identium®
Creating identities for all

USER MANUAL

UDR-111

UHF Desktop Reader

USB reader-writer, Identium Desktop Reader and the SDK — set-up, daily use and troubleshooting



THIS MANUAL COVERS	VERSION	PLATFORM
UDR-111 reader (model IDTS-111UDR)	—	USB to a Windows PC
Identium Desktop Reader	1.4.6	Windows 10/11 (64-bit) and macOS (Apple Silicon and Intel)
Identium UDR-111 SDK — C#, Java, Delphi 7, Visual C++ 2010 samples	1.0	Windows x86 / x64

Document version 1.3 · November 2026

Identium Tech Solutions Pvt. Ltd.
Plot No. 5, First Floor, Raja Dhirsain Marg, Garhi, Sant Nagar, East of Kailash, New Delhi 110065, India
+91 97181 99516 · 011-4714 7839 · WhatsApp +91 82877 32647 · info@identium.in

www.identiumone.com · www.indiarfidstore.com
Creating identities for all

The UDR-111 (model IDTS-111UDR) is a compact USB UHF RFID reader-writer for reading, testing and encoding EPC Gen2 tags at one PC. A single USB cable gives it power and data. It works in two ways: as a USB keyboard that types each tag number into Excel, a web form or an ERP screen, or under the control of software that reads and writes tags. Its software is **Identium Desktop Reader**, Identium's own Windows application, which drives the UDR-111 from version 1.4.0 onwards. This manual covers the hardware, both working modes, everyday work in Identium Desktop Reader 1.4.6 and a quick start for developers using the SDK.

CONTENTS

01/ Equipment Diagram
02/ Connecting and Powering the Reader
03/ Indicators
04/ Two Working Modes
05/ Install Identium Desktop Reader
06/ Connect and Find Your Way Around
07/ Read Tags
08/ Keyboard Mode: Type Tag Numbers into Excel
09/ Write a New EPC to One Tag
10/ Batch Write: Encode a Run of Tags
11/ Reader Settings: Power, Region and Buzzer
12/ For Developers: SDK Quick Start
13/ Integrate the SDK in Your Application
14/ Precautions
15/ Parameters
16/ Troubleshooting
17/ Support

01/ Equipment Diagram



UDR-111 desktop reader with its USB cable

1. **Read area** – the top surface; place the tag here
2. **Light ring** – the reader's LED
3. **USB cable** – power and data from the PC
4. **Housing** – zinc alloy and acrylic, 110 × 70 × 15 mm, 150 g

The reader is powered and controlled through the USB cable. There are no switches or buttons on it: working mode, output format, radio power and the buzzer are all set from the PC with Identium Desktop Reader, or by your own application through the SDK.

02/ Connecting and Powering the Reader

1. Plug the reader's USB cable into a free USB port on the PC. It takes its power from the port; no power supply is needed.
2. Windows recognises it as a standard USB device – no driver has to be installed.
3. To switch the reader off, unplug the USB cable.

TIP

Place the reader flat on the desk beside the keyboard, with the top surface free. Only tags that you want to read should be on or near it.

NOTE

A reader that was left in Keyboard Mode starts typing tag numbers as soon as it has power, with no software running. That is normal – see section 04.

03/ Indicators

Indicator	What it tells you
Buzzer	Sounds when a tag is read. Switch it off, or back on, under **Settings › RFID Settings › Buzzer** in Identium Desktop Reader (section 11); the **Beep** box on the Read tab does the same for a read you start there.
LED	Confirms a tag read, together with the buzzer.

> **NOTE**
> During a batch write the app takes the buzzer over and beeps once per tag written, instead of once per read (section 10).

04/ Two Working Modes

Keyboard Mode – the reader types by itself

The reader acts as a USB keyboard. Present a tag and its number is typed wherever the cursor is: Excel, Notepad, a web form or an ERP screen. No programming is needed, and for the two EPC flavours the reader keeps doing it with the app closed. Format, prefix, suffix and the Enter key are set once in the app. This is the reader's *active* mode.

Driven by software – read and write

The reader answers commands. Use Identium Desktop Reader to read tags, write an EPC, encode a run of tags or lock and kill them, or the SDK to build reading and encoding into your own application. This is the reader's *passive* mode.

The two do not clash. While Identium Desktop Reader reads or writes, it holds the reader passive; Keyboard Mode comes back when you disconnect or close the app, and a reader found in Keyboard Mode still connects.

You want to…	Use
Type tag numbers into Excel or a form	Keyboard Mode – section 08
See what a tag holds	The Read tab – section 07
Write a new EPC to one tag	Write › Manual Write – section 09
Encode a run of tags	Write › Batch Write – section 10
Build reading and writing into your own program	The SDK – sections 12 and 13

05/ Install Identium Desktop Reader

Identium Desktop Reader is Identium's own application for the UDR-111 and the 103 desktop readers. Download it from the Identium Resource Hub, **www.identiumone.com/resourcehub/**, where it is listed four times – two builds for Windows and two for macOS:

Row on the Resource Hub	File	Size
Identium Desktop Reader – Installer (recommended on Windows)	`Identium.DesktopReader.Setup.1.4.6.exe`	91.0 MB
Identium Desktop Reader – Portable	`Identium.DesktopReader.1.4.6.Portable.exe`	90.8 MB
Identium Desktop Reader – macOS (Apple Silicon)	`Identium.DesktopReader.1.4.6.arm64.dmg`	109.8 MB
Identium Desktop Reader – macOS (Intel)	`Identium.DesktopReader.1.4.6.x64.dmg`	117.7 MB

Windows

1. Download either Windows file. The **Installer** adds Start-menu and desktop shortcuts; the **Portable** build is a single file that runs from anywhere, including a USB stick.
2. Run the file. If Windows shows a **Windows protected your PC** notice for a new application, click **More info › Run anyway**.
3. Plug the reader into a USB port (section 02) and start the app. No extra driver is needed.
4. The **Connection** window opens by itself – carry on with section 06.

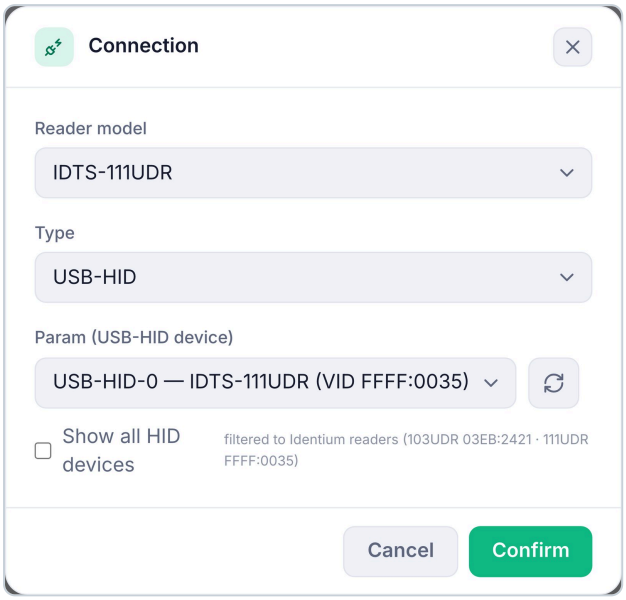
macOS

1. Download the disk image for your Mac: **arm64** for Apple Silicon (M1, M2, M3, M4…), **x64** for Intel Macs.
2. Open it and drag **Identium DesktopReader** into **Applications**.
3. **First launch:** the app is signed but not notarised by Apple, so macOS asks once. Right-click the app › **Open** › **Open**. On macOS 15 and later, if it is blocked, go to **System Settings › Privacy & Security** and click **Open Anyway**.
4. Plug in the reader and launch the app. USB and serial readers need no drivers on macOS either.

Operating system	Windows 10 or 11 (64-bit), or macOS 11 or later (Apple Silicon or Intel)
Connection	A free USB port
Version in this manual	Identium Desktop Reader 1.4.6; the UDR-111 is supported from version 1.4.0

The Resource Hub always carries the current version. Check the version in the app's title bar against the hub row before you report a problem.

06/ Connect and Find Your Way Around

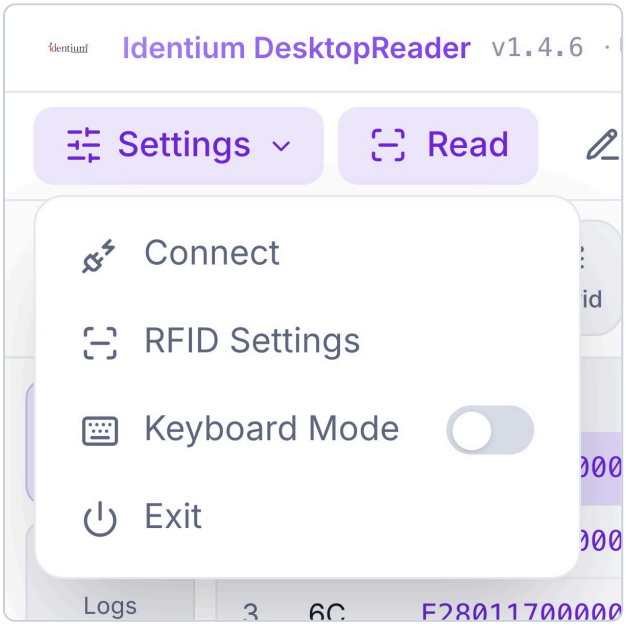


The Connection window with the UDR-111 picked

1. In **Reader model** choose **IDTS-111UDR**. A UDR-111 plugged into USB is recognised and picked for you.
2. Leave **Type** on **USB-HID**.
3. **Param (USB-HID device)** lists the readers found. Pick yours, or press the refresh button beside it. If nothing is listed, tick **Show all HID devices**.
4. Click **Confirm**. The status bar at the bottom reads **Connected: USBHID · USB-HID**, and the title bar reads **UHF Desktop Reader 111**.

> **NOTE**
> The two Identium desktop readers speak different languages, so the app has to know which one it is talking to. On USB it works that out from the device itself (the UDR-111 reports **VID FFFF:0035**).

The window



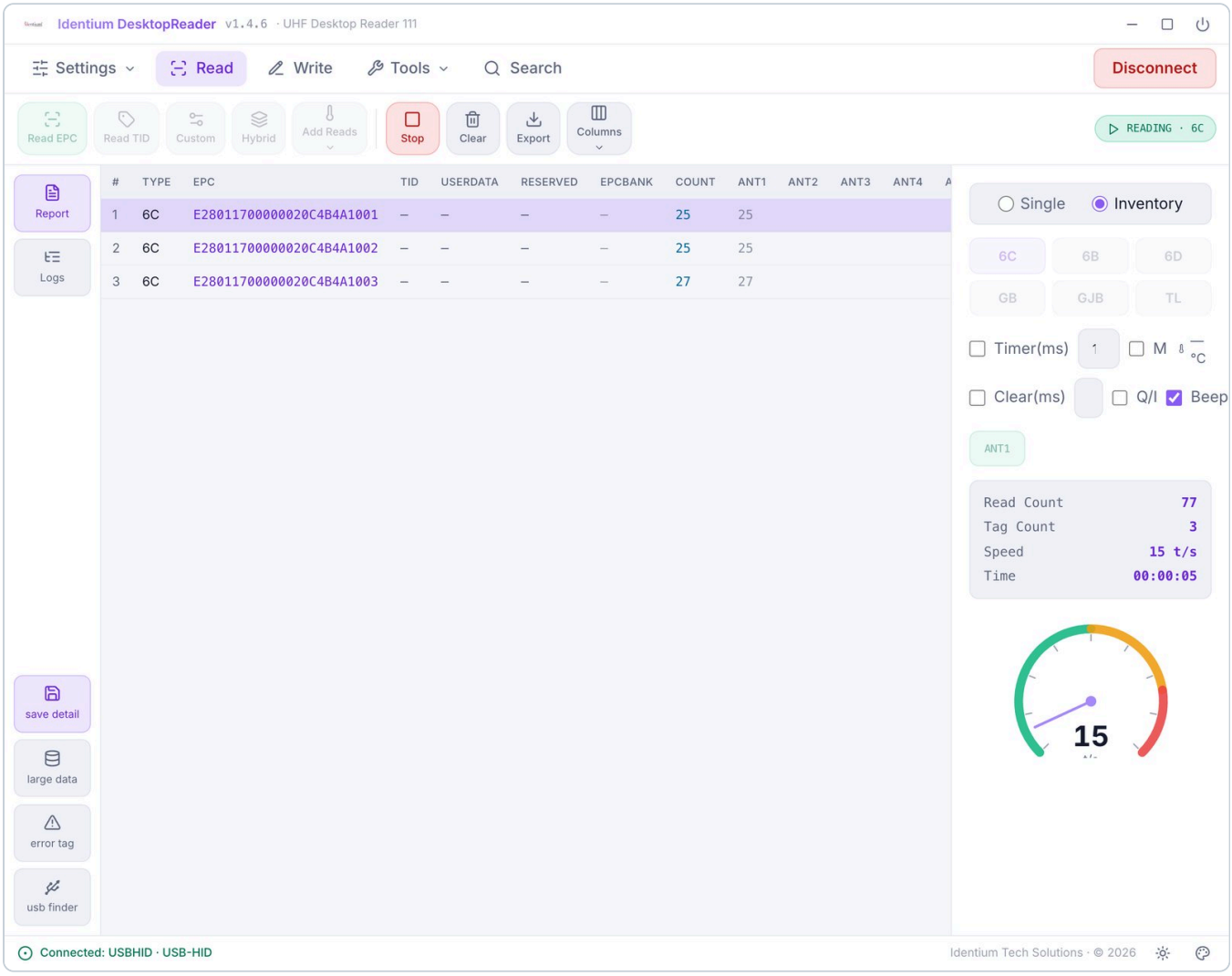
Settings menu with a UDR-111 connected

Menu	What it is for
Settings	**Connect** (the window above), **RFID Settings** (power, region, buzzer – section 11), **Keyboard Mode** (section 08) with a quick on/off switch, and **Exit**
Read	Read tags and watch them arrive – section 07
Write	**Manual Write** for one tag (section 09) and **Batch Write** for a run of tags (section 10)
Tools	Device Info, Custom Command, Clear List, Export Text, Export Excel and Restore Factory Settings
Status bar	Connection state on the left, the last message on the right, and buttons for the light/dark theme and type size

The menus show only what the connected reader can do, so with a UDR-111 the entries for multi-antenna and network hardware are not shown.

07/ Read Tags

1. Click **Read**.
2. On the right, leave **Inventory** selected to keep reading until you stop, or choose **Single** for one read.
3. Place a tag on the reader and click **Read EPC**. The tag appears in the list, and **Read Count**, **Tag Count**, **Speed** and **Time** count up on the right.
4. Click **Read TID** instead to read each tag's TID, the serial number its chip was made with, into the TID column.
5. Click **Stop** when you have finished, **Clear** to empty the list, and **Export** to save the list as a CSV file.



The Read tab: three tags read one after another on the pad, with the live read count, tag count and read rate on the right

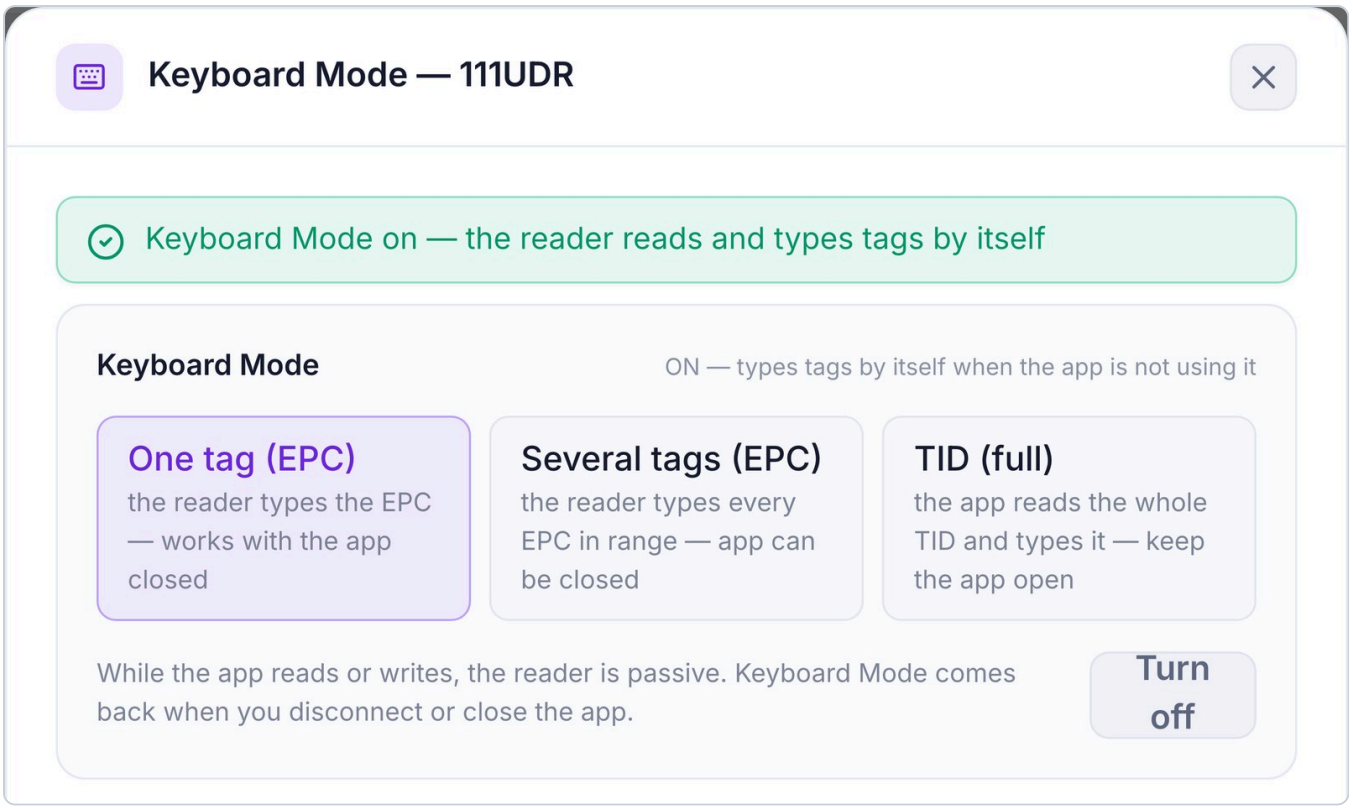
On the Read tab	What it does
Read EPC / **Read TID**	Start reading, returning the EPC, or the EPC and the TID
Custom / **Hybrid**	Read chosen memory areas, and match on a particular EPC, TID or user-memory value
Columns	Choose the columns shown – EPC, TID, user and reserved memory, count, antenna, RSSI, read time and more
6C	The UDR-111 reads EPC Gen2 (ISO 18000-6C) tags, so **6C** is the protocol in use; the other protocol buttons belong to other Identium readers
ANT1	The UDR-111 has one antenna – its read area – so ANT1 is the only antenna
Beep	Ticked, the reader beeps on every tag it reads during this read
Report / **Logs**	Switch between the tag list and the frame log, which shows every command sent and answer received

TIP
The desk pad reads the tag lying on it, one at a time. Lift each tag off before you place the next one, and the list fills up tag by tag.

08/ Keyboard Mode: Type Tag Numbers into Excel

In Keyboard Mode the reader types each tag number wherever the cursor is, exactly as if you had typed it. Nothing has to be installed in Excel, the browser or your ERP.

1. Open **Settings › Keyboard Mode**.
2. Choose what should be typed: **One tag (EPC)**, **Several tags (EPC)** or **TID (full)**.
3. Click **Turn on**. The banner at the top of the dialog confirms it.
4. Close the dialog, click in the cell, box or field where the first number should go, and place a tag on the reader. The buzzer sounds and the number is typed at the cursor.



Settings › Keyboard Mode, switched on with One tag (EPC) chosen

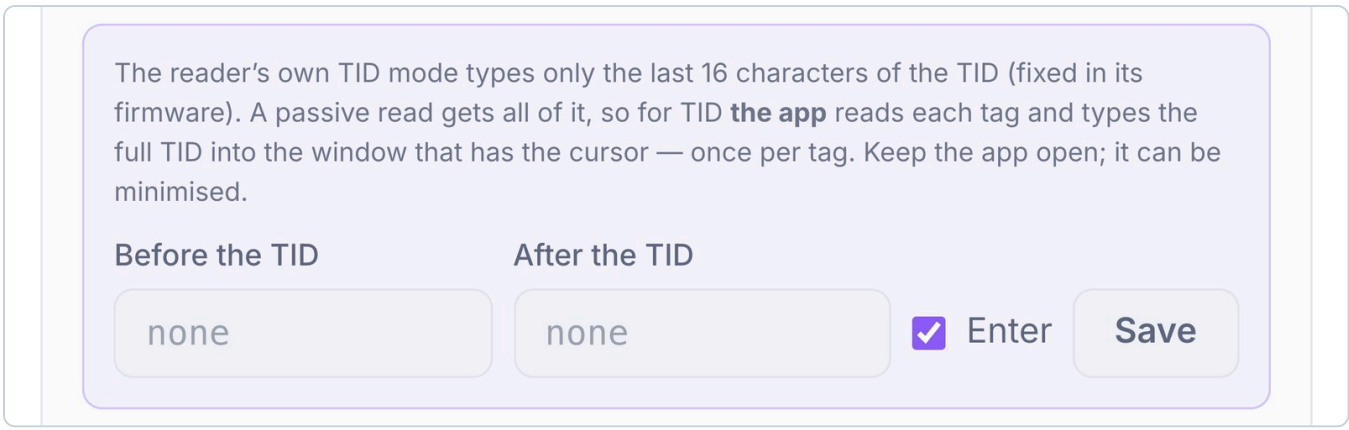
What is typed	Use it for	Does the app have to stay open?
One tag (EPC)	The EPC of the tag on the pad – the everyday setting	No – the reader does the typing itself
Several tags (EPC)	The EPC of every tag in range, one after another	No
TID (full)	The complete TID of each tag – its factory serial number	Yes, but it may be minimised

Typing the full TID

The reader's own TID typing is fixed in its firmware and sends only the last 16 characters of the number. To get all of it, choose **TID (full)**: the app reads each tag itself and types the whole TID, once per tag. Keep Identium Desktop Reader open (minimised is fine). **Before the TID**, **After the TID** and **Enter** set what is typed around the number; click **Save** to keep them.

> **NOTE**
> On a Mac, **TID (full)** needs permission to control the computer, which macOS asks for once: allow **Identium DesktopReader** under **System Settings › Privacy & Security › Accessibility**.

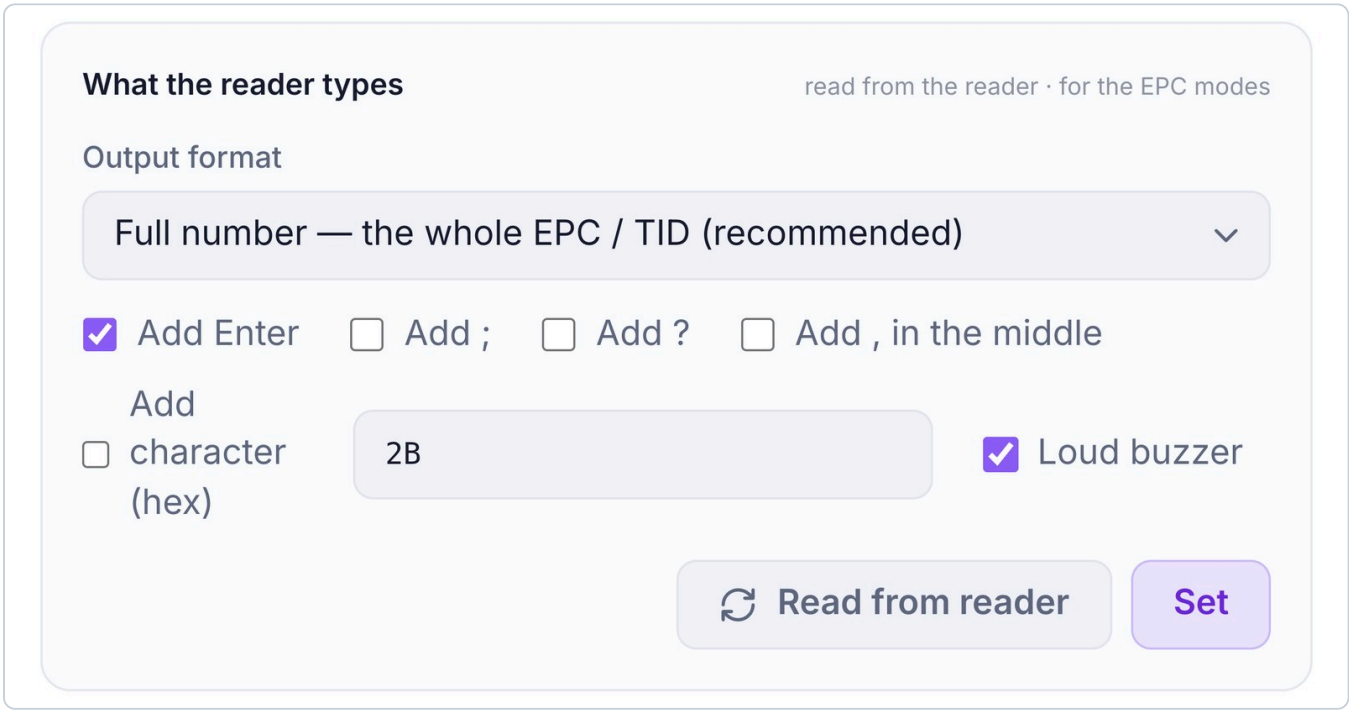
Check the TID, in the section below, reads the tag on the pad with the app and shows the TID and EPC it really holds, so you can compare them with what was typed.



With TID (full) chosen, these boxes set what the app types around each TID

What the reader types

5. Under **Output format** choose **Full number – the whole EPC / TID**. The other formats are short access-card numbers made from the last few bytes of the tag number; the app warns you when one of them is chosen.
6. Tick **Add Enter** so the next number goes to the next cell or line. **Add ;**, **Add ?**, **Add , in the middle** and **Add character (hex)** add those characters to the typed output; check the result in Notepad.
7. **Loud buzzer** sets how loudly the reader beeps as it types.
8. Click **Set** to send the format to the reader. **Read from reader** shows what the reader has stored.
9. Under **Prefix & suffix**, up to four characters each can be typed before and after every tag number. Enter them and click the **Set** button beside each box.



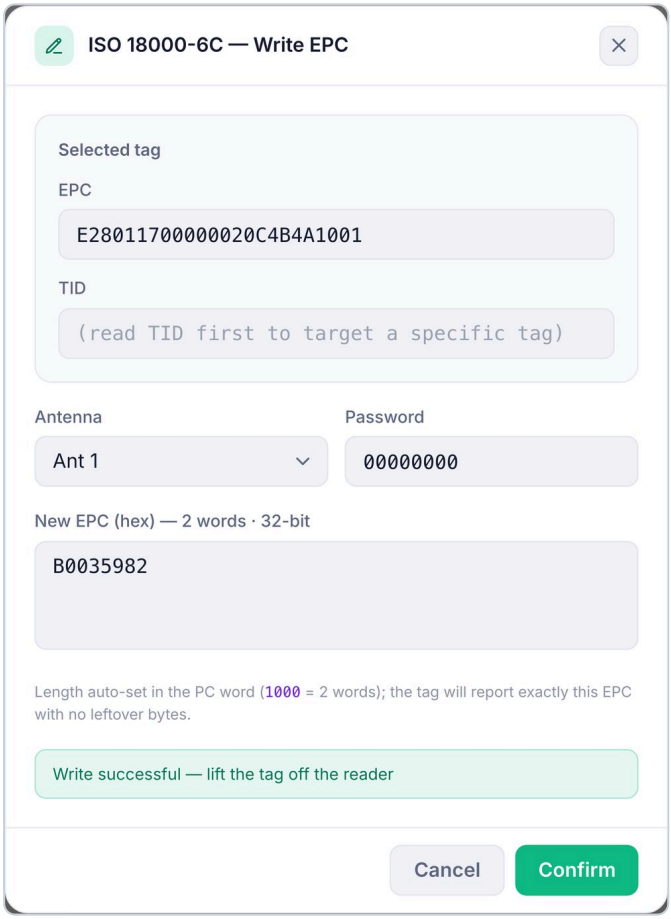
What the reader types: the output format and the characters added to each tag number

TIP

Before a rollout, test the chosen format with a few of your own tags in Notepad, and check the typed numbers against a read on the Read tab.

09/ Write a New EPC to One Tag

1. Place **one** tag on the reader and keep other tags away from it.
2. Click **Read** and then **Read EPC** so the tag appears in the list, then click its row to select it. Click **Stop**.
3. Click **Write**, then **Manual Write**, then the green **EPC** button.
4. The **Selected tag** box shows the tag you picked. Leave **Antenna** on **Ant 1** and **Password** on **00000000** unless the tag has an access password.
5. Type the new EPC in **New EPC (hex)**, in hexadecimal. It must be whole 16-bit words – four hex characters per word – so 8, 12, 16 or 24 characters, up to 31 words. The app shows the word count as you type and sets the tag's own length field to match.
6. Click **Confirm**. **Write successful – lift the tag off the reader** appears under the box.



Write EPC: a four-byte EPC written to the selected tag

CAUTION

Writing replaces the tag's EPC until it is written again. Check the number before you click **Confirm**, and write one tag at a time.

The other write tools

Button	What it does
User	Writes data to the tag's user memory, from a word address you choose
Custom	Writes any memory area, and **locks** or **destroys (kills)** a tag
Super	Reads every memory bank of the tag at once, then writes any one of them

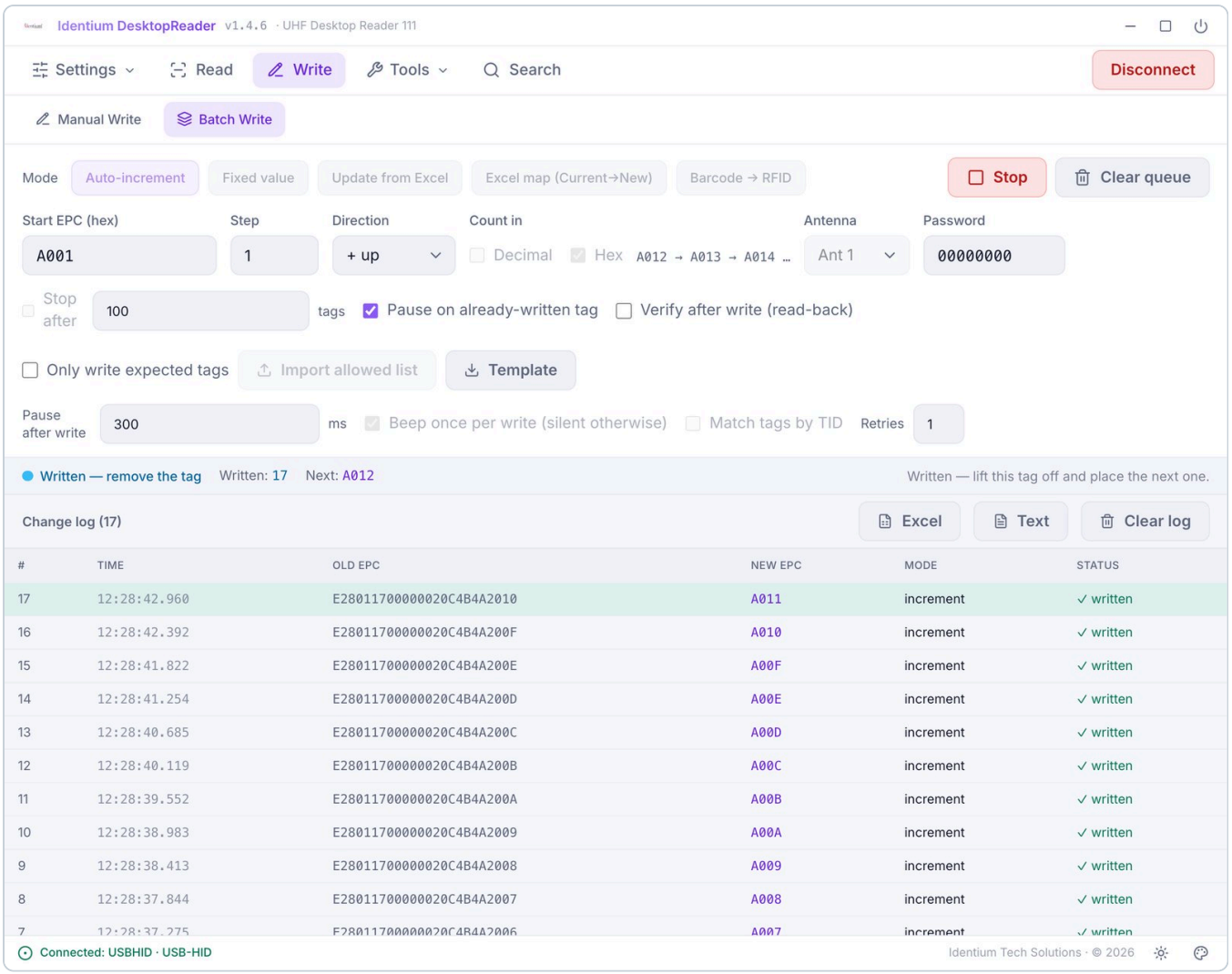
CAUTION

Destroy and the **Permanent lock** / **Permanent unlock** actions cannot be undone. Try them on spare tags first.

10/ Batch Write: Encode a Run of Tags

Batch Write encodes tags one after another without a read step: arm it once, then place a tag, lift it off, place the next. Each write is logged with the old and the new number.

1. Click **Write**, then **Batch Write**.
2. Choose a **Mode**: **Auto-increment** for a numbered series, **Fixed value** to write the same number to every tag, **Update from Excel** for a column of new EPCs, **Excel map (Current→New)** to renumber known tags, or **Barcode → RFID** to turn a scanned barcode into the tag's EPC.
3. For **Auto-increment**, set the **Start EPC (hex)**, the **Step** and the **Direction**, and whether counting runs in **Decimal** or **Hex**. The next few numbers in the series are previewed beside the boxes.
4. Leave **Password** at **00000000** unless your tags have an access password.
5. Click **Start**. The bar under the controls reads **Armed – place a tag**.
6. Place a tag. It is written as soon as it answers, the reader beeps once, and the bar reads **Written – remove the tag**. Lift it off and place the next one.
7. Click **Stop** when the run is done. **Excel** and **Text** save the change log.



Batch Write in Auto-increment mode, part-way through a run: the change log lists every tag written, newest first

Option	What it is for
Stop after … tags	Ends the run after a set number of tags
Pause on already-written tag	Stops and asks before a tag that already carries a number from this run is overwritten
Verify after write (read-back)	Reads each tag back and compares it with what was written
Only write expected tags	Writes only tags whose current number is in a list you import
Pause after write	How long to wait after each write before looking for the next tag (300 ms by default)
Beep once per write	The app takes the buzzer over and beeps once per tag written, instead of once per read
Match tags by TID	Reads the TID as well, so a tag that half-took a write can be told from a fresh one. It costs an extra read per tag
Template	Saves a sample Excel sheet with the columns each import expects

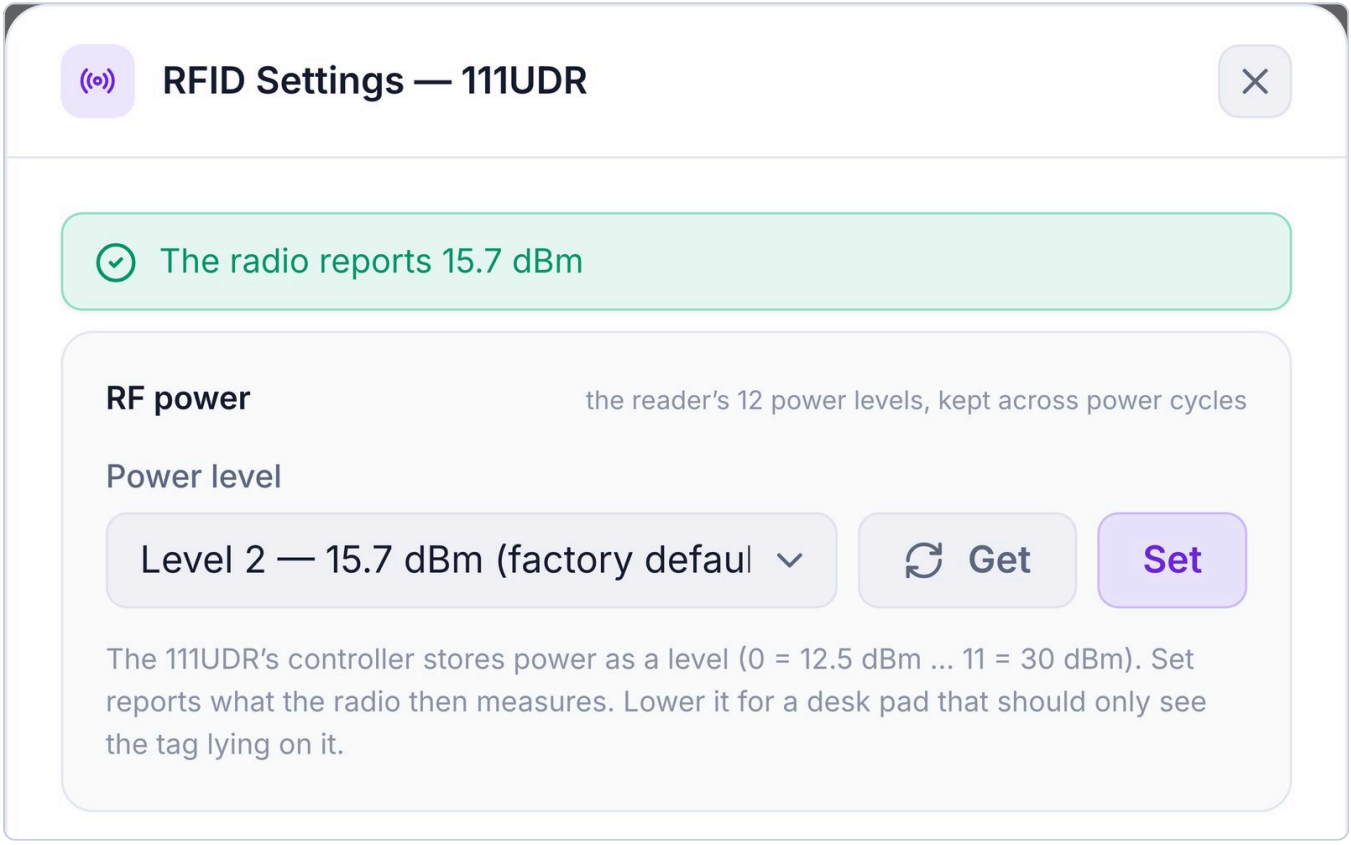
TIP
Run the first few tags of a batch, check them with **Read EPC**, and only then work through the box.

11/ Reader Settings: Power, Region and Buzzer

Open **Settings › RFID Settings**. The dialog covers the radio and the buzzer; everything to do with typing is under **Settings › Keyboard Mode**.

RF power

The reader stores its radio power as one of 12 levels, not as a number of dBm, and keeps it when it is unplugged. The app lists each level with the dBm figure the reader gives it – level 0 is the lowest and level 11 the highest, and level 2 is the factory setting. Choose a level and click **Set**; the message that follows reports what the radio then measures. **Get** reads the current value back.



RFID Settings – the reader's 12 power levels

TIP
Lower the level if the reader picks up tags lying beside it instead of the one on the pad – for example while encoding single tags. Raise it if a tag on the pad is not read at all.

Region

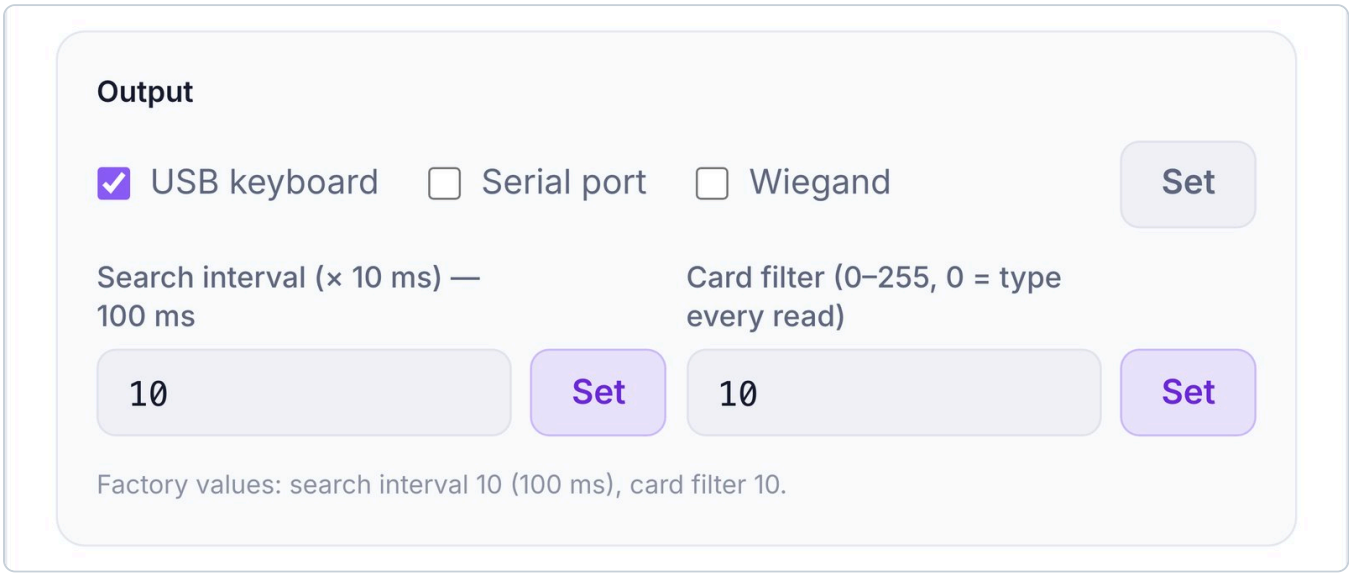
Under **Region & channels**, choose **Europe / India 865-868 MHz** and click **Set**. That is the band the reader must use in India. **Frequency hopping** moves the reader between the channels of the band; leave it on.

Buzzer

Reader beeps switches the reader's buzzer on or off; click **Set** to send it.

Output, read interval and repeat filter

The rest of the reader's own settings are in **Settings › Keyboard Mode**, under **Output**, because they decide how it behaves when it is typing on its own.



Keyboard Mode › Output: where the reader sends what it reads, how often it looks for a tag, and the repeat filter

Setting	Range	Use
USB keyboard / **Serial port** / **Wiegand**	On or off	Where the reader sends each number it reads. Leave **USB keyboard** on for a PC.
Search interval	0–255, in units of 10 ms (factory 10 = 100 ms)	How long the reader waits between read attempts
Card filter	0–255 (factory 10)	0 types the tag on every single read; a higher value stops the same tag being typed again and again

Serial port & Wiegand, lower down, holds the baud rate, the serial end symbol and the Wiegand format for readers wired into a serial or access-control system. Enter a value and click the **Set** button beside it.

> **NOTE**
> The reader reports its output format back to the app, so **Read from reader** always shows the truth. The other settings cannot be read back, so the dialog shows what was last sent in this session. **Tools › Restore Factory Settings** puts power, reader mode, outputs and baud rate back to their defaults.

12/ For Developers: SDK Quick Start

The SDK is a Windows library, `DLLProject_x86.dll` / `DLLProject_x64.dll`, with sample applications and their source code in four languages; the C#, Delphi 7 and Java samples are also ready to run. All functions return `0` for success. It is published on the Resource Hub as **UDR-111 – UHF Desktop Reader SDK**, `Identium-UDR-111-SDK-`

`1.0.zip` (about 8 MB).

What is in the package

Unzip the download and open `Identium-UDR-111-SDK-1.0`. Everything is already unpacked: the API reference in `Docs/`, the two DLLs in `Library/` and the sample applications in `Samples/`. `README.md` lists every folder.

In `Identium-UDR-111-SDK-1.0`	Contents
`Docs/Identium-UDR-111-Interface-Functions.pdf` / `.docx`	API reference: the functions, their parameters and return values (both files have the same, complete content)
`Library/`	`DLLProject_x86.dll` and `DLLProject_x64.dll`
`Samples/`	The sample applications (next table)

In `Samples/`	Contents
`CSharp/RFID_EPC_Project/`	C# sample (Visual Studio, .NET Framework 4.0). `bin/Release` holds ready-built `RFID_EPC_Project_x86.exe`, `RFID_EPC_Project_x64.exe` and both DLLs
`Delphi7/`	Delphi 7 sample with source and a built .exe
`VC2010/`	Visual C++ 2010 sample with source and the `DLLProject.h` / `DLLProject.cpp` wrapper; build it in Visual Studio (`Release/` already holds the DLL)
`Java-x86/` / `Java-x64/`	Java (NetBeans) samples using JNA. Install a 32-bit (x86) or 64-bit (x64) Java 8 or later, then run `dist/start.bat`

Run the sample application

1. Connect the reader. Close Identium Desktop Reader and any other program that uses the reader.
2. Open `Samples/CSharp/RFID_EPC_Project/bin/Release` and run `RFID_EPC_Project_x64.exe` (64-bit Windows) or `RFID_EPC_Project_x86.exe`.
3. On the **Connection & Read EPC** tab choose **USB** and click **Connect**. The button turns green and reads **Disconnect**.
4. Place a tag on the reader and click **Read Single**. The tag appears in the **EPC Table** with its PC, EPC and RSSI.
5. The **Read & Write Tag Memory** tab selects a tag (**Select Parameter**) and reads, writes, locks or kills it. **Test & Modem Setting** holds RSSI scan and modem settings.

> **NOTE**
> With a USB connection the sample enables **Read Single** only. To read continuously, call `ReadSingle()` in a loop from your own code.

> **TIP**
> For everyday work – setting the band, the power level and the buzzer, reading, writing and encoding – use Identium Desktop Reader (sections 05–11). The sample application is there to show the API, not to run a desk.

13/ Integrate the SDK in Your Application

1. Copy `DLLProject_x86.dll` or `DLLProject_x64.dll` from `Library/` next to your program and build for the same platform (x86 or x64).
2. Declare the functions. In C#, reuse `DLLProject.cs` from `Samples/CSharp/RFID_EPC_Project/` and add the conditional compilation symbol `DllImport_x86` or `DllImport_x64` to your project, matching your build platform: the declarations in `DLLProject.cs` are compiled only when one of them is defined (the sample's x86 project defines `DllImport_x86`). In C/C++, add `DLLProject.h` and `DLLProject.cpp` from `Samples/VC2010/RFID_EPC_Project/` to your project, create a `DLLProject` object and call its methods. That wrapper loads `DLLProject_x86.dll`, so build for x86, or change the DLL name in `DLLProject.cpp` for x64.
3. Register a callback and connect: `Connect(2, NULL, callback)` – 2 = USB. Tag reads arrive in the callback as notice frames (Type `0x02`, Command `0x22`).
4. Read with `ReadSingle()`. In the notice data, byte 0 is RSSI, bytes 1–2 are the PC word, then the EPC (length from the PC word), then the CRC.
5. Read and write memory with `SetSelectParam()` (choose the tag by its EPC), then `ReadData()` / `WriteData()`. Banks: 0 RFU, 1 EPC, 2 TID, 3 User. Start addresses and read lengths are in 16-bit words; the sample passes the write data length in bytes.
6. Call `Disconnect()` when you finish.

Function groups in the API reference

The API guide, `Docs/Identium-UDR-111-Interface-Functions.pdf` (also as `.docx`), describes the functions below with their parameters and return values.

Connection and inventory

`Connect`, `Disconnect`, `GetModuleInfo`, `ReadSingle`, `SetSelectParam`, `SetSelectMode`

Tag memory

`ReadData`, `WriteData`, `LockUnlock`, `Kill`, `GetQuery`, `SetQuery`

RF settings

`SetRegion`, `SetRfChannel`, `GetRfChannel`, `SetFhss`, `SetPower`, `GetPower`, `SetCW`, `SetModemPara`, `GetModemPara`, `ScanJammer`, `ScanRSSI`

Chip-specific commands

Tag-chip-specific commands for read protection, EAS (anti-theft bit) and privacy (QT) settings. The function names and parameters are in the API reference.

```
static DLLProject.CallReceive onData = OnData;   // keep this reference alive

static int OnData(byte type, byte cmd, int size, IntPtr p)
{
    // Runs on a worker thread: marshal to the UI thread (the sample uses Invoke).
    if (type == 0x02 && cmd == 0x22 && size > 3)   // tag notice
    {
        byte[] d = new byte[size];
        Marshal.Copy(p, d, 0, size);
        int epcLen = (d[1] / 8) * 2;                // d[0] = RSSI, d[1..2] = PC
        string epc = BitConverter.ToString(d, 3, epcLen).Replace("-", " ");
        Console.WriteLine("EPC " + epc);
    }
    return 1;
}

// Connect over USB (ConnType 2, no connection string). 0 = success.
int rc = DLLProject.Connect(2, null, onData);

// Set the India band (Europe region) and the radio's transmit power.
DLLProject.SetRegion(3);        // Europe region = India band 865–868 MHz
DLLProject.SetPower(2000);      // hundredths of a dBm: 2000 = 20 dBm

// Read one tag. The EPC arrives in OnData.
DLLProject.ReadSingle();

// Close the connection when you finish.
DLLProject.Disconnect();
```

C# outline based on the sample project (RFID_EPC_Project)

CAUTION

In India use the reader on 865–868 MHz only. In the SDK this is the **Europe** region (`SetRegion(3)`): 15 channels, 0.2 MHz apart, all inside the 865–868 MHz band.

CAUTION

`Kill`, and the **Perma Lock** / **Perma Open** options of `LockUnlock`, cannot be undone. Test them on spare tags first.

14/ Precautions

- The housing is rated IP40: use the reader indoors and keep it away from water and other liquids.
- Use it between -20 °C and +60 °C, at up to 80% relative humidity, non-condensing.
- Connect it directly to a USB port on the PC; the reader takes its power from that port.
- When encoding, keep only the tag being written on the reader. Lower the power level under **Settings › RFID Settings** (section 11) to shorten the read range, so that nearby tags are not read or written.
- Keep the reader on the India UHF band, 865–868 MHz: in **Settings › RFID Settings** choose **Europe / India 865–868 MHz** and click **Set** (section 11).
- Do not open or modify the reader. Clean it with a soft, dry cloth.
- Before you lock or destroy tags, try the operation on spare tags; both can be permanent.

15/ Parameters

Model	IDTS-111UDR (UDR-111 UHF Desktop Reader)
Frequency	865–868 MHz (India)
Protocol	EPC Global UHF ISO 18000-6C (EPC Gen2)
RF output power	Adjustable – the reader stores it as one of 12 power levels, set in Identium Desktop Reader
Read range	0–50 cm, depending on the tag
Antenna	One, built in (the read area)
Interface	USB
Operating modes	Answer: read and write · Active: HID keyboard emulation
Reading rate	Software programmable
Read indication	Buzzer and LED
Software	Identium Desktop Reader (Windows and macOS), and the Identium UDR-111 SDK for Windows – both free from the Resource Hub
Power	USB
Material	Zinc alloy and acrylic
Dimensions	110 × 70 × 15 mm
Weight	150 g
IP rating	IP40
Humidity	80%, non-condensing
Operating temperature	-20 °C to +60 °C

Specifications are indicative; confirm the details for your order in the quotation.

16/ Troubleshooting

Problem	Likely cause	What to do
Nothing is typed when a tag is presented	Keyboard Mode is off, or the cursor is not in a text field	Open **Settings › Keyboard Mode** and click **Turn on**. Click into the target field and present the tag again.
The typed number is too short, reversed or in decimal	Output format	In **Settings › Keyboard Mode**, choose **Full number – the whole EPC / TID** and click **Set**.
Numbers run together on one line	No Enter after each number	Tick **Add Enter** and click **Set**.
The same tag is typed again and again	Card filter is 0	In **Keyboard Mode › Output** set a **Card filter** and click **Set**.
Only part of the TID is typed	The reader's own TID typing sends the last 16 characters	Choose **TID (full)** so the app reads and types the whole number, and leave the app open (minimised is fine).

Problem	Likely cause	What to do
The reader is not in the Connection window's device list	Cable, or the list is filtered	Press the refresh button beside the list, tick **Show all HID devices**, or re-plug the USB cable.
Connected, but no tag appears	No read started, or the power level is too low	Click **Read EPC** on the Read tab. If the tag still does not answer, raise the power level under **Settings › RFID Settings** (section 11).
Nearby tags are read, or the wrong tag is written	Read range larger than needed	Keep other tags away from the reader and lower the power level under **Settings › RFID Settings**.
A write is refused	Several tags on the reader, a wrong access password, or a locked tag	Leave one tag only, enter its access password, and try again. In Batch Write the reason is shown beside the tag in the change log.
The reader beeps on every read and you want it quiet	The reader's own buzzer	Untick **Reader beeps** under **Settings › RFID Settings › Buzzer** and click **Set**, or untick **Beep** on the Read tab.
Identium Desktop Reader or the SDK cannot reach the reader	The reader is in use by another program, or a 32-/64-bit mismatch in an SDK build	Close the other program – only one at a time can use the reader. For the SDK, keep the x86 DLL with x86 builds and the x64 DLL with x64 builds. Re-plug the USB cable.

17/ Support

For help with installation, integration or a fault that the steps above do not clear, contact Identium. Please keep the product model, the software version shown in the app and a short description (or screenshot) of the problem ready.

Company	Identium Tech Solutions Pvt. Ltd.
Address	Plot No. 5, First Floor, Raja Dhirsain Marg, Garhi, Sant Nagar, East of Kailash, New Delhi 110065, India
Phone	+91 97181 99516 · 011-4714 7839
WhatsApp	+91 82877 32647
Email	info@identium.in
Software and documents	www.identiumone.com/resourcehub/
Buy hardware and tags	www.indiarfidstore.com

This product and its software are updated from time to time; screens and specifications may change without notice. Check the Resource Hub for the latest version of this manual.