



APP and APC specification 1.03

Allen Coding Printers

DETAIL SPECIFICATION

Table of Contents

Table of Contents	2
Important Notes.....	5
1 Introductory notes.....	6
1.1 Intended audience	6
1.2 Applicability.....	6
2 General Information	7
2.1 Used pictograms	7
2.2 Conventions used in this document.....	7
2.3 Glossary of terms	8
2.4 Companion Documents.....	8
3 Protocol Description	9
3.1 Rationale	9
3.2 General Definitions	9
3.3 Application Printer Protocol (APP)	9
3.3.1 APP header fields	10
3.3.2 APP details	10
3.3.3 APP header minimal C struct definition	11
3.4 Application Printer Commands (APC)	12
3.4.1 General Definitions	12
3.4.2 General COMMAND message payload definition	13
3.4.3 RESPONSE message definition	13
3.4.4 NOTIFICATION message definition.....	13
3.4.5 STREAM message definition	13
4 APC commands and responses.....	15
4.1 General information & device status inquiry	15
4.1.1 ECHO	15
4.1.2 VERSION_INFORMATION	15
4.1.2.1 GET_VERSION_INFORMATION.....	15
4.1.2.5 SET_VERSION_INFORMATION	16
4.1.3 SOFT_REBOOT.....	16
4.1.4 RETRIEVE_DIAGNOSTIC_LOG	17
4.1.5 GET_DEVICE_STATUS.....	17
4.1.6 GET_PENDING_ERRORS	18
4.1.7 GET_PRINT_INFO	19
4.1.8 RAISE_NOTIFICATION.....	19
4.1.9 DEVICE_INFO_GET	19
4.1.10 DEVICE_INFO_LISTKEYS.....	20
4.1.11 DEVICE_INFO_VALUETYPE	21
4.1.12 RESET_ERRORS.....	21
4.1.13 RIBBON_CLEAR.....	22
4.1.14 RESET_RIBBON_COUNTER	22
4.1.15 RESET_SERIALIZATION	22
4.2 Print operations	23
4.2.1 PRINT_TRIGGER.....	23
4.2.2 STOP	23
4.2.3 LABEL_LOAD.....	24
4.2.4 LABEL_POSTVIEW.....	25
4.3 Device configuration	25
4.3.1 Configuration key-value pair definition	25
4.3.2 CONFIG_SET	32

4.3.3	CONFIG_GET	32
4.3.4	CONFIG_LISTKEYS	33
4.3.5	CONFIG_VALUETYPE	34
4.4	File and print label storage operations	35
4.4.1	FILE_STORAGE_PUT	35
4.4.2	FILE_STORAGE_DELETE	36
4.4.3	FILE_STORAGE_LIST	36
4.4.4	FILE_STORAGE_FREE	37
4.4.5	FILE_STORAGE_GET	37
4.4.6	LIST_LABELS	38
4.4.7	LABEL_EXISTS	38
4.5	Variable contents	39
4.5.1	Printer variable type definition	39
4.5.2	VARIABLE_SET	39
4.5.3	VARIABLE_GET	40
4.5.4	VARIABLE_DELETE	40
4.5.5	VARIABLE_LISTKEYS	41
4.6	Font	41
4.6.1	FONT_STORAGE_PUT	41
4.6.2	FONT_STORAGE_DELETE	43
4.6.3	FONT_STORAGE_LIST	43
4.6.4	FONT_STORAGE_GET	44
4.6.5	LIST_FONTS	44
4.6.6	FONT_EXISTS	45
4.6.7	FONT_UPDATE	45
4.6.8	FONT_LOAD	46
4.6.9	FONT_UPDATE_SELECTSLOT	47
4.7	PIC	48
4.7.1	PIC_UPDATESTART	48
4.7.2	PIC_UPDATEPKT	48
4.7.3	PIC_UPDATEFINALIZE	49
4.8	FPGA	49
4.8.1	FPGA_UPDATE	49
4.9	DIAGNOSTICS FUNCTIONS	50
4.9.1	DIAGNOSTICS	50
APC notifications		51
4.10	List of defined NOTIFICATION messages	51
4.10.1	NOTIFY_INIT	51
4.10.2	NOTIFY_HELLO	51
4.10.3	NOTIFY_READY	51
4.10.4	NOTIFY_IDLE	51
4.10.5	NOTIFY_ERROR	52
4.10.6	NOTIFY_WARN	53
4.10.7	NOTIFY_INFO	53
4.10.8	NOTIFY_USERMESSAGE	54
4.10.9	NOTIFY_PRINTED	54
4.10.10	NOTIFY_PRINT_LOST	54
4.10.11	NOTIFY_NEED_VARIABLE_DATA	54
4.10.12	NOTIFY_CASECOVER_CLOSE	55
Functional Part – More parts		56

5	Appendix A – Identifier enumerations.....	57
6	Appendix B – Value type definitions.....	59
7	Error codes	60
8	Command IDs	64
9	Enumerations	65
10	Label definition example.....	67

Important Notes

The content of this document is provided for information purposes only. It is subject to change without notice, and comes without guarantee of correctness or completeness, or fitness for any particular purpose.

This publication or any parts thereof may not be made available in any form to third parties without prior written permission from the legal owner.

Contact Information

Allen Coding GmbH
Friedrich-Bergius-Ring 30
D-97076 Würzburg
Direct +49 931 250 76 - 0
Fax +49 931 250 76 – 50
Email: info@allen.de
Website: www.allen.de

Version Information

This Version: 0.01	Released at: unreleased	Released by:	{EmployeeName, eMail}
---------------------------	--------------------------------	---------------------	-----------------------

Revision History

Version 0.01	Released at: unreleased	Released by:	

1 Introductory notes

1.1 Intended audience

This document is intended for software developers and application engineers that need to communicate with an APC printer through software, or require a reference for the capabilities of the APC communication interface.





It is not intended for a general audience.

1.2 Applicability


The contents of this document applies to the ongoing development work on the “TP-Upgrade” (Codename Anton) printer, and should be applicable in modified form for future XLU-board based software updates (e.g. XL series.).

2 General Information

2.1 Used pictograms

Pictogram	Meaning
	Warning of a general hazard.
	Warning of possible damage.
	Observe the corresponding information in the referenced companion document.
	General note.

This is an example:

CAUTION	
	<p>Potential hazard caused by incorrect firmware files!</p> <p>The use of firmware images not suitable for the device may result in an inoperative (“bricked”) device.</p> <ul style="list-style-type: none">▪ Verify and check all available information before applying a firmware update.

2.2 Conventions used in this document

Code examples and formatted data

```
#include <stdio.h>

int main(void)
{
    printf("Hello, World\n");

    return 0;
}
```

Non-printable characters

All non-printable characters are indicated in 0x-prefixed hexadecimal notation: 0xCAFEBAFE

Definitions

For syntactical definitions, a variant of the Extended Backus-Naur Form is used. Ref. http://en.wikipedia.org/wiki/Extended_Backus%E2%80%93Naur_Form

Specific Conventions

The following identifiers are used to distinguish the on-the-wire data formats:

- Value types are identified by their C name as in `stdint.h`
- Zero-terminated character strings are identified by the word “cstring”
- A 32 bit value is divorced like that: 0xMB_UB_HB_LB
- A 16 bit value is divorced like that: 0xHB_LB
- (for example (int32)1 is 00 00 00 01)
-

2.3 Glossary of terms

Recursion

Synonymous with -> Self-reference

Self-reference

Synonymous with -> Recursion

2.4 Companion Documents

Document Name and Version	Available through

3 Protocol Description

3.1 Rationale

Before individual printer commands can be defined, we must first define a communication protocol that can be used to safely and efficiently exchange commands and notifications over the different communication channels the printer supports.

It is herein introduced with the name “Allen Printer Protocol”, abbreviated as APP.

Individual printer commands are defined as structured payload data for the APP. The corresponding command set is herein introduced with the name “Allen Printer Commands”, abbreviated as APC.

3.2 General Definitions

All communication channels - the dedicated handheld link, the general purpose serial link, the TCP connections and the serial over USB link - are treated equally. No syntactic or semantic differences are permitted to exist based on which communication channel is used for communication with the printer.

A timeout occurs when no data is received within 1500ms while reading a packet. The partial packet is discarded and an error is signaled as a NOTIFICATION message through the APC layer.

3.3 Application Printer Protocol (APP)

The communications protocol is a fairly trivial protocol similar to many existing protocols that must fulfill similar demands.

A packet is defined by a fixed-length header with a magic start byte, a flags byte, and a two bytes long payload length field for a total of four bytes. The maximum payload length is defined to be 32768 bytes as a compromise between message transfer time, round-trip efficiency, the normal data size of a single exchange and memory limitations within the printer. The data format on the line is defined to be standard “network byte order”, big-endian.

Not all communication channels that are supported by the printer guarantee completeness and correctness of transferred data. Therefore, APP defines a per-packet optional CRC23 checksum mechanism.

Some of the data transferred over APP are very efficiently compressed with a simple algorithm, while some of the communications channels are limited to 115200 bps. Therefore, APP defines a per-packet optional compression algorithm.

APP defines two header flags, NOTIFICATION and RESPONSE, which are set only by the printer accordingly with the type of payload delivered. This is done purely for convenience and by convention; the protocol can in theory operate without requiring these flags. The assumption is that observing these flags may provide useful and easily available sorting and routing information to external applications.

3.3.1 APP header fields

Offset	Name	Summary
Byte 0	MAGIC	Magic number 0x80 + protocol version number (current: 0)
Byte 1	FLAGS	Message flags
Byte 1: Bit 0	COMPRESSED	Packet payload is compressed – NYI
Byte 1: Bit 1	CHECKSUMMED	Packet payload is CRC32 checksummed, CRC32 follows after length bytes – NYI
Byte 1: Bit 2	NOTIFICATION	Packet contains a notification message. Convenience flag for message processing.
Byte 1: Bit 3	RESPONSE	Packet contains a response to a previously issued command. Convenience flag for message processing.
Byte 1: Bit 4	RESERVED_4	This flag is used internally by the printer and must always be 0.
Byte 1: Bit 5	RESERVED_5	Reserved for future use. Must always be 0.
Byte 1: Bit 6	FLOW_CONTROL_ESCAPE	Packet data is escaped to support RS232 XOnXOff. – NYI
Byte 1: Bit 7	RESERVED_7	Reserved for future use. Must always be 0.
Byte 2	LENGTH_HIGH	High byte of payload length field.
Byte 3	LENGTH_LOW	Low byte of payload length field.

3.3.2 APP details

3.3.2.1 Protocol versioning

The APP version number is a single sequential number encoded in all packet headers as MAGIC – 0x80.

The current protocol version is 0.

Future protocol versions may:

- Define currently reserved flags for use.
- Implement currently specified but not yet implemented features.
- Extend the header beyond the length field.
- Differ in payload data definitions.

Any changes in the APP and APC specification must cohere with a uniquely defined version number. Changes in the specification are documented in accordance with the principle that all protocol versions remain well-defined by this specification document.

3.3.2.2 COMPRESSED header flag

Compressed messages are comprised of

1. A single byte indicating the compression algorithm used as part of an extended header.
2. The compressed payload data following.

NOTE: The compression algorithm byte being part of an extended header implies that it does not count towards the payload length as defined in the LENGTH fields.

NOTE: Compression support is not specified in detail and not supported in protocol version 0.

3.3.2.3 CHECKSUMMED header flag

Checksummed messages are comprised of

1. The 4 bytes of the 32-bit CRC following the length bytes as part of an extended header.

2. The checksummed payload following the CRC32.

NOTE: The CRC32 being part of an extended header implies that it does not count towards the payload length as defined in the LENGTH fields.

A standard CRC32 algorithm with a polynomial of 0x04C11DB7, Preset-to-all-ones and Post-invert is applied to the entire packet including the header with the 4 byte CRC part itself masked to 0.

NOTE: CRC32 support is not specified in detail and not supported in protocol version 0.

3.3.2.4 FLOW_CONTROL_ESCAPE header flag

To be able to transfer data transparently through software flow control serial links, the message data may be escaped at the transfer level.

Starting with the message length header fields, the following escapes for XOn and XOff must be used if the FLOW_CONTROL_ESCAPE flag is set. The Magic byte and the flags byte itself are specified so that – if active – the XOn and XOff characters cannot occur.

XOn := 0x20, 0x31
 XOff := 0x20, 0x33
 0x20 := 0x20, 0x00

Unescaping can be accomplished by a simple XOR operation with 0x20 on the byte received immediately after removing any received 0x20 bytes.

NOTE: FLOW_CONTROL_ESCAPE support is not supported in protocol version 0.

3.3.2.5 NOTIFICATION header flag

The printer guarantees that any messages sent that contain a “Notification Message” as defined by the APC specification have this header flag set to 1.

The purpose of this flag is purely for the sake of convenience of any external applications that may use this flag reliably to route or sort incoming packets before inspecting the payload in detail. It does not convey any information that could not also reliably be established by inspecting the payload.

This flag has no meaning in messages sent to the printer.

3.3.2.6 RESPONSE header flag

The printer guarantees that any messages sent that contain a “Response to previously issued command” as defined by the APC specification have this header flag set to 1.

The purpose of this flag is purely for the sake of convenience of any external applications that may use this flag reliably to route or sort incoming packets before inspecting the payload in detail. It does not convey any information that could not also reliably be established by inspecting the payload.

This flag has no meaning in messages sent to the printer.

3.3.3 APP header minimal C struct definition

```
struct st_pkt {
    uint8_t magic; // magic byte 0x80 + incremental protocol version number
    (current version: 0)
    union {
        uint8_t flags_uchar; // message flags
    }
};
```

APP and APC specification 1.03

```

    struct {
        uint8_t reserved_7 :1; // reserved
        uint8_t flow_control_escape :1; // RS232 software flow control
escaping - NYI
        uint8_t reserved_5 :1; // reserved
        uint8_t reserved_4 :1; // printer internal
        uint8_t response :1; // packet is a response to an issued
command
        uint8_t notification :1; // packet is an unsolicited
notification message
        uint8_t checksummed :1; // packet is checksummed - NYI
        uint8_t compressed :1; // packet is compressed - NYI
    } flags_fields;
} flags;
union {
    uint16_t length_ushort;
    struct {
        uint8_t length_high; // high byte
        uint8_t length_low; // low byte
    } length_bytes;
} length;
};

```

3.4 Application Printer Commands (APC)

3.4.1 General Definitions

The APC defines four types of messages:

COMMAND messages trigger an action in the printer. They are always directed at the printer.

RESPONSE messages are always direct replies to an issued **COMMAND** and are only sent to the communication channel that the corresponding **COMMAND** was received from. The printer guarantees that a **RESPONSE** is issued within 150ms of receiving any **COMMAND**.

NOTIFICATION messages always originate from the printer and are sent to all channels without expecting a response.

STREAM messages utilize a file-descriptor like printer-issued token and are used to transfer data streams to and from the printer that would be too large to fit into a **COMMAND** payload.

NOTE: The stream mechanism is not finally specified and not supported in protocol version 0, and may be omitted following review.

COMMAND and **RESPONSE** messages utilize a **COMMAND_SEQUENCE_ID** handle mechanism.

An unhandled or unknown **COMMAND** causes an immediate error response.

The printer may send a **NOTIFICATION** message at any time, also while a **COMMAND** on the channel is pending a **RESPONSE**. All **NOTIFICATION** messages are marked with a header flag so that they may be filtered without payload inspection by the receiving component.

Every **COMMAND** will return a corresponding **RESPONSE** message as soon as it is processed. All **RESPONSE** messages are marked with a header flag so that they may be filtered without payload inspection by the receiving component.

Every **COMMAND** takes effect immediately and returns a **RESPONSE** message immediately after taking effect, unless the **COMMAND** specifically uses one of the following two mechanisms:

In cases where a COMMAND may not take effect immediately – e.g. changing the print speed while an intermittent mode print cycle is currently underway - the printer may send an immediate RESPONSE confirming the operation under the guarantee that the COMMAND will become effective at the next possible opportunity – e.g. the print speed is changed as soon as the print cycle completes. Such cases are documented separately.

In cases where a COMMAND may run for a longer time, - e.g. a task is performed in the printer as a background process – the printer will send an immediate RESPONSE confirming that the corresponding process has been initiated successfully, and upon completion or abortion of the task later send a specifically defined NOTIFICATION message. Such cases are documented separately.

All payloads start with a message type identification byte.

3.4.2 General COMMAND message payload definition

Offset	Name	Summary
Byte 0	MESSAGE_TYPE	The value 0x10 indicates a COMMAND message.
Bytes 1 – 2	COMMAND_ID	A 16 bit constant identifying the COMMAND as defined in detail below.
Bytes 3 – 4	COMMAND_SEQUENCE_ID	A 16 bit value assigned by the user, copied to the RESPONSE packet. See RESPONSE message definition.
Any following	DATA	Optional additional fixed or dynamic length data as defined in detail for the individual COMMAND.

3.4.3 RESPONSE message definition

Indication of a successful operation or an error condition

Offset	Name	Summary
Byte 0	MESSAGE_TYPE	The value 0x30 indicates a RESPONSE message.
Bytes 1 – 2	COMMAND_ID	The COMMAND_ID that this RESPONSE is issued for is repeated.
Bytes 3 – 4	COMMAND_SEQUENCE_ID	Copied from the COMMAND message this response is generated for.
Bytes 5 – 6	COMMAND_ERROR	16 bit error code constant as identified for the individual COMMAND. Zero to indicate success.
Any following	DATA	In case of success: Optional additional fixed or dynamic length data as defined in detail for the individual COMMAND. In case of error: Usually, it would be a zero-terminated string offering a concise and precise error description, but may include additional structured data if specified in detail for the individual COMMAND

3.4.4 NOTIFICATION message definition

Offset	Name	Summary
Byte 0	MESSAGE_TYPE	The value 0x40 indicates a NOTIFICATION message.
Bytes 1 – 2	NOTIFICATION_ID	A 16 bit constant identifying the NOTIFICATION as defined in detail below.
Any following	DATA	Optional additional fixed or dynamic length data as defined in detail for the individual NOTIFICATION.

3.4.5 STREAM message definition

Offset	Name	Summary
Byte 0	MESSAGE_TYPE	The value 0x50 indicates a STREAM message.
Byte 1-2	COMMAND_ID	Command ID of the STREAM message

Byte 3-4	COMMAND_SEQUENCE_ID	Identifier of the command message (16 bit value)
Byte 5-6	STREAM_SEQUENCE_ID	A valid stream token previously issued by the printer.
Bytes 7 – 10	STREAM_LENGTH	Total length of the streamed data – unsigned 32 bit integer.
Bytes 11 – 14	STREAM_OFFSET	Offset of the contained data within the stream – unsigned 32 bit integer.
Any following	DATA	Streamed data block contents.

4 APC commands and responses

4.1 General information & device status inquiry

4.1.1 ECHO

4.1.1.1 COMMAND definition

Return the payload data in an echo response, supporting a ping-like functionality if sequence numbers are injected into the payload.

COMMAND_ID: 0x0001

DATA: Any.

4.1.1.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 data	ANY	The COMMAND payload is returned in unmodified form.

4.1.1.3 Negative RESPONSE definition

None defined.

4.1.2 VERSION_INFORMATION

4.1.2.1 GET_VERSION_INFORMATION

4.1.2.2 COMMAND definition

Return printer firmware version information.

COMMAND_ID: 0x0010

DATA: None.

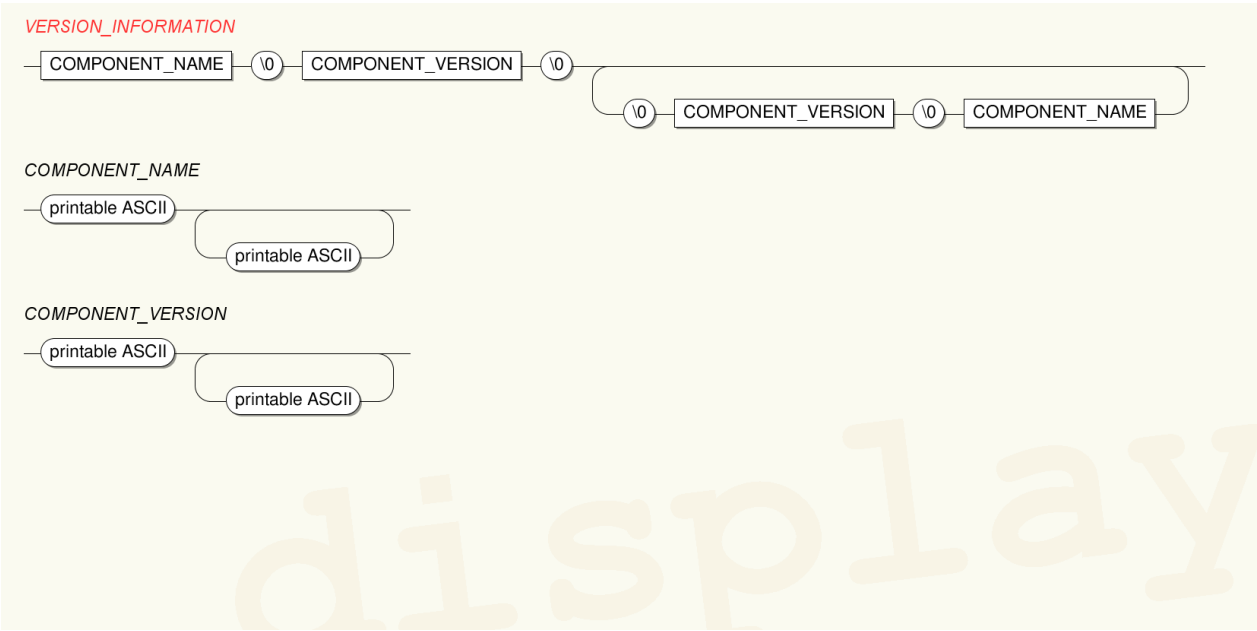
4.1.2.3 Positive RESPONSE definition

Index & Type	Name	Summary
0 cstring	VERSION_INFORMATION	A set of pairs of zero-terminated printable ASCII strings providing version information about all firmware components.

VERSION_INFORMATION = COMPONENT_NAME, COMPONENT_VERSION, { COMPONENT_NAME, COMPONENT_VERSION }

COMPONENT_NAME = ? zero-terminated printable ASCII string ?

COMPONENT_VERSION = ? zero-terminated printable ASCII string ?



4.1.2.4 Negative RESPONSE definition

None defined.

4.1.2.5 SET_VERSION_INFORMATION

4.1.2.6 COMMAND definition

Set the name for the current printer firmware package. i.e. MLI-1.1

COMMAND_ID: 0x0011

DATA: version name.

Index & Type	Name	Summary
0 cstring	VERSION_INFORMATION	Printable zero-terminated ASCII string specifying the version information. No longer than 18 characters.

4.1.2.7 Positive RESPONSE definition

Default response with no data part.

4.1.3 SOFT_REBOOT

4.1.3.1 COMMAND definition

Perform a soft reboot of all printer components.
Execution of this command is deferred with a delay of 500ms.

This command is only accepted in the IDLE state or while the printer is still initializing. The printer can't be safely reset while in the READY or IDLE states.

COMMAND_ID: 0xABBA

DATA: None.

4.1.3.2 Positive RESPONSE definition

Default response with no data part.

4.1.3.3 Negative RESPONSE definition

Error Code	Name	Summary
	REFUSED	The printer will not soft reboot at this time. Wait for the printer to reach a different state first.

4.1.4 RETRIEVE_DIAGNOSTIC_LOG

4.1.4.1 COMMAND definition

Pull the diagnostic log from the specified MCU core and retrieve it in a single data packet.

COMMAND_ID: 0x0020

DATA: MCU Identifier.

Index & Type	Name	Summary
0 uint8	MCU_IDENTIFIER	Identifies the MCU to pull the diagnostic log from. 0 = Routing MCU 1 = Print control MCU 2 – 14 = TPH Management MCUs 1-12

4.1.4.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 cstring	LOG_STRING	A single zero-terminated printable ASCII string up to the maximum packet size.

LOG_STRING = ? zero-terminated printable ASCII string ?

4.1.4.3 Negative RESPONSE definition

Error Code	Name	Summary
	NOT_AVAILABLE	The specified MCU cannot deliver a diagnostic log for any reason.

4.1.5 GET_DEVICE_STATUS

4.1.5.1 COMMAND definition

Return printer status information.

COMMAND_ID: 0x0030

DATA: None.

4.1.5.2 Positive RESPONSE definition

Index & Type	Name	Summary

0 uint8	STATE	ID of the state 0 - uninitialized 1 - initialized 2 - idle 4 - ready 8 - printing
-----------	-------	--

4.1.5.3 Negative RESPONSE definition

None defined.

4.1.6 GET_PENDING_ERRORS

4.1.6.1 COMMAND definition

Return a list of current error conditions on the device.

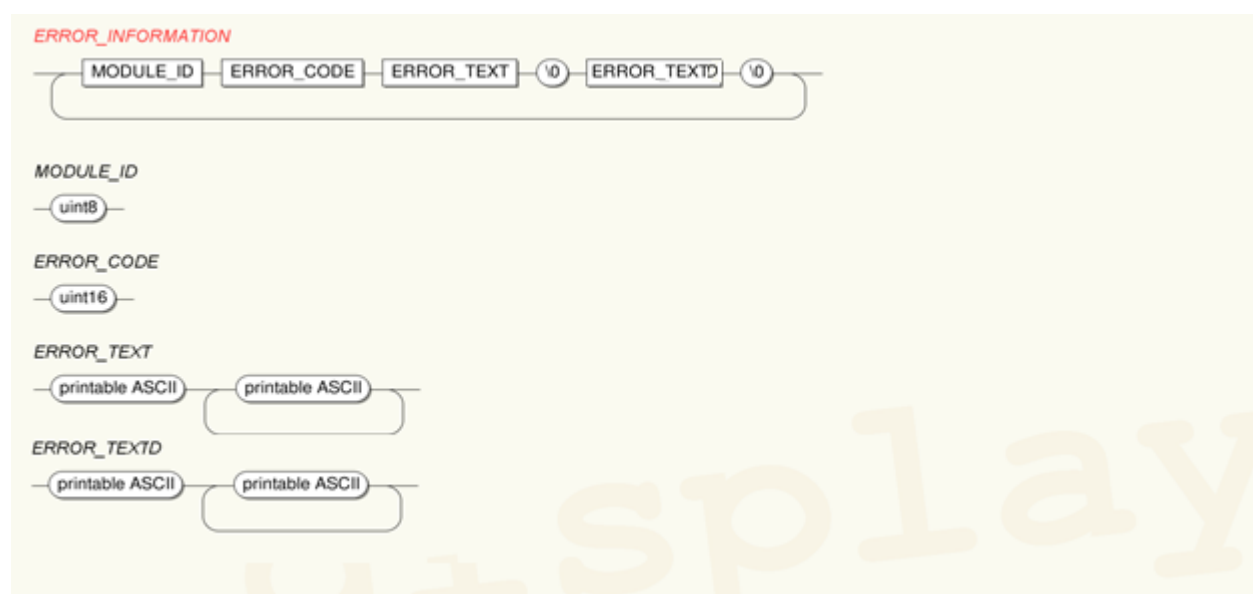
COMMAND_ID: 0x0040

DATA: None.

4.1.6.2 Positive RESPONSE definition

Any number of ERROR_INFORMATION blocks, or no data if no errors are pending.

Index & Type	Name	Summary
0 uint8	MODULE_ID	ID of the module the error occurred in. 0 - Renesas Bootloader 1 - Renesas User Application 2 - Freescale Bootloader 3 - Freescale User Application TBD
1 uint16	ERROR_CODE	Unique error code within the module. Error codes may be re-assigned with different meaning in different modules.
2 cstring	ERROR_TEXT	A printable ASCII string with error information.
3 cstring	ERROR_DETAILS	A printable ASCII string with error detail information.



4.1.6.3 Negative RESPONSE definition

None defined.

4.1.7 GET_PRINT_INFO

4.1.7.1 COMMAND definition

Return information about the printing process currently underway.

COMMAND_ID: 0x0050

DATA: TPH number, starting from 0

4.1.7.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 uint16	PRINTS_PERFORMED	Number of print cycles successfully completed since the last time the PRINTS_TO_PERFORM value was set.
1 uint16	PRINTS_TO_PERFORM	Preset number of print cycles after which the printer automatically enters the STOPPED state. May be 0 if no limit is desired.
2 cstring	LABEL_NAME	A single zero-terminated UTF-8 string containing the active label name for the head.

4.1.7.3 Negative RESPONSE definition

Error Code	Name	Summary
	NOT_READY	The device is not in a READY state and therefore cannot infer information about printing.

4.1.8 RAISE_NOTIFICATION

4.1.8.1 COMMAND definition

Raises a notification message of type NOTIFY_INFO to be dispatched to all channels.

COMMAND_ID: 0x1000

DATA: NOTIFY_INFO code & data fields to be raised.

Index & Type	Name	Summary
0 uint16	INFO_CODE	An information code as defined by the NOTIFY_INFO message.
1 as specified	INFO_DATA	Optional data block individually defined by the information message definition.

4.1.8.2 Positive RESPONSE definition

Default response with no data part.

4.1.9 DEVICE_INFO_GET

4.1.9.1 COMMAND definition

Get a configuration value.

COMMAND_ID: 0x0032

DATA: Configuration key.

Index & Type	Name	Summary
0 cstring	DEVICE_INFO_KEY	Printable zero-terminated ASCII string specifying the device info key. No longer than 18 characters.

4.1.9.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 as specified	DEVICE_INFO_VALUE	As specified below

4.1.9.3 Device info key-value pair definition

Name	Summary	Unit	Type	idle only	Debug
TPH_UNITS	Bitmask: Information about which pic has a tph connected.	-	uint32	0	0
TEMPERATURE	Detected by print head	°C	VECTOR<int16>*12	1	0
TINRANGE	for debug issues and for internal use only		VECTOR<bool(8bit)>	1	1
MAINLOOPCNT	for debug issues and for internal use only		VECTOR<uint32>	1	1
HYSTERESISBEHAVIOR	for debug issues and for internal use only		Uint8	0	1
MEAS- URED_CARRIAGE_DISTANCE	Max available carriage distance measured at boot up		Uint32<2400>	0	1
TPH_TYPE	Info about the installed print head		uint8	0	0
MACHINE_PRINTS_TOTAL	Total prints done		uint32	0	0
CURRENT_PRESSURE	Current air pressure	0.1 bar	uint32	0	0

4.1.10 DEVICE_INFO_LISTKEYS

4.1.10.1 COMMAND definition

Retrieve a list of all configuration keys known and supported.

COMMAND_ID: 0x0033

DATA: None.

4.1.10.2 Positive RESPONSE definition

Index & Type	Name	Summary
Repeats cstring	DEVICE_INFO_KEY	Printable zero-terminated ASCII string specifying the device info key. No longer than 18 characters.

4.1.11 DEVICE_INFO_VALUETYPE

4.1.11.1 COMMAND definition

Get value type information for a configuration key.

COMMAND_ID: 0x0034

DATA: Configuration key.

Index & Type	Name	Summary
0 cstring	DEVICE_INFO_KEY	Printable zero-terminated ASCII string specifying the configuration key. No longer than 18 characters.

4.1.11.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 Boolean (uint8)	IDLE_ONLY	Value can only be read when machine is not printing
1 Boolean (uint8)	DEBUG	Device info key is used for internal use only
2 cstring	UNIT	Unit of measurement, printable zero-terminated ASCII string as specified below.
3 cstring	TYPE_RANGE	Value type, printable zero-terminated ASCII string as specified in Appendix B

4.1.12 RESET_ERRORS

4.1.12.1 COMMAND definition

Reset pending errors

COMMAND_ID: 0x0041

DATA: None.

4.1.12.2 Positive RESPONSE definition

Default response with no data part

4.1.12.3 Negative RESPONSE definition

Index & Type	Name	Summary
	NOT_READY	The device is not in a READY state and therefore error cannot be reset.

4.1.13 RIBBON_CLEAR

4.1.13.1 COMMAND definition

Reset pending errors

COMMAND_ID: 0x0045

DATA: None.

4.1.13.2 Positive RESPONSE definition

Default response with no data part

4.1.13.3 Negative RESPONSE definition

Index & Type	Name	Summary
	NOT_READY	The device is not in a READY state and therefore error cannot be reset.

4.1.14 RESET_RIBBON_COUNTER

4.1.14.1 COMMAND definition

Reset pending errors

COMMAND_ID: 0x0046

DATA: None.

4.1.14.2 Positive RESPONSE definition

Default response with no data part

4.1.14.3 Negative RESPONSE definition

Index & Type	Name	Summary
	NOT_READY	The device is not in a READY state and therefore error cannot be reset.

4.1.15 RESET_SERIALIZATION

4.1.15.1 COMMAND definition

Reset serialization elements

COMMAND_ID: 0x0047

DATA: None.

4.1.15.2 Positive RESPONSE definition

Default response with no data part

4.1.15.3 Negative RESPONSE definition

Index & Type	Name	Summary
	NOT_READY	The device is not in a READY state and therefore error cannot be reset.

4.2 Print operations

4.2.1 PRINT_TRIGGER

4.2.1.1 COMMAND definition

Perform the same action as if the external print signal was triggered.

COMMAND_ID: 0x0080

DATA: None.

4.2.1.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 uint16	PRINTS_PERFORMED	Number of already triggered print cycles since entering the ready state. NOTE: Does not and cannot include the print cycle just triggered, which may still abort.
1 uint16	PRINTS_TO_PERFORM	Preset number of print cycles after which the printer automatically enters the STOPPED state. May be 0 if no limit is desired.

4.2.1.3 Negative RESPONSE definition

Error Code	Name	Summary
	NOT_READY	The device is not in a READY state and therefore cannot print.

4.2.2 STOP

4.2.2.1 COMMAND definition

Unload label, reset print cycle information, raise cylinder, idle all motors - enter STOPPED state.

Execution of this command is deferred until the current print cycle is completed or aborted.

While a response is generated immediately, this does not imply that the printer has already changed state. Watch the relevant NOTIFICATION messages to determine when the printer has stopped.

COMMAND_ID: 0x0090

DATA: unload format

Index & Type	Name	Summary
0 uint8	UNLOAD_FORMAT	0=preserve format for next startup 1=permanent unload

4.2.2.2 Positive RESPONSE definition

Index & Type	Name	Summary
--------------	------	---------

APP and APC specification 1.03

0 uint16	PRINTS_PERFORMED	Number of the triggered print cycle since entering the ready state.
------------	------------------	---

4.2.2.3 Negative RESPONSE definition

Error Code	Name	Summary
	NOT_READY	The device is not in a READY state and therefore cannot STOP.

4.2.3 LABEL_LOAD

4.2.3.1 COMMAND definition

Prepare a label to be printed with the specified quantity. If successful, the printer enters the READY state waiting for a print-trigger command.

*Normally, this command only requires the DATA fields to retrieve an already saved label in the file storage. However, an external labels also can be sent by specifying the OPTIONAL fields.

** If external label is used, it's not saved in the file storage, it's only prepared to be printed in the current session. To save a label in the file storage, please refer to section 4.4.1

COMMAND_ID: 0x00A0

DATA: Print Quantity, Label name

OPTIONAL: Parser, Label Length, Label Offset, Label Contents

Index & Type	Name	Summary
0 cstring	LABEL_NAME	Name of the label, max. 18 chars
1 uint16	PRINTS_TO_PERFORM	Preset number of print cycles after which the printer automatically enters the STOPPED state. May be 0 if no limit is desired.
2 uint8	PARSER	0=NONE (RAW bmp 1bpp) 1=MLI01
3 uint32	LABEL_LENGTH	Total length of the label data in bytes
4 uint32	LABEL_OFFSET	Represents the position of data in the label
5 repeats byte	LABEL_CONTENTS	Contents of the label

4.2.3.2 Positive RESPONSE definition

Default response with no data part.

4.2.3.3 Negative RESPONSE definition

Error Code	Name	Summary
	NO_LABEL	The label was not found by label storage.
	STORAGE_ERROR	Label storage has reported an unhandled error.
	ALREADY_READY	The printer is already READY and must be STOPPED before processing a PRINT command.
	LABEL_ERROR_SHORT	Signature of label is incorrect
	LABEL_ERROR_FORMAT_UNKNOWN	
	LABEL_ERROR_OOM	Memory allocation for label length failed.
	LABEL_ERROR_LONG	Label is bigger than expected length
	LABEL_ERROR_NOT_CONTIGUOUS	Expected label offset not found
	LABEL_OK_INCOMPLETE	First package is sent successfully, but whole label is still incomplete.
	LABEL_PARSE_ERROR	Label transfer is successful, but content could not be parsed
	LABEL_ERROR_PARSER_UNKNOWN	Label transfer is successful, but parser is unknown

4.2.4 LABEL_POSTVIEW

4.2.4.1 COMMAND definitions

Get a file from the file storage.

COMMAND_ID: 0x0202

DATA: File name.

Index & Type	Name	Summary
0 uint8	PRINthead_NUMBER	Get actual post view data for selected printhead
1 uint8	IMAGEFORMAT	0=RAW MONOCHROME BITMAP 1=PNG

4.2.4.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 uint16	STREAM_ID	IF > 0 the file will be sent later as a stream message with STREAM_ID
1 uint32	POSTVIEW_LENGTH	Total file length in bytes
2 uint32	POSTVIEW_OFFSET	Represents the position in the file. If this value is zero, it's a new file or an existing file to be overwritten.
3 repeats byte	POSTVIEW_BYTES	File Content

4.3 Device configuration

A configuration key must be uniquely defined and, once defined, a given key must not be redefined in future protocol versions.

4.3.1 Configuration key-value pair definition

Name	Summary	Unit	Type Range	restart needed	idle only	debug	Default Value
IM_FORWARD_HEAD_SPEED	Desired speed of the forward print head movement for intermittent mode print operations.	Mm/s	uint16 50-300 depending on TPH	0	1	0	50
IM_RETRACT_HEAD_SPEED	Desired speed of the retraction print head movement for intermittent mode print operations.	Mm/s	uint16 50-400	0	0	0	250

TPH_CONTRAST	Arbitrarily scaled value for print intensity, aka print contrast.	-	uint16 0-100 step 10	0	1	0	50
TPH_RESISTANCE	Installation parameter for expected print head resistance value. This represents the internal impedance of the printing element. This value must match the resistance of the print head in use. Only for print heads which support setting the resistance	Ω (ohm)	VECTOR <uint16>*12 array of 12 uint16 values ranging from 1,063 – 2,052 depends on tph	0	1	0	All fields: absolute minimal resistance value for tph which can be read from pic, depending on tph
TPH_HYSTERESIS PARAMETER_A	for debug issues and for internal use only	-	uint32 Range of uint32	0	1	1	Read from pic, depend on tph
TPH_HYSTERESIS PARAMETER_B	for debug issues and for internal use only	-	uint32 Range of uint32	0	1	1	Read from pic, depends on tph
TPH_HYSTERESIS PARAMETER_C	for debug issues and for internal use only	-	uint32 Range of uint32	0	1	1	Read from pic, depends on tph
TPH_TEMPERATURE_HIGH	for debug issues and for internal use only	°C	int16 0-70	0	1	1	Read from pic, depends on tph
TPH_TEMPERATURE_LOW	for debug issues and for internal use only	°C	int16 0-70	0	1	1	Read from pic, depends on tph
TPH_PRINTTIMEOUT	for debug issues and for internal use only	ms	uint16 0-1000	0	1	1	Read from pic, depends on tph

TPH_USERMASK	User-configured active print heads. Setting a TPH_USERMASK that includes print heads that are not available to the hardware results in an error response.	-	VECTOR <uint8>*12 array of 12 uint8 values ranging from 0-1	0	1	0	0
DATETIME.UTC	The printer clock, always set as UTC. Day is a two-digit numeric value (01-31) Month is a two-digit numeric value (01-12) Year is a four-digit numeric value Hour is a two-digit numeric value (00-23) Minutes is a two-digit numeric value (00-59) Seconds is a two-digit numeric value (00-59)	-	String yyyyMMddHHmmss	0	0	0	1990-01-01 -00-00-00
NETWORK_MAC_ID	MAC address	-	string 6 bytes in hex - valid MAC	1	0	0	02AABB CCDDEE
NETWORK_GATEWAY	Gateway for TCP/IP connections		uint8 0,0,0,0	0	0	0	0
NETWORK_SUBNET_MASK	Subnet mask for TCP/IP connections		uint8 255,255,255,0	0	0	0	0
NETWORK_IP	IP for TCP/IP connections		uint8 192, 168, 173, 68	0	0	0	0
NETWORK_PORT	PORT for TCP/IP connections		Uint16 1024-49151	0	0	0	30001

RIBBON_USE	Using ribbon 0 – Printing without using ribbon 1 – Printing with ribbon installed.	-	uint8 0 1	0	0	0	0
RIBBON_SPOOL_DISTANCE_CLEAR	Ribbon spooling distance for a complete ribbon clear	1/12 th mm	uint16 0-11988*2	0	0	0	0
RIBBON_SPOOL_DISTANCE_FULL	Ribbon spooling distance for a “full spool”	1/12 th mm	uint16 0-11988	0	0	0	0
RIBBON_SPOOL_DISTANCE_PARTIAL	Ribbon spooling distance for a “partial spool”	1/12 th mm	uint16 0-2,400	0	0	0	0
RIBBON_SPOOL_NUM_PARTIALS	Number of Partial spools to be performed	-	uint8 Range of uint8	0	0	0	0
RIBBON_SPOOL_SPEED_FULL	Ribbon spooling top speed for a “full spool”	mm/s	uint16 100-2000	0	0	0	1500
RIBBON_SPOOL_SPEED_CLEAR	Ribbon spooling top speed for a “clear spool”	mm/s	uint16 100-2000	0	0	0	1200
RIBBON_SPOOL_SPEED_PARTIAL	Ribbon spooling top speed for a “partial spool”	mm/s	uint16 100-2,000	0	0	0	110
PRINT_LENGTH	Length of the useable print area, not counting acceleration and stopping distances.	1/12 th mm	uint16 0-2400	0	0	0	0
AUTO_LENGTH	Printer automatically sets the print length for the loaded label.	-	uint8 0 1	0	0	0	1
AUTO_INIT	Printer automatically loads last valid format after power on. This value is initialized after a firmware	-	uint8 0 1	0	0	0	1

	update to 0						
AUTO_READY	Printer automatically attempts to enter the “ready” state on power up. This value is initialized after a firmware update to 0	-	uint8 0 1	0	0	0	0
AUTO_PRINT	Once the first print signal is received, the printer will keep printing until it encounters a stop condition. This value is initialized after a firmware update to 0	-	uint8 0 1	0	0	0	0
AUTO_START	Printer automatically starts printing after power on. This value is initialized after a firmware update to 0	-	uint8 0 1	0	0	0	0
AUTO_TRIM	Printer automatically ignores the whitespace of the top of the print format	-	uint8 0 1	0	0	0	1
DELAY_AUTO_PRINT	Delay time between automatic prints when AUTO_PRINT is active.	ms	uint16 0-10,000	0	0	0	0
DELAY_CARRIAGE_MOVE_PRINT	Delay time between the beginning of the activation (lowering) of the print heads and the beginning of carriage print movement.	ms	uint16 0-5,000	0	0	0	0

DELAY_STAGGERED_PRINTHEAD_ACTIVATION	Delay time between the activation (lowering) of individual print heads.	ms	uint16 5-500	0	0	0	0
DELAY_EXTERNAL_PRINT_SIGNAL	Delay between occurrence of external GPIO print signal and print sequence start.	ms	uint16 0-10,000	0	0	0	0
DELAY_CARRIAGE_MOVE_RETURN	Delay time between the the deactivation (raising) of the print-heads and the beginning of carriage return movement. In ms.	ms	uint16 0-5,000	0	0	0	0
DELAY_RIBBON_SPOOL_PARTIAL	Delay time between the the deactivation (raising) of the print-heads and the beginning of ribbon partial spooling. In ms.	ms	uint16 0-5,000	0	0	0	20
DELAY_RIBBON_SPOOL_FULL	Delay time between the the deactivation (raising) of the print heads and the beginning of ribbon full spooling. In ms.	ms	uint16 0-5,000	0	0	0	10
ACCELERATION_OFFSET	Offset in mm before print start	1/12 th mm	uint16 0-240	36	0	0	0
FORMAT_OFFSET_X	Global format offset X	1/12 th mm	int16 -672 to 672	0	0	0	0
FORMAT_OFFSET_Y	Global format offset Y	1/12 th mm	int_16 -2400 to 2400	40	0	0	0
TPH_FORMAT_OFFSET_X	Print head format offset X	1/12 th mm	int16 -672 to 672	0	0	0	0

TPH_FORMAT_OFFSET_Y	Print head format offset Y	1/12th mm	int16 -2400 to 2400	0	0	0	0
NOTIFICATION_PRINTED	print notification (NOTIFY_PRINTED) after each print cycle	-	uint8 0 1	0	0	0	0
LOCALE	Localization and language-specific values such as weekdays	-	string 3 bytes EN,DE,FR,ES,TW, KR,RU,SE,TR,NL, PT,IT,IS,HU	0	0	0	en
MACHINE_PRINTS	Prints counter	-	UInt32	0	0	0	0
NETWORK_DHCP_ENABLED	Enable DHCP		uint8 0 1	0	0	0	0
ACCELERATION_OFFSET	Offset at begin of print format	12/mm	uint16 0-240	0	1	0	0
SLOW_DOWN_OFFSET	Offset at end of print format	12/mm	uint16 0-100	0	1	0	0
TPH_ASSEMBLY_DISTANCE	Distance between 2 print heads	12/mm	uint16 36-28800	0	0	0	0
AUTO_RIBBON_PARAMETERS	Enable automatic calculation of spool parameters		uint8 0 1	0	0	0	0
FORMAT_FONTNAME	Current font name		ASCII 18	0	0	0	0
RIBBON_WARNING_LENGTH	Printer raises a warning if the calculated ribbon length gets lower than this value	m	UInt16 0-10000	0	0	0	25
RIBBON_TOTAL_LENGTH	Total length of installed ribbons	m	UInt16 0-10000	0	0	0	600
RIBBON_WARNING_ENABLED	Enable raising ribbon warning	-	uint8 0 1	0	0	0	0

TPH_APPLIED_POWER_FOR_LE VELS	Power levels for rohm print heads		VECTOR <uint8>*6 array of 6 uint8 values ranging from 0-100				50*6
----------------------------------	---	--	---	--	--	--	------

4.3.2 CONFIG_SET

4.3.2.1 COMMAND definition

Set a configuration value.

COMMAND_ID: 0x00B0

DATA: Configuration key, associated data.

Index & Type	Name	Summary
0 cstring	CONFIGURATION_KEY	Printable zero-terminated ASCII string specifying the configuration key. No longer than 18 characters.
1 as specified	CONFIGURATION_VALUE	As specified above.

4.3.2.2 Positive RESPONSE definition

Default response with no data part.

4.3.2.3 Negative RESPONSE definition

Error Code	Name	Summary
	INVALIDPAYLOAD	The payload contained incorrect data (length, layout, undetermined sting fields, etc.)
	INVALID_KEY	The key identifier contains characters that are not printable ASCII.
	UNKNOWN_KEY	The configuration key is not known by the printer.
	IDLE_ONLY_KEY	The configuration key setting requires the printer to be in the IDLE state, but it was not.
	VALUE_OUT_OF_RANGE	The configuration value was outside of the range of accepted values. Detail may be provided in the RESPONSE data part as a zero-terminated printable ASCII string.

4.3.3 CONFIG_GET

4.3.3.1 COMMAND definition

Get a configuration value.

COMMAND_ID: 0x00B1

DATA: Configuration key.

Index & Type	Name	Summary
0 cstring	CONFIGURATION_KEY	Printable zero-terminated ASCII string specifying the configuration key. No longer than 18 characters.

4.3.3.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 as specified	CONFIGURATION_VALUE	As specified above.

4.3.3.3 Negative RESPONSE definition

Error Code	Name	Summary
	INVALID_KEY	The key identifier contains characters that are not printable ASCII.
	UNKNOWN_KEY	The configuration key is not known by the printer.

4.3.4 CONFIG_LISTKEYS

4.3.4.1 COMMAND definition

Retrieve a list of all configuration keys known and supported.

COMMAND_ID: 0x00B2

DATA: None.

4.3.4.2 Positive RESPONSE definition

Index & Type	Name	Summary
repeats cstring	CONFIGURATION_KEY	Printable zero-terminated ASCII string specifying the configuration key. No longer than 18 characters.

4.3.4.3 Negative RESPONSE definition

None defined.

4.3.5 CONFIG_VALUETYPE

4.3.5.1 COMMAND definition

Get value type information for a configuration key.

COMMAND_ID: 0x00B3

DATA: Configuration key.

Index & Type	Name	Summary
0 cstring	CONFIGURATION_KEY	Printable zero-terminated ASCII string specifying the configuration key. No longer than 18 characters.

4.3.5.2 Positive RESPONSE definition

Index & Type	Name	Summary
1 Boolean (uint8)	RESTART_NEEDED	Printer needs to be restarted before activating a new configuration value.
1 Boolean (uint8)	IDLE_ONLY	Value can only be changed when machine is not printing
2 Boolean (uint8)	DEBUG	Configuration key is used for internal use only
3 cstring	UNIT	Unit of measurement, printable zero-terminated ASCII string as specified below.
4 cstring	TYPE_RANGE	Value type and range, printable zero-terminated ASCII string as specified in Appendix B.
5 repeats byte	DEFAULT_VALUE	The configuration default value bytes as defined in TYPE_RANGE

4.3.5.3 Negative RESPONSE definition

Error Code	Name	Summary
	INVALID_KEY	The key identifier contains characters that are not printable ASCII.
	UNKNOWN_KEY	The configuration key is not known by the printer.

4.4 File and print label storage operations

4.4.1 FILE_STORAGE_PUT

4.4.1.1 COMMAND definitions

Put file in the file storage.

COMMAND_ID: 0x0070

DATA: File information (Name, Length, offset, overwrite flag, content)

Index & Type	Name	Summary
0 cstring	FILE_NAME	Printable zero-terminated ASCII string specifying the file name including the extension. No longer than 18 characters.
1 uint32	FILE_LENGTH	Total file length in bytes
2 uint32	FILE_OFFSET	Represents the position in the file. If this value is zero, it's a new file or an existing file to be overwritten.
3 uint8	FILE_OVERWRITE_FLAG	If set, the file should be overwritten
4 repeats byte	FILE_BYTES	File content

4.4.1.2 Positive RESPONSE definition

Default response with no data part.

4.4.1.3 Negative RESPONSE definition

Error Code	Name	Summary
	FILE_STORAGE_ERROR	File was not received and stored

4.4.2 FILE_STORAGE_DELETE

4.4.2.1 COMMAND definitions

Delete a file from the file storage.

COMMAND_ID: 0x0071

DATA: File name.

Index & Type	Name	Summary
0 cstring	FILE_NAME	Printable zero-terminated ASCII string specifying the file name including the extension. No longer than 18 characters.

4.4.2.2 Positive RESPONSE definition

Default response with no data part.

4.4.2.3 Negative RESPONSE definition

Error Code	Name	Summary
	FILE_STORAGE_ERROR	File was not deleted

4.4.3 FILE_STORAGE_LIST

4.4.3.1 COMMAND definitions

List files of a selected type or all files in the file storage.

COMMAND_ID: 0x0072

DATA: File type and display mode.

Index & Type	Name	Summary
0 cstring	FILE_TYPE	Printable zero-terminated ASCII string specifying the file type: "*" – All files are listed. Following default suffixes are used for formats: "mli" - Binary UTF8-encoded Mli print format "raw" - Raw file
1 uint8	MODE	0x00 = FS_DIR_MODE_FILENAME ASCII value for MFS display mode * = MFS_ATTR_ANY r = MFS_ATTR_READ_ONLY s = MFS_ATTR_SYSTEM_FILE h = MFS_ATTR_HIDDEN_FILE d = MFS_ATTR_DIR_NAME v = MFS_ATTR_VOLUME_NAME a = MFS_ATTR_ARCHIVE u = FS_DIR_MODE_UNIX m = FS_DIR_MODE_MS DOS f = FS_DIR_MODE_FILENAME

4.4.3.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 cstring	FILE_NAMES	Zero-terminated ASCII string specifying the files of selected type in the file storage, delimited with /r/n
1 uint32	LIST_LENGTH	Total list length in bytes
2 uint32	LIST_OFFSET	Represents the position in the list. If this value is zero, it's a new list.
3 repeats byte	LIST_BYTES	List Content

4.4.3.3 Negative RESPONSE definition

Error Code	Name	Summary
	FILE_STORAGE_ERROR	Error reading file storage list

4.4.4 FILE_STORAGE_FREE

4.4.4.1 COMMAND definitions

Delete files by type from the file storage.

COMMAND_ID: 0x0073

DATA: File type.

Index & Type	Name	Summary
1 cstring	FILE_TYPE	printable zero-terminated ASCII string specifying the file type: "*" - all files are deleted "mli" - all binary UTF8-encoded MLI print format are deleted "raw" - all Raw files are deleted

4.4.4.2 Positive RESPONSE definition

Default response with no data part.

4.4.4.3 Negative RESPONSE definition

Error Code	Name	Summary
	FILE_STORAGE_ERROR	Error while deleting the files.

4.4.5 FILE_STORAGE_GET

4.4.5.1 COMMAND definitions

Get a file from the file storage.

COMMAND_ID: 0x0074

DATA: File name.

Index & Type	Name	Summary
0 cstring	FILE_NAME	Printable zero-terminated ASCII string specifying the file name including the extension. No longer than 18 characters.

4.4.5.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 uint16	STREAM_ID	IF > 0 the file will be sent later as a stream message with STREAM_ID
1 uint32	FILE_LENGTH	Total file length in bytes
2 uint32	FILE_OFFSET	Represents the position in the file. If this value is zero, it's a new file or an existing file to be overwritten.
3 repeats byte	FILE_BYTES	File Content

4.4.5.3 Negative RESPONSE definition

Error Code	Name	Summary
	FILE_STORAGE_ERROR	File was not sent

4.4.6 LIST_LABELS

4.4.6.1 COMMAND definitions

List all the Labels' names from the storage.

COMMAND_ID: 0x0075

DATA: File name.

4.4.6.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 cstring	FILE_NAMES	Zero-terminated ASCII string specifying the files of selected type in the file storage, delimited with /r/n
1 uint32	LIST_LENGTH	Total list length in bytes
2 uint32	LIST_OFFSET	Represents the position in the list. If this value is zero, it's a new list.
3 repeats byte	LIST_BYTES	List Content

4.4.6.3 Negative RESPONSE definition

Error Code	Name	Summary
	FILE_STORAGE_ERROR	Error reading file storage list

4.4.7 LABEL_EXISTS

4.4.7.1 COMMAND definitions

Checks if a label exists in the storage.

Index & Type	Name	Summary
0 cstring	FILE_NAME	Printable zero-terminated ASCII string specifying the file name including the extension. No longer than 18 characters.

4.4.7.2 Positive RESPONSE definition

Default response with no data part.

4.4.7.3 Negative RESPONSE definition

Error Code	Name	Summary
	< APC_ERRORS >	APC Error code, if label does not exists APC_ERROR_SD_FILE_EXISTS = 0x0012

4.5 Variable contents

4.5.1 Printer variable type definition

A variable can be marked VOLATILE, which results in the variable key being deleted after being used in as many print cycles as indicated.

Currently only simple zero terminated string-type variables are supported.

Future support may include

- formatted numbers
- triggered counters
- timed counters
- counters with pre-defined sequences (e.g. "A1", "B1", "B2", "A2", ...)
- date and time offsets
- ...?

4.5.2 VARIABLE_SET

4.5.2.1 COMMAND definition

Set a printer-wide variable value for use in labels.

COMMAND_ID: 0x00C0

DATA: variable key, associated data.

Index & Type	Name	Summary
0 uint8	VARIABLE_TYPE	1=VARIABLE_TEXT 2= VARIABLE_COUNTER 3= VARIABLE_DATE 4= VARIABLE_TEXT (Append serialization buffer)
1 cstring	VARIABLE_NAME	Printable zero-terminated ASCII string specifying the variable key. No longer than 18 characters.
2 as specified	VARIABLE_VALUE	Printable zero-terminated ASCII string specifying the variable key. Multiple values for serialization could be set, delimited with ESC (27) . No longer than 40 characters per string

4.5.2.2 Positive RESPONSE definition

Default response with no data part.

4.5.2.3 Negative RESPONSE definition

Error Code	Name	Summary
------------	------	---------

	INVALID_KEY	The key identifier contains characters that are not printable ASCII.
	VALUE_OUT_OF_RANGE	The value was longer than 4 kilobytes.
	UNKNOWN_KEY	The key identifier was not found

4.5.3 VARIABLE_GET

4.5.3.1 COMMAND definition

Get a variable value.

COMMAND_ID: 0x00C1

DATA: Variable contents.

Index & Type	Name	Summary
1 cstring	VARIABLE_KEY	Printable zero-terminated ASCII string specifying the variable key. No longer than 18 characters.

4.5.3.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 uint8	VARIABLE_TYPE	1=VARIABLE_TEXT 2= VARIABLE_COUNTER 3= VARIABLE_DATE
1 as specified	VARIABLE_VALUE	As specified above.

4.5.3.3 Negative RESPONSE definition

Error Code	Name	Summary
	UNKNOWN_KEY	The configuration key is not known by the printer.

4.5.4 VARIABLE_DELETE

4.5.4.1 COMMAND definitions

Delete a variable from the global variable storage.

COMMAND_ID: 0x00C3

DATA: VARIABLE name.

Index & Type	Name	Summary
0 cstring	LABEL_NAME	Printable zero-terminated ASCII string specifying the label name. All stored variable content for the specified label are reset to their original values. If no LABEL_NAME is specified, all stored variables for all labels are reset to their original values.

4.5.4.2 Positive RESPONSE definition

Default response with no data part.

4.5.4.3 Negative RESPONSE definition

Error Code	Name	Summary
	VARIABLE_NOT_FOUND	Variable name was not found or is not a global variable

4.5.5 VARIABLE_LISTKEYS

4.5.5.1 COMMAND definition

Retrieve a list of all variable keys currently set.

COMMAND_ID: 0x00C2

DATA: list mode

1 string	LABEL_NAME	Printable zero-terminated ASCII string defining the label name For the variable to be displayed. If label name is empty, the global variables are listed
------------	------------	--

4.5.5.2 Positive RESPONSE definition

Index & Type	Name	Summary
repeats cstring	VARIABLE_KEY	Printable zero-terminated ASCII string specifying the configuration key. No longer than 18 characters.

4.5.5.3 Negative RESPONSE definition

None defined.

4.6 Font

--	--	--

4.6.1 FONT_STORAGE_PUT

4.6.1.1 COMMAND definitions

Put font in the font storage of the Freescale processor.

COMMAND_ID: 0x0088

DATA: Font information (Name, Length, offset, overwrite flag, content)

Index & Type	Name	Summary
0 cstring	FONT_NAME	Printable zero-terminated ASCII string specifying the font name including the extension. No longer than 18 characters.
1 uint32	FONT_LENGTH	Total font length in bytes
2 uint32	FONT_OFFSET	Represents the position in the font. If this value is zero, it's a new font or an existing font to be overwritten.
3 uint8	FONT_OVERWRITE_FLAG	If set, the font should be overwritten

4 repeats byte	FONT_BYTES	Font content
-----------------------	------------	--------------

4.6.1.2 Positive RESPONSE definition

Default response with no data part.

4.6.1.3 Negative RESPONSE definition

Error Code	Name	Summary
	FONT_STORAGE_ERROR	Font was not received and stored

4.6.2 FONT_STORAGE_DELETE

4.6.2.1 COMMAND definitions

Delete a font from the font storage.

COMMAND_ID: 0x0081

DATA: Font name.

Index & Type	Name	Summary
0 cstring	FONT_NAME	Printable zero-terminated ASCII string specifying the font name including the extension. No longer than 18 characters.

4.6.2.2 Positive RESPONSE definition

Default response with no data part.

4.6.2.3 Negative RESPONSE definition

Error Code	Name	Summary
	FONT_STORAGE_ERROR	Font was not deleted

4.6.3 FONT_STORAGE_LIST

4.6.3.1 COMMAND definitions

List fonts of a selected type or all fonts in the font storage.

COMMAND_ID: 0x0082

DATA: Font type and display mode.

Index & Type	Name	Summary
0 cstring	FONT_TYPE	Printable zero-terminated ASCII string specifying the font type: "*" – All fonts are listed. Following default suffixes are used for formats: "mli" - Binary UTF8-encoded Mli print format "raw" - Raw font
1 uint8	MODE	0x00 = FS_DIR_MODE_FONTNAME ASCII value for MFS display mode * = MFS_ATTR_ANY r = MFS_ATTR_READ_ONLY s = MFS_ATTR_SYSTEM_FONT h = MFS_ATTR_HIDDEN_FONT d = MFS_ATTR_DIR_NAME v = MFS_ATTR_VOLUME_NAME a = MFS_ATTR_ARCHIVE u = FS_DIR_MODE_UNIX m = FS_DIR_MODE_MSDOS f = FS_DIR_MODE_FONTNAME

4.6.3.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 cstring	FONT_NAMES	Zero-terminated ASCII string specifying the fonts of selected type in the font storage, delimited with /r/n
1 uint32	LIST_LENGTH	Total list length in bytes
2 uint32	LIST_OFFSET	Represents the position in the list. If this value is zero, it's a new list.
3 repeats byte	LIST_BYTES	List Content

4.6.3.3 Negative RESPONSE definition

Error Code	Name	Summary
	FONT_STORAGE_ERROR	Error reading font storage list

4.6.4 FONT_STORAGE_GET

4.6.4.1 COMMAND definitions

Get a font from the font storage.

COMMAND_ID: 0x0084

DATA: Font name.

Index & Type	Name	Summary
0 cstring	FONT_NAME	Printable zero-terminated ASCII string specifying the font name including the extension. No longer than 18 characters.

4.6.4.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 uint16	STREAM_ID	IF > 0 the font will be sent later as a stream message with STREAM_ID
1 uint32	FONT_LENGTH	Total font length in bytes
2 uint32	FONT_OFFSET	Represents the position in the font. If this value is zero, it's a new font or an existing font to be overwritten.
3 repeats byte	FONT_BYTES	Font Content

4.6.4.3 Negative RESPONSE definition

Error Code	Name	Summary
	FONT_STORAGE_ERROR	Font was not sent

4.6.5 LIST_FONTS

4.6.5.1 COMMAND definitions

List all the Fonts' names from the storage.

COMMAND_ID: 0x0085

DATA: Font name.

4.6.5.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 cstring	FONT_NAMES	Zero-terminated ASCII string specifying the fonts of selected type in the font storage, delimited with /r/n
1 uint32	LIST_LENGTH	Total list length in bytes
2 uint32	LIST_OFFSET	Represents the position in the list. If this value is zero, it's a new list.
3 repeats byte	LIST_BYTES	List Content

4.6.5.3 Negative RESPONSE definition

Error Code	Name	Summary
	FONT_STORAGE_ERROR	Error reading font storage list

4.6.6 FONT_EXISTS

4.6.6.1 COMMAND definitions

Checks if a font exists in the storage.

Index & Type	Name	Summary
0 cstring	FONT_NAME	Printable zero-terminated ASCII string specifying the font name including the extension. No longer than 18 characters.

4.6.6.2 Positive RESPONSE definition

Default response with no data part.

4.6.6.3 Negative RESPONSE definition

Error Code	Name	Summary
	< APC_ERRORS >	APC Error code, if font does not exists APC_ERROR_SD_FONT_EXISTS = 0x0012

4.6.7 FONT_UPDATE

4.6.7.1 COMMAND definitions

Put file in the font storage. For 512K Boards the font is stored always permanent into serial flash

COMMAND_ID: 0x00FA

DATA: Fixed Font (Length, offset, content)

Index & Type	Name	Summary
0 uint32	CHECKSUM	CRC-32 of Font content (ISO 3309)
1 uint32	FONT_DATA_LENGTH	Total file length in bytes
2 uint32	FONT_DATA_OFFSET	Represents the position in the file. If this value is zero, it's a new file or an existing file to be overwritten.
3 repeats byte	FONT_BYTES	Font content

4.6.7.2 Positive RESPONSE definition

Default response with no data part.

4.6.7.3 Negative RESPONSE definition

Error Code	Name	Summary
	ERROR_FONT_OK_INCOMPLETE	waiting for continued font data
	ERROR_FONT_LONG	font data size too large
	ERROR_FONT_NOT_CONTIGUOUS	font data offsets are not contiguous
	ERROR_FONT_OOM	font memory allocation error
	ERROR_FONT_CRC_MISMATCH	font CRC mismatch
	ERROR_FONT_SAVE	font could not be saved
	ERROR_FONT_NULL	font is null
	ERROR_FONT_USER	user font load error from sd
	ERROR_FONT_TOO_BIG	user font not matching into SF
	ERROR_FONT_FONTNUMBER	user font number wrong, do a FONT_UPDATE_SELECTSLOT before

4.6.8 FONT_LOAD

4.6.8.1 COMMAND definition

Prepare a font to be printed with the current label. The font persists as default font for all following labels

COMMAND_ID: 0x0087

DATA: Font name

OPTIONAL: Font type, Font Length, Font Offset, Font Contents

Index & Type	Name	Summary
0 cstring	FONT_NAME	Name of the font, max. 18 chars
1 uint8	FONT_LOAD_MODE	0=LOAD 1=LOAD_AND_PERSIST_TO_SERIALFLASH
2 uint8	FONT_TYPE	0=NOT_SPECIFIED 1=TTF
3 uint32	FONT_LENGTH	Total length of the font data in bytes
4 uint32	FONT_OFFSET	Represents the position of data in the font
4 repeats byte	FONT_CONTENT	Content of the font

4.6.8.2 Positive RESPONSE definition

Default response with no data part.

4.6.8.3 Negative RESPONSE definition

Error Code	Name	Summary
	NO_FONT	The font was not found by label storage.
	STORAGE_ERROR	Font storage has reported an unhandled error.
	FONT_ERROR_FORMAT_UNKNOWN	
	FONT_ERROR_OOM	Memory allocation for font length failed.
	FONT_ERROR_LONG	Font is bigger than expected length
	FONT_ERROR_NOT_CONTIGUOUS	Expected label offset not found

	FONT_OK_INCOMPLETE	First package is sent successfully, but whole label is still incomplete.
	FONT_FONTTYPE_ERROR	Font transfer is successful, but content could not be verified
	FONT_ERROR_FONTTYPE_UNKNOWN	Font transfer is successful, but fonttype is unknown
	ERROR_COMMAND_NOT_IDLE	Not in state idle

4.6.9 FONT_UPDATE_SELECTSLOT

4.6.9.1 COMMAND definitions

Put file in the file storage.

COMMAND_ID: 0x00FA

DATA: Fixed Font (Length, offset, content)

Index & Type	Name	Summary
0 uint8	SLOT_NR	Select slot number before updating the serial flash with a font 1=SLOT1 2=SLOT2

4.6.9.2 Positive RESPONSE definition

Default response with no data part.

4.6.9.3 Negative RESPONSE definition

Error Code	Name	Summary
	ERROR_FONT_OK_INCOMPLETE	waiting for continued font data
	ERROR_FONT_LONG	font data size too large
	ERROR_FONT_NOT_CONTIGUOUS	font data offsets are not contiguous
	ERROR_FONT_OOM	font memory allocation error
	ERROR_FONT_CRC_MISMATCH	font CRC mismatch
	ERROR_FONT_SAVE	font could not be saved
	ERROR_FONT_NULL	font is null
	ERROR_FONT_USER	user font load error from sd
	ERROR_FONT_TOO_BIG	user font not matching into SF
	ERROR_FONT_FONTNUMBER	user font number wrong

4.7 PIC

Notes:

- A 32 bit value is divorced like that: 0xMB_UB_HB_LB
- A 16 bit value is divorced like that: 0xHB_LB
- Flash, program flash, etc. means that part of PIC Flash which is allocated for the application.
- Can only performed when printer status information is idle or below.

4.7.1 PIC_UPDATESTART

4.7.1.1 COMMAND definitions

Switch PIC to boot-loader stage, if in application stage and erase that part of the program flash, which is allocated for the application.

COMMAND_ID: 0x00F0

DATA: none

4.7.1.2 Positive RESPONSE definition

Default response with no data part.

4.7.1.3 Negative RESPONSE definition

Error Code	Name	Summary
0x0001	UPDATE_ERROR	Error during entering bootloader mode or erasing program flash

4.7.2 PIC_UPDATEPKT

4.7.2.1 COMMAND definitions

The update packet is derived from the hex file to be programmed in the following manner:

The hex file is in Intel hex format. Each line represents a hexadecimal record starting with a colon (:) and is in ASCII format. The colon has to be discarded and the remaining data must be converted from ASCII to hexadecimal.

The length of one packet must not exceed 97 Bytes.

COMMAND_ID: 0x00F1

DATA: Firmware bytes

Index & Type	Name	Summary
0 Bytes 1 – 2	PACKET_LENGTH	length of the packet to transfer
2 repeats byte	PACKET_BYTES	PIC update bytes

4.7.2.2 Positive RESPONSE definition

Default response with no data part.

4.7.2.3 Negative RESPONSE definition

Error Code	Name	Summary
------------	------	---------

0x0001	UPDATE_ERROR	Error/Problems while transferring update packet
--------	--------------	---

4.7.3 PIC_UPDATEFINALIZE

4.7.3.1 COMMAND definitions

After programming: check for errors during update and verify the content of the programmed Flash. PIC remain in boot-loader state.

COMMAND_ID: 0x00F2

DATA: Checksum

Index & Type	Name	Summary
0 byte	ADRS_LB	start address LB
1 byte	ADRS_HB	start address HB
2 byte	ADRS_UB	start address UB
3 byte	ADRS_MB	start address MB
4 byte	NUMBYTES_LB	Number of bytes LB
5 byte	NUMBYTES_HB	Number of bytes HB
6 byte	NUMBYTES_UB	Number of bytes UB
7 byte	NUMBYTES_MB	Number of bytes MB
8 byte	CHECKSUM_LB	CRC the programmed flash should have LB
9 byte	CHECKSUM_HB	CRC the programmed flash should have MB

4.7.3.2 Positive RESPONSE definition

Default response with no data part.

4.7.3.3 Negative RESPONSE definition

Error Code	Name	Summary
	UPDATE_ERROR	The PIC Update has failed (Checksum does not fit, other errors occurred during update, entering application state)

4.8 FPGA

4.8.1 FPGA_UPDATE

4.8.1.1 COMMAND definitions

Put file in the file storage.

COMMAND_ID: 0x00F3

DATA: FPGA firmware (Length, offset, content)

Index & Type	Name	Summary
0 uint32	CHECKSUM	CRC-32 of FPGA firmware content (ISO 3309)
1 uint32	FPGA_DATA_LENGTH	Total file length in bytes
2 uint32	FPGA_DATA_OFFSET	Represents the position in the file. If this value is zero, it's a new file or an existing file to be overwritten.
3 repeats byte	FPGA_BYTES	FPGA firmware content

4.8.1.2 Positive RESPONSE definition

Default response with no data part.

4.8.1.3 Negative RESPONSE definition

Error Code	Name	Summary
	FPGA_STORAGE_ERROR	FPGA firmware was not received or stored

4.9 DIAGNOSTICS FUNCTIONS

4.9.1 DIAGNOSTICS

4.9.1.1 COMMAND definitions

Functions for printer diagnostics

COMMAND_ID: 0x00F4

DATA: Diagnose (Command, timeout, additional data length, additional data)

Index & Type	Name	Summary
0 uint8	DIAGNOSTICS_COMMAND	0 = LEAVE_DIAGNOSTICS_MODE 1 = ENTER_DIAGNOSTICS_MODE 2 = RIBBON_MOTOR_ON 3 = RIBBON_MOTOR_OFF 4 = PERFORM_CARRIAGE_MOVE 5 = PRESTROKE_DOWN 6 = PRESTROKE_UP 7 = PRINTSTROKE_DOWN 8 = PRINTSTROKE_UP 9 = GPIO_IN_RIBBON_RIGHT 10 = GPIO_IN_RIBBON_LEFT 11 = GPIO_IN_MACHINE_COVER 12 = GPIO_IN_HOME 13 = GPIO_IN_CARRIAGE_END 14 = GPIO_OUT_ALL 15 = GPIO_OUT_EXTERN_ERROR 16 = GPIO_OUT_EXTERN_READYTOPRINT 17 = GPIO_OUT_EXTERN_PRINTED 18 = GPIO_OUT_EXTERN_WARNING 19 = GPIO_OUT_EXTERN_INVERTED_PRINTED 20 = GPIO_OUT_TPH_POWER_STBY 21 = GPIO_OUT_ONBOARD_LED
1 uint32	DIAGNOSTICS_DATA_LENGTH	Length of data for diagnostics command 0 = default
2 repeats byte	DIAGNOSTICS_DATA_BYTES	Data to perform diagnostics command for 4 = Length carriage move

4.9.1.2 Positive RESPONSE definition

Index & Type	Name	Summary
0 uint8	RESULT	GPIO Bits or 0 = Command was not successful

		1 = Command successful
--	--	------------------------

4.9.1.3 Negative RESPONSE definition

Error Code	Name	Summary
	DIAGNOSTICS_ERROR	Command was not successful
	DIAGNOSTICS_TIMEOUT	Command timeout detected

APC notifications

4.10 List of defined NOTIFICATION messages

4.10.1 NOTIFY_INIT

4.10.1.1 NOTIFICATION definition

Indicates that the printer has started the regular boot process and the communication interfaces have initialized. The firmware has loaded and initialized communications, but is not yet fully operational. Wait for a NOTIFY_HELLO message before issuing a COMMAND.

NOTIFICATION_ID: 0x0010

DATA: None.

4.10.2 NOTIFY_HELLO

4.10.2.1 NOTIFICATION definition

Indicates that the printer has completed the boot process and is operational. The printer will send a NOTIFY_READY or NOTIFY_IDLE message following this message to indicate the boot-up state.

NOTIFICATION_ID: 0x0020

DATA: None.

4.10.3 NOTIFY_READY

4.10.3.1 NOTIFICATION definition

Indicates that the printer is ready to print a label. When in this state, the printer will attempt to print when a print trigger event occurs.

NOTIFICATION_ID: 0x0030

DATA: Name of label in print.

Index & Type	Name	Summary
0 cstring	LABEL_NAME	Name of the label that is ready for printing.

4.10.4 NOTIFY_IDLE

4.10.4.1 NOTIFICATION definition

Indicates that the printer is stopped. No attempt to print will occur when a print trigger event occurs.

NOTIFICATION_ID: 0x0040

DATA: None.

4.10.5 NOTIFY_ERROR

4.10.5.1 NOTIFICATION definition

Indicates that the printer has encountered an immediate error condition. This message does not in itself imply a transition to the STOPPED state, but it is usually preceded by a NOTIFY_IDLE message.

NOTIFICATION_ID: 0x0050

DATA: Error information.

Index & Type	Name	Summary
0 uint8	MODULE_ID	ID of the module the error occurred in. 0 - Renesas Bootloader 1 - Renesas User Application 2 - Freescale Bootloader 3 - Freescale User Application TBD
1 uint16	ERROR_CODE	Unique error code within the module. Error codes may be re-assigned with different meaning in different modules.
2 as specified	ERROR_DATA	Optional data block individually defined by the error message definition.

List of MODULE_ID (Renesas User Application) ERROR_CODE values and their meaning:

`APC_ERROR_CARRIAGE_LOST` = 1,
Carriage got lost.
`APC_ERROR_CASE_SENSOR` = 2,
Case is open.
`APC_ERROR_RIBBON_SENSOR` = 3,
Ribbon break is detected.
`APC_ERROR_HC_INIT_FAILED` = 4,
Hysteresis controller encountered problems during initialization.
`APC_ERROR_HC_CONFIG_FAILED` = 5,
Hysteresis controller encountered problems during configuration.
`APC_ERROR_NVRAM_RESET_OCCURED` = 6,
NVRAM configuration restore encountered problems, all configurations has been reset to defaults.

List of MODULE_ID (Freescale User Application) ERROR_CODE values and their meaning:

`APC_ERROR_OPENING_GPIO` = 1,
`APC_ERROR_SD_INIT` = 2,
`APC_ERROR_SD_PROTECTED` = 3,
`APC_ERROR_SD_MFS_INIT` = 4,
`APC_ERROR_SD_MFS_OPEN` = 5,
`APC_ERROR_SD_MFS_COMPATIBLE` = 6,
`APC_ERROR_SD_NOT_INSTALLED` = 7,
`APC_ERROR_SD_MFS_FORMAT_DIR` = 8,
`APC_ERROR_SD_UNINSTALLING_MFS` = 9,
`APC_ERROR_SD_FORMAT` = 10,
`APC_ERROR_SD_CREATE_DIR` = 11,
`APC_ERROR_SD_CHANGE_DIR` = 12,
`APC_ERROR_SD_ASSIGNING_MEMORY` = 13,
`APC_ERROR_SD_PATH` = 14,

```
APC_ERROR_SD_FILE_IO = 15,
APC_ERROR_SD_CANT_OPEN_FILE = 16,
APC_ERROR_SD_DELETING_FILE = 17,
```

4.10.6 NOTIFY_WARN

4.10.6.1 NOTIFICATION definition

Indicates that the printer has encountered a condition that may be cause for an ERROR if allowed to continue unchecked and should be addressed.

NOTIFICATION_ID: 0x0060

DATA: Warning information.

Index & Type	Name	Summary
0 uint8	MODULE_ID	ID of the module the error occurred in. 0 - Renesas Bootloader 1 - Renesas User Application 2 - Freescale Bootloader 3 - Freescale User Application TBD
1 uint16	WARNING_CODE	Unique warning code within the module. Warning codes may be re-assigned with different meaning in different modules.
2 as specified	WARNING_DATA	Optional data block individually defined by the warning message definition.

List of MODULE_ID (Renesas User Application) WARNING_CODE values and their meaning:

```
APC_WARNING_UNDEFINED = 0,
```

List of MODULE_ID (Freescale User Application) WARNING_CODE values and their meaning:

```
APC_WARNING_UNDEFINED = 0,
```

4.10.7 NOTIFY_INFO

4.10.7.1 NOTIFICATION definition

Indicates that the printer has encountered a condition that may be of interest for connected Applications.

NOTIFICATION_ID: 0x0070

DATA: Optional information.

Index & Type	Name	Summary
0 uint8	MODULE_ID	ID of the module the error occurred in. 0 - Renesas Bootloader 1 - Renesas User Application 2 - Freescale Bootloader 3 - Freescale User Application TBD
1 uint16	INFO_CODE	Unique Info code within the module. Info codes may be re-assigned with different meaning in different modules.
2 as specified	INFO_DATA	Optional data block individually defined by the information message definition.

List of MODULE_ID (Renesas User Application) INFO_CODE values and their meaning:

APC_INFO_UNHANDLED_MSGTYPE = 1,

An unknown message type was received.

APC_INFO_UNHANDLED_COMMAND = 2,

An unknown command was received.

List of MODULE_ID (Freescale User Application) INFO_CODE values and their meaning:

APC_INFO_UNHANDLED_MSGTYPE = 1,

An unknown message type was received.

APC_INFO_UNHANDLED_COMMAND = 2,

An unknown command was received.

4.10.8 NOTIFY_USERMESSAGE

4.10.8.1 NOTIFICATION definition

Indicates that the printer has encountered a condition that may be of interest for the User.

NOTIFICATION_ID: 0x0080

DATA: Warning information.

Index & Type	Name	Summary
0 cstring	MESSAGE	A message to display

4.10.9 NOTIFY_PRINTED

4.10.9.1 NOTIFICATION definition

Indicates that the printer has successfully processed and completed one print trigger event.

NOTIFICATION_ID: 0x1000

DATA: Warning information.

Index & Type	Name	Summary
0 uint16	PRINTS_PERFORMED	Number of successfully completed print cycles since entering the ready state.

4.10.10 NOTIFY_PRINT_LOST

4.10.10.1 NOTIFICATION definition

Indicates that the printer was unable to complete a requested print cycle – e.g. an external print cycle or print command was received, but the printer is unable to service the request.

NOTIFICATION_ID: 0x1200

DATA: Warning information.

4.10.11 NOTIFY_NEED_VARIABLE_DATA

4.10.11.1 NOTIFICATION definition

Indicates that the printer needs variable data set before it can process any further print cycles of the loaded label.

NOTIFICATION_ID: 0x2000

DATA: Information about missing variable IDs.

Index & Type	Name	Summary
repeats cstring	VARIABLE_IDENTIFIERS	List of zero-terminated printer- variable identifier keys that are required to be set before printing can proceed.

4.10.12 NOTIFY_CASECOVER_CLOSE

4.10.12.1 NOTIFICATION definition

Indicates that the printer needs variable data set before it can process any further print cycles of the loaded label.

NOTIFICATION_ID: 0x3000

DATA: Warning information.

Functional Part – More parts

5 Appendix A – Identifier enumerations

```
public enum COMMANDID
{
    ECHO = 0x0001, //
    GET_VERSION_INFORMATION = 0x0010,
    SOFT_REBOOT = 0xABBA,
    RETRIEVE_DIAGNOSTIC_LOG = 0x0020,
    GET_DEVICE_STATUS = 0x0030,

    DEVICE_INFO_GET = 0x0032,
    DEVICE_INFO_LISTKEYS = 0x0033,
    DEVICE_INFO_VALUETYPE = 0x0034,
    GET_PRINTHEAD_STATUS = 0x0035,

    GET_PENDING_ERRORS = 0x0040,
    GET_PRINT_INFO = 0x0050,

    FILE_STORAGE_PUT = 0x0070,
    FILE_STORAGE_DELETE = 0x0071,
    FILE_STORAGE_LIST = 0x0072,
    FILE_STORAGE_FREE = 0x0073,
    FILE_STORAGE_GET = 0x0074,
    LIST_LABELS = 0x0075,

    PRINT_TRIGGER = 0x0080,
    STOP = 0x0090,
    LABEL_LOAD = 0x00A0,
    LABEL_POSTVIEW = 0x0202,

    CONFIG_SET = 0x00B0,
    CONFIG_GET = 0x00B1,
    CONFIG_LISTKEYS = 0x00B2,
    CONFIG_VALUETYPE = 0x00B3,

    VARIABLE_SET = 0x00C0,
    VARIABLE_GET = 0x00C1,
    VARIABLE_LISTKEYS = 0x00C2,
    VARIABLE_DELETE = 0x00C3,

    FONT_ADD_FONT = 0x00FA,
    FONT_TRANSFERPKT = 0x00FB,
    FONT_PERSIST = 0x00FC,
    FONT_DELETE = 0x00FD,
    FONT_LIST = 0x00FE,

    PIC_UPDATESTART = 0x00F0,
    PIC_UPDATEPKT = 0x00F1,
    PIC_UPDATEFINALIZE = 0x00F2,

    FPGA_UPDATE = 0x00F3,
    DIAGNOSTICS = 0x00F4,

    RESET_DEFAULTS = 0x00FF,
    RAISENOTIFICATION = 0x1000,
}
```

```
public enum NOTIFICATIONID
{
    NOTIFY_UNDEFINED = 0x0000,
    NOTIFY_HELLO = 0x0020,
    NOTIFY_READY = 0x0030,
    NOTIFY_IDLE = 0x0040,
    NOTIFY_ERROR = 0x0050,
    NOTIFY_WARN = 0x0060,
    NOTIFY_INFO = 0x0070,
    NOTIFY_USERMESSAGE = 0x0080,
    NOTIFY_PRINTED = 0x1000,
    NOTIFY_PRINT_LOST = 0x1200,
    NOTIFY_NEED_VARIABLE_DATA = 0x2000,
}

public enum MESSAGETYPES
{
    NOTDEFINED = 0x0000,
    COMMAND = 0x0010,
    RESPONSE = 0x0030,
    NOTIFICATION = 0x0040,
    STREAM = 0x0050,
}

public enum DEVICE_STATUS
{
    UNINITIALIZED = 0x0000,
    INITIALIZED = 0x0001,
    IDLE = 0x0002,
    READY = 0x0004,
    PRINTING = 0x0008
}

public enum LOAD_AND_PRINT_RESPONSE
{
    NO_LABEL = 0x0001,
    STORAGE_ERROR = 0x0002,
    ALREADY_READY = 0x0003,
    VARIABLE_ID_NOT_FOUND = 0x0004,
    VARIABLE_ID_INVALID = 0x0005,
    VARIABLE_VALUE_INVALID,
    PRINT_HEADS_INVALID = 0x0007,
}
```

6 Appendix B – Value type definitions

The following VALUE_TYPE_DEFs are supported:

Simple types:

ASCII[max[,min]] – a printable ASCII string.
 UTF8[max[,min]] – a printable UTF-8 string.
 onoff – a binary toggle.
 uint8[max[,min]]
 uint16[max[,min]]
 uint32[max[,min]]
 int8[max[,min]]
 int16[max[,min]]
 int32[max[,min]]
 datetime[max[,min]] – a time_t
 time[max[,min]]
 date[max[,min]]
 timespan[max[,min]]
 listtypes
 TUPLE<any defined above;...>
 VECTOR<any defined above>

int8	8	–128, –127, ..., 127
uint8	8	0, 1, ..., 255
int16	16	–32 768, –32 767, ..., 32 767
uint16	16	0, 1, ..., 65 535
int32	32	–2 147 483 648, ..., 2 147 483 647
uint32	32	0, 1, ..., 4 294 967 295

The following units are supported:

1/12 mm
 mm
 cm
 m
 km
 Ω
 V
 %
 pt
 mm/s
 qty
 °C
 bytes

7 Error codes

```
#define ERROR_NONE 0

#define ERROR_BASE_CMD 0x1000
#define ERROR_BASE_FPGA 0x1100
#define ERROR_BASE_FPGA_FIRMWARE 0x1200
#define ERROR_BASE_MACHINE 0x1300
#define ERROR_BASE_HYST 0x1400
#define ERROR_BASE_FONT_RENDER 0x1500
#define ERROR_BASE_SERIAL_FLASH 0x1600
#define ERROR_BASE_LABEL 0x1700
#define ERROR_BASE_APC 0x1800
#define ERROR_BASE_NVRAM 0x1A00
#define ERROR_BASE_FONT 0x1B00
#define ERROR_BASE_IIC 0x1C00
#define ERROR_BASE_BARCODE 0x1D00
#define ERROR_BASE_CONFIG 0x1E00
#define ERROR_BASE_STEPDRV 0x1F00
#define ERROR_BASE_INTERRUPT 0x1FF0
#define ERROR_BASE_WARNING 0x2000
#define ERROR_DIAGNOSTICS 0x2100

enum ERROR_WARNING {
    WARNING_LOW_RIBBON = ERROR_BASE_WARNING,
    WARNING_LOW_SERIALIZATION_BUFFER,
    WARNING_RENDER_SLOW, ///< label rendering takes longer time
    WARNING_PRINTSIGNAL_TOO_FAST, ///< print signals come in too fast
};

enum ERROR_INTERRUPT {
    ERROR_INTERRUPT_ILLEGAL_CODE = ERROR_BASE_INTERRUPT,
    ERROR_INTERRUPT_ILLEGAL_SLOT,
    ERROR_INTERRUPT_CPU_ADDRESS,
    ERROR_INTERRUPT_BANK_OVERFLOW,
    ERROR_INTERRUPT_BANK_UNDERFLOW,
    ERROR_INTERRUPT_DIVIDE_BY_ZERO,
    ERROR_INTERRUPT_DIVIDE_OVERFLOW,
    ERROR_INTERRUPT_MEMORY_ALLOCATION
};

enum ERROR_STEPDRV {
    ERROR_STEPDRV_RAMP_STEPS = ERROR_BASE_STEPDRV,
    ERROR_STEPDRV_RAMP_SPEEDMIN, ///< error speed value < MIN_SPEED_PPS
    ERROR_STEPDRV_RAMP_SPEEDMAX, ///< error speed value > MAX_SPEED_PPS
    ERROR_STEPDRV_RAMP_SPEEDRANGE, ///< error ramp start or end speed value > max speed
    ERROR_STEPDRV_RAMP_RAMPUP, ///< error ramp_up NULL
    ERROR_STEPDRV_RAMP_RAMPDOWN, ///< error ramp_down NULL
    ERROR_STEPDRV_RAMP_RAMPUPDATA, ///< error ramp_up->rampdata NULL
    ERROR_STEPDRV_RAMP_RAMPDOWNDATA, ///< error ramp_down->rampdata NULL
    ERROR_STEPDRV_GO_STOP, ///< error stepper not STOPPED at go
    ERROR_STEPDRV_GO_STATE ///< error invalid initial state for go
};

enum ERROR_BARCODE {
    ERROR_BARCODE_RSS = ERROR_BASE_BARCODE, ///< error rendering RSS barcode
    ERROR_BARCODE_EAN,
    ERROR_BARCODE_ITF14,
    ERROR_BARCODE_CODE,
    ERROR_BARCODE_DATAMATRIX,
    ERROR_BARCODE_QR,
    ERROR_BARCODE_HANXIN,
};

enum ERROR_LABEL {
    ERROR_LABEL_OK_INCOMPLETE = ERROR_BASE_LABEL, ///< command is waiting for continued label data
    ERROR_LABEL_SHORT, ///< label received is too short to contain valid label data
    ERROR_LABEL_OOM, ///< label memory allocation error
    ERROR_LABEL_LONG, ///< extra data received beyond expected length of label
};
```

```

ERROR_LABEL_NOT_CONTIGUOUS, ///< label data offsets are not contiguous
ERROR_LABEL_PARSE, ///< a parse error occurred when processing the label
ERROR_LABEL_PARSER_UNKNOWN, ///< the label parser id was not understood
ERROR_LABEL_NOT_LOADED, ///< no label has been loaded
ERROR_LABEL_NOT_READY,
ERROR_LABEL_BOOT,
ERROR_LABEL_EXISTS,
ERROR_LABEL_NULL,
ERROR_LABEL_SERIALIZATION,
ERROR_LABEL_OFFSET,
ERROR_LABEL_FONT,
};

enum COMMAND_ERROR_ID {
    ERROR_COMMAND_INVALIDPAYLOAD = ERROR_BASE_CMD, ///< the payload contained incorrectly formatted data
    ERROR_COMMAND_NOT_READY, ///< the command would require the printer to be in the READY state, but it was not
    ERROR_COMMAND_REFUSED, ///< the command was refused and can not be processed
    ERROR_COMMAND_NOT_IDLE, ///< the command would require the printer to be in the IDLE state, but it was not
    ERROR_COMMAND_INVALID_KEY, ///< the key in a key-value based command was invalid
    ERROR_COMMAND_UNKNOWN_KEY, ///< the key in a key-value based command was not found
    ERROR_COMMAND_VALUE_OUT_OF_RANGE, ///< a value was out of range
    ERROR_COMMAND_PIC_UPDATE, ///< PIC_UPDATESTART, PIC_UPDATEPKT, PIC_UPDATEFINALIZE
    ERROR_COMMAND_SERIALIZEDBUFFER, ///< Serial buffer full
};

enum ERROR_APC {
    ERROR_APC_UNHANDLED_COMMAND = ERROR_BASE_APC ///< the command was not understood
};

enum ERROR_SERIAL_FLASH {
    ERROR_SERIAL_FLASH_GENERIC = ERROR_BASE_SERIAL_FLASH ///< generic serial flash error
};

enum ERROR_FONT_RENDER {
    ERROR_FONT_CRC = ERROR_BASE_FONT_RENDER, ///< the stored font CRC is incorrect
    ERROR_FONT_ALLOC, ///< not enough memory to load font
    ERROR_FONT_FONTFUSION,
    ERROR_FONT_SIZE
};

enum ERROR_HYST {
    ERROR_HYST_GENERIC = ERROR_BASE_HYST, ///< generic heating management error
    ERROR_HYST_CRCCMD, ///< heating management command checksum error
    ERROR_HYST_ERRCNT, ///< heating management internal controller error
    ERROR_HYST_SYSTEM, ///< heating management system error
    ERROR_HYST_CONFIGDATA, ///< heating management configuration error
    ERROR_HYST_VERSION, ///< heating management version mismatch error
    ERROR_HYST_NO_HC, ///< heating management error: no controller
    ERROR_HYST_NO_UNIT, ///< heating management error: no unit
    ERROR_HYST_DATARANGE, ///< heating management error: data range
    ERROR_HYST_CMD_INVALID, ///< heating management: invalid controller command
    ERROR_HYST_CRCFLASH, ///< heating management flash CRC error
    ERROR_HYST_DEFAULTS_SET, ///< heating management: defaults applied
    ERROR_HYST_NO_APPLICATION, ///< heating management: no firmware installed
    ERROR_HYST_NEWVALUE, ///< setting new value failed
    ERROR_HYST_AUTOCORRECTION ///< autocorrection applied for power correction
};

enum ERROR_MACHINE {
    ERROR_MACHINE_RANGE_LOW = ERROR_BASE_MACHINE, ///< machine configuration value too low
    ERROR_MACHINE_RANGE_HIGH, ///< machine configuration value too high
    ERROR_MACHINE_INVALIDTRANSITION, ///< machine can not transition into the requested state from the current state
    ERROR_MACHINE_CARRIAGE_LOST, ///< carriage was not found at home sensor when it should be
    ERROR_MACHINE_CASE_SENSOR, ///< machine cover sensor open
    ERROR_MACHINE_RIBBON_SENSOR, ///< a ribbon sensor is open
    ERROR_MACHINE_CARRIAGE_END, ///< carriage is at end sensor, please decrease the print length
    ERROR_MACHINE_PRINTSPEED, ///< print speed could not be applied
    ERROR_MACHINE_DENSITY, ///< density could not be applied
    ERROR_MACHINE_PRINTLENGTH,
    ERROR_MACHINE_PRESSURE_SENSOR, ///< no air pressure
};

```

```

ERROR_MACHINE_PRINTCYCLE_LOST, ///< print signal could not be applied
ERROR_MACHINE_RIBBON_SPOOL,
ERROR_MACHINE_DEFAULTS_APPLIED
};

enum ERROR_FPGA_FIRMWARE {
    ERROR_FPGA_FIRMWARE_OK_INCOMPLETE = ERROR_BASE_FPGA_FIRMWARE, ///< waiting for continued FPGA data
    ERROR_FPGA_FIRMWARE_LONG, ///< extra data received beyond expected length of FPGA
    ERROR_FPGA_FIRMWARE_OOM, ///< FPGA memory allocation error
    ERROR_FPGA_FIRMWARE_NOT_CONTIGUOUS, ///< FPGA data offsets are not contiguous
    ERROR_FPGA_FIRMWARE_SIZE, ///< FPGA size error
    ERROR_FPGA_FIRMWARE_CRC_MISMATCH, ///< FPGA CRC mismatch
    ERROR_FPGA_FIRMWARE_SAVE, ///< FPGA firmware could not be saved
    ERROR_FPGA_FIRMWARE_LOAD_FAILED, ///< FPGA did not load successfully
    ERROR_FPGA_FIRMWARE_NULL ///< FPGA null
};

enum ERROR_CONFIG
{
    ERROR_CONFIG_LOCALE = ERROR_BASE_CONFIG //Error setting locale
};

enum ERROR_NVRAM
{
    ERROR_NVRAM_VERSION_MISMATCH = ERROR_BASE_NVRAM, ///< NVRAM struct version mark mismatch
    ERROR_NVRAM_CRC_MISMATCH ///< NVRAM CRC error
};

enum ERROR_FPGA {
    ERROR_FPGA_RANGE_LOW = ERROR_BASE_FPGA, ///< FPGA value too low
    ERROR_FPGA_RANGE_HIGH ///< FPGA value too high
};

enum ERROR_FONT {
    ERROR_FONT_OK_INCOMPLETE = ERROR_BASE_FONT, ///< waiting for continued font data
    ERROR_FONT_LONG, ///< font data size too large
    ERROR_FONT_NOT_CONTIGUOUS, ///< font data offsets are not contiguous
    ERROR_FONT_OOM, ///< font memory allocation error
    ERROR_FONT_CRC_MISMATCH, ///< font CRC mismatch
    ERROR_FONT_SAVE, ///< font could not be saved
    ERROR_FONT_NULL, ///< font is null
    ERROR_FONT_USER, ///< user font load error from sd
    ERROR_FONT_TOO_BIG, ///< user font not matching into SF
    ERROR_FONT_FONTNUMBER ///< user font number wrong
};

enum ERROR_IIC {
    ERROR_IIC = ERROR_BASE_IIC ///< iic bus error
};

```

```
enum APC_ERRORS {
    APC_NO_ERROR = 0x0000,

    ///=== SD_card error codes ===
    APC_ERROR_OPENING_GPIO = 0x0001,
    APC_ERROR_SD_INIT = 0x0002,
    APC_ERROR_SD_PROTECTED = 0x0003,
    APC_ERROR_SD_MFS_INIT = 0x0004,
    APC_ERROR_SD_MFS_OPEN = 0x0005,
    APC_ERROR_SD_MFS_COMPATIBLE = 0x0006,
    APC_ERROR_SD_NOT_INSTALLED = 0x0007,
    APC_ERROR_SD_MFS_FORMAT_DIR = 0x0008,
    APC_ERROR_SD_UNINSTALLING_MFS = 0x0009,
    APC_ERROR_SD_FORMAT = 0x000A,
    APC_ERROR_SD_CREATE_DIR = 0x000B,
    APC_ERROR_SD_CHANGE_DIR = 0x000C,
    APC_ERROR_SD_ASSIGNING_MEMORY = 0x000D,
    APC_ERROR_SD_PATH = 0x000E,
    APC_ERROR_SD_FILE_IO = 0x000F,
    APC_ERROR_SD_CANNOT_OPEN_FILE = 0x0010,
    APC_ERROR_SD_DELETING_FILE = 0x0011,
    APC_ERROR_SD_FILE_EXISTS = 0x0012
}

typedef enum APC_WARNINGS {
    APC_WARNING_LOW_RIBBON = ERROR_BASE_WARNING,
    APC_ERROR_LOW_SERIALIZATION_BUFFER

enum COMMAND_ERROR_ID {
    CMD_NO_ERROR = 0x0000
    CMD_ERROR_INVALIDPAYLOAD = 0x1000 // the payload contained incorrect data (length, layout, unterminated string fields
    CMD_ERROR_NOT_READY = 0x0001 // the command would require the printer to be in the READY state, but it was not.
    CMD_ERROR_REFUSED = 0x000, // the command was refused; it cannot be processed at this time, usually because the printer
needs to be in a different state. refer to specification.
    CMD_ERROR_NOT_IDLE = 0x000 // the command would require the printer to be in the IDLE state, but it was not.

    CMD_ERROR_INVALID_KEY = 0x0001 // the key in a key-value based command was invalid
    CMD_ERROR_UNKNOWN_KEY = 0x0002 // the key in a key-value based command was not found
    CMD_ERROR_VALUE_OUT_OF_RANGE = 0x0006 // a value was out of range

    CMD_ERROR_FILE_STORAGE = 0x0001 // a generic error accessing the internal sd card with file operations
    CMD_ERROR_FONT_STORAGE = 0x0001 // a generic error accessing the internal sd card with font operations
}
```

8 Command IDs

```

ECHO = 0x0001,
GET_VERSION_INFORMATION = 0x0010,
SET_VERSION_INFORMATION = 0x0011,
SOFT_REBOOT = 0xABBA,
RETRIEVE_DIAGNOSTIC_LOG = 0x0020,
GET_DEVICE_STATUS = 0x0030,
GET_FREE_MEMORY = 0x0031,

DEVICE_INFO_GET = 0x0032,
DEVICE_INFO_LISTKEYS = 0x0033,
DEVICE_INFO_VALUETYPE = 0x0034,
GET_PRINthead_STATUS = 0x0035,

GET_PENDING_ERRORS = 0x0040,
RESET_ERRORS = 0x0041,
RIBBON_CLEAR = 0x0045,
RESET_RIBBON_COUNTER = 0x0046,
RESET_SERIALIZATION = 0x0047,
GET_PRINT_INFO = 0x0050,

FILE_STORAGE_PUT = 0x0070,
FILE_STORAGE_DELETE = 0x0071,
FILE_STORAGE_LIST = 0x0072,
FILE_STORAGE_FREE = 0x0073,
FILE_STORAGE_GET = 0x0074,
LIST_LABELS = 0x0075,
LABEL_EXISTS = 0x0076,

PRINT_TRIGGER = 0x0080,
STOP = 0x0090,
LABEL_LOAD = 0x00A0,
LABEL_POSTVIEW = 0x0202,

CONFIG_SET = 0x00B0,
CONFIG_GET = 0x00B1,
CONFIG_LISTKEYS = 0x00B2,
CONFIG_VALUETYPE = 0x00B3,

VARIABLE_SET = 0x00C0,
VARIABLE_GET = 0x00C1,
VARIABLE_LISTKEYS = 0x00C2,
VARIABLE_DELETE = 0x00C3,

FONT_STORAGE_PUT = 0x0088,
FONT_STORAGE_DELETE = 0x0081,
FONT_STORAGE_LIST = 0x0082,
FONT_STORAGE_GET = 0x0084,
LIST_FONTS = 0x0085,
FONT_EXISTS = 0x0086,
FONT_LOAD = 0x0087,
FONT_UPDATE = 0x00FA,
FONT_UPDATE_SELECTSLOT = 0x00FB,

PIC_UPDATESTART = 0x00F0,
PIC_UPDATEPKT = 0x00F1,
PIC_UPDATEFINALIZE = 0x00F2,
DIAGNOSTICS = 0x00F4,

FPGA_UPDATE = 0x00F3,

RESET_DEFAULTS = 0x00FF,
RAISENOTIFICATION = 0x1000,

```


9 Enumerations

```
enum MLI_BARCODE_TYPE
{
    QR_CODE = 0, //0
    DATAMATRIX, //1
    RSS14, //2
    RSS_EXP, //3
    RSS_LTD, //4
    RSS14STACK, //5
    RSS14STACK_OMNI, //6
    RSS_EXPSTACK, //7
    CODE128, //8
    EAN128, //9
    EAN14, //10
    EAN18, //11
    NVE18, //12
    EAN13, //13
    SSCC18, //14
    EAN8, //15
    ISBN, //16
    UPCA, //17
    UPCE, //18
    ITF14, //19
    CODE11, //20
    CODE39, //21
    CODE93 //22
}

@OPTION_3 for DATAMATRIX":
DM_SQUARE=100;
DM_DMRE = 101;

enum COUNTERTYPE {
    COUNTER_VARIANT_LEADING_DIGITS_SPACES = 0,
    COUNTER_VARIANT_VALUE_CUT_TO_LENGTH,
    COUNTER_VARIANT_LEADING_DIGITS_ZERO
};
```

```
enum DATETYPE {
DATE_VARIANT_DAY = 0,
    DATE_VARIANT_DAY_WITHLEADINGZERO,
    DATE_VARIANT_MONTH,
    DATE_VARIANT_MONTH_WITHLEADINGZERO,
    DATE_VARIANT_MONTHNAME_3CHARACTERS,
    DATE_VARIANT_MONTHNAME,
    DATE_VARIANT_YEAR_1_2CHARS,
    DATE_VARIANT_YEAR_1CHAR,
    DATE_VARIANT_YEAR_WITHLEADINGZERO_2CHARS,
    DATE_VARIANT_YEAR_4CHARS,
    DATE_VARIANT_DAYCODE,
    DATE_VARIANT_DAYCODE_WITHLEADINGZERO_3CHARS,
    DATE_VARIANT_DAYCODE_3CHARS_29FEB_366,
    DATE_VARIANT_CALENDARWEEK,
    DATE_VARIANT_CALENDARWEEK_WITHLEADINGZERO,
    DATE_VARIANT_DAYOFWEEK_FIRSTDAY_MONDAY,
    DATE_VARIANT_DAYOFWEEK_FIRSTDAY_SUNDAY,
    DATE_VARIANT_DAYOFWEEK_3CHARS,
    DATE_VARIANT_DAYOFWEEK,
    DATE_VARIANT_HOUR24_WITHLEADINGZERO,
    DATE_VARIANT_HOUR24,
    DATE_VARIANT_HOUR12_WITHLEADINGZERO,
    DATE_VARIANT_HOUR12,
    DATE_VARIANT_MINUTE_WITHLEADINGZERO,
    DATE_VARIANT_MINUTE,
    DATE_VARIANT_SECOND_WITHLEADINGZERO,
    DATE_VARIANT_SECOND,
    DATE_VARIANT_AMPM,
    DATE_VARIANT_AMPM_WITHSPACE,
    DATE_VARIANT_MM_DD_YY,
    DATE_VARIANT_HH_MM,
    DATE_VARIANT_HH_MM_SS
};

// 0= [D:1] 1 Day
// 1= [D:2] 01 DayWithLeadingZero (2 chars)
// 2= [M:1] 1 Month
// 3= [M:2] 01 MonthWithLeadingZero
// 4= [M:3] JAN MonthName (3characters)
// 5= [M:4] JANUAR MonthName
// 6= [A:0] 15 Year (1-2 chars)
// 7= [A:1] 9 Year (1 char)
// 8= [A:2] 05 YearWithLeadingZero (2 chars)
// 9= [A:4] 2020 Year (4 chars)
// 10= [J:1] 1 DayCode (1-3 chars)
// 11= [J:3] 001 DayCodeWithLeadingZero (3 chars)
// 12= [J:4] 366 DayCode (3 chars, 29 Feb = 366))
// 13= [W:1] 1 CalendarWeek
// 14= [W:2] 01 CalendarWeekWithLeadingZero
// 15= [L:1] 1 DayOfWeek (first day monday)
// 16= [L:2] 1 DayOfWeek (first day sunday)
// 17= [L:3] MON DayOfWeek (3 chars)
// 18= [L:4] MONDAY DayOfWeek
// 19= [H:1] 01 Hour24WithLeadingZero
// 20= [H:2] 24 Hour24
// 21= [H:3] 01 Hour12WithLeadingZero
// 22= [H:4] 12 Hour12WithLeadingZero
// 23= [MM:1] 01 MinuteWithLeadingZero
// 24= [MM:2] 1 Minute
// 25= [S:1] 01 SecondWithLeadingZero
// 26= [S:2] 1 Second
// 27= [T:1] AM AMPM
// 28= [T:2] AM AMPM with space
```

10 Label definition example

```

MLi00
#
# Label name: Varcnt.mli
# Renderer: a:design2 version 4.2.50826
# generated 9/21/2015 2:04 PM
#
SET RIBBON_SPOOL_DISTANCE_FULL=4104
SET RIBBON_SPOOL_DISTANCE_PARTIAL=353
SET RIBBON_USE=1
#
#VAR_COUNTER @NAME @COUNTERTYPE @START @END @STEP @INITIALVALUE @REPEATS @PADLENGTH
VAR_COUNTER "COUNTER1" 1 0 1 1000 1 0 4
#
#VAR_TEXT @NAME @CONTENT
VAR_TEXT "Variable1" "123"
#
#VAR_DATE @NAME @DATETYPE @OFFEST_DAYS @OFFSET_MONTHS @OFFSET_YEARS
VAR_DATE "DATE1_1" 0 0 0 0
VAR_DATE "DATE1_2" 3 0 0 0
VAR_DATE "DATE1_3" 9 0 0 0
#
#TXT @HEAD @X @Y @SIZE_PT @WIDTH_PCT @BEHAVIOR @CONTENT
#      @BEHAVIOR
#      0 = printdirection normal
#      1 = printdirection 90°
#      2 = printdirection 180°
#      3 = printdirection 270°
#      4 = printdirection normal, inverted
#      5 = printdirection 90°, inverted
#      6 = printdirection 180°, inverted
#      7 = printdirection 270°, inverted

TXT * 36 84 10 100 0 "Text1"
TXT * 373 87 10 100 0 "{COUNTER1}"
TXT * 198 88 10 100 0 "{Variable1}"
TXT * 210 204 10 100 0 "{Variable1}{COUNTER1}"
TXT * 361 202 10 100 0 "{DATE1_1}.{DATE1_2}.{DATE1_3}"

```

```
#BARCODE @HEAD @X @Y @BARCODETYPE @ORIENTATION @MODULE_SIZE_PT @HEIGHT_PT @OPTION_0
@OPTION_1 @OPTION_2 @INVERTED @INPUTMODE @TXT_ORIENTATION @TXT_SIZE_PT @BORDER_WIDTH_PT
@CONTENT
```

```
# @BARCODETYPE
# QR_CODE = 0
# DATAMATRIX = 1
# RSS14, = 2
# RSS_EXP = 3
# RSS_LTD = 4
# RSS14STACK = 5
# RSS14STACK_OMNI = 6
# RSS_EXPSTACK = 7
# CODE128 = 8
# EAN128 = 9
# EAN14 = 10
# EAN18 = 11
# NVE18 = 12
# EAN13 = 13
# SSCC18 = 14
# EAN8 =15
# ISBN = 16
# UPCA = 17
# UPCE = 18
# ITF14 = 19
# CODE11 = 20
# CODE39 = 21
# CODE93 = 22
# CODE2_5 = 23
# HANXIN = 24
```

```
# @ORIENTATION
#      0 = printdirection normal
#      1 = printdirection 90°
#      2 = printdirection 180°
#      3 = printdirection 270°
```

```
# @INVERTED
#      0 = normal
#      1 = inverted
```

```
@INPUTMODE
#      0 = normal
#      1 = unicode
#      2 = GS1
#      3 = KANJI
#      4 = SJIS
```

```
# @BORDER_WIDTH_PT
#      0 = no border
```

```
BARCODE * 102 62 15 0 4 49 2 0 0 0 1 1 10 0 "{COUNTER1}"
```

```
#QR BARCODE
# @OPTION_1 ECC-Level: 1=L, 2=M, 3=Q, 4=H
# @OPTION_2 size: 1-40
```

```
# Auto //0
# 21x21 //1
# 25x25 //2
# 29x29 //3
# 33x33 //4
# 37x37 //5
# 41x41 //6
# 45x45 //7
# 49x49 //8
# 53x53 //9
# 57x57 //10
# 61x61 //11
# 65x65 //12
# 69x69 //13
# 73x73 //14
# 77x77 //15
# 81x81 //16
# 85x85 //17
# 89x89 //18
# 93x93 //19
# 97x97 //20
# 101x101 //21
# 105x105 //22
# 109x109 //23
# 113x113 //24
# 117x117 //25
# 121x121 //26
# 125x125 //27
# 129x129 //28
# 133x133 //29
# 137x137 //30
# 141x141 //31
# 145x145 //32
# 149x149 //33
# 153x153 //34
# 157x157 //35
# 161x161 //36
# 165x165 //37
# 169x169 //38
# 173x173 //39
# 177x177 //40
BARCODE * 10 0 0 0 6 105 2 1 1 0 1 1 15 0 "1111111111111"

#GS1 SQUARE DATAMATRIX BARCODE
# @OPTION_1 MODE: 0=square, 1=rectangle
# @OPTION_2 size: 1-42
#1: 10x10 , 3*/1, /*
#2: 12x12 , 5*/3, /*
#3: 14x14 , 8*/5, /*
#4: 16x16 , 12*/7, /*
#5: 18x18 , 18*/9, /*
#6: 20x20 , 22*/12, /*
#7: 22x22 , 30*/14, /*
#8: 24x24 , 36*/16, /*
#9: 26x26 , 44*/21, /*
#10: 32x32 , 62*/25, /*
```

```
#11: 36x36 , 86*/28, /*
#12: 40x40 ,114*/30, /*
#13: 44x44 ,144*/31, /*
#14: 48x48 ,174*/32, /*
#15: 52x52 ,204*/33, /*
#16: 64x64 ,280*/34, /*
#17: 72x72 ,368*/35, /*
#18: 80x80 ,456*/36, /*
#19: 88x88 ,576*/37, /*
#20: 96x96 ,696*/38, /*
#21:104x104,816*/39, /*
#22:120x120,1050*/40, /*
#23:132x132,1304*/41, /*
#24:144x144,1558*/2, /*
#25: 8x18 , 5*/4, /*
#26: 8x32 , 10*/6, /*
#27: 12x26 , 16*/10, /*
#28: 12x36 , 22*/13, /*
#29: 16x36 , 32*/17, /*
#30: 16x48 , 49*/8, /*
#31: 8x48 , 18*/11, /*
#32: 8x64 , 24*/15, /*
#33: 12x64 , 43*/22, /*
#34: 16x64 , 62*/18, /*
#35: 24x32 , 49*/20, /*
#36: 24x36 , 55*/24, /*
#37: 24x48 , 80*/27, /*
#38: 24x64 ,108*/19, /*
#39: 26x32 , 52*/23, /*
#40: 26x40 , 70*/26, /*
#41: 26x48 , 90*/29, /*
#42: 26x64 ,118*/0
BARCODE * 307 294 1 0 7 92 2 0 1 100 2 1 5 100 "[01]12345678901234[12]121212"
```

```
#GS1 HanXin BARCODE
# @OPTION_1 MODE: 0=square
# @OPTION_2 size: 1-88
# "23x23" : 1
# "25x25" : 2
# "27x27" : 3
# "29x29" : 4
# "31x31" : 5
# "33x33" : 6
# "35x35" : 7
# "37x37" : 8
# "39x39" : 9
# "41x41" : 10
# "43x43" : 11
# "45x45" : 12
# "47x47" : 13
# "49x49" : 14
# "51x51" : 15
# "53x53" : 16
# "55x55" : 17
# "57x57" : 18
# "59x59" : 19
# "61x61" : 20
```

"63x63" : 21
"65x65" : 22
"67x67" : 23
"69x69" : 24
"71x71" : 25
"73x73" : 26
"75x75" : 27
"77x77" : 28
"79x79" : 29
"81x81" : 30
"83x83" : 31
"85x85" : 32
"87x87" : 33
"89x89" : 34
"91x91" : 35
"93x93" : 36
"95x95" : 37
"97x97" : 38
"99x99" : 39
"101x101" : 40
"103x103" : 41
"105x105" : 42
"107x107" : 43
"109x109" : 44
"111x111" : 45
"113x113" : 46
"115x115" : 47
"117x117" : 48
"119x119" : 49
"121x121" : 50
"123x123" : 51
"125x125" : 52
"127x127" : 53
"129x129" : 54
"131x131" : 55
"133x133" : 56
"135x135" : 57
"137x137" : 58
"139x139" : 59
"141x141" : 60
"143x143" : 61
"145x145" : 62
"147x147" : 63
"149x149" : 64
"151x151" : 65
"153x153" : 66
"155x155" : 67
"157x157" : 68
"159x159" : 69
"161x161" : 70
"163x163" : 71
"165x165" : 72
"167x167" : 73
"169x169" : 74
"171x171" : 75
"173x173" : 76

```
# "175x175" : 77
# "177x177" : 78
# "179x179" : 79
# "181x181" : 80
# "183x183" : 81
# "185x185" : 82
# "187x187" : 83
# "189x189" : 84
# "191x191" : 85
# "193x193" : 86
# "195x195" : 87
# "197x197" : 88
BARCODE * 262 76 24 0 6 175 0 1 7 0 1 1 15 0 "123123"
```

```
#GFX @HEAD @X @Y @HEIGHT @WIDTH @TRANSPARENT @BASE64MONO1PPIMAGEDATA"
GFX * 457 44 53 24 1 AAAAAAAAAAAAAAAAAAAAAAAAAAAf-
gAD/8AP/+Af/+A/gcB+AED8AAD4AAHwAAHwAAPgAAPgAA///4///w///wPAAAPAAA///w///g///gPgAAHgAAHwAAHwAA
D4AAD8AAB+AEA/gcAf/8AP/8AD/8AAfgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
```

```
#VAR_CAL @NAME @TYPE @OFFEST_DAYS @OFFSET_MONTHS @OFFSET_YEARS "@TABLE"
```

MLi01

```
#VAR_DATE @NAME @DATETYPE @OFFSET_HOURS @OFFEST_DAYS @OFFSET_MONTHS @OFFSET_YEARS
```

```
#VAR_DATE "DATE1_1" 0 0 0 0 0
```

```
#VAR_DATE "DATE1_2" 3 0 0 0 0
```

```
#VAR_DATE "DATE1_3" 9 0 0 0 0
```

```
#VAR_CAL @NAME @TYPE @OFFEST_DAYS @OFFSET_MONTHS @OFFSET_YEARS "@TABLE"
```

```
VAR_CAL "DATECAL1_1" 5 6 0 0 0 "A\B\C\D\E\F\G\H\I\J\K\L\M\N\O\P\Q\R\S\T\U\V\W\X\Y\Z"
```

```
TXT * 396 91 10 100 0 "Schicht"
```

```
TXT * 392 180 10 100 0 "{DATECAL1_1}"
```