

Electronics Component Inventory for Hobbyists

From a Spreadsheet to a Full Inventory System

How to choose an inventory solution without
creating unnecessary complexity or locking yourself in

UFElectronics.eu

Version 1.0
8th of October 2026 - 1617

<https://ufelectronics.eu>

Contents

1	Introduction	2
2	Start with the simplest thing that works	2
3	But spreadsheets have a limit	3
4	The solution ladder	4
5	Local applications	4
6	Some interesting options	5
6.1	Spreadsheet / LibreOffice Calc	5
6.2	Binner	5
6.3	Component Catalog	6
6.4	Part-DB	6
6.5	PartHub	7
6.6	Cloud services	7
7	Software and resources	8
7.1	General-purpose alternatives	8
8	Comparing the approaches	9
9	The most important comparison: migration	10
10	The migration ladder	10
11	CSV is your escape hatch for the core inventory data	11
11.1	The data model matters	11
12	Don't make the spreadsheet too clever	12
13	How much inventory do you actually have?	13
14	What type of hobbyist are you?	13
14.1	The "I just need to find my resistors" hobbyist	13
14.2	The "I have drawers everywhere" hobbyist	13
14.3	The "I build lots of projects" hobbyist	14
14.4	The "I have a serious electronics lab" hobbyist	14
14.5	The "I'm building products" hobbyist	14
14.6	The "I don't want to maintain computers" hobbyist	14
15	Component inventory versus product catalogue	15
16	A practical strategy	16
17	My personal rule	17
18	Backups: Your inventory is data	17
18.1	What is a backup?	18
18.2	Export is not the same as backup	18
18.3	A backup should be automatic if possible	19
18.4	Do not forget the restore test	19
18.5	What about cloud and external services?	19
18.6	The simplest backup strategy	20
19	Conclusion	20

1 Introduction

How much software do you actually need to keep track of your electronic components?

If you are an electronics hobbyist, the answer may be: surprisingly little.

You might have a few drawers of resistors, capacitors, transistors and ICs. A spreadsheet may be perfectly adequate.

But what happens when the collection grows?

And what if you want:

- Bills of Materials (BOMs)
- automatic stock deduction
- storage locations
- supplier information
- datasheets
- barcodes
- labels
- KiCad integration
- project management
- multiple users

There is a whole range of solutions between:

“I have an Excel file.”

and:

“I need to maintain a server.”

The interesting part is that you do not necessarily have to make that jump all at once.

2 Start with the simplest thing that works

For a hobbyist, a spreadsheet is not a bad inventory system.

A simple table might contain:

- Part number / MPN
- Manufacturer
- Description
- Value
- Package
- Quantity
- Storage location

- Supplier
- Supplier part number
- Datasheet URL
- Notes

For a small collection, this can be all you need.

Spreadsheet software also has some enormous advantages:

- No installation
- No server
- No database administration
- Very easy to customize
- Easy to understand
- Easy to back up
- Easy to export
- CSV is supported almost everywhere

And perhaps most importantly:

You already know how to use it.

For a hobbyist, that matters.

3 But spreadsheets have a limit

The problem starts when the spreadsheet becomes an application that you are manually programming.

For example, you may eventually want to know:

- Where exactly is this component?
- How many do I have?
- Which BOMs use it?
- What happens to stock when I build a board?
- Which supplier sells it?
- What was the price?
- Do I have an alternative part?
- Which components are below minimum stock?
- Can I scan a barcode?
- Can I print storage labels?
- Can I connect the inventory to KiCad?

You can do some of these things in a spreadsheet.

You can even do surprisingly many of them.

But at some point you are essentially building your own inventory application inside Excel or LibreOffice Calc.

That is the point where specialized software starts to make sense.

4 The solution ladder

It is useful to think about the available solutions as a ladder.

Level	Typical solution	Best for
1	Spreadsheet / CSV	Small collection and simple inventory
2	Local inventory application	Growing hobby collection
3	Electronics inventory system	Serious component and BOM management
4	Self-hosted server application	Multiple users and advanced workflows
5	Cloud service	Users who do not want to maintain anything

You do not have to start at level 4 just because level 4 exists.

5 Local applications

There is an interesting middle ground that is easy to overlook.

You do not necessarily need a server.

Applications such as Binner are designed specifically for electronics inventory and can run locally.

Another approach is a local-first application such as Component Catalog.

Some local applications technically use a local web interface, but that is very different from having to administer an Internet-facing server.

There is no need for:

- a VPS
- nginx configuration
- a public IP address
- an SSL certificate
- public database administration
- server security maintenance

You simply run the application on your own computer.

For a hobbyist, that distinction is important.

6 Some interesting options

The following projects represent different points on the spectrum. They should be regarded as examples rather than a definitive list.

Software features, licensing and free plans can change, so current documentation should always be checked before making a decision.

6.1 Spreadsheet / LibreOffice Calc

The simplest solution.

It is difficult to beat for a small collection.

The important trick is to keep the data structured.

Do not put:

10k resistors in drawer A4

into one cell.

Instead use separate fields:

MPN	Value	Package	Quantity	Location
RC0603FR-0710KL	10k	0603	137	A4

This looks boring.

That is exactly why it is good.

A boring, structured dataset is easy to migrate later.

6.2 Binner

Binner is an electronics inventory system intended for component management.

It provides functionality such as:

- component inventory
- storage locations
- projects
- BOMs
- supplier information
- datasheets
- barcode support
- labels
- KiCad integration
- import/export

It is particularly interesting if you have outgrown your spreadsheet but do not want to become a server administrator.

6.3 Component Catalog

Component Catalog takes a somewhat different approach.

It is very focused on the physical component collection:

- boxes
- drawers
- compartments
- quantities
- locations
- datasheets
- photographs
- 3D models
- BOMs
- labels
- KiCad-related data

This is appealing if the main problem is:

“I know I own this component. I just don’t know where I put it.”

For electronics, the physical storage location is often almost as important as the part number. Knowing that you own 200 capacitors is much less useful if you cannot find them.

6.4 Part-DB

Part-DB is a more mature, full-featured electronics inventory system.

It goes considerably further into:

- component data
- suppliers
- stock
- storage locations
- BOMs
- datasheets
- parametric information
- APIs
- KiCad integration
- external component data

It is a very interesting choice once you are talking about serious component inventory.

The trade-off is that it is a server-oriented application, so it introduces more complexity than a local spreadsheet or desktop-style application.

6.5 PartHub

PartHub is an open-source electronics inventory and BOM system based on PHP/Laravel. It can be self-hosted, and a hosted service is also available.

It is particularly interesting for people who already know the Laravel/PHP ecosystem.

It includes functionality such as:

- inventory
- storage locations
- BOMs
- stock history
- CSV import
- barcode scanning
- multiple users

For a developer, the fact that it is built on familiar technology can make it attractive.

6.6 Cloud services

There is also the cloud approach.

PartsBox is one example of a browser-based electronics inventory and BOM service.

The attraction is obvious:

- No server
- No database administration
- No Docker
- No SSL configuration
- No server backups to configure
- No software updates to administer

You simply use the service.

The trade-off is that your data lives in somebody else's system, and free plans may have limitations.

Cloud and lock-in are related, but they are not the same thing. A cloud service with good export facilities may be much easier to leave than a locally installed system with a proprietary data format.

7 Software and resources

The following projects and services are discussed in this guide. Always check the current project documentation for the latest features, requirements, licensing and pricing.

- **Binner** - Electronics parts inventory system for makers, hobbyists and professionals.

<https://github.com/replaysMike/Binner>

Website: <https://binner.io/>

- **Component Catalog** - Local-first inventory system focused on electronic components, physical storage locations, cases, BOM matching and labels.

<https://github.com/mikeymcfish/component-catalog-app>

- **Part-DB** - Open-source inventory management system for electronic components, with BOMs, storage locations, suppliers, import/export and KiCad integration.

<https://github.com/Part-DB/Part-DB-server>

Documentation: <https://docs.part-db.de/>

- **PartHub** - Open-source electronics inventory and BOM management system based on PHP/Laravel.

<https://github.com/osterchrisi/PartHub>

- **PartsBox** - Browser-based electronics inventory and BOM management service, including a free Maker plan.

<https://partsbox.com/>

7.1 General-purpose alternatives

A spreadsheet application such as LibreOffice Calc or Microsoft Excel remains a perfectly valid starting point for a small inventory.

LibreOffice Calc: <https://www.libreoffice.org/discover/calc/>

Microsoft Excel: <https://www.microsoft.com/microsoft-365/excel>

8 Comparing the approaches

The following table is a deliberately broad comparison.

The table is intended as a first-pass comparison rather than a feature certification.

"Built-in" means that the software provides functionality in this area; the depth and implementation can differ considerably between products.

The exact capabilities of individual projects can change over time.

	Spreadsheet	Binner	Component Catalog	Part-DB /	
PartHub	Cloud				
Cost	Free / low	Free	Free	Free	Free tier / paid
Runs locally	Yes	Yes	Yes	Possible	No
Internet required	No	No	No	No, if self-hosted	Usually yes
Server administration	None	None	None	Yes, if self-hosted	None
Simple part list	Excellent	Excellent	Excellent	Good	Excellent
Storage locations	Manual	Built-in	Excellent	Excellent	Built-in
BOMs	Manual	Built-in	Built-in	Excellent	Excellent
Stock deduction	Manual	Built-in	Built-in	Built-in	Built-in
Supplier data	Manual	Built-in	Built-in	Excellent	Excellent
Datasheets	Links	Built-in	Built-in	Built-in	Built-in
Barcode / labels	DIY	Built-in	Built-in	Built-in	Built-in
KiCad integration	DIY	Yes	Limited	Yes / Future	Depends
Customization	Excellent	Moderate	Moderate	High	Limited
Learning curve	Very low	Low	Low	Higher	Low
Data export	Excellent	Good	Good	Good	Depends

* Component Catalog runs a local web server, but this is very different from maintaining an Internet-facing server. The user does not need to administer a remote server.

* Component Catalog provides KiCad-related functionality, but this is not the same as a full inventory/KiCad integration. PartHub currently lists KiCad support as future development.

9 The most important comparison: migration

There is another criterion that may be more important than the feature list:

How difficult is it to leave?

A system can have every feature you want and still be a bad choice if your data is effectively trapped inside it.

This gives us another useful comparison.

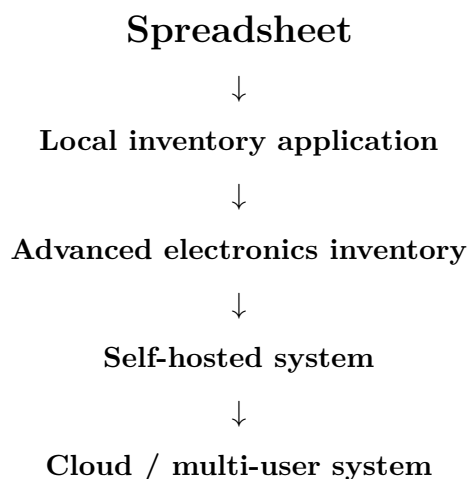
Solution	Data portability	Migration difficulty	Lock-in risk
CSV / spreadsheet	Excellent	Very easy	Very low
SQLite / local database	Excellent	Easy	Very low
Binner	Good	Easy / moderate	Low
Component Catalog	Good	Moderate	Low
Part-DB	Good	Moderate	Low
PartHub	Good	Moderate	Low / medium
Cloud service	Depends on export	Moderate / hard	Medium
Closed proprietary system	Potentially poor	Potentially hard	High

These ratings are general guidance rather than guarantees. Before committing significant data to a system, check its current export facilities and the formats in which your data can be retrieved.

10 The migration ladder

This leads to what I think is the most useful way of looking at the whole subject.

You can start with:



The important thing is to make sure that moving upwards does not require throwing everything away.

For example:

```
LibreOffice Calc
  |
  | CSV
  v
Binner / local application
  |
  | export
  v
Part-DB / advanced inventory
  |
  | API / database / export
  v
More advanced system
```

The exact migration path will depend on the software, but the principle is sound.

11 CSV is your escape hatch for the core inventory data

If I were starting from scratch, I would deliberately make the basic inventory data portable.

Keep the important fields in simple columns:

- Part ID
- Manufacturer
- Manufacturer part number
- Description
- Value
- Package
- Quantity
- Location
- Supplier
- Supplier part number
- Datasheet
- Notes

11.1 The data model matters

At first, an inventory may appear to need only:

Part + quantity + location

As the collection becomes more sophisticated, other information can become important:

- minimum stock level

- supplier
- alternative parts
- purchase price
- lot or date code
- condition or status
- available and allocated stock
- package and footprint
- notes and documentation

Total quantity is not always the same as available quantity.

As projects become more important, an inventory system may need to distinguish between stock that is available, allocated to a project, reserved, or otherwise unavailable.

For some components, especially moisture-sensitive or traceability-critical parts, lot and storage information can also matter.

The important point is not that every hobbyist needs all of these fields.

It is that the inventory system should make it possible to add them when the workflow requires them.

Then periodically export the data to CSV.

CSV is an excellent escape hatch for the core inventory data, but it is not necessarily a complete representation of an inventory system.

BOMs, stock history, multiple locations, attachments, photographs and other relationships may require additional exports or backups.

Even if you later adopt a sophisticated database, you still have a relatively universal representation of your inventory.

12 Don't make the spreadsheet too clever

This is perhaps the biggest practical recommendation.

If you know that your spreadsheet may eventually become something else, do not build a huge monster of formulas, macros and special formatting.

Keep the underlying data boring.

Use separate columns.

Use consistent values.

Do not mix several pieces of information into one cell.

Do not make the location part of the description.

Do not make the quantity part of the part name.

Do not rely on a particular spreadsheet program to understand your data.

Think:

Database first, spreadsheet second.

Even though you are using a spreadsheet.

13 How much inventory do you actually have?

The best solution also depends enormously on the size of the collection.

Collection / workflow	Sensible starting point
Less than 100 parts	Spreadsheet
100–1,000 parts	Spreadsheet or local inventory
1,000–5,000 parts	Local inventory becomes attractive
5,000+ parts	Dedicated inventory system becomes increasingly useful
Multiple projects / BOMs	Dedicated system
Multiple users	Server or cloud system
Selling components commercially	Proper inventory / e-commerce integration

These are not hard boundaries.

A person with 200 components and 30 BOMs may benefit more from specialized software than someone with 5,000 random components.

Workflow complexity matters at least as much as the number of parts.

14 What type of hobbyist are you?

The most useful question may therefore not be:

“Which is the best inventory application?”

It may be:

“What kind of problem am I actually trying to solve?”

14.1 The “I just need to find my resistors” hobbyist

Use a spreadsheet.

Do not over-engineer it.

14.2 The “I have drawers everywhere” hobbyist

Try a local component inventory application.

Storage locations become much more important.

14.3 The “I build lots of projects” hobbyist

Look for:

- BOM support
- stock deduction
- project management
- supplier information

14.4 The “I have a serious electronics lab” hobbyist

A dedicated electronics inventory system starts making sense.

14.5 The “I’m building products” hobbyist

Now inventory, BOMs, purchasing, suppliers, manufacturing and possibly e-commerce start becoming connected problems.

14.6 The “I don’t want to maintain computers” hobbyist

Use a cloud service.

You are essentially paying somebody else to be your system administrator.

15 Component inventory versus product catalogue

For a small electronics business, there is another important distinction.

Your **component inventory** is not necessarily the same thing as your **public product catalogue**.

For example:

PRIVATE INVENTORY

Resistors
Capacitors
Transistors
Connectors
PCBs
Screws
Wire
etc.

|
v

BOMs / manufacturing

|
v

PUBLIC PRODUCTS

BJT Experimentation Board
555 Timer Board
Multivibrator Board
Differential Amplifier Board
etc.

The private inventory may be perfectly happy in a dedicated inventory system or even a spreadsheet.

The public products may belong in WooCommerce or another e-commerce system.

Trying to make one application do absolutely everything can create more complexity than it solves.

16 A practical strategy

If starting from scratch, I would use a simple strategy.

1. Start with a spreadsheet if it solves the problem.
2. Keep the data structured.
3. Use stable identifiers such as manufacturer part numbers where appropriate.
4. Keep quantities and locations in separate fields.
5. Avoid unnecessary macros and proprietary spreadsheet logic.
6. Export a CSV backup periodically.
7. Move to a local inventory application when the spreadsheet becomes inconvenient.
8. Move to a dedicated electronics inventory system when BOMs, stock management and suppliers become important.
9. Move to a server or cloud solution when collaboration or remote access actually justifies it.

This approach avoids paying the complexity cost before you actually need the functionality.

17 My personal rule

I would use this rule:

Start with the lowest level that solves today's problem - but choose a data format that doesn't prevent tomorrow's solution.

That means there is nothing wrong with starting with Excel or LibreOffice.

In fact, it may be the smartest choice.

If you later discover that you are spending more time maintaining the spreadsheet than managing components, move up.

And if you eventually need:

- BOMs
- automatic stock deduction
- supplier integration
- barcodes
- KiCad integration
- multiple users
- purchasing

move up again.

The important thing is not to choose the most powerful system today.

It is to make sure that:

Today's simple solution does not become tomorrow's prison.

18 Backups: Your inventory is data

There is one subject that is easy to overlook when choosing an inventory system:

What happens if the computer, disk or service disappears?

An inventory database may represent years of work.

It may contain:

- thousands of components
- carefully entered part numbers
- storage locations
- supplier information
- prices
- BOMs
- project information

- datasheets and documents

Replacing the software is one problem.

Losing all of that information is a much bigger one.

18.1 What is a backup?

A backup is a separate copy of your data that can be used to restore the original if the original is lost, damaged or corrupted.

For a local inventory system, simply making another copy on the same disk is not enough.

For example:

Computer → another directory on the same hard disk

is not a very useful backup if that hard disk fails.

A more useful approach is:

Computer → separate storage device

or:

Computer → another computer

or:

Computer → cloud storage

The important principle is that the backup should be stored on a different digital storage unit from the original.

That could be:

- an external hard disk or SSD
- another computer
- a NAS
- a USB storage device
- cloud storage
- another suitable storage service

For particularly important data, keeping more than one backup and having at least one copy in a different physical location is even better.

18.2 Export is not the same as backup

An export and a backup serve different purposes.

An export is primarily about portability. For example, a CSV file can make it possible to move the core inventory data to another application.

A backup is primarily about recovery. It should allow you to recover from accidental deletion, disk failure, corruption or other data loss.

Ideally, you want both.

18.3 A backup should be automatic if possible

The best backup is usually the one that happens without you having to remember to do it.

For example, a local inventory database could be backed up automatically every day to an external drive or another computer.

If the inventory is stored as CSV files, automatic file synchronization or scheduled copying can provide a simple backup mechanism.

The exact method is less important than having a system that actually runs regularly.

18.4 Do not forget the restore test

A backup is only useful if it can actually be restored.

It is therefore worth occasionally checking that:

- the backup exists
- the files can be opened
- the database can be restored
- the exported data is readable
- the backup is not corrupted

A backup that has never been tested is an assumption, not a guarantee.

18.5 What about cloud and external services?

Cloud services and externally hosted inventory systems change the situation, but they do not eliminate the backup question.

If you use an external service, find out:

- Whether they perform backups
- How frequently backups are made
- How long backups are retained
- Whether backups are stored separately from the main system
- Whether you can export your own data
- Whether you can restore your data yourself
- What happens if the service is discontinued

Do not assume that:

“It is in the cloud” = “It is backed up.”

A cloud provider may have excellent backup procedures, but that is something to verify rather than assume.

And even if the provider has reliable backups, maintaining your own export of important data can still be a very good idea.

18.6 The simplest backup strategy

For a hobbyist, the backup strategy does not have to be complicated.

For example:

```
Main computer
|
+-----> External drive
|
+-----> Cloud storage
```

If the inventory is important enough, an additional copy can be kept somewhere physically separate.

The same principle applies whether the inventory consists of a single CSV file or a sophisticated database.

The software can be replaced. Your data may not be.

This is another reason to prefer inventory systems that provide good export facilities.

If you can export your inventory to a standard format such as CSV, you have an additional layer of protection against both technical failure and vendor lock-in.

CSV is an excellent escape hatch for the core inventory data, but it is not necessarily a complete representation of an inventory system.

BOMs, stock history, multiple locations, attachments, photographs and other relationships may require additional exports or backups.

19 Conclusion

There is no universally “best” electronics inventory system.

For one hobbyist, a spreadsheet is exactly right.

For another, it becomes a daily frustration.

For someone with thousands of components and many projects, a dedicated electronics inventory system may save enormous amounts of time.

And for somebody who simply does not want to maintain software, a cloud service may be worth the trade-off.

The important thing is to recognize that these are not completely separate choices.

They can be stages in a progression.

Simple -> Structured -> Specialized -> Connected

Start simple.

Keep the underlying data portable.

Move upwards when the complexity of the work justifies it.

And keep an escape route.

UFElectronics.eu
<https://ufelectronics.eu>

Version 1.0 - 8th of October 2026 - 1617