



Smart Board 1150

Keyboard with Contact Smart Card Reader

SOFTWARE DEVELOPER GUIDE

Document number: 64410276
Document revision: 1.04
Issue date: Dec 09, 2025

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Product Description	1
1.3	Key Features	1
1.4	Related Documents	2
2	Getting Started	3
3	Interfaces	4
3.1	Host Interface	4
3.2	User Interface	4
3.3	Contact Card Interface	5
4	Configuration	8
4.1	Vendor Commands	8
4.2	Firmware Update	14
5	Appendix	15
5.1	Enable CCID Escape Commands	15
6	Contact	16
7	Revision History	17

1 Introduction

1.1 Purpose

This document serves as a guide for developers who wish to integrate contact Smart Cards according to ISO/IEC 7816-3 or configure the keyboard scanning behavior on the Smart Board 1150.

1.2 Product Description

The Smart Board 1150 (in the following referred to by its abbreviation JK-A1150) is a keyboard with built-in contact Smart Card reader.

1.3 Key Features

- **ISO/IEC 7816-3 T=0, T=1**
- **Easy Integration**
 - Plug and Play without special driver installations
- **Power Management**
 - fully compliant with the low power modes as specified in USB 2.0
 - the host may turn the JK-A1150 off to conserve power
 - the JK-A1150 may wake the host from suspend
- **Firmware Update**
 - option to update the device firmware

1.4 Related Documents

HID	Device Class Definition for Human Interface Devices (HID) Firmware Specification—5/27/01 - Version 1.11
CCID	CCID Specification for Integrated Circuit(s) Cards Interface Devices Revision 1.1, April 22nd, 2005
ISO/IEC 7816-3	Identification cards - Integrated circuit cards Part 3: Cards with contacts - Electrical interface and transmission protocols Third edition - 2006-11-01
ISO/IEC 7816-4	Identification cards – Integrated circuit cards Part 4: Organization, security and commands for interchange Second edition - 2005-01-15

2 Getting Started

Simply plug the JK-A1150 into a free USB port on your PC to get started.

No manual driver installation is necessary and every CCID compliant driver should work with the Smart Card interface on the JK-A1150.

However, take note that most CCID drivers do not allow the execution of CCID Escape commands per default. In case the use of CCID Escape commands is required, refer to *Enable CCID Escape Commands* on how to enable them.

3 Interfaces

3.1 Host Interface

3.1.1 USB

The JK-A1150 provides a USB 2.0 Full-speed (12 Mbits/s) interface, and enumerates as a composite device with the following device classes:

- HID (Human Interface Device, v1.11)
- CCID (Integrated Circuit Cards Interface Device, v1.1)
- DFU (Device Firmware Upgrade, v1.1)

The device firmware may be updated via the DFU interface. See section *Firmware Update* for more information on the update process.

Table 3.1: JK-A1150 Endpoint Assignment

Endpoint	Description	Type	Max Packet Size
EP0	Control EP	Interrupt	64b
EP1	HID Boot Reports	Interrupt IN	8b
EP2	Smart Card	Bulk OUT	64b
EP3	Smart Card	Bulk IN	64b
EP4	Card Presence	Interrupt IN	8b
EP5	HID Media Reports	Interrupt IN	3b

3.2 User Interface

3.2.1 Keyboard

The JK-A1150 provides a keyboard interface according to USB Device Class Definition for Human Interface Devices (HID Specification) version 1.11. It implements two USB protocol stack interfaces:

- Keyboard with BIOS support
- Media & Power Keys interface according to Advanced Configuration and Power Interface

The JK-A1150 is compliant with the Microsoft Copilot Specification and provides a Copilot key between right Alt and right Ctrl.

The key debounce behavior of the JK-A1150 is configurable via CCID vendor commands. This allows for optimal configuration based on the personal typing speed. See *Set Debounce Configuration* for details.

3.2.2 LED

The JK-A1150 is configured with a total of four LEDs. From left to right, these are Num Lock, Caps Lock, CCID Data, and PIN.

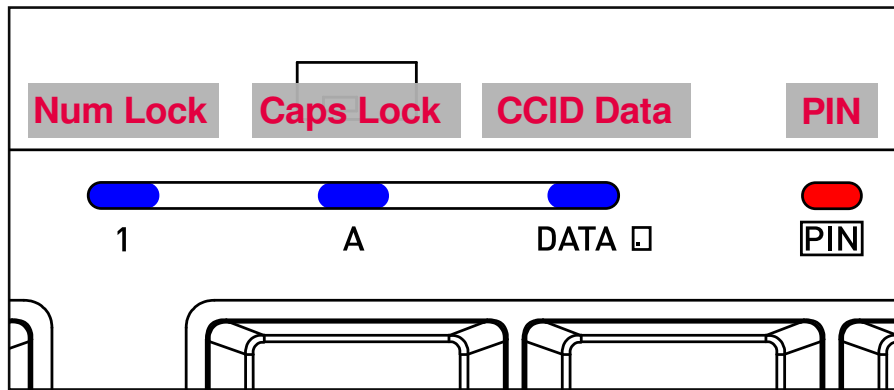


Fig. 3.1: LED placement on the JK-A1150

Num Lock

The Num Lock LED indicates the current function of numeric keypad. Its state is controlled by operating system.

Caps Lock

The Caps Lock LED indicates input mode in which all typed keys are uppercase. Its state is controlled by the operating system.

CCID Data

The CCID Data LED indicates the state of the smart card interface. It is ON when a smart card is inserted and activated. A typical host will deactivate a card after a couple of seconds when it is not in use. In this case, the LED is turned OFF.

Table 3.2: JK-A1150 Status LED

LED State	Meaning
OFF	No card inserted or card inactive
ON	A card is inserted and active
Fast Blink	Exchange with Smart Card ongoing
Slow Blink	Smart Card Error

When a Smart Card is powered and a data exchange is ongoing, the CCID Data LED blinks fast. On electrical errors such as overcurrent, the CCID Data LED blinks slowly until the offending Smart Card is removed from the slot.

PIN

The PIN LED is reserved for future use and currently does not have any functionality.

3.3 Contact Card Interface

The JK-A1150 is compliant with the CCID and ISO/IEC 7816-3 specifications. It provides the following functionality:

- Supported protocols: T=0 and T=1
- Supported baudrates: up to 500K baud (TA1 0x97 @ 4MHz)
- Data Exchange Level: TPDU
- Operation Mode: ISO
- Automatic voltage selection

- Automatic determination of protocol parameters from the card's ATR
- Automatic PPS performed internally
- Automatic IFSD Exchange in T=1

3.3.1 Card Activation

Before data can be exchanged with a contact Smart Card, the card must be activated first. The card activation is typically initiated by the host driver after it has been informed of a change in card presence. The JK-A1150 then powers on the card and selects the ideal voltage class. After the card has been activated, the host may set the protocol parameters or let the JK-A1150 decide internally.

The typical activation procedure is as follows:

1. Card is inserted into a contact slot.
2. Reader notifies host of the card presence
3. Reader waits for PC_to_RDR_PowerOn from the host driver
4. Reader selects a supply voltage and powers up the card
5. Reader waits for PC_to_RDR_SetParameters
6. Reader performs PPS
7. Card is ready

The JK-A1150 supports all card classes specified in ISO/IEC 7816-3. The voltage selection process begins at class C (1.8V) going up, until a valid ATR is received from the card.

The JK-A1150 is capable of determining the best protocol parameters for the card internally. Still, the host driver may change card parameters and select any of the card's supported protocols via PC_to_RDR_SetParameters. If no PC_to_RDR_SetParameters was received before the first data packet the JK-A1150 selects proper parameters and performs the PPS internally.

The following flowchart shows the typical activation procedure as described above:

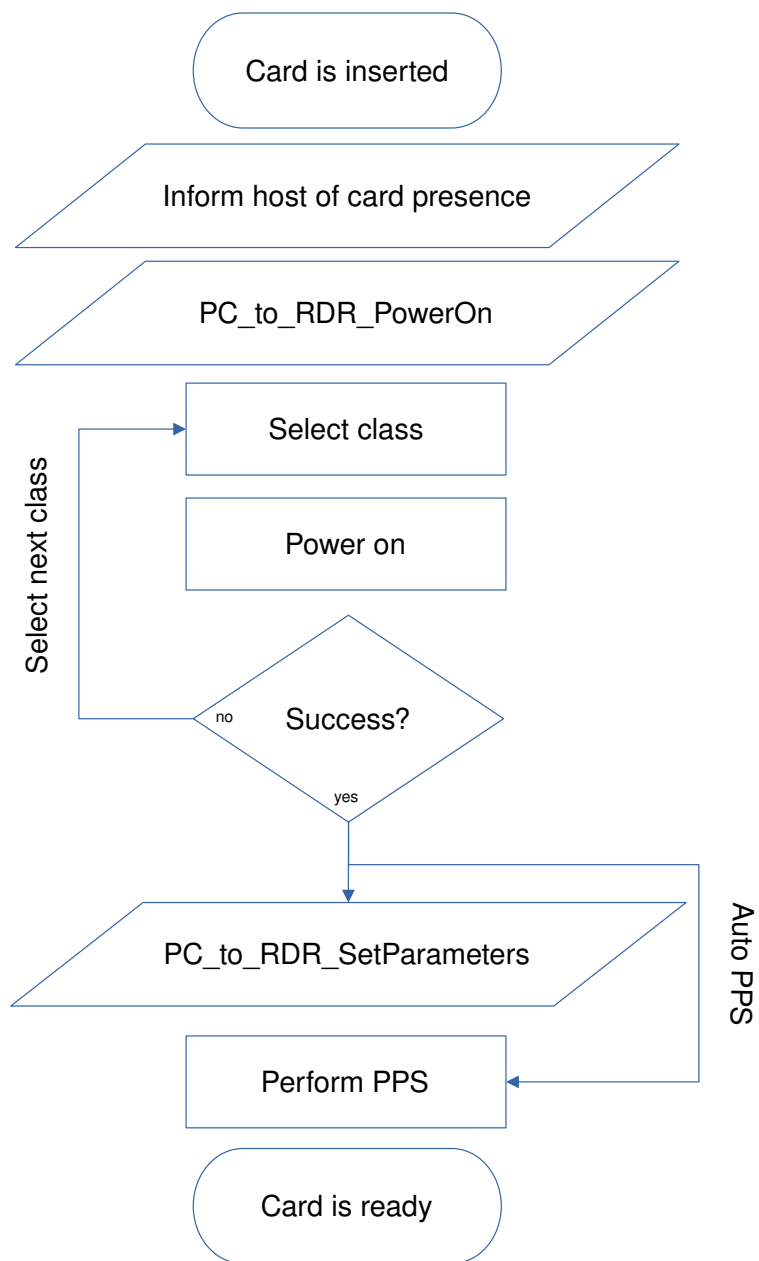


Fig. 3.2: Card Activation Procedure

4 Configuration

4.1 Vendor Commands

Vendor Commands are used to provide additional features not specified in CCID or PC/SC. This may include simple status queries, or configuration of the reader itself. Vendor commands are supported via CCID Escape.

The **CCID Escape** command is the official method to exchange data with the reader directly. No Smart Card is required, but CCID Escape commands must first be enabled in the host driver. Refer to *Enable CCID Escape Commands*.

Several vendor-specific commands are defined to retrieve information about the JK-A1150. Every command shares a defined header, as seen below. The command-specific payload is an ASN.1 DER TLV to request specific configuration items.

Table 4.1: JK-A1150 Generic Vendor APDU

CLA	INS	P1	P2	LC	DATA	LE
FF	70	04	6A	xx	command specific DER TLV	xx

Table 4.2: JK-A1150 Generic Vendor Response

DATA	SW1 SW2
command specific	ISO7816 response code

The response for all vendor commands consists of an ASN.1 DER TLV encoded response payload using the same tag as the request, followed by two status bytes SW1SW2. These are always transmitted as the last two bytes of the response and convey the result of the operation. Refer to ISO/IEC 7816-4 for all defined response codes. A selection of response codes relevant to the Vendor commands are listed in *Vendor response codes* below.

Command and response payload encoding explained based on the Vendor command:

```
→ FF 70 04 6A 06 E0 04 DF 82 00 00 00 Request APDU "Vendor"
← DF 82 00 06 43 48 45 52 52 59 90 00 Response APDU encodes "CHERRY"
```

The request APDU consists of the following parts:

- FF 70 04 6A (General Header)
- 06 (length of payload)
- E0 04 (ASN.1 TAG: GET operation)
- DF 82 00 (ASN.1 TAG: request-specific)
- 00 (length of expected response; return all available data)

The response APDU consists of the following parts:

- DF 82 00 (ASN.1 TAG of request)
- 06 (length of payload)
- 43 48 45 52 52 59 (payload: CHERRY, ASCII encoded)
- 90 00 (response status OK)

Table 4.3: Vendor response codes

SW1 SW2	Comment
90 00	Success
67 00	Error, wrong Le
6B 00	Error, wrong P1 P2 (VID)
6A 80	Invalid data provided
64 00	No data available

4.1.1 Commands

Overview of supported Vendor commands

Command	Type	Tag	Get/Set	Example Result	APDU
Vendor	String	100h	GET	CHERRY	FF 70 04 6A 06 E0 04 DF 82 00 00 00
Firmware Label	String	102h	GET	<full label>	FF 70 04 6A 06 E0 04 DF 82 02 00 00
Firmware Version	2 Octets	103h	GET	1.04	FF 70 04 6A 06 E0 04 DF 82 03 00 00
FW Source Reference	10 Octets	105h	GET	SHA1	FF 70 04 6A 06 E0 04 DF 82 05 00 00
Get Debounce Config	3 Octets	106h	GET	<config>	FF 70 04 6A 06 E0 04 DF 82 06 00 00
Set Debounce Config	<none>	106h	SET	<status>	FF 70 04 6A 09 E1 07 DF 82 06 03 xx xx xx 00
Bootloader Label	String	202h	GET	<full label>	FF 70 04 6A 06 E0 04 DF 84 02 00 00
Bootloader Version	2 Octets	203h	GET	1.04	FF 70 04 6A 06 E0 04 DF 84 03 00 00
BL Source Reference	10 Octets	205h	GET	SHA1	FF 70 04 6A 06 E0 04 DF 84 05 00 00

Vendor

Used to retrieve the vendor name, CHERRY.

Command	CLA	INS	P1	P2	LC	DATA	LE
Vendor	FF	70	04	6A	06	E0 04 DF 82 00 00	00

The response contains the ASCII vendor name.

Example Exchange returns CHERRY:

```
→ FF 70 04 6A 06 E0 04 DF 82 00 00 00
← DF 82 00 06 43 48 45 52 52 59 90 00
```

Firmware Label

Used to retrieve the ASCII firmware label. The firmware label is constructed from:

- Device name (JK-A1150)
- Label id (FW)
- Memory type ([F]lash / [R]om)
- FW version (major.minor-patch)
- FW source ref (first 5 bytes)

Command	CLA	INS	P1	P2	LC	DATA	LE
FW Label	FF	70	04	6A	06	E0 04 DF 82 02 00	00

Example Exchange returns JK-A1150_FW_F-1.06-0-GD0F2DBF6E5:

```
→ FF 70 04 6A 06 E0 04 DF 82 02 00 00
← DF 82 02 20 4A 4B 2D 41 31 31 35 30 5F 46 57 5F 46 2D 31 2E 30 36 2D 30 2D 47 44 30 46 32 44 42 46
→ 36 45 35 90 00
```

Firmware Version

Used to retrieve the firmware version as two bytes in format `major.minor`, each interpreted as an unsigned integer.

Command	CLA	INS	P1	P2	LC	DATA	LE
FW Version	FF	70	04	6A	06	E0 04 DF 82 03 00	00

Example Exchange returns FW version 1.11:

```
→ FF 70 04 6A 06 E0 04 DF 82 03 00 00
← DF 82 03 02 01 0B 90 00
```

Firmware Source Reference

Used to retrieve the firmware source reference (SHA1 Hash) in hexadecimal and capped to 10 bytes. The first five bytes of the source reference are also part of the FW Label response.

Command	CLA	INS	P1	P2	LC	DATA	LE
FW Source Ref	FF	70	04	6A	06	E0 04 DF 82 05 00	00

Example Exchange returns FW source ref 03B2FC51AA1D61822FAC:

```
→ FF 70 04 6A 06 E0 04 DF 82 05 00 00
← DF 82 05 0A 03 B2 FC 51 AA 1D 61 82 2F AC 90 00
```

Get Debounce Configuration

Used to retrieve the currently set Debounce configuration.

Command	CLA	INS	P1	P2	LC	DATA	LE
Get Debounce Config	FF	70	04	6A	06	E0 04 DF 82 06 00	00

Example Exchange returns the current configuration values 010006:

```
→ FF 70 04 6A 06 E0 04 DF 82 06 00 00
← DF 82 06 03 01 00 06 90 00
```

Set Debounce Configuration

Used to apply a Debounce configuration. The configuration payload consists of three bytes:

- b2 corresponds to True Eager config value (default: 01)
- b1 corresponds to Defer Press config value (default: 00)
- b0 corresponds to Defer Release config value (default: 06)

The allowed range for each of these values is 0x00 - 0x0C.

Each of the three values is used to configure a specific state in the debounce process and the values correspond to multiples of the scan interval, which is set to 4ms.

Command	CLA	INS	P1	P2	LC	DATA	LE
Set Debounce Config	FF	70	04	6A	09	E1 07 DF 82 06 03 XX XX XX	00

Example Exchange returns only the status of the request:

```
→ FF 70 04 6A 09 E1 07 DF 82 06 03 01 02 03 00
← DF 82 06 00 90 00
```

Example of behavioral description when configured with default values

Initial state: Key is pressed.

With a Defer Release configuration of 06, the JK-A1150 will report a key as released only after it has not been pressed for 6 scan intervals. With a Defer Press configuration of 00, a key will be reported as pressed immediately when the key-press is detected. After this, the JK-A1150 ignores key state changes for 01 scan intervals (True Eager), before entering the key release detection again.

See the Diagram *Key Debouncing Algorithm* on the next page for more information.

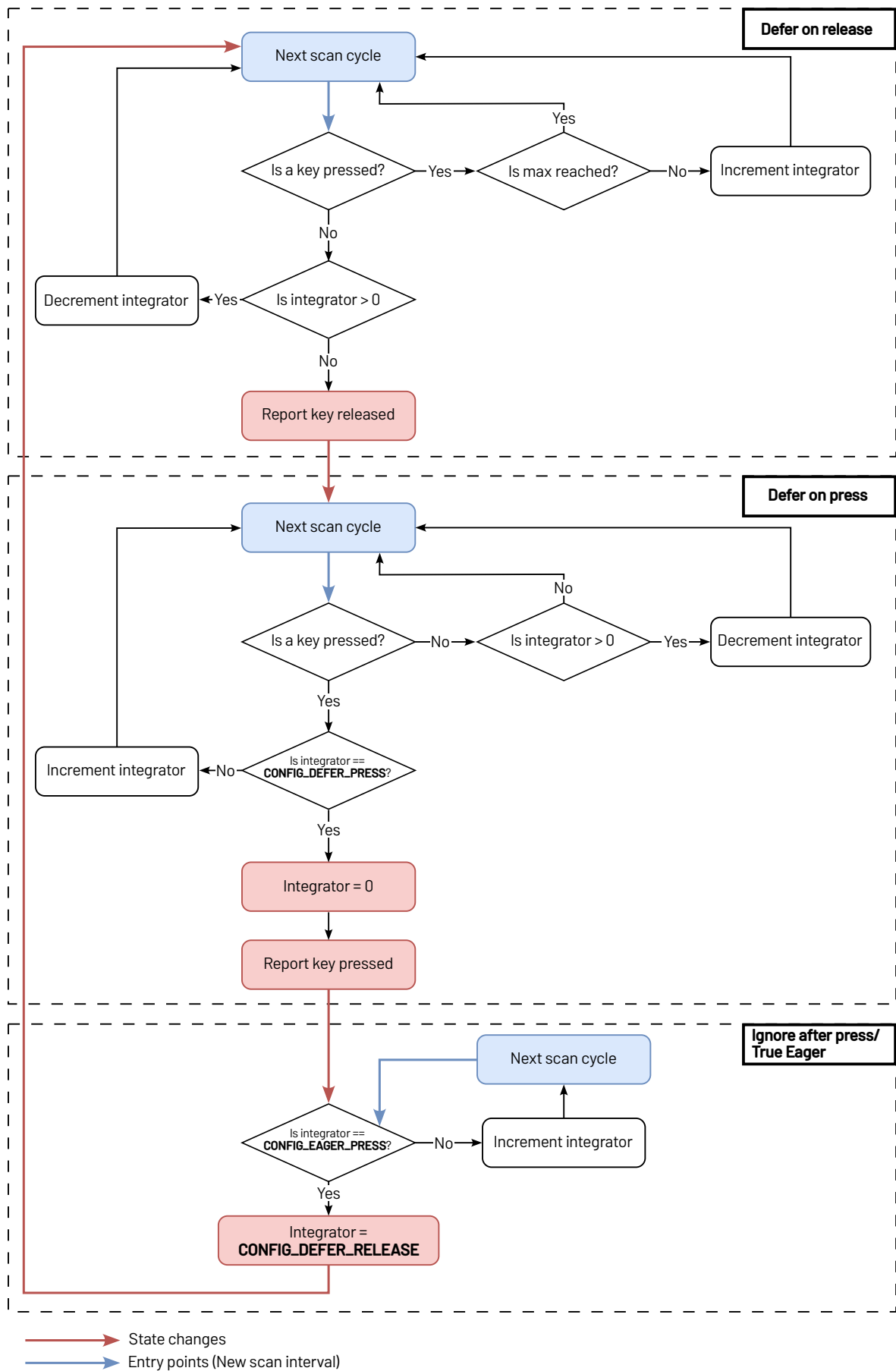


Fig. 4.1: Key Debouncing Algorithm

Bootloader Label

Used to retrieve the ASCII bootloader firmware label. The label consists of:

- Device name (JK-A1150)
- Label id (BL)
- Memory type ([F]lash / [R]om)
- FW version (major.minor-patch)
- FW source ref (first 5 bytes)

Command	CLA	INS	P1	P2	LC	DATA	LE
BL Label	FF	70	04	6A	06	E0 04 DF 84 02 00	00

Example Exchange returns JK-A1150_BL_F-1.06-0-GD0F2DBF6E5:

```
→ FF 70 04 6A 06 E0 04 DF 84 02 00 00
← DF 84 02 20 4A 4B 2D 41 31 31 35 30 5F 42 4C 5F 46 2D 31 2E 30 36 2D 30 2D 47 44 30 46 32 44 42 46
→ 36 45 35 90 00
```

Bootloader Version

Used to retrieve the bootloader firmware version as two hexadecimal bytes in format `major.minor`.

Command	CLA	INS	P1	P2	LC	DATA	LE
BL Version	FF	70	04	6A	06	E0 04 DF 84 03 00	00

Example Exchange returns bootloader FW version 1.04:

```
→ FF 70 04 6A 06 E0 04 DF 84 03 00 00
← DF 84 03 02 01 04 90 00
```

Bootloader Source Reference

Used to retrieve the bootloader firmware source reference (SHA1 Hash) in hexadecimal and capped to 10 bytes. The first five bytes of the source reference are also part of the BL Label response.

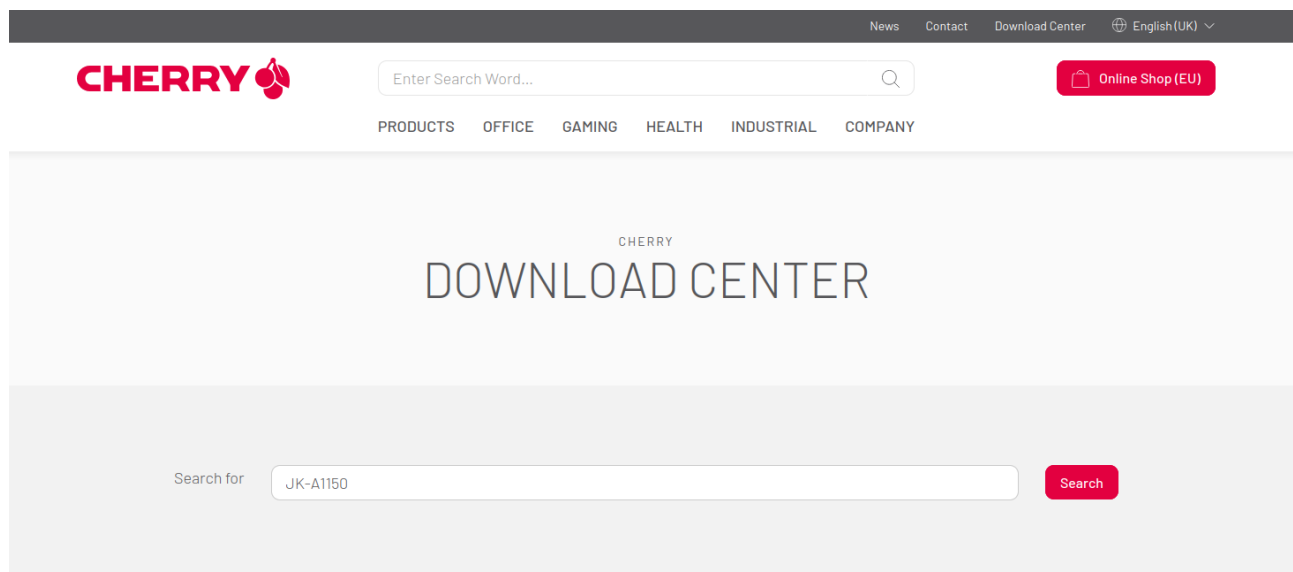
Command	CLA	INS	P1	P2	LC	DATA	LE
BL Source Ref	FF	70	04	6A	06	E0 04 DF 84 05 00	00

Example Exchange returns bootloader FW source ref 03B2FC51AA1D61822FAC:

```
→ FF 70 04 6A 06 E0 04 DF 84 05 00 00
← DF 84 05 0A 03 B2 FC 51 AA 1D 61 82 2F AC 90 00
```

4.2 Firmware Update

The JK-A1150 supports firmware updates via USB. Only officially signed firmware updates are accepted by the device. Please check the CHERRY Download Center to see if there is new firmware available for your JK-A1150.



In case new firmware is available, please download the ZIP file and unpack it to your hard drive.

4.2.1 Update via the Cherry Management Software

On Windows, the JK-A1150 can be updated using the CHERRY Management SW, which is provided alongside the firmware update in the downloaded ZIP file.

To start the update, start the EXE file and follow the instructions.

4.2.2 Update via the Command Line Interface

On systems supported by libusb, the JK-A1150 can be updated via the command-line tool *dfu-util*.

DFU stands for Device Firmware Upgrade and provides a standardized way to update the firmware of USB devices. *dfu-util* can be installed via the package manager on most Linux distributions, or via homebrew on macOS. For more information as well as Windows binaries, see the *dfu-util* homepage.

To update the JK-A1150, open a command shell and call

```
# dfu-util -D /path/to/firmware-update-file
```

Make sure the device is not unplugged during the update. A loss of power at this point can lead to failures. The JK-A1150 will automatically perform a reboot after the firmware update is completed successfully.

5 Appendix

5.1 Enable CCID Escape Commands

5.1.1 Windows

To send an Escape command to a reader, the DWORD registry value `EscapeCommandEnable` must be added and set to a non-zero value under one of the following keys.

- `HKLM\SYSTEM\CurrentControlSet\Enum\USB\VID_046A&PID_0201&MI_02\*\Device Parameters` (prior to Windows 7)
- `HKLM\SYSTEM\CurrentControlSet\Enum\USB\VID_046A&PID_0201&MI_02\*\Device Parameters\WUDFUsbccidDriver` (Windows 7 and later)

Note that on Windows 7 and later, the key `WUDFUsbccidDriver` must already exist. Do not create it yourself. If a reader with a different serial number is connected to the computer, the operation must be repeated.

For more information see: <http://msdn.microsoft.com/en-us/windows/hardware/gg487509.aspx>

5.1.2 macOS

macOS has built-in CCID support starting from macOS Sonoma. With Sonoma version 14.3, the built-in CCID driver reports support for CCID Escape, but commands may not be executed as expected. In case of issues, it is possible to switch to the Linux CCID driver, as was the norm prior to macOS Sonoma.

To see which driver is currently in use, enter the following command into a shell:

```
defaults read /Library/Preferences/com.apple.security.smartcard.plist useIFDCCID
```

If the `useIFDCCID` key exists and is set to 1, then the Linux CCID driver is already active. Refer to *Linux* to enable CCID Escape commands.

Otherwise, it is easily possible to switch to the Linux CCID driver, by calling:

```
sudo defaults write /Library/Preferences/com.apple.security.smartcard useIFDCCID -bool yes
```

For more information, see: <https://blog.apdu.fr/posts/2023/11/apple-own-ccid-driver-in-sonoma/> and <https://blog.apdu.fr/posts/2024/02/macos-sonoma-bug-scardcontrol-part-2/>

5.1.3 Linux

This section also applies to macOS if the Linux CCID driver is used.

The Linux CCID driver supports the CCID Escape command since revision 5991, though the command is not enabled per default. To allow the use of CCID Escape, modify the value of `ifdDriverOptions` in the driver's `Info.plist` accordingly.

If the bit 0 of `ifdDriverOptions` is set, then the driver will allow the use of CCID Escape:

```
[ ... ]
<key>ifdDriverOptions</key>
<string>0x0001</string>

←— Possible values for ifdDriverOptions
1: DRIVER_OPTION_CCID_EXCHANGE_AUTHORIZED
   the CCID Exchange command is allowed. You can use it through
   SCardControl(hCard, IOCTL_SMARTCARD_VENDOR_IFD_EXCHANGE, ...)
[ ... ]
```

For more information see: <https://blog.apdu.fr/posts/2011/10/featureccidesccommand/>

6 Contact

Europe

Cherry Europe GmbH
Cherrystraße 2
91275 Auerbach/Opf.
Germany

Web: <https://www.cherry.de/en-gb>
Email: info@cherry.de

USA

Cherry Americas LLC
9522 58th Place
Suite 100
Kenosha, WI 53144
United States

Phone: +1 262 942 6508
Web: <https://www.cherryamericas.com>
Email: sales@cherryamericas.com

7 Revision History

Date	Revision	Major changes
Sep, 2025	v1.0.1	Initial release
Sep, 2025	v1.0.2	Update supported Vendor Command
Oct, 2025	v1.0.3	Update firmware version description
Dec, 2025	v1.0.4	Update Cherry Americas address