Installation and Programming Manual

BioMax2 / BiomaxPlus

KBio2-Online

v. 2.11

Kimaldi Electronics, S.L.
Ctra. Rubí, 292-B Pol.Ind. Can Guitard
08228 Terrassa (Barcelona) **CIF** B61802302

www.kimaldi.com
Tel: 937 361 510 **Fax**: 937 361 511
E-mail: kimaldi@kimaldi.com

Table of Contents

1. About this Manual

This manual applies to *BioMax2* and *KBio2-Online* access and presence control electronics, firmware versions 0x61.43 and 0x62.43 respectively.

2. Description

BioMax2/KBio2 electronics are designed as Online access and/or presence control terminals..

- *BioMax2* is versatile enough to control up to two Clock&Data readers and two RS-232 devices, whether RFID readers or FIM modules for biometric identification. It is also equipped with a 2x20-character alphanumeric display and a 4x4 numerical keypad.
- *KBio2-Online* is a more compact piece of hardware, only supporting one Clock&Data reader and two RS232 devices (RFID readers or FIM modules). User interface is simpler, solely based on LEDs.
- Default connectivity is based on both TCP/IP and UDP. Alternatively, communication may be carried out via RS232. KiWi2 electronics can also be connected for WiFi communications.
- It is possible to add the *BiomaxPlus-DB* device, intended for Semi-Offline access control. That is, the system switches to Offline mode whenever communication with main Host is lost. That way, it is BioMax2 itself managing and logging access activity, until communication with main Host is recovered.

BioMax2 and *KBio2-Online* electronics are recommended for all access control applications on electric locks, displays, leds, buzzers and ticker printers, etc. requiring online operation. In particular, whenever biometrical identification is required, the search engine may be installed in the main Host[1] or in the FIM module.

BioMax2 and *KBio2-Online* electronics may be controlled with serial commands or with the help of the Active-X, Kimaldi Bio-OCX controller or its equivalent .NET version, the latter method being recommended.

Except for their hardware differences, *BioMax2* and *KBio2-Online* are equivalent in terms of functionality and programming interface. With regards to *BioMaxPlus*, it behaves as BioMax2 in all aspects, although it offers additional features.

1. Nitgen's eNSearch and eNBSP libraries are available for that purpose.

2.1. BioMax2 electronics

The *BioMax2* electronics board has the following hardware resources:

- RJ45 Ethernet connection (10/100-BaseT)
- UART0: First port RS-232 (by default, for Host connection).
- UART1: Second port RS-232 (by default, for biometric sensor connection).
- UART2: Third port, 3V3 TTL levels (reserved as a socket for *BioMaxPlus-DB* or other special cases).
- Two Clock&Data ports
- Four relays
- Two LED outputs (Green and Red)
- One output for an external buzzer.
- One current-limited 5 VDC outlet
- One connector to a 4x4 keypad
- One 2x20-character, alphanumeric, backlit display connector
- One card-mounted buzzer
- Four digital inputs
- Connection to Optical Barrier
- One socket for KiWi2 Converter (UART0) or *BioMaxPlus-DB* (UART2).

2.2. KBio2-Online electronics

The *KBio2-Online* electronics board has the following hardware resources:

- RJ45 Ethernet connection (10/100-BaseT)
- UART0: First port RS-232 (by default, for Host connection).
- UART1: Second port RS-232 (by default, for biometric sensor connection).
- UART2: Third port, 3V3 TTL levels (reserved as a socket for *BioMaxPlus-DB* or other special cases).
- One Clock&Data ports
- One relay
- One current-limited 5 VDC outlet
- One connector to a 3-key pad with 3 LEDs
- One card-mounted buzzer
- One digital input
- Connection to Optical Barrier
- One socket for KiWi2 Converter (UART0) or *BioMaxPlus-DB* (UART2).

3. Technical Specifications

3.1. BioMax2

Size:	OEM board: 134 mm x 106 mm x 25 mm
Supply Voltage:	5 VDC. ± 10%
Maximum consumption [1]:	450 mA
Ethernet Interface:	10/100 Base-T
RS232 Interface:	Baud Rate between 9600 and 57600; n,8,1
5VDC Output:	Limited to 1 Amp by means of a retriggerable fuse. Maximum recommended current to external readers: 500 mA.
User interface:	2x20-character, backlit LCD Display; 4x4 numerical keypad, buzzer on-board; output drivers for 2 LEDs (green and red) and external buzzer.
Optical Barrier:	4-pin connector for LED and phototransistor [2]
Relay contacts:	4 relay contacts, normally open, 24V / 1A
Digital inputs:	4 relay-type digital inputs. In open circuit (contact open) their logic value will be 0. In contact with GND (contact closed), logic value 1.
Clock&Data Inputs:	2 ports for Clock&Data readers, ABA-Track 2 encoding.

3.2. KBio2 Online

Size:	OEM board: 96 mm x 59 mm x 25 mm
Supply Voltage:	5 VDC. ± 10%
Maximum consumption [3]:	400 mA
Ethernet Interface:	10/100 Base-T
RS232 Interface:	Baud Rate between 9600 and 57600; n,8,1
5VDC Output:	Limited to 1 Amp by means of a retriggerable fuse. Maximum recommended current to external readers: 500 mA.
User Interface:	3-key pad with 3 LEDs and buzzer.

1. Consumption of the raw board, without peripheral devices connected to it.
2. The LED optical barrier has been discontinued on devices featuring FIM5360 or equivalent. See section 5.3 for further information
3. Consumption of the raw board, without peripheral devices connected to it.

Optical Barrier:	4-pin connector for LED and phototransistor [1].
Relay contact:	1 relay contact, normally open, 24V / 1A
Digital input:	1 relay-type digital input. In open circuit (contact open) its logic value will be 0. In contact with GND (contact closed), logic value 1.
Clock&Data Input:	1 port for Clock&Data reader, ABA-Track 2 encoding.

3.3. Addressing

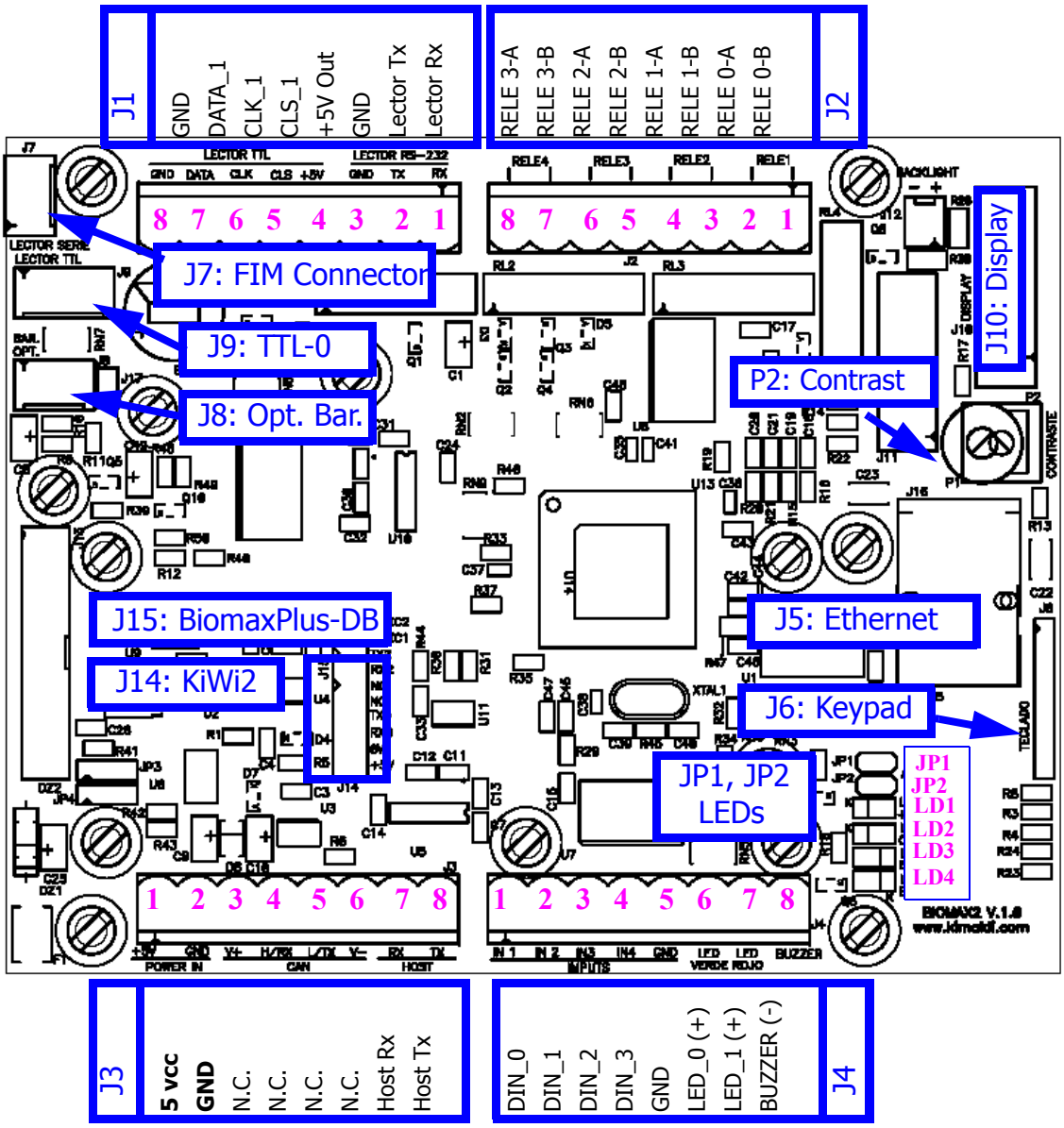
BioMax2 and *KBio2-Online* electronics have two jumpers (JMP1 and JMP2) to encode the ID Application in the KSP protocol.

See chapter 6.3.: "Routing and Localisation" for further details.

1. The LED optical barrier has been discontinued on devices featuring FIM5360 or equivalent. See section 5.3 for further information

4. Installation

4.1. Connector signals, BioMax2



WARNING !!!!

BioMax2 electronics board requires +5 Vdc supply

Take the necessary antistatic precautions when handling this product to avoid damaging the sensitive electronic devices.

4.2. Connection details, BioMax2

J3 - Main Connector

Pin 1 - Positive power supply pole (**5 Vcc**)

Pin 2 - Negative power supply pole (GND) (Pin no. 5, female SubD connector)

Pin 3 - Not connected

Pin 4 - Not connected / UART_2 - Rx, 3V3 TTL, in special cases

Pin 5 - Not connected / UART_2 - Tx, 3V3 TTL, in special cases

Pin 6 - Not connected

Pin 7 - UART_0 - Rx, RS232 to Host (Pin nº 3, female SubD connector)

Pin 8 - UART_0 - Tx, RS232 to Host (Pin nº 2, female SubD connector)

J2 - Relay Connector

Pin 1 - Pole A of Relay 0

Pin 2 - Pole B of Relay 0

Pin 3 - Pole A of Relay 1

Pin 4 - Pole B of Relay 1

Pin 5 - Pole A of Relay 2

Pin 6 - Pole A of Relay 2

Pin 7 - Pole A of Relay 3

Pin 8 - Pole B of Relay 3

J4 - Digital Input/Output connector

Pin 1 - Digital Input 0

Pin 2 - Digital Input 1

Pin 3 - Digital Input 2

Pin 4 - Digital Input 3

Pin 5 - LED common cathode - GND

Pin 6 - Anode (+) for LED 0 (Green)

Pin 7 - Anode (+) for LED 1 (Red)

Pin 8 - Cathode (-) for External Buzzer (5 V)

J1 - Connector to Card Readers

Pin 1 - UART_1 - Rx, RS-232 to reader (Pin nº 2, male SubD connector)

Pin 2 - UART_1 - Tx, RS-232 to reader (Pin nº 3, male SubD connector)

Pin 3 - Negative power supply pole (GND) (Pin no. 5, male SubD connector)

Pin 4 - +5 VDC Output

Pin 5 - CLS Signal (TTL_1)

Pin 6 - Clock Signal (TTL_1)

Pin 7 - Data Signal (TTL_1)

Pin 8 - Negative power supply pole (GND)

J5: Ethernet Connector (RJ45 connection)

J9 - TTL_0 connector to Card Reader: NdCAN-compatible

- Pin 1 - Data Signal (TTL_0)
- Pin 2 - Clock Signal (TTL_0)
- Pin 3 - CLS Signal (TTL_0)
- Pin 4 - +5 VDC Output
- Pin 5 - Negative power supply pole (GND)

J7: Connection to FIM biometric identification module or to generic RS-232 reader

- Pin 1 - +5 VDC Output
- Pin 2 - UART_1 - Tx 232 signal to reader or FIM Module
- Pin 3 - UART_1 - Rx 232 signal to reader or FIM Module
- Pin 4 - Power supply negative pole (GND)

J8: JST Connector to Kimaldi Optical Barrier

- Pin 1 - Anode (+) of the emitting LED
- Pin 2 - Cathode (-) of the emitting LED
- Pin 3 - Collector (+) of the receiving phototransistor
- Pin 4 - Emitter (GND) of the receiving phototransistor

P2: Display Contrast: adjustment potentiometer

JP1, JP2: Jumpers to select KSP Application_ID (see Subsection 6.3.2.)

LEDs showing the electronics status:

- LD1 - 3V3 supply operating
- LD2 - Rx/Tx Instructions or Events between *BioMax2* and the Host computer
- LD3 - Red - Ethernet Link Error
- LD4 - Green - Ethernet Activity (when blinking).

In addition, Keypad and Display connections are available for the OEM version of the equipment:

J6: 4x4 Keypad connector

Pin 1 - Column 3 (OUT)	Pin 5 - Row 0 (IN)
Pin 2 - Column 2 (OUT)	Pin 6 - Row 1 (IN)
Pin 3 - Column 1 (OUT)	Pin 7 - Row 2 (IN)
Pin 4 - Column 0 (OUT)	Pin 8 - Row 3 (IN)

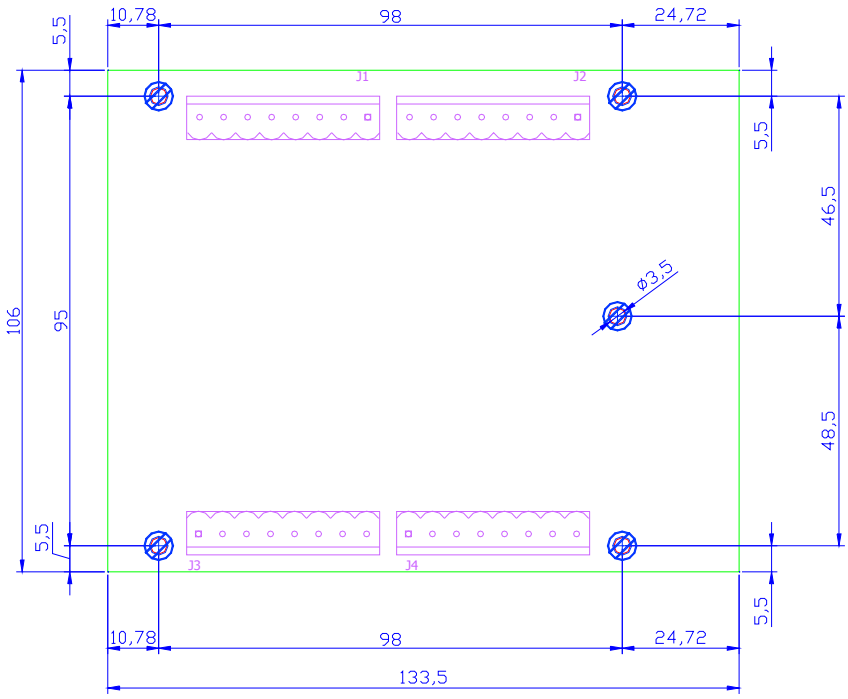
J10: Connector to 20x2 Display, with incorporated Backlit

J14 - Connector/socket for KiWi2 Converter (UART0):

- Pin 1 - No Connect
- Pin 2 - No Connect
- Pin 3 - UART_0 - Tx 232 signal to Host
- Pin 4 - UART_0 - Rx 232 signal to Host

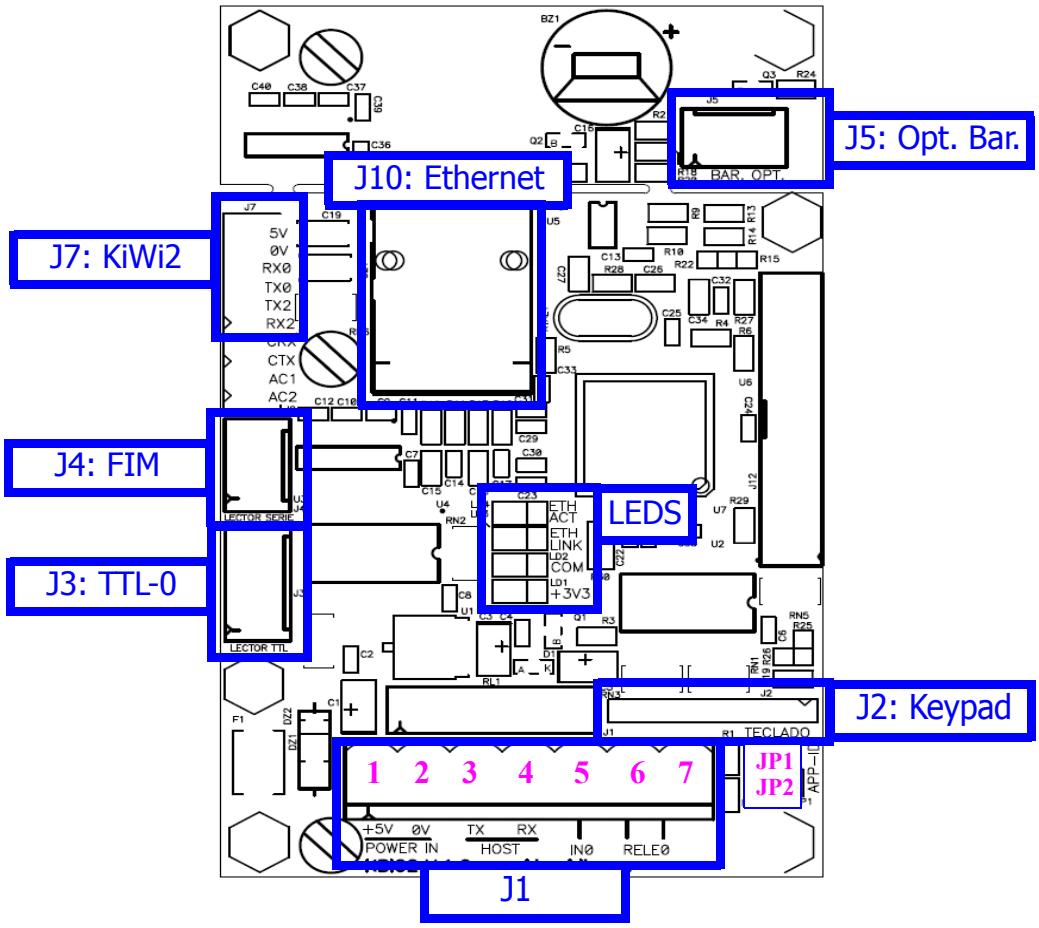
Pin 5 - GND
Pin 6 - 5VDC input/output (*only for KiWi2, BioMaxPlus-DB*).

4.3. Mechanical Outline, BioMax2



Measurements in mm.

4.4. Connector signals, KBio2-Online



WARNING !!!!

KBio2 electronics board requires +5 Vdc supply

Take the necessary antistatic precautions when handling this product to avoid damaging the sensitive electronic devices.

4.5. Connection details, KBio2-Online

J1 - Main Connector:

Pin 1 - Positive Power Supply Pole (**+5 VDC**).
Pin 2 - Negative power supply pole (GND) (Pin no. 5 SubD female connector).
Pin 3 - UART_0 - Tx 232 Signal to Host (Pin no. 2, SubD female connector).
Pin 4 - UART_0 - Rx 232 Signal to Host (Pin no. 3, SubD female connector).
Pin 5 - Digital Input 0.
Pin 6 - Pole A of Relay 0.
Pin 7 - Pole B of Relay 0.

J2: Keypad connector:

Pin 1 - Red LED Anode (+).
Pin 2 - Green LED Anode (+).
Pin 3 - Yellow LED Anode (+).
Pin 4 - Common Cathode (GND).
Pin 5 - Keypad supply (+5 VDC).
Pin 6 - IN Key.
Pin 7 - F1 Key.
Pin 8 - F2 Key.

J3 - TTL_0 Connector to card Reader (NdCAN compatible):

Pin 1 - Data signal (TTL_0)
Pin 2 - Clock signal (TTL_0)
Pin 3 - CLS signal (TTL_0)
Pin 4 - Output +5 VDC
Pin 5 - Negative power supply pole (GND)

J4: Connection to FIM biometric identification module or to generic RS-232 reader

Pin 1 - +5 VDC Output
Pin 2 - UART_1 - Tx 232 signal to reader or FIM Module
Pin 3 - UART_1 - Rx 232 signal to reader or FIM Module
Pin 4 - Power supply negative pole (GND)

J5: JST Connector to Kimaldi Optical Barrier

Pin 1 - Anode (+) of the emitting LED
Pin 2 - Cathode (-) of the emitting LED
Pin 3 - Collector (+) of the receiving phototransistor
Pin 4 - Emitter (GND) of the receiving phototransistor

J7 - Connector/socket for KiWi2 Converter (UART0):

Pin 1 - No Connect
Pin 2 - No Connect
Pin 3 - UART_0 - Tx 232 signal to Host
Pin 4 - UART_0 - Rx 232 signal to Host
Pin 5 - GND
Pin 6 - 5VDC input/output (*only for KiWi2, BioMaxPlus-DB*).

J10: Ethernet Connector (RJ45 connection)

JP1, JP2: Jumpers to select KSP Application_ID (see Subsection 6.3.1.)

LEDs showing the electronics status:

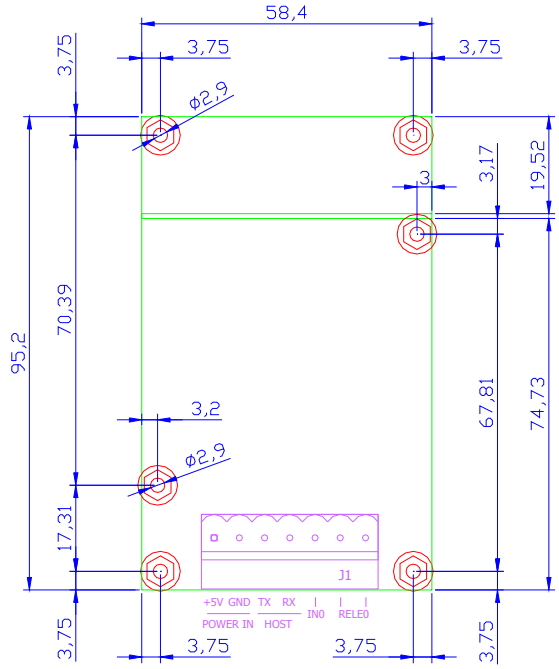
LD1 - 3V3 supply operating

LD2 - Rx/Tx Instructions or Events between *BioMax2* and the Host computer

LD3 - Red - Ethernet Link Error

LD4 - Green - Ethernet Activity (when blinking).

4.6. Mechanical Outline, KBio2-Online



Measurements in mm.

5. Operation

Thanks to its IP connectivity, *BioMax2* and *KBio2-Online* electronics make it possible to install a network of biometric sensors or different types of card readers, accompanied by the corresponding actuators or user interfaces. All the peripherals are Host-managed, except for the biometric search engine, which may optionally be located in the FIM itself [1].

Normal operations consist of sending the events that occur to the Host. Normally, these will be linked to a user ID. The user can simply put his or her finger on the biometric sensor or start the identification by presenting the card or pressing any key. From hereon, the Host will launch the identification and save its result. Once the identity of the person is known, it will be possible to act on an access relay, display a personalised message on screen or even ask for a code using the keycard.

An aspect that deserves special attention is the possibility to connect a choice of RS-232 peripheral devices to *BioMax2*, as indicated in chapter 7.

5.1. Connection of a FIM module

The most usual implementation of *BioMax2* is *BioMax2-FP*, which features a FIM module for biometric identification. The same statement applies to *KBio2-Online*. In both cases, several types of use are allowed:

- Host-based template database:
  - **1-to-N identification (1:N):** the user places his or her finger on top of the biometric sensor. An optical barrier activation event occurs, after which the Host orders the capture of the user's fingerprint. The host will compare the scanned template with all the fingerprints in its database (in the order of 25,000 users).
  - **1-to-1 Identification (1:1):** the user is identified using the card or by entering a code. The Host receives the identification code and retrieves the corresponding identification code from its database. This fingerprint is sent to the biometric sensor, after which the sensor itself scans the user's finger to check its identify with a 1 to 1 comparison (“Instant Matching”).
- FIM-based template database:
  - **1-to-N identification (1:N):** the user places his or her finger on top of the biometric sensor. An optical barrier activation event occurs, after which the Host orders the capture and immediate identification of the user's fingerprint.
  - **1-to-1 Identification (1:1):** the user is identified using the card or by entering a code. The Host receives the identification code and triggers a comparison between a template in the FIM module and the scanned fingerprint.

1. Offline behaviour is allowed when the search engine is located in the FIM. Typically, the offline mode is enhanced by means of the BiomaxPlus-DB auxiliary board. See section 5.5

5.2. Connection to a configurable RS-232 reader

Kimaldi Electronics offers a series of readers (for example, KRD13M and SC42) that have automatic operating modes and that also allow for two-way communications between the Host and the reader (normally for configuration tasks).

BioMax2 and *KBio2-Online* are capable of capturing the events generated by these readers and applying them with the same treatment as if they came from a Clock&Data reader: automatic buzzer activation and LEDs and multiread filtering, etc.. These readings are notified from the Card Reading event.

At the same time, two-way communications are permitted between the Host and the reader from the combination of the Send to UART command and the Reception from UART event.

5.3. Finger presence detection

BioMax2 and *KBio2-Online* electronics support two different means of finger presence detection, depending on the installed hardware:

- **LED optical barrier:** This is the legacy method, the only one available for old FIM versions (FIM20xx, FIM22xx y FIM30xx). The LED barrier is going to be obsoleted. Please find more details in Subsection 5.3.1.
- **Auto-On funcion:** Newer FIM versions (FIM 5360 and later) provide an *Auto-Identify* function based on an embedded capacitive sensor. This is going to be the default detection method for *BioMax2* and *KBio2-Online*. Please find more details in Subsection 5.3.2.

5.3.1. Optical Barrier

BioMax2 and *KBio2* terminals allow you to control an optical barrier to detect the presence of a finger on the biometric sensor. In addition to the normal operating conditions, the optical barrier manages two possible error situations:

1.- *Permanent obstruction of the beam.* In its idle state, the optical barrier detects the beam of light from the LED emitter. If it does not detect the beam for some time, it may be due to a mechanical or electronic fault. In normal conditions, 2 or 3 consecutive identification attempts can be made, without removing your finger from the biometric sensor, before this error is triggered. This parameter can be configured, as explained in the chapter 8.

2.- *Interference by external light sources* . If you expose the terminal to extreme light conditions, the receptor element can be saturated, so it cannot detect the presence of a finger.

Both permanently blocking the beam and saturating the receptor trigger an error situation, in the event of which the Host will receive a notification (event Optical Barrier Status, code 0xED). The yellow LED on *KBio2-Online*'s User Interface ("Ready") will

flash to indicate this. In this situation, you must press the green key ("Enter") to initiate biometric identification.

It will automatically return to normal when the beam of light is no longer obstructed or the external interference is removed (event Optical Barrier Status, code 0x0D).

In order to activate this detection method, it is necessary to configure the following parameters:

- Parameter$03 (*CFG_Opt_T_Watchdog*): it allows activation of the LED optical barrier. So:
  - LED operation: `$80` (LED optical barrier on, obstruction timeout to 12.8 sec.).

If a second FIM module is connected, a special cable assembly allows using a Digital Input to detect finger presence.

5.3.2. FIM Module

BioMax2 and *KBio2-Online* also allow finger presence detection by means of the capacitive sensor embedded both in the OPP06 optical sensor and the capacitive sensors that can be attached to the FIM5360 module (and later).

If two FIM devices are connected to a single *BioMax2* or *KBio2-Online*, the same detection mechanism can be used.

In order to activate this detection method, the following parameters must be correctly configured:

- Parameter$03 (*CFG_Opt_T_Watchdog*): it allows to select the finger detection method, or disable it altogether.
  - Required value: `$FF` (Auto-On enabled).
- Parameter$1B (*CFG_Serial_Port_Reader*): it is necessary to configure UART1 for a FIM module, at the correct baud rate.
  - Possible values: `$88` to `$8B`.
  - Recommended value: `$8A`.
- Parameter$1D (*CFG_Main_Reader*): the FIM module must be declared here.
  - Required value: `$05` (FIM module on UART1 for the main reader).

Optionally, a second FIM module with Auto-On detection may be connected. In this event, UART0 will be used:

- Parameter$1A (*CFG_Serial_Port_Host*):it is necessary to configure UART0 for a FIM module, at the correct baud rate.
  - Possible values: `$88` to `$8B`.
  - Recommended value: `$8A`.
- Parameter$1E (*CFG_Aux_Reader*): the second FIM module must be declared here.
  - Required value: `$04` (FIM module on UART0 for the auxiliary reader).

5.3.3. Setup and operation

Once configured, the *BioMax2* or *KBio2-Online* electronics must be reset (use command ApplyCFG), in order to start operation.

Optical barrier

The LED optical barrier will operate immediately, and it will be promptly generate a Key Pressing event as soon as the infra-red beam is blocked. Any key value may be announced, as configured through Parameter$05 (*CFG_OPT_KeyCode*). Its default value is `$41` ('A' : green key).

Upon finger detection, operation is suspended during a guard time period, which is configured through Parameter$02 (*CFG_OPT_T_GUARD*). Such guard time may be overridden from the application by disabling the optical barrier prior to communication with the FIM module (Disable optical barrier). When done, re-enable operation (Enable optical barrier) to resume detection.

FIM Auto-On

Automatic finger detection by the biometric sensor is a more complex process:

- Upon startup, a boot-up time is required for the FIM module. We may configure such value through Parameter$0E (*CFG_IDNT_-BOOTUP_TIME*). The recommended value depends on the FIM model and the database capacity.
- If a finger is presented during the boot-up period, an Optical Barrier Status event showing status `$FC`, will appear.
- After boot-up, Auto-On starts automatically. If we force Enable optical barrier, we get its current status in response. Value `$00` is the correct operation status, while `$FD` means that boot-up operation did not end correctly, so Auto-On is not available. That can be due to a communication error (baudrate mismatch, broken cable...) or to an old version of FIM that does not support this feature.
- Finger detection occurs when the finger is placed on the surface of the sensor. If there is no clear perception of change (for instance, if the finger does is not removed between one failed identification and the next retry), the finger is not detected. Therefore, it is necessary to remove and re-place the finger for each identification that we try. Upon detection, a Key Pressing event appears. The value of the pressed key is configurable through Parameter$05 (*CFG_OPT_KeyCode*), and its default value is `$41` ('A' : green key).
- Upon finger detection, operation is suspended during a guard time period, which is configured through Parameter$02 (*CFG_OPT_T_GUARD*). Such guard time may be overridden from the application by disabling the optical barrier prior to communication with the FIM module (Disable optical barrier). When done, re-enable operation (Enable optical barrier) to resume detection.

- If Auto-On could not be enabled, an Optical Barrier Status event appears every time we contact the FIM module through Send to UART.
- If two FIM modules are connected to BioMax2, both may activate the Auto-On mechanism. In such case, the key value that is reported by the auxiliary FIM is one count higher than the one of the main FIM (by default, values `$41` for the main FIM and `$42` for the auxiliary FIM are used, corresponding to green and red keys on the BioMax2 keypad).

5.4. Connection to an auxiliary board, Bio-OCX protocol

It is worth remarking that various electronic devices, based on protocol Bio-OCX, can be easily connected to the RS-232 ports. That provides Ethernet connectivity to legacy electronics such as the old *BioMax* and *KBio*, which may be converted to KSP nodes when connected to UART0 of *BioMax2* or *KBio2-Online*.

We may also adapt to very particular installations, for instance by connecting a *Series-16R board* for additional relays.

5.5. BiomaxPlus-DB module

Kimaldi Electronics offers the *BiomaxPlus-DB* module, which provides an offline behaviour, so that it acts as a back-up host when the main one is out of reach.

The *BioMax2* + *BiomaxPlus-DB* assembly is commercially known as *BiomaxPlus*.

The *BioMaxPlus-DB* module is recommended for online access control applications, where offline control must take over if communication to the main Host is lost. In particular, if biometric identification is required, the template database must be located in the FIM module as well.

When using the *BioMaxPlus-DB* module, the following parameters must be correctly configured:

- Parameter$1C: UART2 is configured to link with *BioMaxPlus-DB*.
  - Recommended value: `$D1` (19200 baud communication)
- Parameter$12: It sets the maximum response time (in tenths of second), allowed for Host response, upon event issued by *BioMax2* (key pressed, finger detected, etc.). If the Host does not reply promptly, it is assumed to be lost, and operation mode switches to Offline.
  - Recommended value: `$0A` (1 second)
- Parameter$13: It sets the maximum time between Host instructions (we assume that the Host is regularly sending Keep-Alive frames to *BioMax2*). After such period of time, the Host is assumed to be lost, and operation mode switches to Offline.
  - Recommended value: `$64` (10 seconds)

Commands are available to switch between Online and Offline mode and vice-versa: SetModeOnline and SetModeOffline.

In order for the system to operate properly, the Host must periodically send the SetModeOnline command to BioMaxPlus, so that it keeps its online operating status. Time between such frames must be lower than the one configured on Parameter$13.

5.6. Others

Another basic aspect of *BioMax2* and *KBio2-Online* electronics is the configuration of its interface with the Host. It has several communication channels, as described in chapter 6.: "Comunicación con el Host"

All the aspects related to the operating and connectivity of *BioMax2* and *KBio2-Online* electronics can be configured as explained in chapter 8.: "BioMax2 / KBio2-Online electronics configuration" .

It also incorporates a function for diagnosing faults in the hardware and for remote firmware updating.

5.6.1. Information on exceptions

If an exception occurs in the operation of the microcontroller, the information on it is stored in the non-volatile memory of the *BioMax2 / KBio2-Online* electronics. The second time an exception occurs, the machine restarts automatically.

The most normal thing will be enabling the ApplyCFG event via the *CFG_ECHO* parameter (see Subsection 8.1.2.) and notifying Kimaldi Electronics Technical Service if this event responds to a non-null void frame. The same information can be retrieved at any time using the GetCrashInfo instruction and can then be deleted with ClearCrashInfo.

5.6.2. Remote FW update

Although not directly related to its functionality, it is worth mentioning the possibility to reprogram *BioMax2* or *KBio2-Online*. That must be done by the installation supervisor, either directly through TCP/IP, or through UART0 whenever the IP stack itself must be reprogrammed (UART2 is also a possibility, by means of a special USB-to-UART cable).

Please refer to Kimaldi Partner's Area for further details.

6. Comunicación con el Host

BioMax2 and *KBio2-Online* electronics allows for several communication channels with the Host. Establishing this communication is the first step towards correctly configuring the electronics. *BioMax2* / *KBio2-Online* electronics is permanently listening to these channels and will return a response via the channel from which the command was received.

Configuration parameters discussed in this chapter will be:

- Parameter$0D: *CFG_EventCH*. It allows configuring the communications channel that routes the initial events, before the first frame is received from the Host (for instance, Key Pressing). Default value: `$FF`
- Parameter$1A: *CFG_Serial_Port_Host*. Default value: `$91`
- Parameter$1C: *CFG_Serial_Port_Aux*. Default value: `$F0` (disabled).

The detailed programming model may be found on Table 3 and chapter 8.1.

6.1. Communication with the Host via RS-232

The UART0 port corresponds to communications with the Host via RS-232 by default. This communication is established at 19200 bps, with 8 bits, no parity and one stop bit by default. The serial frames will consist of ASCII-Hex values according to the frame format generated by the Bio-OCX controller (see chapter 9).

From the *CFG_Serial_Port_Host* parameter, however, there are other alternatives:

- Communication speed can be configured to 9600, 19200, 38400 or 57600 baud. To do so, the *CFG_Serial_Port_Host* byte should take the `$90`, `$91`, `$92` or `$93` values respectively.
- Values other than those indicated above are used for many purposes, as will be seen later. In these cases, the UART0 port stops interacting with the Host and the UART2 automatically takes its place.

In cases where the use of adapter cards for other protocols is required (e.g. WiFi, TCP/IP and Bluetooth, etc.), this will be done from this UART0 port. Whenever connecting to the Host via UART0, it is advisable to suitably configure the `$10` parameter of the IP configuration (see Subsection 8.2.1.).

The *CFG_EventCH* parameter should take the `$00` value for the *BioMax2* / *KBio2-Online* electronics to spontaneously transmit Online frames through this channel.

The UART2 is enabled to maintain connectivity with the Host in configurations in which *CFG_Serial_Port_Host* takes a value other than `$90`, `$91`, `$92`, etc. Then, *CFG_Serial_Port_Aux* automatically takes the `$91` value, which allows connection to the Host in the event of emergencies (e.g. if you are unable to reconfigure IP communications via the Kimaldi Localisation Service). Given that the UART2 operates at TTL levels, a USB adaptor cable supplied by Kimaldi is necessary.

6.2. IP Communications

To access the IP communication services, the network parameters must be configured correctly. Configuration can be carried out using the Bio-OCX protocol, via RS-232. In this case, we shall deal with the parameters described in section 8.2. Alternatively, the Kimaldi Localisation Service detailed below can be used. Once the equipment is configured, there will be one TCP/IP Socket and two different protocols via UDP Sockets.

6.2.1. Kimaldi Localisation Service

It is possible to detect the *BioMax2 / KBio2-Online* electronics connected to your local area network using the Kimaldi Localisation Service:

- IP configuration: once the unit is located from its MAC Address, it is possible to configure its IP parameters and restart the unit.
- The localisation request from the Host to the *BioMax2 / KBio2-Online* electronics is made via port 2000.
- The frame reception in the Host from the *BioMax2 / KBio2-Online* electronics is made using port 2001.

There is a DLL that allows this Service to be integrated into any software application. For further information on the low-level protocol, please see the SLK Programming Manual.

6.2.2. Bio-OCX protocol via TCP (TCP Server)

The Bio-OCX protocol is also available via a TCP Socket, which makes it fully compatible with the KiWi2 convertor module:

- *BioMax2 / KBio2-Online* electronics is in server mode. Therefore, frames from any Host can be accepted (please see *SLK-Safety* in section 8.2) The connection socket will be started by the Host. *BioMax2 / KBio2-Online* electronics will only generate TCP events while this socket is enabled.
- Commands from the Host to the *BioMax2 / KBio2-Online* electronics are transmitted through an arbitrary Host port (*Local Port* of the Host).
- The frames are received at the *BioMax2 / KBio2-Online* electronics via port 1001 (*Remote Port* from the Host).

The serial frames will consist of ASCII-Hex values according to the frame format generated by the Bio-OCX controller (see chapter 9).

The *CFG_EventCH* parameter is not applicable in this case, as TCP events cannot be generated spontaneously (i.e. events will not be generated until the Host has started the socket).

In contrast, it is possible to have an open TCP against a different computer to the one declared as the Remote Host. Great care must be taken in this case, as there could be two Hosts acting simultaneously on the *BioMax2 / KBio2-Online* electronics.

6.2.3. Bio-OCX protocol via UDP

Finally, the Bio-OCX protocol is also available via a UDP Socket:

- Commands from the Host to the *BioMax2 / KBio2-Online* electronics are transmitted via the port chosen from the Host (*Local Port* of the Host). From a *BioMax2 / KBio2-Online* electronics viewpoint, this is the Remote Host Port that can be configured (see section 8.2, Parameter$07) and that is 0 by default (automatic assignment).
- The frames are received at the *BioMax2 / KBio2-Online* electronics via port 5500 (*Remote Port* from the Host).

The frame format will be binary and will not have the `<STX>, <ETX>` delimiters.

The *CFG_EventCH* parameter should take the `$07` value for the *BioMax2 / KBio2-Online* Electronics to spontaneously transmit Online frames through this channel.

6.2.4. KSP protocol via UDP

Finally, *BioMax2 / KBio2-Online* electronics allows for work with the new *Kimaldi Stackable Protocol* (KSP). KSP operates via another UDP Socket:

- Commands from the Host to the *BioMax2 / KBio2-Online* electronics are always transmitted via Host port 6001 and using Broadcast IP frames. This means that a special frame format will be used that will allow us to specify the node at KSP level.
- The frames are received at the *BioMax2 / KBio2-Online* electronics via port 6000 (*Remote Port* from the Host).
- One significant advantage in relation to the Bio-OCX protocol is that the Host may receive all Online Event frames generated from any KSP mode via just one Socket on port 6001. These frames also carry the address of the node to have generated them.

There is an ActiveX control to interact with the *BioMax2 / KBio2-Online* electronics via KSP. Please see the corresponding manual for further details.

The *CFG_EventCH* parameter should take the `$08` value for the *BioMax2 / KBio2-Online* Electronics to spontaneously transmit Online frames through this channel.

6.2.5. KSP protocol via TCP (TCP-client)

On Subsection 6.2.2. we saw the TCP socket working in Server Mode. That means, requests coming from any host can be accepted. The opposite option is to enable a server application on the Host side (that is achieved throug KSP-OCX or the equivalent .NET library) and to configure *BioMax2* or *KBio2-Online* as clients. In order to do so, let us:

- Define the *IP-RemoteHost* (the same that is used for UDP sockets).
- Define the Host's port to which *BioMax2 / KBio2-Online* will connect (*TCP Port-RemoteHost*). Typically, that is port number `6601`.
- Communication to *BioMax2* o *KBio2-Online* is done through port number `6600`.

The *CFG_EventCH* parameter should take the `$0C` value for the *BioMax2 / KBio2-Online* Electronics to spontaneously transmit Online frames through this channel.

6.3. Routing and Localisation

BioMax2 / KBio2-Online electronics can be part of a local IP network, supporting several communication protocols. There are several routing modes for this reason. The different *BioMax2 / KBio2-Online* electronics addresses can be consulted in the configuration instructions (also see section 8.2) or Kimaldi Localisation Service (see SLK - Programming Manual).

6.3.1. Routing hardware

In terms of Hardware, the *BioMax2 / KBio2-Online* electronics have two jumpers (JP2 and JP1; see chapter 4.2.: "Connection details, BioMax2" or chapter 4.4.: "Connector signals, KBio2-Online") to encode the high part (*Application_ID*) of the bus address. Its low part is encoded through a byte in the configuration.

The extended address of the node shall be binary encoded for the 10 bits: Jumpers (2,1), *CFG_KSP_AddrLo*. With JP2 being the heaviest bit, followed by JP1 and then the configuration byte. The jumpers take logic value 1 when the bridge is on and 0 if not.

The *BioMax2 / KBio2-Online* electronics reads the configuration of the jumpers on receiving the power supply. Therefore, if the address is to be changed, the power supply should be switched off for a moment or and an ApplyCFG instruction sent for the changes to take effect.

6.3.2. KSP Address

If we communicate with the *BioMax2 / KBio2-Online* electronics via KSP (Subsection 6.2.4.), the address corresponding to it is determined in the following way:

- *KSP Application:* this has a value between 11 and 14 (`$0B` and `$0E`), obtained by adding 11 to the JP1, JP2 value.
- *KSP ID:* this is an 8 bit value ranging from 1 to 254 (`$01` and `$FE`) that is expressed by the configuration byte *CFG_KSP_AddrLo*.

6.3.3. MAC Address

The MAC Address of each *BioMax2 / KBio2-Online* electronics is unique and assigned by the manufacturer. It is labelled above the electronics in hexadecimal format.

6.3.4. IP Address

The IP address of each *BioMax2 / KBio2-Online* electronics will be user-assigned based on the characteristics of the local area to which it is connected. This is generally a configuration field, which can be user-modified through the Kimaldi Localisation Service or any service for connection to the Host (see all configuration fields in section 8.2).

7. Communication with RS-232 devices

BioMax2 and *KBio2-Online* electronics have two RS-232 ports that can be used for different purposes.

As explained in chapter 6.1, UART0 maintains the possibility of connection to the Host so that total compatibility between *BioMax* and *BioMax2* is ensured. However, communication to Host will normally occur through the IP ports, so that UART1 and usually UART0 will be available for connection to peripheral RS-232 devices.

To date, these devices will be of the following types:

7.1. Biometric identification module

BioMax2 and *KBio2-Online* electronics are specially suited for biometric identification. For that purpose, a FIM module must be connected to UART1 (connector J7 on chapter 4.1 for BioMax2; connector J4 on chapter 4.4 for KBio2), so that we can capture and verify biometric templates [1].

The template database for biometric identification may be stored in the Host (eNBSP libraries from Nitgen), or in the FIM module itself, depending on the particular needs of the installation.

The Bio-OCX controller allows a simple operation with the FIM modules. We must also configure the following parameters:

- Parameter$1B: UART1 configuration.
  - Recommended value: `$8A` (FIM module at 38400 baud)
- Parameter$1D: Main FIM activation.
  - Recommended value: `$05` (it allows Auto-On and GCR commands)
  - Legacy value: `$FF` (LED optical barrier)

More information on Subsection 8.1.1.

7.2. RS-232 Reader

7.2.1. "Read-only" readers

A reader that transmits the user identification code through the serial port can normally be connected either to UART0 or UART1. This would be, for example, a proximity reader, magnetic band or bar code, etc. In these cases, the data received will be processed the same as if it were to arrive via Clock&Data ports. This means that the green and red LEDs can be used to provide information on the success or error of the reading process, that multiread filtering can be applied and that in any case a Card Reading event will be generated. Subsection 8.1.4. shows that the data is processed in the same way, whether it

1. We may also connect the FIM module to UART0, but then we must use Terminal Block J3 (BioMax2), and configure Parameter$1A *CFG_Serial_Port_Host* to $8A, and Parameter$1D to $04.

is from a Clock&Data port or from a suitably configured UART (*CFG_Serial_Port_XXXX* parameters to value `$40` or `$41`, see Table 3).

7.2.2. Online readers

Nevertheless, more sophisticated readers do exist. These are those that can generate an event with the identification code but that, at the same time, allow for two-way communications with the Host to be programmed.

Within this type are the Kimaldi readers KRD13M and S42, which have programmable automatic modes capable of extracting a data sequence from a Mifare or SLE4442 card, respectively. In this case, the host may issue instructions to manage reader configuration:

- The data frames are sent via the Send to UART instruction.
- The responses are received via the Reception from UART event.
- Codetag events (ASCII frames) are received via Card Reading.
- The data format is binary, doing without `<STX>, <ETX>.`
- Connection via UART0 is established after configuring the Parameter$1A *CFG_Serial_Port_Host* to the value `$40`, `$41` or `$42`, according to the baud rate. The maximum length of the frame is **48 bytes**.
- Connection via UART1 is established after configuring the Parameter$1B *CFG_Serial_Port_Reader* to the value `$40`, `$41` or `$42`, according to the baud rate. The maximum length of the frame is **48 bytes**.
- Moreover, identification events will be treated as in "Read-only" readers: Card Reading event, automatic LED and beeper activation, multiread filtering...

7.2.3. Omnikey 55xx Readers

Starting on FW version 0x61.39, Omnikey 55xx Plug&Play RS-232 readers also allow the Card Reading functionality, along with the rest of the features explained in "Read-only" readers. Online, bidirectional communication is also supported.

In order to do that, the following parameters must be configured:

- The corresponding UART (Parameter$1A *CFG_Serial_Port_Host* or Parameter$1B *CFG_Serial_Port_Reader)* to value `$30` or `$31`, according to the baud rate (see Table 3).
- We must also activate the corresponding UART bit at the *CFG_Reader_Enable* parameter, and related ones (See Subsection 8.1.4., Card reading port configuration).
- In particular, we must configure Parameter$0C *CFG_Multiread_TMO* (also on Subsection 8.1.4.), in order to define the appropriate behaviour of the multiple Card Reading events, when the card is permanently placed on the reader.

7.3. Gateway to Bio-OCX or KPLUS2 electronics

A variety of Kimaldi electronic boards do not provide for integrated IP communication, since they only have a serial port. *BioMax2/KBio2-Online* may act as a gateway to indirectly connect them to a LAN. Moreover, it is a convenient way to enhance funcionality of *BioMax2/KBio2-Online*.

7.3.1. Supported electronics, Bio-OCX protocol

The following models can be connected to *BioMax2/KBio2-Online*:

- Serial-xR card: -1R, -4R, 16R (must be configured as Bio-OCX).
- BioMax, BioMax-FP.
- KBio: normally in its Online version, also Offline.

7.3.2. Supported electronics, KPLUS2 protocol

Starting on FW 0x61.43 KRD13Mv2 readers through KPLUS2 protocol can be connected. The goal is to implement Online Mifare readers, allowing high-speed read/ write transactions to Mifare cards, over IP.

7.3.3. Communication via Bio-OCX UDP

When using Bio-OCX protocol over UDP (port 5500 on *BioMax2/KBio2*):

- We contact the peripheral electronics via Send to UART.
- Answers and events are collected through Reception from UART.
- The frame format will be as follows, in binary and without <STX>,<ETX>:

```
<OPC><NA_high><NA_low><ARG_0><ARG_1>....<ARG_NA><CRC>
```

- For BioMax, KBio, the recommended connection is through UART0. Parameter$1A *CFG_Serial_Port_Host* must take value $99. Maximum frame length is **912 bytes**, suitable to manage the attached FIM module.
- Serial-xR boards may be connected through UART0 or UART1. Parameter$1A *CFG_Serial_Port_Host* or Parameter$1B *CFG_Serial_Port_Reader* must take value $98. Maximum frame length is **49 bytes**.
- KRD13Mv2 readers may be connected through UART0 or UART1. Parameter$1A *CFG_Serial_Port_Host* or Parameter$1B *CFG_Serial_Port_Reader* may take values $20, $21 or $22 according to the baud rate. For maximum speed, 38400 baud recommended ($22). Maximum frame length is **49 bytes**.
- Frame format for KPLUS2 (KRD13Mv2) is as follows, in binary and without <STX>,<ETX>:

```
<OPC><NA><ARG_0><ARG_1>....<ARG_NA><CRC>
```

7.3.4. Communication via KSP UDP

When using protocol KSP-UDP (port 6000 on *BioMax2/KBio2*), everything explained in the previous section still applies:

- Parameters *CFG_Serial_Port_Host* or *CFG_Serial_Port_Reader* must be properly configured (`$98` or `$99` according to the baud rate for Bio-OCX, `$20`, `$21` o `$22` according to the baud rate for KRD13Mv2).
- Send to UART and Reception from UART may be used for communication.

However, a more convenient method exists:

- Our *BioMax2/KBio2* board has a KSP address (see Subsection 6.3.2.), made of an Application_ID between 11 and 14, plus Node_ID between 1 and 254. For instance, 14.4 (`$0E.$04` in Hexadecimal)
- In such case, the peripheral board connected to UART0 will have a KSP address incremented in 64 units (`$40`) as its Application_ID. For instance, 78.4 (`$4E.$04`).
- And the peripheral board connected to UART1 will have a KSP address incremented in 128 units (`$80`) as its Application_ID. For instance, 142.4 (`$8E.$04`).

That way, we may operate with up to 3 different electronic boards, through KSP (but only the main one will appear on Kimaldi Localisation Service)

8. BioMax2 / KBio2-Online electronics configuration

BioMax2 or *KBio2-Online* electronics configuration consists of two parts:

- Parameter array: these are an array of bytes that allow certain machine characteristics to be defined in terms of function (timings, parameter enabling/ disabling, etc.).
- IP configuration: data required for machine operations in a local area network (Ethernet).

BioMax2 / *KBio2-Online* electronics configuration is stored in EEPROM memory, which provides it with around 10,000 read/write cycles. This should be taken into account when managing the configuration parameters.

8.1. Parameter array

All the machine's parameters are listed below:

Table 1: BioMax2 / KBio2-Online parameter list

No.	Description	Further info
$01	Defines the events reported to the Host	See Subsection 8.1.2.
$02	Guard time, in tenths of a second, between finger detection events.	See Subsection 8.1.3.
$03	Watchdog, in tenths of a second, for the detection of the permanent blocking of the optical barrier.	See Subsection 8.1.3.
$04	Time-Out, in tenths of a second, of reception of data by the UART (FIM or RS232 reader).	See Subsection 8.1.2.
$05	The optical barrier will generate a Key Pressing event corresponding to a configurable key value.	See Subsection 8.1.3.
$06	Duration, in hundredths of a second, of the buzzing of the internal beeper to indicate the pressing of a key.	See Subsection 8.1.5.
$07	Duration, in hundredths of a second, of the buzzing of the internal beeper to indicate an error.	See Subsection 8.1.5.
$08	Enables RS-232 (UART0, UART1) and Clock&Data (TTL_0, TTL_1) ports.	See Subsection 8.1.4.
$09	Configuration byte of the automatic beeper enabling after a correct Clock&Data reading.	See Subsection 8.1.4.
$0A	Configuration byte of the automatic beeper enabling after an incorrect Clock&Data reading.	See Subsection 8.1.4.
$0B	Configuration of the multiread filtering.	See Subsection 8.1.4.

$0C	Time-out in tenths of a second of the multiread filtering, to detect that a card has left the reading field.	See Subsection 8.1.4.
$0D	Configures the communications channel through which the event frames are sent.	See Subsection 8.1.2.
$0E	FIM Boot-Up time	
$0F	Configure TTL_x ports as Wiegand Inputs	See Subsection 8.1.4.
$10 a $11	Reserved	
$12	Host reply Time-Out, upon event generated by BioMax2 / KBio2-Online, before switching to Offline Mode	See Subsection 5.5.
$13	No-traffic Time-Out, before switching to Offline Mode	See Subsection 5.5.
$14	Display response Time-Out	See Subsection 8.1.6.
$15 to $19	Reserved	
$1A	Determines the protocol and Baud Rate assigned to the UART0. See Table 3.	See Subsection 8.1.1.
$1B	Determines the protocol and Baud Rate assigned to the UART1. See Table 3.	See Subsection 8.1.1.
$1C	Determines the protocol and Baud Rate assigned to the UART2. See Table 3.	See Subsection 8.1.1.
$1D	Enables main FIM with Auto-On detection	See Subsection 8.1.3.
$1E	Enables auxiliary FIM with Auto-On detection	See Subsection 8.1.3.
$1F	Time, in tenths of a second, to indicate correct reading with the green LED.	See Subsection 8.1.5.
$20	Time, in tenths of a second, to indicate incorrect reading with the red LED.	See Subsection 8.1.5.
$21	Time, in tenths of a second, to indicate correct reading with the buzzing of the external beeper.	See Subsection 8.1.5.
$22	Time, in tenths of a second, to indicate incorrect reading with the buzzing of the external beeper.	See Subsection 8.1.5.
$23	Reserved for future uses.	

8.1.1. Serial port configuration (UART0, UART1, UART2)

The serial ports have a different transmission capacity according to the programmed mode:

- UART0 allows frames of up to 912 bytes in all cases
- UART1 allows frames of up to 912 bytes in transparent mode (configuration values `$88` to `$8C`), but only up to 48 bytes in the other modes.
- UART2 allows frames of up to 48 bytes, but up to 912 bytes when connected to he Host. UART2 is a special port, only available at TTL levels (3.3V), intended for connection to the *BiomaxPlus-DB* module, for Offline behaviour add-on.

The behaviour of the serial ports is defined from the `$1A`, `$1B` and `$1C` configuration parameters:

Table 2: RS-232 port configuration

Num		Description	Def.
$1A	*CFG_Serial_Port_Host*	Determines the protocol and Baud Rate assigned to the UART0. Refer to Table 3	$91
$1B	*CFG_Serial_Port_Reader*	Determines the protocol and Baud Rate assigned to the UART1. Refer to Table 3	$88 [a]
$1C	*CFG_Serial_Port_Aux*	Determines the protocol and Baud Rate assigned to the UART2. Refer to Table 3	$F0

a. OEM electronic boards are configured at 9600 baud by default. *BioMax2* terminals (already assembled) are configured at the optimum speed for the FIM, typically 38400 baud ($8A).

Depending on the function associated to the port and baud rate, the values to be assigned are detailed in the following table.

Table 3: Reader configuration codes

Tipo de lector	*CFG_Serial_Port_XXX* [a]	*Detalles*
Communication with the Host, Bio-OCX protocol	$90: 9600 baud $91: 19200 baud $92: 38400 baud	Only UART0 or alternatively UART2 [b]
FIM module [c] / RS-232 Online Reader	$88: 9600 baud $89: 19200 baud $8A: 38400 baud $8B: 57600 baud	Transparent frames, no formatting considerations. - UART0 up to 912 bytes - UART1 up to 912 bytes
RS-232 Card Reader (Automatic Mode)	$40: 9600 baud $41: 19200 baud	*CFG_Reader_Enable* [d] must be enabled too
Omnikey 55xx Reader (Automatic Mode)	$30: 9600 baud $31: 19200 baud	*CFG_Reader_Enable* and *CFG_Multiread_TMO* [e] must be configured too
KRD13Mv2 Reader (Online Mode)	$20: 9600 baud $21: 19200 baud $22: 38400 baud	Disable corresponding UART on parameter *CFG_Reader_Enable*
Serial-xR card	$98: 9600 baud	UART0 or UART1
KBio / BioMax	$99: 19200 baud	Only UART0 (912 bytes)
Communication to *BiomaxPlus-DB* module	$D0: 9600 baud $D1: 19200 baud $D2: 38400 baud $D3: 57600 baud	Only UART2
UART disabled	$F0	

a. *CFG_Serial_Port_Reader* corresponds to UART1 and *CFG_Serial_Port_Host* to UART0
b. If UART2 is disabled, it automatically reconfigures to connect to the Host ($91), when UART0 is not configured for that role.
c. Configuration parameters related to FIM modules are typically set during the assembly process and should not be modified. The FIM module typically connects to UART1.
d. Recommended for readers with Automatic Mode (for instance, KRD13M or SC42). See chapter 7.2 and Subsection 8.1.4.
e. Use with Omnikey 55xx Plug&Play RS-232. See chapter 7.2 and Subsection 8.1.4.

8.1.2. Event configuration

BioMax2 and *KBio2-Online* electronics are able to start communications to the Host when any event occurs in the machine (e.g. a key is pressed). The following parameters can be configured:

Table 4: Event configuration

No.		Description	Def.
$01	*CFG_ECHO*	Defines the events reported to the Host	$50
$04	*CFG_ SCITimeout*	Time-out of the reception of the UART from the RS232 reader (UART1, UART0)	$FF
$0D	*CFG_EventCH*	Communications channel through which the electronics will generate the event frame. Refer to Table 5	$FF

The following section explains the meaning of each of these configuration bytes.

Parameter$01: **ECHO**. This generates an OpCode for each status change (digital inputs, time-locking relays and optical barrier, among others). We can configure from which type of signal we want to obtain the Echo.

Permitted values:

- Bit 3: ECHO ON – Reset Output.
- Bit 4: ECHO ON – Status change in digital inputs.
- Bit 5: ECHO ON – Ends digital output timing (Relay, LED, buzzer).
- Bit 6: ECHO ON – Optical barrier (notifies erroneous status)
- Bit 7 (MSB): ECHO ON – Reserved (keep at 0)
- 0x00: ECHO OFF: - silent (only responds if it is asked for the digital inputs).

Default value:

- 0x50: ECHO ON, for digital inputs and optical barrier

Parameter$04: **SCI Timeout**. This configures the maximum time-out between consecutive characters received by the terminal from the RS232 reader to be interpreted as part of the same command frame.

Permitted values:

- 0x01 .. 0xFE: Time-out value in tenths of a second.
- 0x00, 0xFF: Automatic time-out value (from 150 ms for 9600 baud and 20 ms for 115200 baud)

Default value:

- 0xFF: Automatic time-out.

Parameter$0D: **Canal de Eventos**. Normally, the *BioMax2/KBio2* electronics always responds through the same channel from which the instruction is received. However, there are situations (typically after Reset) where there is no prior instruction and therefore we have to refer to *CFG_EventCH* to determine this output channel.

Normalmente, la *BioMax2/KBio2* responde siempre por el mismo canal de donde procede la instrucción. Sin embargo, existen situaciones (típicamente, después de Reset) donde no existe una instrucción previa y por tanto debemos referirnos a *CFG_EventCH* para determinar dicho canal de salida.

Table 5: Event Channel identifier

CFG_EventCH	**Communications channel**
$00	UART0
$07	UDP - Bio-OCX (port 5500) [a]
$08	UDP - KSP (port 6000)
$0C	TCP - KSP (port 6600)
$FF	Ninguno

a. This requires specifying a remote port other than 0

Default value:

- 0xFF: Events are not generated until the Host has started communicating.

Note that a spontaneous TCP/IP output cannot be defined.

8.1.3. Optical barrier configuration

Table 6: Optical barrier configuration

Num		Description	Def.
$02	*CFG_OPT_T_GUARD*	Time-out between finger detection events	$14
$03	*CFG_OPT_T_WATCHDOG*	Watchdog for permanent blockage detection	$00 $FF
$05	*CFG_OPT_KeyCode*	The optical barrier will generate a Key Pressing event corresponding to a configurable key value	$41
$1D	*CFG_Main_Reader*	Auto-On Finger Detection is enabled for the Main FIM (UART1)	$FF $05
$1E	*CFG_Aux_Reader*	Auto-On Finger Detection is enabled if an auxiliary FIM (UART0) is connected	$FF

Parameter$02: **Optical Guard Time**. This configures the optical barrier guard time: when the optical barrier has detected the presence of the finger, it is disabled for a few seconds. This allows for biometric identification, with no new finger detection events occurring. If the guard time runs out and the finger is still blocking the optical barrier, a new event will be generated.

Permitted values:

- 0x00 .. 0xFE: Guard time value in tenths of a second.

Default value:

- 0x14: Guard time of 2 seconds.

Parameter$03: **Optical Watchdog Timer**. In its standby state, the optical barrier detects the beam of light from the LED transmitter. If it does not detect the beam for some time, it may be due to a mechanical or electronic fault. In this situation, you must press the green key (“Enter”) to initiate biometric identification. When the Watchdog Timer of the optical barrier is triggered, the Host receives a notification (Optical Barrier Status event, code 0xED). It will automatically return to normal when the beam of light is no longer obstructed (Optical Barrier Status event, code 0x0D). The default value of the Watchdog Timer allows for 2 or 3 consecutive identification attempts to be made without removing your finger from the biometric sensor.

This parameter is also used to enable Auto-On detection, or to disable finger detection altogether.

Permitted values:
- 0x00: This disables the optical barrier.
- 0x01 .. 0xFE: Delay value in tenths of second.
- 0xFF: Detect finger presence through the FIM module (Auto-ON).

Default value:
- 0xFF: Auto-On Finger Detection through FIM.

Legacy value:
- 0x80: This disables the optical barrier after 12.8 seconds without detecting the beam (only for terminals with old FIM models and LED optical barrier).

Parameter$05: **Optical Key Code**. The presence of a finger blocking the optical barrier will generate an event equivalent to a Key Pressing. The key code is configurable, although we recommend comparing finger presence to the pressing of the green key. For this reason, the default value is 0x41 (corresponding to 'A').

Permitted values:
- 0x00 .. 0xFF: Any, although we recommend an alphanumerical value.

Default value:
- 0x41: 'A', is equivalent to the pressing of the green "Enter" key.

Parameter$1D: **Main Reader**. If a biometric FIM module is connected and Auto-On Finger Detection must be enabled, we must configure this parameter.

Permitted values:
- 0x04, 0x05: Main FIM connected to UART0, UART1 (respectively).
- 0x00, 0xFF or other: Auto-On detection disabled.

Default value:
- 0x05: Main FIM connected to UART1 (typically, J7 connector on BioMax2).

Parameter$1E: **Auxiliary Reader**. If a second biometric FIM module is connected and Auto-On Finger Detection must be enabled, we must configure this parameter.

Permitted values:
- 0x04, 0x05: Main FIM connected to UART0, UART1 (respectively).
- 0x00, 0xFF or other: Auto-On detection disabled.

Default value:
- 0x00: No Auxiliary FIM by default.

8.1.4. Card reading port configuration

Table 7: Card reading port configuration

Num		Description	Def.
$08	*CFG_Reader_Enable*	Enables RS-232 (UART0, UART1) and Clock&Data (TTL_0, TTL_1) ports	$8C
$09	*CFG_Reader_BuzOK*	Automatic activation of the Beeper after a correct Clock&Data reading	$0C
$0A	*CFG_Reader_BuzError*	Automatic activation of the Beeper after an incorrect Clock&Data reading	$0C
$0B	*CFG_Reader_Multiread*	Multiread filtering	$00
$0C	*CFG_Multiread_TMO*	Time-out in tenths of a second of the multiread filtering to detect that a card has left the reading field.	$28

Parameter$08: **Reader Enable**. Enables and configures the operation of the Clock&Data ports.

Permitted values:

- Bit 0 (LSB): UART0 ON – card reader enabled in UART0. This also requires that *CFG_Serial_Port_Host* is configured accordingly (e.g., with a value of `$40`; refer to Table 3)
- Bit 1: UART1 ON – card reader enabled in UART1. This also requires that *CFG_Serial_Port_Reader* is configured accordingly (e.g., with a value of `$40`; refer to Table 3)
- Bit 2: TTL_0 ON – Clock&Data TTL_0 enabled.
- Bit 3: TTL_1 ON – Clock&Data TTL_1 enabled.
- Bit 4: `'$'` before UART0 – `'$'` is in front of the data received via UART0.
- Bit 5: `'&'` before UART1 – `'&'` is in front of the data received via UART1.
- Bit 6: `'*'` before TTL_0 – `'*'` is in front of the data received via TTL_0.
- Bit 7: `'#'` before TTL_1 – `'#'` is in front of the data received via TTL_1.

Default value:

- 0x8C: TTL_0, TTL_1 enabled; `'#'` before TTL_1, TTL_0 without prefix.

Parameter$09: **Buzzer on OK**. Buzzer configuration for correct readings. We can enable the buzzer assembled on the *BioMax2 / KBio2-Online* (internal), or a set of external buzzer and green LED, so that they are enabled when a correct reading has occurred.

Permitted values:

- Bit 0 (LSB): Internal buzzer enabled for UART0.
- Bit 1: Internal buzzer enabled for UART1.
- Bit 2: Internal buzzer enabled for TTL_0.
- Bit 3: Internal buzzer enabled for TTL_1.
- Bit 4: External green LED and buzzer enabled for UART0.
- Bit 5: External green LED and buzzer enabled for UART1.
- Bit 6: External green LED and buzzer enabled for TTL_0.
- Bit 7: External green LED and buzzer enabled for TTL_1.

Default value:

- 0x0C: Internal buzzer enabled for TTL_0 and TTL_1.

Parameter$0A: **Buzzer on Fail**. Buzzer configuration for incorrect readings. We can enable the buzzer assembled on the *BioMax2 / KBio2-Online* (internal), or a set of external buzzer and red LED, so that they are enabled when an incorrect reading has occurred.

- Bit 0 (LSB): Internal buzzer enabled for UART0.
- Bit 1: Internal buzzer enabled for UART1.
- Bit 2: Internal buzzer enabled for TTL_0.
- Bit 3: Internal buzzer enabled for TTL_1.
- Bit 4: External red LED and buzzer enabled for UART0.
- Bit 5: External red LED and buzzer enabled for UART1.
- Bit 6: External red LED and buzzer enabled for TTL_0.
- Bit 7: External red LED and buzzer enabled for TTL_1.

Default value:

- 0x0C: Internal buzzer enabled for TTL_0 and TTL_1.

Parameter$0B: **Multiread**. This enables multiread filtering of the Clock&Data and RS232 inputs. Multiread filtering is useful for when we have a reader configured in multiread mode connected to the card (every X time the reader checks the presence of the card and if it is present it generates an event with the card's identification information). This mode allows for filtering all the events that correspond to a same card and only generates events when it varies with the reading data.

Permitted values:
- Bit 0 (LSB): Multiread filter enabled for UART0.
- Bit 1: Multiread filter enabled for UART1.
- Bit 2: Multiread filter enabled for TTL_0.
- Bit 3: Multiread filter enabled for TTL_1.
- Bits 4, 5: When set, EMC FW filter for TTL_0, TTL_1 (respectively) is disabled. *Set only when recommended by Kimaldi.*
- Bits 6, 7: not in use

Default value:
- 0x00: Multiread filter disabled, EMC FW filter enabled.

Parameter$0C: **Multiread time-out**. This defines the time required for determining that a card has left the field of a reader configured in multiread. When this time is up, the *BioMax2 / KBio2-Online* electronics generates a Card Reading event with a null identification value (refer to Subsection 9.6.18.).

Permitted values:
- The time-out value should exceed the reading repetition time.

Default value:
- 0x28: Null Card Reading event 4 seconds after the last correct reading.

Parameter$0F: **Wiegand configuration**. The Clock&Data ports may be turned into Wiegand inputs. As usual, *BioMax2/KBio2-Online* generates a Card Reading event whenever a correct Wiegand frame is received (parity bits are verified). Prefix insertion or beeper activation are configured through the same parameters and bits that correspond to TTL_x (parameters `$08` to `$0A`).

Permitted values:
- Bits 0 to 2: incoming Wiegand Format (See Wiegand format explanation, in Appendix A.3.3). Possible values:
  - *b000* = `$0`: Free Wiegand format. *No parity check.*
  - *b001* = `$1`: Wiegand-26 bits format
  - *b010* = `$2`: Wiegand-34 bits format
- Bit 3: Reserved.
- Bit 4: when set, port TTL_0 is configured as Wiegand input.
- Bit 5: when set, port TTL_1 is configured as Wiegand input.
- Bits 6, 7: Reserved.

Default value:
- 0x00: Wiegand reception disabled.

8.1.5. Digital output timing

Table 8: Digital output timing

Num		Description	Def.
$1F	*CFG_TempGREEN_OK*	Green LED timing upon correct reading	$0A
$20	*CFG_TempRED_Fail*	Red LED timing upon incorrect reading	$14
$21	*CFG_TempBuzz_OK*	Buzzer timing upon correct reading	$02
$22	*CFG_TempBuzz_Fail*	Buzzer timing upon incorrect reading	$06
$23	*CFG_TempBacklit*	Display backlighting	$32

For all the parameters described in the table, the configuration value corresponds to the time in tenths of second that they are to remain enabled:

- Whenever a port enabled in *CFG_Reader_BuzOK* receives a correct reading, the external green LED (*CFG_TempGREEN_OK*) and the buzzer will activate (*CFG_TempBuzz_OK*) .
- Whenever a port enabled in *CFG_Reader_ BuzError* receives an incorrect reading, the external red LED (*CFG_TempRED_Fail*) and the buzzer will activate (*CFG_TempBuzz_Fail*).
- When the display is written on (Write Display), the backlighting will be automatically enabled.

Permitted values:

- 0xFF: Permanently enabled.
- 0x01 .. 0xFE: Timing in tenths of a second.
- 0x00: Timing disabled.

Default value: depending on parameter (see table)

Internal buzzer timings may also be configured (FW 0x61.33 and later):

Table 9: Internal buzzer timing

Num		Description	Def.
$06	*CFG_TempBuzz1_OK*	Internal buzzer, correct key pressing	$06
$07	*CFG_TempBuzz1_Fail*	Internal buzzer, upon error	$78

For all the parameters described in the table, the configuration value corresponds to the time in multiples of 10 ms that they are to remain enabled:

- With a correct reading for which *CFG_Reader_BuzOK* is enabled, the internal buzzer (*CFG_TempBuzz_OK*) will be triggered.
- With an incorrect reading for which *CFG_Reader_ BuzError* is enabled, the internal buzzer (*CFG_Temp-Buzz_Fail*) will be triggered.

Permitted values:

- 0xFF: Permanently enabled.
- 0x01 .. 0xFE: Timing in **multiples of 10 milliseconds**.
- 0x00: Timing disabled.

Default value: depending on parameter (see table)

8.1.6. Others

Table 10: Other configuration parameters

No.		Description	Def.
$14	*CFG-UI_DisplayTMO*	Maximum time-out of the Display	$0F

Parameter$14: **Display time-out**. Time-out (in cycles) after writing on the Display.

Permitted values:

- 0xFF: legacy value. This is a very conservative Time-out.
- 0x0F: recommended value.
- 0x02: minimum value. Use it if no display is connected.

Default value:

- 0x0F: recommended value.

8.2. IP configuration - TCP and UDP sockets

IP communications via the UDP Socket are configured using a different kind of parameter array and are numbered as follows:

- `$01` - *IP-Client*: Four bytes indicating the address of the *BioMax2 / KBio2-Online* electronics being configured.
- `$02` - *IP-Gateway*: Four bytes, indicating the gateway address of the IP network.
- `$03` - *IP-NetMask*: Four bytes, configuring the submask of the IP network.
- `$04` - *IP-Server*: Four bytes, indicating the server address of the IP network.
- `$05` - *IP-RemoteHost*: Four bytes, indicating the Host address from which we will communicate with the *BioMax2 / KBio2-Online* electronics terminal via a UDP Socket. To communicate via the TCP/IP Socket, this parameter does not have to be specified although it is advisable (increases machine safety).
- `$06` - *IP-Context*: Four bytes, specifying the IP context. This parameter is maintained for compatibility with other Kimaldi products.
- `$07` - *Port-RemoteHost:* Two bytes, encoding four BCD digits, which indicate the Host port with which communication with the *BioMax2 / KBio2-Online* electronics is to be established (Bio-OCX protocol via UDP). Its default value is `5501`. If its value is `0000`[1], the *BioMax2 / KBio2-Online* electronics will automatically respond to the Host port originating the frame.
- `$08` - *SLK-Safety:* if the most significant bit is at 1, only TCP/IP communications from the IP declared as *IP-RemoteHost* are permitted. If the most significant bit is at 0, the TCP/IP communication can be established from any host. For security reasons, it is advisable to filter the Host IP so that the default value of this parameter is `$FF`. The seven least significant bits are reserved for future use.
- `$09` - *DHCP*: If it is "TRUE", the server is responsible for assigning the IP address to the *BioMax2 / KBio2-Online* electronics. If it is "FALSE", the IP address will be permanent and we will set this ourselves.
- `$0A` - *MAC-Address*: The 6 bytes of the MAC Address. This value can be accessed as read only, not write.
- `$0B` - *KSP-Address*: Two bytes that encode the KSP address. The high part is obtained by adding 0x0B to the value encoded by the JP1, JP2 jumpers. The low part is directly the *KSP-Address_Low* byte (parameter `$0F`, next). This value can be accessed as read only, not write.
- `$0C` - Not applicable in *BioMax2 / KBio2-Online* electronics.
- `$0D` - *Description*: This is a string of text of up to 40 characters that enables us to identify the terminal in human language (e.g. "Door Terminal 1").

1. Warning: if the value of the Remote Host Port, Bio-OCX UDP, is 0, events will not be generated via this protocol until a frame is sent from the Host

- `$0E` - *TCP_TimeOut*: This is a byte that encodes the time in seconds between transmission retries. In other words, after the transmission of a frame, the *BioMax2 / KBio2-Online* electronics will wait for Acknowledgement from the Host during *TCP_TimeOut* seconds. After this time, the frame will be resent up to 3 times. After 3 failed retries, the *BioMax2 / KBio2-Online* electronics will close the TCP/IP socket. This same parameter controls the *KeepAlive* frame of the *BioMax2 / KBio2-Online* electronics: every *10 x TCP_TimeOut* seconds, the *BioMax2 / KBio2-Online* electronics will send a frame with the ASCII $06 byte to check that communications with the Host remain valid. If the frame *KeepAlive* is not acknowledged after 3 retries, the Socket will also be closed.

  If *TCP_TimeOut* is 0, no retries are made, *KeepAlive* frames are not generated and the socket can only be closed from the Host.

- `$0F` - *KSP-Address_Low*: This byte encodes Node_ID in a KSP network. It would be equivalent to the 8 bit DIP-Switch of Kimaldi CAN system products.
- `$10` - *CFG_ePHY*: This byte allows for the type of Ethernet connection available for our electronics to be consulted and specified in advance.
- `$1E` - *TCP Port-RemoteHost:* Four BCD digits that indicate the Host port through which TCP communications (Bio-OCX protocol via TCP (TCP Server) or KSP-TCP communications (KSP protocol via TCP (TCP-client)) will be established . Its default value is `0000`, so that the enabled protocol is Bio-OCX (TCP Server). If a different value is encoded (typically, `6601`), TCP-Client will be enabled.
- `$1F` - *KSPPort-RemoteHost:* Four BCD digits that indicate the Host port through which communications with *BioMax2 / KBio2-Online* electronics will be established (KSP protocol via UDP). Its default value is `6001` and it requires a value other than `0000`.

It is important to remember that changes in the IP configuration will not take effect until after the machine is restarted, which can be done by simply sending the ApplyCFG instruction.

8.2.1. ePHY configuration

During the start-up phase and before starting IP communications, the *BioMax2 / KBio2-Online* electronics must negotiate the transmission speed to the subnetwork where it is connected. The process is as follows:

1. Self-negotiation: this allows for connection to 10 BaseT or 100 BaseT, Half-Duplex.
2. Manual negotiation to 100 BaseT, Half-Duplex.
3. Manual negotiation to 10 BaseT, Half-Duplex.
4. Let us assume that there is no available IP connection. PHy is configured to 100 BaseT.

After an ApplyCFG application, the Ethernet connection should be renegotiated, which could last a few seconds in the worst case scenario (where there is no Ethernet connection). The correct configuration of the $10 parameter (*CFG_ePHY*) allows for the electronics start-up time to be optimised.

Table 11: ePHY configuration values

CFG_ePHY	**Ethernet negotiation**
$08	Ethernet connection disabled
$07	Allows all cases (default value)
$06	Self-negotiation, 100 BaseT
$05	Self-negotiation, 10 BaseT
$02	Manual negotiation, 100 BaseT
$01	Manual negotiation, 10 BaseT
+ $10	Add to the vaules above, in order to exclude compatibility to Gigabit Ethernet.

The normal operation of this parameter is as follows:

1. Based on the default value (`$07`), which takes us through the tree negotiation steps: self-negotiation (`$04`), manual negotiation to 100 BaseT (`$02`) and manual negotiation to 10 BaseT (`$01`).
2. After obtaining the negotiation, the value of the *CFG_ePHY* parameter reflects its result (e.g. a value of `$06`), indicating that self-negotiation to 100 BaseT has been successful.
3. If a speed cannot be negotiated, the value `$07` is maintained, which leaves all the alternatives open for the next time the electronics is restarted. Likewise, ePHY is configured to 100 BaseT, as it is the most common speed.

The improvements introduced with this parameter are visible for the following cases:

- 10 BaseT with no self-negotiation capacity: value `$01` can be set manually and initialisation of the electronics will be faster.
- Connect to the Host via the UART: value `$08` can be set manually and the initialisation of the electronics will be faster.

Collateral effects: if the connection speed is to be changed, make sure value `$07` is restored or restart the electronics a couple of times.

In general, the *CFG_ePHY* parameter does not have to be modified and consulting it will indicate the type of Ethernet connection established.

9. Communications Protocol

The communications protocol with the *BioMax2 / KBio2-Online* electronics consists of two aspects:

- Frame format
- Instructions (OpCode + Arguments) and events.

Communications between the *BioMax2 / KBio2-Online* electronics and the Host can be via several communication channels and, in fact, the communications protocol will differ in each case. ActiveX Kimaldi Bio-OCX (*PrKBIOMAXctl*) controls can be used, which encapsulate all the details of the protocol and instructions.

If, however, KSP (see Subsection 6.2.4.) is used or your own serial frames are generated, bear in mind the frame format and instructions detailed in the following sections.

9.1. Bio-OCX RS-232 or TCP/IP frame format

If the serial port, the TCP/IP socket or an external converter (e.g. RS232-WiFi) is used, the frames should be in the following format:

<STX><OPC><NA><ARG><CRC><ETX>

Where:

<STX>: [1 byte] Indicates the start of the frame. It is the ASCII character 0x02.

<OPC>: [2 char] This parameter indicates the meaning of the frame and enables you to interpret the data it contains. It is a byte value expressed by two hexadecimal ASCII characters (ranging from '0' to '9' or from 'A' to 'F'), representing a hexadecimal two-digit value.

<NA>: [4 char:] This parameter indicates the number of bytes encoded in the data field. It is a 16 bit value expressed by four hexadecimal characters representing a hexadecimal four-digit value.

<ARG>: [2 x NA char] This contains the frame information. This field will be made up of <NA> pairs of hexadecimal ASCII characters representing each of the <NA> information bytes.

<CRC>: [2 char] This field contains a byte value resulting from adding all the ASCII values from the frame in module 255 except for <STX>, <CRC> and <ETX>. Expressed by two hexadecimal ASCII characters.

<ETX>: [1 byte] This indicates the end of the frame. It is the ASCII

character 0x03

To speed up data transmission to the serial ports (e.g. for biometric data transmission), the UDP frames have the Extra Data field that encodes the end of the frame:

<OPC><NA><ARG><CRC><ETB>[Length][Data]<CRC>

<ETB>: Extra Data start indicator (*End Transmission Block*, ASCII $17).

[Length]: [4 char:] This parameter indicates the number of bytes that are encoded in the Extra Data data field. It is a 16 bit value expressed by four hexadecimal characters representing a hexadecimal four-digit value. The maximum supported length is 912 .

[Data]: This contains the Extra Data information. This field will be made up by [Length] pairs of hexadecimal ASCII characters representing each of the [Length] information bytes.

<CRC>: [1 byte] This field contains a byte value resulting from adding all the ASCII values of [Length] and [Data] in module 255 (<ETB> is not included). Expressed by two hexadecimal ASCII characters.

9.2. Bio-OCX UDP frame format

If the *BioMax2 / KBio2-Online* electronics 5500 IP port is used, the frames should be in the following format:

<OPC><NA><ARG><CRC>

Where:

<OPC>: [2 char] This parameter indicates the meaning of the frame and enables you to interpret the data it contains. It is a binary byte.

<NA>: [4 char:] This parameter indicates the number of bytes that are encoded in the data field. It is a 16 bit value expressed in binary format.

<ARG>: [2 x NA char] This contains the frame information. This field will be made up of <NO> bytes with information directly encoded in binary format.

<CRC>: [2 char] This field contains a byte value resulting from adding the rest of the bytes of the frame in module 255 except for <CRC>.

To speed up data transmission to the serial ports (e.g. for biometric data transmission), the UDP frames have the Extra Data field that encodes the end of the frame:

```
<OPC><NA><ARG><CRC><ETB>&b[Length][Data]
```

`<ETB>`: Extra Data start indicator (*End Transmission Block*, ASCII `$17`).

'`&b`': Binary format Extra-Data indicator via UDP.

`[Length]`: 2 bytes in binary format. The maximum supported length is 912 .

`[Data]`: String of `[Length]` bytes. This contains the information directly expressed in binary format.

9.3. KSP UDP frame format

If the *BioMax2 / KBio2-Online* electronics 6000 IP port is used, the frames should be in the following format:

```
<AppID><NodeID><KSP_Opc><Len><Lbl><OPC><ARG>
```

Where:

`<AppID>`: [1 byte] Application identifier (see Subsection 6.3.2.)

`<NodeID>`: [1 byte] Node identifier (see Subsection 6.3.2.)

`<KSP_Opc>`: [1 byte] Its value is `$F8` for the data frames.

`<Len>`: [1 byte] Length afterwards. Its value is 2 + NA.

`<Lbl>`: [1 byte] Value that is always different, indicating the frame number ("label").

`<OPC>`: [1 byte] This parameter indicates the meaning of the frame and enables you to interpret the data it contains. It is a binary byte.

`<ARG>`: [NA bytes] This contains the frame information. This field will be made up of <NA> bytes with information directly encoded in binary format.

In some cases, the *BioMax2 / KBio2-Online* electronics will simply transfer the data received from a slave reader. In this case, this data can be encapsulated as Extra Data at the end of the KSP frame:

```
<Frame_KSP><ETB>&b[Length][Data]
```

`<ETB>`: Extra Data start indicator (*End Transmission Block*, ASCII `$17`).

'&b': Binary format Extra-Drive indicator via UDP.

[Length]: 2 bytes in binary format. The maximum supported length is 912 .

[Data]: String of [Length] bytes. This contains the information directly expressed in binary format.

The KSP protocol is quite complex and it is therefore advisable to use the controller supplied by Kimaldi, prKSPdll. The programming interface of this controller is low-level and it must be sent a chain made up by the OpCode and the instruction arguments, [<OPC><ARG>], as detailed in the following sections.

9.4. KSP TCP frame format

If the *BioMax2 / KBio2-Online* electronics 6600 IP port is used, the frames should be in the following format:

<AppID><NodeID><KSP_Opc><Len><Lbl><OPC><ARG>

That is, the frame is identical to KSP-UDP, but this time in ASCII-Hex format. Therefore, <AppID> has a length of two bytes and so on. Given the complexity of the KSP protocol and the TCP-Server functionality, it is recommended to use the BioKSP-OCX controller provided by Kimaldi, or its .NET equivalent.

9.5. Common Instructions for Online Management

We will first present the instructions that work with any electronics equipped with the Bio-OCX protocol: BioMax, *BioMax2 / KBio2-Online* electronics, KBio, Serie-xR Card.

9.5.1. Communications Test

Instruction	
OP	0x00
NA	0x0000
ARG	None.

Response	
OP	0x00
NA	0x0000
ARG	None.

Function: Checks communications with one node.

9.5.2. Reset

Instruction	
OP	0x01
NA	0x0000
ARG	None

Response	
OP	0x01
NA	0x0000
ARG	None

Function: Initialises one node.

9.5.3. Firmware Version

Instruction	
OP	0x02
NA	0x0000
ARG	None

Response	
OP	0x02
NA	0x0002
ARG	Identification code of the firmware version. Byte 1: FW Major (0x61 for *BioMax2*, 0x62 for*KBio2-Online*) Byte 2: FW Minor

Function: returns the firmware version of the node.

9.5.4. GetCrashInfo

Instruction	
OP	0x0E
NA	0x0000
ARG	None

Response	
OP	0x0E
NA	0x0000 or 0x0014
ARG	If 0 arguments, there is no available information. If 20 arguments: Bytes 1..20: information for Kimaldi Electronics.

Function: It reads the information stored in the non-volatile memory after an exception has been generated.

9.5.5. ClearCrashInfo

Instruction	
OP	0x0F
NA	0x0000
ARG	None

Response	

OP	0x0F
NA	0x0000
ARG	None

Function: It deletes the information stored in the non-volatile memory after an exception has been generated.

9.6. Specific BioMax2/KBio2-Online instructions

9.6.1. ReadCFG_Byte

Instruction	
OP	0x0A
NA	0x0001
ARG	Byte 1: configuration parameter number (see section 8.1)

Response	
OP	0x0A
NA	0x0002
ARG	Byte 1: configuration parameter number (see section 8.1) >Byte 2: parameter value

Function: It reads the value of a configuration parameter stored in the non-volatile memory. See chapter 8.: "BioMax2 / KBio2-Online electronics configuration" .

9.6.2. WriteCFG_Byte

Instruction	
OP	0x0B
NA	0x0002
ARG	Byte 1: configuration parameter number (see section 8.1) Byte 2: parameter value

Response	
OP	0x0B
NA	0x0001
ARG	Byte 1: configuration parameter number (see section 8.1)

Function: Writes configuration stored in non-volatile memory. See chapter 8.: "BioMax2 / KBio2-Online electronics configuration" .

9.6.3. ReadCFG_IP

Instruction	
OP	0x0C
NA	0x0001
ARG	Byte 1: IP parameter number (see section 8.2)

Response	
OP	0x0C
NA	> 0x0001
ARG	Byte 1: IP parameter number (see section 8.2) Byte 2. n : parameter value

Function: It reads the value of an IP configuration parameter stored in the non-volatile memory. See chapter 8.: "BioMax2 / KBio2-Online electronics configuration" .

9.6.4. WriteCFG_IP

Instruction	
OP	0x0D
NA	> 0x0001

ARG	Byte 1: IP parameter number (see section 8.2) Byte 2. n : parameter value

Response	
OP	0x0D
NA	0x0001
ARG	Byte 1: IP parameter number (see section 8.2)

Function: It writes an IP configuration parameter, storing it in the non-volatile memory. See chapter 8.: "BioMax2 / KBio2-Online electronics configuration" .

9.6.5. FactoryCFG

Instruction	
OP	0x08
NA	0x0000 or 0x0001
ARG	Byte 1: 0x00: machine parameters 0x01: IP configuration > 0x01: machine parameters If NO = 0, this is equivalent to byte 1 = 0x00 (machine parameters)

Response	
OP	0x08
NA	0x0000
ARG	None

Function: It restores the initial factory configuration, which is made up of two parts (see chapter 8): IP parameters.

9.6.6. ApplyCFG

Instruction	
OP	0x09
NA	0x0000
ARG	None

Response	None (a soft-reset of the entire machine occurs)

Function: It restarts the machine for the new configuration values to be applied.

Event	
OP	0x09
NA	0x0000 or 0x0014
ARG	If 0 arguments, normal start-up. If 20 arguments, automatic restart after an exception: Bytes 1..20: information for Kimaldi Electronics.

If bit 3 of *CFG_ECHO* is enabled (see Subsection 8.1.2.), the *BioMax2 / KBio2-Online* electronics may "return the response" to the ApplyCFG instruction. In reality, this will be the first event after electronics start-up and, therefore, it will only be received if the *CFG_EventCH* parameter is correctly configured and, in the event of communications via UDP, if a remote port other than `0000` is defined.

In general, if this event is enabled, it will be seen when the electronics is started even in the event of a restart due to an exception. In the latter case, information for the Kimaldi Electronic Technical Service will be obtained.

9.6.7. Activate Led/Beeper Time

Instruction	
OP	0x30
NA	0x0002
ARG	Byte 1: 0x00: Green LED 0x01: Red LED 0x02: external beeper 0x03: internal beeper Byte 2: enabling time 0x00 to 0xFE tenths of a second (only from 0x00 to 0x19 for internal beeper)

Response	
OP	0x30
NA	0x0000
ARG	None

Function: Activates one of the LEDs or the beeper for the specified time.

Event	
OP	0x30
NA	0x0002
ARG	Byte 1: 0x00: Green LED 0x01: Red LED 0x02: external beeper 0x03: internal beeper indicating that its timing has ended Byte 2: always 0x00

If bit 5 of *CFG_ECHO* is enabled (see Subsection 8.1.2.), the *BioMax2 / KBio2-Online* electronics can indicate the end of a timing. In this case, an event is generated indicating the number of the relay to have returned to standby status.

9.6.8. Switch Led/Beeper

Instruction	
OP	0x31
NA	0x0002
ARG	Byte 1: 0x00: Green LED 0x01: Red LED 0x02: external beeper 0x03: internal beeper Byte 2: 0x00: off. <>0x00: on

Response	
OP	0x31
NA	0x0000
ARG	None

Function: This switches one of the LEDs or the beeper on or off.

9.6.9. Close Relay

Instruction	
OP	0x40
NA	0x0002
ARG	Byte 1: relay number to be enabled (from 0 to 3 in DB module) Byte 2: enabling time 0x00 to 0xFE tenths of second

Response	
OP	0x40
NA	0x0000
ARG	None

Function: Closes designated relay for the specified time.

Event	
OP	0x40
NA	0x0002
ARG	Byte 1: relay number to end the timing (from 0 to 3 in DB module) Byte 2: always 0x00

If bit 5 of *CFG_ECHO* is enabled (see Subsection 8.1.2.), the *BioMax2 / KBio2-Online* electronics can notify the end of a timing. In this case, an event is generated indicating the number of the relay to have returned to standby status.

9.6.10. Switch Relay

Instruction	
OP	0x41
NA	0x0002
ARG	Byte 1: relay number to be enabled (from 0 to 3 in DB module) Byte 2: 0x00: open. <>0x00: enabled (closed)

Response	
OP	0x41
NA	0x0000
ARG	None

Function: Closes or opens designated relay.

9.6.11. Clear Display

Instruction	
OP	0x10
NA	0x0000
ARG	None

Response	
OP	0x11
NA	0x0000
ARG	None

Function: Deletes the information from the Display

9.6.12. Write Display

Instruction	
OP	0x11
NA	0x0028
ARG	Bytes 1 to 20: Text on the first row Bytes 21 to 40: Text on the second row

Response	
OP	0x11
NA	0x0000
ARG	None

Function: It shows the specified text via the display in 2 rows of 20 characters. At the same time, the display is automatically backlit during the time specified by *CFG_TempBacklit* (see section 8.1).

9.6.13. Backlit Time

Instruction	
OP	0x12
NA	0x0001
ARG	Byte 1: enabling time 0x00 to 0xFE tenths of second

Response	
OP	0x12
NA	0x0000
ARG	None

Function: It enables the display backlighting during the specified time.

9.6.14. Switch Backlit

Instruction	
OP	0x13
NA	0x0001
ARG	Byte 1: 0x00: off. <>0x00: on

Response	
OP	0x13
NA	0x0000
ARG	None

Function: It permanently enables or disabled the Display backlighting.

9.6.15. Write Display, Backlit

Instruction	
OP	0x14
NA	0x002A
ARG	Byte 1: Backlit time (from 0x00 to 0xFE tenths of a sec.) Byte 2: Timing of the internal buzzer (from 0x00 to 0x19 tenths) Bytes 3 to 22: Text of the first row Bytes 23 to 42: Text of the second row

Response	
OP	0x14
NA	0x0000
ARG	None

Function: It shows the specified text via the display in 2 rows of 20 characters. The timing of the backlighting and the buzzer assembled in the *BioMax2 / KBio2-Online* electronics are also specified.

9.6.16. Digital input status

Instruction	
OP	0x60
NA	0x0000
ARG	None

Response	
OP	0x60
NA	0x0001
ARG	Bit 0: Status of DIN 0 Bit 1: Status of DIN 1 Bit 2: Status of DIN 2 Bit 3: Status of DIN 3

Function: It returns a byte, showing the status of the four digital inputs (0 to 3). If the bit is at 0, the input is in open circuit. If the bit is at 1, the input is shorted to GND.

Event	
OP	0x60
NA	0x0001
ARG	Bit 0: Status of DIN 0 Bit 1: Status of DIN 1 Bit 2: Status of DIN 2 Bit 3: Status of DIN 3

If bit 4 of the *CFG_ECHO* is enabled (see Subsection 8.1.2.), the *BioMax2 / KBio2-Online* electronics can notify of the change in the digital input status via an event with the same OpCode.

9.6.17. Multiread filter status

Instruction	
OP	0x61
NA	0x0001
ARG	Byte 1: 0: Status of UART_0 (only if it is configured as serial reader) 1: Status of UART_1 (only if it is configured as serial reader) 2: Status of TTL_0 3: Status of TTL_1

Response	
OP	0x61
NA	0x0001 or >0x0001
ARG	Byte 1: port from which the status is announced Bytes 2 to <NA>: Code of the present card

Function: It allows for the status of the card reader to be questioned to confirm whether there is a card in the field that it is being filtered in multiread mode. If there is no card in the reader field, it will return just one argument byte.

The reader code may be made up of decimal ASCII characters (from '0' to '9') or ASCII-Hex (from '0' to '9' and from 'A' to 'F') characters. It may also include an indicative prefix of the reader to have read the card (see Subsection 8.1.4.).

9.6.18. Card Reading

Event	
OP	0x81
NA	0x0000 (erroneous reading) > 0x0000 (correct reading)
ARG	Bytes 1 to <NA>: Code of the read card

Function: It returns the code read by the card reader. It may be made up of decimal ASCII characters (from '0' to '9') or ASCII-Hex (from '0' to '9' and from 'A' to 'F') characters. It may also include an indicative prefix of the reader to have read the card (see Subsection 8.1.4.).

If the reading was erroneous, the frame is returned with 0 arguments. This will enable the Host to act on the LEDs or the Beeper to inform users of the success of failure of the operation.

If the multiread filter is enabled, a code based on special characters (e.g. '**********' for TTL_0) can be obtained to indicate that the card has been removed from the reader.

9.6.19. Key Pressing

Event	
OP	0x80
NA	0x0001
ARG	Key code:

Function: Returns key code pressed for Online management according to the following table:

Table 12: Key codes

Key pressed	Code assigned
Numerical (0 to 9)	0x30 to 0x39 ('0' to '9')
Green key ("Enter")	0x41 = 'A'
Red key ("Exit")	0x42 = 'B'
F1	0x43 = 'C'
F2	0x44 = 'D'
Enter	0x45 = 'E'
CLR	0x46 = 'F'
Optical barrier	*CFG_OPT_KeyCode* ('A' by default)

Note that the presence of a finger on the optical barrier is also announced via this event (normally via code 0x41 = 'A', configurable).

9.6.20. Send to UART

Instruction	
OP	0x51
NA	0x0001
ARG	Byte 1: 0x00 to 0x02: UART0 to UART2
Extra Data	Frame sent to the RS232 reader from the Host

Response	
OP	0x51
NA	0x0000
ARG	

Function: Encapsulates a frame to be sent from the Host to the RS232 reader via the UART of the *BioMax2 / KBio2-Online* electronics.

9.6.21. Reception from UART

Event	
OP	0x83
NA	0x0001
ARG	Byte 1: 0x02
Extra Data	Frame that sends the RS232 reader to the Host

Function: Encapsulates a frame to be sent to the Host from the RS232 reader via the UART of the *BioMax2 / KBio2-Online* electronics.

9.6.22. Optical Barrier Status

Event	
OP	0x4B
NA	0x01
ARG	Status code of the optical barrier 0x0D: Back to normal 0xED: Interference or blocking of the optical barrier

Function: If the path of the optical barrier is blocked for some time or there is interference from an external light source, this is communicated through OpCode 0x4B. When the situation returns to normal, the same OpCode with Arg = 0x0D indicates this. See Subsection 5.3.2. and the explanation of the *CFG-UI_OPT_T_WATCHDOG* byte for further details. chapter 8

9.6.23. Enable optical barrier

Instruction	
OP	0x4A
NA	0x0000
ARG	None

Response	
OP	0x4A
NA	0x0001
ARG	Status of the optical barrier

Function: Enables the detection LED of the optical barrier This command can be used to allow for biometric identification simply by placing a finger on the sensor.

The optical barrier is self-managed by default; therefore this instruction is useful to:

- Activate biometric identification after reading a proximity card.
- Force the fast recovery of the optical barrier after a biometric identification (the normal recovery time is set using the *CFG-UI_OPT_T_GUARD* Byte of the configuration, see Subsection 8.1.3. or is indefinitely suspended using the Disable optical barrier instruction).

9.6.24. Disable optical barrier

Instruction	
OP	0x4C
NA	0x0000
ARG	None

Response	
OP	0x4C
NA	0x0001
ARG	Status of the optical barrier

Function: Disables the detection LED of the optical barrier There are several reasons for using this instruction:

- When the scanning of a finger print has started to prevent new warnings.
- In the case of 1:1 identification, the user is expected to start identification using the proximity reader.
- In the case of presence control, the user is expected to press F1 or F2 prior to biometric identification.

9.6.25. Encapsulate GCR

Instruction	
OP	0xBD
NA	0x0003 or above
ARG	See section 10.1

Response	
OP	0xBD
NA	0x0005 or above
ARG	See section 10.1

Function: it encapsulates instruction frames in GCR format (Kreta-Classic format). These GCR instructions allow you to address the DB and UI modules separately and to implement the different FingerVein sensor identification functions (see chapter 10.: "Biometric identification" for further details).

9.7. Error Codes

BioMax2 / KBio2-Online electronics can also respond to any of the instructions it is sent with one of the following codes:

- OpCode *0xFD: Frame Delay* (resend later).
- OpCode *0xFE: Frame Error* (badly-composed frame or argument out of range).
- OpCode *0xFF: Instruction Error* (invalid OpCode).

10. Biometric identification

The *BiomaxPlus* communications protocol provides a set of high-level instructions to simply handle the biometric sensor.

The following instructions will be used as arguments in the Encapsulate GCR command indicated in Subsection 9.6.25.

These GCR instructions are initially intended to work with a main FIM and the *BiomaxPlus-DB* auxiliary board. However, they also work, though partially, for one or two FIM modules, even when no *BiomaxPlus-DB* is present. In order to achieve that, Parameter$1D (associated to commands `'71x'`) and optionally Parameter$1E (associated to commands `'72x'`) must be properly configured (that is, with a value of `$04` or `$05`).

If the available FIM module does not support Auto-On detection, the following instructions may still be used, but Parameter$03 must be set to `$80` or similar.

10.1. Basic instructions

10.1.1. Retrieve sensor version

Instruction	'714' / '724'	
Description	'714'	Instruction code. Intermediate digit addresses Main FIM ('1') or auxiliary ('2').
		No argument

Response	'814tttaaaabbbb' / '824tttaaaabbbb'	
Description	'814'	Response code. Intermediate digit addresses Main FIM ('1') or auxiliary ('2').
	'ttt'	Sensor type. Possible values: FIM - FIM-type biometric sensor.
	'aaaa'	Information on sensor part number. - Usual value: '5360'
	'bbbb'	Information on FW version. - Example: '0114' (versión 1.14)

Function: It allows for the type of biometric sensor installed in the terminal and its firmware version to be retrieved.

10.2. Capture of biometric information

10.2.1. Capture of biometric information

Instruction	'713' / '723'	
Description	'713'	Instruction code. Intermediate digit addresses Main FIM ('1') or auxiliary ('2').
	nn	Optional argument specifies the number of templates to scan ('01' or '02'). If no argument, one template is scanned when in FIM30 emulation mode, and two scans when in FIM20 emulation mode.

Response	'813rr' - ExtraData: 'hhh...hhh' / '823rr' - ExtraData: 'hhh...hhh'	
Description	'813'	Response code. Intermediate digit addresses Main FIM ('1') or auxiliary ('2').
	'rr'	Operation result. Possible values: 11 - Operation completed correctly.
	'hhh...hhh'	The captured template is sent on the Extra-Data field of the response frame. Template length depends on emulation mode (FIM20 vs. FIM30).

Función: It captures templates from user's fingerprint scan operation.

10.3. Sensor database management

10.3.1. Add user

Instruction	‘564nnnnnnnnn0000000000t’ - ExtraData: “hhh...hhh”	
Description	‘564’	Instruction code.
	‘nnnnnnnnnn’	User identifier.
	‘0000000000’	Reserved.
	‘t’	Template destination. Possible values: ‘1’, ‘2’, ‘3’ - Send template to Main FIM (‘1’), to Auxiliary FIM (‘2’) or to both (‘3’).
	‘hhh...hhh’	Biometric information of the user to be registered. Only used where ‘t=1’, ‘t=2’, ‘t=3’. This information travels through the Extra-data field of the frame and has been previously captured through ‘713’ command.

Response	‘664rr’	
Description	‘664’	Response code
	‘rr’	Operation result. Possible values: 11 - Operation completed correctly.

Function: It allows for a user's biometric information to be registered in the sensor’s database, as well as in *BiomaxPlus-DB* if installed.

10.3.2. Request number of users

Instruction	‘715’ / ‘725’	
Description	‘715’	Instruction code. Intermediate digit addresses Main FIM (‘1’) or auxiliary (‘2’).
		No argument

Response	‘815nnnn’ / ‘825nnnn’	
Description	‘815’	Response code. Intermediate digit addresses Main FIM (‘1’) or auxiliary (‘2’).
	‘nnnn’	Number of users stored in the sensor’s database.

Function: it returns the number of users stored in the sensor’s database.

10.3.3. Delete user

Instruction	'562nnnnnnnnn'	
Description	'562'	Instruction code.
	'nnnnnnnnn'	Identifier of the user to be deleted.

Response	'662rr'	
Description	'662'	Response code
	'rr'	Operation result. Possible values: 11 - Operation completed correctly.

Function: it allows for the biometric data of a user to be deleted from the sensor's database, as well as from *BiomaxPlus-DB*. If two FIM modules are connected, information is deleted from both, with no other choices available.

10.3.4. Delete all users

Instruction	'561'	
Description	'561'	Instruction code.
		None

Response	'661rr'	
Description	'661'	Response code
	'rr'	Operation result. 11 - Operation completed correctly.

Function: it allows for the biometric data of all users to be deleted from the sensor's database, as well as from *BiomaxPlus-DB*. If two FIM modules are connected, information is deleted from both, with no other choices available.

10.4. User identification

10.4.1. Instant Matching

Instruction	'71700' - ExtraData: "hhh...hhh" / '72700' - ExtraData: "hhh...hhh"	
Description	'717'	Instruction code. Intermediate digit addresses Main FIM ('1') or auxiliary ('2').
	'hhh...hhh'	Biometric information to use in the comparison.

Response	'817rrqq' / '827rrqq'	
Description	'817'	Response code. Intermediate digit addresses Main FIM ('1') or auxiliary ('2').
	'rr'	Operation result. Possible values: 00 - Incorrect matching 11 - Correct matching.
	'qq'	Reserved for future use. - Default value: 'FF'

Function: the user's finger is scanned, and the retrieved template is compared against the information provided by the same instruction.

10.4.2. 1:N Identification

Instruction	'716' / '726'	
Description	'716'	Instruction code. Intermediate digit addresses Main FIM ('1') or auxiliary ('2').
		No argument

Response	'816rrqqnnnnnnnnnn' / '826rrqqnnnnnnnnnn'	
Description	'816'	Response code. Intermediate digit addresses Main FIM ('1') or auxiliary ('2').
	'rr'	Operation result. Possible values: 11 - Correct identification
	'qq'	Reserved for future use. - Default value: 'FF'
	'nnnnnnnnnn'	User's identifier (FP-ID)

Function: it allows for a user whose biometric information is stored in the sensor's database to be identified. That is, the user's scanned fingerprint is compared against all templates in the FIM module's database. The matching template has an identifier (FP-ID) that is returned to the Host.

10.4.3. 1:1 Identification

Instruction	'716nnnnnnnnn' / '726nnnnnnnnn'	
Description	'716'	Instruction code. Intermediate digit addresses Main FIM ('1') or auxiliary ('2').
	'nnnnnnnnn'	User's identifier (FP-ID)

Response	'816rrqq' / '826rrq'	
Description	'816'	Response code. Intermediate digit addresses Main FIM ('1') or auxiliary ('2').
	'rr'	Operation result. Possible values: 11 - Correct identification
	'qq'	Reserved for future use. - Default value: 'FF'

Function: it allows for a user whose biometric information is stored in the sensor's database to be identified. In 1:1 identification, the user's scanned fingerprint is compared against one single template, that corresponds to the provided FP-ID. The matching result (true or false) is returned.

Appendix A. Application Notes

N.B. Appendix A.1 and Appendix A.2 not translated to English yet.

A.1. BioMax2-FP + Mag + Mifare

Vamos a ver a continuación una posible configuración bastante completa basada en la electrónica *BioMax2*:

- Electrónica *BioMax2*, con conectividad IP al Host
- Módulo de identificación biométrico - FIM)
- Lector de banda magnética
- Lector Mifare, Kimaldi KRD13M

En la aplicación que vamos a detallar podría corresponder al acceso principal de un club deportivo. Cada socio del club tiene una tarjeta de acceso, que históricamente era de banda magnética pero que actualmente se está migrando a Mifare. Para evitar suplantaciones, la identificación es biométrica.

Por tanto, el usuario puede identificarse directamente (1:N) o bien presentando previamente su tarjeta, ya sea la antigua tarjeta de banda magnética, o bien la nueva tarjeta Mifare, que codifica el número de socio en su bloque número 1.

La electrónica *BioMax2* controla un torno de doble sentido, de forma que el acceso al recinto se completa con un display, mientras que la salida debe realizarse presionando un pulsador.

A.1.1. Hardware connections

La electrónica *BioMax2* va integrada dentro de un torno, y usamos la cueva biométrica Kimaldi para integración del sensor biométrico. Conexionado (See chapter 4.: "Installation"):

- Fuente de alimentación de 5 Volt, 2 Amp, para todo el conjunto.
  - 5Vcc a Pin 1, conector J3
  - GND a Pin 2, conector J3
- Lector de Banda Magnética: suponemos un lector de pista 2, con salida TTL
  - Conectar directamente a J9 (TTL_0)
- Lector Mifare, Kimaldi KRD13M
  - 5Vcc desde Pin 1, conector J3. En KRD13M: pin 2 del bornero
  - GND desde Pin 2, conector J3. En KRD13M: pin 1 del bornero
  - Rx UART0 desde Pin 7, J3, conecta a Tx del KRD13M, pin 4 del bornero
  - Tx UART0 desde Pin 8, J3, conecta a Rx del KRD13M, pin 3 del bornero
- Módulo biométrico - FIM
  - Conectar directamente a J7 (UART1)
- Barrera óptica
  - Conectar directamente a J8.
- Relés
  - Conectar según convenga (conector J2). Contactos normalmente abiertos.
- Pulsador
  - Conectar entre pin 1 (DIN_0) y pin 5 (GND) del conector J4
- Display
  - Conectar directamente a J10
  - Ajuste de contraste a través de potenciómetro, P2
- Cable Ethernet
  - Conectar directamente a J5

A.1.2. Configuration of the terminal device

En primer lugar, hay que establecer comunicación con el equipo. Esto se consigue a través del Kimaldi Localisation Service. Existe un programa Demo que nos puede servir, una vez instalado el paquete OCX-DLL.

A continuación, contactaremos con el equipo a través del protocolo Bio-OCX UDP o bien de KSP-UDP. En cualquier caso, los parámetros a configurar son los siguientes:

Table 13: Ejemplo de configuración de lectores

Número parám.	Nombre del parámetro	Valor	Detalles
$1A	*CFG_PuertoSerieHost*	$40	KRD13M, a 9600 baud
$1B	*CFG_PuertoSerieLector*	$8A	FIM a 38400 baud
$08	*CFG_Reader_ Enable*	$55	Activados lectores: - UART0: prefijo '$' - TTL0: prefijo '*'
$09	*CFG_Reader_ BuzOK*	$05	Beeper BioMax2
$0A	*CFG_Reader_ BuzError*	$05	Beeper BioMax2

Cada uno de los valores anteriores se escriben a través de:

- Instrucción: WriteCFG_Byte
- Argumentos: `<Número de parámetro> <Valor del parámetro>`

Aplicar la configuración (ApplyCFG) antes de proseguir.

A.1.3. Configuration of KRD13M reader

Podemos configurar el lector KRD13M para que lea automáticamente el bloque 1 de la tarjeta Mifare, recorte la lectura a 10 bytes y los mande por el puerto serie:

Table 14: Ejemplo de configuración de KRD13M

Trama a mandar	Explicación
$0A060001000100091B	Leeremos el bloque 1, posiciones de 0 a 9, con la clave 1 como Key A
$200701FFFFFFFFFFFF22	Escribimos la clave 1 en EEPROM ($FFFFFFFFFFFF en el ejemplo)
$040600056402**02**0AFF7E	Configuramos el byte 4 al modo automático $02

Cada una de las instrucciones anteriores se mandarán al lector de la forma siguiente:

- Instrucción: Enviar a UART
- Argumento: Byte 1 = $00 (es decir, transmisión a UART0)
- Extra Data:
  - A través de Bio-OCX: '>>0A060001000100091B'
  - La trama real que se manda en Extra Data es, en Hexadecimal y a partir de <ETB> (incluido):

1726620009**0A060001000100091B**

A.1.4. 1:N Identification

Si el usuario coloca directamente el dedo sobre el sensor biométrico, iniciaremos una identificación 1:N en el Host. La secuencia de instrucciones será como sigue:

1. Recibimos el suceso Pulsación de tecla, con argumento de valor 'A' (en general, el valor configurado en *CFG_OPT_KeyCode*).
2. Suspendemos el funcionamiento de la barrera óptica, para impedir que genere nuevos avisos durante el proceso de identificación: Desactivar barrera óptica.
3. Generamos el mensaje oportuno a través del Display: WriteDisplay.
4. Escaneamos huella:
   - Instrucción: Enviar a UART
   - Argumento: Byte 1 = $01 (es decir, transmisión a UART1)
   - Extra Data:
     - A través de Bio-OCX: 'FIM16'
     - La trama real que se manda en Extra Data es, en Hexadecimal y a partir de <ETB> (incluido):

17266200197E0000001600000000000000000000000000000000000016

6. Al cabo de unos segundos, recibimos la información de la huella:
   - Suceso: Recepción desde UART
   - Argumento: Byte 1 = $01 (es decir, recepción desde UART1)
   - Extra Data:
     - A través de Bio-OCX: recibimos una cadena de datos.
     - La trama real que se recibe en Extra Data es, en Hexadecimal y a partir de <ETB> (incluido):

17266201B17E0000001600000001000000000000019400000000000000AC
00000003<400 Bytes de datos de la huella>

7. A partir de la trama anterior, extraemos la huella, la convertimos a formato eNBSP e instruimos al Host para que haga una identificación 1:N. Obtendremos un User-ID, en función del cual decidimos activar un relé, sacar un mensaje por el Display, etc.
8. Restablecemos el funcionamiento de la barrera óptica: Activar barrera óptica

A.1.5. 1:1 Identification

En este caso, la identificación se inicia presentando la tarjeta de usuario:

1. Recibimos un suceso Lectura de Tarjeta. Tal como está configurado el equipo, podemos distinguir la procedencia de la lectura: prefijo '$' para lector KRD13M, prefijo '*' para el lector de banda magnética.
2. Accedemos a la base de datos eNBSP para recuperar la huella del usuario que se ha identificado. Deberemos convertir la huella a formato FIM-HV (404 bytes). Si la huella está disponible, iniciamos el proceso de verificación.
4. Suspendemos el funcionamiento de la barrera óptica, para impedir que genere nuevos avisos durante el proceso de identificación: Desactivar barrera óptica.
5. Generamos el mensaje oportuno a través del Display: WriteDisplay.
6. Verificamos huella ("*Instant Matching*"):
   - Instrucción: Enviar a UART
   - Argumento: Byte 1 = `$01` (es decir, transmisión a UART1)
   - Extra Data:
     - A través de Bio-OCX: `'FIM15'`
     - La trama real que se manda en Extra Data es, en Hexadecimal y a partir de `<ETB>` (incluido):

```
17266201B17E00000015000000000000000000000194000000000000000AA
          00000003<400 Bytes de datos de la huella>
```

8. Los datos mandados se comparan con la huella, que se escanea a continuación. Al cabo de unos segundos, recibimos el resultado de la operación:
   - Suceso: Recepción desde UART
   - Argumento: Byte 1 = `$01` (es decir, recepción desde UART1)

• Extra Data:
- A través de Bio-OCX: recibimos una cadena de datos.
- La trama real que se recibe en Extra Data es, en Hexadecimal y a partir de `<ETB>` (incluido):

```
17266200197E00000015 00000001 00000000000000000000000000000016
```

- `Param1` es `0x01` en caso positivo, `0x02` en caso negativo, `0x07` si se excedió el tiempo.

9. En función del resultado anterior, decidimos activar un relé, sacar un mensaje por el Display, etc.
10. Restablecemos el funcionamiento de la barrera óptica: Activar barrera óptica

A.2. BioMax2-FP Prox + KBio Online RS-232

En el siguiente capítulo, nos vamos a plantear la siguiente situación:

- Un terminal *BioMax2-FP* Prox, con conectividad IP al Host.
- Un terminal KBio Online, conectado al Host a través de la *BioMax2*.

En esta aplicación, el punto de acceso requiere identificación biométrica con proximidad, tanto para entrar como para salir. No obstante, la entrada ofrece una interficie de usuario más completa (display y teclado), mientras que la salida interacciona con el usuario simplemente a base de LEDs y señalización acústica (beeper).

Si bien la electrónica *BioMax2* permite la conexión de hasta dos módulos de identificación biométrica y dos lectores Clock&Data, en algunos casos es más cómodo instalar una electrónica auxiliar para simplificar el cableado de la instalación.

Ante una instalación de este tipo, será suficiente con que una de las dos electrónicas esté dotada de conectividad IP.

A.2.1. Hardware connections

Asumimos en este caso que los terminales individuales están ya correctamente ensamblados en sus respectivas cajas. Para conectar los terminales a través de RS-232:

- Fuente de alimentación de 5 Volt, 2 Amp, para todo el conjunto.
  - 5Vcc a Pin 1, conector J3
  - GND a Pin 2, conector J3
- Cable RS-232 DB9 Macho, para conectar a KBio:
  - GND desde Pin 2, conector J3, al pin 5 del DB9-Macho
  - Rx UART0 desde Pin 7, J3, al pin 2 del DB9-Macho
  - Tx UART0 desde Pin 8, J3, al pin 3 del DB9-Macho
- Conectar el terminal KBio Online Prox, con su fuente de alimentación de 12 Volt y su cable RS-232 DB9-Hembra.

A.2.2. Device configuration

En primer lugar, hay que establecer comunicación con el equipo. Esto se consigue a través del Kimaldi Localisation Service. Existe un programa Demo que nos puede servir, una vez instalado el paquete OCX-DLL.

A continuación, contactaremos con el equipo a través del protocolo Bio-OCX UDP o bien de KSP-UDP. En cualquier caso, los parámetros a configurar son los siguientes:

Table 15: Ejemplo de configuración con KBio auxiliar

Número parám.	Nombre del parámetro	Valor	Detalles
$1A	*CFG_PuertoSerieHost*	$99	KBio, a 19200 baud
$08	*CFG_Reader_ Enable*	$04	Activados lectores: - TTL0: sin prefijo
$09	*CFG_Reader_ BuzOK*	$04	Beeper *BioMax2*
$0A	*CFG_Reader_ BuzError*	$04	Beeper *BioMax2*

Cada uno de los valores anteriores se escriben a través de:

- Instrucción: WriteCFG_Byte
- Argumentos: `<Número de parámetro> <Valor del parámetro>`

Aplicar la configuración (ApplyCFG) antes de proseguir.

A.2.3. Communication through Bio-OCX UDP

A través del controlador Active-X, la comunicación con el terminal *BioMax2* es inmediata. Por el contrario, para comunicarse con el terminal KBio es necesario componer la trama serie para transmitirla como "Extra Data" de la instrucción Enviar a UART.

Así por ejemplo, una instrucción Test de comunicaciones dirigida a la KBio:

- Instrucción: Enviar a UART
- Argumento: Byte 1 = `$00` (es decir, transmisión a UART0)
- Extra Data:
  - A través de Bio-OCX: `'>>00000020'`
  - La trama real que se manda en Extra Data es, en Hexadecimal y a partir de `<ETB>` (incluido):

`1726620004`**`00000020`**

- La trama entera UDP:

`5100010087172662000400000020`

La Response del terminal KBio nos llegará de la siguiente forma:

- Suceso: Recepción desde UART
- Argumento: Byte 1 = `$00` (es decir, recepción desde UART0)
- Extra Data:
  - A través de Bio-OCX: recibimos una cadena de datos.
  - La trama real que se recibe en Extra Data es, en Hexadecimal y a partir de `<ETB>` (incluido):

`1726620004`**`00000020`**

- La trama entera UDP:

`830001008C172662000400000020`

La comunicación con el FIM del terminal KBio se realizará mediante la instrucción Enviar a FIM encapsulada dentro de Extra Data. Así, por ejemplo, podemos interrogarlo en referencia a su versión de FW:

- Instrucción: Enviar a UART
- Argumento: Byte 1 = `$00` (es decir, transmisión a UART0)

• Extra Data:
- A través de Bio-OCX:

'>>5000197E00000004000000000000000000000000000000000000004B3'

- La trama real que se manda en Extra Data es, en Hexadecimal y a partir de <ETB> (incluido):

172662001D**5000197E00000004000000000000000000000000000000000000004B3**

• La trama entera UDP:

5100010087172662001D**500019**
7E00000004000000000000000000000000000000000000004B3

A.2.4. Communication through KSP UDP

La electrónica *BioMax2* ofrece una pasarela KSP a RS-232, de modo que el terminal KBio conectado a la UART0 se comporta, a nivel de funcionalidad, como un nodo KSP-UDP (no es visible, no obstante, a través del Kimaldi Localisation Service).

Para trabajar a través de KSP-UDP, solamente hace falta tener definida una dirección KSP distinta para cada terminal. La *BioMax2* ofrece un valor de *Node_ID* por defecto, que se puede modificar a través del parámetro de configuración IP *KSP-Address_Low*. La parte alta de la dirección KSP (*Application_ID*) se deriva de los *jumpers* JP1 y JP2 (See chapter 4.1):

- JP1 es el bit de menor peso, JP2 el de mayor, dando lugar a valores entre 0 y 3 (0: *jumper* abierto; 1: *jumper* cerrado).
- *Application_ID* de la BioMax2: 0x0B + *jumpers* (es decir, un valor entre 11 y 14).
- *Application_ID* de la KBio: 0x4B + *jumpers* (es decir, un valor entre 75 y 78).

Por tanto, la comunicación con ambos terminales se realiza mediante las mismas instrucciones, con la bien entendida salvedad de que la dirección KSP va a ser distinta según el nodo al que nos dirijamos:

- Instrucción Test de comunicaciones a BioMax2 en formato KSP-UDP, trama entera:

0D01F8027A**00**

- Instrucción Test de comunicaciones a KBio en formato KSP-UDP, trama entera:

4D01F8027B**00**

Observemos que en ambos casos mandamos el OpCode 0x00, una vez al dispositivo 0x0D.01 y la siguiente al dispositivo 0x4D.01. La etiqueta de la trama deberá ser distinta cada vez (0x7A y 0x7B, respectivamente).

Para finalizar el ejemplo, y en comparación con el apartado anterior, la comunicación con el módulo FIM conectado al terminal KBio se realizaría de la siguiente forma:

- Instrucción: Enviar a UART
- Argumento: Byte 1 = `$00` (es decir, transmisión a UART0)
- Extra Data:
  - La trama real que se manda en Extra Data es, en Hexadecimal y a partir de `<ETB>` (incluido):

```
17266200197E00000004000000000000000000000000000000000004
```

- Instrucción `CMD_GET_FIRMWARE_VERSION2` a KBio en formato KSP-UDP, trama entera:

```
4D01F8034251001726620019
7E00000004000000000000000000000000000000000004
```

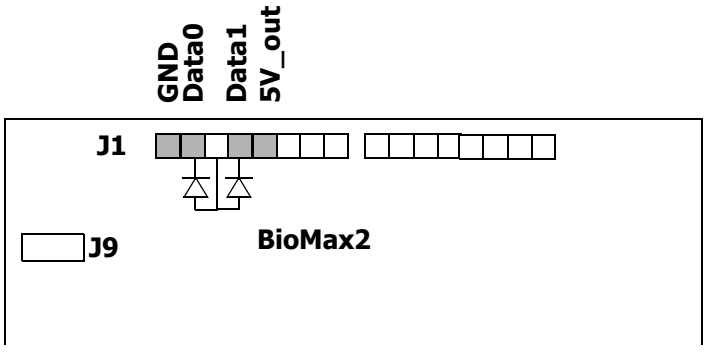
A.3. Wiegand Reception

Starting on FW version x61.41.60 on *BioMax2*, Wiegand reception is also possible. In order to achieve that, some HW and FW adjustments are required, as explained below.

A.3.1. Hardware configuration

In order to receive a Wiegand frame, we must use terminal block J1 on *BioMax2*. Please note that D1, D0 inputs overlap CLS and DAT respectively, formerly related to TTL_1. Additionally, two signal diodes are required between both pins and the central one, formerly TTL_1-CLK.

Figure 1: Input arrangement for Wiegand, BioMax2



A.3.2. Firmware Configuration

The Wiegand input is, effectively, the TTL_1 input with some tricky adjustments. For that reason, once Parameter$0F is configured as stated below, the incoming data will be treated by *BioMax2* just as if it were TTL_1. Implications will be seen further ahead.

Port activation

As said, Wiegand Input will be enabled through Parameter$0F, *CFG_Wiegand*, as explained in Subsection 8.1.4.

Following the case depicted in Figure 1:, let *CFG_Wiegand* take value `$21` (i.e, 26-bit Wiegand through TTL_1 port).

Configuration of the Main Reader

We shall use Parameter$08, *CFG_Reader_Enable*, to issue a Card Reading event upon reception of a Wiegand frame. In a simple implementation, value `$08` on that parameter will enable Wiegand reception through TTL_1.

A.3.3. Wiegand format explanation

Wiegand-26 bits format

The Wiegand-26 bit format allows a user-ID which contains 6 ASCII-Hex characters. That is to say, user-ID codes numbered between `"0000000001"` and `"0000FFFFFE"` will be admitted.

Data format is as follows:

- Bit 0: first parity bit, "even" type. Its value is such, that the 13 first bits together (bit 0 included) contain an even number of ones (`'1'`).
- Bits 1 to 24: data corresponding to the 6 least significant characters of the user-ID, transformed to binary.
- Bit 25: last parity bit, "odd" type. Its value is such, that the 13 last bits together (bit 25 included) contain an odd number of ones (`'1'`).

Wiegand-34 bits format

There is no standard format for 34 bits, however we are keeping the same criteria as above, for an 8 ASCII-Hex character user-ID. Therefore, user-ID codes in the range between `"0000000001"` and `"00FFFFFFFE"` will be admitted.

Data format is as follows:

- Bit 0: first parity bit, "even" type. Its value is such, that the 17 first bits together (bit 0 included) contain an even number of ones (`'1'`).
- Bits 1 to 32: data corresponding to the 8 least significant characters of the user-ID, transformed to binary.

- Bit 33: last parity bit, “odd” type. Its value is such, that the 17 last bits together (bit 33 included) contain an odd number of ones (`'1'`).

Free format

A Card Reading event is issued, directly transferring all bits received through the Wiegand port, with no format processing at all. For that reason, whatever parity bits that might exist are also included in the Card Reading data.

No parity check is performed, so caution must be taken if the incoming data is to be used for access control or other applications where security is required: *data integrity is not guaranteed in Free Format mode*.

A.3.4. Configuration example

The simplest case consists of enabling a Wiegand-26 bit reader through TTL_1:

- Enable Wiegand-26 bit input on TTL_1 port: Parameter$0F to `$21`
- Enable TTL_1 for Card Reading, no prefix: Parameter$08 to `$08`
- Enable beeper if correct reading: Parameter$09 to `$88`
- Enable beeper if incorrect reading: Parameter$0A to `$88`

Appendix B. Troubleshooting

1. The BioMax2 / KBio2-Online electronics does not open the session via UDP, although it is detected via SLK

Using the Kimaldi Localisation Service, select the terminal in question and edit the network configuration. Make sure that:

- The terminal's IP is properly assigned (there are no duplicities, you obtain response to 'ping'). If you are in a network with dynamic IP assignment, set the *'DHCP'* field of the IP configuration to 'TRUE' (`'01'`).
- The submask and gateway are correctly configured.
- The server's IP is only necessary if DHCP is enabled.
- The Host to contact the *BioMax2 / KBio2-Online* electronics is displayed in the *Remote Host*.field
- The Host contacts the *BioMax2 / KBio2-Online* electronics via its 5501 port in the case of Bio-OCX UDP and via its 6001 port in the case of UDP-KSP.
- If Bio-OCX UDP is used to contact several *BioMax2 / KBio2-Online* electronics at the same time, the Host will require a different port for each one. In this case, configure the Host port in the *Port-RemoteHost* field. Value 0 configures the suitable port automatically with the limitation that it does not generate events until the Host has started communications.

If configuration parameters are to be modified, remember to apply them (ApplyCFG) before assessing their effectiveness. If the DHCP is enabled, IP initialisation may take several seconds.

2. The FIM module does not recognize the user's fingerprint.

Biometric devices require some caution:

- When the user identifies, the fingertip skin must be clean, not very dry and not very humid. The finger tip must be positioned straight at the terminal's front panel, so that the center of the fingertip is centered on the sensor.
- Extreme climate conditions (summer or winter) may vary hydrating conditions on our skin.
- Those people involved in intense manual work (manipulation of water, detergents, paints, ...) should consider the use of gloves to protect their fingertip skin.

2. No warning is received when placing a finger on the biometric sensor

It the optical barrier has stopped working, an Optical Barrier Status event will have been generated at some moment to indicate the error status. The most frequent causes of this fault are:

- The finger is left on the biometric sensor for too long. Key Pressing events are generated until a Watchdog appears and equipment switches to error status. It returns to normal when the finger has been removed.
- There is too much ambient light on the biometric sensor. This light stops the optical barrier from working correctly and probably also stops the FingerVein sensor from making a correct identification. We advise you to protect the *BioMax2 / KBio2-Online* terminal against excessive light, especially sunlight.
- Dirt on the optical barrier. Clean both lenses of the optical barrier with a wet cloth.

Even if we cannot solve the problem, an alternative way exists: the green key (for example) could be used to trigger fingerprint identification as well.

4. BioMax2 / KBio2-Online electronics does not recognise the proximity card

Ensure the following:

- The reader is correctly connected to the *BioMax2 / KBio2-Online* electronics, either via the Clock&Data or the RS-232. See chapter 4.1.: "Connector signals, BioMax2" , chapter 4.4.: "Connector signals, KBio2-Online" .
- In the case of UART0, make sure that no two readers are simultaneously connected to the same port, one through J14 connector and the other through terminal block J3.
- In the case of UART1, make sure that no two readers are simultaneously connected to the same port, one through J7 connector and the other through terminal block J1.
- In the case of UART0 / UART1, check that their configuration parameters are correct (see Subsection 8.1.1.). In particular, ensure that the UART and reader transmission speeds coincide.
- In any event (RS-232 or Clock&Data), check that the reader configuration parameters have the ports required enabled (see Subsection 8.1.4.)

5. BioMax2 / KBio2-Online electronics does not generate events

To generate an event, *BioMax2 / KBio2-Online* electronics must know several things:

- Via which protocol to generate the event (Bio-OCX UART0, Bio-OCX IP, KSP-IP): it is always generated depending on the last communication received from

the Host. After a reset, and until no communication is received from the Host, this decision is based on the *CFG_EventCH* parameter (see Subsection 8.1.2.).

- In the case of IP communications, the Host IP address to which the frame is to be sent must be determined. This is the *IP-RemoteHost* parameter indicated in section 8.2.
- In the case of Bio-OCX UDP, the Host IP port must also be known. This will be port 5501 by default.
- Nevertheless, if several *BioMax2 / KBio2-Online* electronics are being routed, these should be connected to different IP ports in the Host. If they are assigned automatically by the Host, simply specify a value of `0000` in the *Port-RemoteHost* parameter (section 8.2). In this case, *BioMax2 / KBio2-Online* electronics cannot determine the destination port until it receives a frame from the Host and, therefore, no events will be received. This could be a limitation when faced with an unexpected power cut, as when the power supply is restored, *BioMax2 / KBio2-Online* electronics will stop generating events until it receives an instruction from the Host. This limitation can be solved by using the KSP-UDP protocol, as the event is then transmitted via the Host's 6001 port from any KSP node.
- In the case of Bio-OCX TCP, events cannot be generated until the Host has opened the TCP/IP Socket. It is not possible to configure the *CFG_EventCH* parameter (see Subsection 8.1.2.) so that events are spontaneously generated via TCP/IP after a reset.

6. BioMax2 / KBio2-Online electronics communicates very slowly

There are three possible causes:

- The display is not connected or does not respond correctly. In this case, *BioMax2 / KBio2-Online* electronics is waiting until the end of Display Time-out (*CFG-UI_DisplayTMO,* see Subsection 8.1.6.). If the display is not connected, the time-out value may be lowered: 0x0F is an advisable value.
- A computer is trying to communicate with *BioMax2 / KBio2-Online* electronics via TCP/IP. If the IP filter is not enabled (see *SLK-Safety* in section 8.2), the *BioMax2 / KBio2-Online* electronics will generate rejection frames to each request. Enable the Remote Host IP filter to improve performance and increase safety.

Many configuration parameters are being written from the application. Each write requires an EEPROM recording time, which slows down the response times. It is not advisable to write frequently on the EEPROM.

7. No Ethernet communication can be established

If it is not possible to communicate with *BioMax2 / KBio2-Online* via Ethernet, try the following:

- Check the status of the red ETH_LINK LED, next to the Ethernet connector
- If the LED is off, physical link has been established. Then, make sure that there is no duplicate IP address, and that IP configuration is correct. It should be possible to contact the electronics board by means of the KSearch application, at least when the Broadcast response is selected.

If the ETH_LINK LED is on, the physical link could not be achieved. Then:

- Try replacing the Ethernet cable, or check the Ethernet switch status.
- If *BioMax2 / KBio2-Online* was connected at 100 Base-T and then we try a 10 Base-T connection (or viceversa), try with a hard reset of the electronics.
- If the LAN is NOT Gigabit Ethernet compatible, we may restrict negotiation options by activating bit 4 (add `$10`) on parameter *CFG_ePHY* (See Table 11 in Subsection 8.2.1.).
- We may choose to reconfigure parameter *CFG_ePHY* to value `$07`, so that a full re-negotiation starts with the Ethernet switch (after hard-reset).

List of Tables

Revision Control

Nº Versión	Fecha	Description
Version 2.10	31 october 2012	FW 0x61.43.63 (BioMax2) / FW 0x62.43.63 (KBio2-Online): First version in English.
Version 2.11	8 noviembre 2012	FW 0x61.43.65 (BioMax2) / FW 0x62.43.65 (KBio2-Online): more options on Parameter$0B.

Notes

FIM biometric modules (*FIM20xx*, *FIM30xx* and *FIM5360*), as well as *eNBSP* software, are products belonging to Nitgen Co., Ltd.

Omnikey readers are products belonging to HiD Global Corporation.

KiWi Ethernet, *KiWi Wifi* and *KRD13M* are products belonging to Kimaldi Electronics, S.L.

This symbol indicates that Waste Electrical and Electronic Equipment (WEEE) should be disposed of separately to domestic waste. This measure is adopted to encourage reuse, recycling, and other ways of recovery, and to prevent possible damage to the environment and personal health. Go to a recycling plant to dispose of this product. If in doubt, contact your distributor or consult section "Environ ment" in our website at www.kimaldi.com

This information is only applicable to consumers in the European Union. In other countries, contact the local authorities to see whether this product can be recycled.