo PID parameters

Sets the parameters for the servo on the single PU port

Byte	Function / value		
0 (command)	0x53 – Set PID parameters		
1	Port index		
2-37	- outLP (float) - D_LP (float) - speed_LP (float) - Kp, Ki, Kd (3x float) - limI (float) - reference rate limit (float) (step of reference change at 10 ms intervals) - limOut (int8) - DeadbandOut (int8) - DeadbandOutBoost (int8) - valid mode (equal to port mode selected)		

Controller structure (PI-D)

Output of the PID controller can be additionally filtered, as well as differentiated input speed and D-part of the controller. Output is conditioned by applying the deadband, anti-friction boost and saturation. All parameters are configurable. First order IIR low-pass filters are used for filtering.
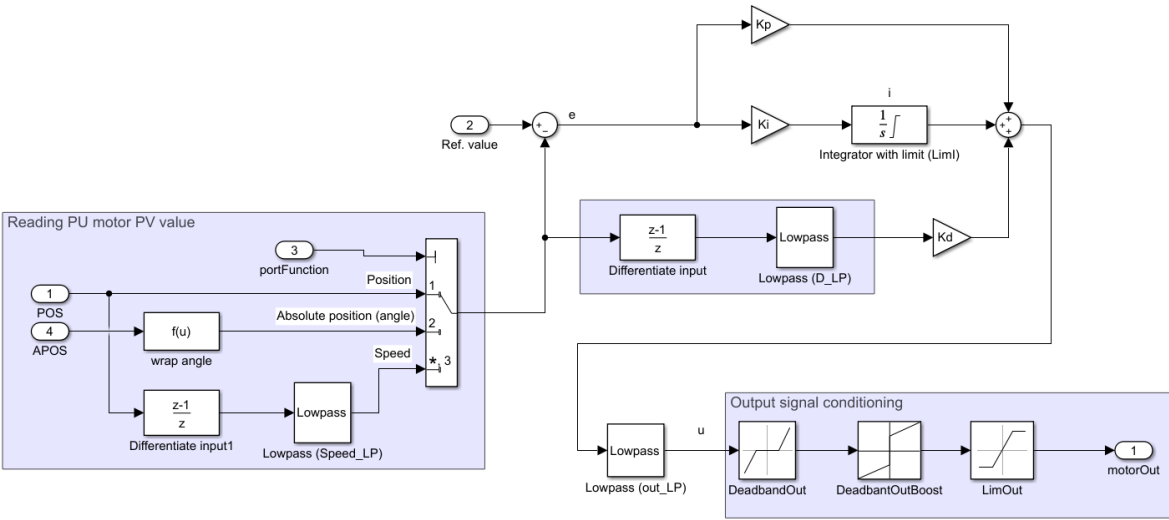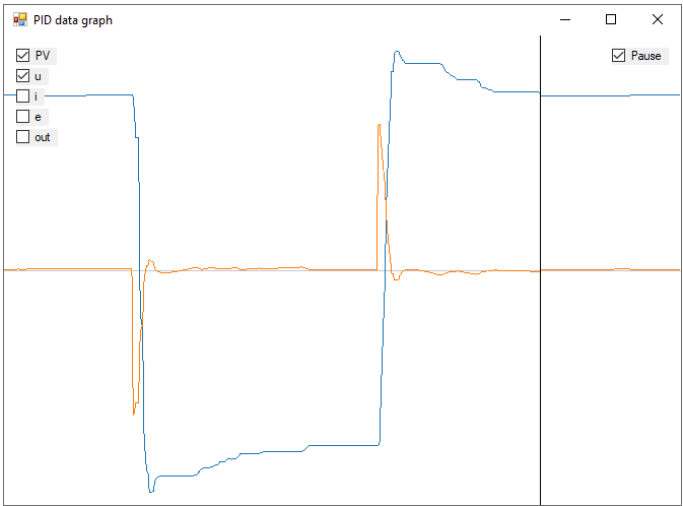
PID controller is running at 100 Hz.

Figure 2: Controller structure

<u>Default servo parameters</u>

Parameter	Position servo	Speed servo	Generic PID	Parameter range
Kp	0.4	0.8	Kp	$[0 - N/A)$
Ki	0.01	0.06	Ki	$[0 - N/A)$
Kd	-0.8	-3	Kd	$(-N/A - 0]$
outLP (K_{LP})	0	0.5	0	$[0 - 1)$
D_LP (K_{LP})	0.9	0.5	0	$[0 - 1)$
speed_LP (K_{LP})	/	0.9	0	$[0 - 1)$
DeadbandOut	2	2	0	$[0 - 127]$
DeadbandOutBoost	2	2	0	$[0 - 127]$
LimI	20	127	127	$[0 - 127]$
LimOut	100	127	127	$(0 - 127]$
Reference rate limit	N/A	N/A	N/A	$[0 - N/A]$

Parameters can be tuned using the test application ('Enable PID state reports' button), which shows the relevant controller parameters in realtime (BLE notification update period should be set to 25 ms or lower).

<u>Parameter description</u>

Kp, Ki and Kd are standard PID controller parameters (see the controller structure in Figure 2). Continuous-style structure is used, implementing discretization via the Euler method in integrator part of the controller. The integrator part of the PID controller is saturated at LimI to address the integrator runaway.

There are three low-pass filters implemented (output filter, speed differentiation, D-part filter in PID), controlled via out_LP, speed_LP and D_LP. Low-pass filter parameter are calculated, as shown below ($Ts = 0.01s$) for desired f_{LP} $[Hz]$ (low-pass filter cut-off frequency).

$$K_{LP} = 1 - \frac{Ts}{Ts + \frac{1}{2\pi f_{LP}}}$$

There is additional rate-limit filter on the reference input (controlled via reference rate limit parameter), which limits the rate of change on the reference input. The values of the parameter are in reference change per one timestep (10 ms). It is beneficial to limit the reference rate in positional servo functionality since it allows using much higher gains in the controller without risking large overshoots of the signal. Higher gains result in better static load handling and improved steady state and input tracking response.

Another block of filters conditions the output signal – output deadband (to prevent the motors to be continually actuated back and forth in small changes due to noise), deadband out boost (to prevent the motors from sticking due to static friction and improving the responsiveness of the system) and output limiter (to limit the signal going to the motor to the valid range).

0xA1 Activate shelf mode

Activate the shelf mode in the device immediately. This will disconnect the battery from the device and will require the user to connect the charger in order to wake-up the device.

Byte	Function / value
0 (command)	0xA1 – Shelf mode

0xAC Check charger settings

Byte	Function / value
0 (command)	0xAC – Check charger settings

Response

Byte	Function / value
0 (command)	0xAC
1	

UART data exchange (motors 1 – 4)

Motor channels 1 to 4 support UART communication. UART communication on the target motor port is enabled by writing 1 to the dedicated characteristic CCCD (enable notifications). Data is sent by writing to the characteristic and received via notifications.

The UART data exchange is only functional when port function 0 is selected. If PU port function is enabled on the target port, UART data is handled by the device itself. UART is disabled if CCCD on the characteristic is disabled.