User guide
HEXAPAD 10 UHF READER
HEXAPAD 10 BC UHF READER

Review 1

BLANK
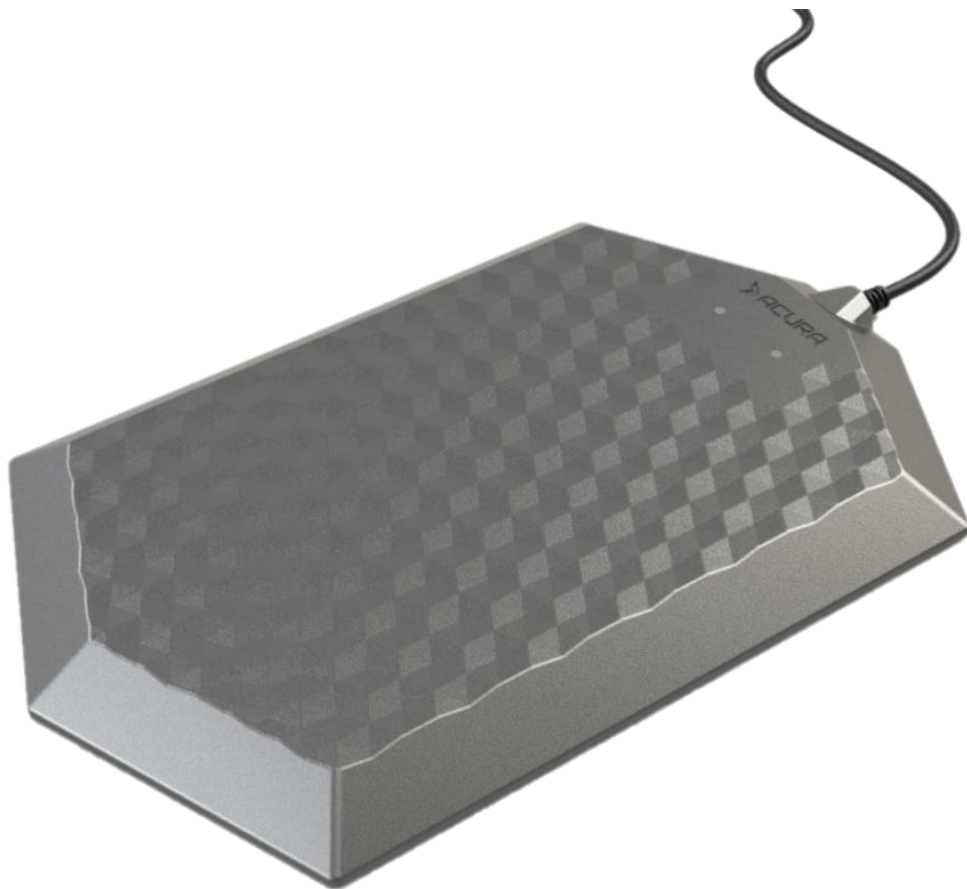
1 Products covered by this manual

This user guide belongs to the following product:

Reader	Code
HEXAPAD 10 USB GRAY	100.624
HEXAPAD-10 USB BLACK	100.629
HEXAPAD-10 USB WHITE	100.628
HEXAPAD-10 BC	100.646

1.1 Approval notes

National Telecommunications Agency (ANATEL):

The HEXAPAD reader was tested and approved in accordance with the Regulation for Certification and Homologation of Telecommunications Products, approved by Anatel Resolution No. 242 of November 30, 2000.
Types: Radio Frequency Identification Systems - Category II.
Service/Application: Restricted Radiation Radiocommunication.
"This equipment is not entitled to protection against harmful interference and may not cause interference to properly authorized systems."

Federal Communication Commission Interference Statement (FCC)

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

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:
- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

⚠ **FCC Caution:** Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

⚠️ **MPE Warning:** Radiation Exposure Statement – This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 21cm between the radiator & your body. Contains FCCID: QV5MERCURY6EN

Authorized Antenna

General Characteristics of the Integrated Antenna of HexaPad 10.

Vendor	Acura Technologies
Model	Integrated Near Field Antenna
Frequency Range (MHz)	915-928 MHz
Circular Gain (dBiC)	2.1 dBiC
Polarization	Circular (RHCP)
Type	Patch Near Field Antenna

1.2 Legal Notice

While all efforts have been made to ensure that this document and the information contained therein are correct, ACURA Technologies and any other parties involved in the creation of this document declare that it is provided "as is", without any express or implied warranty, including, but not limited to, any warranties that the use of the information contained herein has infringed any right, legitimacy or fitness to the purpose, and therefore disclaim any liability, directly or indirectly, for loss or damage related to the use of this document.

The information contained in this document may be changed without notice.

1.3 About Acura

Since the end of the 1990s, ACURA has been the pioneer in the radio frequency identification (RFID) market in Brazil and Latin America, and has successfully explored, since the late 1990s, its large-scale adoption in the most diverse sectors of the economy, from mining to steel, agriculture to food processing, from logistics to retail, from transportation to the distribution chain, access control to asset management. Promoter of new technologies, innovative, agile, and focused on the feasibility of cutting-edge projects.

Technological Development and Commercial Office
Wall Street Business
Av. Antartico, 381 - Jardim do Mar, São Bernardo do Campo - SP, 09726-150 (11)3028-4600

2 Introduction

2.1 About this guide

This user guide provides detailed information about the reader and is divided into three major parts, Part 1 with information for the reader installation, Part 2 with operation information, operation and configuration of the reader in Standalone mode and Part 3 with operation and operation information of the reader in Transparent mode, including programming information.

2.2 Conventions used in this manual

This guide uses some typographic conventions:

> ⚠️ **WARNING: Important warnings that require user attention.**

> **NOTE:**   Information and tips regarding the subject you are in.

2.3 Terms of definitions

The term "tag" used in this document refers to any RFID tag that follows the standard "EPCglobal UHF Class 1 Generation 2".
The term "EPC id" used in this document refers to the tag id that is in the EPC memory bank. The term "GS1 standard" used in this document refers to the "EPC Tag Data Standard version 1.11" which is the GS1 tag encoding standard.
The term "CDC" is the Acronym for "Communication Device Class", "ACM" is the Acronym for "Abstract Control Model", "HID" is the Acronym for "Human Interface Device" both used in the USB communication "Universal Serial Bus".
The term "ASCII" is the acronym for "American Standard Code for Information Interchange" and refers to a table with values for conversion of readable characters.
The term "<CR>" used in this manual represents the "Carriage Return" of the ASCII table and the ENTER key of the keyboard when mentioned in the commands.

2.4 Target audience

This document is intended for technicians who will be responsible for installing the reader. Before installing, configuring, and operating the reader, you should be familiar with:
   ● Data communication between devices, including the USB and Serial interface;
   ● Definitions of configuration, positioning of Reader and RFID Antenna;
   ● Basic knowledge about driver configuration in Operating Systems;
   ● Basic knowledge about the UHF EPC Gen2 protocol.

3 Introduction to equipment

The UHF HEXAPAD reader is a desktop reader with UHF (Ultra High Frequency) (UHF) RFID technology developed with a narrow reading field Near Field antenna, specific for checkout operations. Below are the main features:

- Its integrated antenna is of the **Near Field type**, restricting tag readings in a region above the reader, minimizing unwanted readings of nearby tags on the sides of the reader;
- It can work in two distinct ways, Standalone and Transparent:
  - ○ *Standalone Mode*:
    - Does tag readings automatically;
    - Turn off/turn on RF, read tag, or stay at rest;
    - Sends the results of the reads in the following interfaces:
      - Via keyboard emulation (native);
      - Via serial communication;
      - Via keyboard emulation + serial communication;
    - Various configuration parameters via serial terminal;
      - Reading power;
      - RF times;
      - Gen2 parameters, (tag >> reader) and (reader << tag);
      - Reader information;
      - Read operations;
      - Read filter by RSSI value of the tag;
    - Decodes standard Tags **GS1 SGTIN-96 and SGTIN-198** for checkout applications, sends in the following formats:
      - gs1string, example "(01)03608449920322(21)1540341";
      - gs1epcuri, example "urn:epc:tag:sgtin-96:1.360844.0992032.1540341";
      - gs1epcpureuri, example "urn:epc:id:sgtin:360844.0992032.1540341";
      - gs1gtin13, example "3608449920322";
      - gs1sgtin13, example "3608449920322 1540341";
    - Decodes tags in the Acura Autoid System standard for registering tags for vehicle identification application:
      - autoid + autoidcs
        - ○ wiegand26, example "172 13259";
        - ○ magstripe, example "00001034695627";
    - Decodes tags in the ASCII standard;
    - Option not to decode and send the value in epc hexa of the tag;
    - In addition to reading and decoding the Tag's EPC, you can read any tag memory bank;
  - ○ *Transparent Mode*:
    - The reader must be commanded by a Software that uses MercuryAPI, a complete API for reading and writing tags in the following languages:
      - C, .Net (C#) and Java;
- Compatible with Windows 10, Mac and Linux without driver required, for Windows 7 and 8 the driver is provided for installation;
- Wide variation of reading power: 0 to 27dBm (above 15dBm, use a Y extender cable for an additional power input).

4 Table of revisions

4.1 User guide table

Revision	Month /year date	Description
2	01/2020	● Added information about new firmware 1.1.3. ○ New retag command to enable and disable tag read (RF) in standalone mode.
1	06/2019	● Creation of this document.

4.2 Reader firmware table

Revision	Month /year date	Description
1.1.4	04/2020	● New "echochar" command with "on" and "off", to turn off the character echo via serial with the function of facilitating the development of software that will use the commands, so if the "echochar" is off, the software will only receive the command response. [Standalone mode]. ● Fixed error in the "summary" command in "gen2blf" when the "tagop" was pa, was shown 250. [Standalone mode].
1.1.3	01/2020	● New "readtag" command with "on" and "off", to turn off the rf and turn back on in standalone mode. ● New combination of Capslock (twice fast) to execute the readtag command. [Standalone mode]. ● Beep when running the readtag. [Standalone mode]. ● Beep on recurrency of readmode via CapsLock. [Standalone mode]. ● Tag read beep when readmode is serial and there is no serial communication open. [Standalone mode]. ● Improvement in transparent-autonomous transition and vice versa aiming at better reader stability.
1.1.2	07/2019	● Minor fixes and improved reader stability.
1.0.6	03/2019	● Fixed internal communication timing error that generated reader instability if RFON time was set to more than 500ms.
1.0.2	10/2018	● Creation and release of the firmware.

4.3 Readers hardware table

Revision	Month/year date	Description
V1	02/2021	● Creation of the HEXAPAD reader.

5 Installation notes and important warnings

⚠️ **AVISO:**	The maximum reading distance of passive tags per reader is up to 50cm. The effective reading distance of passive tags varies according to each installation and environment and may change according to. ● Material on which the tag is installed. ● Tag placement; ● Reader targeting in relation to the tag; ● Electromagnetic Interference caused by other equipament installed nearby

NOTE: It is recommended that reading tests are performed before the reader is installed at the place where the reader is to be installed.

---

[1] The reading distance may vary depending on the tag model, usage mode, and the environment.

Part 1 - Overview and installation

Equipment Overview Installation

6 Equipment overview

6.1 Hardware – HEXAPAD
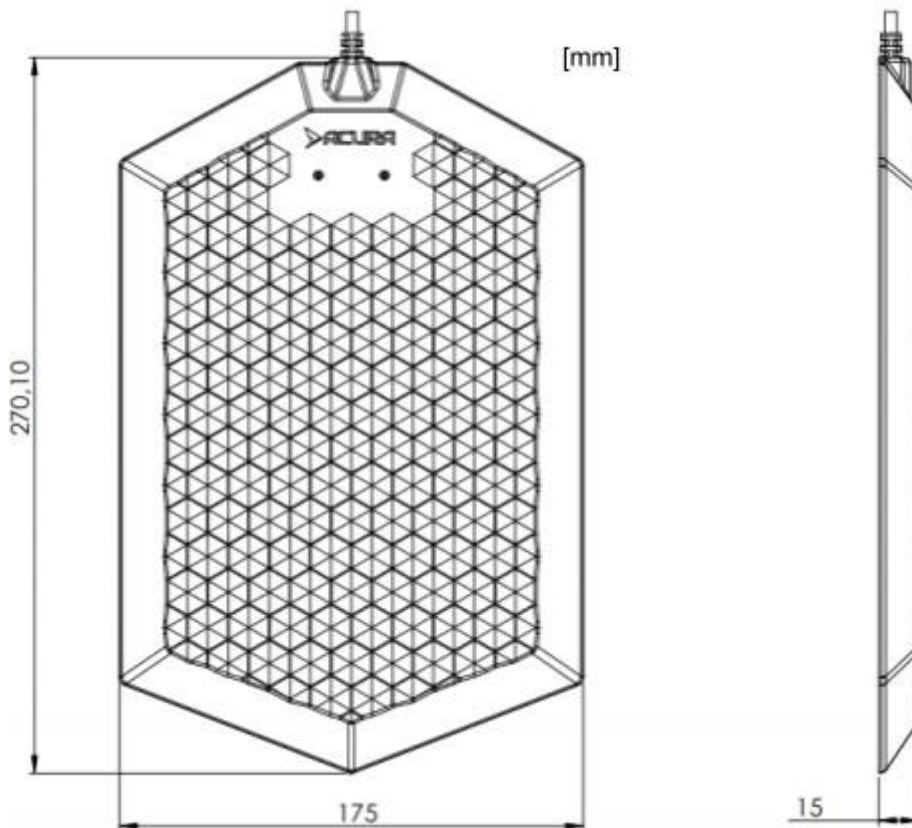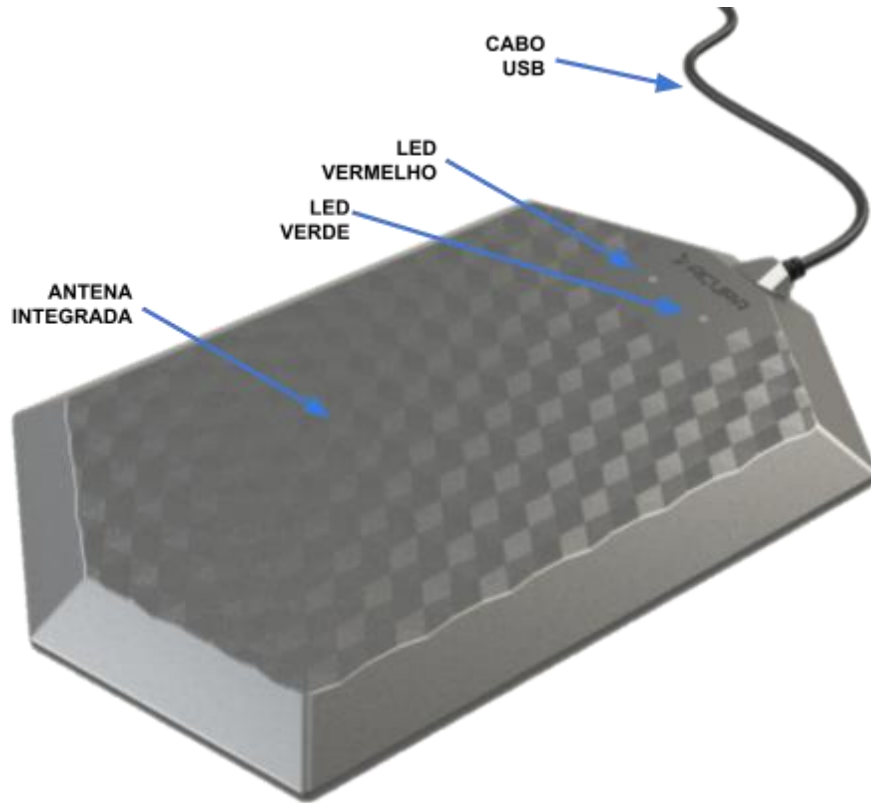
6.2 Hardware – Hexapad 10 BC

BAR CODE READER

RED LED

USB CABLE

INTEGRATED ANTENNA

GREEN LED

HexaPad

[mm]

270,10

175

15
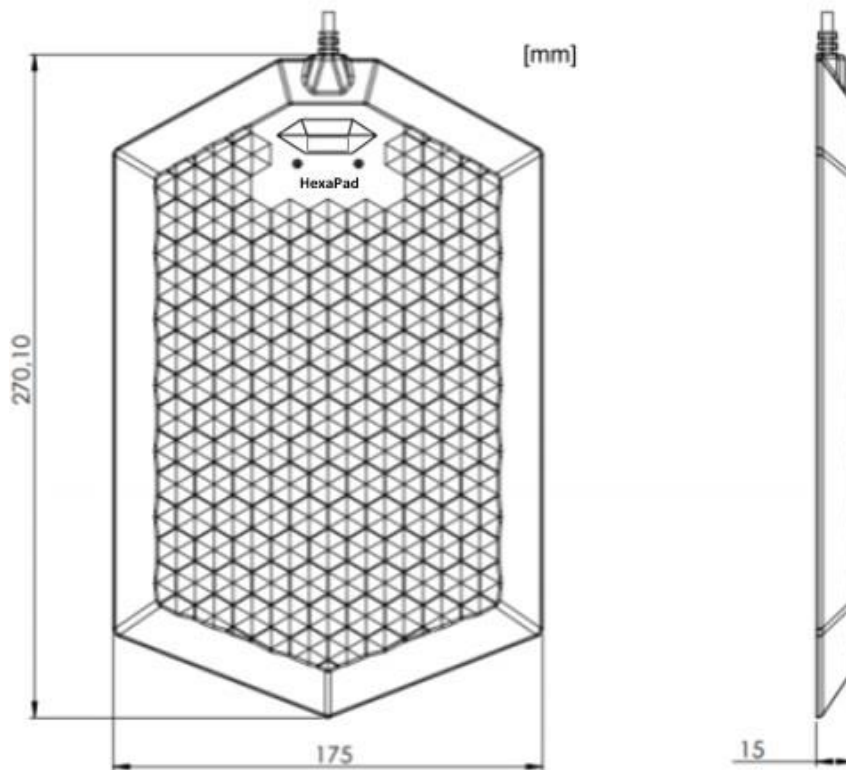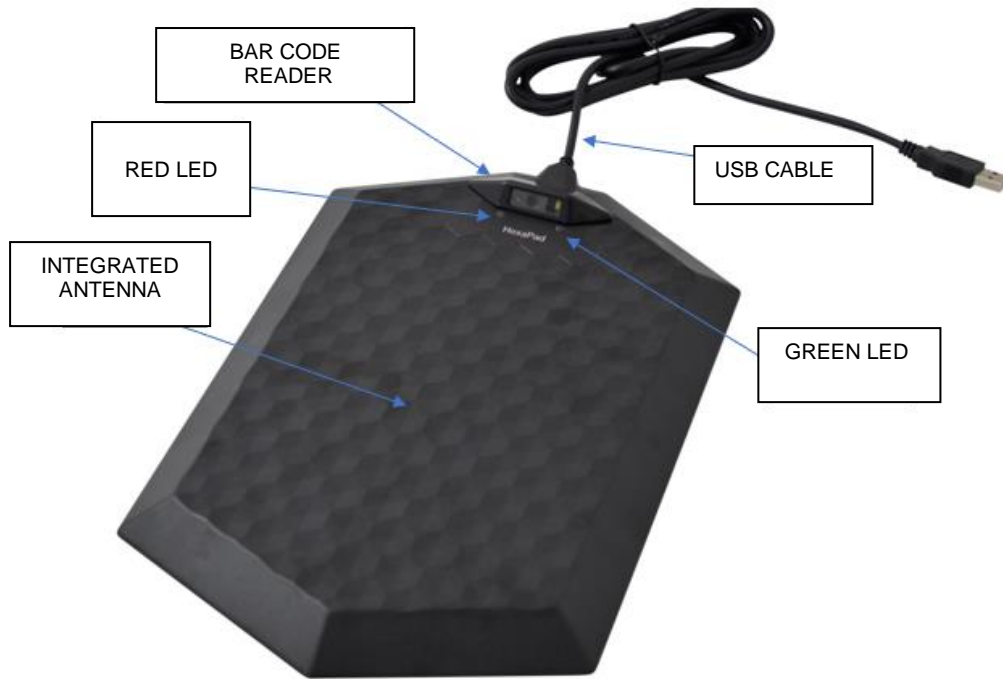
6.2.1   Barcode reading angulation

Field Vision

35º

31,5º

Field Vision

41º
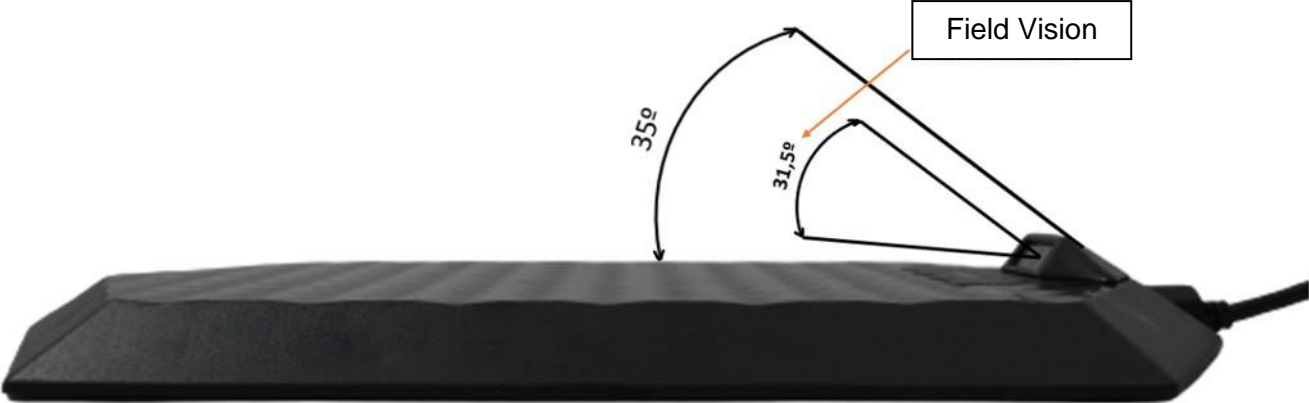
6.2.2 Distance specifications for barcode reading

For the barcode reading to be performed it is necessary to respect the distance and angulation in relation to the field of view between the scanner and the size of the code you want to read, so the larger the code is the further away it needs to be positioned from the scanner.

Examples:

1) <u>Small barcode</u>:

2) Big carcode:

NÃO VALE COMO RECIBO	
Vencimento ▮	Total a pagar (R$) ▮

N° fatura	Data de emissão	Conta referente a	N° instalação	Consumo (kWh)
00000000	▮	▮	▮	▮

000000000000 000000000000 0000000000000 000000000000
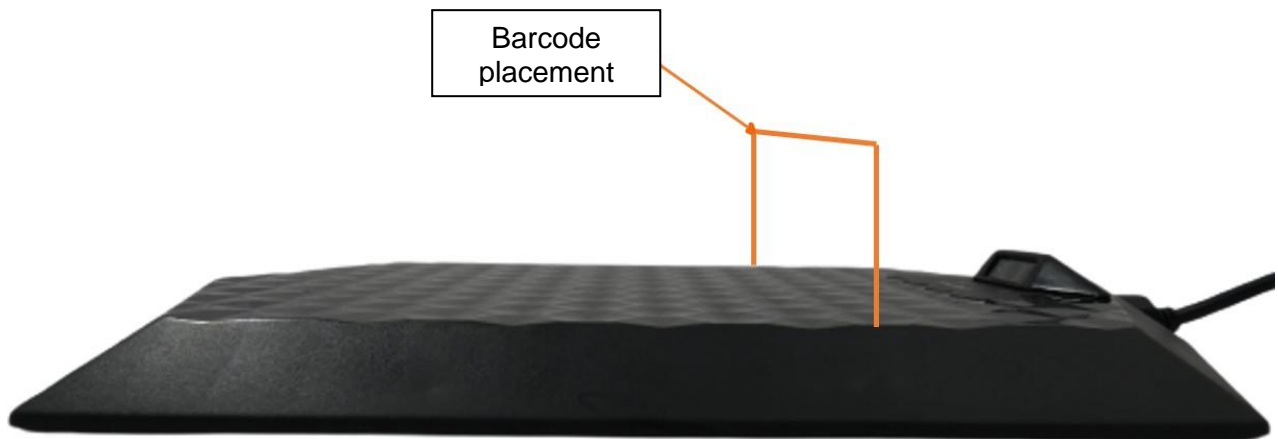
Barcode
placement
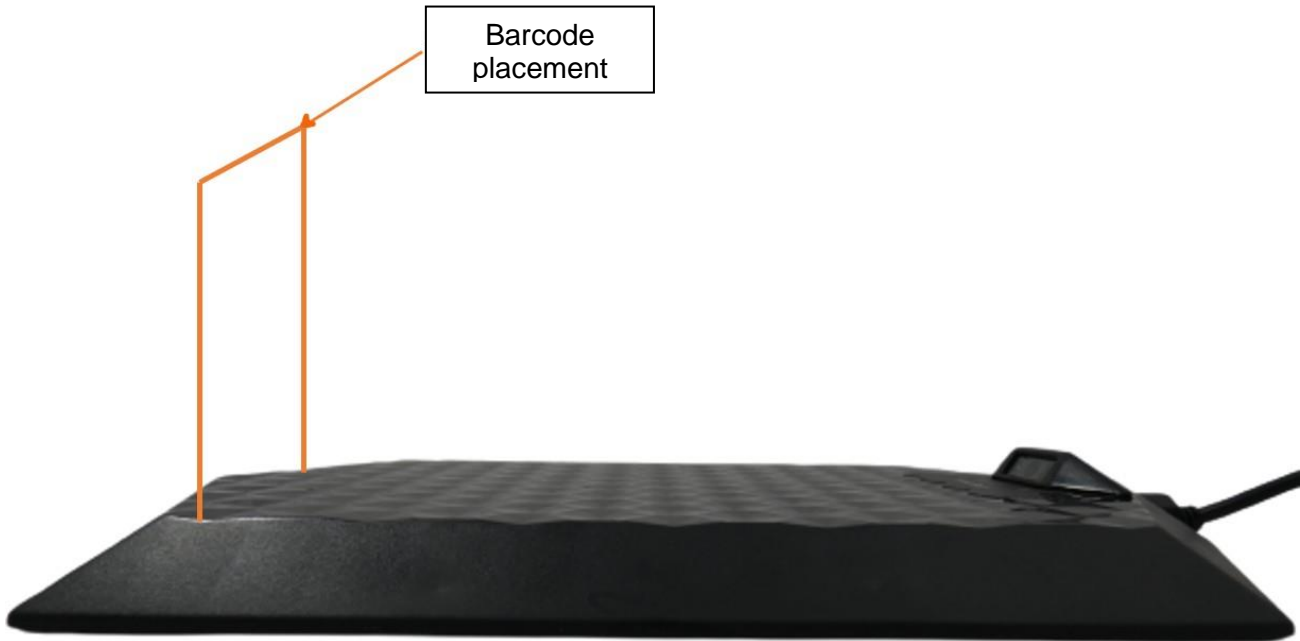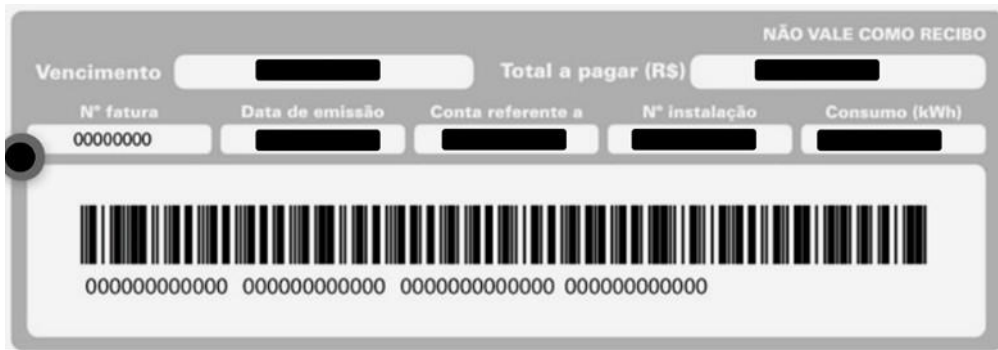
6.3 General reader characteristics

Transponder Protocols	
Protocol	EPCGlobal Gen2 (ISO 18000-6C). EPCGlobal Gen2 V2 (ISO 18000-63) (transparent mode).
Interface RF	
RF output power	0 to 27 dBm. 1dBm resolution.
Regulation	ANATEL (BR) 915 - 928 MHz.
Mode/Modulation/RF Encoding	Frequency Hopping / PR-ASK / M2, M4, M8.
Backscatter Link Frequency (BLF)	250KHz.
Performance	
Maximum reading distance[2]	50cm.
Data / Control Interface	
Connectors	**Communication and Food:** Usb connector type A male
Communication Interface	**USB HID keyboard:** The reader is recognized by the Operating System (Windows, Mac, and Linux) as a KEYBOARD-type HID device. **USB Serial CDC:[3]** When the reader is plugged into the USB, a Serial port is created automatically (Windows 10, Mac, Linux).
Programming/Configuration	**Standalone Mode:** Configuration through serial communication with ASCII commands via terminal (Putty, Minicom, Screen, etc). You don't need the use of API/SDK. Tag read only. **Transparent Mode:** The reader now responds to a Software that uses the Mercury API and has tag reading and writing features.
Protection	
ESD protection	Passive filter passes high (attenuation of 40dB in the ESD discharge band).

---

[2] The reading distance may vary depending on the tag used and the reader's usage environment.

[3] Windows 7 and 8 will require driver installation.

Energy	
Power (Direct Current)	Input voltage: 5VCC +/- 1%. (Host USB port). Maximum Supply Ripple: 200mVpp @20MHz.
Consumption (Direct Current)	Maximum 2.5W in standalone mode. Maximum 6W in transparent mode.

Physical characteristics	
Degree of protection of the reader	Internal use, does not have weather protection.
Integrated antenna	2.1 dBic
Dimensions	263x160x15mm
Operating temperature	-10°C to + 65°C
Storage temperature	-10°C to + 70°C
Relative humidity of the air	95%
Fixation	Table reader, no support for fixing.

6.3.1 General features of the integrated antenna

Electrical characteristics	
Frequency range	915 - 928 MHz
Gain	2.1 dBi
Polarization	Circular (RHCP)
Kind	NearField Antenna

6.2.1.1 Integrated antenna

NOTE: Top of the reader, causing tag readings on the sides near the reader to be minimized and if a very sensitive tag can still be read, its RSSI level is considerably lower than that same tag in a position above the reader.

NOTE: To increase the rejection of side readings, the RSSI filter can be used using the "rssifilter" command. More details in "8.6 Increasing rejection of side readings"

6.2.1.2 Radiation diagram (integrated antenna)
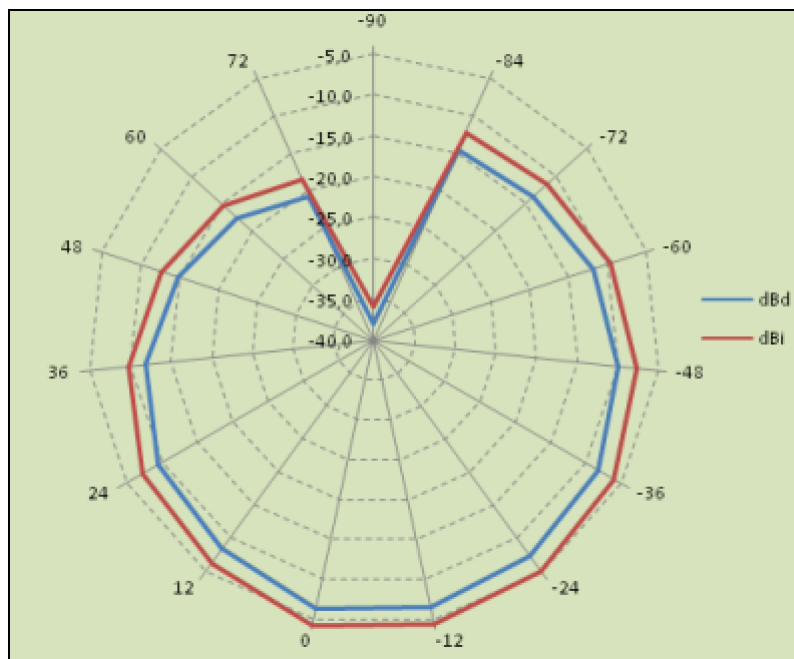
6.2.1.3 VSWR (integrated antenna)

MHz	VSWR
900	1,37
905	1,45
910	1,48
915	1,49
920	1,44
925	1,4
930	1,37

6.2.1.4 Gain Far Field

MHz	Ganho FF	
	dBd	dBi
900	-11,9	-9,69
905	-9,7	-7,49
910	-7,7	-5,49
915	-6,5	-4,29
920	-6,3	-4,09
925	-6,4	-4,19
930	-6,5	-4,29
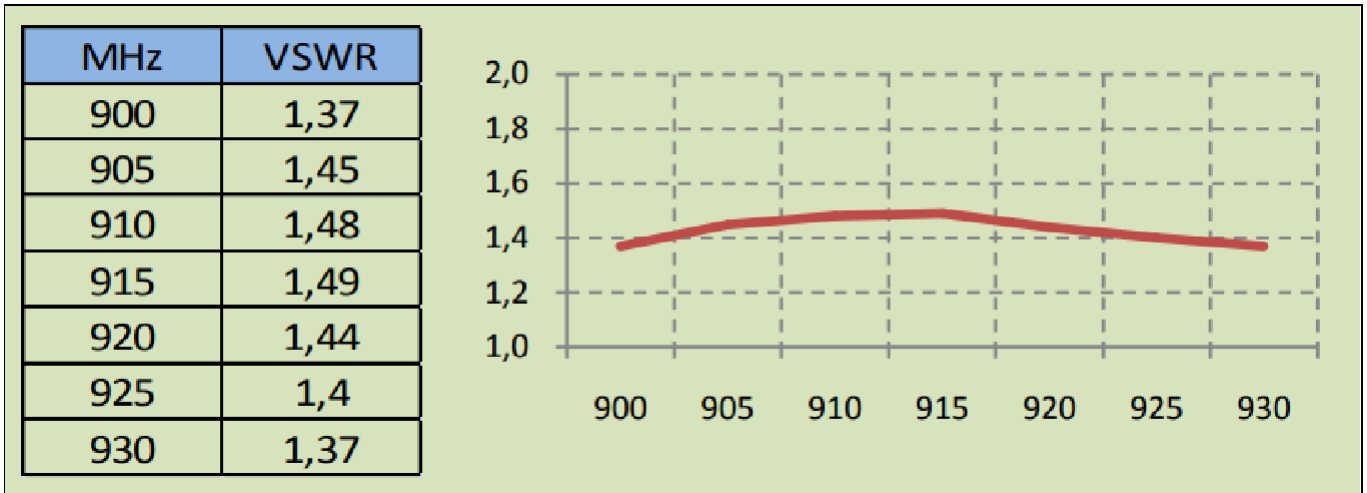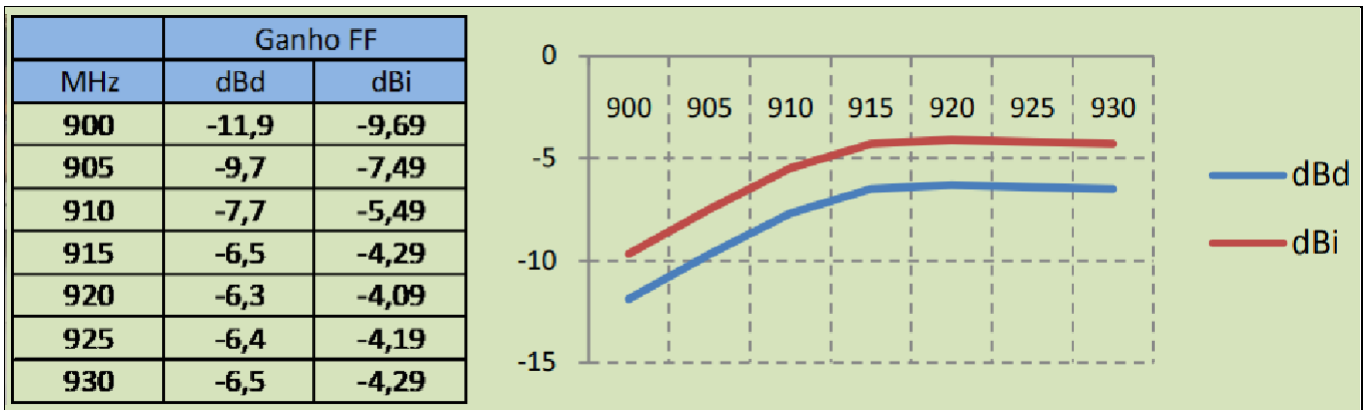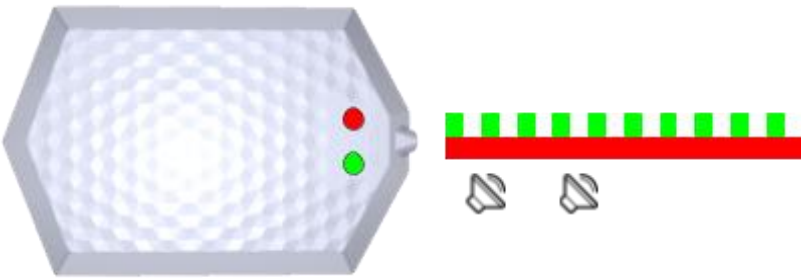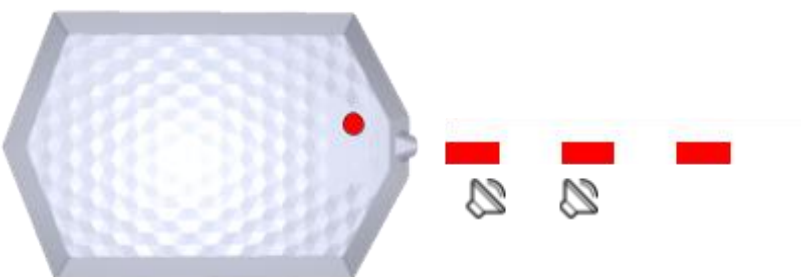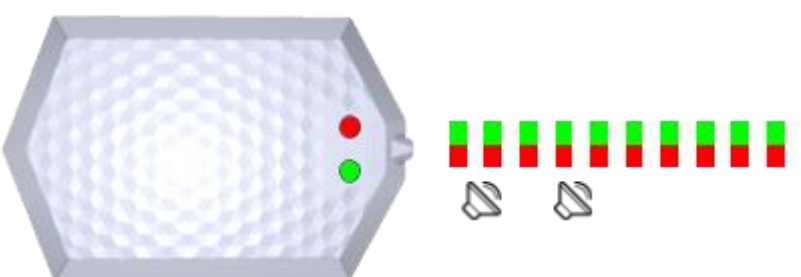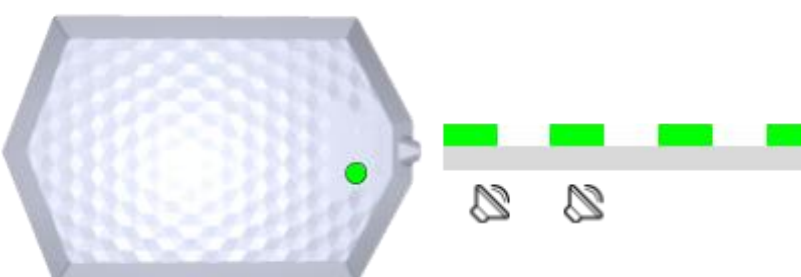
7 Installation

7.1 Turning on the reader

After connecting the reader to a USB port, the following combinations of the LEDs can be:

	Beep and Green Led flashes fast and red LED lit: Reader running in Standalone **mode,** trying to read tags.
	Beep and Green Led turned off and red Led blinking slowly: Reader running **in** Standalone mode, with RF disabled, **not performing tag reading**.
	Beep and green and red LED flash fast simultaneously: Reader energized however without having booted usb communication with the Operating System. Reader is not in operation until communication with the Operating System occurs.
	Beep and Green Led flashes slowly and Red Led turned off: Reader in Transparent **mode**. Note: In Transparent mode, the red Led and Beep are controlled by the user's Software.

On computers running Windows Operating System prior to 10, you will need to install a driver for the creation of the COMxx serial communication port.
For Linux, Android, and Mac Operating Systems, the driver is not required.

When the reader is turned on, the following devices will be created in the Operating System:
- HID class type keyboard;
- Serial port, (COM&LPT on Windows), (CDC on Linux, Android, and Mac)

Since the HID class keyboard type will be used by the reader for keyboard emulation when sending the reading results in Standalone mode, the Serial Port will serve as communication and configuration in Standalone and Transparent modes.

7.1.1 Identifying serial port (Windows)

After turning on the reader, go to the "Control Panel" or with the "Windows + X" key open the "**Device Manager**". In "**Ports (COM and LPT)**", a virtual serial communication port should appear 4s after the time the reader is connected, in addition to a new HID Keyboard type device in "**Keyboards**".
The images below show that the reader has been recognized by the Operating System and is ready to work.

Windows 7 - OK
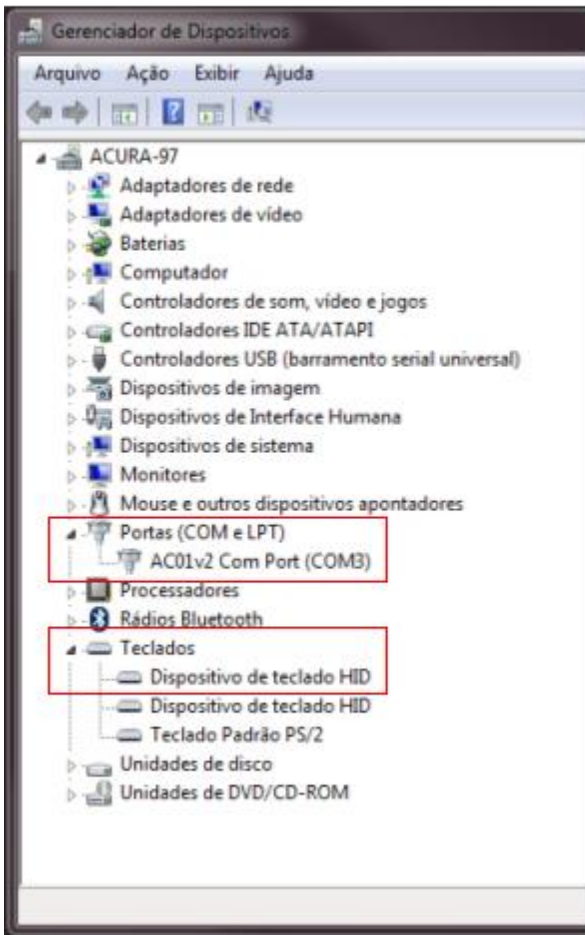
Windows 10 - OK

If Windows does not recognize the reader, as in the image below device manager, the reader driver must be installed following the following steps.

7.1.1.1 Installing driver for Windows

The Driver for Windows download must be done through the link:
https://drive.google.com/file/d/1PJBSm4KYy0w-L7tRK-BRwGBatgpwLF3r/view?usp=sharing

7.1.1.1.1 Step 1: Device detection

1. Connect the reader to your computer's USB port;
2. In the "Control Panel" or "Windows + X" key open the *"Device Manager"*;
3. Under "*Other Devices*" the reader connected to the computer should appear, as follows:
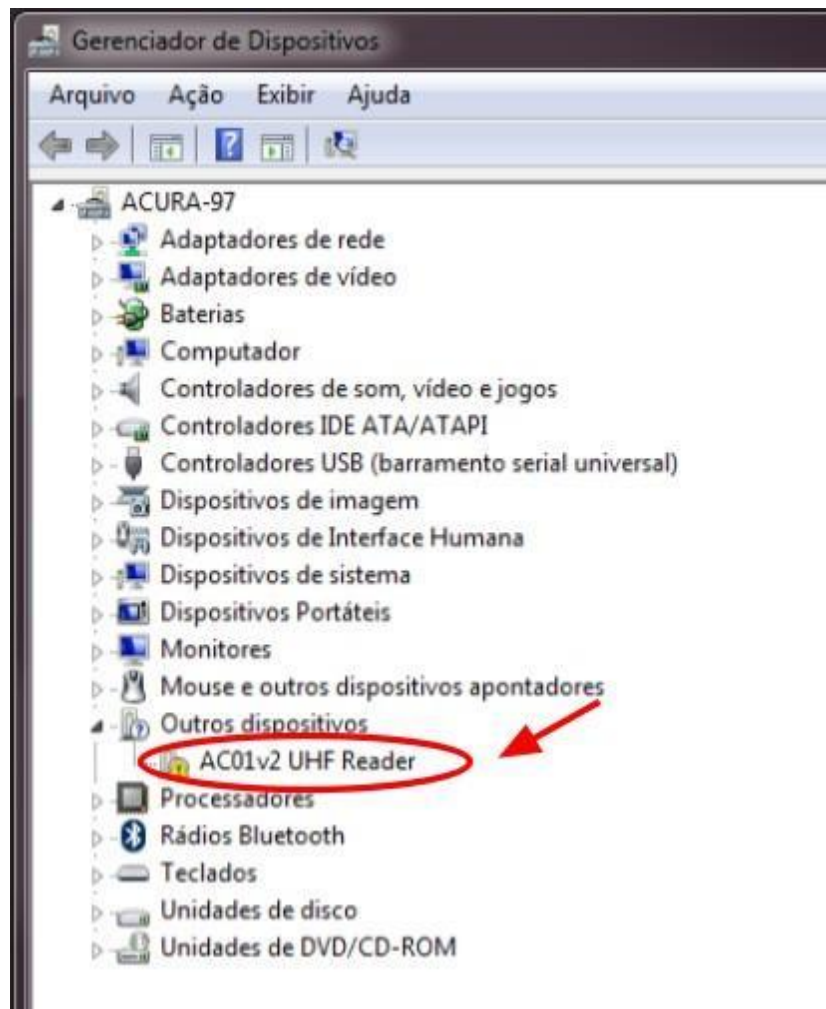
| NOTE: | For Windows 10 or above, Linux and Mac OS do not need driver installation |

7.1.1.1.2 Step 2: Load the driver

1. Right-click, go to "*Update Driver...* ":
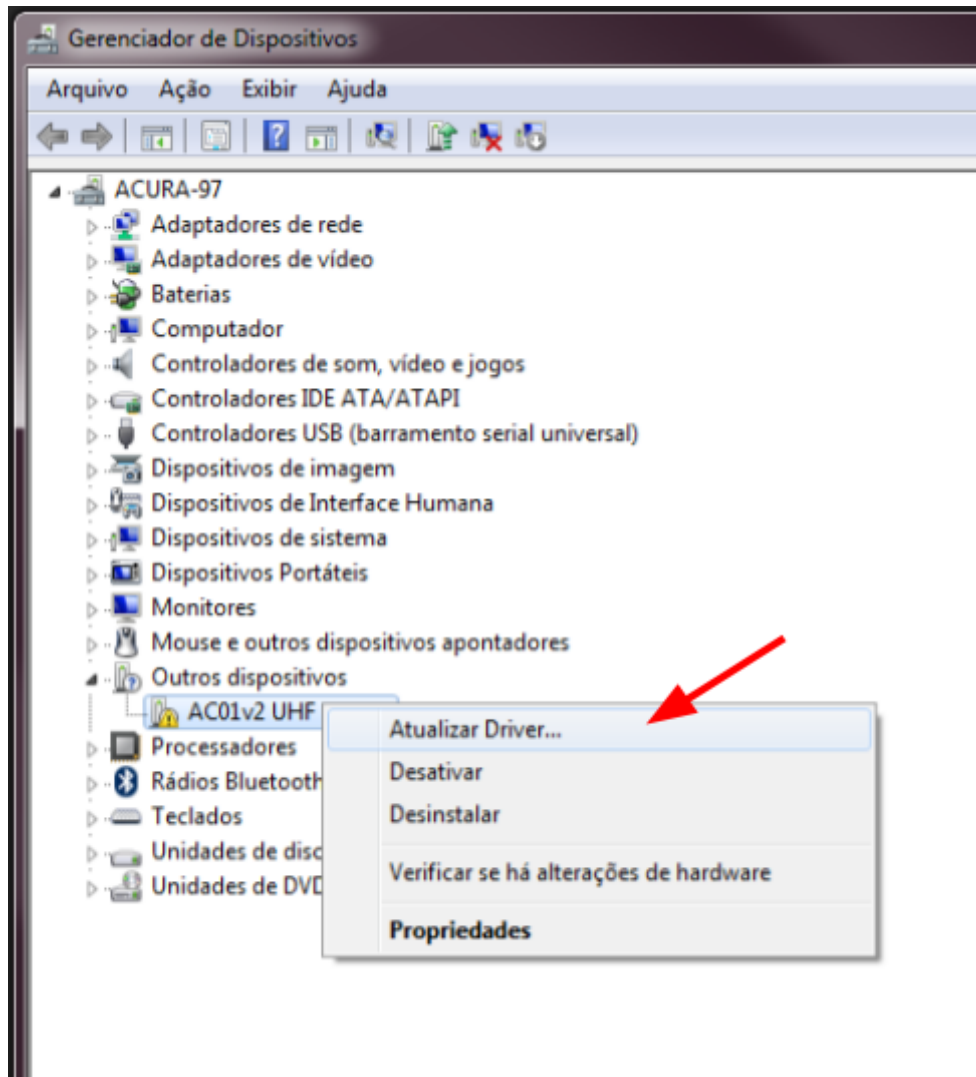
2. Then , "*Search for drive software on your computer*":
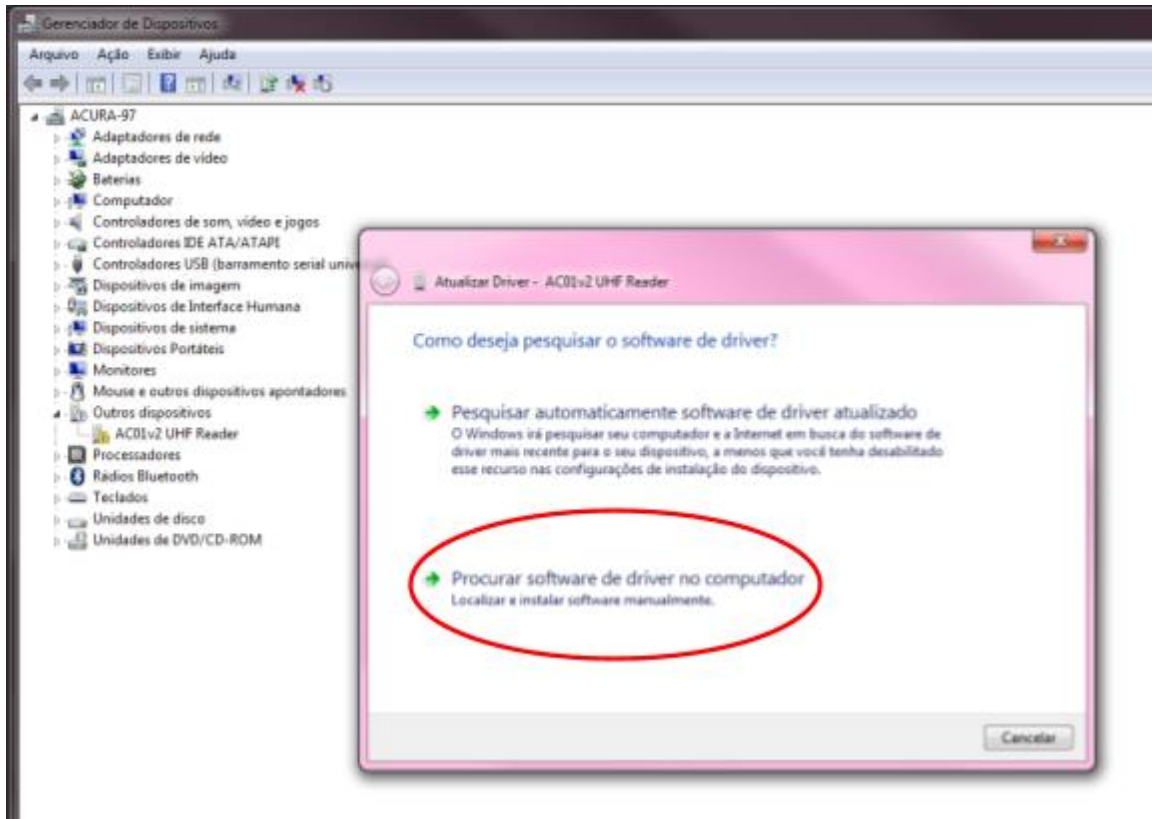
3. Later, select "*Allow me to choose from a list of device drivers on your computer*":
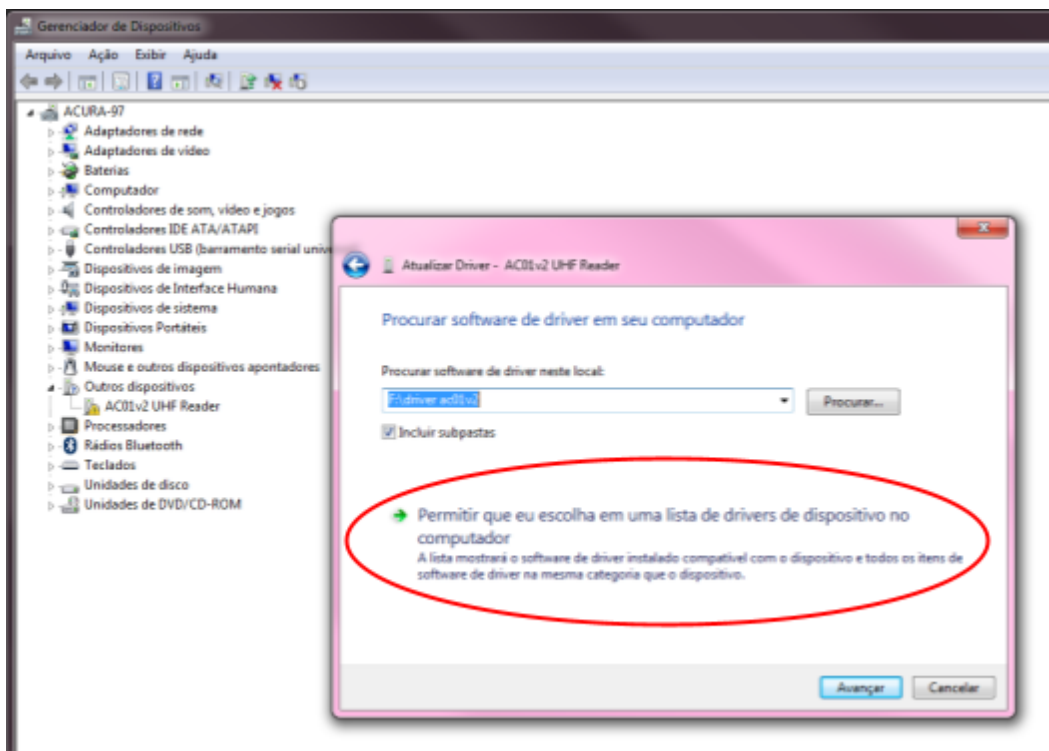
4. Under "Common hardware types" select "***Ports (COM and LPT)***" and click ***Next***:
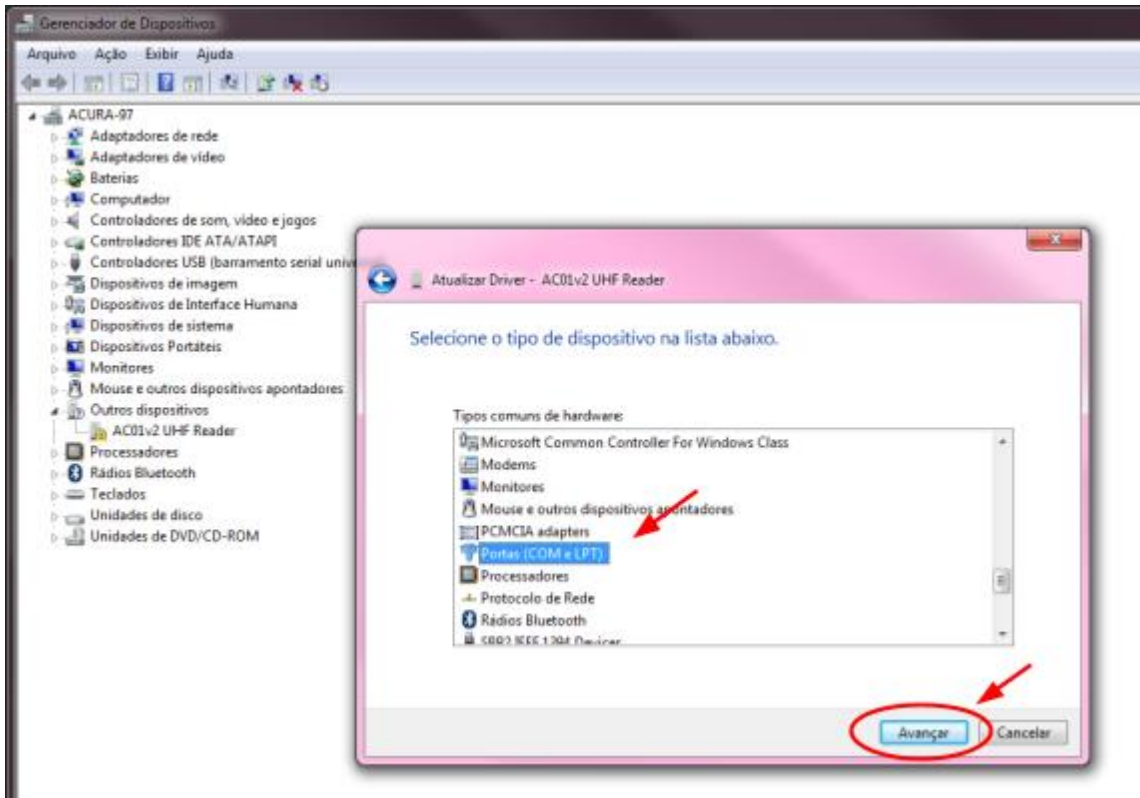
5. Click on "***With Disc...*** ";
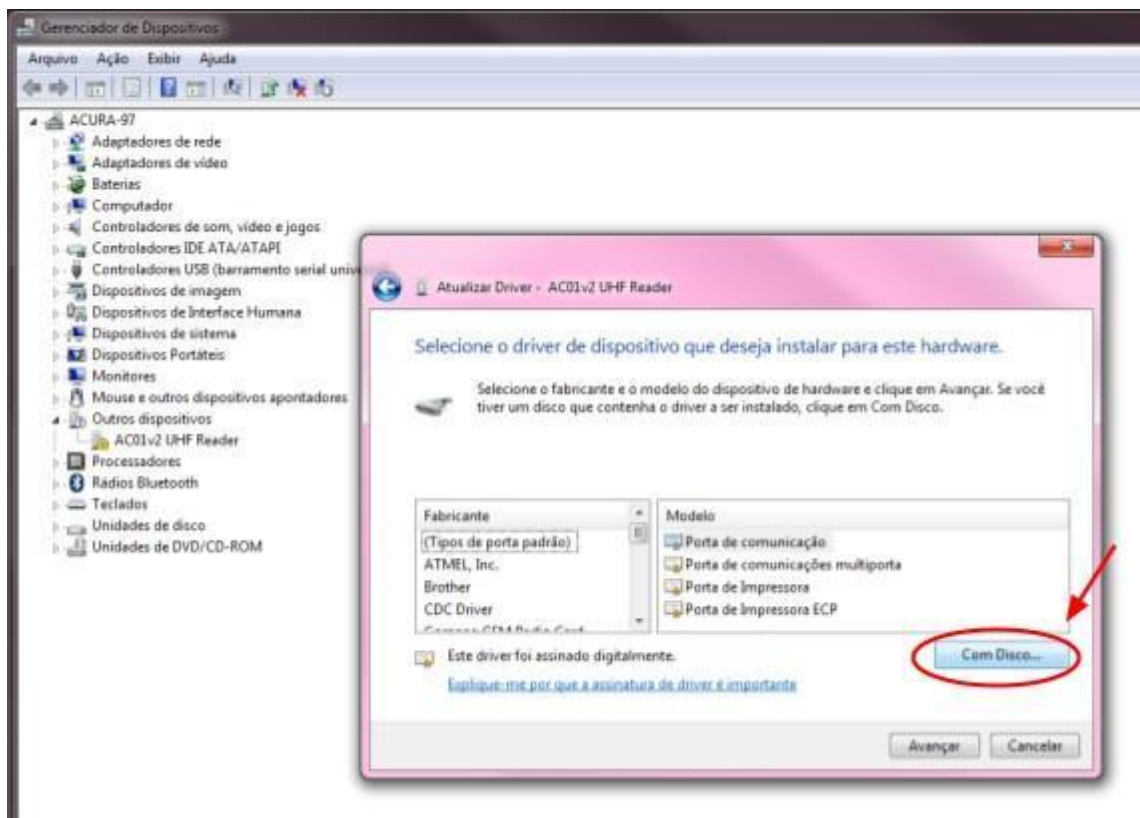
6. Under "*Install from Disk*", go to "*Search...* ", select the file for update and click "*OK*" to confirm. A page similar to the following should appear on your computer. Click "*Next*";
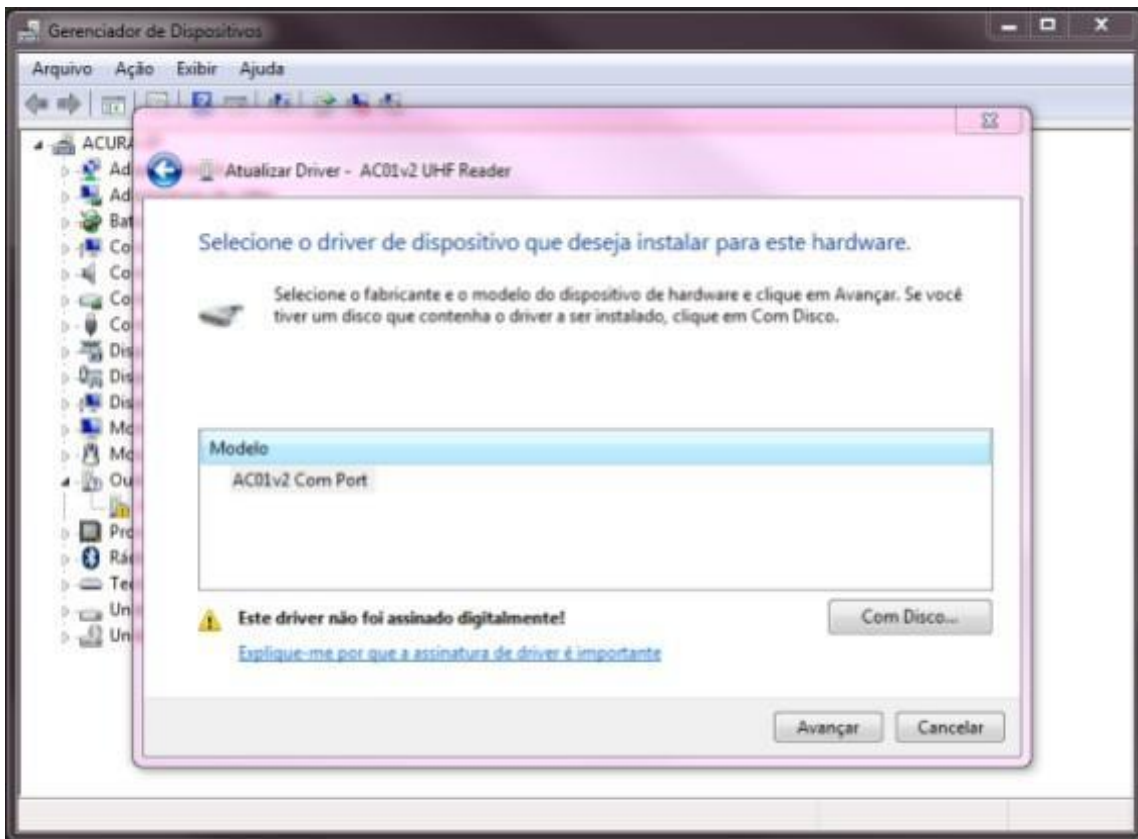
7. A driver update warning should appear. To proceed, click "*Yes*" to continue installing the driver;

8. A Windows security alert saying that "Windows cannot check the editor of this driver software" appears, to continue, click "**Install this driver software anyway**";
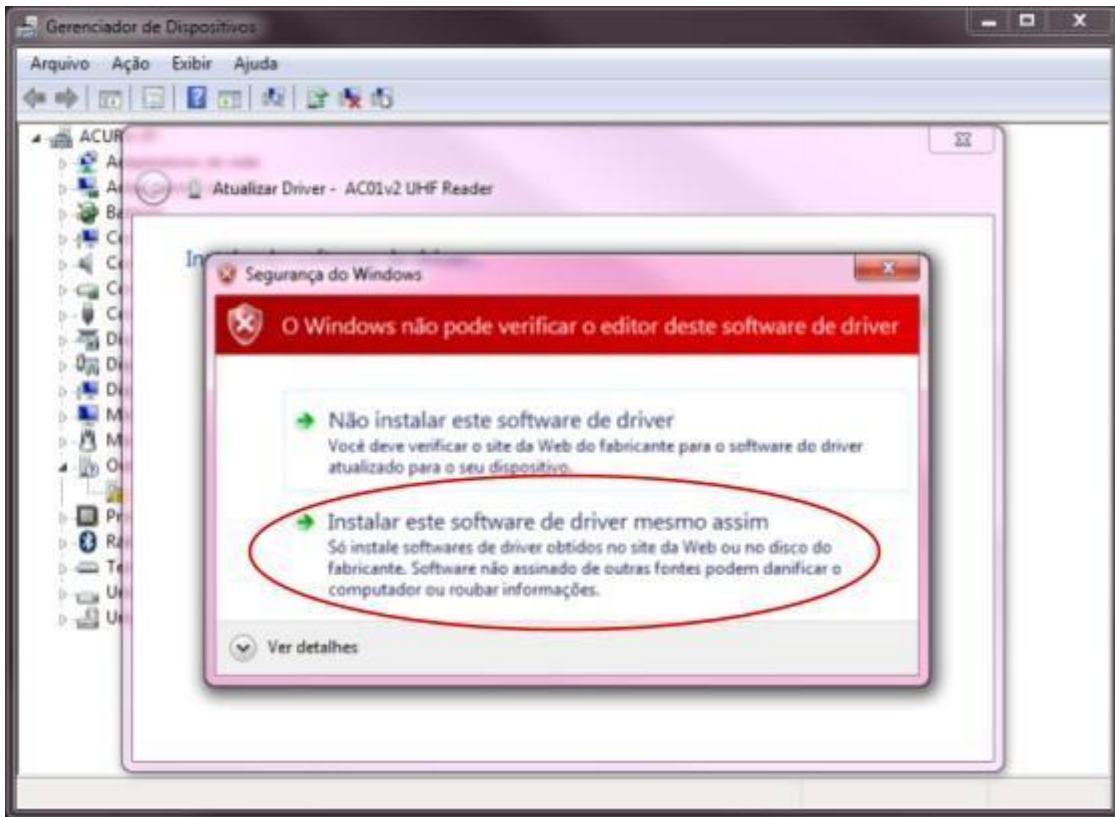
9. After this procedure, wait for installation;
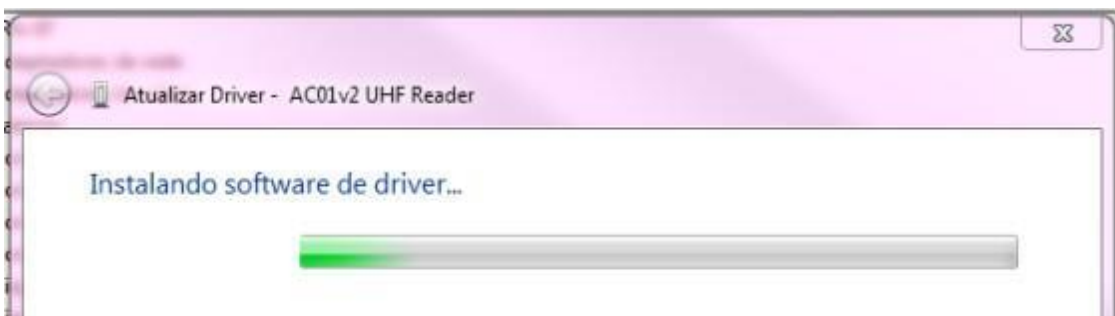
10. Once installed, the successful update message appears, thus ending the installation of the driver software for the AC01v2 With Port device.
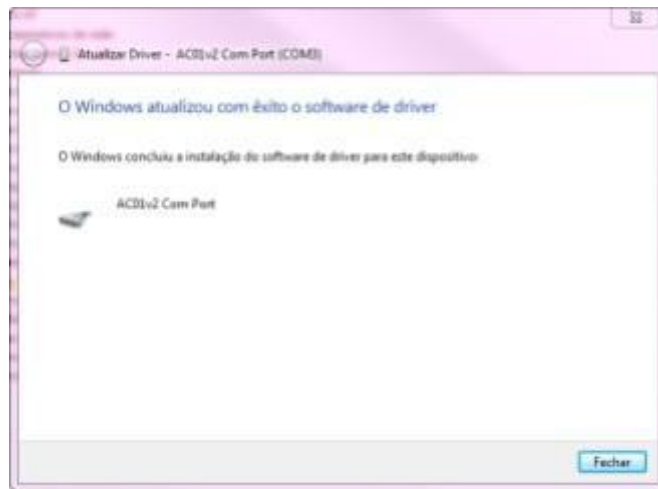
7.1.2 Identifying serial port (Linux)

To identify the name of the serial port created, right after the reader is connected to usb, type in the Terminal:

$dmesg | tail | Grep
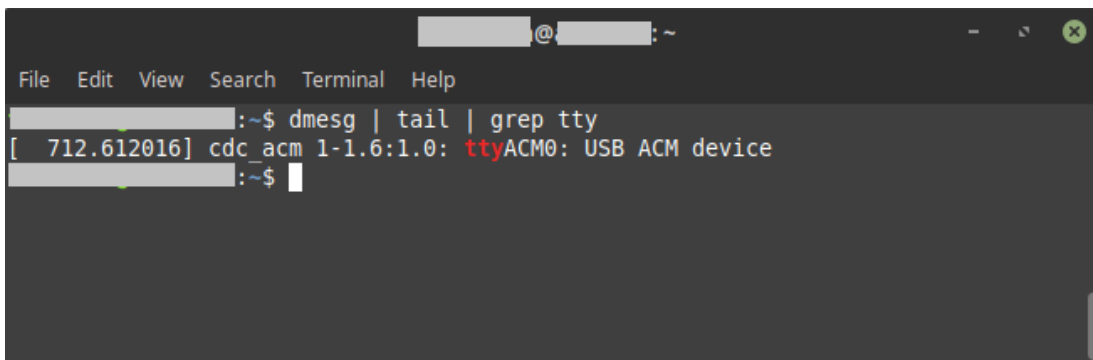
In the example above, serial communication with the reader will be through **/dev/ttyACM0**.

7.1.3 Identifying serial port (Mac OS)

To identify the name of the serial port created, right after the reader is connected to usb, type in the Terminal:

$ls/dev/tty.usb*
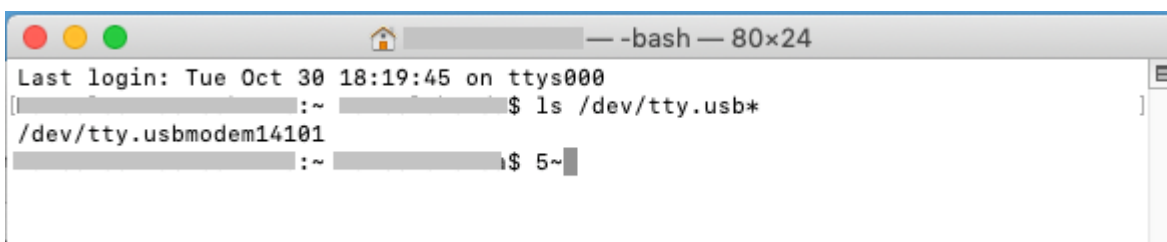
In the example above, serial communication with the reader will be via **/dev/tty.usbmodem14101**.

Part 2 - Standalone Mode

Reader in Standalone

Mode Reader Settings

Usage Examples

`

8 Reader in Standalone mode

In Standalone mode, the reader is constantly trying to read a tag and when a tag is read the information is sent through the configured communication interface. From firmware 1.1.3, the option to disable tag reading has been added by turning off the RF signal via the "**readtag**" command or **CapsLock key** and activate again, thus leaving the reader idle or tag reading.

Visual feedback from the reader when you are in the **Standalone mode by performing tag reading**:

	Green LED flashes fast and red LED lit
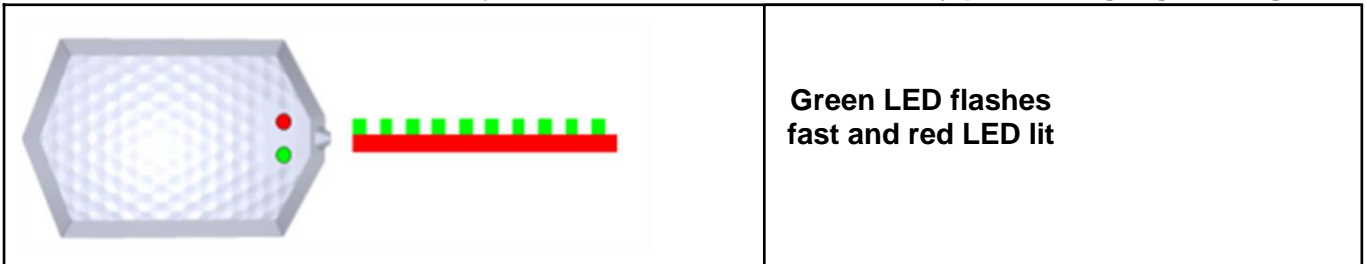
Visual feedback from the reader when you are in the **Standalone mode not performing tag reading, idle**

	Green LED off and Red LED flashes slowly
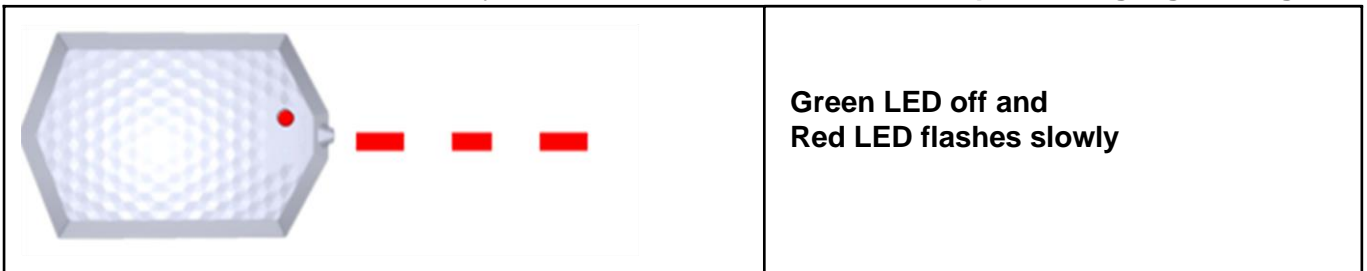
8.1 Turning RF on and off in standalone mode

To turn RF on and off, the following options can be used:
- A.   Use the **readtag command**, examples:
    - a.   The reader is in Standalone mode and the "**readtag off<CR>**" command is sent, so the reader disables the RF and no longer reads the tag.
    - b.   The reader is in Standalone mode and the "**read on<CR>**" command is sent, so the reader activates the RF and re-reads tags.
- B.   Use the "Caps Lock" key (Windows and Linux only), example:
    - a.   To switch between one operating mode and another, simply *quickly click the CapsLock key 2 times in sequence and* check the reader's LEDs and beep to see the change.

8.2 General diagram of standalone mode
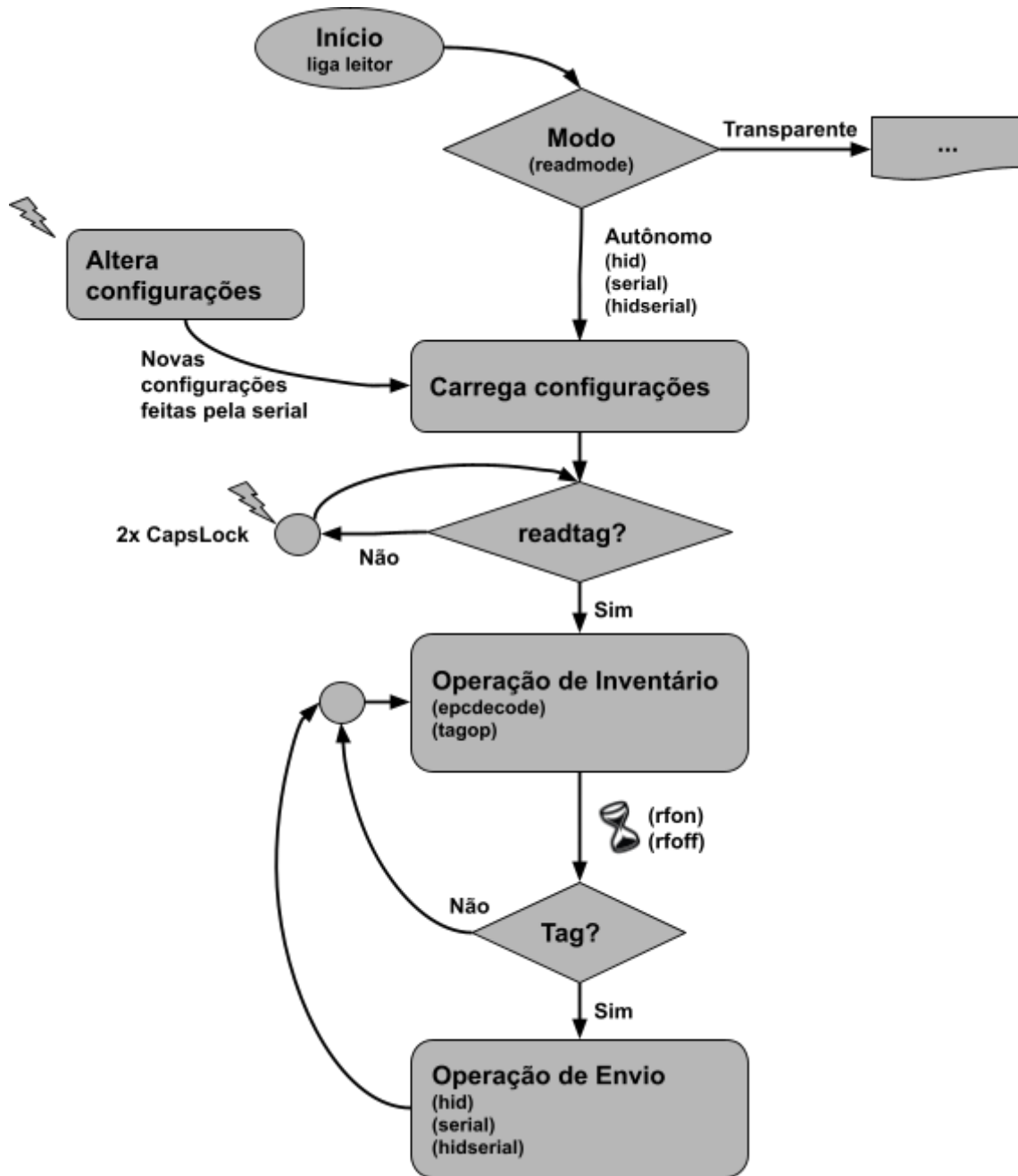
The reader with the most basic configuration reads only the EPC id of the tag in hexadecimal format. The EPC id of the tag will always be read and reported by the reader and its decoding can be changed. In addition to reading the EPC id it is possible to read together any memory bank of the UHF EPC Gen2 tag, as well as the amount of reading and its RSSI signal level. All of this information is sent by the reader in a string formatted as explained in the next chapter.

8.3 Reading result format

In Standalone mode, the reader sends a string formatted with the tag read results in the chosen communication interface.
Below is the format of string:
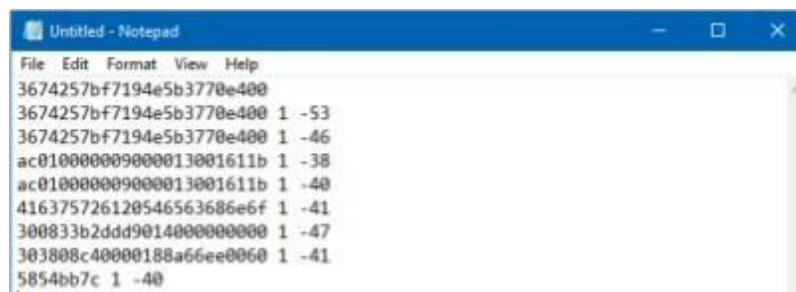
<epcdecode><separator><tagop><separator><readcount><separator><rssi><endofline>
<epcdecode> EPC id read result with the decoding specified by the **epcdecode command**. *Note: Always reported by the reader in Standalone mode.*
< separator> Separator character specified by the **separator command**. *Note: It is only reported by the reader if there are more fields in addition to <epcdecode> to report.*
<tagop> Result of reading the operation specified by the **tagop command**. *Note: Optional.*
<readcount> Amount of reading of the same tag in an inventory period. The option to show or not the value is made by the **reportreadcount command**. *Note: Optional.*
<rssi> Rf signal level value in dBm and negative of tag response. The option to show or not the value is made by the **reportrssi command**. *Note: Optional.*
<endofline> End-of-line type configured by the **endofline command**. *Note: Reported by the reader in Standalone mode if configured other than none.*

⚠	**AVISO:**	**The size of the result string may vary depending on the tags read, so if it is a User Software that will receive this string, always treat it using the <separator> and <endofline> fields to separate the desired fields.**

Examples of reading result:

8.4 Configuring the reader

When the reader is in Standalone mode, the following settings can be performed:
- Type of communication;
- Read the tag or stand at rest (idle);
- Reading power;
- Reading time;
- Rest time;
- EPC id decoding type of the tag;
- Read operation on the memory bank of the tag;
- EPC Gen2 protocol parameters
  - BLF;
  - Tari;
  - Encoding;
  - Session;
  - Target;
  - Q (Anti-collision);
  - Access password;
- Character that classifys fields from the result string;
- End-of-line type of the result string;
- Typing speed in keyboard emulation;
- Filter by RSSI;
- Show or not the RSSI value of the read tag;
- Shows or not the amount of reading of the read tag;
- Tag reading beep;
- Check the internal temperature;

8.4.1 How to set up the reader?

The reader in Standalone mode must be configured through ASCII commands sent by the serial port created on the Operating System when the reader is turned on.

> **NOTE:** To find out which serial port the reader will respond to, please refer to

8.4.1.1 Operating logic

Setting up the READER via ASCII messages works with the Message/Response framework:
- A. An external Host sends a command to the reader;
- B. The reader interprets and executes the command inserted into the received message;
- C. The reader responds to the external Host.

Messages and replies use ASCII characters with a carriage return terminator **<CR>**, which is [ENTER]. Messages can be cashed, low, or both.

> The reader echoes each character typed in the serial terminal and each new line the reader sends the character "**>**" to indicate the cursor
>
> **NOTE:**

8.4.1.2 Syntax of messages and replies

Below is the syntax of messages from Host to Reader:

VALUE<CR COMMAND>	Changes the value of the command/parameter.
COMMAND<CR>	Reads the value of the command/parameter.

⚠️ ALERT	**Only one value must be passed to each command, the reader will not accept more than one value per command.**

Below is the syntax of reader response messages to host:

RESULT<CR>	Returns the result of the execution of the last command sent to the reader.

8.4.1.3 How to send the commands to the reader? (Windows)

Reader settings in Standalone mode can be run through terminal emulation software. The best known free and free code is **Putty** whose download link follows below.
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html To
set up Putty follow the following steps:

1. Open Putty and select "Connection type:" "Serial", "Serial line" port that the reader generated, and "Speed" 115200.
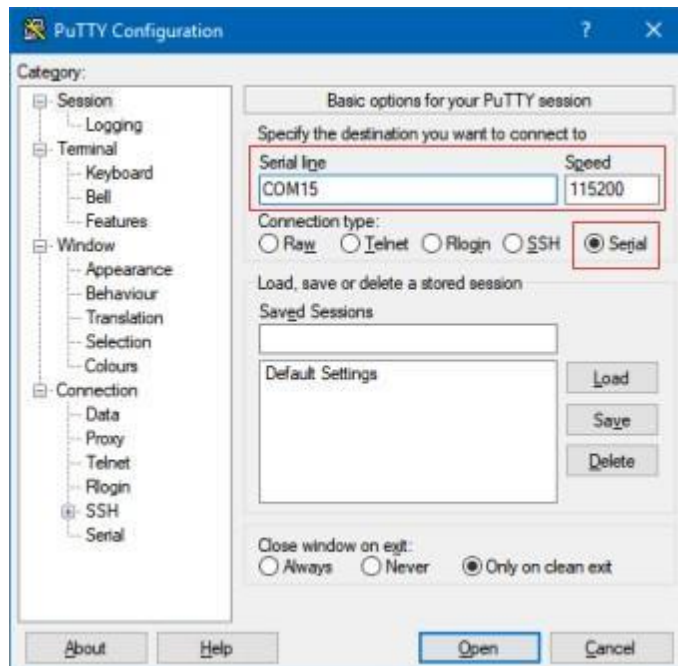
2. Click "Open" to open the terminal screen for typing commands.

8.4.1.4 How to send the commands to the reader? (Linux and Mac OS)
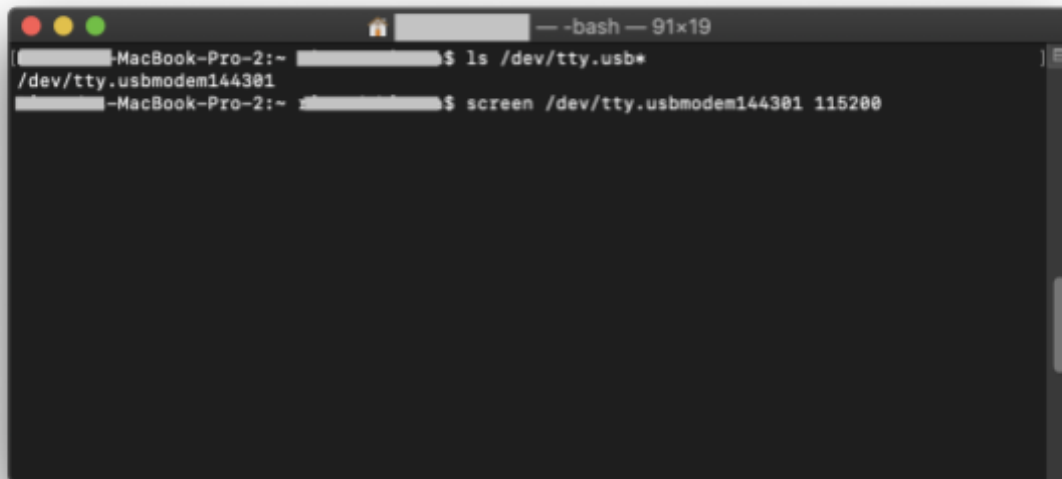
On Linux operating systems you can also use Putty as explained above for Windows.
For Mac OS and Linux you can use the "screen" program as below.

> **NOTE:** To find out which serial port the reader will respond to, please refer to 7.1 Turning on the

(Mac OS)
$ **screen /dev/tty.usbmodemXXXXXX 115200**
Where "XXXXXX" is a unique identifier number generated by the operating system.

(Linux) using screen.
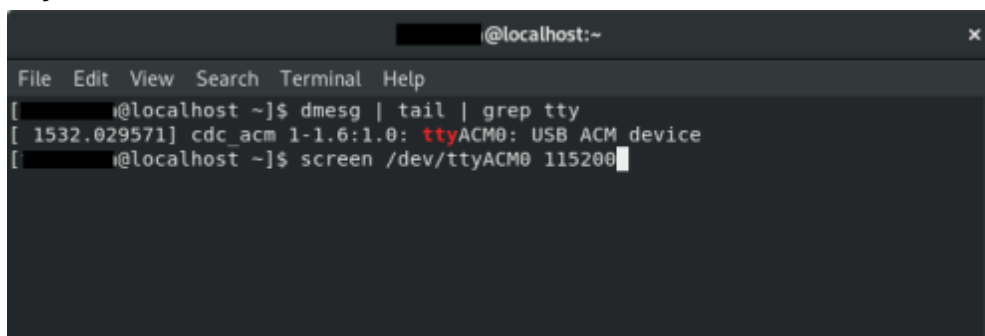
> **NOTE:** On linux to have permission to use the serial port, run the command with super user permission (sudo) or add your user to the dialout group using the following command:
>
> $ **sudo usermod -a -G dialout $USER**
>
> To take effect, you need to restart the system.

To connect using the screen:
$ **screen /dev/ttyACMx**
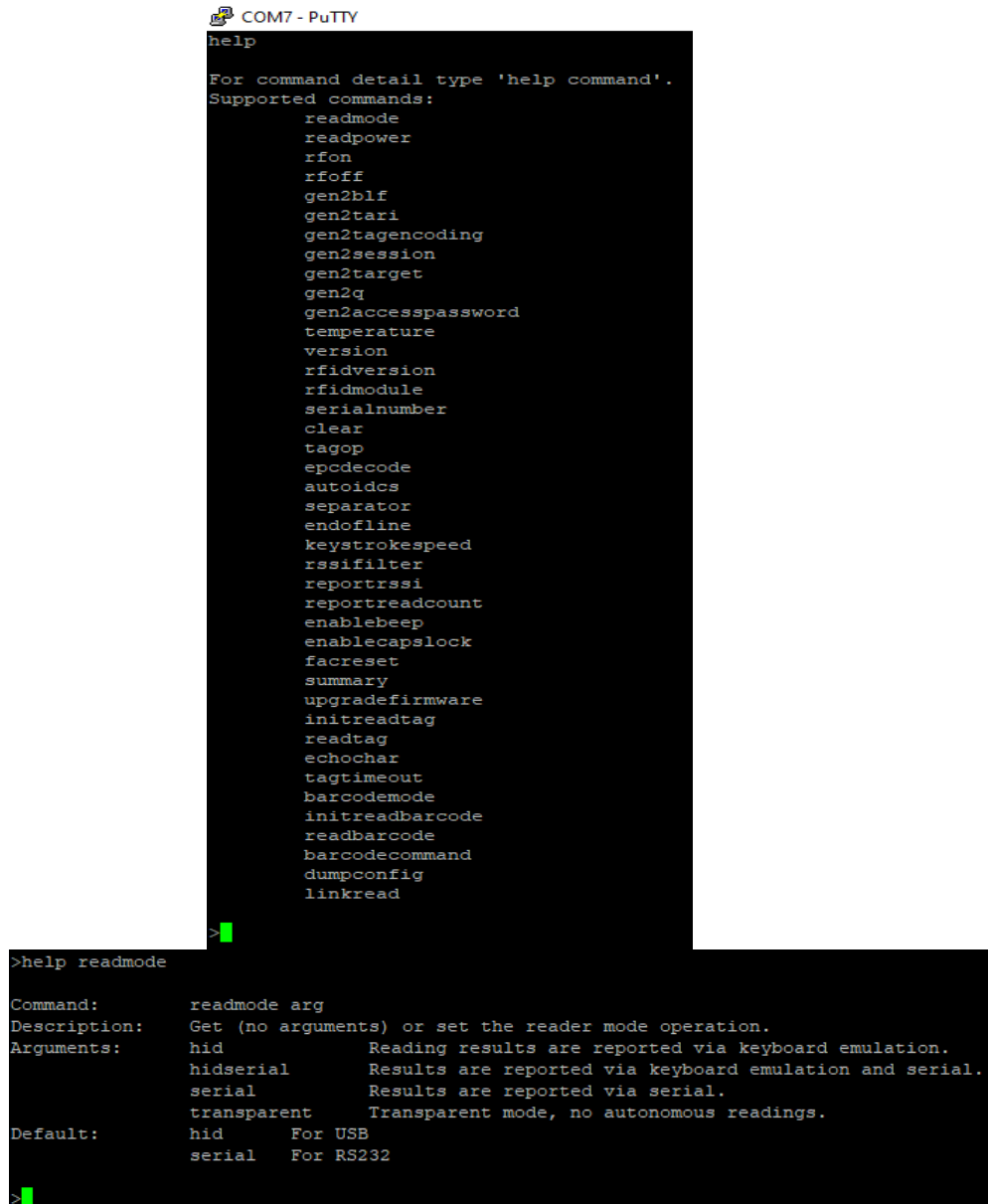
8.5 Configuration commands

8.5.1 Important tip

It is not always easy to remember the name of the command or the accepted values, so there is **the help** command that if executed without any value lists all the commands supported by the reader and to get details of a specific command just **type help command<CR>**, this is a quick way to access the configuration information of the reader. Below is images from the help command.

```
COM7 - PuTTY
help

For command detail type 'help command'.
Supported commands:
        readmode
        readpower
        rfon
        rfoff
        gen2blf
        gen2tari
        gen2tagencoding
        gen2session
        gen2target
        gen2q
        gen2accesspassword
        temperature
        version
        rfidversion
        rfidmodule
        serialnumber
        clear
        tagop
        epcdecode
        autoidcs
        separator
        endofline
        keystrokespeed
        rssifilter
        reportrssi
        reportreadcount
        enablebeep
        enablecapslock
        facreset
        summary
        upgradefirmware
        initreadtag
        readtag
        echochar
        tagtimeout
        barcodemode
        initreadbarcode
        readbarcode
        barcodecommand
        dumpconfig
        linkread
>
```

```
>help readmode

Command:        readmode arg
Description:    Get (no arguments) or set the reader mode operation.
Arguments:      hid             Reading results are reported via keyboard emulation.
                hidserial       Results are reported via keyboard emulation and serial.
                serial          Results are reported via serial.
                transparent     Transparent mode, no autonomous readings.
Default:        hid     For USB
                serial  For RS232

>
```

8.5.2 Summary of configuration commands and default values

Command	Description	Pattern
Readmode	Reading mode and communication interface.	Hid
readpower	Reading power in dBm.	15
rfon	Reading time in ms.	250
rfoff	Downtime in ms.	250
gen2blf	EPC Gen2 Backscatter Link Frequency in kHz.	250
Gen2tari	EPC Gen2 Tari, pulse duration of one bit in us.	25
gen2tagencoding	EPC Gen2 tag-reader communication encoding.	M4
Gen2session	EPC Gen2 tag section.	s1
gen2target	EPC Gen2 tag target.	the
Gen2q	EPC Gen2 tag amount in the reading field.	dynamic
gen2accesspassword	EPC Gen2 tag access password.	00000000
Temperature	Temperature in °C of the reader.	
Version	Firmware version of the reader driver.	
rfidversion	Firmware version of the RF module of the reader.	
rfidmodule	Name of the RF module of the reader.	
serialnumber	Serial number of the reader.	
Tagop	Tag read operation.	None
Epcdecode	Decoding the EPC tag id.	None
autoidcs	Security code for Acura's Autoid tags.	9000001
separator	Result string separator character.	' ' (space)
endoline	End-of-line type of the result string.	Windows
keystrokespeed	Typing speed of hid interface.	Fast
rssifilter	Filter by RSSI value of the tag.	0
reportrssi	Reports the RSSI value of the read tag.	Off
reportreadcount	Reports the amount of tag reading in the "rfon" period.	Off
Enablebeep	Tag reading beep.	on
enablecapslock	Shortcut via "CapsLock" key for switching standalone and transparent modes.	on
Help	Help text for each command.	
summary	Summary of the values of the reader settings.	
facreset	Returns all settings to the factory default.	
upgradefirmware	Starts the firmware update process.	
readtag	Turns RF on or off in standalone mode	on
echochar	Turns character echo on or off in Standalone mode.	on
tagtimeout	Time out	0 ms
barcodemode	Barcode reading mode	Hid
initreadbarcode	Initialize barcode reading	on
readbarcode	Barcode reading	on
linkread		Inverse

8.5.3 Response messages to commands

When receiving a command, the reader executes and responds with one of the following messages:

Kind	Description
Success	When the value entered is valid and different from the current value written to the reader: **ok - parameter has been changed.<CR>** When the value reported is valid and equal to the current value written to the reader: **ok - parameter already has this value.<CR>** When the facreset command is executed: **!!! Factory Reset in action !!! The reader will be restarted.<CR>**
Errors	**error - command not found<CR>** When a non-existent command is sent to the reader. **error - value out of range<CR>** When the value entered is not between those specified and accepted by the reader. **error - too many args.<CR>** When more than one value is passed to the command. **error - value not supported for this RF Module.<CR>** When a value is passed not supported by the RF module. **error - gen2 wrong parameters combination.<CR>** When a value not supported by the ALREADY CONFIGURED BLF is passed. **error - internal buffer full.<CR>** When the amount of characters before <CR> is greater than 50. **error - command not supported for this RF Module.<CR>** Informed command is not supported by the internal RF module. **error - command not supported for this comm interface.<CR>** Informed command is not supported by the communication interface used.

8.5.4 Detailing configuration commands

8.5.4.1 readmode - Read mode

Command	**readmode** *value*
Function	Check (no value) or change the mode of operation of the reader.
Description	The reader can function autonomously with automatic reads, or Transparent where a Software can have full control of the reader for tag reads and writes.
Pattern	**Hid**
Values	**Hid** (Self-employed) Keyboard emulation mode, when a tag is read the result is sent to the computer as if it were typed quickly. **hidserial** (Self-employed) Keyboard emulation mode and USB Serial, the reading result is sent via keyboard emulation and also via serial communication on the USB virtual port. **serial** (Self-employed) Serial communication mode, the read result is sent via serial communication on the USB virtual port. **transparent** - The reader must be controlled by a Software that has full control of the reader to perform readings and writing scans of tags.
Example	To check how it works: **readmode<CR>** **HID<CR>** To change the mode of operation (Standalone reporting via serial only): **serial readmode<CR>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

8.5.4.2 readtag- Turns RF on or off, tag reading in standalone mode

Command	**readtag** *value*
Function	Check (no value) or turn RF on/off.
Description	Enables or disables tag reading by shutting down the reader's RF signal.
Pattern	**on**
Values	**on** **off**
Notes	By pressing 2x fast the CapsLock key runs this command cyclically on/off.
Example	To check the value: **readtag\<CR\>** **off\<CR\>** To disable RF and tag reading: **retag off\<CR\>** **ok - parameter has been changed.\<CR\>**
Errors	When the value reported is not among the specified: **error - value out of range\<CR\>** When more than one value is passed to the command: **error - too many args.\<CR\>**

`

8.5.4.3 readpower - Read power

Command	**readpower** *value*
Function	Check (no value) or change the read power in dBm.
Description	The reader in Standalone mode can have the read power range from 0 to 15 dBm.
Pattern	**10 dBm**
Values	**0 .. 15**
Example	To check the reading power: **readpower<CR>** **12<CR>** To change the reading power to 5 dBm: **5<>CR** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

> **NOTE:** The reader will reach and its power consumption, we always recommend adjusting the best reading power according to your application.

`

8.5.4.4 rfon - Reading time

Command	**rfon** *value*
Function	Check (no value) or change the read time in ms (milliseconds).
Description	The reading time is equivalent to the time when the reader transmits RF power to its antenna in an attempt to read a tag. When you have too many tags to read at the same time, the reading time should be increased.
Pattern	**250 ms**
Values	**50 .. 5000**
Example	To check the reading time: **rfon\<CR\>** **250\<CR\>** To change the reading time to 500 ms (0.5 s): **500\<CR rfon\>** **ok - parameter has been changed.\<CR\>**
Errors	When the value reported is not among the specified: **error - value out of range\<CR\>** When more than one value is passed to the command: **error - too many args.\<CR\>**

NOTE: The reading results are reported by the reader after the sum of rfon rfoff times.

NOTE: If the application requires reading multiple tags at the same time, leave the rfon above 250ms to give time to energize all tags in the reader field.

8.5.4.5 rfoff - Downtime

Command	**rfoff** *value*
Function	Check (no value) or change the read interval time in ms (milliseconds).
Description	The reading interval time is equivalent to the time when the reader turns off the RF power on your antenna.
Pattern	**250 ms**
Values	**0 .. 1000**
Example	To check the read interval time: **rfoff<CR>** **250<CR>** To change the reading interval time to 500 ms (0.5 s): **500<CR >** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

NOTE: The reading results are reported by the reader after the sum of rfon rfoff times.

8.5.4.6 gen2blf - EPC Gen2 Backscatter Frequency Link

Command	**gen2blf** *value*
Function	Check (no value) or change the "Backscatter Link Frequency" parameter of the Gen2 pattern.
Description	Blf is the data rate in KHz of transmission by air between a tag and the reader.
Pattern	**250 Khz**
Values	**250** **320** **640**
Notes	By changing the BLF to 640 or 320 KHz, the tari value and tag encoding will automatically change to 6.25 us and fm0 respectively.
Example	To check the BLF value: **gen2blf<CR>** **250<CR>** To change the BLF value to 640 KHz: **640<CR gen2blf>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

⚠️ **AVISO:** **Changing the gen2blf will imply automatic change by the reader of another EPC gen2 parameters, which are gen2tari and gen2tagencolding**

`

8.5.4.7 gen2tari - EPC Gen2 Tari pulse time

Command	**gen2tari** *value*
Function	Check (no value) or change the "Tari" parameter of the Gen2 pattern.
Description	The Tari is time in us (microseconds) of the pulse of each bit transmitted by the tag.
Pattern	**25 U.S.**
Values	**6.25** **12.5** **25**
Notes	Values 12.5 us and 25 us can only be set if the gen2blf is set to 250 KHz. The values 6.25 us and 12.5 us are supported only by the internal RF module micro-module type, which can be checked by rfidmodule command.
Example	To check the Tari value: **gen2tari<CR>** **25<CR>** To change the value of Tari to 12.5 us: **gen2tari 12.5<CR>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>** When a value not supported by the RF module is passed: **error - value not supported for this RF Module.<CR>** When a value not supported by the already configured BLF is passed: **error - gen2 wrong parameters combination.<CR>**

8.5.4.8 gen2tagencoding - EPC

Command	**gen2tagencoding** *value*
Function	Check (no value) or change the "Tag data encoding" parameter of the Gen2 pattern.
Description	It is the type of encoding of the data sent by the air by the tag.
Pattern	**m4 (Miller 4)**
Values	**fm0** **m2** **m4** **m8**
Notes	m2, m4 and m8 can only be configured if the blf is 250 KHz. fm0 has a low read sensitivity rate but a higher speed in the decoding of each bit. m8 has a high read sensitivity rate but a lower speed in decoding each bit.
Example	To check the Data encoding value: **gen2tagencoding<CR>** **m4<CR>** To change the value of the encoding tag to m8: **gen2tagencoding m8<CR>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>** When a value not supported by the already configured BLF is passed: **error - gen2 wrong parameters combination.<CR>**

8.5.4.9 gen2session - EPC Gen2 tag section

Command	**gen2session** *value*
Function	Check (no value) or change the "Session" parameter of the Gen2 pattern.
Description	It is the type of session in which the reader will read a tag.
Pattern	**s1**
Values	**s0** **s1** **s2** **s3**
Notes	The combination of gen2session and gen2target values reflects the read repetition rate of the same tag.
Example	To check the Session value: **gen2session<CR>** **s1<CR>** To change the session value to s0: **s0<CR gen2session>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

> **NOTE:**   Use **gen2session and gen2target to** manipulate the read rate of the same tag, as described in 8.5 Read rate of the same tag.

8.5.4.10 gen2target - EPC Tagert tagert

Command	**gen2target** *value*
Function	Check (no value) or change the "Target" parameter of the Gen2 pattern.
Description	It is the flag of each tag session.
Pattern	**the**
Values	**the** Reading session flag tags in state A. **b** Reading session flag tags in state B. **Ab** Reading flag tags from the session in state A, then in B. **Ba** Read tags with session flag in state B, then in A.
Notes	The combination of gen2session and gen2target values reflects the read repetition rate of the same tag.
Example	To check the Target value: **gen2target<CR>** **a<CR>** To change the target value to ab: **ab<CR gen2target>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

> **NOTE:** Use **gen2session and gen2target to** manipulate the read rate of the same tag, as described in 8.5 Read rate of the same tag.

8.5.4.11 gen2q - EPC Gen2 amount of tags in the reading field

Command	**gen2q** *value*
Function	Check (no value) or change the "Q" parameter of the Gen2 pattern.
Description	It is a numeric value that the reader uses to regulate the probability of a tag's response, used for anti-collision when multiple tags are in the reader's reading field. 2^Q equals the amount of tags to be read at the same time.
Pattern	**dynamic**

Values	**dynamic**
	     The reader automatically adjusts the Q value during readings.
	
	0 .. 15
	     Fixed value for 2^Q number of tags in the reading field of the reader.

Notes	If the amount of tags to be read at the same time varies, leave the q value in dynamic, so the reader automatically adjusts the best value for each reading.
Example	To check the Q value: **gen2q<CR>** **dynamic<CR>** To change the value q to 3, with 8 tags in the reading field: **3<CR gen2q>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

> **NOTE:** Use **gen2q as** dynamic **if** the number of tags in the reading field varies or is not known.

8.5.4.12 gen2accesspassword - EPC Gen2 tag access password

Command	**gen2accesspassword** *value*
Function	Check (no value) or change the "Access password" parameter of the Gen2 pattern.
Description	Is the 32-bit (8-digit) password of the hex tag.
Pattern	**00000000**
Values	**00000000 .. ffffff**
Notes	To read the reserved memory bank of the tag with the tagop command, the password must be the same as the tag.
Example	To check access passrowd: **gen2accesspassword<CR>** **00000000<CR>** To change the password to 12345678: **gen2accesspassword 12345678<CR>** **ok - parameter has been** **changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

NOTE: To read the protected memory region of the tag, Reserved, the tag access password must be reported to the reader via this command.

8.5.4.13 tagop - Tag reading operation

Command	**tagop** *value*
Function	Check (no value) or change the type of read operation of a tag.
Description	With the type of read operation it is possible to read any memory bank of the tag and also tags of the Autoid System of Acura.
Pattern	**None**

Values	**None**
	        No memory bank read operation.
	autoid
	        Reads acura's Autoid System tags.
	reserved:wp:wl
	        Reads the Reserved memory bank of the tag.
	epc:wp:wl
	        Reads the Epc memory bank from the tag.
	tid:wp:wl
	        Reads the Tid memory bank of the tag.
	user:wp:wl
	        Reads the User memory bank of the tag.

Notes	**wp** - word pointer, indicates the initial position in word of the reading in the specified memory bank. **wl** - word length, indicates the number of words to read in the specified memory bank. Maximum value for wp and wl is 32. To read the entire memory bank, simply specify zero for wp and wl.
Example	To read all the contents of the User memory bank of the tag: **tagop user:0:0<CR>** **ok - parameter has been changed.<CR>** To read four words from the Tid memory bank from the second word: **tagop tid:2:4<CR>** **ok - parameter has been changed.<CR>** To not read any specific tag bank, just the result of the EPC: **tagop none<CR>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

> **NOTE:**   To read the entire memory bank, simply specify zero **for wp and wl**.

8.5.4.14 epcdecode - EPC id decoding of the tag

Command	**epcdecode** *value*
Function	Check (no value) or change the decryption type for the EPCID reading of the tag.
Description	The tag to be read may be encoded for a particular application, for example, for checkout applications the tag will be encoded in the GS1 standard, for maintenance applications the tag may be encoded in ASCII. Tag EPCID decoding returns a readable result for each application.
Pattern	**None**

Values	**None**
	        No decoding is performed and the epcid result of the tag is reported in hexadecimal.
	GS1string
	        Standard GS1 that returns the "GS1 element string".
	GS1epcuri
	        GS1 pattern that returns the "EPC Tag URI".
	GS1EPCPUREURI
	        Standard GS1 that returns the "EPC Pure Identity URI".
	GS1Gtin13
	        Standard GS1 that returns code GTIN13 without the serial number.
	GS1Sgtin13
	        Standard GS1 that returns code GTIN13 + serial number.
	ASCII
	        Decodes the EPCID of the tag according to the Ascii table.
	Wiegand26
	        Decodes the last 3 bytes of tag EPCID in Sitecode + ID.
	magstripe
	        Decodes the last 4 bytes of the EPCID of the tag in decimal.

Notes	For the GS1 standard, the sgtin-96 and sgtin-198 schemes are supported. The wiegand26 and magstripe values can be used in conjunction with the autoid type tagop.
Example	To read the GTIN13 + serial number of a tag encoded in the GS1 standard: **epcdecode gs1sgtin13<CR>** **ok - parameter has been changed.<CR>** To read the GS1 element string of a tag encoded in the GS1 pattern: **epcdecode gs1string<CR>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

8.5.4.15 autoidcs - Acura Autoid System Security Code

Command	**autoidcs** *value*
Function	Check (no value) or change the value of the Acura Autoid System Security Code.
Description	It's a six-digit number
Pattern	**900001**
Values	**000001 .. 999999**
Notes	This command should only be used if tagop is autoid.
Example	To check the autoidcs: **autoidcs<CR>** **900001<CR>** To change the Autoid security code to 900158: **autoidcs 900158<CR>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

NOTE: This command is used only if the application is to read the Tags of the Autoid System of Acura, tagop configured as autoid.

NOTE: The value of the security code is printed on the tag, please refer to 8.6.2. Registrator for acura's Autoid System.

8.5.4.16 separator - Result string separation character

Command	**separator** *value*
Function	Check (no value) or change the separation character of the fields of the reading result of a tag.
Description	It is a character that classifts the fields in the string from the tag read result.
Pattern	**' ' (space)**

Values	**space**
	        To have a space as a separator.
	Char
	        Any readable character in the Ascii table.

Notes	For space, you must write space as a value for the command.
Example	To check the separator: **separator\<CR\>** **space\<CR\>** To change the tab to '|': **separator |\<CR\>** **ok - parameter has been changed.\<CR\>**
Errors	When the value reported is not among the specified: **error - value out of range\<CR\>** When more than one value is passed to the command: **error - too many args.\<CR\>**

NOTE: The picking character in conjunction with the end of the line should be used to split the result string by the Software that will receive this string.

8.5.4.17 endofline - End of line of result string

Command	**endofline** *value*
Function	Check (no value) or change the end-of-line character.
Description	It is an unreadable character that is at the end of the result string and represents the end of the line for the Operating System.
Pattern	**Windows**

Values	**windows:qt**
	    Standard Windows end-of-line <CR><LF>.
	unix:qt
	    Standard Unix end-of-line <LF>.
	Macintosh:QT
	    Mac<CR standard end<>.
	Tab:QT
	    End of line will be the Tab tab character.
	None
	    No end of line in the result string.

Notes	**qt** - Amount of end-of-line repetition, values 1 through 10. If **:qt is** omitted, it is considered only an end of line.
Example	To check which line end is configured: **endofline<CR>** **windows<CR>** To change the end of line to Tab and repeat 3 times: **endoline tab:3<CR>** **ok - parameter has been changed.<CR>** To change the end of line to windows and repeat 5 times: **endoline windows:5<CR>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

NOTE: The picking character in conjunction with the end of the line should be used to split the result string by the Software that will receive this string.

8.5.4.18 keystrokespeed - Keyboard emulation typing speed

Command	**keystrokespeed** *value*
Function	Check (no value) or change the typing speed of the keyboard emulation.
Description	The typing speed in keyboard emulation, when readmode is as hid or hidserial, can be changed to three different values.
Pattern	**Fast**

Values	**Fast**
	        Quick typing, period between typing T=4ms.
	Medium
	        Average typing, period between typing T=32ms.
	slow
	        Slow typing, period between typing T=64ms.

Example	To check the typing speed: **keystrokespeed<CR>** **fast<CR>** To change the speed to slow: **slow<CR keystrokespeed>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

> This value should be changed only if the Host will receive the result string via
> **NOTE:**   Keyboard emulation presents problems due to the typing speed exercised by
> the reader.

`

8.5.4.2 echochar- Turns the character echo on or off

Command	**echochar** *value*
Function	Check (no value) or turn echo on/off.
Description	Enables or disables the echo of each byte sent by serial communication in Standalone mode.
Pattern	**on**
Values	**on** **off**
Notes	When set to off, typing in a "Putty" terminal will not appear if the character echo is not configured in the terminal software.
Example	To check the value: **echochar<CR>** **off<CR>** To disable RF and tag reading: **echochar off<CR>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

8.5.4.19 rssifilter - Reading filter by RSSI

Command	**rssifilter** *value*
Function	Check (no value) or change the cutoff value of the RSSI filter for reading.
Description	The signal level of the tag indicates whether the tag is near or away from the antenna, when closer to the antenna, the stronger the RSSI (closer to zero) and vice versa. If a value is specified for the filter, only tags with RSSI equal to or greater than that specified will be reported by the reader. The signal level of the tag is measured in dBm and is a negative value, i.e. the power in Watts is always less than 1mW.
Pattern	**0**
Values	**0..99**
Notes	To disable the filter, write the value 0. The specified value should not contain the minus sign.
Example	To check the filter value: **rssifilter<CR>** **0<CR>** To change the filter value to 45: **45<CR rssifilter>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

> **NOTE:** The filter by the RSSI value of the tag is used to restrict unwanted reading areas around the reader. It should always be used in conjunction with the reading power.

`

8.5.4.20 reportrssi - Reports the RSSI value of the read tag

Command	**value reportrssi**
Function	Check (no value) or change the sending of the RSSI value of the read tag.
Description	Enables or disables sending the RSSI value of the tag read in the result string.
Pattern	**Off**
Values	**on** **off**
Notes	The rssi value is reported with the negative sign '-'.
Example	To check the value: **reportrssi<CR>** **off<CR>** To change the value to have the RSSI value in the result string: **reportrssi on<CR>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

8.5.4.21 reportreadcount - Reports the amount of tag reading

Command	**reportreadcount** *value*
Function	Check (no value) or change the sending of the read amount of the tag.
Description	Enables or disables sending the read amount of the tag in the "rfon" period in the result string.
Pattern	**Off**
Values	**on** **off**
Notes	The maximum value of the read amount is 255.
Example	To check the value: **reportreadcount<CR>** **off<CR>** To change the value to have the amount of reading in the result string: **reportreadcount on<CR>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

`

8.5.4.22 enablebeep - Sound warning

Command	**enablebeep** *value*
Function	Check (no value) or change the beep with each reading of the tag.
Description	Enables or disables beeping with each tag reading.
Pattern	**on**
Values	**on** **off**
Example	To check the value: **enablebeep<CR>** **off<CR>** To turn off the tag read warning sound: **enablebeep off<CR>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

8.5.4.23 enablecapslock - Shortcut to change read mode

Command	**enablecapslock** *value*
Function	Check (no value) or change the use of the shortcut to change read mode.
Description	Enables or disables the use of the "CapsLock" 5x quick-type key, to switch from Standalone to Transparent mode and vice versa.
Pattern	**Off**
Values	**on** **off**
Notes	This functionality works only and Windows and Linux operating systems. Alternative to exit transparent mode is to send the command "**$ off<CR>**" to the reader.
Example	To check the value: **enablecapslock<CR>** **off<CR>** To change the value to enable shortcut use: **enablecapslock on<CR>** **ok - parameter has been changed.<CR>**
Errors	When the value reported is not among the specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

NOTE: For more details on changing the reader's operating mode please refer To 8.4 Switching standalone mode to Transparent and vice versa.

`

8.5.4.24 help - Help on commands
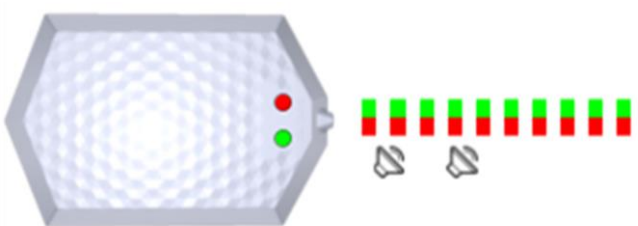
Command	**help** *value*
Function	Lists all commands (no value) or shows detailed help to the last command.
Description	For detailed help on a particular command just type: "help command<CR>".
Example	To check the commands accepted by the reader: **help<CR>** **command list... <CR>** For detailed help on the tagop command: **help tagop<CR>** **Command:tagop arg<CR>** **Description: Get (no arguments) or set the read tag operation.<CR>** **Arguments: reserved:wp:wl Reserved bank.*<CR>** ** epc:wp:wlEPC bank.*<CR>** ** tid:wp:wlTID bank.*<CR>** ** user:wp:wlUser bank.*<CR>** ** autoidAutoid tags.<CR>** ** noneNo tagop.<CR>** ** * wp - word pointer, wl - word length, max 32, menbank:0:0 entire bank.** **Default: none<CR>**
Errors	When the value reported is not among the specified: **error - command not found<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

8.5.4.25 summary - Summary of the values configured in the reader

Command	summary
Function	Lists all the values configured in the reader.
Errors	When a value is passed to the command: **error - too many args.<CR>**
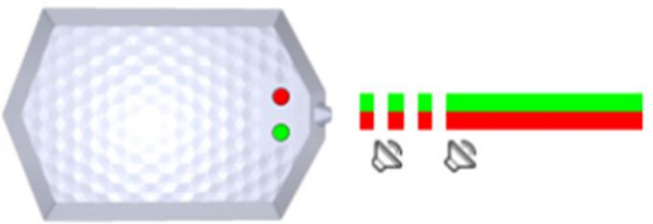
8.5.4.26 facreset - Factory Reset

Command	facreset
Function	Returns all values to the factory default.
Errors	When a value is passed to the command: **error - too many args.<CR>**

	Beep and green and red LED flash fast simultaneously for 5s to warn you that you are in the process of factory reset.

8.5.4.27 upgradefirmware - Firmware update

Command	**upgradefirmware**
Function	Starts the reader firmware update process.
Example	To check the commands accepted by the reader: **help<CR>** **command list... <CR>** **>upgradefirmware** !!! Bootloader will be executed !!! The reader will be restarted.
Errors	When a value is passed to the command: **error - too many args.<CR>**

Beep and Green and Red LED flash fast simultaneously 3 times and then the green and red LED remain lit continuously to warn you that it is in the process of updating firmware.

8.6 Switching Standalone mode to Transparent and vice versa

For changing the mode of operation of the reader the following options can be used:
- A. Use the **readmode command**, examples:
    - a. The reader is in Standalone mode and the "**readmode transparent\<CR>**" command is sent, so the reader enters Transparent operation mode.
    - b. The reader is in transparent mode and the **$off\<CR> command** is sent, so the reader enters standalone mode with hid interface.

> **NOTE: When**   in transparent mode, the reader does not echo the characters typed in the

- B. Use the "Caps Lock" key (Windows and Linux only), example:
    - a. Close any serial communication with the reader;
    - b. To switch between one operating mode and another, simply *quickly click the CapsLock key 5 times in sequence* and check the reader's LEDs to see the change.

⚠	**ALERT**	**The Caps Lock shortcut for switching the operating** mode only **works if and only if the serial port of the reader is not connected (not being used by any Software).**

`

8.7 Same tag read rate

The read rate of the same tag must be manipulated through the following commands:
- **Gen2session**
- **gen2target**
- **rfon**
- **rfoff**

The commands mentioned above change the parameters of the UHF EPC Gen2 protocol and have an effect on communication between the tag and reader.

Combinations of values for the Session and Target of the tag reflect how often a tag that has already responded to an inventory will respond again, in other words, how fast a tag is reassing.

Examples:

A. I would only read only once the EPC id of a tag and that same EPC id does not repeat itself while the tag is in the reading field of the reader. Configure as follows: **gen2session s2<CR>**

   gen2target a<CR>

B. I would like to read the same tag again and again while it is in the reader reading field. Set up as follows:

   gen2session s0<CR> gen2target ab<CR>

   In this case, what will determine the read repetition rate are the **rfon and rfoff commands**, i.e. the sum of the times will dictate the repetition rate. The highest possible read repetition rate is every 50ms:

   rfon 50<CR>
   rfoff 0<CR>

⚠️ ALERT	**The tag read rate can be much faster than the rate of sending the results by keyboard emulation (hid), that is, if the chosen read rate is high and the chosen interface is keyboard emulation, the reader will store the reading result and will be sending it even after the tag is removed from the reading field.**

8.8 Increasing rejection of side readings
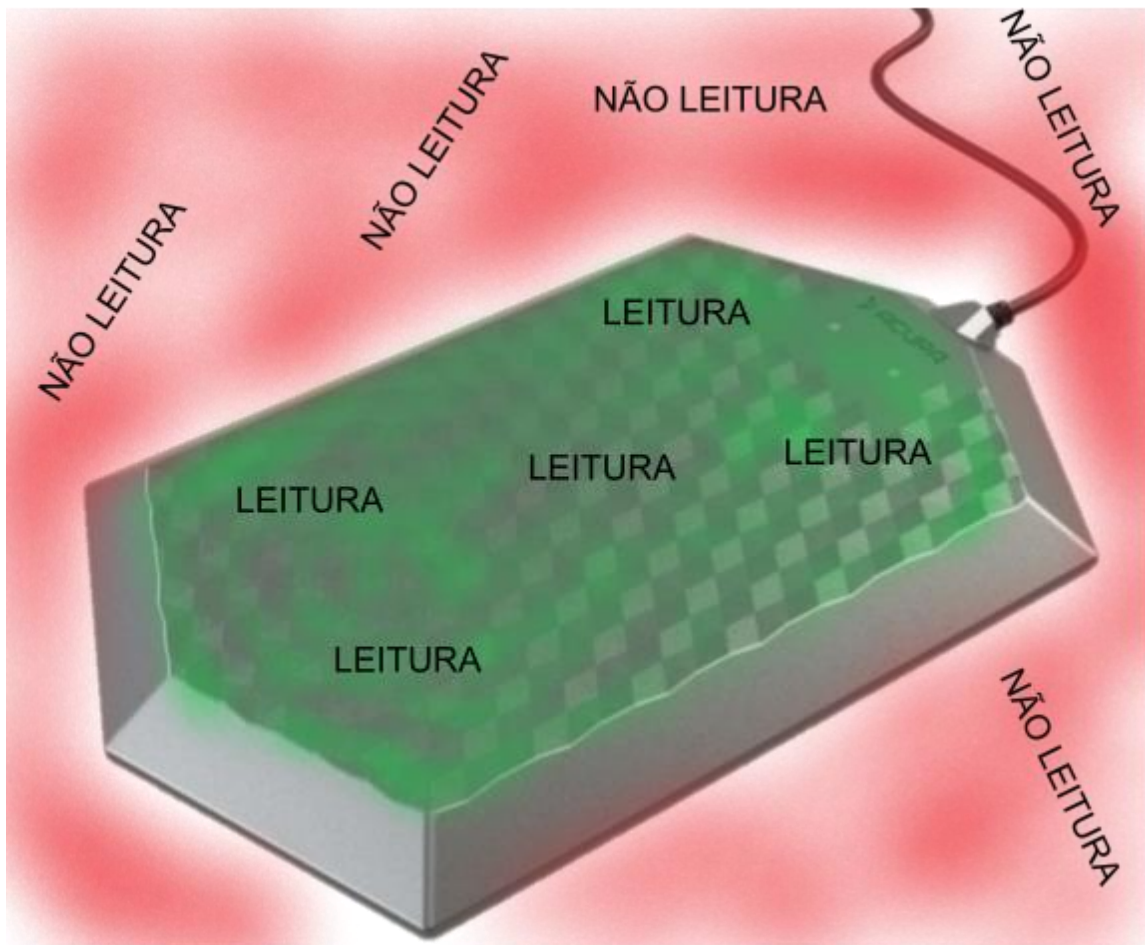
To obtain a reading field as close as possible to the figure above, the RSSI reading filter should be used in conjunction with the reading power setting. With these adjustment plus the integrated near field antenna the rejection of side readings greatly increases.

8.8.1 How to measure the RSSI cutoff value?

The cut-off value of the tag reading signal level should be obtained in a practical manner, that is, with the tags that will be used in the application and reading tests checking the rssi value reported on the sides and on top of the reader.

> To obtain the RSSI value of each read you take, use the "**reportrssi on"** command.
> **NOTE:** To configure the reader to report readings only above an RSSI value, use the command "**rssifilter xx**", where xx is the desired value.

> **NOTE:** In order to facilitate the execution of rssi level readings, the repetition rate of reading of the Tags can be increased as described in "8.5 Reading rate of the same tag"

8.9 Examples of use (Standalone mode)

Below are some examples of using and applying the reader with the details of the configuration commands.

8.9.1 Checkout - Standard GS1/EAN tags

For applications that use UHF EPC Gen2 tags encoded according to *the GS1 EPC Tag Data Standards (TDS) 1.11* standard, the HEXAPAD reader can be used and supports the following tag encodings:

- **Sgtin-96**
- **Sgtin-198**

The results of reading the encoded tag can be sent in the following formats:

- **GS1 element string**, example "(01)03608449920322(21)1540341";
- **EPC Tag URI**, example "urn:epc:tag:sgtin-96:1.360844.0992032.1540341";
- **EPC Pure Identity URI**, example "urn:epc:id:sgtin:360844.0992032.1540341";
- **GTIN13 without the serial number**, example "3608449920322";
- **GTIN13 + serial number**, example "3608449920322 1540341";

8.9.1.1 Read tag encoded in SGTIN-96/198 and show result in GS1 element string format

Configure the reader as follows:

epcdecode gs1string<CR>
(to decode sgtin-96 or sgtin-198 and send in GS1 element string format)

8.9.1.2 Read tag encoded in SGTIN-96/198 and show result in EPC Tag URI format

Configure the reader as follows:

epcdecode gs1epcuri<CR>
(to decode sgtin-96 or sgtin-198 and send in EPC Tag URI format)

For the other formats just use the **epcdecode command** with the equivalent format that can be checked in 8.3.16 epcdecode - EPC id decoding of the tag.

⚠ **ALERT**	**The keyboard /language layout configured in the Operating System can change the ":" character of the result string of the epcdoce type gs1epcuri and gs1pureepcuri.**

8.9.2 Registrator for Acura's Autoid System

The HEXAPAD reader can be used to read the secure Acura tags of the Autoid System in vehicle identification applications.

The reader can be used as a tag registrant, facilitating and avoiding errors in the data entry of the site access control system.

The following settings must be made on the reader:

- Configure the reader with the same Security Code (CS) that is written to the tag:
    - **autoidcs 000818<CR>**
- To read an Autoid tag in wiegand26 format, set:
    - **wiegand26<CR epcdecode>**
- Set up read operation mode:
    - **autoid <CR tagop>**
- Configure the reader to emulate keyboard:
    - **hid<CR readmode>**

Below is an image of a secure tag tag from Acura's Autoid System:
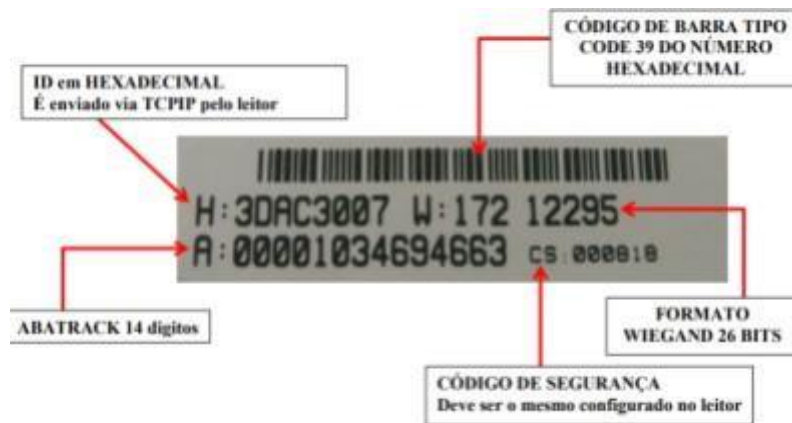
Reader settings:

8.9.3 Read TID memory bank and EPC tag id

With the HEXAPAD reader it is possible to read any memory bank of the EPC gen2 tag, (reserved, epc, tid and user), just configure by **the tagop** command in which database is desired to read along with the initial position and the size of the data (in word) that you want to read.

The syntax of the **tagop** command is explained in detail in . Examples:

   A. Read the entire TID bank of the tag plus the EPC id in hexa:

              tagop

      tid:0:0<CR>

      epcdecode none<CR>

      Example of reading two tags with different sizes from the TID bank:
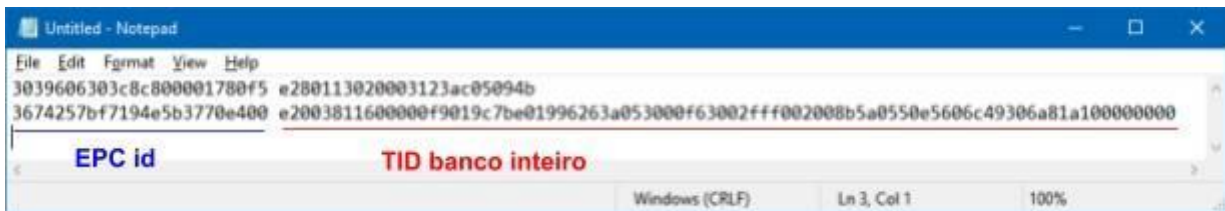
   B. Read from the TID bank 4 words skipping the first 2 words and decoding the EPC id by the ASCII table:

      tagop tid:2:4<CR>

      epcdecode ascii<CR>

      Example of reading two tags:

In addition to TID banking, RESERVED, USER, and EPC banks can also be read in conjunction with the EPC tag id.

> **NOTE:** The count unit in the EPC Gen2 standard is word, which is equivalent to 2 bytes or 16 bits.

> **NOTE:** For reserved bank, if the tag has a recorded access password other than 0x00000000, the tag password must be configured on the reader by the **gen2accesspassword**.

> **NOTE:** More details of commands in 8.3 Configuration commands.

Part 3 - Transparent Mode

Reader in Transparent mode

Programming and development of Software

Software demo - Ura

Controlling Led and Beep

`

9 Reader in Transparent mode

The reader in Transparent mode starts to respond through the serial to a Software developed using the Mercury API, which is a library package for C#/.Net, Java and C languages, with this development package it is possible to develop software that can have full control of the reader for reading and writing UHF EPC Gen2 tags (ISO 18000-6C) and UHF EPC Gen2 V2 (ISO 18000-63).

Visual feedback from the reader when in Transparent mode:

| | **Green LED flashes slowly and Red LED turned off** |
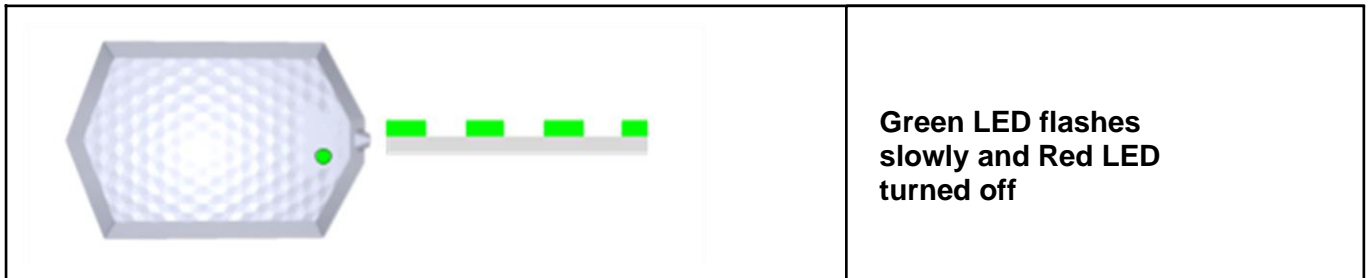
As a factory default, the mode of operation of the reader is standalone mode, to change to Transparent mode please refer:
8.4 Switching Standalone mode to Transparent and vice versa;
8.3.4 readmode - Read mode
The red LED and the beep of the reader are now controlled by the User Software by manipulating two GPIOs available in MercuryAPI, plus details in 9.2.2. Controlling the Led and Beep.
Communication between the User Software and the reader is serial through the port created by the Operating System at the time the reader is connected to the USB, to identify such port, please consult 7.1 Connecting the reader.

> **NOTE: In**   Transparent mode, the HID keyboard-type interface is not used by the reader.

9.1 Software Programming and Development

For the development of a control software for the reader it is necessary to use the MercuryAPI available in the languages C#/.Net, Java and C, with this development package it is possible to develop software that can control the reader in its communication interface.
The most current and recommended version for development can be found at:
https://www.jadaktech.com/products/thingmagic-rfid/thingmagic-mercury-api/

In the API file package, there are several examples and source code in all languages. This development package has no additional cost and is open source.

Full API documentation: https://www.jadaktech.com/products/thingmagic-rfid/thingmagic-mercury-api/

9.2 Software demo - URA

The reader in Transparent mode can be tested using Jadak's "Universal Reader Assistant" software, the software download, and its manual are available at the links below:

User guide:
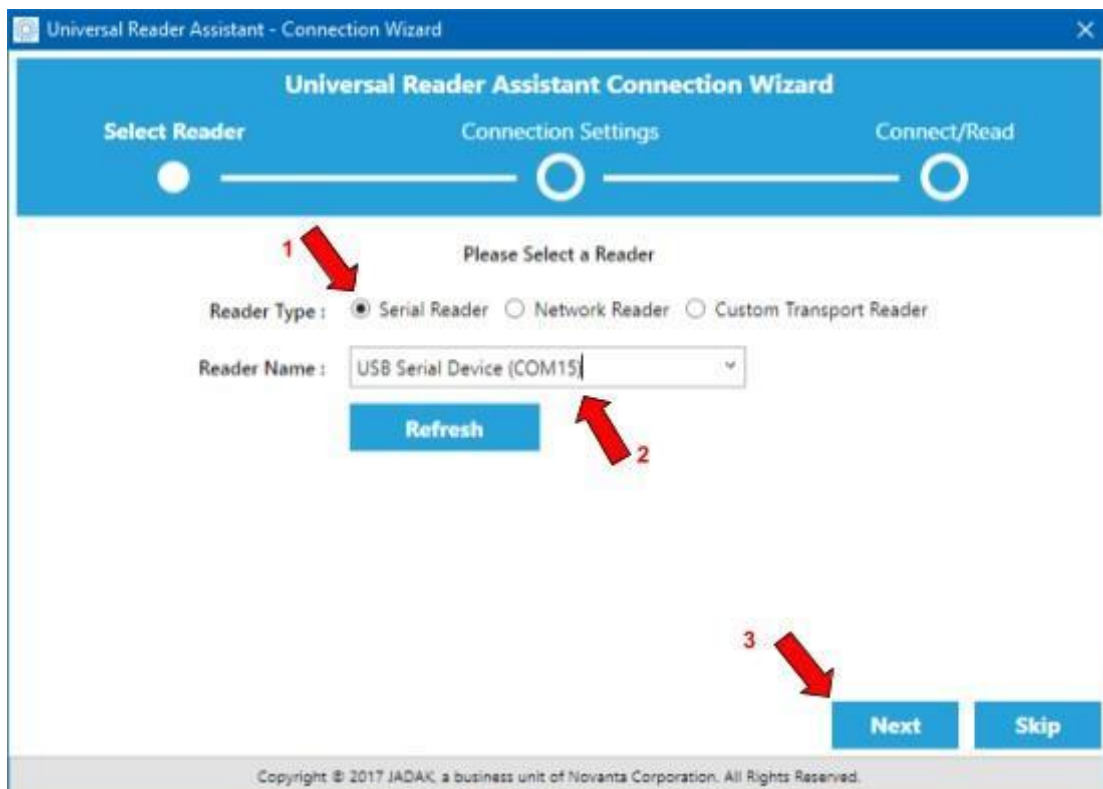https://www.jadaktech.com/products/thingmagic-rfid/thingmagic-universal-reader-assistant/

URA Software:
https://www.jadaktech.com/products/thingmagic-rfid/thingmagic-universal-reader-assistant/
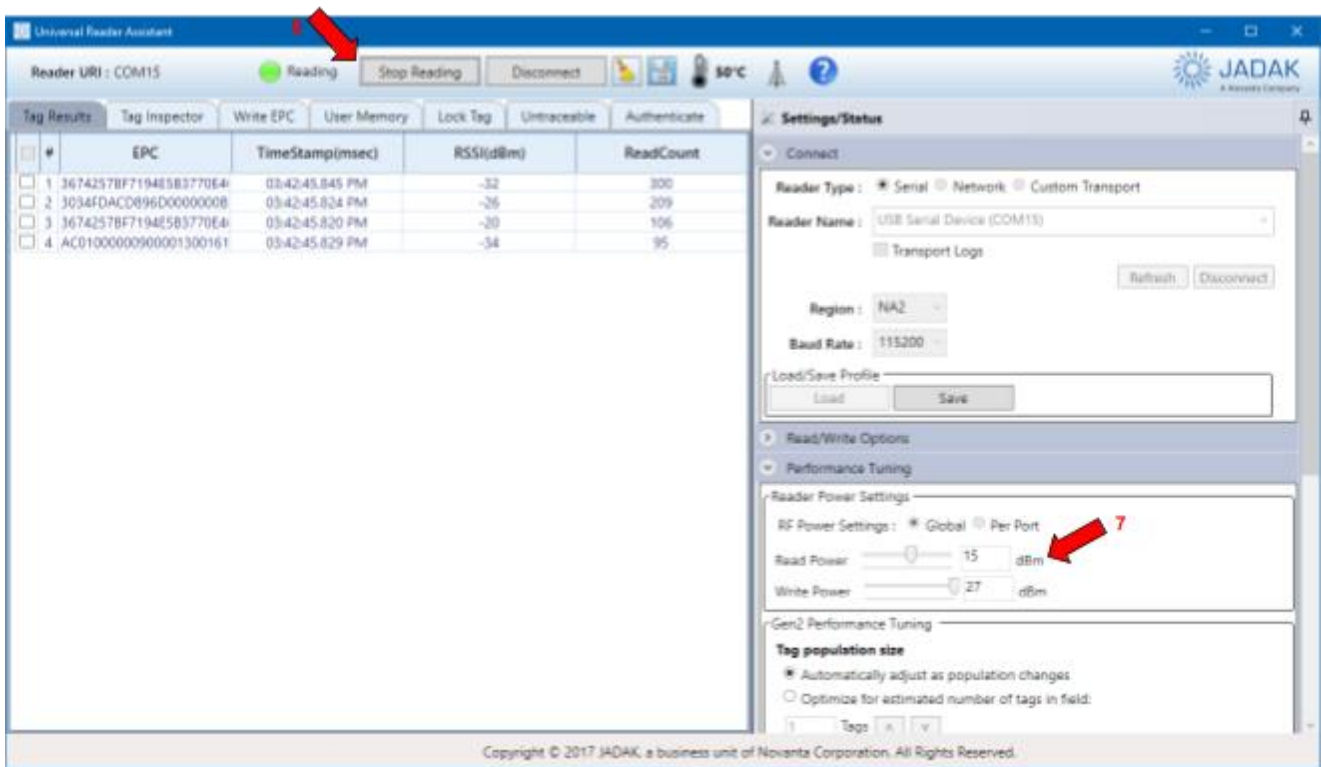
9.2.1 Instructions for Demo Software - IVR

NOTE: Make sure the reader is in Transparent mode. See 9 Reader in Transparent mode.

Run the "Universal Reader Assistant".

Once connected, lower the read power to 15dBm if the reader power is only from a USB port and then read.

9.2.2 Controlling the Led and Beep

In Transparent mode, the control of the red Led and beep must be done by the User Software using MercuryAPI.
Below is the usage map for the respective GPIOs:

Output #	Type feedback	Drive (On)
GPIO1	Red Led	High
GPIO2	Beep	High

Below is image using ivr to manipulate the red LED and the beep of the reader in Transparent mode.
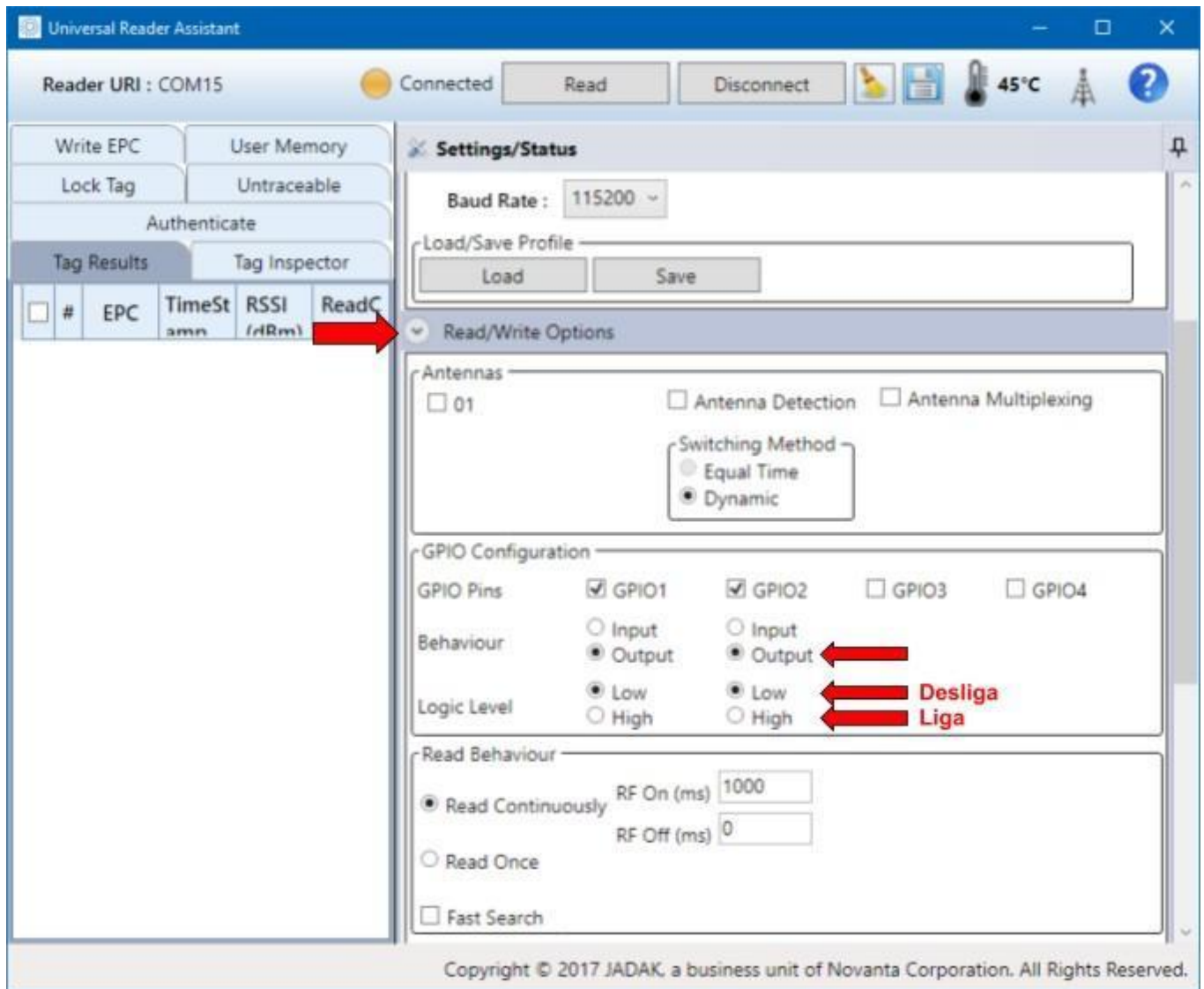
NOTE: In the MercuryAPI file package for all languages there is an example for manipulating the Reader's GPIO called "**Gpiocommands**".

10 Part 4 - Update the firmware

How to update the firmware of the reader? (Windows)

11 How to update the firmware of the reader? (Windows)

11.1 Update firmware by serial port

This procedure requires attention and prior knowledge of some features of the reader, which are found in the manual, so if necessary return in the manual according to the links indicated to consult questions. Before starting the procedure make sure of some specifications:

- If you have installed on your machine the software "**Putty**". If you do not already have it, please refer to 8.2.1.3 How to send the commands to the reader? (Windows) to continue;
- If the reader is in "Standalone **"** mode. For the mode in which the reader is in, please refer to 7.1. Turning on the reader;
- Which serial port the reader will respond to when it is connected on your computer. To find out which serial port the reader will respond to, please refer to 7.1.1. Identifying the serial port (Windows)."

11.1.1 Step one. Send the update command to the reader

1. Open putty;
2. Select "Connection type:" "Serial", "Serial line" port that the reader generated, and "Speed" 115200;
3. Click "Open" to open the terminal screen for typing commands;

4. Type "**help**" without argument to verify that the reader is connected correctly;

5. Type "*upgradefirmware*" without argument and the following screen should appear;

|	**Beep and green and red LED will flash fast simultaneously 3 times and then the green and red LED will remain lit continuously to warn you that you are in the process of updating firmware.**

NOTE: Currently the reader is in "*Firmware Update" mode*, so no

Disconnect the reader from the computer until the upgrade process is complete.

6. A message from "***Installing device...***" will appear, wait until the end of the process.

Windows 8 - OK
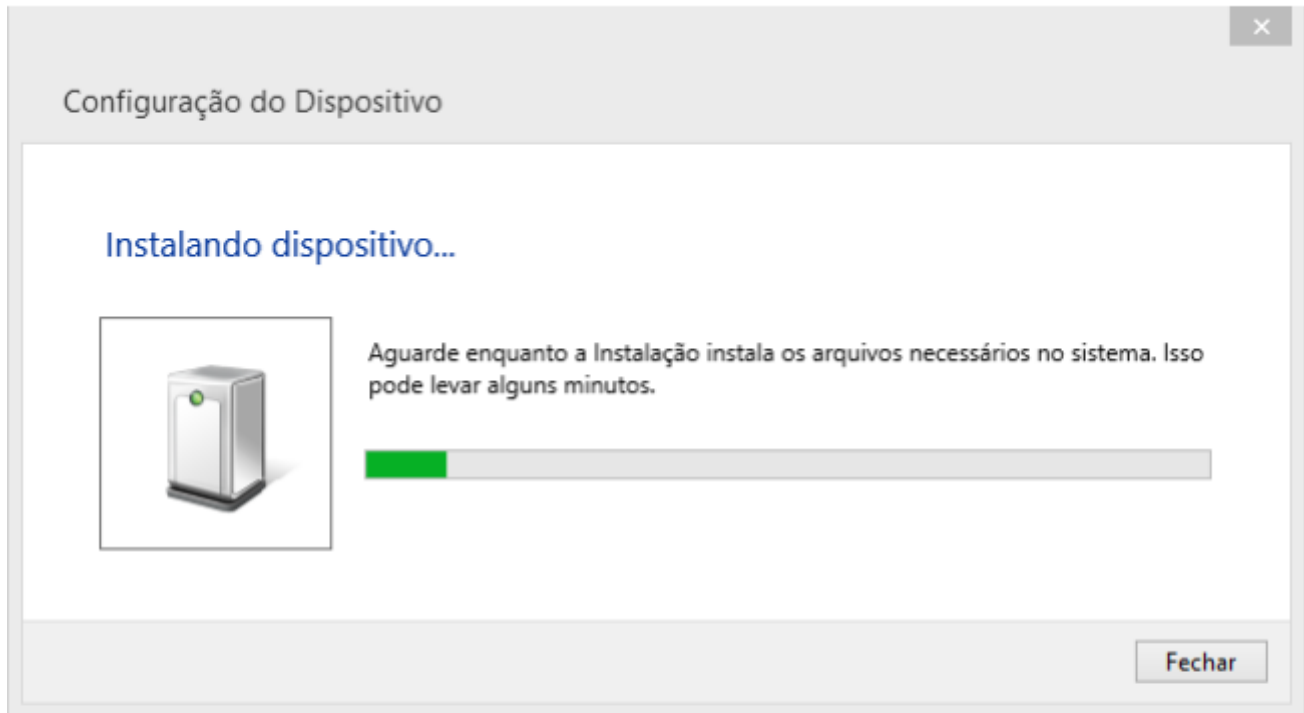
Windows 7 - OK
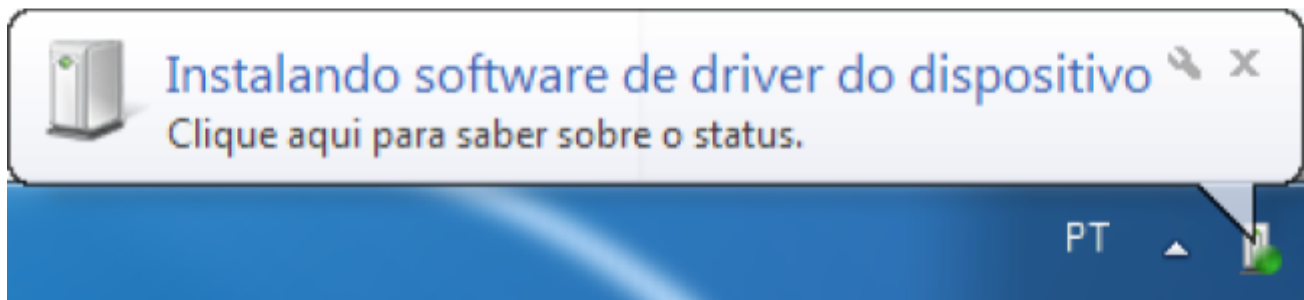
NOTE: If you use Windows 10 the driver will be installed automatically.

7. Make sure the driver has been successfully installed, go to the "Control Panel" or with the "Windows + X" key open *the "Device Manager*". In "*Ports (COM and LPT)*", a virtual serial communication port "*Bossa Program Port (COM...)* " should appear.
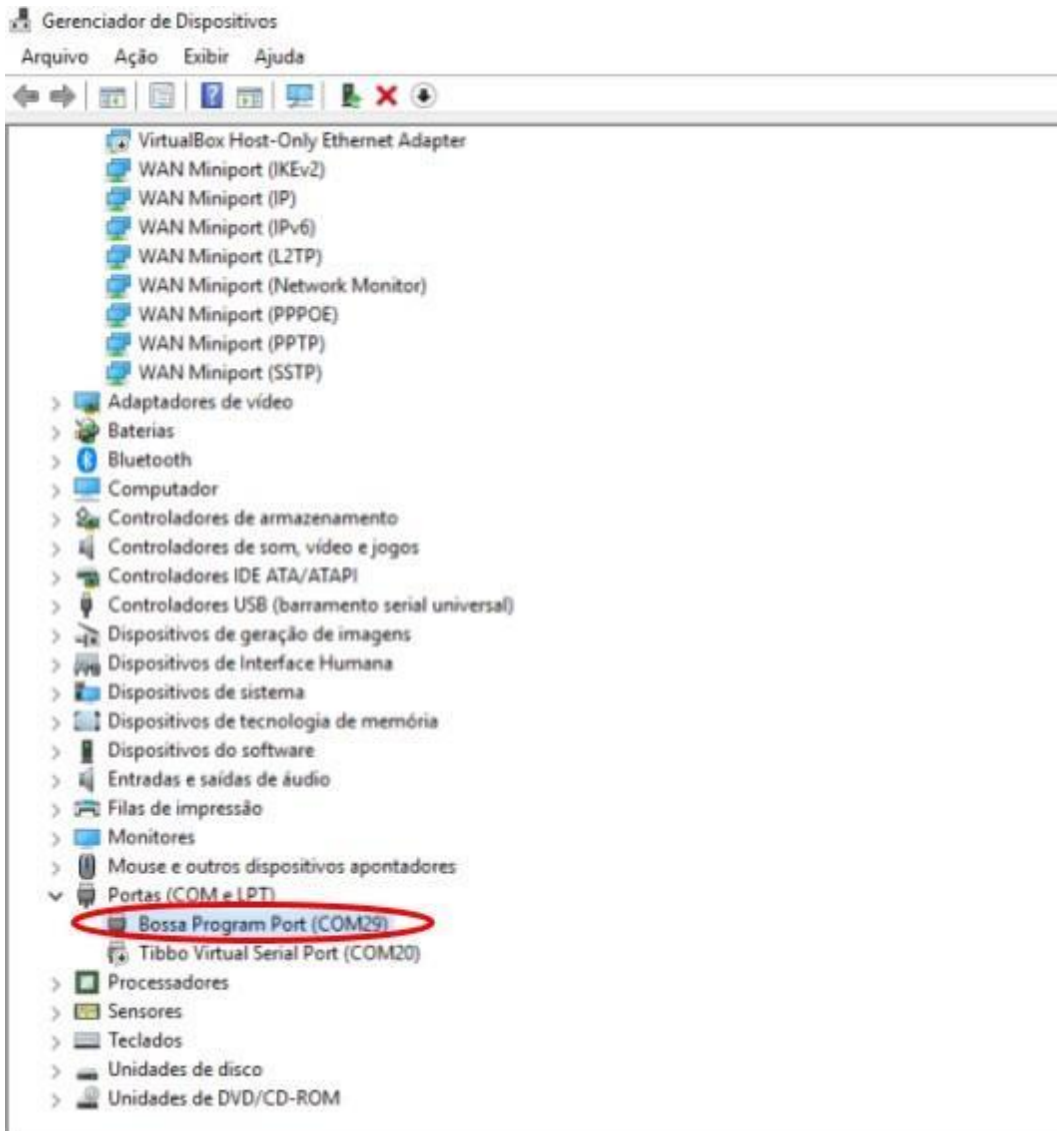
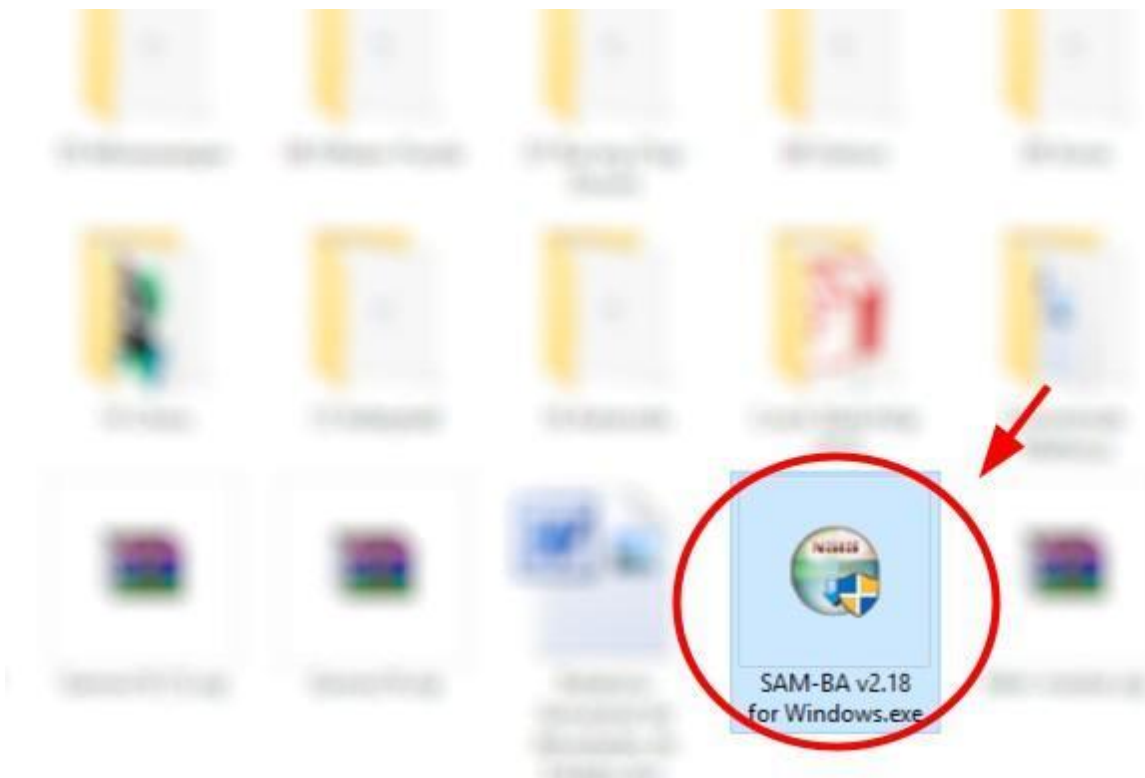## 11.1.2 Step two. Installation of the software "Acura Sam-ba" (Windows)

This procedure requires attention and is divided into two steps, the execution of the software installer "**SAM-BA v2.18 for Windows.exe**" and the unpacking of the folder that will be used in the firmware installation process.

### 11.1.2.1 Step 1. Software installation

1. Download the software installer from this link:
https://drive.google.com/file/d/1JpiUY_6CKPEYa-L-Bx0rnE1xzz1Uh-1U/view

2. Go to downloads, or where the file was saved and run the installer. A message of "Do you want to allow this app from an unknown vendor to make changes to your Windows?" message will appear, click "yes."

3. The software installation screen will appear, click "Next >", "I Agree", "Next >", "Next >", "Install", and wait for the software to install.
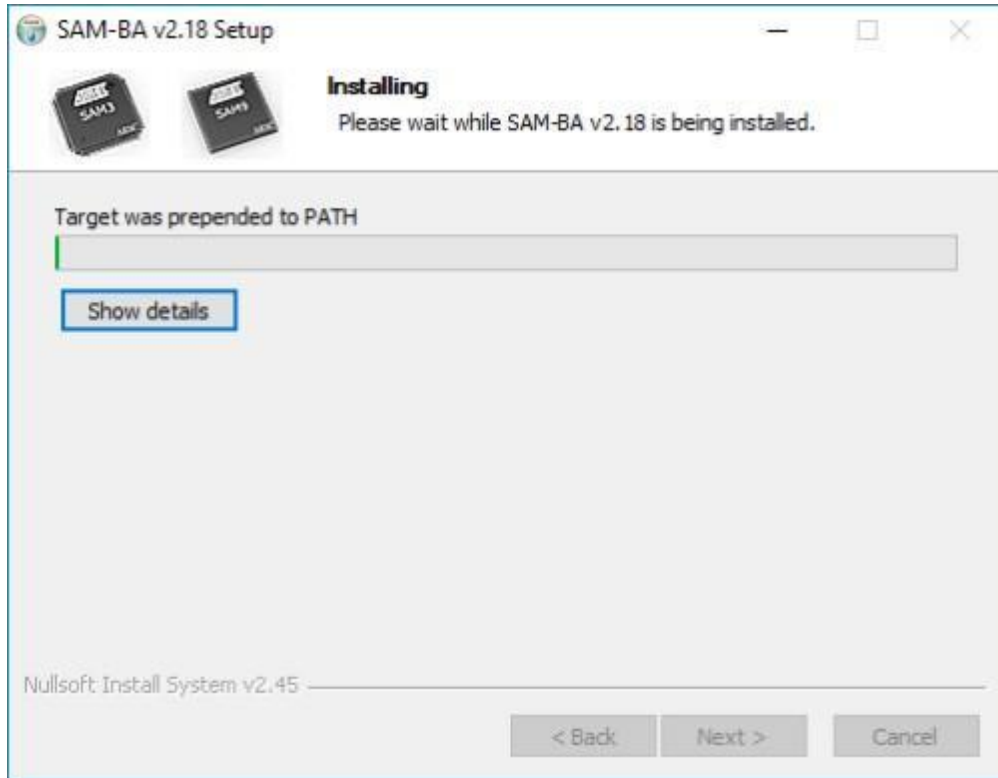
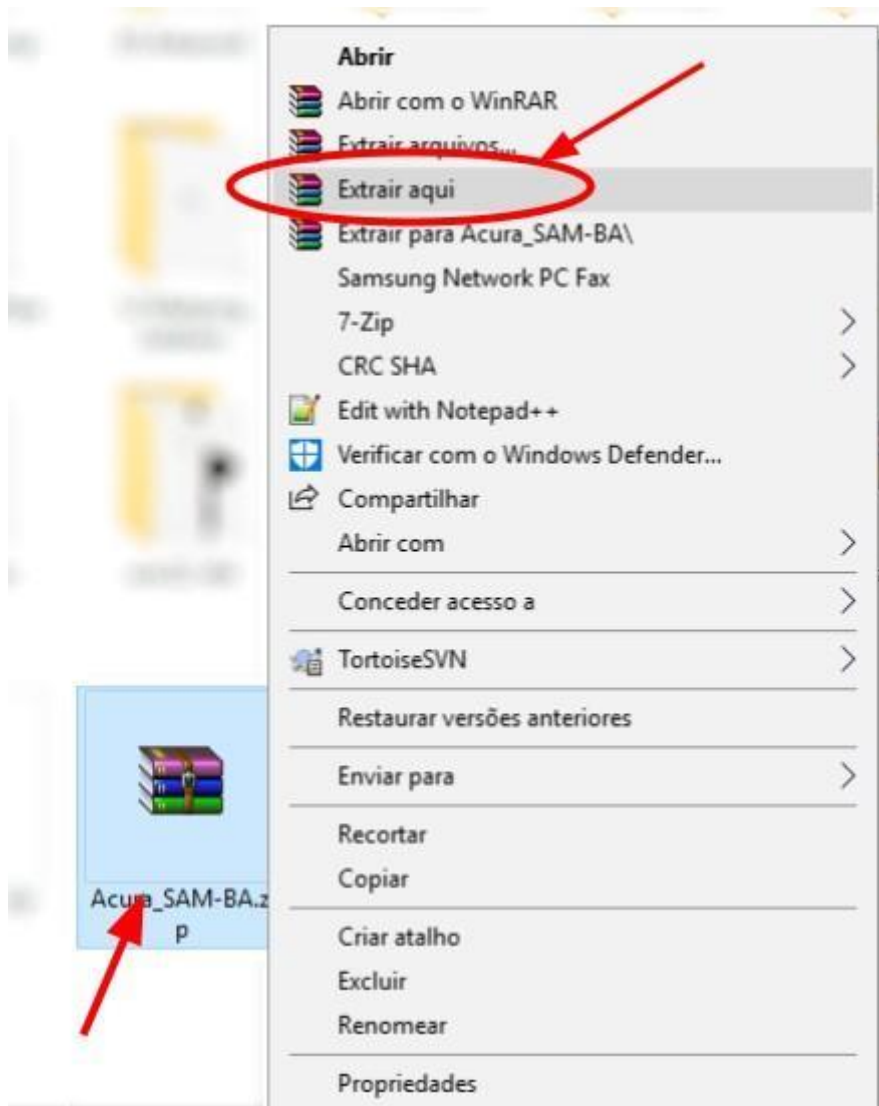4. After installation click "Next >", "Next >" and "Finish" to finish.

11.1.2.2 Step two. Unpack the software folder

1. Download the zipped folder of the software by the link:
https://drive.google.com/file/d/1aRQhdlHt2z8v4rRvrHB1G3Vc8Bkdm1We/view

2. Go to downloads, or where the file was saved, right-click, and click "***Extract here***".


⚠️ **WARNING: It is extremely important to perform the two steps carefully.**

11.1.3 Step three. Load the firmware

The software that will be used for the firmware update will always be what is inside the folder "**Acura-SAM-BA**" that was downloaded and unpacked in section 10.1.2.2 Step 2. Unpack the software folder.

1. Inside the uncompressed folder "**Acura_SAM-BA**" double-click on the software "**Acura_sam-ba.exe**";
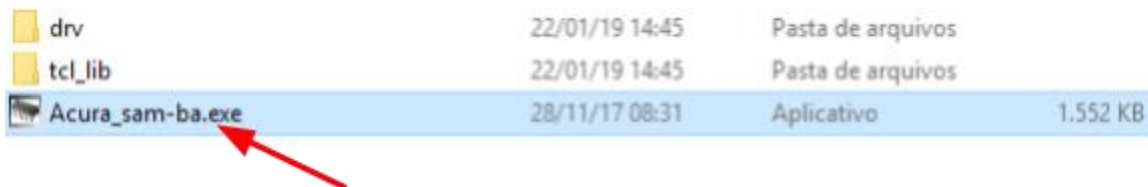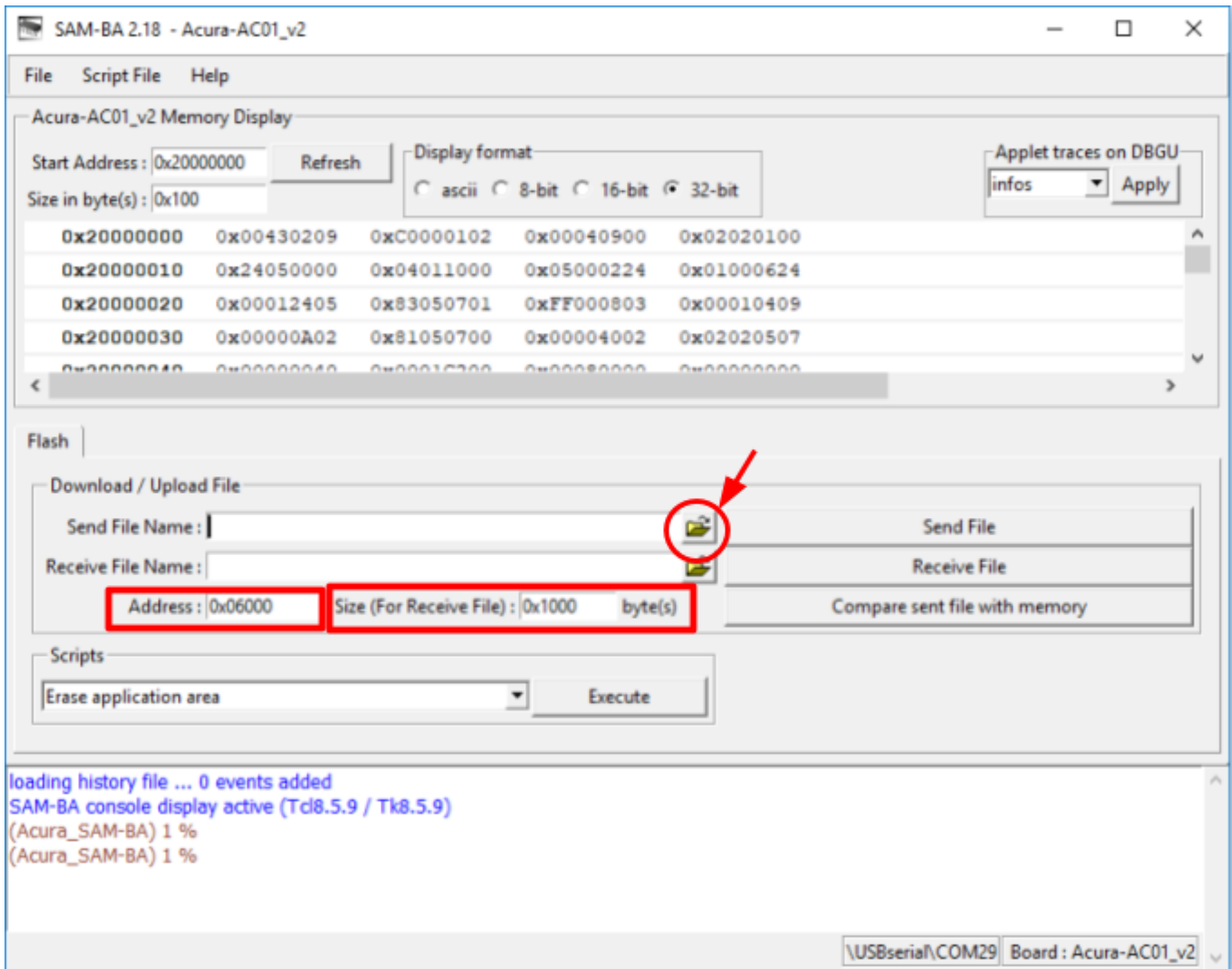

> **NOTE:** This software will only work if you have run and installed the software "**SAM-BA v2.18 for Windows.exe**" reported in section 10.1.2.1. Step 1. Installer execution.

2. Click "**Connect**";
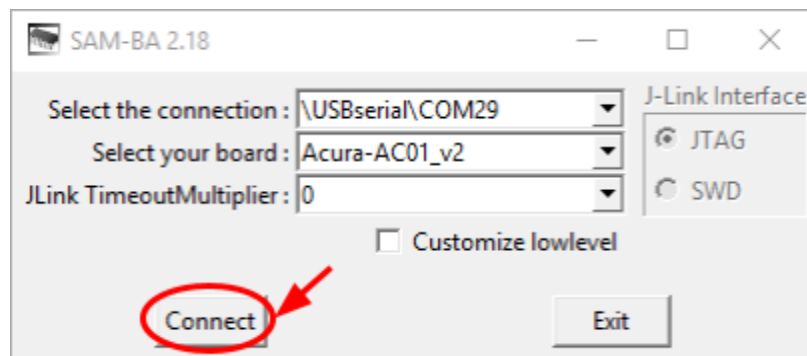
3. Click the "***Send File Name:***" icon and select the updated firmware file;

⚠ AVISO:	It is extremely important not to modify the parameters "*Address:*" "*0x06000*" and "*Size (For Receive File):*" "*0x1000*" before sending the firmware to the reader. Modifying these parameters will damage the player.

4. Make sure that **the parameters "Address:" "0x06000" and** "**Size (For Receive File):"**
    "0x1000" agree and send the updated firmware to the reader by clicking on "Send File" and
    wait for the procedure to complete 100%;
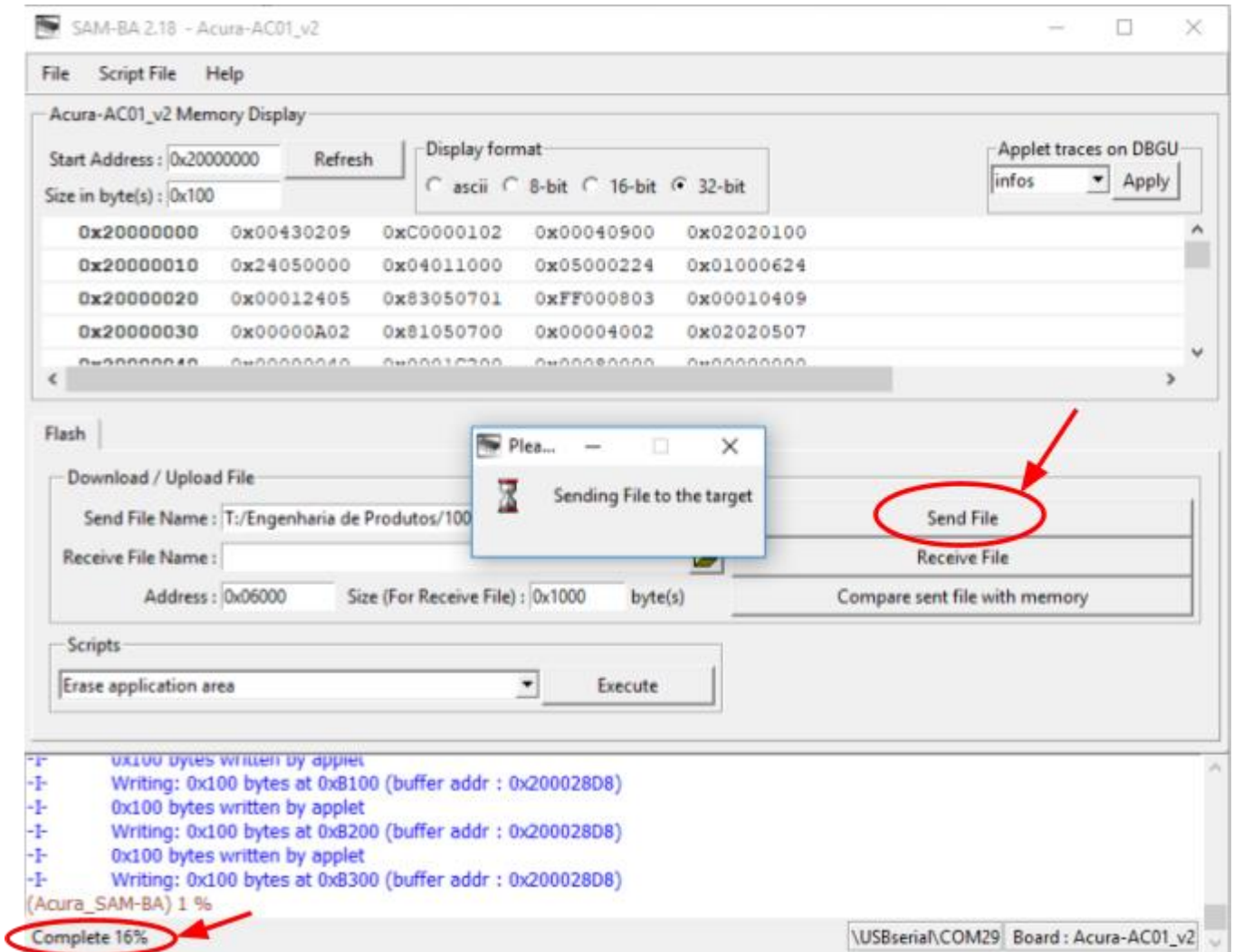
NOTE: It usually takes 10 seconds to update the firmware.

5. Disconnect and plug the reader back into the USB port and the reader will return to normal
    operation with the updated firmware.

Since the end of the 1990s, ACURA Global has been the pioneer in the radio frequency identification (RFID) market in Brazil and Latin America, and has successfully explored, since the late 1990s, its large-scale adoption in the most diverse sectors of the economy, from mining to steel, agriculture to food processing, from logistics to retail, from transport to the distribution chain, from access control to asset management. Promoter of new technologies, innovative, agile, and focused on the feasibility of cutting-edge projects.

Commercial and Development Office
Av. Antarctica, 381 14th floor
São Bernardo do Campo - SP
ZIP Code: 09726-150
11 3028-4600

FACTORY MG
Av. Nicolau Cesarino, 4197, Extrema - MG, 37640-000

www.ACURA.com.br