User Guide

AcuPad-50 Mux

USB

http://www.acura.com.br

BLANK

1 Products covered by this guide.

This guide contains information of the following products:

Model	Code	Details
AcuPad-50 Mux	100.659	*Inside the box with 2 SMA connectors.*

1.1 Approval note.

National Telecommunications Agency (ANATEL):

The Acupad-50 Mux, was tested and approved in accordance with the Regulationfor 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 Antennas for AcuPad-50 MUX.

Vendor	Model	Type	Polarization	Max Linear Gain (dBi)
Mti Wireless Edge	MT009S08LH/C	Patch	Circular	6.5
Mti Wireless Edge	MT-262006/TRH/A/K/F	Patch	Circular	7.0

Specification of the cable and connectors, used between the reader and antenna:

For MT009S08LH/C model:

Cable Length: 5m

SMA Male connector (antenna):

RP-SMA Male Connector (reader):

For MT-262006/TRH/A/K/F model:

Cable Length: 1.5m

RP-TNC Male (antenna):

RP-SMA Male Connector (reader):

Professional installation instruction

1. Installation personal
This product is designed for specific application and needs to be installed by a qualified personal who has RF and related rule knowledge. The general user shall not attempt to install or change the setting.

2. Installation location
The product shall be installed at a location where the radiating antenna can be kept 21cm from nearby person in normal operation condition to meet regulatory RF exposure requirement.

3. External antenna
Use only the antennas which have been approved by the applicant. The non-approved antenna(s) may produce unwanted spurious or excessive RF transmitting power which may lead to the violation of FCC limit and is prohibited.

4. Installation procedure
Please refer to user's manual for the detail.

5. Warning
Please carefully select the installation position and make sure that the final output power does not exceed the limit set force in relevant rules. The violation of the rule could lead to serious federal penalty.

1.2 Legal notice

While every effort has been made to ensure that this document and the information contained therein is 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 use of the information contained herein does not infringe any right, legitimacy or fitness for purpose, and therefore waives any liability, direct or indirect, for loss or damage related to the use of this document.
The information contained in this document is subject to change 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 guide provides detailed instructions for installing, connecting, configuring, operating, and troubleshooting the AcuPad-50 Mux. This guide is divided in four parts, **Part 1** with information for installationof the reader, **Part 2** with the operating information, reader working and configuration in the AutonomousMode, **Part 3** with the reader operating and working information in Transparent Mode and **Part 4** with the reader firmware upgrade information.

2.2 Document conventions

Throughout this document some important information should be highlighted using the following notes:

WARNING: **Important notices that require user attention.**

NOTE: Important information and tips regarding the subject.

2.3 Definition terms

The term “tag” used in this document refers to any RFID tag that follows the pattern “EPCglobal UHF Class 1 Generation 2”.
The term “EPC id” used in this document refers to the Tag’s id in the EPC memory bank.
The term “Pattern GS1” used in this document refers to the “EPC Tag Data Standard version 1.11” that is the GS1 tags coding pattern.
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 “Universal Serial Bus” USB communication.
The term “ASCII” stands for “American Standard Code for Information Interchange” and refers to a table with values for converting readable characters. The term “<CR>” used in this manual represents the “Carriage Return” of the ASCII table and the ENTER key on the keyboard when mentioned in commands.

2.4 Target audience

This document is intended for helping the professional in set up and install the AcuPad-50 Mux. Before attempting to install, configure, and operate this product, you should be familiar with the following terms:

- Device communication parameters including USB and Serial communications.
- Basic knowledge about network setting for Windows OS.
- Basic knowledge about RF antenna radiation and positioning.
- Basic knowledge about the UHF EPC Gen2 protocol.

3 Introduction to equipment

The reader AcuPad-50 Mux is a table or wall reader with UHF RFID technology (Radio Frequency Identification) (Ultra High Frequency) with short dimensions and flexible use, it has useful features and functionality for the solutions of tag registration, Checkout, etc. Below follows the main features:

- It can work in two different ways, Autonomous and Transparent:
  - **Autonomous mode:**
    - Read tags automatically;
    - Send the read results on the following interfaces;
      - Keyboard emulation (Native);
      - Serial communication;
      - Keyboard emulation + serial communication;
    - Many configuration parameters serial terminal;
      - Read power;
      - RF time;
      - Parameters Gen2, (tag >> reader) and (reader << tag);
      - Reader information;
      - Read operations;
      - Reader filter by the tag RSSI value;
    - Decodes tags GS1 SGTIN-96 and SGTIN-198 pattern for checkout applications, send 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”;
    - Read the PA tags (SJ5511 - Artifact), toll (only the AC-01M V2);
    - Decodes tags Autoid System pattern in the Acura for tag registration for vehicles identification application:
      - autoid + autoidcs
        - Wiegand26, example “172 13259”;
        - Magstripe, example “00001034695627”;
    - Decodes tags ASCII pattern;
    - Option of not decoding and sending the EPC value of the tag in hex;
    - Reports which antenna the tag was read;
    - In addition to reading and decoding the tag's EPC, it is possible to read any database tag memory;
  - **Transparent mode:**
    - The reader must be controlled by a Software that uses the MercuryAPI, an API for reading and writing tags in the following languages:
      - C, .Net (C#) and Java;
- Compatible with Windows 10, Mac and Linux with no driver needed, for Windows 7 and 8 or driver is provided for installation;
- It has reduced dimensions, with holes compatible with a 4x2” light box;
- Wide range of read power: 0 to 30dBm (using a Y extender cable for an additional power input).

4 Revision tables

4.1 Document revision table

Revision	Date mm/yyyy	Description
3	02/2023	• Format adjustment.
2	01/2020	• Added information about new firmware 1.1.3. ◦ New readtag command to enable and disable reading of tags (RF) in standalone mode.
1	06/2019	• Creation of this document.

4.2 Firmware revision table

Revision	Date mm/yyyy	Description
1.1.3	01/2020	• New "readtag" command with "on" and "off", to turn rf off and on again in standalone mode; • New Capslock combination (twice fast) to run the readtag command; • Audible warning on readtag execution; • Sound warning when changing readmode via CapsLock; • Tag read beep when readmode is serial and there is no open serial; • Improved transparent-standalone transition and vice versa aiming at better stability of the reader;
1.1.2	07/2019	• Added support for the Micro module; • Added commands to read PA, toll tags; • Added several commands regarding antenna detection; • Minor fixes and improved player stability.
1.0.6	03/2019	• Fixed internal communication timing error that generated reader instability if the RFO time was set to more than 500ms.
1.0.2	10/2018	• Firmware creation and release.

4.3 Hardware version table

Version	Date mm/yyyy	Description
V1	06/2022	• Creating the AcuPad-50 Mux

5 Installation notes and important warnings

WARNING:

The maximum reading distance varies depending on the antenna used in the reader. The effective reading distance of passive tags varies according to each installation and environment, and may change depending on:

- Material on which the tag is installed;
- Tag position;
- Reader direction in relation to tag;
- Electromagnetic interference caused by other equipment installed nearby.

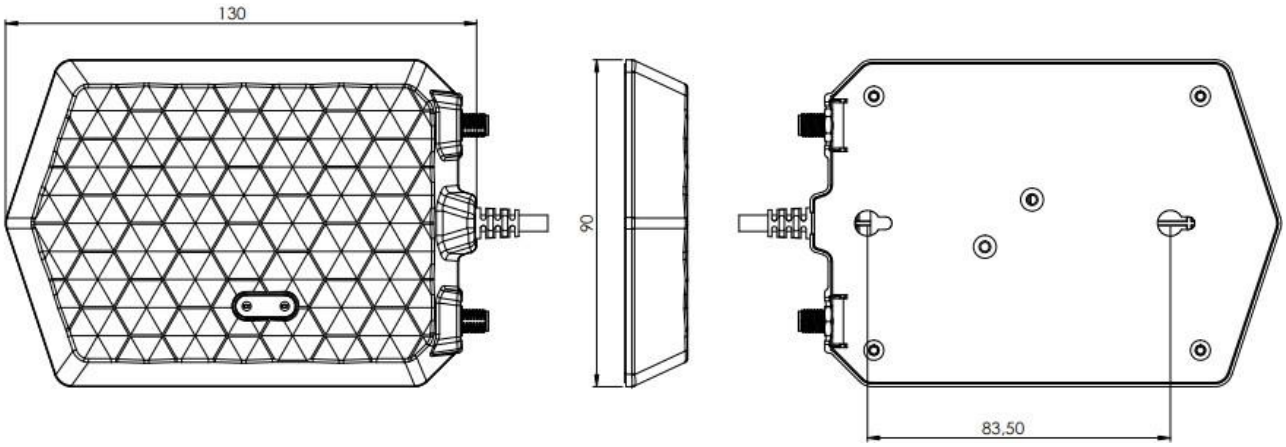
NOTE: It is recommended that before installing the reader, reading tests are carried out in the place where the reader is to be installed.

Part 1 - Overview and installation

Equipment Overview

6 Equipment overview

6.1 Hardware - AcuPad-50 Mux



6.4 Reader general features

Transponder protocols	
Protocol	EPCGlobal Gen2 (ISO 18000-6C). EPCGlobal Gen2 V2 (ISO 18000-63) (transparent mode). Artifact protocol (SJ5511).
Interface RF	
RF Power	0 a 27 dBm
Regulation	ANATEL (BR) 915 – 928 MHz FCC
Mode/ Modulation/ RF codification	Frequency Hopping / PR-ASK / M2, M4, M8.
Backscatter Link Frequency (BLF)	250KHz.
Performance	
Max read distance[1]	Up to 3 meters with 8,5 dBic antenna.
Data / control interface	
Connectors	**Power and communication:** USB type A connector male (USB model).
Interface communication	**USB HID keyboard:** The reader is recognized by the operating system (Windows, Mac, Linux) as a HID dispositive as a keyboard. **USB Serial CDC:**[2] When the reader is plugged into the USB, a Serial port is created automatically (Windows 10, Mac, Linux).
Programming/Configuration	**Autonomous mode:** Configuration via serial communication with commands ASCII via terminal (Putty, Minicom, Screen, etc). It is not necessary to use the API/SDK. Tag read only. **Transparent mode:** The reader starts responding to a Software that uses Mercury API and has tag reading and writing functionality.

[1]The reading distance may vary depending on the antenna connected to the reader, power supply, the tag used and the environment in which the reader is used.
2 Windows 7 and 8 will require driver installation.

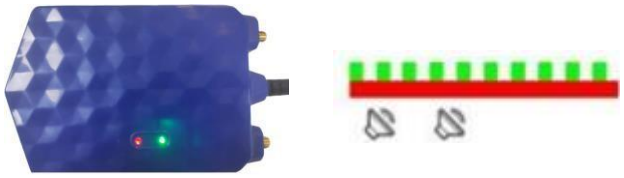
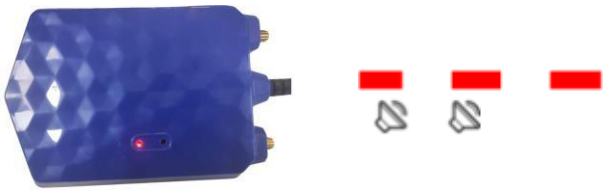
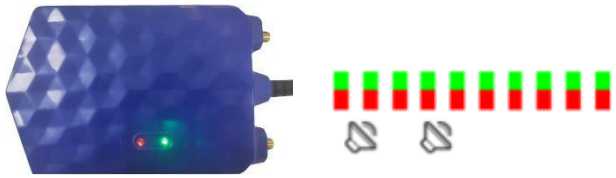
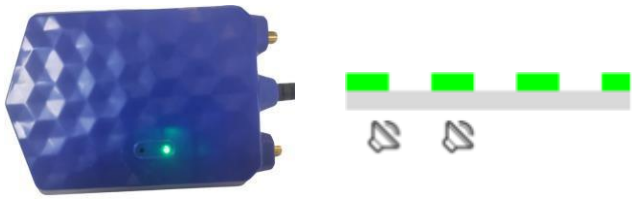
Protection	
Protection against ESD	Passive high pass filter (40dB attenuation in the ESD discharge).

Energy (Continuation)	
Power (Direct Current)	Input voltage: 5VDC +/- 1%. (Host USB port). Maximum Ripple from the source: 200mVpp @20MHz.
Consumption (Direct Current)	Maximum 2.5W in autonomous mode. Maximum 6W in transparent mode.
Physical characteristics	
Reader protection level	Indoor use, no weather protection.
RF Connector	2x RP-SMA Female
Dimensions	125x90x17mm
Operating temperature	-10°C a + 65°C
Storage temperature	-10°C a + 70°C
Air relative humidity	95%
Fixation	Reader with holes for fixation.

7 Installation

7.1 Turning on the reader

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

	Beep and greed led flashing quickly and red led on: Reader is working in **autonomous mode**, trying to read tags.
	Beep and green led off and red led flashing slowly: Reader is working in **autonomous mode**, with RF disabled, **no reading tags.**
	Beep and green and red led flashing quickly simultaneously: Reader energized but not having initialized USB communication with the Operating System. reader no is in operation until communication takes place with the Operational system.
	Beep and green led flashing slowly and red led off: Reader is working in **transparent mode**. Obs. In the transparent mode, the red led and the beep are controlled by the user software.

On computers with Windows Operating System older than 10, it will be necessary to install a driver for creating the COMxx serial communication port.
For Linux, Android and Mac OS, the driver is not required.
When the reader is turned on, the following devices will be created in the Operating System:

- HID class keyboard type;
- Serial port, (COM&LPT in windows), (CDC in Linux, Android and Mac).

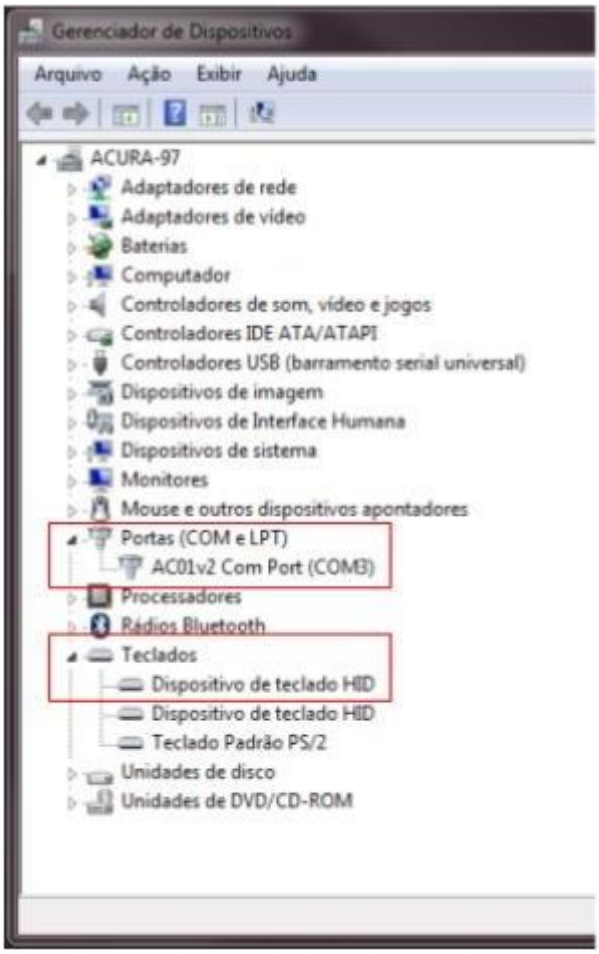
Since the keyboard type HID class will be used by the reader for keyboard emulation when sending the reading results in Autonomous mode and the Serial Port will serve as communication and configuration in the Autonomous and Transparent modes.

7.1.1 Identifying serial port (USB) (Windows)

After turning on the reader, go to “Control Panel” or with the keys “Windows + X” open the “**Device Manager**”. In “**Ports (COM and LPT)**”, a virtual serial communication port should appear 4s after the moment the reader is plugged in, plus 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

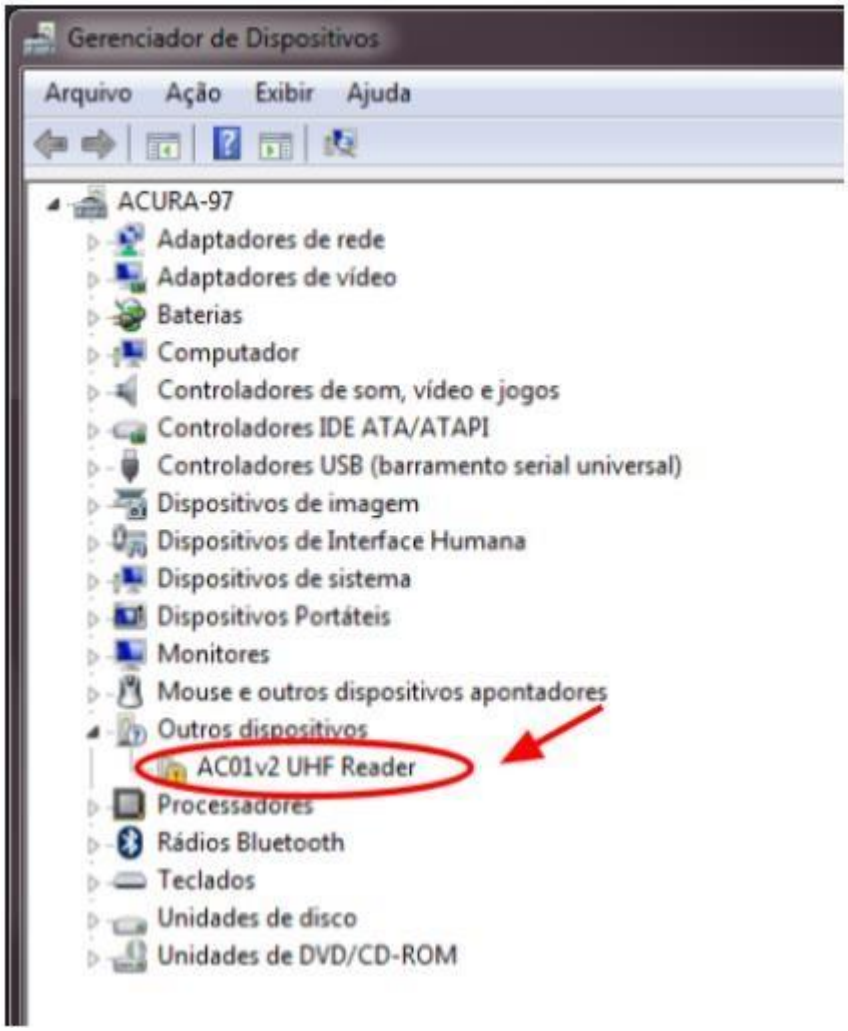
If Windows does not recognize the reader, as in the image below Device Manager, the Reader driver must be installed following the steps below:

7.1.1.1 Installing driver for Windows

The Driver for Windows must be downloaded through the link:
https://drive.google.com/file/d/1PJBSm4KYy0w-L7tRK-BRwGBatqpwLF3r/view

7.1.1.1.1 Step 1: Device detection

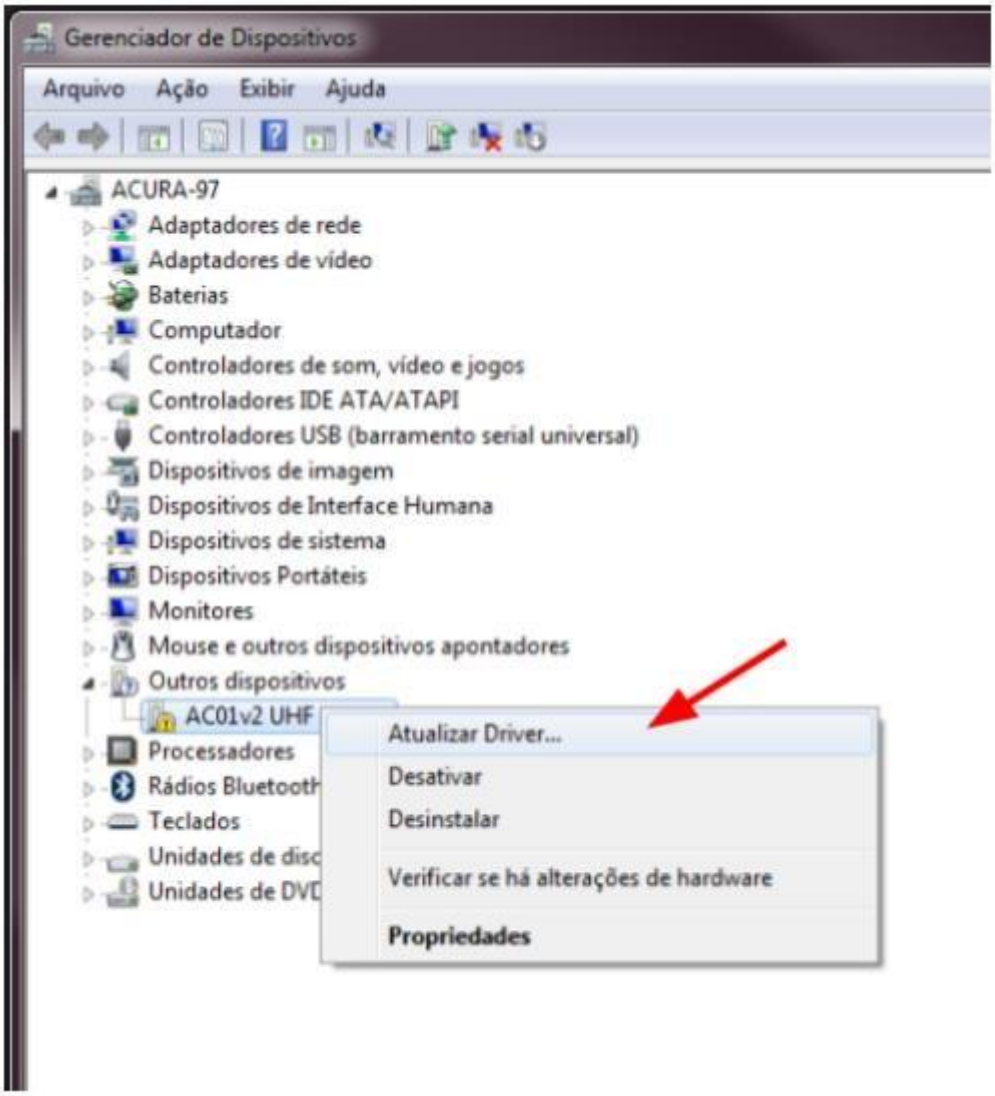
1. Connect the reader to the computer's USB port;
2. In the "Control Panel" or key "Windows + X" open the "**Device Manager**";
3. In “**Other Devices**” the reader connected to the computer should appear, as follows:



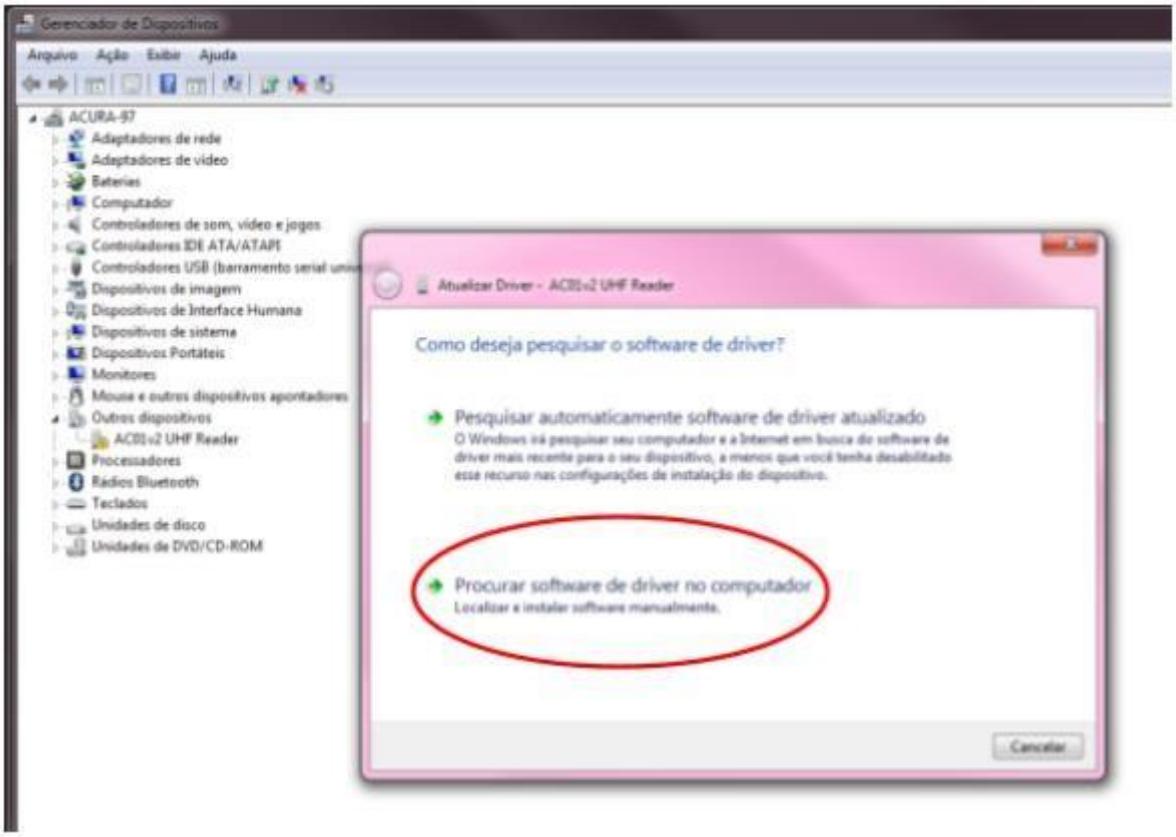
NOTE: For Windows 10 or above, Linux and Mac OS no driver installation required

7.1.1.1.2 Step 2: Load the driver

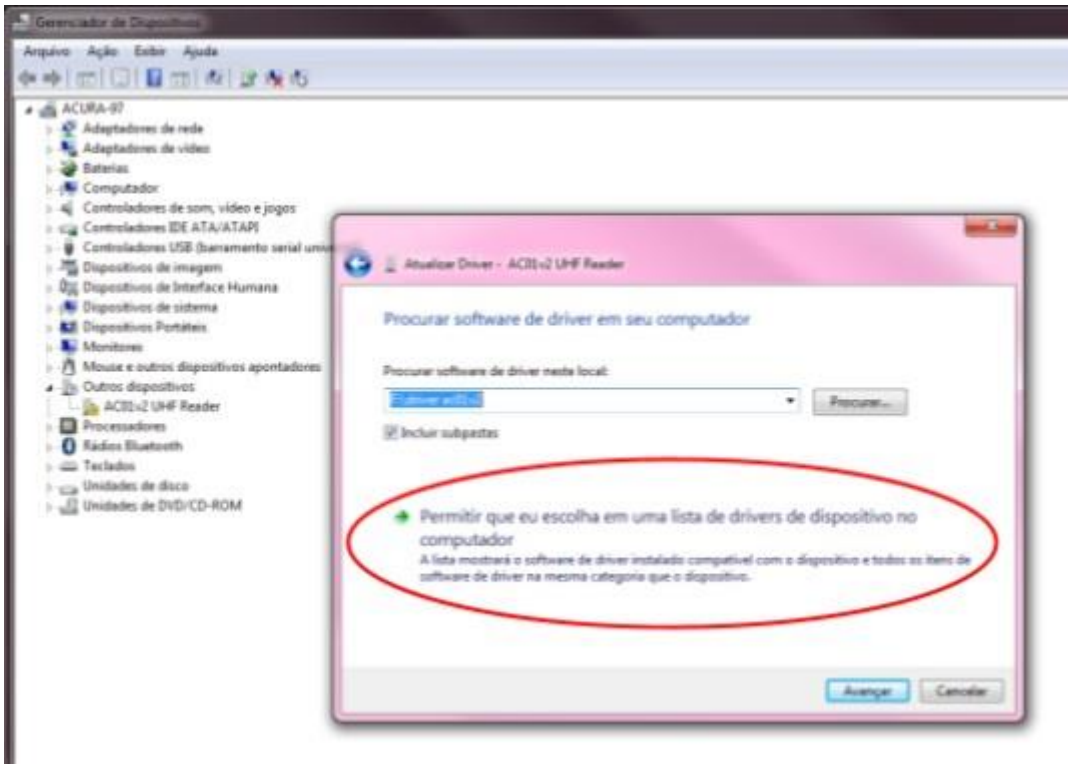
1. With the right mouse button, go to "**Update Driver...**":



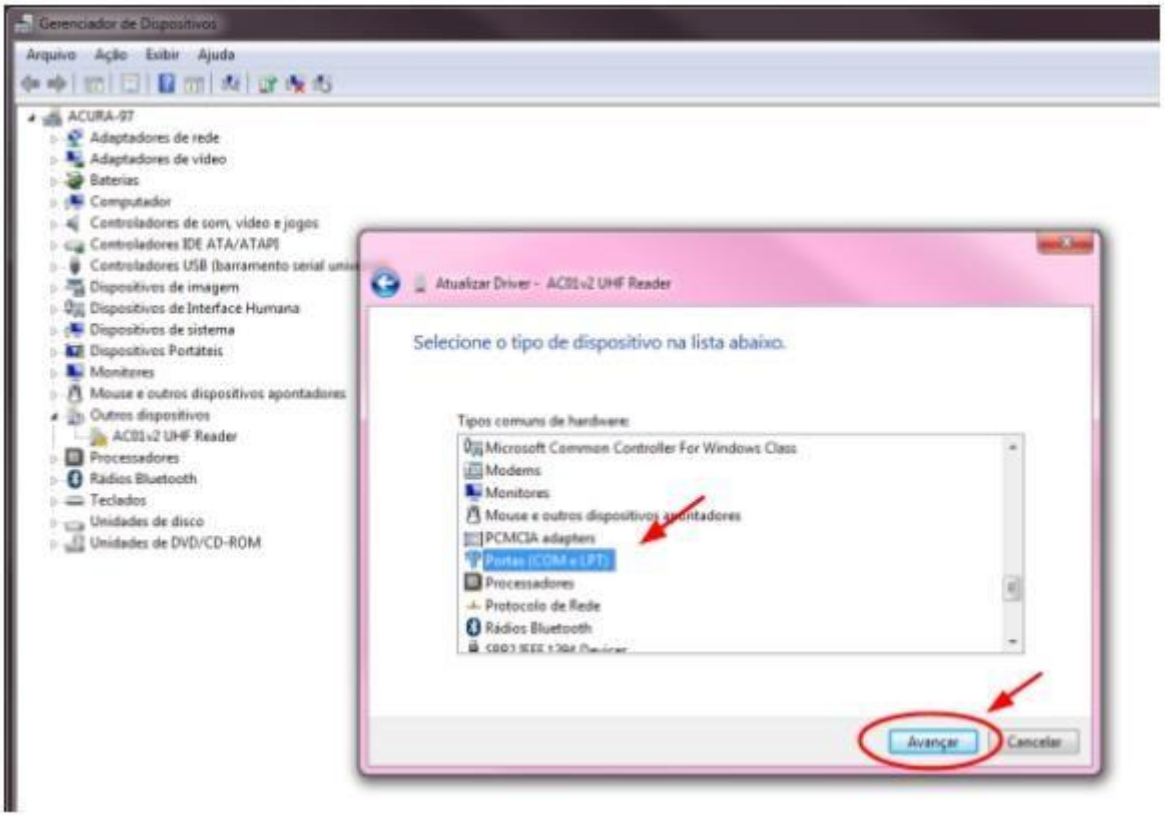
2. Then “Browse my computer for driver software”:



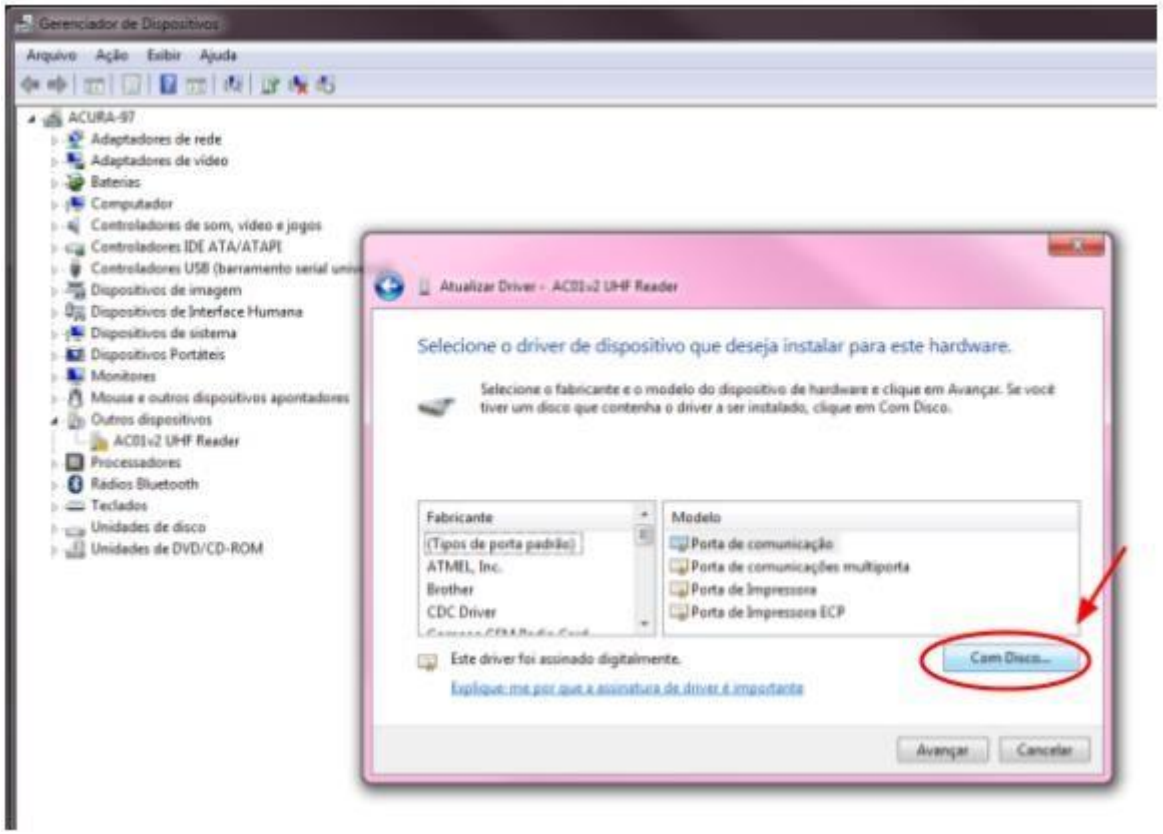
3. Afterwards, select “Allow me to choose from a list of device drivers on the computer”:



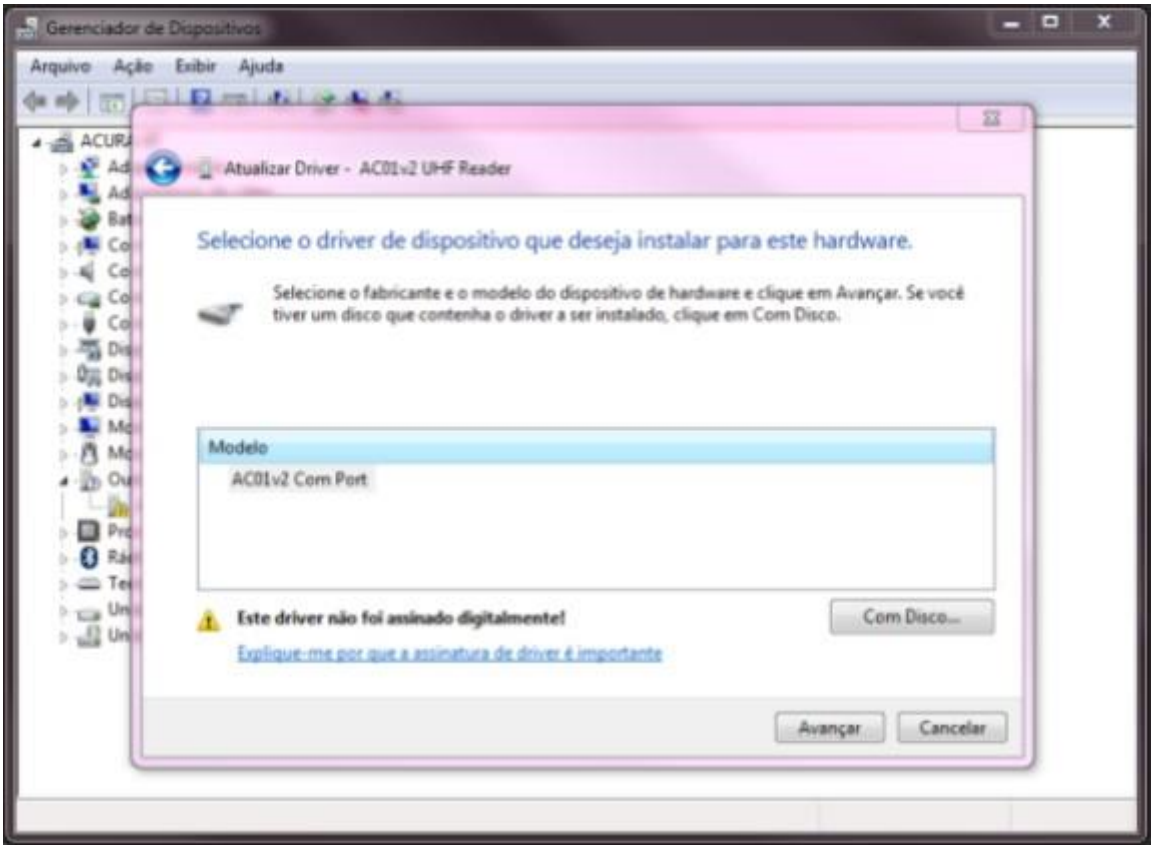
4. Under “**Common types of hardware**” select “Ports (COM and LPT)” and click “**Next**”:



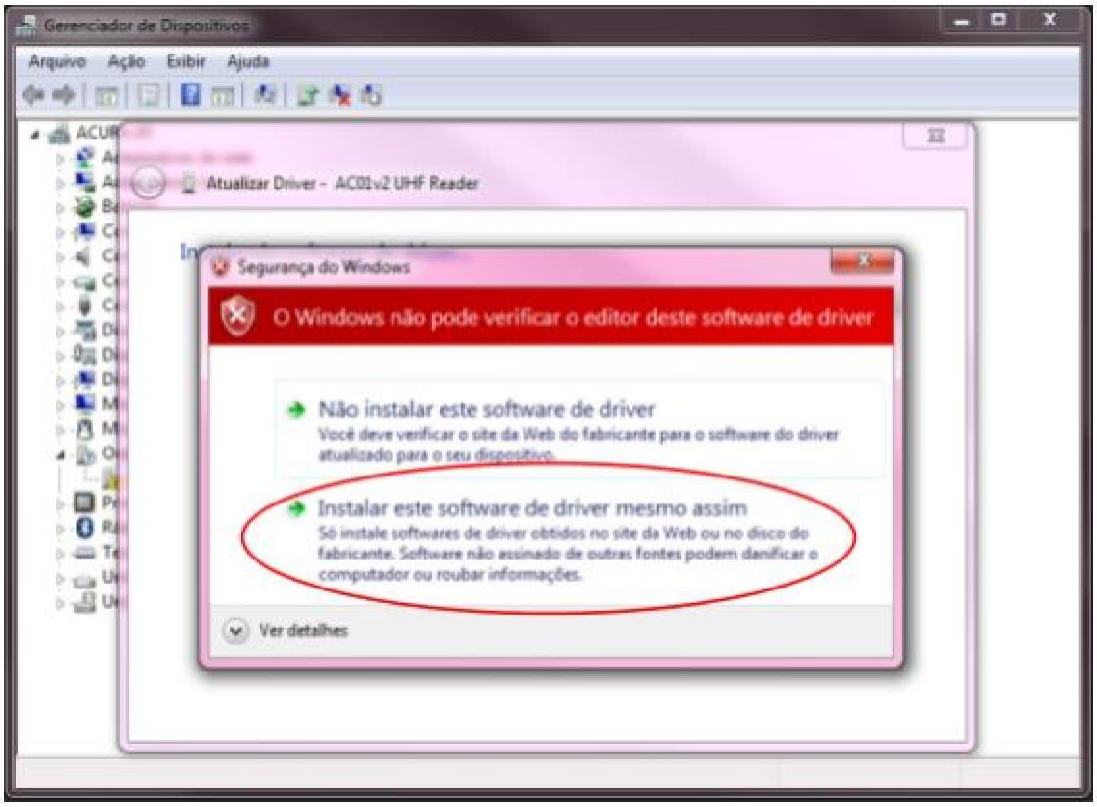
5. Click on “**Have Disk...**”;



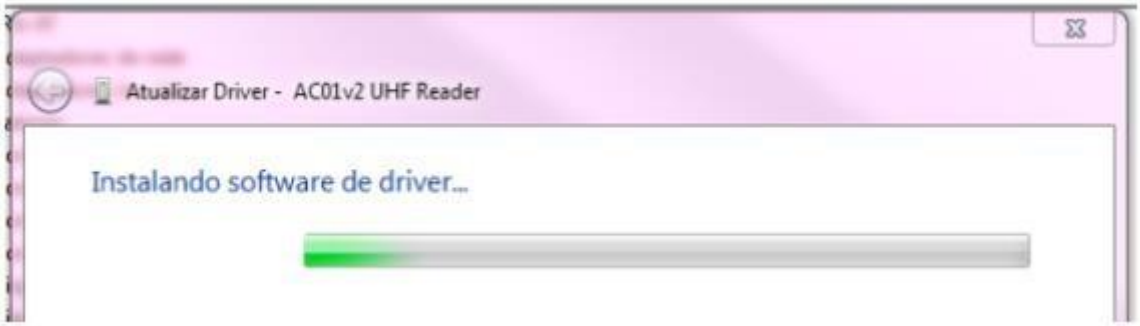
6. In "**Install from Disk**", go to "**Browse...**", select the file to update and click on “**OK**” to confirm. A page similar to the following should appear on your computer. Click on “**Next**”;



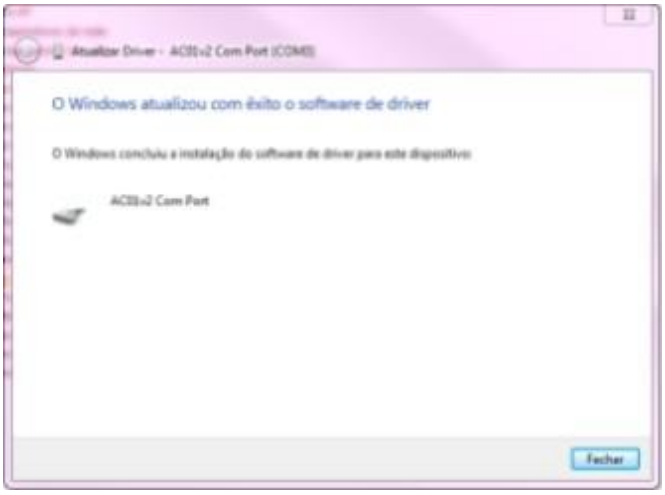
7. A driver update notice should appear. To proceed, click “Yes” to continue installing the driver;
8. A Windows security alert saying “Windows cannot verify the publisher of this driver software” appears, to continue click on “**Install this driver software even though**";



9. After this procedure, wait for the installation;



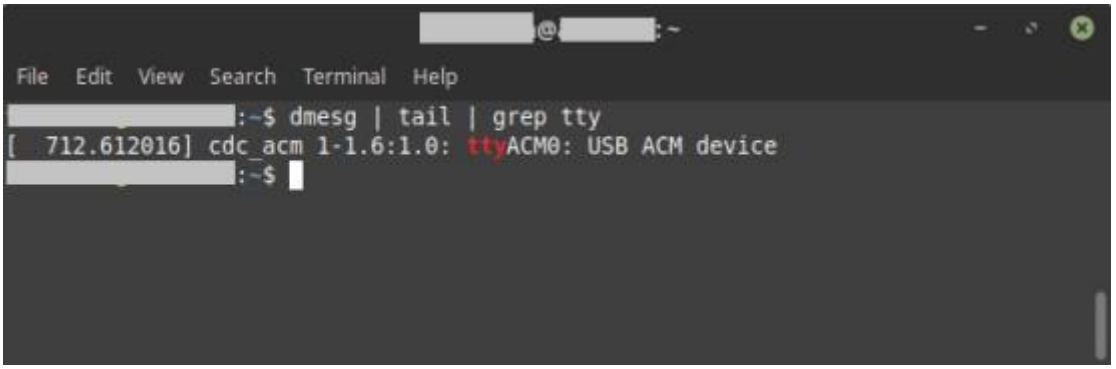
10. Once installed, the update successfully completed message appears, ending like this installing the driver software for the AC-01 v2 Com Port device.



7.1.2 identifying serial port (USB) (Linux)

To identify the name of the serial port created, right after connecting the reader to the USB, type in Terminal:

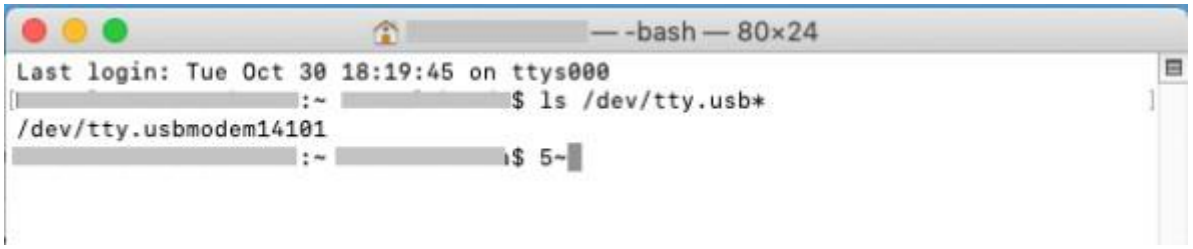
$ dmesg | tail | grep tty



In the example above, the serial communication with the reader will be through **/dev/ttyACM0**.

7.1.3 identifying serial port (USB) (Mac OS)

To identify the name of the serial port created, right after connecting the reader to the USB, type in Terminal:
$ ls /dev/tty.usb*



In the example above, the serial communication with the reader will be through /**dev/tty.usbmodem14101**.

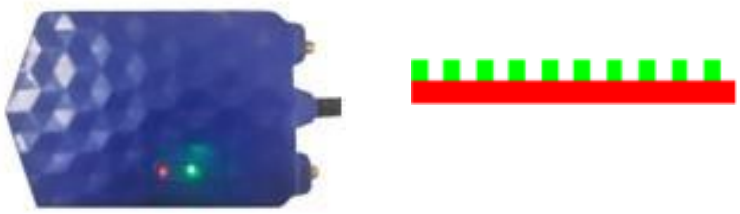
Part 2 - Autonomous mode

Reader in autonomous mode
Reader configuration
Use example

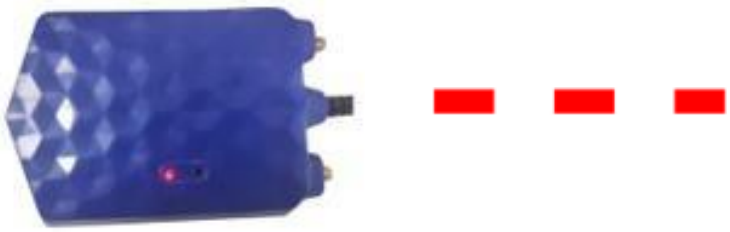
8 Reader in autonomous mode

In Autonomous 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. As of firmware 1.1.3 it was added the option to disable the reading of tags by turning off the RF signal through the command "**readtag**" or **CapsLock** key and activate again, being able to leave the reader in idle or in tag reading.

Visual feedback from the reader when in **Autonomous mode performing tag reading**:

	Green Led flashes fast and Red Led on

Reader visual feedback when in Autonomous mode not performing tag reading, at idle:

	Green Led off and Red Led flashes slowly

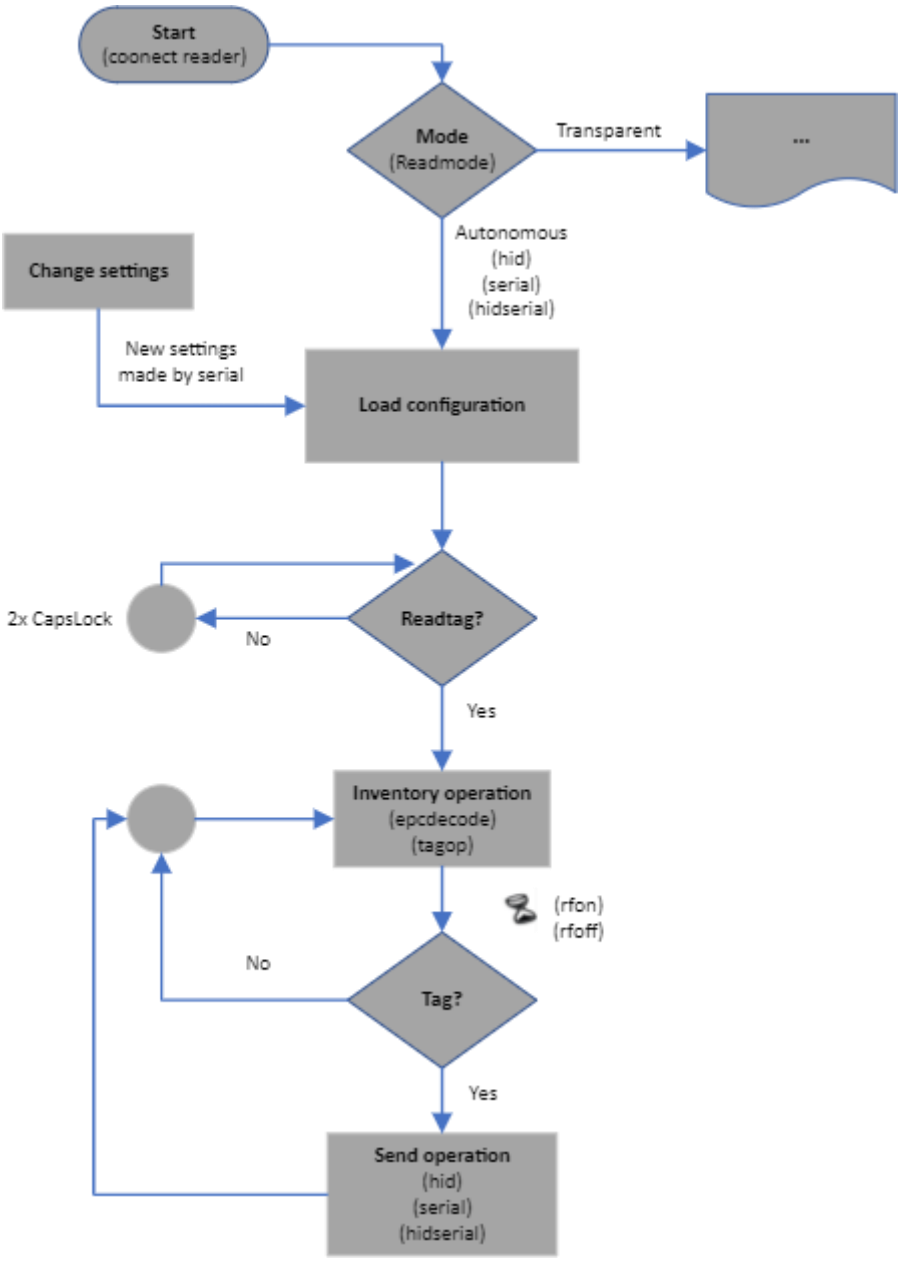
8.1 Activating and deactivating RF in autonomous mode

To enable and disable RF, the following options can be used:

A. Use the **readtag** command, examples:
   a. The reader is in Autonomous mode and the command "**readtag off<CR>**" is sent, this way the reader deactivates the RF and does not read the tag anymore.
   b. The reader is in Autonomous mode and the command "**readtag on<CR>**" is sent, this way the reader activates the RF and goes back to reading tags.

B. Use the "Caps Lock" key (Windows and Linux only), for example:
   a. To switch between one operating mode and another, just quickly **click 2 times in sequence the CapsLock key** and check the LEDs and beep of the reader to see the change.

8.2 General diagram of the autonomous 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. Besides reading the EPC id it is possible to read together any memory bank of the UHF EPC Gen2 tag, such as the amount of reading and its RSSI signal level. All this information is sent by the reader into a String formatted as explained in the next chapter.

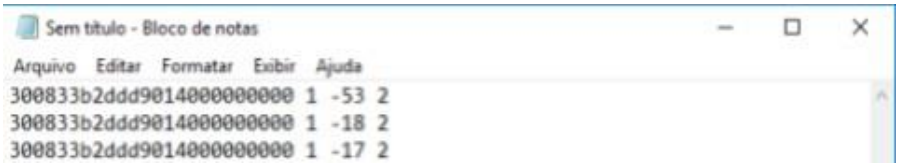
8.3 Read result format

In Autonomous mode, the reader sends a String formatted with the tag reading results in the chosen communication interface.
Below is the format of the String:

<epcdecode><separator><tagop><separator><readcount><separator><rssi><separator><antenna><endofline>
<epcdecode> Result of reading the EPC id with the decoding specified by the **epcdecode** command. *Note: Always reported by the reader in Autonomous mode.*
<separator> Separator character specified by the **separator** command. *Note: It is only reported by the reader if there are more fields than <epcdecode> to report.*
<tagop> Read result of 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> Value of the RF signal level in dBm and negative of the tag response. The option to show or not the value is made by the **reportrssi** command. *Note: Optional.*
<antenna> Value referring to the antenna that reads the tag. The option to show or not the value is made by the **reportantenna** command. *Note: Optional.*
<endofline> End of line type set by the **endofline** command. *Note: Reported by reader in Standalone mode if set other than none.*

WARNING: Result string size may vary depending on 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 read results:



8.4 Configuring the reader

When the reader is in autonomous mode, the following configs can be done:

- Communication type;
- Read or standalone (idle);
- Read power;
- Read time;
- Standalone time;
- Tag EPC id decodification type;
- Read operation of the tag memory bank;
- EPC Gen2 protocol parameters
  - BLF;
  - Tari;
  - Encoding;
  - Session;
  - Target;
  - Q (collision avoidance);
  - Access password;
- Character that separates the fields of the result string;
- String result end of line type;
- Keyboard emulation digitation speed;
- RSSI filter;
- Show or not the RSSI value of the read tag;
- Show or not the amount of reading of the read tag;
- Read tag warning sound;
- Check the internal temperature;
- Decodification type of the PA tag read;
- Show or not which antenna read the tag.

8.4.1 How to set up the reader?

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

NOTE:	To know which serial port the reader will respond to, please consult 7.1 Turning on the reader.

8.4.1.1 Operating logic

Reader configuration via ASCII messages works with the Message/Response structure:

A. The external host send a command for the reader;
B. The reader interprets and execute the command in the message received;
C. The reader answers the external host;

Messages and replies use ASCII characters with a carriage return **<CR>** terminator.
Messages can be in uppercase, lowercase, or both.

NOTE:	The reader echoes each character typed into the serial terminal and each newline the reader sends the character “**>**” to indicate the cursor.

8.4.1.2 Messages and answers sintaxe

Below is the syntax of the messages from the Host to the Reader:

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

WARNING: Only one value must be passed to each command, the reader does not will accept more than one value per command.

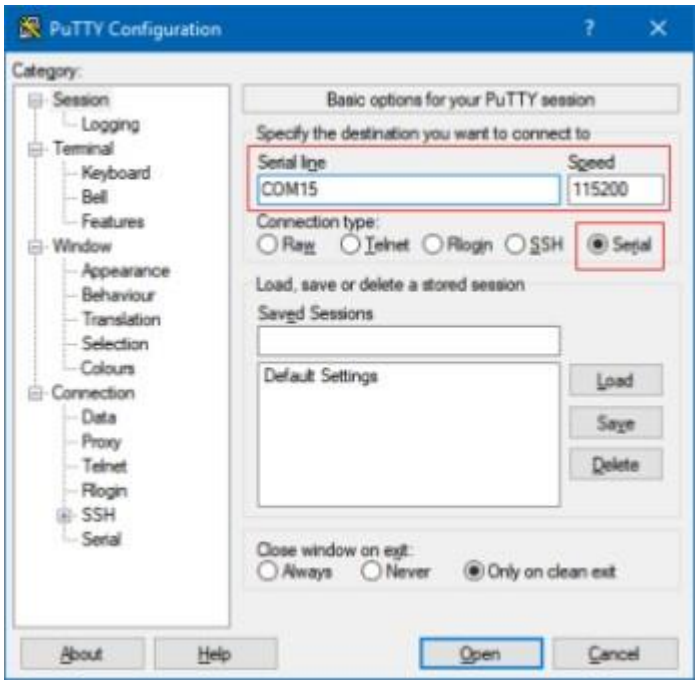
Below is the syntax of the response messages from the Reader to the Host:

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

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

Reader settings in Autonomous mode can be performed through a software terminal emulation. The best known free and open-source code is Putty whose Download link see below.
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
To config the Putty follow the steps:

1. Open the Putty and select “Connection type:” “Serial”, “Serial line” the port that the reader generated and “Speed” 115200.



2. Click in “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 the Windows.
For Mac OS and Linux you can use the “screen” program as shown below.

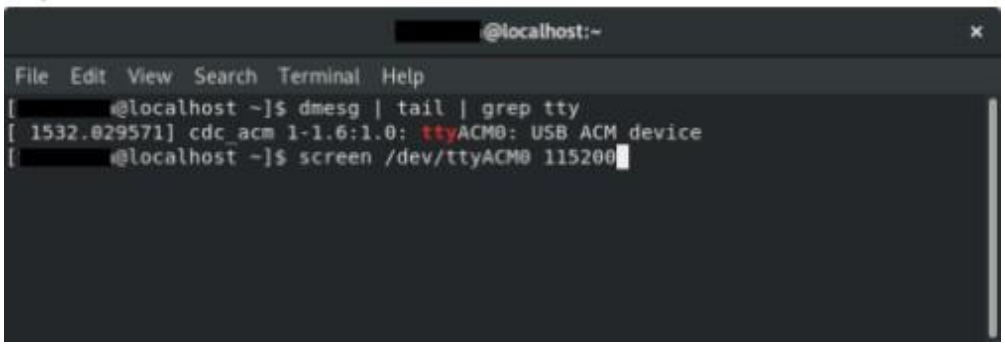
> **NOTE:** To know which serial port the reader will respond to, please consult 7.1 Turning on the reader.

(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 permission of super user (sudo) or add your user to the dialout group using the following command:
> $ sudo usermod -a -G dialout $USER
> To take effect, you must restart the system.

To connect using the screen:
$ screen /dev/ttyACMx



8.5 Configuration commands

8.5.1 Important tip

It's not always easy to remember the command name or accepted values, that's why there is the command **help** which if executed with no value lists all 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 scanner's configuration information. Below are images of the help command.

8.5.2 Configuration commands and default values summary

Command	Description	Pattern	Interface
readmode	Reading mode and communication interface.	hid	USB/RS232
readpower	Read 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, one-bit pulse duration in us.	25	
gen2tagencoding	EPC Gen2 encoding of tag-reader communication.	m4	
gen2session	EPC Gen2 tag section.	s1	
gen2target	EPC Gen2 target of the tag.	a	
gen2q	EPC Gen2 tag quantity in the reading field.	dynamic	
gen2accesspassword	EPC Gen2 tag access password.	00000000	
temperature	Temperature in °C of the reader.		
version	Reader controller firmware version.		
rfidversion	Reader RF module firmware version.		
rfidmodule	Reader RF module name.		
serialnumber	Reader serial number.		
tagop	Tag read operation.	none	USB/RS232
epcdecode	Decoding the tag's EPC id.	none	
autoidcs	Security code for Acura Autoid tags.	9000001	
separator	Result string separator character.	‘ ‘ (space)	
endofline	End-of-line type of the result string.	windows	
keystrokespeed	Hid interface typing speed.	fast	USB
rssifilter	Filter by tag's RSSI value.	0	USB/RS232
reportrssi	Reports the RSSI value of the read tag.	off	USB/RS232
reportreadcount	Reports the amount of tag reading in the period of "rfon".	off	
enablebeep	Tag reading audible warning.	on	
enablecapslock	Shortcut via “CapsLock” key for switching modes Autonomous and Transparent.	on	USB
help	Help text for each command.		USB/RS232

summary	Summary of the values of the reader		
facreset	Returns all settings to the default of the factory.		
upgradefirmware	Starts the firmware update process.		USB
rs232baudrate	Reader BaudRate configuration.	115200	USB/RS232
paresult	Configures the data format for reading PA tags.	raw	
detectionlimit	Return loss threshold in dBm for detection antenna.	7	
returnlossinterval	Return loss measurement range in seconds.	10	
returnloss	Last measured return loss from RF ports.		
reportantenna	Return from the antenna that the tag was read.	off	
antennaport	Defines the antenna(s) to be used.	auto	
readtag	Enables or disables RF in standalone mode.	on	

8.5.3 Command return messages

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

Type	Description
Success	When the value informed is valid and different from the current one recorded in the reader: ok - parameter has been changed.<CR> When the value informed is valid and equal to the current one recorded in the reader: ok - parameter already has this value.<CR> When the faceset command is run: !!! Factory Reset in action !!! The reader will be restarted.<CR>
Error	When a non-existent command is sent to the reader: error - command not found<CR> When the value entered is not among those specified and accepted by the reader: 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> When the number of characters before <CR> is greater than 50: error - internal buffer full.<CR>

8.5.4 Detailing of the configuration commands

8.5.4.1 readmode - Read mode

Command	**readmode** *value*
Function	Check (no value) or change the reader's operating mode.
Description	The reader can work Autonomous with automatic readings, or Transparent where a Software can have full reader control for tag readings and writings.
Pattern	**hid**

Values	• **hid**
	  ◦ (Autonomous) Keyboard emulation mode, when a tag is read the result is sent to the computer as if it were typed quickly.
	• **hidserial**
	  ◦ (Autonomous) Keyboard emulation mode and USB Serial, the result of reading is sent via keyboard emulation and also via communication serial to the virtual USB port.
	• **serial**
	  ◦ (Autonomous) Serial communication mode, reading result is sent via serial communication on the virtual USB port.
	• **transparent**
	  ◦ The reader must be controlled by a Software that has full control of the reader to read and write tags.

Example	To check the operating mode: **readmode<CR>** **hid<CR>** To change the working mode (Standalone reporting via serial only): **readmode serial<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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 - Activate or deactivate the RF, autonomous mode tag read

Command	**readtag** *value*
Function	Check (no value) or enable/disable RF.
Description	Enables or disables the reading of tags by turning off the RF signal from the reader.
Pattern	**on**
Values	**on** **off**
Notes	Pressing the CapsLock key twice quickly executes this command in a cyclic on/off way.
Example	To check the value: **readtag<CR>** **off<CR>** To disable RF and tag reading: **readtag off<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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 reading power in dBm.
Description	The reader in Autonomous mode can have the read power range from 0 to 15 dBm.
Pattern	**10 dBm**
Values	**0 .. 15**
Example	To check read power: **readpower<CR>** **12<CR>** To change the readout power to 5 dBm: **readpower 5<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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 power directly influences the tag reading distance, the temperature at the scanner will achieve and its power consumption, we always recommend setting 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 reading time in ms (milliseconds).
Description	Read time is the time the reader transmits RF energy to your antenna in an attempt to read a tag. When you have many tags to read at the same time, the reading time must be increased.
Pattern	**250 ms**
Values	**50 .. 5000**
Example	To check the read time: **rfon<CR>** **250<CR>** To change the reading time to 500 ms (0.5 s): **rfon 500<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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 the rfon rfoff times.

NOTE: If the application requires reading multiple tags at the same time, leave the rfon above 250ms to allow 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 reading time in ms (milliseconds).
Description	The read interval time is the time the reader turns off RF power on your antenna.
Pattern	**250 ms**
Values	**0 .. 1000**
Example	To check the read time: **rfoff<CR>** **250<CR>** To change the reading time to 500 ms (0.5 s): **rfoff 500<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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 the rfon rfoff times.

8.5.4.6 gen2blf - EPC Gen2 Backscatter Link Frequency

Command	**gen2blf** *value*
Function	Check (no value) or change “Backscatter Link Frequency” parameter from default Gen2.
Description	The BLF is the data rate in KHz of the transmission over the air between a tag and the reader.
Pattern	**250 Khz**
Values	**250** **320** **640**
Notes	Changing the BLF to 640 or 320 KHz, the Tari value and tag encoding will be automatically changed to 6.25 us and fm0 respectively.
Example	To check the BLF value: **gen2blf<CR>** **250<CR>** To change the BLF value to 640 Hz: **gen2blf 640<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

WARNING: Changing the gen2blf will imply the automatic change by the reader of other EPC gen2 parameters, which are the gen2tari and gen2tagencoding.

8.5.4.7 gen2tari - EPC Gen2 Tari pulse time

Command	**gen2tari** *value*
Function	Check (no value) or change the “Tari” parameter from the Gen2 default.
Description	The Tari is the time in use (microseconds) of the pulse of each bit transmitted by the tag.
Pattern	**25 us**
Values	**6.25** **12.5** **25**
Notes	The values 12.5 us and 25 us can only be configured if gen2blf is configured to 250 kHz. The values 6.25 us and 12.5 us are only supported by the internal RF module type micro-module, which can be verified by the rfidmodule command.
Example	To check the Tari value: **gen2tari<CR>** **25<CR>** To change the Tari value to 12.5 us: **gen2tari 12.5<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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>**

WARNING: The Tari value must be combined with the BLF value.

8.5.4.8 gen2tagencoding - EPC Gen2 codification

Command	**gen2tagencoding** *value*
Function	Check (no value) or change the “Tag data encoding” parameter from the Gen2 standard.
Description	It is the encoding type of the data sent over 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 decoding each bit. m8 has a high read sensitivity rate but a slower speed in decoding each bit.
Example	To check the Data encoding value: **gen2tagencoding<CR>** **m4<CR>** To change the encoding tag value to m8: **gen2tagencoding m8<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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>**

WARNING: The value of the encoding Tag must be combined with the value of the BLF.

8.5.4.9 gen2session - EPC Gen2 tag session

Command	**gen2session** *value*
Function	Check (no value) or change the “Session” parameter from the Gen2 default.
Description	It is the type of session in which the reader will read a tag.
Pattern	**s1**
Values	**s1** **s2** **s3** **s4**
Notes	The combination of gen2session and gen2target values reflects the repetition rate of reading the same tag.
Example	To check the Session value: **gen2session<CR>** **s1<CR>** To change the session value to s0: **gen2session s0<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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 8.7 Same tag read rate.

8.5.4.10 gen2target - EPC Gen2 tag target

Command	**gen2target** *value*
Function	Check (no value) or change the “Target” parameter from the Gen2 default.
Description	It is the flag of each session of the tag.
Pattern	**a**
Values	**a** Reading tags with session flag in state A. **b** Reading tags with session flag in state B. **ab** Reading tags with session flag in state A, then in B. **ba** Reading tags with session flag in state B, then in A.
Notes	The combination of gen2session and gen2target values reflects the repetition rate of reading the same tag.
Example	To check the Session value: **gen2target<CR>** **a<CR>** To change the target value for ab: **gen2target ab<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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 8.7 Same tag read rate.

8.5.4.11 gen2q - EPC Gen2 number of tag in the read field

Command	**gen2q** *value*
Function	Check (no value) or change parameter “Q” from Gen2 default.
Description	It is a numerical value that the reader uses to regulate the probability of a tag's response, used for collision avoidance when multiple tags are in the reader's read field. Two Q is equivalent to the number 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 reader's reading field.
Notes	If the number of tags to be read at the same time varies, leave the value q 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 q value to 3, with 8 tags in the read field: **gen2q 3<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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 not be known.

8.5.4.12 gen2accesspassword - EPC Gen2 tag access password

Command	**gen2accesspassword** *value*
Function	Check (no value) or change the “Access password” parameter from the Gen2 default.
Description	Is the tag's 32-bit (8 digits) password in hexadecimal.
Pattern	**00000000**
Values	**00000000 .. ffffffff**
Notes	To read the tag's reserved memory bank with the tagop command, the password must be the same as the tag.
Example	To verify the Access password: **gen2accesspassword<CR>** **00000000<CR>** To change the password to 12345678: **gen2accesspassword 12345678<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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 tag's protected memory region, Reserved, the tag's access password must be informed to the reader through this command.

8.5.4.13 tagop - Tag read 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 from the Acura Autoid System.
Pattern	**none**
Values	**none** No memory bank read operation. **autoid** Reads Acura Autoid System tags. **pa** It reads the Artifact protocol (SJ5511) toll tags. **reserved:wp:wl** Reads the tag's RESERVED memory bank. **epc:wp:wl** Reads the tag's EPC memory bank. **tid:wp:wl** Reads the tag's TID memory bank. **user:wp:wl** Reads the tag's USER memory bank.
Notes	**wp** - word pointer, indicates the initial word position of the reading in the memory bank specified. **wl** - word length, indicates the number of words to be read from the memory bank specified. Maximum value for wp and wl is 32. To read the entire memory bank, just specify zero for wp and wl.
Example	To read the entire contents of the tag's User memory bank: **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 bank from the tag, just the EPC result: **tagop none<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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, just specify **zero for wp and wl**.

8.5.4.14 epcdecode - Tag EPC id decodification

Command	**epcdecode** *value*
Function	Check (no value) or change decoding type for tag EPCID reading.
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 checkout applications. maintenance the tag can be encoded in ASCII. Decoding the tag's EPCID returns a readable result for each application.
Pattern	**none**
Values	**none** No decoding is performed and the tag's EPCID result is reported in hexadecimal. **gs1string** GS1 pattern that returns the “GS1 element string”. **gs1epcuri** GS1 standard that returns the “EPC Tag URI”. **gs1epcpureuri** GS1 pattern that returns the “EPC Pure Identity URI”. **gs1gtin13** GS1 standard that returns the GTIN13 code without the serial number. **gs1sgtin13** GS1 standard that returns the GTIN13 code + the serial number. **acsii** Decodes the tag's EPCID according to the Ascii table. **wiegand26** Decodes the last 3 bytes of the tag's EPCID into Sitecode + ID. **magstripe** Decodes the last 4 bytes of the tag's EPCID 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 tagop type autoid.
Example	To read the GTIN13 number + 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 from a tag encoded in the GS1 standard: **epcdecode gs1string<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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 - Security code of the Acura autoid system

Command	**autoidcs** *value*
Function	Check (no value) or change the value of the Autoid System Security Code of the The cure.
Description	It's a six-digit number
Pattern	**900001**
Values	**000001 .. 999999**
Notes	This command should only be used if the 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>**
Error	When the value entered is not among those 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 only used if the application is to read the Autoid System tags of the Acura, tagop set to autoid.

NOTE:	The security code value is printed on the tag, please refer to 8 8.3 Register for the Acura Autoid System.

8.5.4.16 separator - Separator character of result string

Command	**separator** *value*
Function	Check (no value) or change the separator character of the fields of the result of reading a tag.
Description	It is a character that separates the fields in the tag read result string.
Pattern	‘ ‘ (space)
Values	space To have a space as a separator. ‘char’ Any human-readable character from the Ascii table.
Notes	For space, you must write space as the value for the command.
Example	To check the separator: separator<CR> space<CR> To change the separator to '|' : separator |<CR> ok - parameter has been changed.<CR>
Error	When the value entered is not among those specified: error - value out of range<CR> When more than one value is passed to the command: error - too many args.<CR>

NOTE:	The separator character in conjunction with the end of line must be used to divide 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 end-of-line characters.
Description	It is a non-readable character that sits at the end of the result string and represents the end of line for the Operating System.
Pattern	**windows**
Values	**windows:qt** Windows standard end of line <CR><LF>. **unix:qt** Unix standard line ending <LF>. **macintosh:qt** Mac<CR> standard end of line. **tab:qt** End of line will be the tab character Tab. **none** No end of line in the result string.
Notes	**qt** - Amount of end-of-line repetition, values 1 to 10. If :**qt** is omitted, it will only be considered an end of line.
Example	To check which endpoint is configured: **endofline<CR>** **windows<CR>** To change the end of line to Tab and repeat 3 times: **endofline tab:3<CR>** **ok - parameter has been changed.<CR>** To change the end of line for windows and repeat 5 times: **endofline windows:5<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

NOTE: The separator character in conjunction with the end of line must be used to divide the result string by the Software that will receive this string.

8.5.4.18 keystrokespeed - Typing speed of keyboard emulation

Command	**keystrokespeed** *value*
Function	Check (no value) or change keyboard emulation typing speed.
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** Fast 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 typing speed: **keystrokespeed<CR>** **fast<CR>** To change the speed to slow: **keystrokespeed slow<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

NOTE: This value must be changed only if the Host that will receive the result string via keyboard emulation presents problems due to the typing speed exerted by the reader.

8.5.4.19 rssifilter - RSSI filter read

Command	**rssifilter** *value*
Function	Check (no value) or change RSSI filter cutoff value for reading.
Description	The tag's signal level indicates whether the tag is near or far 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 an RSSI equal to or greater than the specified will be reported by the reader. The tag signal level is measured in dBm and is a negative value, that is, the power in Watts is always less than 1mW.
Pattern	**0**
Values	**0 .. 90**
Notes	To disable the filter, set the value 0. The value specified can not have the negative signal
Example	To check the filter value: **rssifilter<CR>** **0<CR>** To change the filter value to 45: **rssifilter 45<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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 tag's RSSI value is used to restrict unwanted reading areas to the around the reader. It should always be used in conjunction with read power.

8.5.4.20 reportrssi - RSSI value report of the tag

Command	**reportrssi** *value*
Function	Check (no value) or change sending 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>**
Error	When the value entered is not among those 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 - report the amounts of tag reading

Command	**reportreadcount** *value*
Function	Check (no value) or change sending the tag reading amount.
Description	Enables or disables sending the amount of tag reading in the “rfon” period in the string of result.
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>**
Error	When the value entered is not among those 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 - warning sound

Command	**enablebeep** *value*
Function	Check (no value) or change the audible warning at each tag reading.
Description	Enables or disables the audible warning at 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>**
Error	When the value entered is not among those 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 switch reading mode

Command	**enablecapslock** *value*
Function	Check (no value) or change the use of the shortcut to switch reading mode.
Description	Enables or disables the use of the “CapsLock” key 5x fast typing, to change the mode Autonomous to Transparent and vice versa.
Pattern	**off**
Values	**on** **off**
Notes	This functionality only works on Windows and Linux operating systems. The 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 usage: **enablecapslock on<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those 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 6 Switching from Autonomous to Transparent mode and vice versa.

8.5.4.24 help - Help about the commands

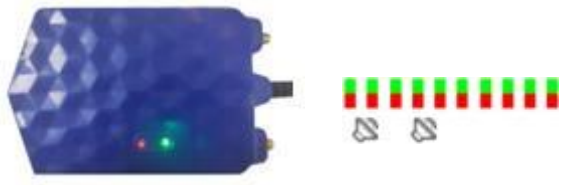
Command	**help** *value*
Function	List all commands (no value) or show detailed help for the passed command.
Description	To get detailed help on a particular command just type: “help command<CR>”.
Example	To check the commands accepted by the reader: **help<CR>** **lista de comando...<CR>** For detailed 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:wl EPC bank.*<CR>** **tid:wp:wl TID bank.*<CR>** **user:wp:wl User bank.*<CR>** **autoid Autoid tags.<CR>** **none No tagop.<CR>** *** wp - word pointer, wl - word length, max 32, menbank:0:0 entire bank.** **Default: none<CR>**
Error	When the value entered is not among those specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

8.5.4.25 summary - Reader configuration values summary

Command	**summary**
Function	Lists all values configured on the reader.
Error	When a value is passed to the command: **error - too many args.<CR>**

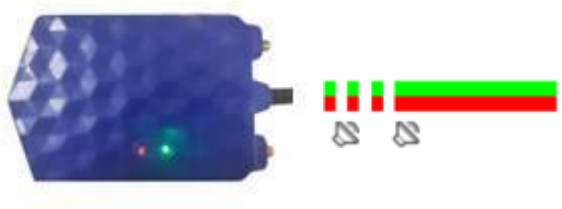
8.5.4.26 facreset - Factory reset

Command	**facreset**
Function	Returns all values to the factory default.
Error	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 let you know that you are in the process of factory reset.

8.5.4.27 upgradefirmware - Firmware upgrade

Command	upgradefirmware
Function	Starts the reader firmware update process.
Example	To check the commands accepted by the reader: help<CR> lista de comando...<CR> >upgradefirmware !!! Bootloader will be executed !!! The reader will be restarted.
Error	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 let you know that you are in the process of firmware update.

8.5.4.28 paresult - PA tag data format

Command	**paresult** *value*
Function	Check (no value) or change the output format of PA tag readings.
Description	Configures the data format of PA tag readings.
Pattern	**raw**
Values	**acura** **honeywell** **raw**
Example	To check the read data format of the reader's PA tags: **paresult<CR>** **raw<CR>** To change the reader's PA tags read data format value: **paresult acura<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

8.5.4.29 detectionlimit - Detection limit

Command	**detectionlimit** *value*
Function	Check (no value) or change the return loss threshold value in dBm to antenna detection at the port.
Description	Sets the return loss threshold in dBm for antenna detection.
Pattern	**7**
Values	**5 .. 15**
Example	To check the return loss threshold value in dBm for antenna detection: **detectionlimit<CR>** **7<CR>** To change the return loss threshold value in dBm for antenna detection: **detectionlimit 5<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

8.5.4.30 returnlossinterval - Return loss interval

Command	**returnlossinterval *value***
Function	Check (no value) or change the return loss measure range value in seconds.
Description	Sets the return loss measurement interval in seconds.
Pattern	**10**
Values	**10 .. 60**
Example	To check the return loss measure interval value in seconds: **returnlossinterval<CR>** **10<CR>** To change the return loss threshold value in dBm for antenna detection: **returnlossinterval 15<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

8.5.4.31 returnloss - Return loss

Command	**returnloss** *value*
Function	Check (no value) the last measured return loss of the rf ports.
Description	Checks the last measured return loss of the rf ports.
Example	To check the return loss measure interval value in seconds: **returnlossinterval<CR>** **0 4<CR>**
Error	When more than one value is passed to the command: **error - too many args.<CR>**

8.5.4.32 reportantenna - Report antenna

Command	**reportantenna** *value*
Function	Check (no value) or change the option to show or not the antenna that the tag was read.
Description	Configures the option to show or not the antenna that the tag was read.
Pattern	**off**
Values	**on** **off**
Example	To check whether or not to show the antenna that the tag was read: **reportantenna<CR>** **off<CR>** To change the value of the option to show or not the antenna that the tag was read: **reportantenna on<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those specified: **error - value out of range<CR>** When more than one value is passed to the command: **error - too many args.<CR>**

8.5.4.33 antennaport - Antenna port

Command	**antennaport** *value*
Function	Check (no value) or change the antenna to be used for reading.
Description	Configure the antenna to be used for reading.
Pattern	**auto**
Values	**auto** **1** **2** **12**
Example	To check the antenna being used for reading.: **antennaport<CR>** **1<CR>** To change the value of the option to show or not the antenna that the tag was read: **antennaport 2<CR>** **ok - parameter has been changed.<CR>**
Error	When the value entered is not among those specified: **error - value out of range<CR>** When more than one value is passed to the command: error - too many args**.<**CR>

NOTE: In “**auto**” mode the reader will detect the antenna by measuring the return loss, through each "**returnlossinterval**" return loss interval along with the value of the detection limit "**detectionlimit**".

8.6 Changing from Autonomous mode for Transparent and vice versa.

To change the reader's operating mode, the following options can be used:

A. Use the **readmode** command, examples:
   a. The reader is in Autonomous mode and the command “**readmode transparent<CR>**” is sent,in this way the reader enters the Transparent mode of operation.
   b. The reader is in transparent mode and the **$ off<CR>** command is sent, so the reader enters Autonomous mode of operation with hid interface.

NOTE:	When in transparent mode, the reader does not echo characters typed into the serial terminal.

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

WARNING:	The Caps Lock shortcut for changing the operating mode only works if and only if the reader's serial port is not connected (not being used by any Software).

NOTE:	When the reader's communication interface is RS-232, switching the operating mode must be done exclusively by the “readmode” command.

8.7 Same tag read rate

The same tag read rate can be controlled by 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 the communication between the tag and the reader.
Combinations of values for the Tag's Session and Target reflect how often a tag that has already responded to an inventory will respond again, in other words, how fast is the replay reading a tag.
Examples:

A. I would like to read only once the EPC id of a tag and that the same EPC id is not repeat while the tag is in the reader's reading field. Configure as follows:
gen2session s2<CR>
gen2target a<CR>

B. I would like to repeatedly read the same tag while it is in the reader's reading field. Configure 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. Highest repeat read rate possible is every 50ms:
rfon 50<CR>
rfoff 0<CR>

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

8.8 Use examples (Autonomous mode)

Below are some examples of use and application of the reader with the details of the commands of settings.

8.8.1 Checkout - pattern GS1/EAN tag

For applications using UHF EPC Gen2 tags encoded according to the GS1 EPC Tag standard Data Standards (TDS) 1.11, the AcuPad-50 Mux reader can be used and supports the following tag encodings:

- **sgtin-96**
- **sgtin-198**

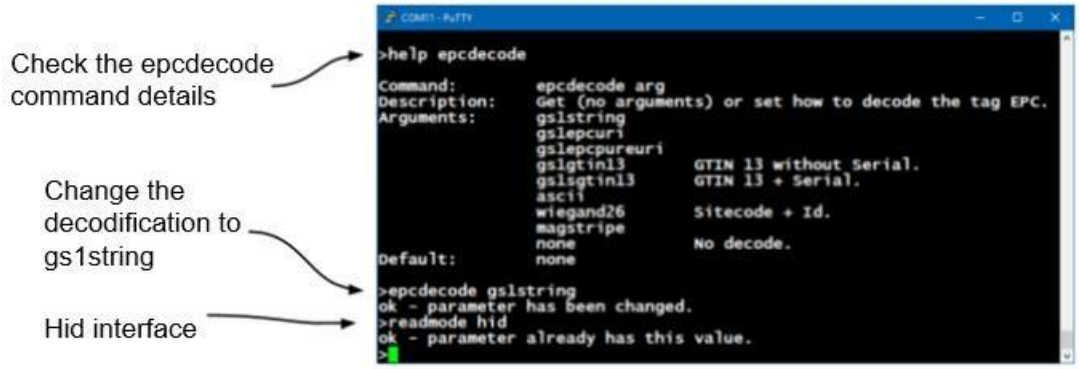
Encoded tag read results can be sent in the following formats:

- **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 serial number, example “3608449920322”;
- **GTIN13 + serial number**, example “3608449920322 1540341”.

8.8.1.1 Read SGTIN-96/198 codified tag and show GS1 element string format result

Configure the reader as follows:

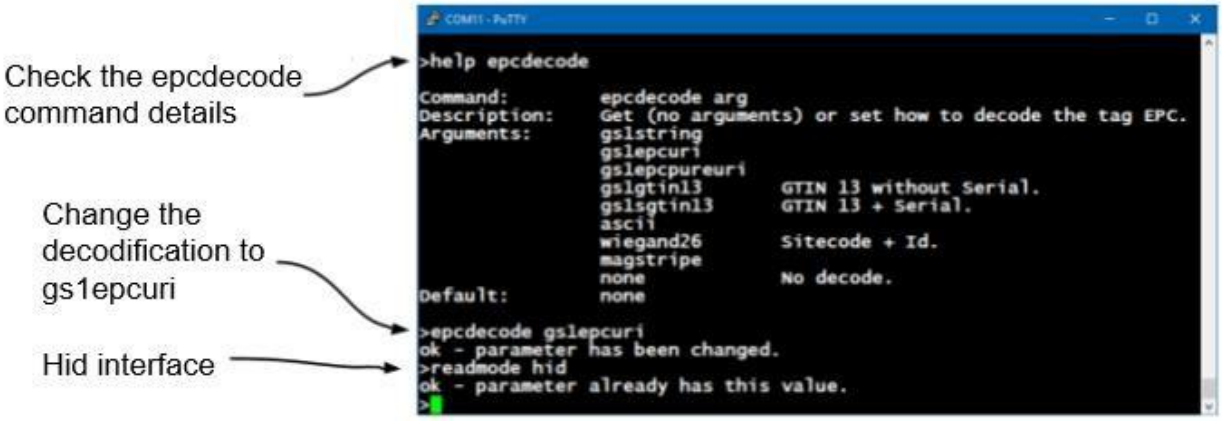
epcdecode gs1string<CR>
(to decode sgtin-96 or sgtin-198 and send in GS1 element string format)

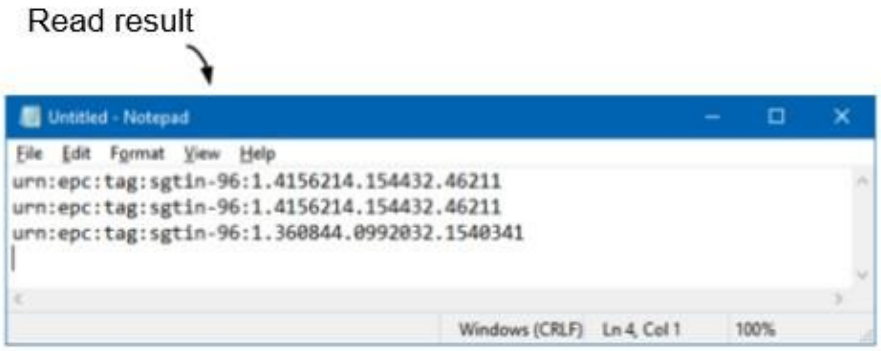


8.8.1.2 Read SGTIN-96/198 codified tag and show EPC tag URI format result

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 8.5.4.14 epcdecode – Tag EPC id decodification.

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

8.8.2 Artifact protocol pattern tags (SJ5511)

For applications using Artifact Protocol (SJ5511) toll tags, the AcuPad-50 Mux can be used to read tags andreport them in the following formats, which can be changed by the "paresult" command:

- **acura**
  - PA-TAG epc:EPCID rn96:R96 data:AES128 rssi:RSSI antenna:ANTID
    Example of reading:
    PA-TAG epc:003019ffffff rn96:3cc96763cd8b3976582ceeb6
    data:f868d50bfa48aae46170cc3b11f2ac5440c9707eb04e0977f75d8219f7a59998
    rssi:-39.0 antenna:1
- **raw**
  - EPCID R96 AES128
    Example of reading:
    003019ffffff 181048b51424cbc7056ca972
    ba9f998f29de7a8e9e8a441977b653bb1b9089fe081a1d359ad9f989531d0943
- **honeywell**
  - EVT:LISTEN-OUT(5):EVT:TAG HEPCID HR96 HAES128 RSSI 0 ANTID
    Example of reading:
    EVT:LISTEN-OUT(5):EVT:TAG H003019ffffff Hdc143e3dd3febabce1a9ba42
    Hbc63e3bf0a51400ebbe74fbb4f5976d1e0a7b121f13639c16b336c61c23329d7 -42.0 0 1

8.8.2.1 Read artifact protocol toll tag (SJ5511)

Configure the reader as follows:

- Configure read operation mode:
  - **tagop pa<CR>**
- Configure the read result format:
  - **paresult acura<CR>**
    (to send in the format “EPCID rn96:R96 data:AES128 rssi:RSSI antenna:ANTID”)

Read result:

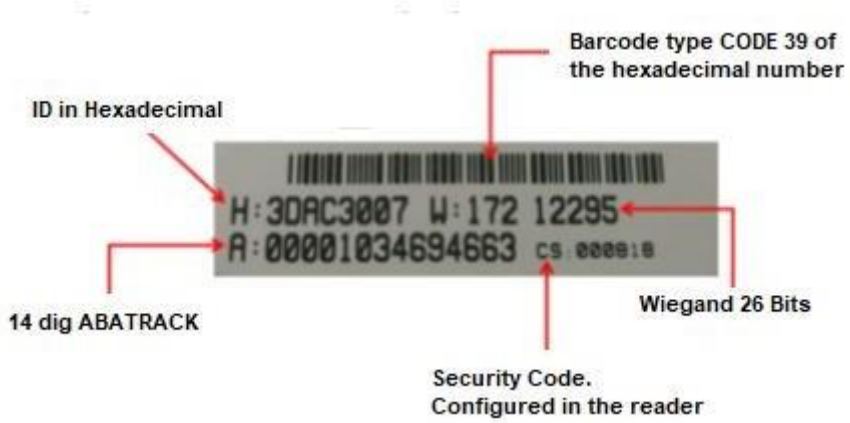
NOTE: To change the reading result format to “honeywell”, or “raw”, just follow the same procedure but send the commands: “**paresult honeywell<CR>**”, or “**paresult raw<CR>**”.

8.8.3 Recorder for the Acura Autoid System

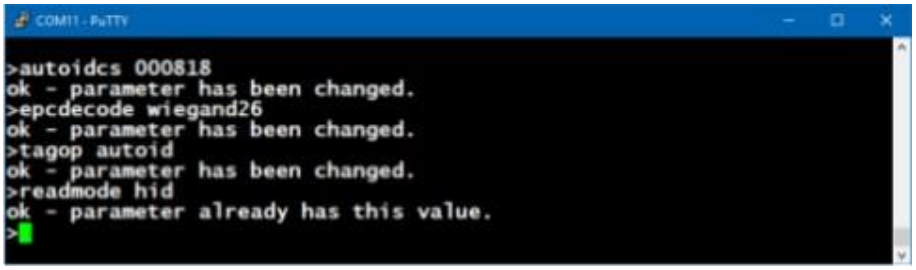
The AcuPad-50 Mux Reader can be used to read Acura Secure Tags from the Autoid System on vehicleidentification applications.
The reader can be used as a tag register, facilitating and avoiding errors in the data entry of the Site access control system.
The following settings must be made on the reader:

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

Below is an image of an Acura Autoid System secure tag tag:



Reader configuration:

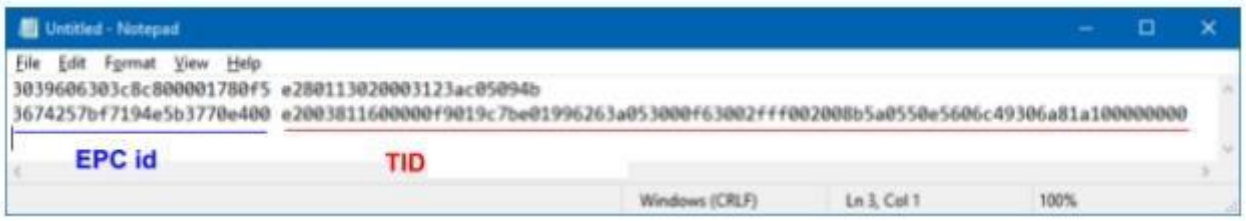


Read result

8.8.4 Read tag TID and EPC id memory bank

With the AcuPad-50 Mux reader it is possible to read any memory bank of the EPC gen2 tag, (reserved, epc,tid and user), just configure the tagop command in which bank you want to read together with the initial position and size of the data (in word) that you want to read.
The syntax of the tagop command is explained in detail in 8.5.4.13 tagop - Tag read operation
Examples:

A. Read the tag's entire TID bank plus the EPC id in hex:
   tagop tid:0:0<CR>
   epcdecode none<CR>
   Example of reading two tags with different TID bank sizes:



B. Read from the TID bank 4 words a skipping the first 2 words and decode the EPC id from the table ASCII:
   tagop tid:2:4<CR>
   epcdecode ascii<CR>
   Example of reading two tags:

In addition to the TID bank, the RESERVED, USER and EPC banks can also be read together with the EPC tag id.

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

NOTE:	For the Reserved bank, if the tag has an access password recorded other than 0x00000000, the tag password must be configured in the reader by the command **gen2accesspassword**.

NOTE:	More details on the commands in 8.5 Configuration commands.

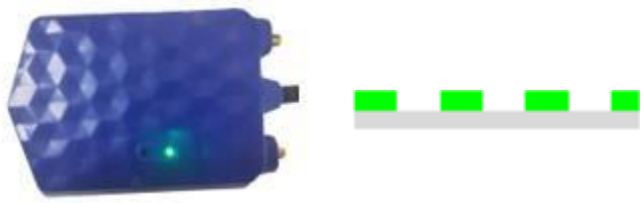
Part 3 - Transparent mode

Reader in transparent mode
Programming and software development
Software demo - URA
Led and beep control

9 Reader in transparent mode

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

Visual reader feedback when in Standalone mode:

	**Green led flashes slowly and Red led off**

By factory default, the scanner's operating mode is Standalone mode, to switch to Transparent mode please consult:
8.6 Changing autonomous mode for transparent and vice versa
8.5.4.1 readmode - Read mode
The red led and the beep of the reader are now controlled by the User Software through the manipulation of two GPIOs available in MercuryAPI, more details in 9.2.2 Led and beep controlling
The communication between the User Software and the reader is serial through the port created by the System Operational when the reader is connected to the USB, to identify such port, please consult 7.1 Turning on the reader.

NOTE: In Transparent mode, the keyboard-type HID 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 C#/.Net, Java and C languages, 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 files package there are several examples and source codes in all languages.
This development package has no additional cost and is open source.

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

9.2 Software demo - URA

The reader in Transparent mode can be tested using the “Universal Reader Assistant” software from Jadak, the software download and its manual are available from the links below:

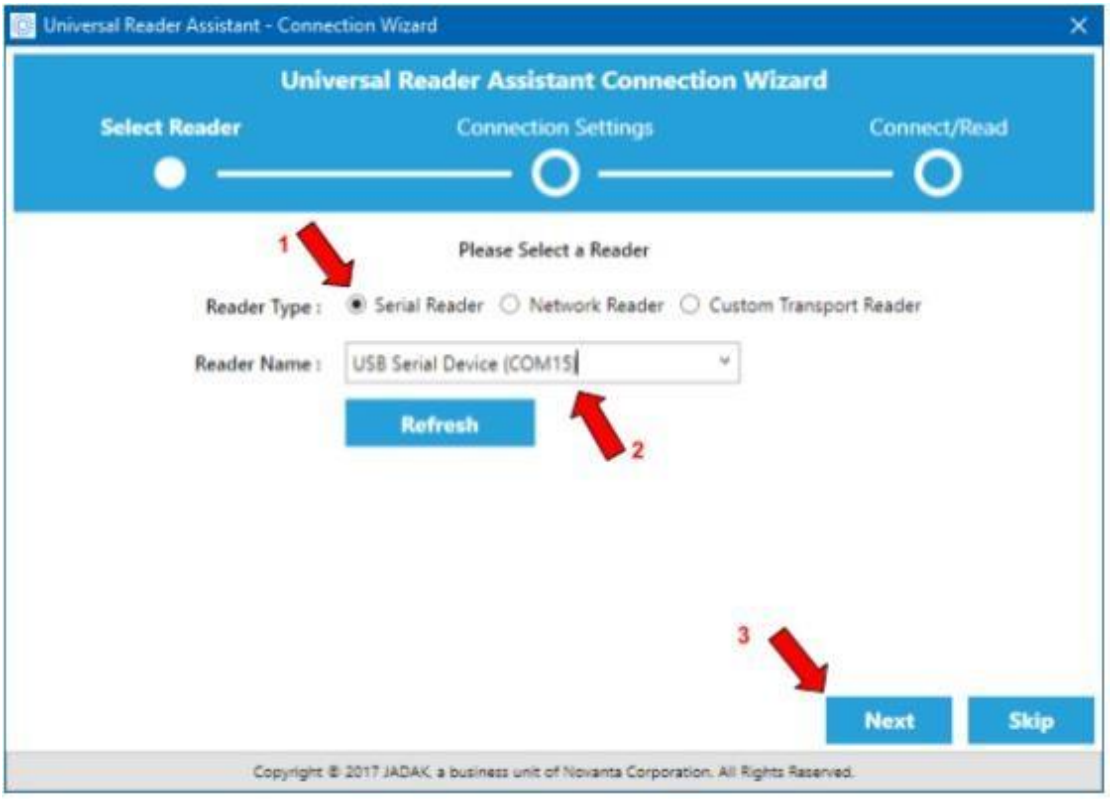
User guide:
https://www.jadaktech.com/products/thingmagic-rfid/thingmagic-universal-reader-assistant/

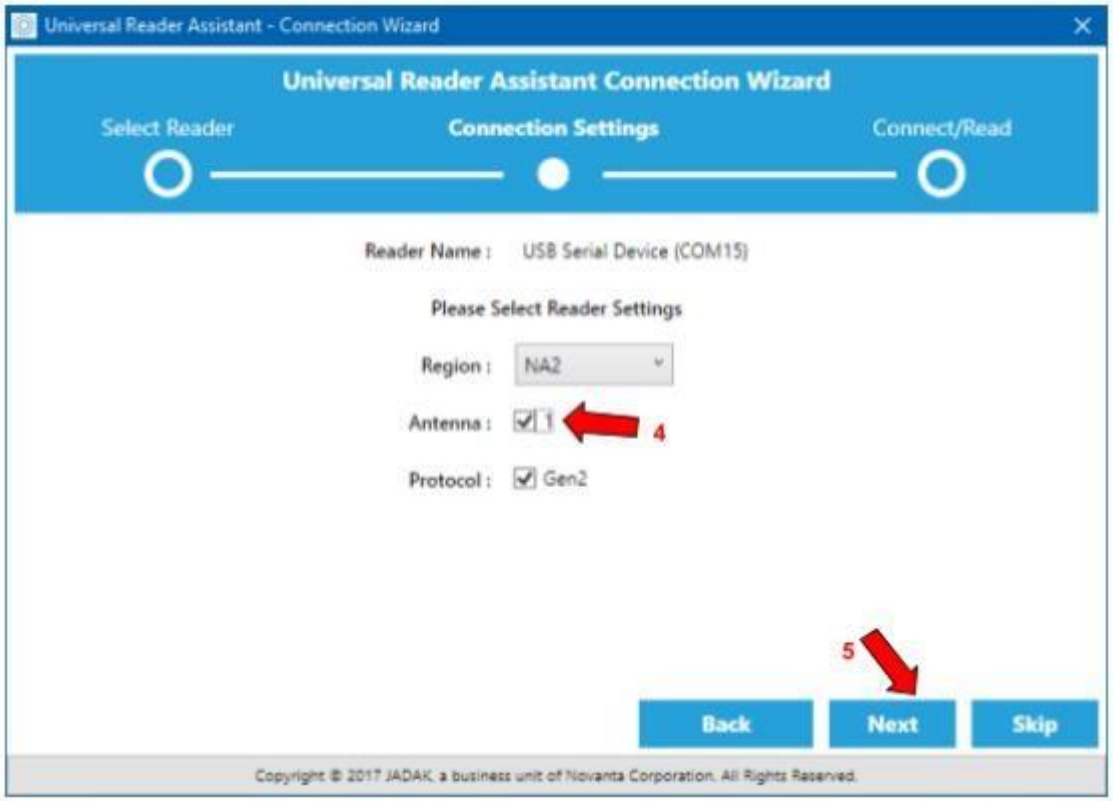
Software URA:
https://www.jadaktech.com/products/thingmagic-rfid/thingmagic-universal-reader-assistant/

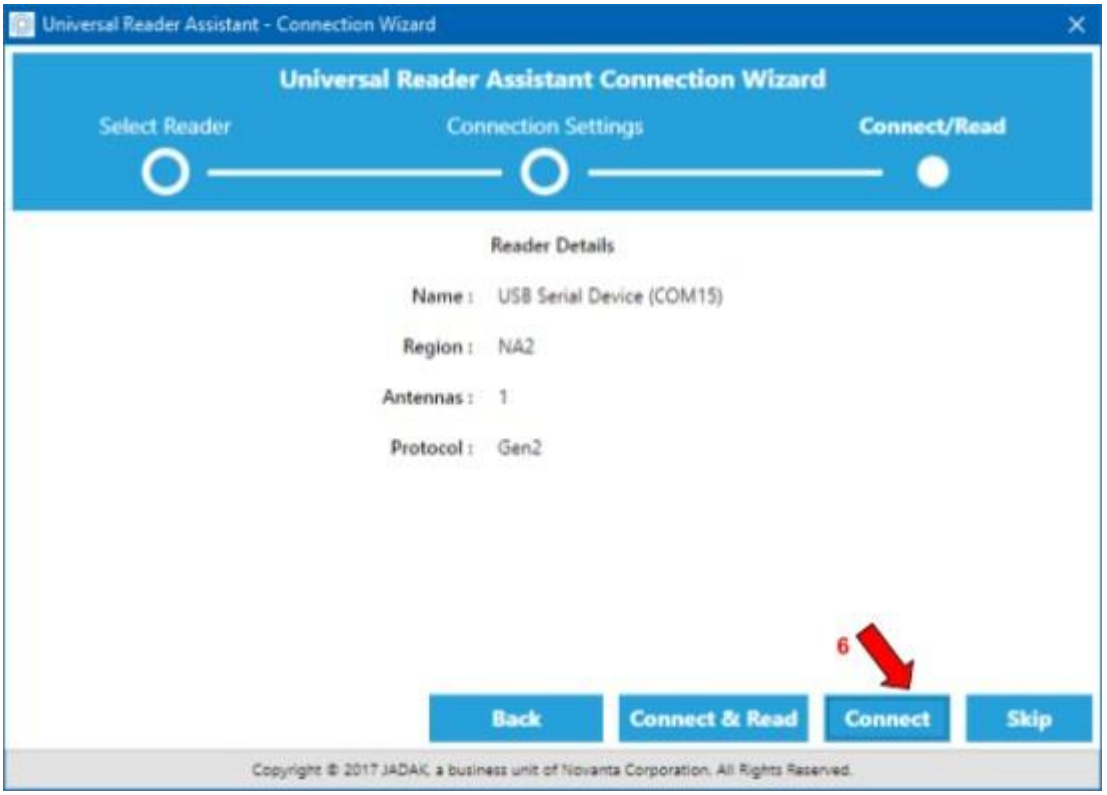
9.2.1 Software demo instructions

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

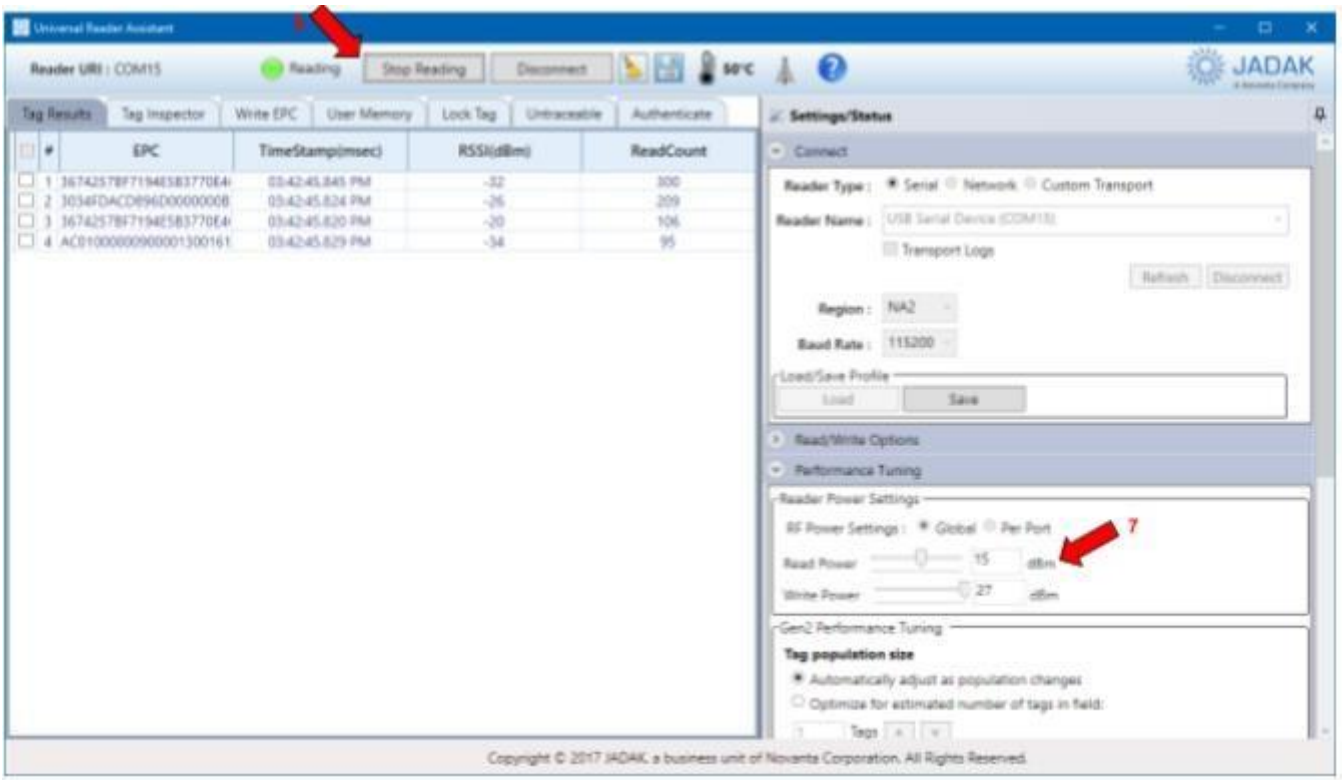
Run “Universal Reader Assistant”.




Universal Reader Assistant - Connection Wizard
Universal Reader Assistant Connection Wizard
Select Reader
Connection Settings
Connect/Read
Reader Name : USB Serial Device (COM15)
Please Select Reader Settings
Region : NA2
Antenna : 1
4
Protocol : Gen2
5
Back
Next
Skip
Copyright © 2017 JADAK, a business unit of Novanta Corporation. All Rights Reserved.


Universal Reader Assistant - Connection Wizard
Universal Reader Assistant Connection Wizard
Select Reader
Connection Settings
Connect/Read
Reader Details
Name : USB Serial Device (COM15)
Region : NA2
Antennas : 1
Protocol : Gen2
6
Back
Connect & Read
Connect
Skip
Copyright © 2017 JADAK, a business unit of Novanta Corporation. All Rights Reserved.

Once connected, reduce the reading power to 15dBm if the reader is powered only by a USB port and then Read.

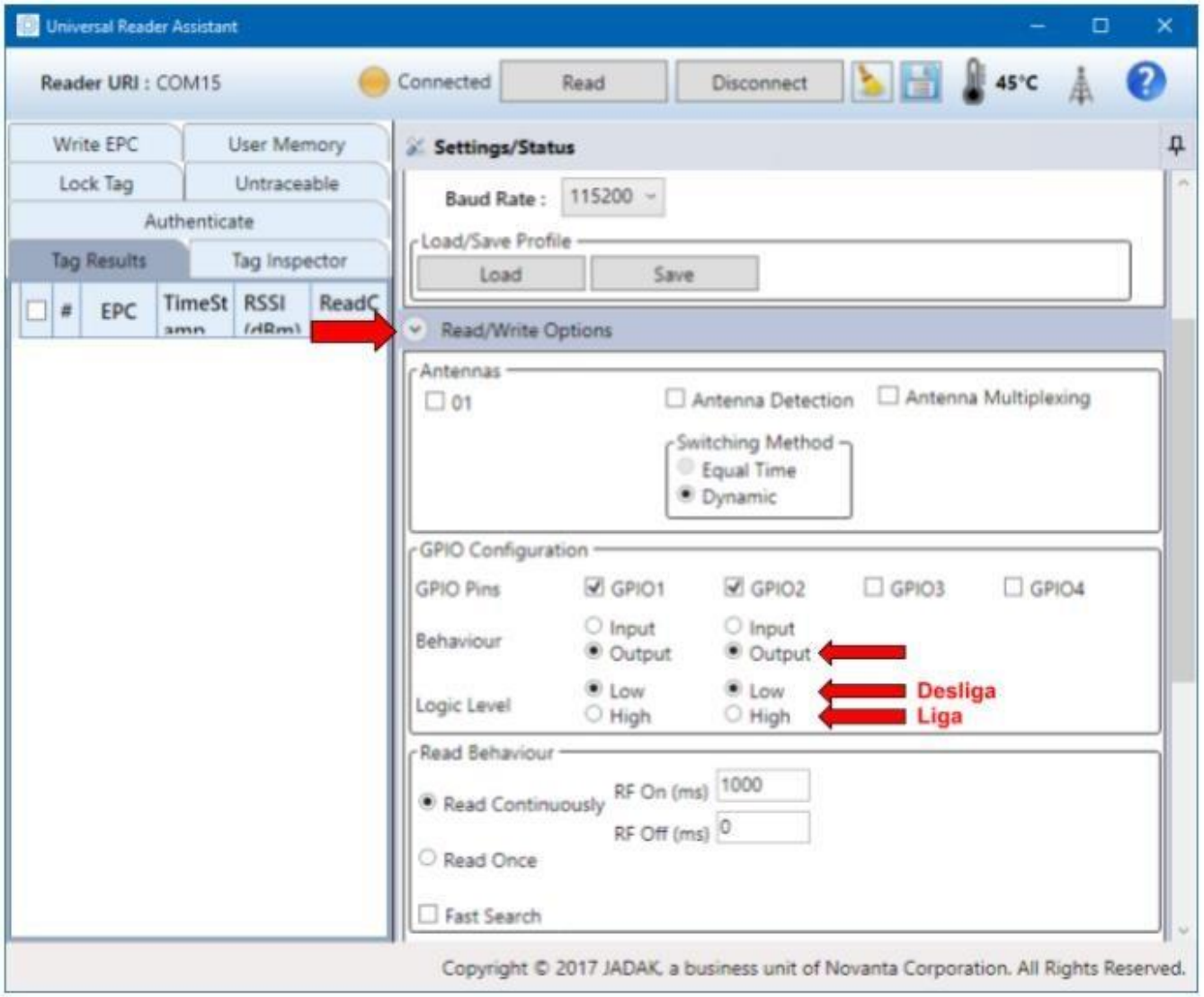


9.2.2 Led and beep controlling

In Transparent mode, the control of the red LED and the Beep must be done by the User Software using the Mercury API.
Below is the usage map for the respective GPIOs:

Out #	Feedback type	Drive (on)
GPIO1	Red led	High (High level)
GPIO2	Beep	High (High level)

Below is the image using the URA to manipulate the red led and the reader's beep in mode Transparent.



NOTE: In the MercuryAPI file package for all languages there is an example for manipulation of the reader's GPIO called “**gpiocommands**”.

Part 4 - Upgrade firmware

How to upgrade the reader firmware? (Windows)

10 How to upgrade the reader firmware (Windows)

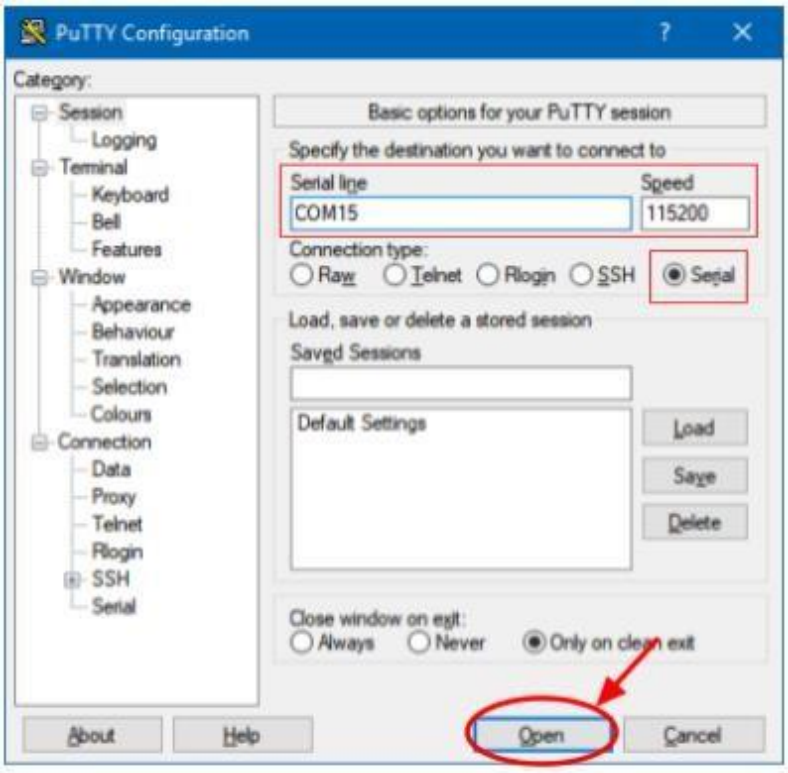
10.1 Upgrade firmware in the serial port

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

- If you have the “Putty” software installed on your machine. If you don't already have it, please see 8.4.1.3 How to send commands to the reader? (Windows) to continue;
- in “Standalone” mode. To find out what mode the reader is in, please see 7.1. Turning on the reader;
- Which serial port will the reader respond to when connected to your computer. to know in which serial port the reader will respond, please refer to 7.1.1. Identifying the serial port (USB) (Windows).

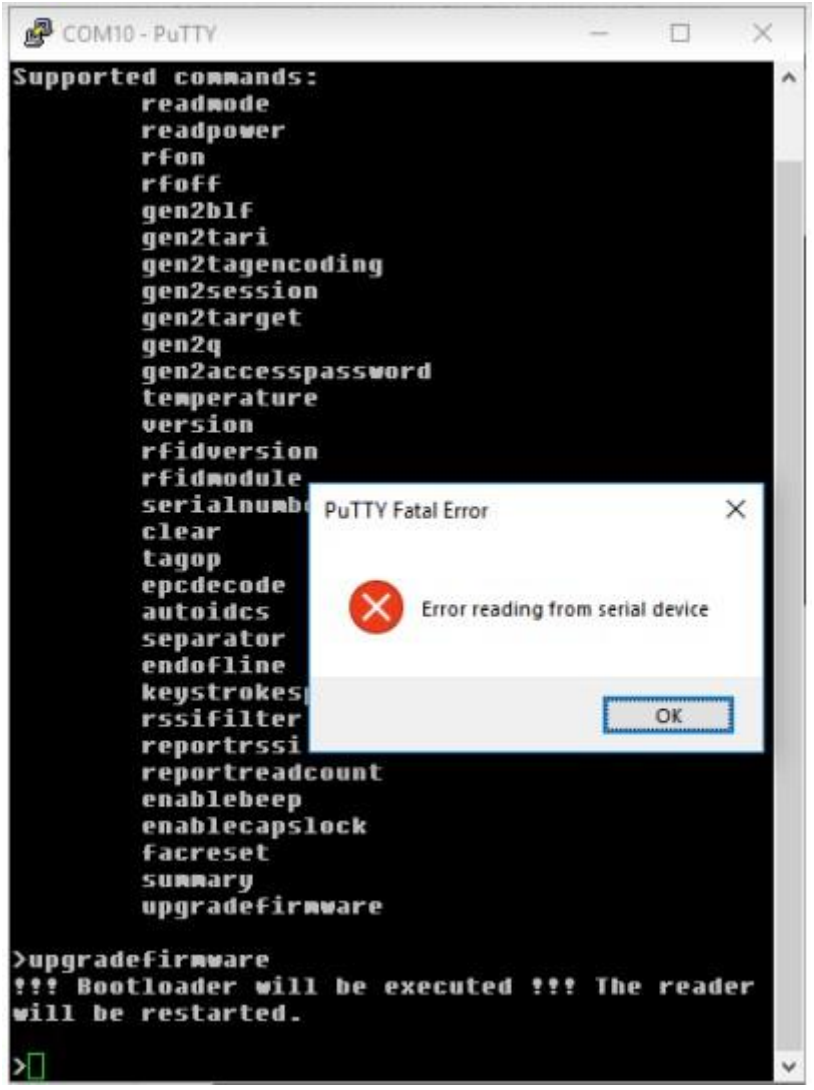
10.1.1 Step 1: Send the update command for the reader

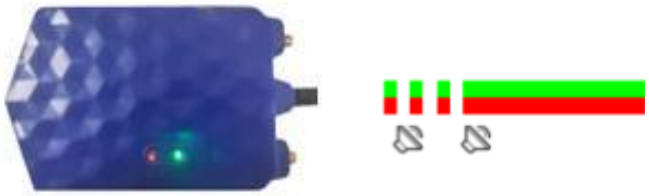
1. Open the Putty;
2. Select “Connection type:” “Serial”, “Serial line” the port that the reader generated and “Speed” 115200;
3. Click in “Open” to open the terminal screen to enter the commands;



4. Enter “**help**” without argument to check if the reader is connected correctly;

5. Type “**upgradefirmware**” without an argument and the following screen should appear;

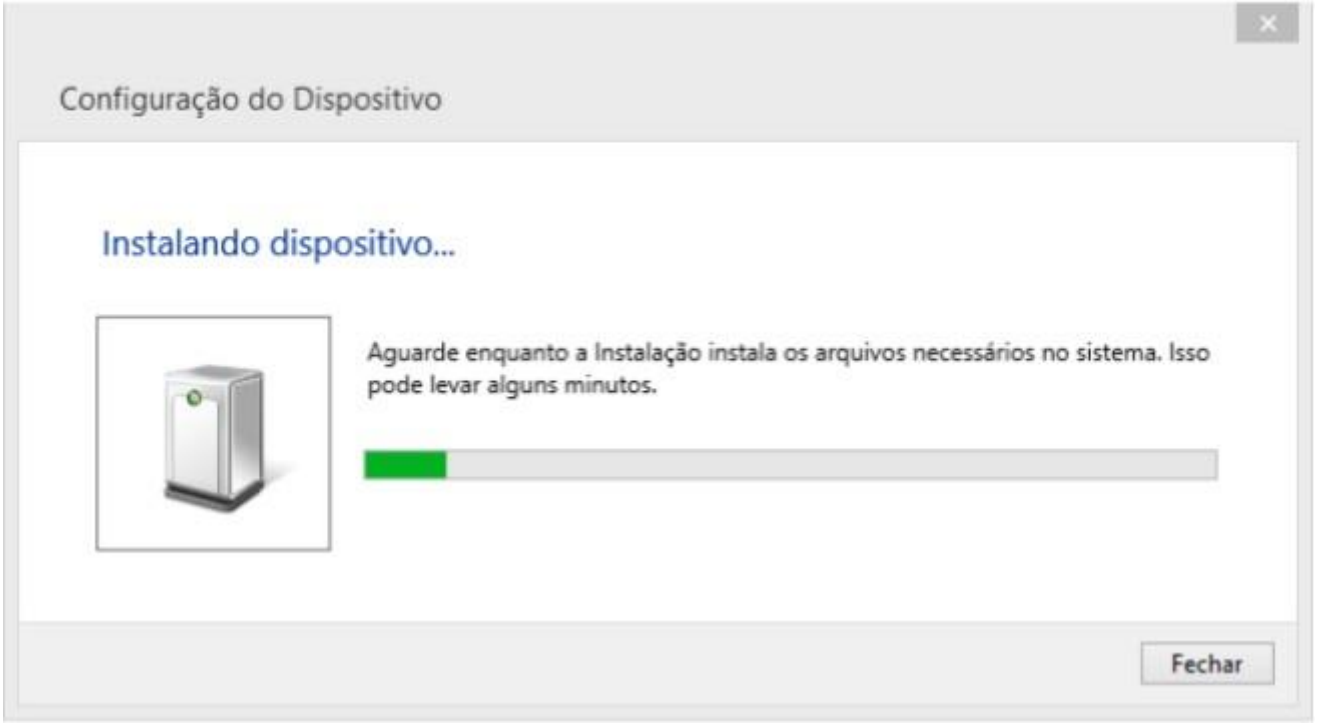


	Beep and green and red LED will flash fast simultaneously 3 times and then the green and red led will remain lit continuously to let you know that you are in the process of firmware update.

NOTE: At this moment the reader is in “**Firmware Update**” mode, therefore, no disconnect the player from the computer until the update process is complete.

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

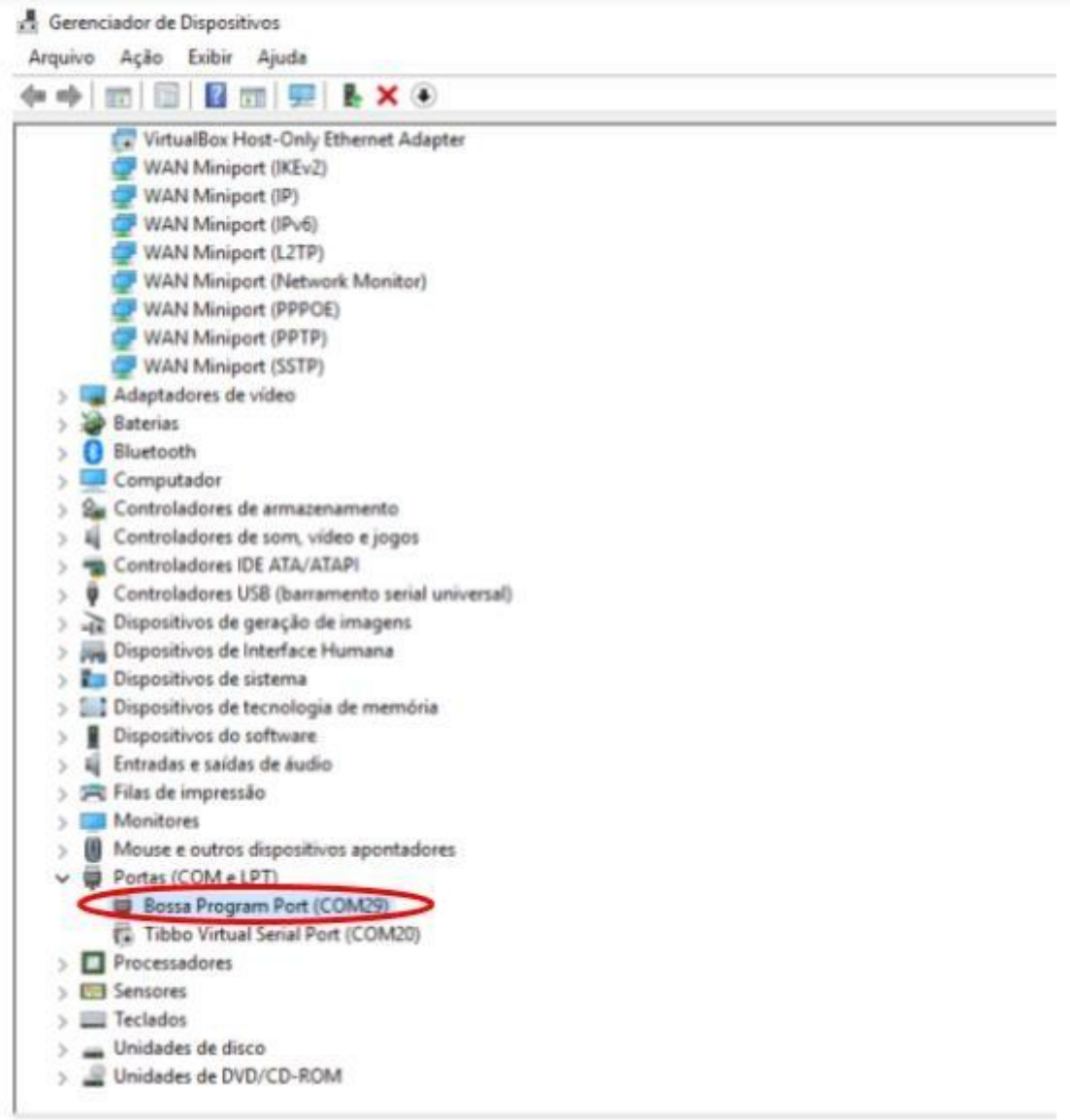
Windows 8 - OK



Windows 7 - OK

NOTE: If you are using Windows 10 the driver will be installed automatically.

7. Make sure that the driver has been successfully installed, go to the “Control Panel” or press the “Windows + X” to open the “**Device Manager**”. Under “**Ports (COM and LPT)**”, a port “**Bossa Program Port (COM...)**” should appear.



10.1.2 Step 2: Software installation “Acura Sam-ba” (Windows)

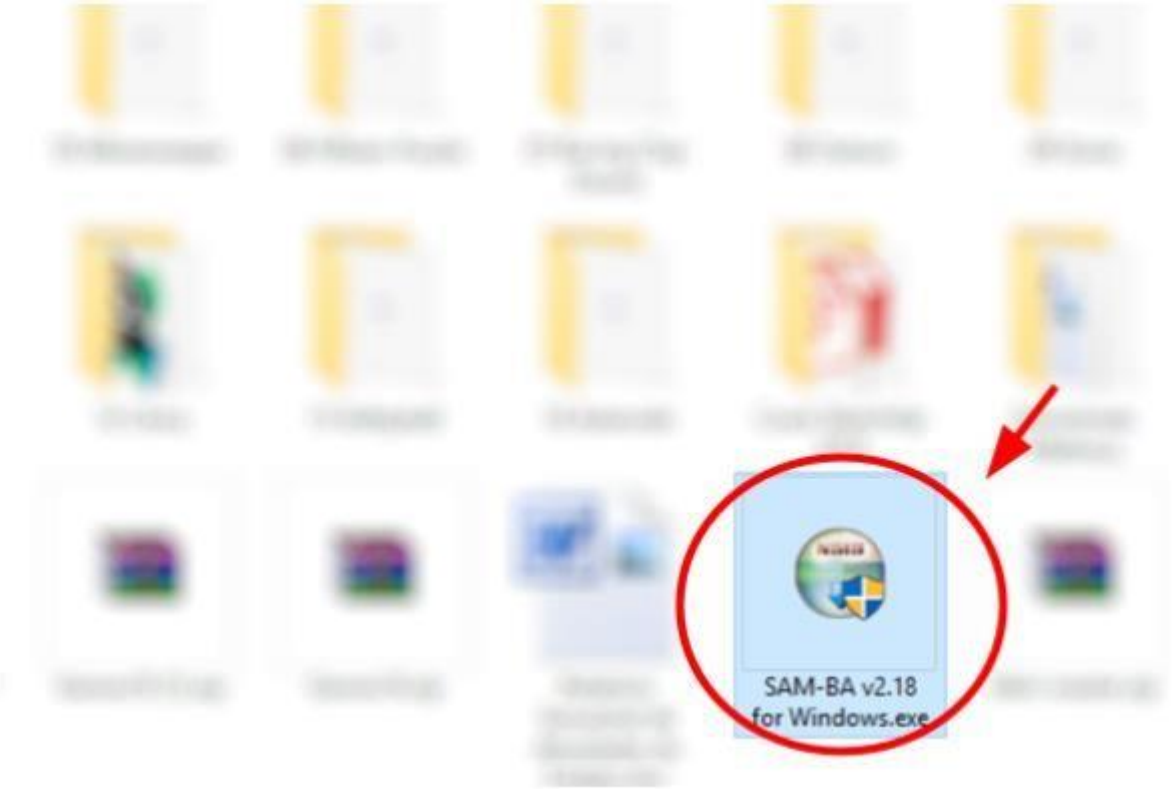
This procedure requires attention and is divided into two steps, running the software installer “**SAM-BA v2.18 for Windows.exe**” and unzipping the folder that will be used in the firmware installation.

10.1.2.1 Step 1: Software installation

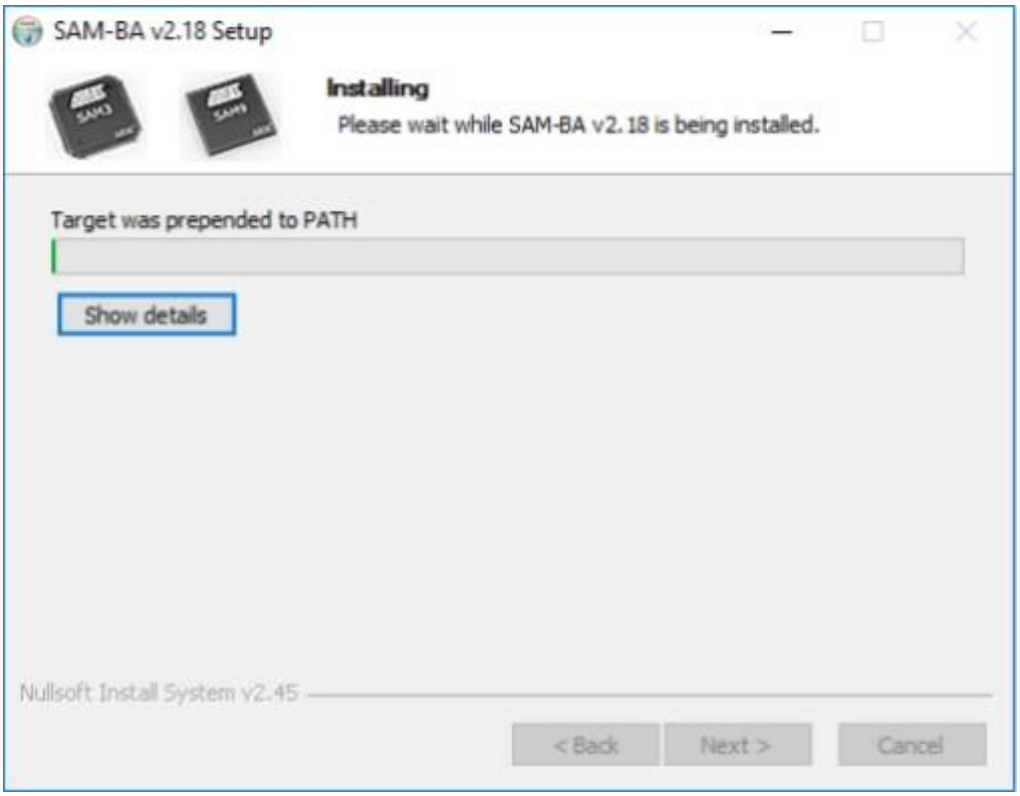
1. Download the software installer from this link: https://drive.google.com/file/d/1VNy4M8sb6WHnMwfciN08O9EEefODngUt/view?usp=share_link

2. Go to downloads, or where the file was saved and run the installer. a message from “Do you want to allow this app from an unknown vendor to make changes to your device?" 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.

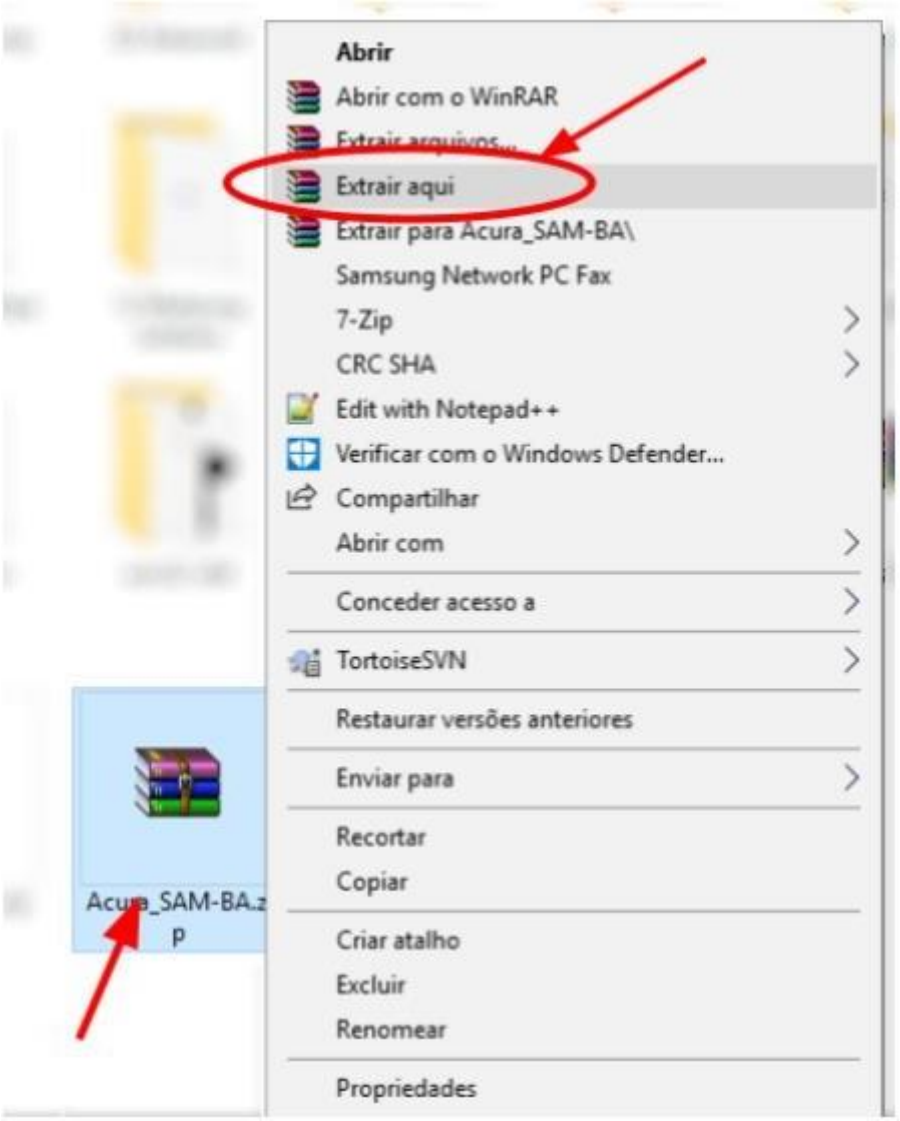


4. After installation click “Next >”, “Next >” and “Finish” to finish.

10.1.2.2 Step 2: Unzip the software folder

1. Download the software zipped folder from the link: https://drive.google.com/file/d/12jV9kI6L37pQ_ZvzLTr7K2HctsLPvBGB/view?usp=share_link

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

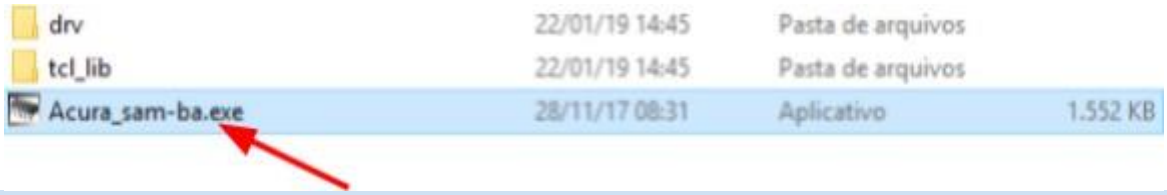


WARNING: It is extremely important to carry out the two steps carefully.

10.1.3 Step 3: Load the firmware

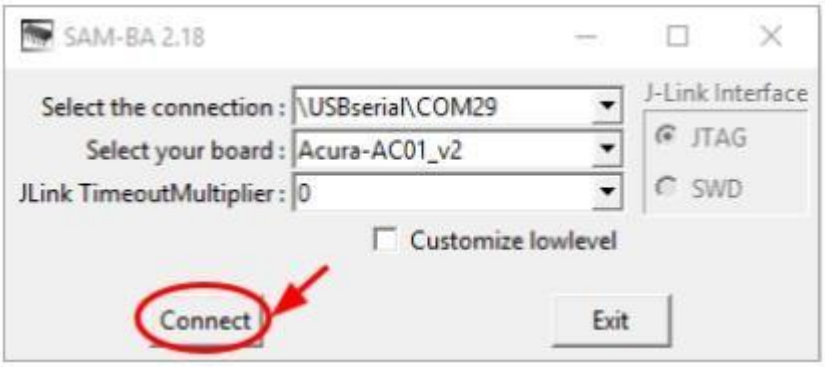
The software that will be used for the firmware update will always be the one inside the folder "**Acura-SAM-BA**" which was downloaded and unzipped in the 10.1.2.2 Step 2: Unzip the software folder.

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

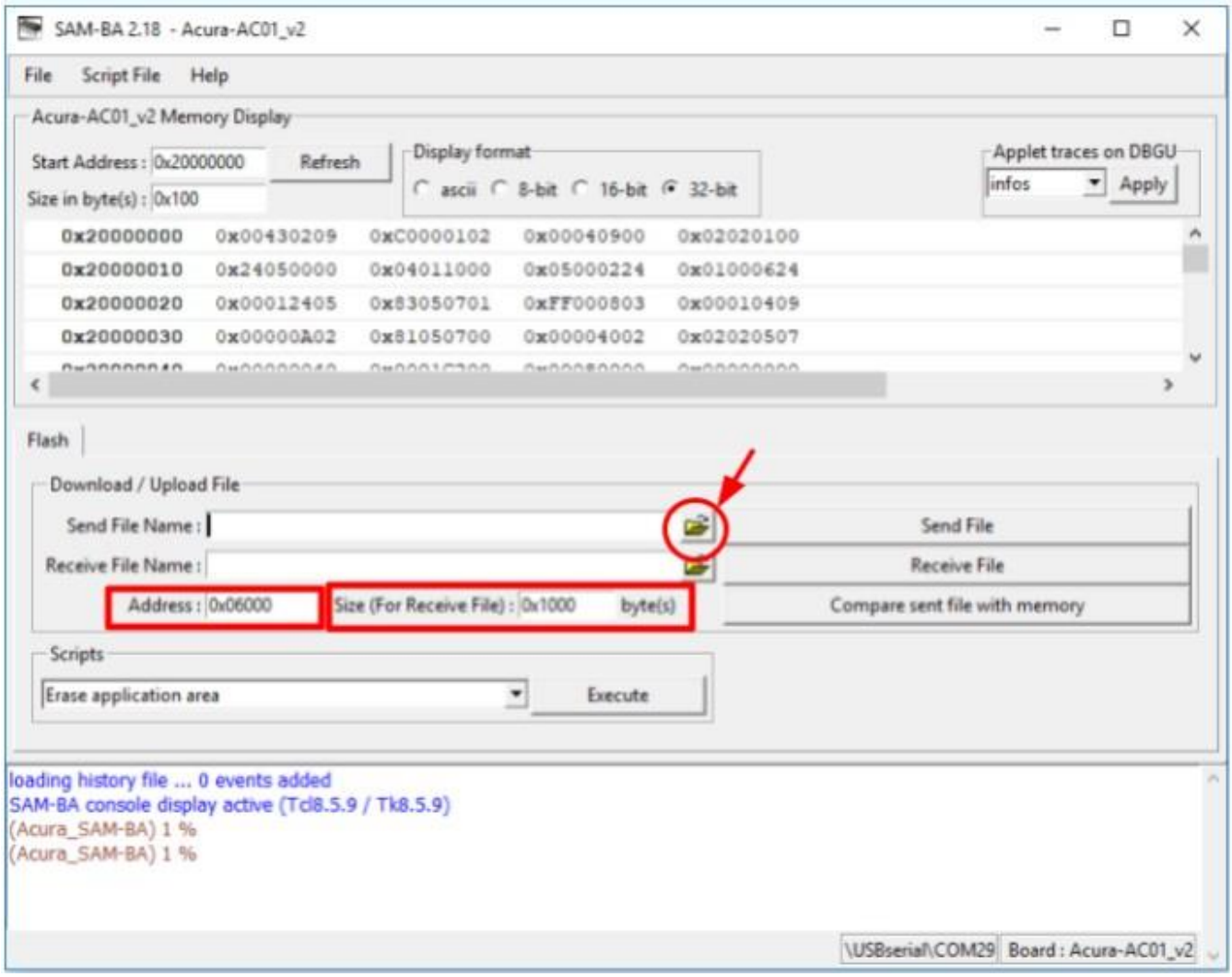


NOTE: This software will only work if you have run and installed the software "**SAM-BA v2.18 for Windows.exe**" informed in the section 10.1.2.1 Step 1: Software installation.

2. Click in "**Connect**"

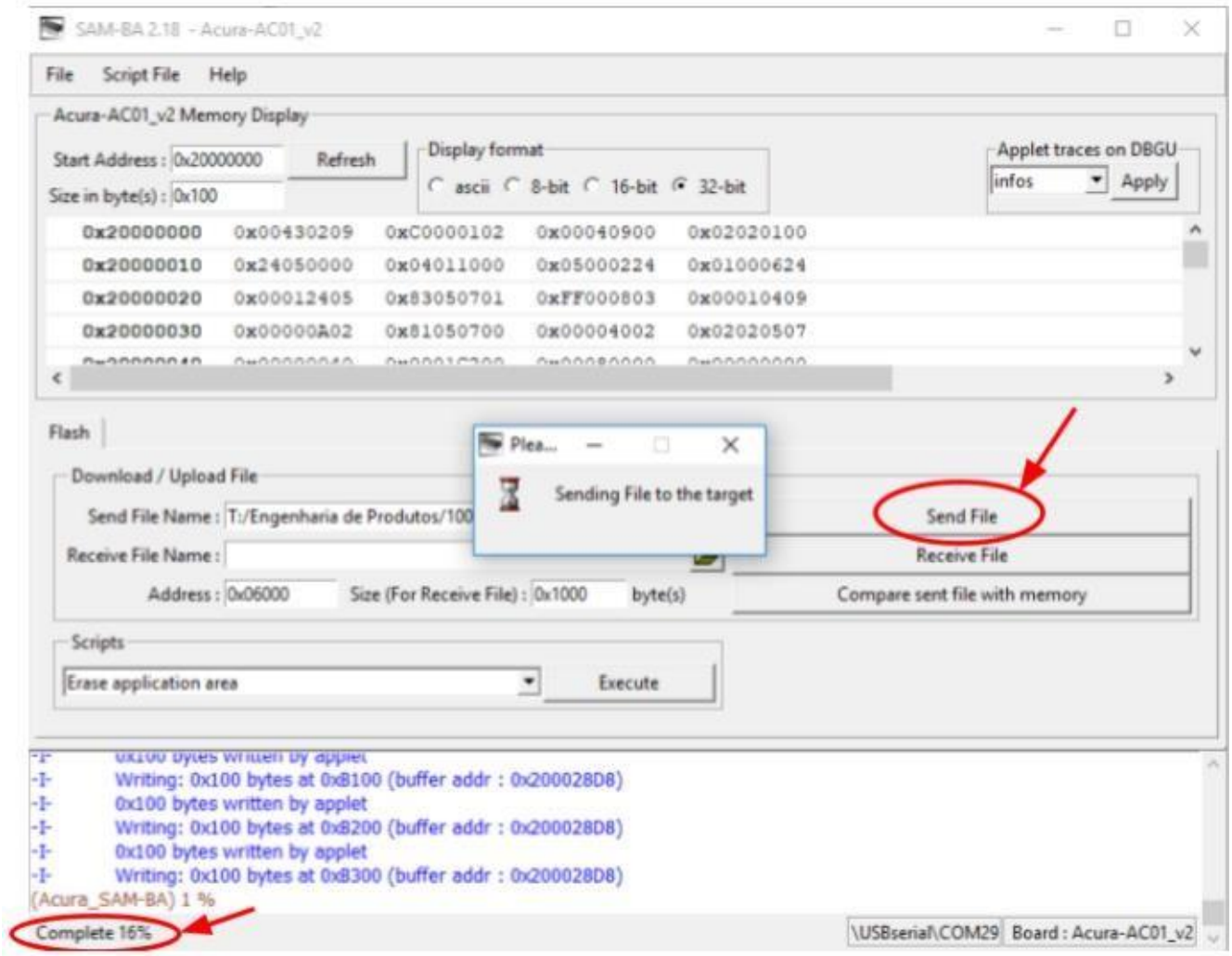


3. Click the “**Send File Name:**” icon and select the updated firmware file;



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

4. Make sure the parameters “Address:” “0x06000” and “Size (For Receive File):” “0x1000” are in agreement and send the updated firmware to the reader by clicking on “Send File” and wait for the procedure to complete 100%;



NOTE: It normally takes 10 seconds to update the firmware.

5. Unplug the scanner and plug it back into the USB port and the scanner will work again normal with the updated firmware.

WARNING: If there is a problem during the update and the reader is not calling or is not responding to commands, please update the reader manually by the internal button, according to the next section.

Part 5 - Regional regulation

11.1 Frequency and operation region

On startup of the AcuPad-50 Mux reader, the operating frequencies (Hop-Table) are configured according tothe region preset in the reader's firmware. for the region of Brazil, the configured frequencies are 902 to907MHz and 915 to 928MHz.

11.2 Reader RF power

For any antenna and cable combination used with this player, the maximum RF power emitted by the reader must be determined by means of the following equation:

Pmax = 36 dbm[5] –	Antenna gain[6] +	Cable loss[7]

The maximum configurable power on the reader is 30 dBm.
Thus, with the antenna and cable data, the power to be configured in the reader is calculated to its use, keeping in mind the maximum and minimum of the reader and respecting the norms regulations in the region where the reader is being used.

WARNING: It is the responsibility of the user of the equipment to correct configuration of the RF output power to comply with the standards regional RF emission.

ACURA Global is the pioneer in the Radio Frequency Identification (RFID) market in Brazil and Latin America, and has successfully pioneered, since the late 1990s, its wide adoption scale in the most diverse sectors of the economy, from mining to steel, from agriculture to food processing, from logistics to retail, from transport to the distribution chain, from control access to asset management. Promoter of new technologies, innovative, agile and focused on viability of cutting-edge projects.

Commercial and Development Office
Av. Antártico, 381 14° andar
São Bernardo do Campo – SP
CEP: 09726-150
11 3028-4600

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

www.ACURA.com.br