BOUT Solutions
Improves your logistic operations

ScanSystems

Installation/User Manual

For system version **v30.00**, and higher

Revision v1.03 (d24-m7-2025)

BOUT
Solutions

Intro

 Please **do not open the Scannerbox/Buttonbox yourself**, and **do not configure the ScanSystem and attached sensors and scanners yourself,** without receiving instructions to do so.

 Please note that this manual is written with newer ScanSystems hardware in mind. Please refer to the **"Bout Solutions - ScanSystems software hardware compatibility list"**, to see what hardware is compatible with ScanSystems firmware **v30**.

This manual will go over everything you need to know to get started with the Bout Solutions - ScanSystems. We will discuss what hardware makes up our product, how to install it onto your forklift/production line, together with Zebra scanners of your choice.

This manual will refer to "your **ScanSystems contact person**" a few times. This contact person can be the retailer/installer of your ScanSystem(s) from the client's point of view. Product retailers/installers can omit these references in most cases.

The target audience of this manual are retailers/product installers. It can also contain useful information for customers, but note that the customer shouldn't have to go trough the setup steps in order to get the system working. This should be done by the installers, except if the customer wants to do it by themselves.

The manual is written for ScanSystems **v30**, and higher. If you have received a newer version of the ScanSystems, please make sure there is no newer manual available for this device. You can find out what software version you have by looking at the sticker on the back on either the Scannerbox, or Buttonbox. Please note that older versions didn't have this displayed on the back. In the troubleshooting section, you will find more ways to know what version you have.

BOUT Solutions

Contents

Glossary

Possibly foreign concepts are explained here in order of simple to advanced (what concepts you need to know first to know the next one), and then alphabetically if possible.

2D codes: 2D codes can be Code128 (1D), QR codes, and dot matrix codes for example. All 1D/2D codes are referred to as 2D codes, since they are always printed as a 2D image.

ASCII/Extended ASCII: American Standard Code for Information Interchange. A commonly used standard to represent mainly English symbols, letters, and numbers within a computer system.

Booting: The process a computer device goes through from the point when it receives power, to where it starts being operational. In ScanSystems case, operational means that the ScanSystems is waiting for a trigger input for example.

Configuration/settings/user settings: In the context of this document, a configuration means a collection of settings which the ScanSystems can be configured with. There are two types of settings the ScanSystems can work with, **system settings**, and **user settings**. When we are talking about a configuration/settings/user settings, it means we are always talking about the user settings of the ScanSystems. The system settings are reserved for special variables like the data necessary to connect Scannerbox, and Buttonbox together, or to enable/disable demo mode.

EPC (Electronic Product Code): An RFID tag has two identifiers. The EPC is its rewritable identifier. You can put any data here, as long as it's size stays the same as what the EPC length for that tag type should be.

<ETX>, and <ESC>: These are ASCII control characters. <ETX> = HEX0x03, and <ESC> = HEX0x1B. They are used for special functions within the ScanSystems. Please look at the manual of the scanner(s) you are planning to use, to see how you can output these characters with it/them.

IO: Inputs/outputs. Physical ports on a device, which it can use to communicate with humans/other devices, and humans/other devices with it. In the ScanSystems case, this can be the Scannerboxes scanner connectors, sensor connector, or status LED.

Single/double IO sensor: A type of sensor with one or two output lines, which in the case of the ScanSystems; can be attached to the Scannerbox in order to receive triggers to start scanning.

PCB: Printed Circuit Board. The guts of a lot of electronic devices including the ScanSystems. Most components needed to make the device functional are attached to this board in one way or another.

Peer-to-peer Wi-Fi: A type of Wi-Fi connection that can be limited to only the devices who directly communicate with each other (peers). This type of connection doesn't need to be integrated into a client's network, and is standalone.

Silkscreen (PCB): The text that is printed on top of the PCB, it denotes the names of components on the PCB, and some extra information.

Special commands: Commands that can be generated via the **ScanSystems Configuration tool**, in order to execute specialized functions in your ScanSystems device, like enabling demo mode, or factory resetting the system.

1. Introducing ScanSystems

The Bout Solutions ScanSystems is a plug-and-play solution designed to simplify and streamline logistics operations that require 2D or RFID code scanning. With its smart wireless scanning technology, ScanSystems is ideal for both moving equipment like forklifts and fixed setups such as production lines. It boosts efficiency, improves safety, and reduces costs across your entire operation.

Compact, Wireless, and Easy to Install

ScanSystems consists of two compact units: the Scannerbox and the Buttonbox, connected wirelessly to eliminate the need for complex and fragile cabling. The Scannerbox is commonly placed next to the scanners on a moving part such as the forklift carriage, while the Buttonbox is mounted inside the cabin and connected to the terminal with a single USB cable.

This wireless design avoids the common issue of cable wear in moving systems, speeding up installation and increasing system reliability.

True Plug and Play

No drivers. No software installation. No external Wi-Fi needed. ScanSystems operates on a closed private network, meaning you can simply connect the batteries to the Scannerbox, plug in the Buttonbox, and you're ready to go. It's fast, intuitive, and completely hassle-free.

Safer and Faster Scanning

Scanners can be triggered manually using the Buttonbox or automatically by sensors. Operators no longer need to step out of the cabin or lean dangerously outside to scan barcodes. This dramatically reduces safety risks and speeds up operations, especially in high-paced environments like warehouses and distribution centers.

1.1. Operating principle and intended function

ScanSystems is designed to enable wireless communication between a scanner and a vehicle terminal in industrial environments, particularly forklifts.

The system allows data from a barcode scanner to be transmitted wirelessly, eliminating the need for cabling through moving parts of the vehicle (such as the forklift mast), reducing mechanical wear and simplifying installation. The basic workflow is described below. For a more detailed explanation, please refer to chapter 2.4.

Basic Workflow:

1. User Action
   The forklift driver presses a button on the Buttonbox to initiate a scanning process/the external sensor attached to the scannerbox sensor triggers the scannerbox directly to initiate a scanning process.
2. Signal Transmission
   The Buttonbox sends a wireless trigger signal to the Scannerbox (applicable when triggering the Scannerbox via the Buttonbox).
3. Scanning
   The Scannerbox receives the signal and captures data from a connected barcode scanner or external sensor.
4. Wireless Response
   The Scannerbox wirelessly transmits the scanned data back to the Buttonbox.
5. Output to Terminal
   The Buttonbox prints the received data via a HID, or USB serial to the forklift terminal or computer.

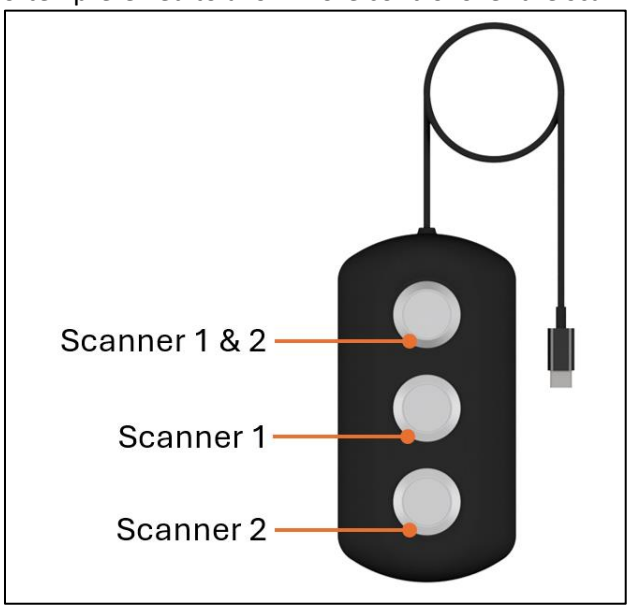
Power Supply

- Scannerbox: Powered via external power supply or rechargeable batteries (not included).
- Buttonbox: Powered through a USB connection from the forklift terminal.

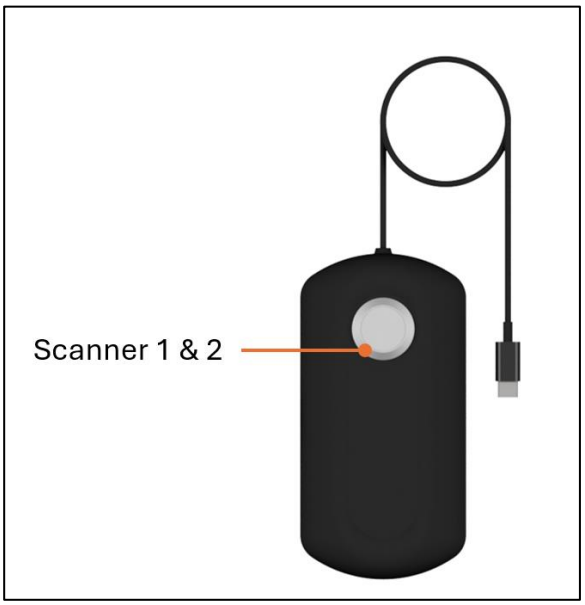
The system operates via **peer-to-peer Wi-Fi**, this does not require external Wi-Fi infrastructure.

1.2. Hardware and Compatibility

ScanSystems consists of two main units, the Scannerbox and the Buttonbox. The Buttonbox is available with 1, 2, or 3 buttons. The picture below "Buttonbox with one button" shows the basic version with one button, while the extended version with three buttons is shown in the other picture below "Buttonbox with three buttons" . For systems that require only one scanner, a Buttonbox with a single button is always used. When the system includes two scanners, a Buttonbox with 2 or 3 buttons is often preferred to allow more control over the scanners.



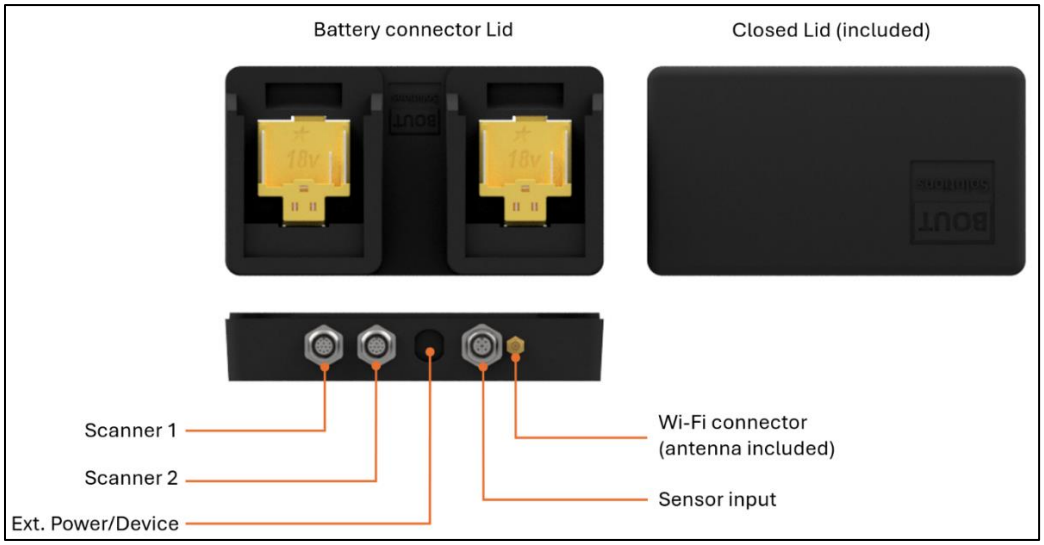
Buttonbox with one button



Buttonbox with three buttons

The Buttonbox connects via a USB cable, and each unit comes with a USB-C to USB-An adapter. The Buttonbox can communicate via HID or USB serial to a terminal, or another output device.

The Scannerbox connects wirelessly to the Buttonbox. It executes commands received from the Buttonbox. Up to two scanners can be connected to the Scannerbox, as shown in the image below. In addition, a sensor can be connected that sends a trigger signal to automate the process. Replacing the battery connector with a closed lid, and adding an external power connector is also possible. This modification can be applied, when an external power supply is available on the forklift board. The port can also be used for external devices, including the gimbal, which can be found on our website: www.boutsolutions.nl.



Scannerbox hardware layout

For the detailed system specifications, please refer to the datasheet included as Attachment 3. Hardware details, including input/output configurations, are explained in Attachment 2.

The ScanSystem is compatible with any serial scanner that requires only a trigger signal and transmits data at 9600 baud. It works especially well with Zebra FS series scanners as well as specialized devices such as the Bout Solutions RFID scanner.

Commonly the UC2000-L2-E6-V15 sensor from Pepperl+Fuchs is used. It is an ultrasonic sensor known for its robustness in handling warehouse packages and works seamlessly with the ScanSystem.

All compatible products can be found on our website at www.boutsolutions.nl. To request the full product list and detailed pricing information, please contact us directly.

2. How the ScanSystems functions in depth

This chapter will teach you the inns, and outs of the ScanSystems firmware. You will learn how the Scannerbox, and Buttonbox function, talk to each other, and what you can do with it.

 **When reading this chapter, the manual will presume that you know the component names of external and internal IO, described in Attachment 2.** The explanation will use these names.

2.1. System boot up sequence, and main program cycle

When you power on the Scannerbox, and Buttonbox, the first thing the devices will do is configure their IO. The Scannerbox will check what type of sensors, and scanners are attached, and the Buttonbox will check if it should run in HID, or USB serial mode amongst many other things. This can take up to 4 seconds, after which both devices will go to their main program cycle when not in pairing mode (chapter 2.2). The main program cycle is the cycle that is always ran after the boot sequence. Everything Scannerbox and Buttonbox do from now on happens within this cycle.

By default, the Scannerbox waits for a trigger signal from its attached sensor input, or Buttonbox. The Buttonbox waits until one of its buttons is pressed to send a trigger signal to the Scannerbox, it also waits for data from the Scannerbox in its main program cycle.

 Since Scannerbox, and Buttonbox initialize their IO while booting, **you will have to restart Scannerbox/Buttonbox when making any changes to IO** (install a new scanner, or add, or remove a sensor).

 ScanSystems works with precision timers that would overflow after approximately 45 days, which could cause some weird behavior. To completely eliminate possible strange system behavior, **the Scannerbox, and Buttonbox will reboot after 40 days** exactly while being in their main program cycle. You won't notice this (apart from the devices rebooting which takes approx. 8 seconds), since the systems will automatically connect to each other again.

 Please note that **you can't trigger the Buttonbox, and Scannerbox while a scan is being performed until the scan is finished, and data has been processed.** ScanSystems will ignore your triggers while busy.

 **When the Scannerbox and Buttonbox restart, they will forget all their temporary data** - such as duplicate scanner data (chapter 2.4.2.3), scanner data currently in the buffer, etc. It will of course remember its system, and user settings.

2.2 Pairing mode

 Please **do not open the Scannerbox/Buttonbox yourself** without having received instructions to do so**.**

 When pairing Scannerbox, and Buttonbox. **Make sure that you're only pairing one Scannerbox, and one Buttonbox at the same time**, since otherwise you could end up with Buttonboxes being paired with different Scannerboxes.

Pairing mode, is a special mode that Scannerbox, and Buttonbox can be launched into when booting up. When launching pairing mode, Scannerbox, and Buttonbox won't go to their main program cycle after initializing like they normally do. They will launch into a state that searches for another Scannerbox, or Buttonbox to connect to.

Normally, you don't have to pair the Scannerbox, and Buttonbox to each other. This has been done from factory. There are exceptions however. Maybe you find yourself in a situation where you've replaced the Buttonbox, or Scannerbox. Then you'd have to pair Scannerbox to Buttonbox again.

To begin pairing Buttonbox, and Scannerbox, you will need to open up the Buttonbox, and Scannerbox housings. Once you've done that, it should look something like the **Buttonbox internal IO description**, and **Scannerbox internal IO description** images, depicted in Attachment 2 under the **Buttonbox overview**, and **Scannerbox overview** section.

First you will need to make sure both devices are receiving power, and are close to each other. You can check if the devices are receiving power, by checking if the **Power LED** of the Buttonbox, and Scannerbox is lit up. Once you've done this, please now press and hold the **Connection button** on the Scannerbox. While holding this button, also press the **Reset button** on the Scannerbox, and release that button after pressing it. Now wait until a blue flashing light appears on the **Status LED**. When this light appears, you can release the connection button. The Scannerbox is now in pairing mode. Now you can do the same for the Buttonbox, however the built in status LED on the Buttonbox can only display a green color, so when the Buttonbox is in connection mode, its status LED will start flashing green. Once both devices are in pairing mode, they will start looking for each other, and once they find each other, their status LED will start flashing fast, after they will reboot (slow flashing yellow light). Now the devices should be successfully paired. This means you can close up the housings of Scannerbox, and Buttonbox again, and you're finished with the procedure.

2.3. Communication characteristics between Scannerbox, and Buttonbox

Scannerbox and Buttonbox, communicate quite a lot with each other. The most important details about the system's communication will be explained here.

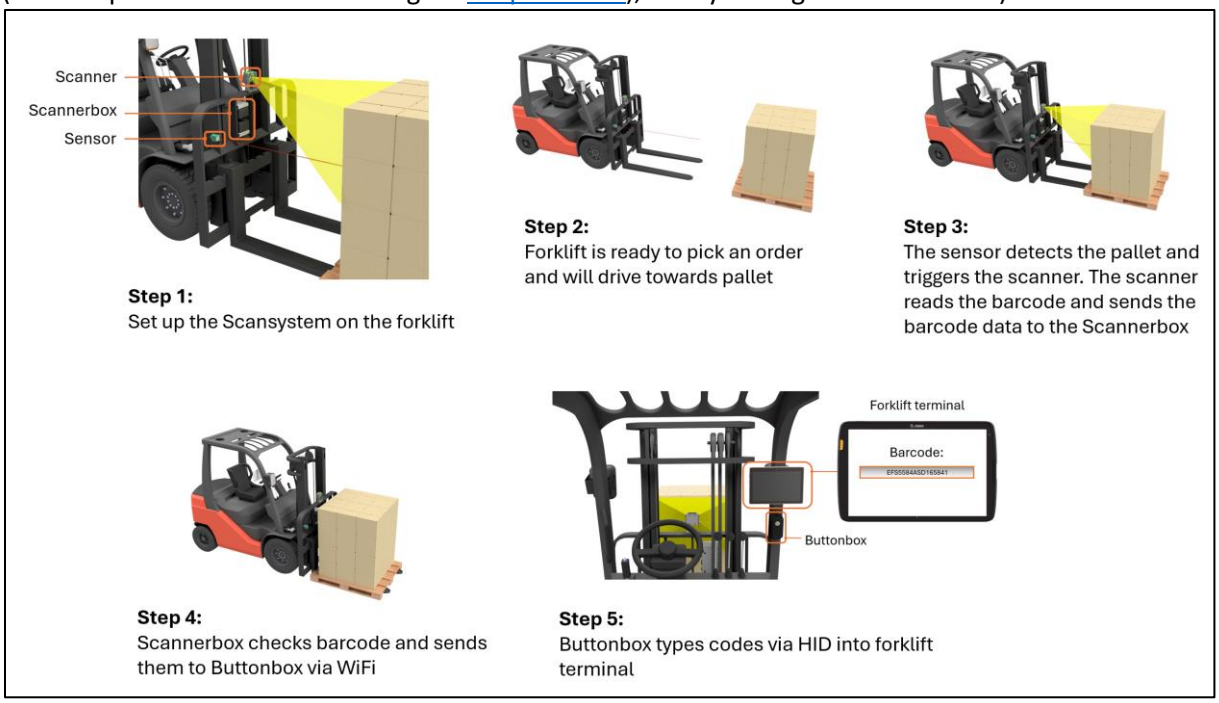
When Buttonbox, and Scannerbox send messages to each other, they will only send this data once without retry. This means that if Buttonbox, and Scannerbox have lost connection with each other, and want to send data to each other, they won't retry sending that data.

While scanning, or when finished scanning, Buttonbox, and Scannerbox communicate this with each other, so you can't trigger the system to perform a new scan when it's already performing a scan. There is a safety timeout, however. If for some reason Buttonbox, and Scannerbox lose communication with each other, they will after a while presume the scanning procedure on the other device has finished. This way they will never lock up. This timeout is dynamically calculated based on the user settings you've set for Scannerbox, and Buttonbox (user settings are discussed later in chapter 2.5).

When losing connection between Scannerbox and Buttonbox, the devices will automatically re-initialize connection with each other when they've found each other again. This also means that if you restart the Scannerbox or Buttonbox at any point for example, they will simply reconnect again.

2.4. The scanning cycle

When the Scannerbox is triggered by its attached sensor input, or the Buttonbox, and the system is not already performing a scan, a scan will be initiated. You can see an example of how a scanning cycle goes (note that depending on the hardware, and **user settings** configuration of your ScanSystems (more explained about user settings in chapter 2.5.1), the cycle might look different):



Scanning cycle example

2.4.1. Triggering the Scannerbox

Note that you can configure the Scannerbox user settings in such a way that all attached RFID scanners will always scan, even when the system isn't looking for data. When the Scannerbox is then triggered for the scanner port this RFID scanner is attached to, it will use the RFID scanner's latest tag **EPC** as its found data when no new tag was found after that.

Depending on the settings set in the Scannerbox, it waits for HI/LO flanks from the **single** and **double IO sensors**, so we will refer to a scanning flank as HI.

A scanning cycle always starts with a trigger. Either this trigger can come from the Buttonbox, or it can come from one of the Scannerboxes sensor inputs. Triggering the ScanSystems only works when an active scan has finished. Otherwise, the ScanSystems will ignore any trigger input. Triggering of the Scannerbox via sensor can be done with a single IO sensor, or a double IO sensor.

When a single IO sensor becomes HI (**Sensor connector** **input 1** pin), the Scannerbox will trigger its assigned scanner(s).

A double IO sensor is a bit more complicated. When **Sensor connector** **input 2A** pin gets HI, the Scannerbox will trigger its assigned scanner(s). Now if **Sensor connector** **input 2B** pin gets HI, and then LO, the Scannerbox will trigger its assigned scanner(s) (which can be the same scanners or different ones depending on the configuration of the Scannerbox).

A trigger from the Buttonbox will trigger whatever scanner is associated to the button of the Buttonbox you pressed.

Ext. IO connector **output 1 and 2** pins, correspond to the **Scanner 1, and 2 connector** **trigger pins**. When scanner 1 is being triggered for example, Ext. IO connector output 1 pin will also be HI. Note that the Ext. IO connector outputs will ignore the output polarity that can be set for the scanner 1, and 2 connector trigger pins (so a trigger will always make the Ext. IO connector output pins HI, and no trigger makes them LO).

2.4.2. Scanning, and processing the scanned data

 Note **that the combined limit of all characters of both scanners connected to the Scannerbox is 4000 characters.** If the system receives more characters than this limit, the scan will be invalid.

When the Scannerbox is triggered, it will trigger scanners associated to that trigger signal. It will then start looking for data from the scanners it triggered. For any attached 2D code scanner, it will look for data via serial communication with a baud rate of 9600. For any attached RFID scanner, it will read **EPC** data from individual tags via serial communication with a baud rate of 115200.

Every time the Scannerbox receives a new chunk of data from one of the scanners attached to its scanner connectors, it will append the new data to the data buffer for that scanner if it's data from a 2D code scanner. If the data is coming from an RFID scanner, it will overwrite the EPC tag data currently in the buffer for that scanner with new EPC tag data if the tag meets your configured user settings (more about user settings will be explained in chapter 2.5.1)).

Data received from a scanner, will either be perceived as printable data (data that will be sent to the Buttonbox to be printed into the clients terminal), or as a **special command** by the Scannerbox. When no printable data, nor a special command has been found yet, the Scannerbox will continue waiting for data until its max scanning time has been reached, after which it will discard the data and stop scanning.

2.4.2.1. Processing printable data

 **2D code scanner's printable data is deemed corrupted when no <ETX> character is found at the end of the scanner's data. RFID scanner's tags only need to contain any EPC.** Printable data received after the <ETX> character will be cut off, including the <EXT> character itself.

Printable data is any scanner data that doesn't contain a **special command**, is perceived as printable data. The Scannerbox however still expects an <ETX> character to be present at the end of each scanner's data for each 2D code scanner attached to the Scannerbox. It expects the EPC of an RFID tag for each RFID scanner attached to the Scannerbox. If this is not the case, the Scannerbox will deem the data of that scanner corrupted, and discard it. All scanner data within a scan, received after the <ETX> character, will be cut off for each 2D code scanner. This character denotes the end of a 2D code scanner's data for this scan. Once the printable data is processed, it is sent to the Buttonbox where it can be printed.

2.4.2.2. Processing special commands

When the Scannerbox is receiving data from either one of its scanners, it checks for a **special command** suffix, and prefix. When this suffix and prefix have been found, the system identifies the data as a special command. The system will then try to execute this special command. When the system is finished executing this command, or when it found a special command prefix, and suffix but no special command, it will finish the scanning cycle afterwards. When the Scannerbox finds special commands from two scanners at the exact same time, it will only execute the special command found in the data of scanner 1. The scanning cycle is now finished.

2.4.2.3. Duplicate scanner data

Now onto duplicate scanner data finding. While processing printable data, the system checks if scanner data from this round is identical to each other, and if scanner data from this round is identical to data from last round (when the Scannerbox is triggered with an external sensor connected). It all depends on your user settings what the system will do when finding duplicate data. Checking against data from last round was implemented to prevent duplicate data from accidental sensor triggers. This is why the system will only compare this data when triggered via an attached sensor. When the Scannerbox was triggered by the Buttonbox, it won't compare data from the current round against data from last round. It will also save empty data to compare against next round when triggered this way, so when the Scannerbox finds new data next round whilst being triggered with an external sensor, the code won't be a duplicate.

When a special command was processed, the last scanned scanner data buffer is untouched, so processing special commands won't clear the last scanned scanner data buffer.

2.4.3. Printing scanned data

 **Using ScanSystems with HID on Windows 11 Notepad with spelling check enabled can cause garbled output data.** Windows 11 Notepad doesn't properly handle HID input very efficiently, so it is best avoided, even on ScanSystems slowest HID typing mode.

 Note that **scanner 1's data is printed first, followed by scanner 2's data.** This means that you can decide the order at which data is printed in, by physically swapping the scanners connected to the scanner 1, and scanner 2 connector port.

 Please note that the **Buttonbox supports typing all extended ASCII text characters via USB serial, but only the following special characters (besides numbers, and letters) when typing with HID:** \ ' " \ ( ) [ ] { } , . ; : | < > ` ~ ! @ # $ % ^ & * - + / = _ ScanSystems can type an ever-growing amount of extended ASCII characters. If you need the system to type characters that are not on this list, please contact your ScanSystems contact person, and we can add them in for you.

 **If** your **scanner data contains** one or more **<ESC> characters**, **the Buttonbox delays typing for 50ms per <ESC> character when it is printing the data from a scanning cycle**. This character is used for typing delays, it will not be typed by the Buttonbox.

When all printable data has arrived at the Buttonbox, the Buttonbox will start printing this data. Depending on if the **HID/USB serial selector jumper** is connected or not (will be further explained in chapter 2.5.1), the Buttonbox will start printing all data via HID, or USB serial. It will also delay typing when it encounters one or more <ESC> characters in the printable data for 50ms. After printing all the data, the scanning cycle is finished.

2.5. All about user settings, and how to change them

The ScanSystems has two types of settings. It has **user settings**, which are useful to the end user. These settings can for example alter how long the Scannerbox will look for scanner data, or how fast the Buttonbox will type its data, or which scanners to trigger when scanning. There's a lot of possibilities here. The ScanSystems also has **system settings**, which the Scannerbox, and Buttonbox for example use to keep track of which Scannerbox, and Buttonbox to connect to, or if they're in demo mode. All user settings can easily be changed with a tool we'll discuss in chapter 2.5.1. System settings will be changed by the Buttonbox, and Scannerbox themselves mostly. An important distinction between user, and system settings, is that system settings can't be reset, and user settings can.

2.5.1. Configuring the system with user settings

If you want to change how your ScanSystems functions by for example making the time it will scan longer, or which scanners it will trigger when scanning etc. You can do this by generating a 2D code, containing all kinds of user settings. You can then scan this code with scanners attached to your Scannerbox, after which the Scannerbox will apply the settings from this 2D code to itself, and the Buttonbox. You can easily configure the system with the **ScanSystems Configuration Tool**, you can download this tool from www.boutsolutions.nl, or ask your ScanSystems contact person about it. Make sure to download/ask for the correct version of the application for your ScanSystems firmware. This manual won't explain how to generate a 2D code containing these settings yourself, since the ScanSystems Configuration Tool can do all of this for you, so it is never necessary to do this by hand.

The Buttonbox also has one hardware based user setting. This setting allows you to select if the Buttonbox should print its data via HID, or USB Serial. If you want the Buttonbox to type its data via HID, you can short the two pins pointing to the center of the Buttonbox PCB (a bridge connector shorting these two pins should already be present on the PCB) of the **HID/USB serial selector jumper**. If you want the Buttonbox to output USB serial data, then remove this short.

2.5.2. Resetting the user settings to their default values

When wanting to reset the user settings to their default values, you have two options at your disposal. You can either do this via the aforementioned (chapter 2.5.1) **ScanSystems Configuration Tool**, or by opening up the system, and resetting it physically. The preferred method to reset settings is via the ScanSystems Configuration Tool. System settings can't be reset to their defaults.

2.5.2.1. Resetting the user settings via the ScanSystems Configuration Tool

You can reset all user settings of Scannerbox, and Buttonbox at the same time, via a **special command** through the **ScanSystems Configuration Tool**. Just head to the **Special commands screen**, and select the "**Factory reset**" special command option. The Scannerbox should now perform a factory reset of the user settings for itself, and it will also factory reset the user settings on the Buttonbox.

2.5.2.2. Resetting the user settings via hardware

Please **do not open the Scannerbox/Buttonbox yourself** without having received instructions to do so**.**

Reset both Scannerbox, and Buttonbox via hardware, if you're resetting either one of them. This is important, since Scannerbox, and Buttonbox have some settings which are synced between each other, and need to stay synced in order to keep the devices in a good working order.

After resetting the user settings on the ScanSystems via hardware, **don't forget to put the dipswitches back facing towards the numbers.** When you don't do this, the Scannerbox, and Buttonbox won't boot.

Resetting user settings on a Scannerbox, or Buttonbox via hardware, will only reset that specific device. This is unlike with the special command method which resets both devices at the same time.

To reset Scannerbox, or Buttonbox user settings via hardware, you will first need to open the casing of the system(s) you would like to reset. Once you've done that, it should look something like the **Buttonbox internal IO description**, and **Scannerbox internal IO description** images, depicted in Attachment 2 under the **Buttonbox overview**, and **Scannerbox overview** section. Make sure the system(s) you want to reset are receiving power. You can check if the system is receiving power by checking if the **power LED** is lit up. If the system is receiving power, please wait 5 seconds. You will now need to set all **dipswitches** away from the numbers. The status light on you system will now light up for a few seconds, after which it starts blinking and goes out. This means the system started initiating a factory reset was rebooting, and is now waiting for you to put the dipswitches back facing towards the numbers.

2.6. Status indicator

 Please **do not open the Scannerbox/Buttonbox yourself** without having received instructions to do so**.**

 **The built in status LED of the Buttonbox can only display a green color.**

The Scannerbox, and Buttonbox are equipped with an onboard **status LED**. At the time of writing this manual, the Buttonbox, and Scannerbox only have status LEDs mounted at their PCB's, which can't be seen outside of their casing. To check the status LEDs of Scannerbox, and Buttonbox, you will need to open the casing of the systems. You can find the location of the status LEDs in the **Buttonbox internal IO description**, and **Scannerbox internal IO description** images, depicted in Attachment 2 under the **Buttonbox overview**, and **Scannerbox overview** section.

The table depicted below this text, contains all possible status codes (light sequences) the Buttonbox and Scannerbox can display. Even though the Buttonbox status LED only displays a green color, care has been put into designing the status codes speeds, and frequencies, so it should be easy to deduct the status, by looking at these variables. The status codes are sorted by system alphabetically.

System	Light sequence	Description
Buttonbox	Blinks white once for 200ms	Trigger generated. This trigger will be sent to Scannerbox
	Blinks cyan twice for 500ms	Settings found, and applied
	Blinks red once for 4000ms	Factory reset initiated
	Blinks yellow 3 times for 600ms	System will reboot
	Blinks yellow infinitely after trying to reboot for 600ms	System can't reboot
	Blinks green once for 2000ms	Diagnostics mode continues
	Blinks green 5 times for 200ms	HID print mode was initiated instead of USB serial during the boot process
	Blinks red infinitely for 1000ms	Wi-Fi error during the boot process, please reboot the device
	Blinks blue indefinitely for 400ms	Search for Scannerbox in connection mode started
	Blinks blue indefinitely for 200ms	Scannerbox has been found in connection mode, currently communicating
	Blinks blue 5 times for 50ms	Buttonbox has been successfully paired to Scannerbox in connection mode
	Blinks green twice for 20ms	Buttonbox printed all data received from Scannerbox
	Blinks red twice for 60ms	Buttonbox didn't print data received from Scannerbox
Scannerbox	Blinks white once for 200ms	Scan started with one scanner
	Blinks white twice for 100ms	Scan started with two scanners
	Blinks cyan twice for 500ms	Settings found, and applied
	Blinks red once for 4000ms	Factory reset initiated
	Blinks yellow 3 times for 600ms	System will reboot
	Blinks yellow infinitely after trying to reboot for 600ms	System can't reboot

	Blinks pink once for 12ms	A chunk of scanner data has been found
	Blinks green twice for 20ms	Scanning cycle finished, and printable data was sent to the Buttonbox
	Blinks red twice for 20ms	Scanning cycle finished, but no printable data was sent to the Buttonbox
	Blinks red 4 times for 50ms	Demo mode was disabled
	Blinks red 8 times for 50ms	Demo mode was enabled
	Blinks green once for 2000ms	Diagnostics mode started
	Blinks red infinitely for 1000ms	Wi-Fi error during the boot process, please reboot the device
	Blinks blue indefinitely for 400ms	Search for Buttonbox in connection mode started
	Blinks blue indefinitely for 200ms	Buttonbox has been found in connection mode, currently communicating
	Blinks blue 5 times for 50ms	Buttonbox has been successfully paired to Scannerbox in connection mode

ScanSystems status code light table

2.7. Demo mode

ScanSystems has a demo mode. When demo mode is activated on a ScanSystems, the system will be locked for two minutes each time the Scannerbox boots/restarts. After these two minutes, the system is completely usable for 30 minutes. After 30 minutes, the system will start appending "DEMO" as a text behind all scanner data. After 45 minutes, the system will lock up, and you have to restart the Scannerbox again (by cutting power, and supplying it again for example).

When the ScanSystems is locked, and you try to perform a scan, the Buttonbox will print that the system is in demo mode. There is a key attached to this message, which you will need if you want to unlock the system. Note that each time the Scannerbox is restarted, a new key will be generated. If you want to unlock your ScanSystems, please contact your ScanSystems contact person about this.

When the ScanSystem is locked in a demo mode lock, some special commands like unlocking the system from the demo mode still work.

2.8. Updating the ScanSystems firmware

 Please **do not open the Scannerbox/Buttonbox yourself**, and **do not update the ScanSystem yourself,** without receiving instructions to do so.

If you want to see if you can update your ScanSystems to a newer firmware revision, please contact your ScanSystems contact person about the process to update your ScanSystems. Please also have a look at the following documents: **"Bout Solutions - ScanSystems Updater Tool manual"**, **"Bout Solutions - ScanSystems software hardware compatibility list"**. Both documents can be acquired by contacting your ScanSystems contact person.

 **A new version of the ScanSystems might have deleted/renamed some settings, or changed other properties of them.** In this case, the ScanSystems will automatically reset these settings to its new default values. **This means that after updating the ScanSystems, it might behave differently than what you're used to.** You can always choose to perform a factory reset, if you want to start over.

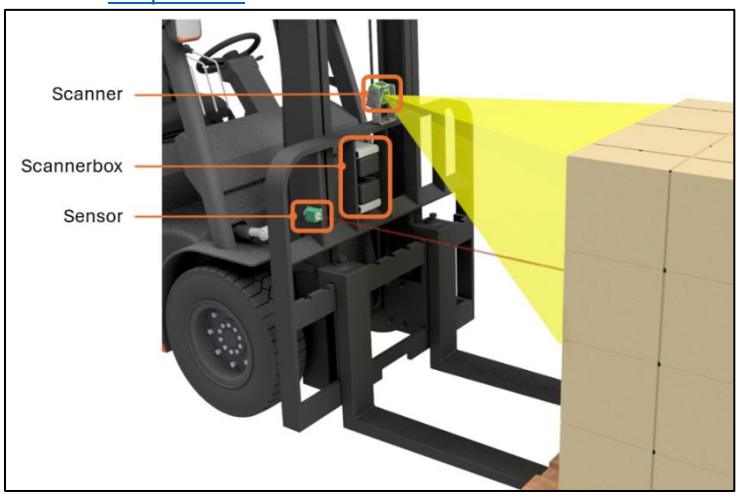
 **Updating the ScanSystems might lead to a total system reset** (all system, and user settings will be reset) depending on if a new firmware version uses a new partition table. You can find info about these partition tables in the **"Bout Solutions - ScanSystems software hardware compatibility list"**.

3. Getting started

This chapter describes the process of preparing the system for full operation on a forklift. It includes hardware installation guidelines, scanner configuration with references to supporting materials such as instructional videos, manuals, and preconfigured ZJobs, as well as important parameter settings including the ETX end character and job timeout. The chapter concludes with a guide on configuring the ScanSystems.

3.1. Installing the hardware

This section explains how to mount the ScanSystems hardware. In most cases, the system is installed on a forklift. The image below shows an example of a forklift equipped with ScanSystems, including a connected scanner and sensor. Details on how these components are wired to the Scannerbox can be found in chapter 1.2.



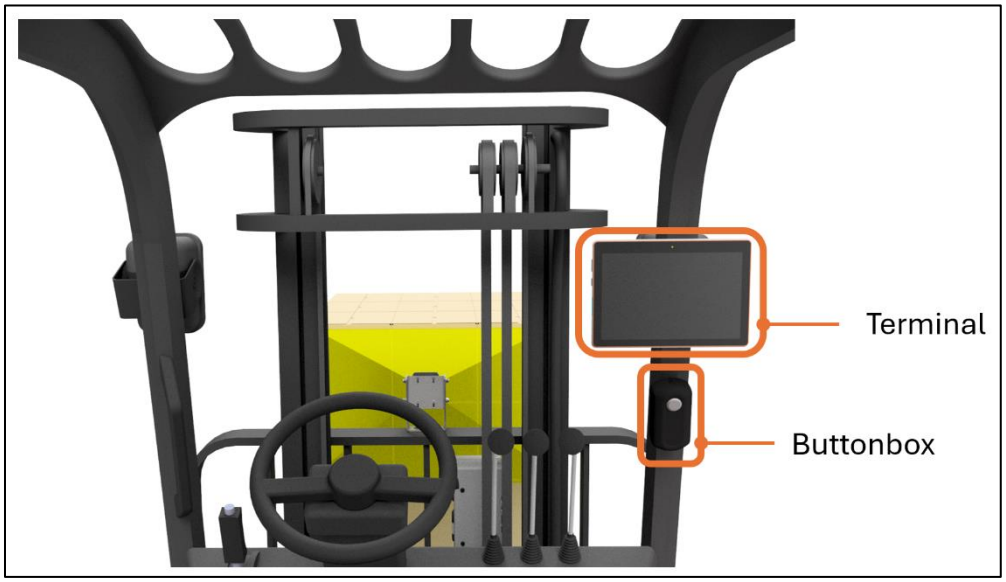
ScanSystems example installation on a forklift

In this configuration, the scanner is mounted using the “50101 Fixed Vertical FS4x” while the ScanSystem is secured with the “50118 Fixed Battery Power”. Both mounting solutions are part of Bout Solutions’ product portfolio and are available at www.boutsolutions.nl. The system in this setup is battery powered. With one battery and a single scanner connected, it can operate continuously for approximately 15 hours.

The ScanSystems is equipped with two battery slots, allowing for hot-swapping batteries without power interruption. This ensures continuous operation during battery changes.

During installation, it is essential to ensure that any load handled by the forklift does not come in contact with, or cause damage to the installed components. Proper positioning and secure mounting are crucial not only to prevent interference between the load and the equipment, but also to avoid errors caused by scanners shifting position and failing to capture the correct field of view. This ensures both safe operation and accurate system performance.

The Buttonbox is the easiest component to install. Simply connect it to the terminal and ensure that it functions as a keyboard input within the terminal environment. The image below shows the driver's view with the terminal and Buttonbox clearly visible.



ScanSystems forklift drivers view

3.2. Setting up the scanners

Scanners need to be focused at the correct distance to ensure reliable reading. Each scanner, and lens configuration has a specific focal point, and it is essential that this matches the location where the pallet is being scanned. For the Zebra FS40 Wide Angle, this focus range is typically between 80 and 120 cm.

The same principle applies to ultrasonic sensors, which are commonly used because they are not affected by reflections, lighting conditions, or color differences. Ultrasonic sensors have a blind zone, which is a limited area directly in front of the sensor where object detection is not possible. It is important to keep this area clear to prevent false triggers.

When the system is automated with a sensor, the forklift often picks up loads with fast, continuous motion. To ensure scanning happens at the right moment, we recommend setting the sensor trigger point slightly further forward. This way, the scan is activated only when the pallet is fully within range.

3.3. Configuring the ScanSystems

The 2D (smart) scanners are configured using the manufacturer's software, for Zebra devices this is Zebra Aurora. In Aurora, you can create jobs to save and copy configurations. Our RFID scanner is automatically preconfigured. This paragraph explains the process of manual configuration, the use of standard jobs, and the role of the configuration tool in setting up the system.

Manual Configuration of 2D Scanners:
It is important that the scanners are correctly configured to work together with ScanSystems. The video on www.boutsolutions.nl explains how to configure the scanners using Zebra Aurora. The step-by-step guide to manually configure the system is provided here, if you can't watch the video for any reason:

1. Download the software that supports your scanner.
2. Configure your communication settings:
   - Protocol: RS-232
   - Baud rate: 9600
3. Configure your I/O settings:
   - Activate GPIO 0
   - Set GPIO 0 as a normally acting input
   - Input is rising edge
4. Create a Job:
   - Capture your image
     - Set trigger to GPIO 0, level continuous, rising edge
     - Ensure your acquisition settings are suitable for your environment
   - Build the job
     - Use an active timeout of 2000ms (compatible with the default ScanSystems user settings)
     - Make sure QR codes can be scanned, including the codes relevant for your application
     - Locate advanced filtering options for the scanner and ensure you add <ETX> to the end of your data string

 Do not use any **<ETX>** characters in your scanner output as delimiter for example, only as an end suffix. Otherwise, the system will think that the end of the output from one of the scanners has been reached. The Scannerbox basically sees the <ETX> as a stop bit. All data after <ETX> will be cut off.

Standard ZJobs:
Pre-configured ZJobs are available for the Zebra fixed scanner range. These ZJobs can be downloaded from www.boutsolutions.nl. However, the acquisition settings cannot be preset, as they depend on the specific codes and the environment.

ScanSystems Configuration Tool:
The configuration tool was developed to enable flexible deployment of the ScanSystem and to provide a tailored solution for a wide range of logistical challenges. The tool is quite comprehensive and has its own user manual, which can be found at www.boutsolutions.nl. You can also ask your ScanSystems contact person about this tool.

The configuration tool functions as a parameter list that can be filled in according to your requirements. Using default parameters, the system can already be operated. Once the parameters are set, a QR code can be generated. This QR code can be scanned by a scanner connected to our system. After scanning, the parameters are applied to the system, and it will operate as configured.

 When using the configuration tool**, it is necessary to scan QR codes to configure the system.** Make sure the scanner is configured with the correct symbologies to read QR codes.

4. Troubleshooting

If you encounter problems with your ScanSystems, please look at possible solutions in this chapter before contacting your ScanSystems contact person.

When reading this chapter, the manual will presume that you know the component names of external and internal IO, described in Attachment 2. The explanation will use these names.

When contacting your ScanSystems contact person, always provide them with information about what problem you're facing, and which troubleshooting steps you have taken if any.

4.1. Troubleshooting problems

In chapter 4.1.1, you will find a list of all problems you could troubleshoot yourself. Problems are sorted in order of severity (**High**, **Medium**, and **Low**). Start troubleshooting from the highest severity problems to the lowest one. When you follow a troubleshooting path, it may direct you to a **procedure** (chapter 4.2). A procedure is an action that is standard for different types of problems, so in order to reduce the document size and avoid writing the same procedures for parts of different problems; these are grouped in the Procedures chapter (chapter 4.2).

Make sure before, and after troubleshooting a problem to reset the device to the following state:

- Provide power to the system.
- Disconnect any edge devices from the system like scanners, or sensors, and connect them again.
- The device should be fully assembled (housing is closed, but screws don't have to be inserted per se for quick troubleshooting purposes).

4.1.1 List with possible problems

High:

Scanners don't seem to get triggered by Scannerbox:

When you trigger your Scannerbox, and it doesn't seem to trigger the scanners attached to its scanner ports, there could be a variety of problems at play.

Firstly, it's a good idea to check if your scanner(s) really isn't/aren't getting triggered. The Zebra FS series scanners have a few lights on top to indicate the status of the scanner. When an FS scanner is triggered, it doesn't necessarily generate a sound, or blink it's build in lighting. What it does do is show you an LED on the top which lights up while it's scanning. Look at the LED's of the scanner, trigger it, and see what happens. If no LED is lighting up, then please look at the trigger configuration of your scanner in procedure 3. If this doesn't fix the problem, please continue reading.

If you have an external sensor attached to the Scannerbox, please try triggering it with that sensor instead of the Buttonbox, since there could be a connection problem.

If scanning with the sensor works, then you might need to reconnect Scannerbox, and Buttonbox again. You can do this by following chapter 2.2. If this doesn't resolve the issue, it could also be that your Scannerbox, and Buttonbox firmware versions different from each other. You can check for this by following procedure 4, and running the **system summary** command. If the firmware versions of your Buttonbox, and Scannerbox match, please contact your ScanSystems contact person. If they don't match, also contact your ScanSystems contact person, since either the Buttonbox, or Scannerbox will need an update, so the devices firmware versions match again.
If using the sensor to trigger the Scannerbox also doesn't work (or if you can't test with an external sensor), please read along.

It is a good idea to restart the ScanSystems at this point. Please unplug your Buttonbox from the terminal it's attached to, and also remove power from the Scannerbox. Now plug the Buttonbox into the terminal again, and provide power to the Scannerbox again. If this resolved the issue, please report this information to your ScanSystems contact person via procedure 1, since you might have encountered a firmware bug (assuming you haven't swapped a 2D code scanner for an RFID scanner, and vice versa without restarting, which can cause this kind of behavior). If restarting the ScanSystems didn't work, please read further along.

Now we will need to start opening up the system, to check if the ScanSystems is actually receiving power. To do this you will need to follow procedure 5. If the Scannerbox, or Buttonbox. If the system isn't receiving power, and you can't test using different power sources, please contact your ScanSystems contact person. If the system is receiving power, continue reading.

While having the Scannerbox, and Buttonbox opened up, we can check its **Status LED**s. The location of these LEDs is described in Attachment 2 under the **Scannerbox overview**, and **Buttonbox overview**. Please perform a trigger with Buttonbox, or external sensor attached to the Scannerbox, and see if the Status LEDs light up. You can also see what the status of the system is when the LEDs light up, by looking at the following table in chapter 2.6. If the LEDs of the Scannerbox light up when triggering it, something might be wrong with the attached scanners, or cable connecting them to the system. Contact your ScanSystems contact person in this case. When the LEDs don't light up, the system might be locked since the Scannerboxes/Buttonboxes **dipswitches** are all set away from the numbers depicted on the switches. This means the ScanSystems won't boot, since these switches are used for a physical factory reset (chapter 2.5.2.2). If setting these switches back towards the numbers doesn't work, please contact your ScanSystems contact person.

High:

Scanners are being triggered, but nothing happens afterwards:

When you trigger your Scannerbox, and it triggers the scanners attached to its scanner ports, but nothing happens afterwards, there are a few possible causes.

The first possible problem we need to rule out is a wrong configuration of scanners. Please follow procedure 3 to see if your scanner(s) is/are configured correctly.

If your scanners are configured correctly, it's a good idea to restart the ScanSystems at this point. Please unplug your Buttonbox from the terminal it's attached to, and also remove power from the Scannerbox. Now plug the Buttonbox into the terminal again, and provide power to the Scannerbox again. If this resolved the issue, please report this information to your ScanSystems contact person via procedure 1, since you might have encountered a firmware bug (assuming you haven't swapped a 2D code scanner for an RFID scanner, and vice versa without restarting, which can cause this kind of behavior). If restarting the ScanSystems didn't work, please read further along.

It now is a good idea to check if the connection between Scannerbox, and Buttonbox is working as it should. Follow procedure 4 to run **diagnostics**. If the Buttonbox outputs data to your screen as seen in procedure 4, this means your Scannerbox ,and Buttonbox are properly connected to each other, and the Buttonbox can output data to your terminal.
If you don't see any output from diagnostics, make sure that your Buttonbox is put in the correct output mode (HID, or USB serial output mode). You can set in which mode the Buttonbox should print data by following chapter 2.5.1. If this doesn't solve the problem, please read further along. It could also be that Scannerbox, and Buttonbox aren't properly connected to each other. Please follow chapter 2.2 to reconnect Scannerbox, and Buttonbox to each other. If this doesn't make a difference, contact your ScanSystems contact person.
If you do see output from diagnostics, it could be that you've configured the ScanSystems with some settings that might prevent the data from getting through. You can try a factory reset of the settings at that point (chapter 2.5.2), or look at the settings of your system, by running the **system summary** command when following procedure 4. If this doesn't help, please contact your ScanSystems contact person.

Low:

When triggered, the ScanSystems types demo mode data:

When you trigger the Scannerbox, and the Buttonbox prints either of the following messages to your screen: "ScanSystems is in demo mode. Wait **x**m until demo starts, or unlock ScanSystems. Demo key: **xxxxxxxx**", or "ScanSystems demo ended, Restart devices, and wait **x**m until demo starts, or unlock ScanSystems. Demo key: **xxxxxxxx**". This means your ScanSystems is in demo mode. You will need to contact your ScanSystems contact person to unlock the system from demo mode with your **demo key**. Note that each time you restart the Scannerbox, a new demo key is generated. You can also continue the system in demo mode if you follow the instructions the ScanSystems printed to your screen.

4.2. Procedures

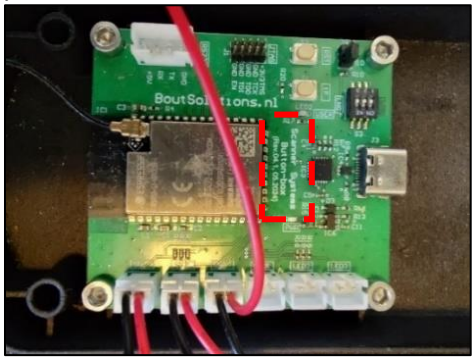
1 (Contact your ScanSystems contact person about the problem):
Please contact your ScanSystems contact person, and provide them with the following information:

- Explain that you have encountered a problem with the ScanSystems, and that you followed **x** troubleshooting steps.
- Explain what hardware you're using (what is the scanner/sensor configuration on your ScanSystems, are you powering the Scannerbox with external batteries, or via external power). What terminal do you have the Buttonbox connected to.
- What ScanSystems firmware are you using, and what hardware revision (you can find this by following **procedure 2**).
- Describe how we might be able to reproduce the problem if possible (did ScanSystems work normally before, and what happened after it stopped working normally).

2 (Retrieve your ScanSystems hardware revision, and firmware version):
Finding your ScanSystems hardware version
To get to know the hardware revision of your ScanSystems, simply open the casing of the Scannerbox, and Buttonbox. The Scannerbox PCB should look something like in the image below. The area marked in **red**, shows you where you can find the hardware revision of the Buttonbox.

Scannerbox PCB

The Buttonbox PCB should look similar to the image below. The area marked in **red**, shows you where you can find the hardware revision of the Buttonbox.


Buttonbox PCB

Finding your ScanSystems firmware version:
To find out what firmware the ScanSystems is running, there are several ways to know depending on how the ScanSystems system is you have.

If you know your ScanSystems isn't updated, you can use several methods to find out the firmware version currently running on your ScanSystems.

On the latest versions of the ScanSystems hardware, you can find on the label on the back of the Buttonbox, or Scannerbox if you got a newer version of the system. If you got an older version, you can look up when the system was ordered, since Bout Solutions might know when that version was deployed on your system. You can also find the release dates of the ScanSystems firmware in Attachment 1. There's one other indicator. Starting with **v24**, when triggering all scanners, the LED on the Scannerbox PCB lights up white instead of the old versions where it lights up blue. If you have a Buttonbox with the red PCB revision, the version on the system has to be lower than **v28.2**, since after **v28.2**, the red Buttonbox PCB revision is not supported anymore.

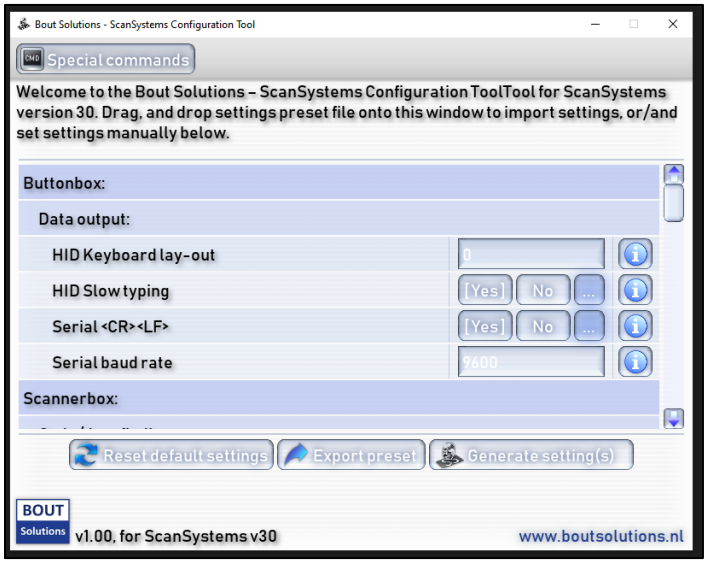
When your ScanSystems has been updated before to **v30**, or higher, you can find out by running a special command that will tell you the current firmware version on your ScanSystems devices. You can find out how to generate, and scan special commands into your ScanSystems by reading the **"Bout Solutions - ScanSystems Configuration Tool manual"** for the latest ScanSystems version. The special command that will print the firmware version of Scannerbox, and Buttonbox is the "**System summary"** command. This command should be preserved throughout ScanSystems firmware versions, so it will probably work, no matter what version of the **ScanSystems Configuration Tool** you use. If scanning this special command doesn't work, please contact your ScanSystems contact person.

3 (Check if your scanner(s) is/are configured correctly):
In order for 2D code scanner(s) to work with the ScanSystems, you need them to be configured so that it/they will be triggered by a short pulse on a rising edge, via the **Scanner connectors trig. pin**. This pin should connect to the IO of your Scanner(s). Besides this, the Scanner(s) should also send an **<ETX>** character at the end of their data transmission (after typing all scanned codes, delimiters, and suffixes etc.). This character is very important. Without it, the Scannerbox discards the data for this scanner, since it thinks it's corrupted. Also make sure your scanner(s) is/are configured to scan the correct symbologies. Lastly, RS232 output on your scanner needs to be enabled at a baud rate of 9600.

If you are using Zebra FS series scanners with the ScanSystems, there are a few things you can do to make sure they are configured correctly, besides the aforementioned information. On our site: www.boutsolutions.nl, you can find an instruction video on how to setup Zebra FS series scanners. If you've just updated your Zebra scanner, factory reset it. This is important, and is also explained in the FS series manual. The scanners have the tendency to not output RS232 anymore after a firmware update without factory resetting them. You can also easily use one of our custom ZJobs (corresponding to your ScanSystems firmware version) on the scanners, which are guaranteed to work for the ScanSystems. You can read more about this in chapter 3.3.

4 (Run diagnostics/system summary):

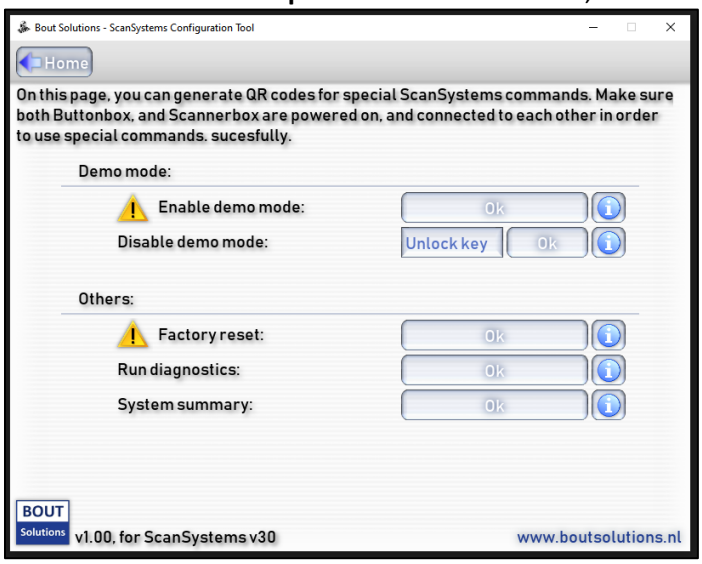
Please start the **ScanSystems Configuration Tool**. You can download this tool from www.boutsolutions.nl, or ask your ScanSystems contact person about it. Make sure to download/ask for the correct version of the application for your ScanSystems firmware. Once you've opened the tool, you are greeted with the **Main screen** like in the picture below (it could differ a bit from what you might see depending on the revision of the application):



ScanSystems Configuration Tool - Main screen

Now click on the "**Special commands**" button in the top left corner of the main screen.

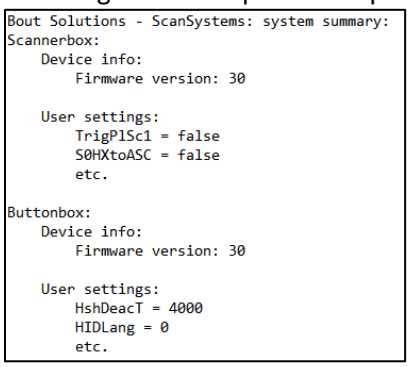
You will now be at the **Special commands** screen, like in the picture below:



ScanSystems Configuration Tool - Special commands screen

From here, you can run either a **system summary**, or **diagnostics**. A system summary will get you the firmware version the Scannerbox, and Buttonbox are currently running, and a summary of all the **user settings** values the ScanSystems is currently using. The diagnostics **special command**, will test the connection between Scannerbox, and Buttonbox, and output some test data. Just click on the "**Ok**" button for the command you want to execute, and read the instructions described in the message that now pops up. After that, you can scan the generated special command QR code to execute the desired command on your ScanSystems.

The image below depicts example output from the "system summary" special command:



```
Bout Solutions - ScanSystems: system summary:
Scannerbox:
    Device info:
        Firmware version: 30

    User settings:
        TrigP1Sc1 = false
        S0HXtoASC = false
        etc.

Buttonbox:
    Device info:
        Firmware version: 30

    User settings:
        HshDeacT = 4000
        HIDLang = 0
        etc.
```

ScanSystems system summary export example (shortened ver.)

You can see the firmware version under the Scannerboxes, and Buttonboxes "**Device info**" section.

When running the "diagnostics" special command, the Buttonbox should print the following text to your screen: "Bout Solutions - ScanSystems: Diagnostics success."

5 (Check if ScanSystems is powered on):
First remove the cover from your Buttonbox, and Scannerbox (you don't need to disconnect power from the devices). Now you should see the PCB of the Scannerbox, and Buttonbox as shown in the **Buttonbox internal IO description**, and **Scannerbox internal IO description** images, depicted in Attachment 2 under the **Buttonbox overview**, and **Scannerbox overview** section. If the **Power LEDs** on the devices are lit up green, the devices are receiving power.

Attachment 1. (System summary settings translation table)

When you execute the "**System summary**" special command in the ScanSystems, the Buttonbox will print a report of some system variables of Scannerbox and Buttonbox, including what their current user settings are set to.

However, the names (keys) of the user settings the ScanSystems uses internally are shortened names of the settings names the **Bout Solutions - ScanSystems Configuration Tool** uses. This is due to efficiency reasons. You can see this in the picture below, of a ScanSystems system summary:

```
Bout Solutions - ScanSystems: system summary:
Scannerbox:
    Device info:
        Firmware version: 30

    User settings:
        TrigP1Sc1 = false
        S0HXtoASC = false
        etc.

Buttonbox:
    Device info:
        Firmware version: 30

    User settings:
        HshDeacT = 4000
        HIDLang = 0
        etc.
```

ScanSystems system summary export example (shortened ver.)

Below is a table containing the translations between the settings keys used by the ScanSystems internally, and the Configuration Tool. This way you can deduct which settings are what from the System summary. These settings are sorted alphabetically the way they are sorted in the Configuration Tool. As you might be able to see, the Configuration Tool also doesn't show all settings the ScanSystem has to offer, since these settings are automatically calculated by the Configuration Tool. These settings are colored a **dark yellow**. The ScanSystems will print out these hidden settings during the system summary, but you can ignore them.

Setting name in Configuration Tool	Setting name on ScanSystems
Buttonbox:	
Data output:	
HID Keyboard lay-out	HIDLang
HID Slow typing	HIDST
Serial <CR><LF>	SerialCRLF
Serial baud rate	SerialBaud
Other	
SafetyHandshakeReactivationTime	HshDeacT
Scannerbox:	
Code/data finding:	
Maximum time multiplier to look for data	MaxLpCt
Only look for one scanner's data	LFSSDta
Recognition pattern scanner 1	RcPtnSc0
Recognition pattern scanner 2	RcPtnSc1
Duplicate scanner data:	
Cross check last scanner data for duplicates	CsCkSDLR
Dup. scnr. data allowance cur. round	DSdtaAlw
Keep last scnr. data of non trig. scnr.	KNTSLDta
Remove last dup. data	RmDupSc
HID:	
Scanner data delimiter	Delim
Scanner data prefix	Prefix
Scanner data suffix	Suffix
Other:	
Convert HEX to ext. ASCII scanner 1	S0HXtoASC
Convert HEX to ext. ASCII scanner 2	S1HXtoASC

SafetyHandshakeReactivationTime	HshDeacT
RFID:	
Always look for tags	AGRFID
Min. card str. to register scanner 1	MCSRFID0
Min. card str. to register scanner 2	MCSRFID1
Reader strength [dB/1000]	DbRFID
Working area	AreaRFID
Scanner/trigger related:	
Clear non triggered scnr. data allowance	ClNTSDA
Default scanner trigger by sensor	DfScTrBS
Inverse double IO sensor polarity	InvDSenPol
Inverse single IO sensor polarity	InvNSenPol
Invert trigger polarity scanner 1	TrigPlSc0
Invert trigger polarity scanner 2	TrigPlSc1
Sensor state change time [ms]	SenStChT
Special scanmode	SpScMde
Switch scanner on HI/LO sensor	HiLoSwSc
Trigger mode scanner 1	TrigMdSc0
Trigger mode scanner 2	TrigMdSc1
Trigger on HI/LO sensor	HiLoScn

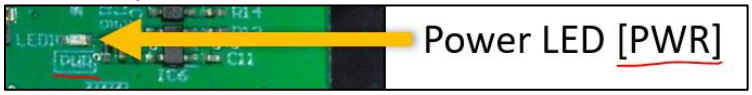
ScanSystems user settings key to Configuration Tool key translation table

Attachment 2. (Scannerbox, and Buttonbox IO overview)

This attachment contains a description of the Scannerboxes, and Buttonboxes external, and relevant internal IO. It is useful to know this information when wanting to connect any type of external power source/device to the ScanSystems, and to connect the ScanSystems to your systems.

The pictures below depict the external and internal IO's of the Scannerbox, and Buttonbox. Relevant IO's contain a footnote in **blue**, so you know which IO is which in the different pictures displayed within this attachment.

Please keep in mind that **you could have a later PCB revision of the Scannerbox/Buttonbox PCB than** what you will see **in this attachment**, but **the IO should generally be the same.** For this reason, most IO's have a label behind their names in square brackets, describing the PCB silkscreen name they have. This silkscreen IO name most likely won't change throughout PCB revisions, thus you can almost always find IO's by their silkscreen name if you have a different looking PCB. In the example picture below you will see the power LED of the Buttonbox: You can see that the silkscreen name of this output (marked in **red** on the PCB) is called "PWR", which corresponds to the IO descriptions silkscreen name between square brackets.

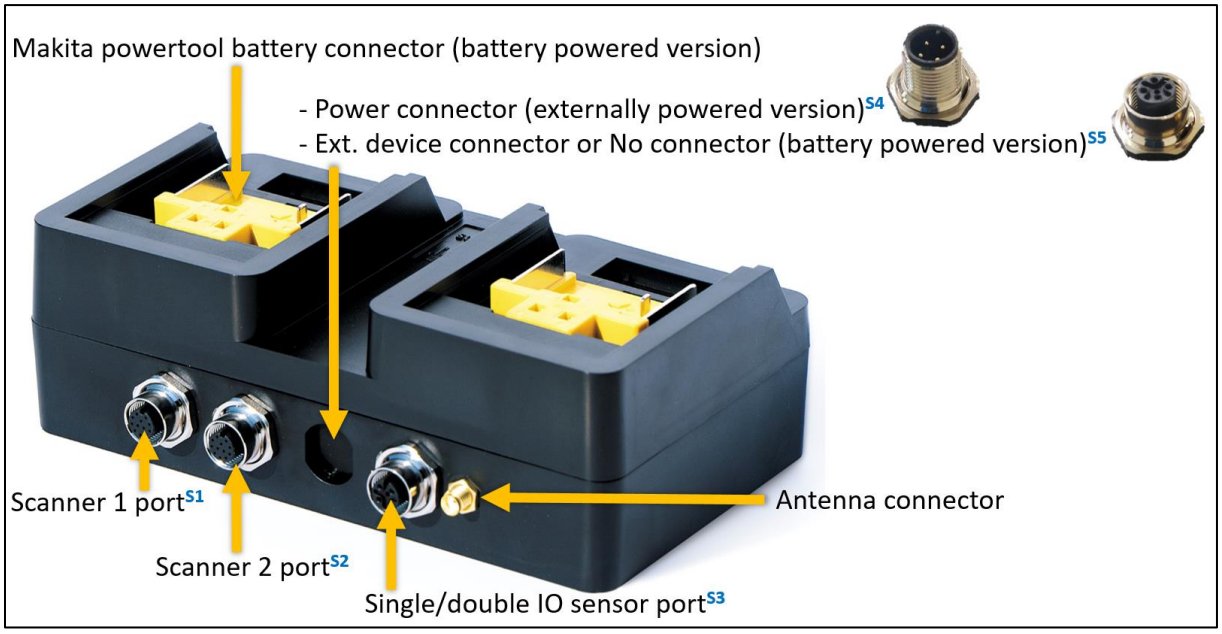


ScanSystems PCB silkscreen IO name mapping example

Throughout the pictures you will also see some device pins are named "**+ vSB**". This means that the system is providing the voltage level you put into the system via batteries/a power connector/(USB connector for the Buttonbox) at that pin. So say you have an 18v battery connected to the Scannerbox, the voltage that will come out to **+ vSB** will also be 18v.

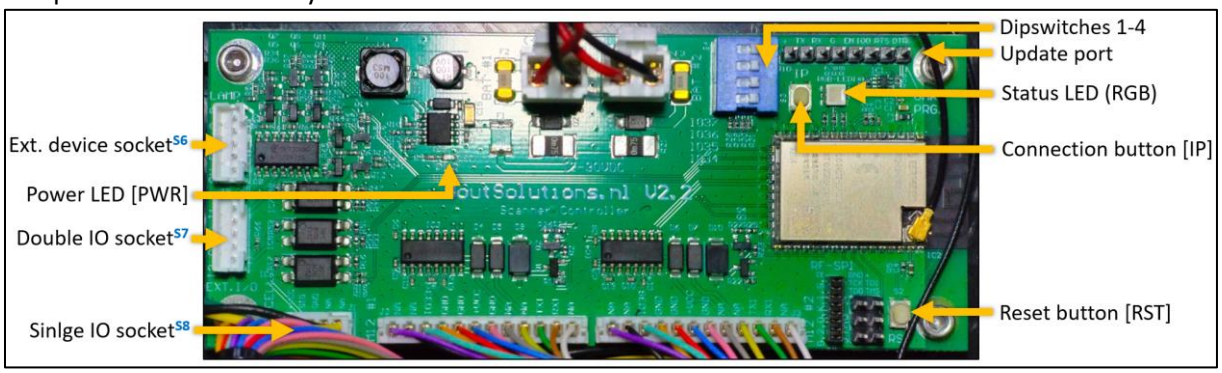
Scannerbox overview

The image below depicts the outside of the battery powered Scannerbox, but this overview is also applicable for the externally powered Scannerbox:



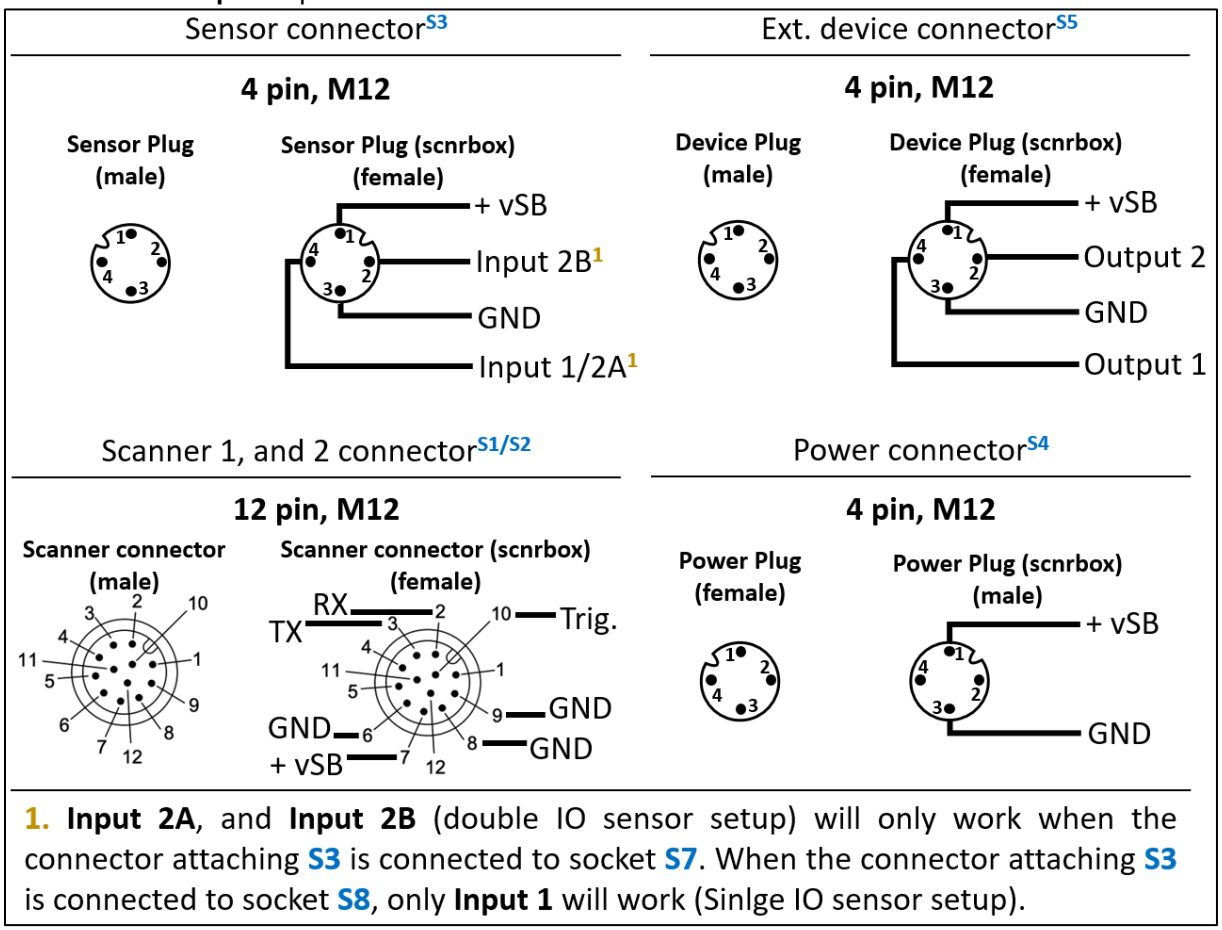
Scannerbox external IO description

The picture below shows you the internal IO's of the Scannerbox:



Scannerbox internal IO description

In the picture below, you can see the pinouts of the different connectors depicted in the "**Scannerbox external IO description**" picture earlier:

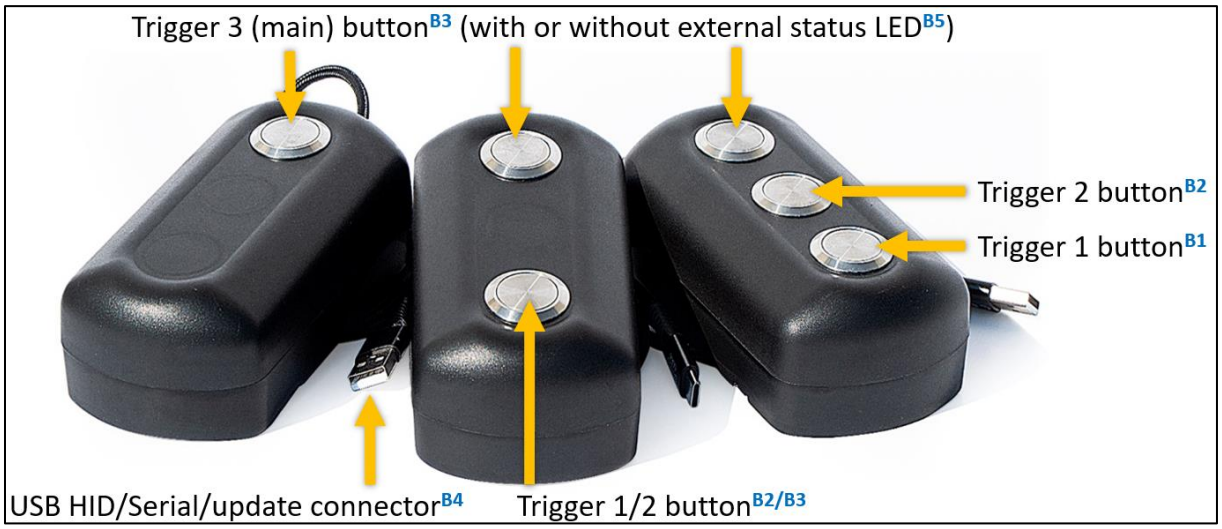


1. **Input 2A**, and **Input 2B** (double IO sensor setup) will only work when the connector attaching **S3** is connected to socket **S7**. When the connector attaching **S3** is connected to socket **S8**, only **Input 1** will work (Sinlge IO sensor setup).

Scannerbox external connector pinout information

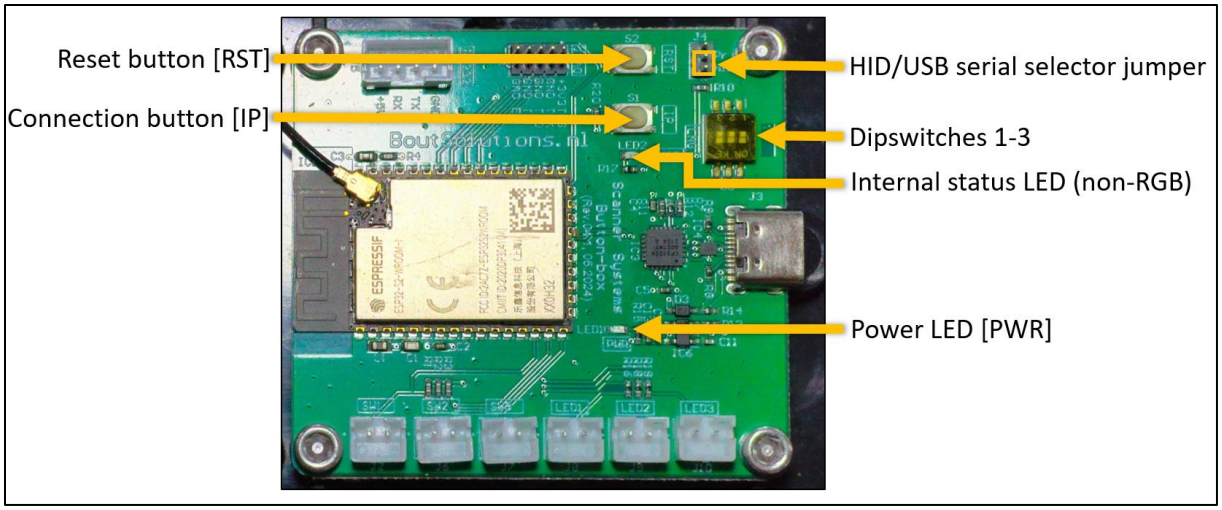
Buttonbox overview

The image below depicts the different default configurations of the Buttonbox you can get, to show you which buttons on which default configurations do what. Of course, this can always be customized if the customer wants it:



Buttonbox external IO description

The picture below shows you the internal IO's of the Buttonbox:



Buttonbox internal IO description

Attachment 3. (ScanSystems specifications)

The Bout Solutions ScanSystems is a plug and play system, which solves all kinds of logistical problems where 2D/RFID code scanning is required on moving systems (like on a forklift), or static applications like production line scanning. It is saves costs and time, by being able to trigger scanners automatically via sensors of choice, and by having a wireless connection between Buttonbox, and Scannerbox.

Plug and play

ScanSystems does not need any drivers, or software to be installed onto your system. It also uses a closed **peer-to-peer** Wi-Fi network to communicate, eliminating the need to login to your Wi-Fi network. The System can be plugged into your system of choice, with just a single USB cable.

Safety and speed increase on moving systems

Scanning of 2D codes, and RFID tags can happen from a distance. Scanners can be triggered safely and automatically by sensor on the fork board, or by hand with the Buttonbox inside the forklift cabin. This saves time in contrast to scanning by hand, and is very safe.

In the case of a moving application like forklift scanning, this means that forklift operators will not have to get out of their cabin in order to scan a code with a hand scanner. Operators, who want to save time by putting their arms at risk by putting it through the forklift cabin, also will not need to do this anymore. This example forklift case is also applicable to other moving warehouse operations!

Eliminating extra costs

The connection between scanners (attached to the Scannerbox), and terminal where data will be input (attached to Buttonbox) is wireless up to 20 meters. This means that applications, which have moving parts between scanner and terminal, do not need cables routed through the system, which will wear out over time due to movement.

Think about a forklift terminal connecting scanners to the fork board. Using a conventional solution, cables would need to be routed through the forklift mast. These cables usually only survive one to two years based on field data, and cost a lot to replace. The ScanSystems eliminate this problem by going wireless.

ScanSystems specifications

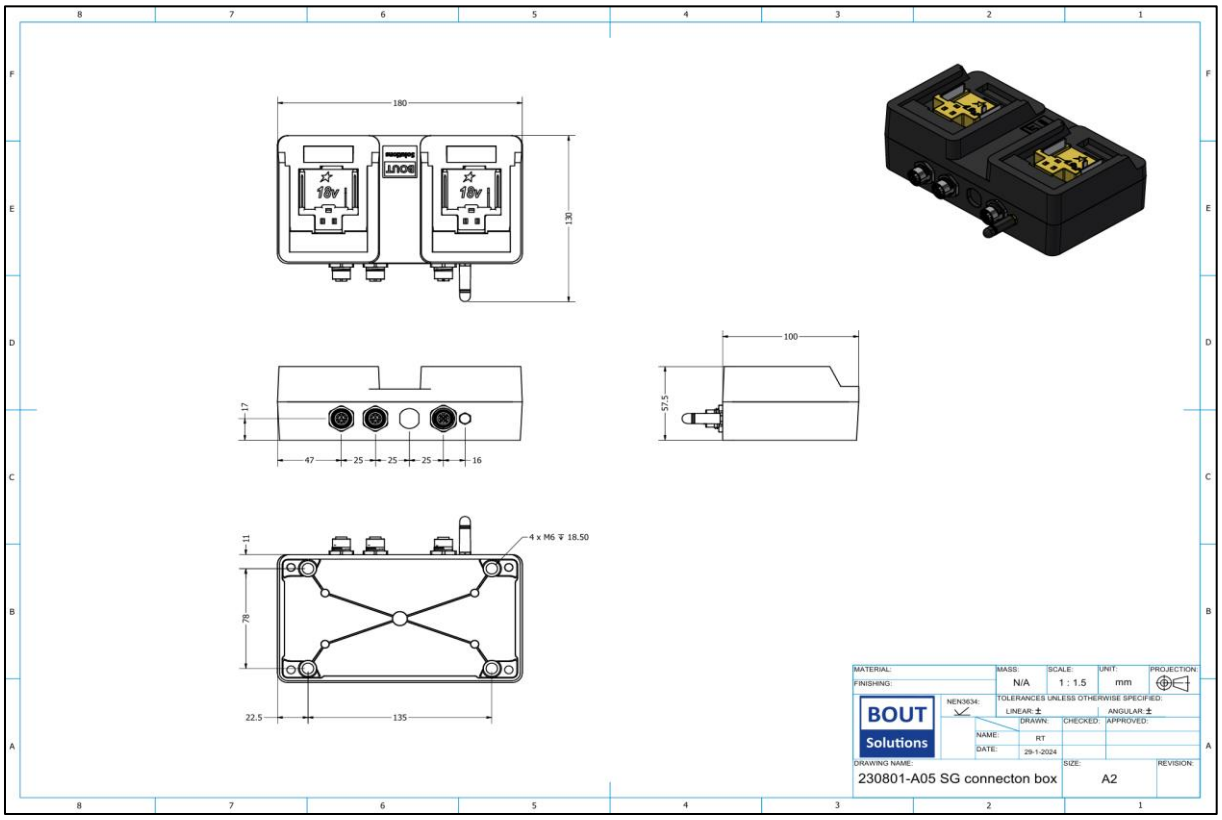
Component	Buttonbox	Scannerbox
Device overview		
Weight	~150g (one button configuration) ~184g (three button configuration)	~447g (battery powered version), or ~346g (externally powered version)
Dimensions (base unit)[1]	- Unit: W:67mm*L:133mm*H:41mm (without cable) - Cable: L:1m	W:180mm*L:100mm*H:57,5mm (battery powered version), or W:180mm*L:100mm*H:43mm (externally powered version)
Power input	Via USB cable	12-30v at 5amps via external power, or battery
Communication between Scannerbox, and Buttonbox	2.4Ghz Wi-Fi, Peer to peer closed network. Communication can reach up to 20m without any metal obstructions between the devices.	2.4Ghz Wi-Fi, Peer to peer closed network. Communication can reach up to 20m without any metal obstructions between the devices, and with the factory default external antenna attached.
Triggers	- Momentary reset button - Momentary connection button - Three dipswitches - 1-3 momentary push buttons	- Momentary reset button - Momentary connection button, four dipswitches
Connectivity		
Power	USB C cable (to USB A with adapter)	Sliding power tool battery connector (battery powered version), or 4 pin M12 female connector (externally powered version)
Scanners	N/A	Two 12 pin M12 female connectors
Sensor input	1-3 momentary built in push buttons	Single, or double IO sensor input via 4 pin M12 female connector
Wi-Fi Antenna	Built in	SMA female connector
Performance		
Battery life	N/A	Performance with one scanner, and one battery attached at 20°, is 16h standby, and 500 scans. The same setup at -20°, will give you a 10h standby time, and 500 scans.
Software		
Updating the system	ScanSystems Updater Tool	ScanSystems Updater Tool
Configuring the system/executing special commands	ScanSystems Configuration Tool	ScanSystems Configuration Tool

ScanSystems specifications table

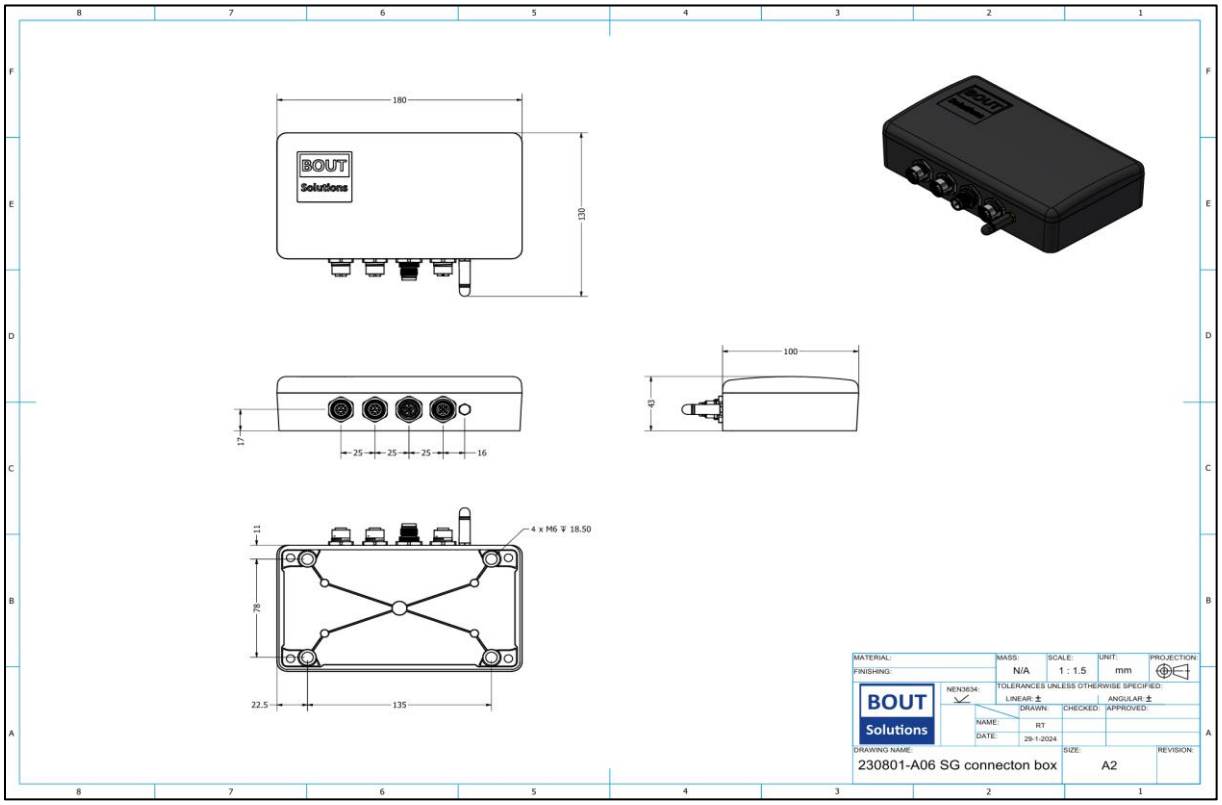
[1]Please refer to the Scannerbox dimensions section within this document, to view all the dimensions of the different versions of the Scannerbox, and Buttonbox in more detail.

Scannerbox dimensions

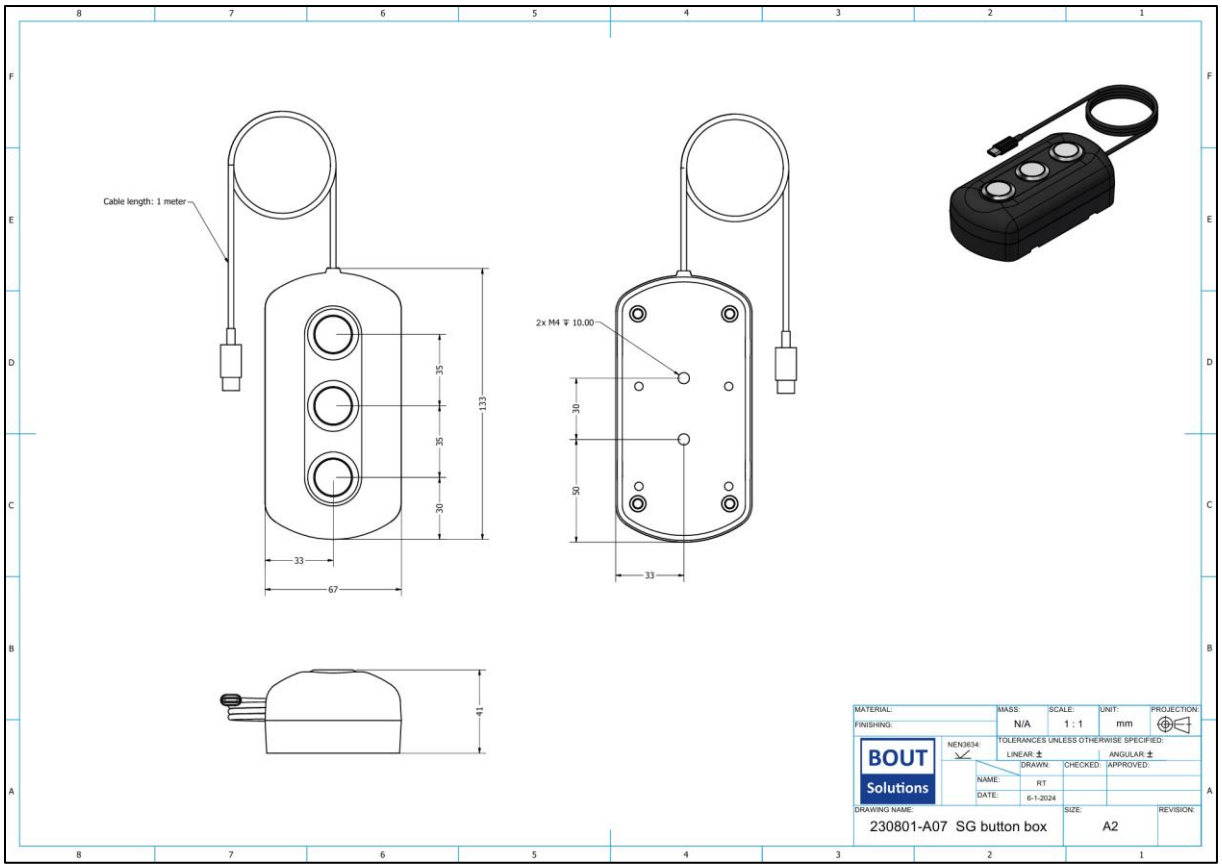
The images below, visualize the dimensions of the different versions of the Scannerbox, and Buttonbox:



Scannerbox (battery powered version) dimensions



Scannerbox (externally powered version) dimensions



Buttonbox (battery powered version) dimensions