

Microcontroller Unit (MCU)

Block Data Sheet

MCU 140616

SPECIFICATIONS

- > **Sampling Rate:** 1, 10, 100 or 1000Hz
- > **Analog Ports:** 6 in (A1-A6) + 1 out (PWM)
- > **Digital Ports:** 2 in (I1&I2) + 2 out (O1&O2)
- > **Auxiliary Ports:** 1 in (ABAT)
- > **Resolution:** 10-bit (A1-A4) + 6-bit (A5&A6)
- > **Communication:** UART
- > **Operating Voltage:** 3.3V
- > **Input Voltage Range:** 3.0-5.5V
- > **Consumption:** ~5mA

FEATURES

- > Real-time data sampling and streaming
- > High performance open-source firmware
- > Crystal oscillator for maximum precision
- > Status and low-battery indicators
- > Cross-talk attenuation
- > Raw data sampling and output
- > Plug & play operation
- > Easy-to-use OpenSignals software & APIs

APPLICATIONS

- > Rapid prototyping of custom hardware
- > Biosignal research
- > Real-time data visualization
- > Real-time data recording

GENERAL DESCRIPTION

Our MCU block is designed for accurate and reliable real-time data streaming over any UART-compatible interface (e.g. Bluetooth, FTDI). Its high performance firmware, made available in open source, can acquire and control up to 6 analog inputs, 1 analog output, 2 digital inputs and 2 digital outputs at up to 1kHz. Together with the OpenSignals software and a wide array of programming APIs, our MCU enables hassle-free access to raw data, making it ideal for rapid prototyping of cross-platform real-time applications.

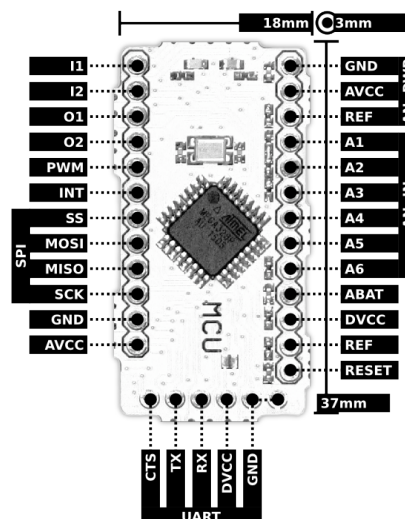


Fig. 1. Pin-out and physical dimensions.

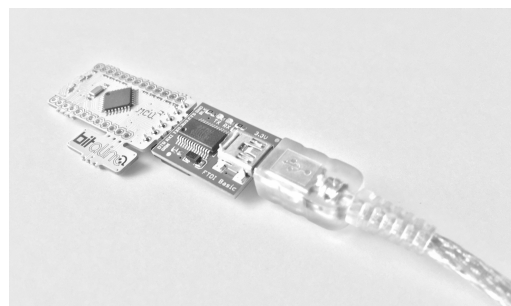


Fig. 2. Our MCU can be used with any UART-compatible accessory (although minor adaptations may be needed).

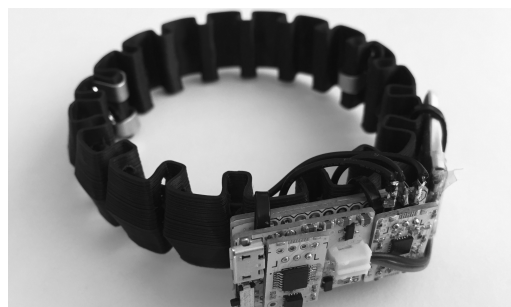


Fig. 3. Prototype wearable device using our MCU and Bluetooth blocks for real-time wireless data streaming.



PLUX – Wireless Biosignals, S.A.
Av. 5 de Outubro, n. 70 – 8.
1050-059 Lisbon, Portugal
bitalino@plux.info
<http://bitalino.com/>

REV A

© 2016 PLUX

This information is provided "as is," and we make no express or implied warranties whatsoever with respect to functionality, operability, use, fitness for a particular purpose, or infringement of rights. We expressly disclaim any liability whatsoever for any direct, indirect, consequential, incidental or special damages, including, without limitation, lost revenues, lost profits, losses resulting from business interruption or loss of data, regardless of the form of action or legal theory under which the liability may be asserted, even if advised of the possibility of such damages.



BEWARE: DIRECT OR INDIRECT COUPLING TO THE MAINS MAY RESULT IN SHOCKING HAZARD



Microcontroller Unit (MCU)

Block Data Sheet

STATE COMMANDS

The BITalino (r)evolution firmware is prepared to receive a set of commands that can be used to control the state of the device (Fig. 4):

- > **Idle:** Corresponds to the state in which the device is when in standby (status LED fades at 0.5Hz); it is the default state when the device is turned on.
- > **Live:** Corresponds to the state in which the device is acquiring and streaming data in real time (LED blinks at 1Hz); setting the channel mask bits (labeled A1-A6) to 0 or 1 deactivates / activates the real-time streaming of the corresponding channel A1-A6 from the device.
- > **Simulated:** This is a facility provided for developers, and in this state the device streams synthetic data generated by the firmware (sine, saw tooth, and square waves, together with a pre-recorded ECG time series); this enables the development and testing of the software layer at the receiver using the actual physical and transport layers, but without requiring sensors to be connected to a user to have values changing on the streamed data.

bits								
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	idle mode (when in live or simulated mode)
A6	A5	A4	A3	A2	A1	0	1	live mode with analog channel selection (when in idle mode)
A6	A5	A4	A3	A2	A1	1	0	simulated mode with analog channel selection (when in idle mode)

Fig. 4. State commands.

DATA PACKETS

While in live or simulated mode, the BITalino (r)evolution firmware streams the acquired data in real time, formatted as a structured sequence of bits corresponding to:

- > **CRC:** 4-bit Cyclic Redundancy Check (CRC) code, useful for the evaluation of the data packet consistency on the receiver.
- > **S:** 4-bit sequential number generated by the firmware to identify the packet, which can be used on the receiver to detect packet loss.
- > **O1 & O2:** State of the digital output ports O1 & O2 on the device.
- > **I1 & I2:** State of the digital input ports I1 & I2 on the device.
- > **A1-A6:** Digital code produced by the ADC for the voltage at the corresponding analog input ports A1-A6; the first four channels arrive with 10-bit resolution (ranging from 0-1023) while the last two arrive with 6-bit (ranging from 0-63)¹.

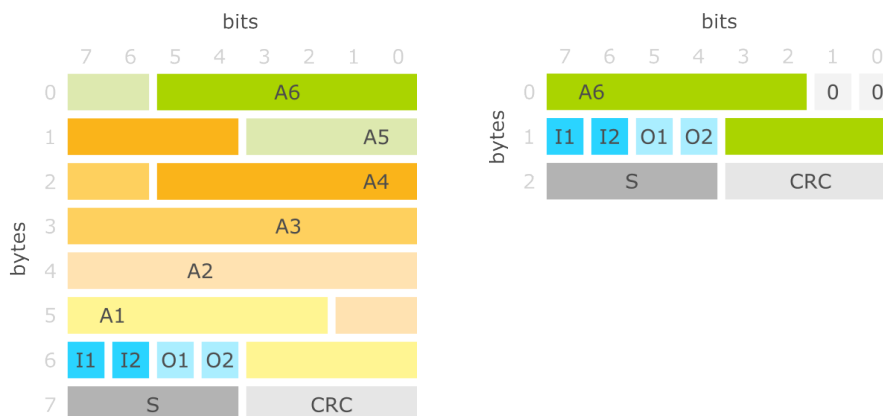


Fig. 5. Data packet structure sent while the device is in live or simulated modes, when all the channels are selected for acquisition (left) and only channel A6 is selected for acquisition (right)¹.

¹ It is important to highlight that A5 and/or A6 may arrive with 10-bit resolution if less than 6 channels are acquired (e.g. when the selected channels are [A1, A2, A3, A5, A6] only A6 has 6-bit, when the selected channels are [A3, A4, A5, A6] both A5 & A6 have 10-bit, etc.).

Microcontroller Unit (MCU)

Block Data Sheet

CONFIGURATION COMMANDS

While in idle mode, the BITalino (r)evolution firmware is also prepared to receive a set of device configuration commands (Fig. 6):

- > **Sampling Rate:** The sampling rate can be selected by setting the bits labeled F to 00 (1Hz), 01 (10Hz), 10 (100Hz), or 11 (1000Hz).
- > **Battery Threshold:** By default, the firmware is prepared to control the low battery LED based on the ABAT analog input readings (i.e. if the ABAT pin from the PWR block is connected to ABAT on the MCU, the firmware monitors the battery level); the battery LED threshold can be set to any 6-bit value (ranging from 0-63), in which values closer to 0 trigger the low battery indication for battery voltages bellow 3.4V (5-10% charge), and values closer to 63 trigger the low battery indication for battery voltages bellow 3.8V (90-95% charge).
- > **Firmware Version:** For compatibility and traceability purposes, it may be useful to know the firmware version running on the device, therefore a command is provided to enable such request, to which the device responds with a string (e.g. "BITalino_v5.0\n").
- > **Device Status:** In some applications, it is useful simply to retrieve the state of the device at a given moment in time, without having it in live or simulated mode (i.e. when real-time data streaming is not a requirement); as such, a command is provided to inquiry the device about its status, to which the device replies with a status packet (as shown in Fig. 7).

bits								
7	6	5	4	3	2	1	0	
F	0	0	0	0	0	1	1	set sampling rate
BATTERY THRESHOLD						0	0	set battery threshold
0	0	0	0	0	1	1	1	send firmware version string
0	0	0	0	1	0	1	1	send device status

Fig. 6. Configuration commands.

STATUS PACKETS

In response to a device status command, the BITalino (r)evolution firmware sends a formatted structured sequence of bits corresponding to (Fig. 7):

- > **O1 & O2:** State of the digital output ports O1 & O2 on the device.
- > **I1 & I2:** State of the digital input ports I1 & I2 on the device.
- > **Battery Threshold:** Low battery LED activation threshold currently defined on the firmware.
- > **A1-A6:** Digital code produced by the ADC for the voltage at the corresponding analog input ports A1-A6; all channels arrive with 10-bit resolution (ranging from 0-1023).
- > **ABAT:** Digital code produced by the ADC for the voltage at the ABAT analog input pin; this channel arrives with 10-bit resolution (ranging from 0-1023).

Microcontroller Unit (MCU)

Block Data Sheet

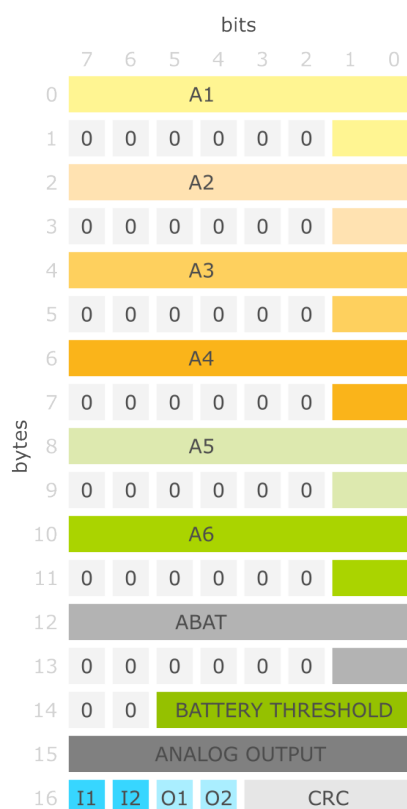


Fig. 7. Status packet structure sent in response to a device status inquiry command².

ACTION COMMANDS

Regardless of the mode in which the device is in, the BITalino (r)evolution firmware provides a set of action commands to control the digital and analog outputs (Fig. 6):

> **Trigger Digital Outputs:** The digital output pins O1 & O2 can be activated / deactivated by setting the pins labeled O1 & O2 to 1 or 0 (the current state of the outputs is sent by the device both in the data packets and state packets).

> **Control Analog Output:** Sending a specific command to the device followed by an 8-bit value sets the output of the PWM, which in turn drives the analog output on the DAC block.

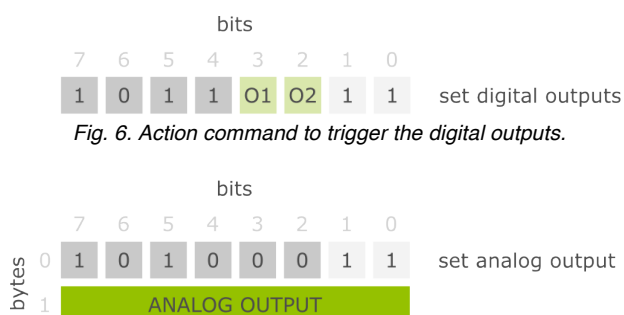


Fig. 7. Action command to set the state of the analog output.

ORDERING GUIDE

Part #	Description
COMP-MCU	Microcontroller (MCU) block pre-programmed with our custom firmware ³ , optimized for real time data acquisition

² Byte 15 is only available in firmware version 5.2 or above

³ <https://github.com/BITalinoWorld/firmware-bitalino-revolution>

Microcontroller Unit (MCU)

Block Data Sheet
