Printer Utility DLL
RFID SUPPLEMENT

(PADLL.DLL & PADLL.H)

Practical Automation Inc.

Version 2.0.0.9 Rev B

PRACTICAL AUTOMATION, INC.
The Alinabal Group of Companies
45 Woodmont Road, P.O. Box 3028, Milford, CT 06460 ▪ Phone: (203) 882-5640 ▪ FAX: (203) 882-5648
www.practicalautomation.com

Table of Contents

Document and DLL Release History

DLL Version	Document Revision	Date	Name	Comments
V2.0.0.3	DRAFT A	2017-12-05	WDW	Created for printers that support RFID and paLMRFIDPrinter
V2.0.0.3	A	2017-12-15	WDW	Adding extra information to Region Parameter Adjusted box colours to match PA Updated timeout note on paLMRFIDPrinter
V2.0.0.3	B	2017-12-17	GLB	Add Query commands and responses
V2.0.0.3	C	2017-12-27	GLB	Add Parameter read and write power elements
V2.0.0.3	D	2018-01-04	GLB	Add TID information
V2.0.0.3	E	2018-03-09	GLB	Add RFID_WRITE_OVERFLOW return code
V2.0.0.3	F	2018-03-21	GLB	Add AFI parameter; add Monza 5 TID note; add new regions.
V2.0.0.5	A	2018-09-20	WDW	Added paGetDLLVersion to dll
V2.0.0.9	A	2018-10-16	WDW	Added paLMPtrRFIDPrinter
V2.0.0.9	B	2019-02-13	GLB	Added RU and SG to Regions table; Correct JP lower limit

OVERVIEW

All Printer Utility DLL (later referred to as DLL) function calls, unless otherwise specified are listed in "Printer Utility DLL Programer's Application Notes" (AN32.LL9.PDF)

This is a supplement document and does not include all the APIs available in the DLL. Instead, this document provides the API commands used with the paLMRFIDPrinter function call provided by the PADLL.DLL dll.

RFID commands are only supported on some devices. At the time of writing, the ITK22G (RFID) is the only device that supports these commands. Check with PA before using these functions as not all devices will support them.

FUNCTION RETURN CODES

```
API functions that return a long return code will return
one of the following constants. These constants are defined in padll.h (C and
C++)

PA_ERR_OK                      = Function completed without error

PA_ERR_PRINTER_NOT_RESPONDING = Printer is not responding to status
                                This is generally due to the printer
                                being powered off, the printer
                                being disconnected or the wrong port
                                selected in the printers panel.

PA_ERR_PIPE_OPEN_FAILED        = Status connection between the application
                                And the printer could not be opened.
                                This is generally due an attempt to communicate
                                with a printer other than the default printer.
                                In this case use paSetNewPrinter to point PADLL
                                Functions to the correct non-default printer.

PA_XXX                         = Return codes specific to a given function.
                                Please refer to individual API for details.
```


PADLL.DLL API

Only a subset of commands are listed here. For full reference refer to AN32nLL9.PDF. At the time of writing, the ITK22G (RFID) is the only device that supports these commands. Check with PA before using these functions as not all devices will support them.

paLMPtrRFIDPrinter

Our RFID support can be treated as a transparent pass through pipe. The commands that the printer supports being received through this pipe are documented in the AN32nLL9_RFID_SUPPLEMENT.PDF document. This function is the same as paLMRFIDPrinter, except that it removes the need to call paSetNewPrinter first. And aids in communicating with multiple printers from different threads.

```
LRESULT APIENTRY paLMPtrRFIDPrinter(char *pszPrinterName, char *pDataToWrite,
DWORD lNoBytesToWrite, char *pResponseBytes, DWORD lMaxResponseLen, DWORD
*plNoBytesRead, DWORD lTimeout);

Declare Function paLMPtrRFIDPrinter Lib "PADLL" (ByVal pszPrinterName As
String, ByVal DataToWrite As String, ByVal NoBytesToWrite As Long, ByRef
ResponseBytes As Byte, ByVal MaxResponseBytesLen As Long, ByRef NoBytesRead As
Long, ByVal Timeout As Long) As Long
```

Parameters

pszPrinterName[in]
Pointer to null terminated printer name.

pCommandBytes
Pointer to a byte buffer that contains the RFID command that needs to be sent.

lLen
The length of the command supplied in pCommandBytes. The maximum length supported in 1024 bytes.

pResponseBytes
Pointer to a byte buffer to receive the response.

lMaxResponseLen
The length of the response buffer supplied in pResponseBytes. The maximum length supported in 1024 bytes.

plResponseLen
Pointer to a variable that contains the length of the response returned.

lTimeout (FOR FUTURE USE)
The maximum length of time that call should wait for a response from the printer.

Return Values

PA_ERR_OK - Successful


```
PA_ERR_PARAM1            - Error with pCommandBytes
PA_ERR_PARAM2            - Error with lLen
PA_ERR_PARAM3            - Error with pResponseBytes
PA_ERR_PARAM4            - Error with lMaxResponseLen or plResponseLen
PA_ERR_PIPE_OPEN_BUSY    - Error opening the pipe/connection to LM
PA_ERR_PIPE_OPEN_FAILED  - Error opening the pipe/connection to LM
PA_ERR_SYSTEM            - Error reading or writing to the pipe/LM
PA_ERR_UNKNOWN_PRINTER   - LM does not support this function
PA_ERR_PIPE_TIMEOUT      - Timeout reading from LM
PA_ERR_PRINTER_NOT_RESPONDING – Printer not responding
```

.NET Support

C#,VB.NET example included in CSGetStatusOnlyLM.zip and VBGetStatusOnlyLM.zip. (Practical Automation toolkit).

paLMRFIDPrinter

Our RFID support can be treated as a transparent pass through pipe. The commands that the printer supports being received through this pipe are documented in this document.

```
LRESULT APIENTRY paLMRFIDPrinter(char *pDataToWrite, DWORD lNoBytesToWrite,
char *pResponseBytes, DWORD lMaxResponseLen, DWORD *plNoBytesRead, DWORD
lTimeout);

Declare Function paLMRFIDPrinter Lib "PADLL" (ByVal DataToWrite As String,
ByVal NoBytesToWrite As Long, ByRef ResponseBytes As Byte, ByVal
MaxResponseBytesLen As Long, ByRef NoBytesRead As Long, ByVal Timeout As Long)
As Long
```

Parameters

pCommandBytes
Pointer to a byte buffer that contains the RFID command that needs to be sent.

lLen
The length of the command supplied in pCommandBytes. The maximum length supported in 1024 bytes.

pResponseBytes
Pointer to a byte buffer to receive the response.

lMaxResponseLen
The length of the response buffer supplied in pResponseBytes. The maximum length supported in 1024 bytes.

plResponseLen
Pointer to a variable that contains the length of the response returned.

lTimeout (FOR FUTURE USE)
The maximum length of time that call should wait for a response from the printer.

Return Values

```
PA_ERR_OK                - Successful
PA_ERR_PARAM1            - Error with pCommandBytes
PA_ERR_PARAM2            - Error with lLen
PA_ERR_PARAM3            - Error with pResponseBytes
PA_ERR_PARAM4            - Error with lMaxResponseLen or plResponseLen
PA_ERR_PIPE_OPEN_BUSY    - Error opening the pipe/connection to LM
PA_ERR_PIPE_OPEN_FAILED  - Error opening the pipe/connection to LM
PA_ERR_SYSTEM            - Error reading or writing to the pipe/LM
PA_ERR_UNKNOWN_PRINTER   - LM does not support this function
PA_ERR_PIPE_TIMEOUT      - Timeout reading from LM
PA_ERR_PRINTER_NOT_RESPONDING – Printer not responding
```


COMMAND FORMAT

These commands are to be sent to the printer using the paLMRFIDPrinter API call through the PADLL.DLL.

These commands expose the functionality of the internal printer RFID API to the printer host. The driver and the PADLL transparently pass these packets between the printer and the host. In the following descriptions, the syntax delineators for visual clarity are not actually sent on the line:

Packet format, host to printer:

```
<esc>RF<RFID command character><length in ASCII Digits>{,<length bytes of data>}
```

Packet format, printer to host:

```
<esc>rf<RFID response character><length in ASCII Digits>{,<length bytes of data>}
```

Parameters

The contents between angle brackets '<>' are a description of the data field they represent. For example, "<esc>" represents the ASCII Escape character, 0x1B. Spaces are for visual clarity of this description and are not part of the protocol.

Depending on command requirements, the data itself may be subdivided into command dependent fields. A command or response with no data should encode the length as '0' (000). If the length is non-zero, a comma must follow to delimit the length from parameter/response data. The curly braces '{' and '}' in the above packet formats indicate the optional data for non-zero lengths.

<esc>
: The ASCII Escape character, 0x1B.

RF
: The RFID command header that is sent. This must always be "RF".

rf
: The RFID command header for data that is received from the printer. This must always be "rf".

<RFID command character>
: The command for this message. This is a single character that is defined as follows:

Command	Description
S	Read Stock Status
W	Write tag data
C	Cancel current write
R	Read tag
M	Move tag
P	Parameter set request for RFID reader parameters
Q	Query reader data

<*RFID response character*>

The response for this message. Generally, the response will be a lower case version of the command that was used when the message was transmitted to the printer. This is a single character that is defined as follows:

Command	Description
s	Stock Status –a single digit ASCII status code
w	Status of current write command
r	Status of current Read tag
m	Result of the move tag operation
p	Parameter set response
q	Query response
x	Unimplemented command

<*length in ASCII Digits*>

The length of ASCII data provided with this command. This length must always be formatted as 3 ASCII digits. So a length of 0 would be encoded as "000". If this value is greater than 0, then there must be a comma "," separating this field from the actual data itself.

<*length bytes of data*>

This is the data sent. And must be the length specified in <length in ASCII Digits>.

COMMANDS

NOTE

ONLY ONE READ/WRITE/SET PARAMETER COMMAND CAN BE PROCESSED BY THE PRINTER AT A TIME. IF A COMMAND IS SENT TO READ/WRITE/SET PARAMETER, THEN THE SAME COMMAND CAN BE SENT WITH A "000" LENGTH TO QUERY THE PROGRESS OF THAT COMMAND. ONCE COMPLETE, THEN THE NEXT READ/WRITE/SET PARAMETER COMMAND CAN BE SENT.

Read Stock Status (S)

This command is used to query the RFID stock status of the printer. For this command, the length is always "000" when requesting status, and the response length is always '001'.

Packet format, host to printer:

```
<esc>RFS000
```

Packet format, printer to host:

```
<esc>rfs001,<status byte>
```

Parameters

<esc>

The ASCII Escape character, 0x1B.

<status byte>

RFID stock status as defined.

Status	Value	Description
RFID_READER_NOT _DETERMINED	'0'	Reader presence unknown
RFID_NO_READER	'1'	No reader installed
RFID_READER_NON_FUNCTIONAL	'2'	Reader installed, but not working
RFID_NO_STOCK	'3'	Reader present, stock is out of paper
RFID_TAGLESS_STOCK	'4'	Non-RFID stock present
RFID_STOCK_LOADED	'5'	RFID stock present, but not ready
RFID_TAG_READY	'6'	RFID Tag in position for reading/writing
RFID_TAG_FAULTY	'7'	Cannot read tag which should be there

Write tag data (W)

This command is used to write data to the tag. A write request with a length of "000" is a write status request.

Packet format, host to printer:

```
<esc>RFW<length in ASCII Digits>,<memory bank>,<data offset>,<length user data><user data bytes>

<esc>RFW000
```

Packet format, printer to host:

```
<esc>rfw001,<status byte>
```

Parameters

<esc>

The ASCII Escape character, 0x1B.

<length in ASCII Digits>

The length of parameter data provided with this command. This length must always be formatted as 3 ASCII digits. So a length of 0 would be encoded as "000". If this value is greater than 0, then there must be a comma "," separating this field from the actual data itself.

<memory bank>

The specific memory bank to write to.

Value	Bank	Description
E	01	EPC (Electronic Product Code) Memory bank
U	11	User Memory bank

<length epc data>

The length of EPC (Electronic Product Code) data supplied.

<epc data bytes>

Data to write to the EPC (Electronic Product Code) memory block. Must be <length epc data>.

<data offset>

Offset to write data to.

<length user data>

The length of user data supplied.

<user data bytes>

Data to write to the user memory block. Must be <length user data>.

<status byte>

The status of the write operation.

Status	Value	Description
RFID_SUCCESS	'0'	Operation occurred successfully
RFID_BUSY	'1'	Operation in progress
RFID_FAILURE	'2'	Failed
RFID_NO_STATUS	'5'	No operation is in process for this query
RFID_INVALID_COMMAND	'6'	Command and/or parameters invalid
RFID_WRITE_OVERFLOW	'7'	Attempt to write more data than bank can hold. No data written to tag.

NOTE

RFID_SUCCESS, RFID_FAILURE, RFID_INVALID_COMMAND, AND RFID_WRITE_OVERFLOW ARE FINAL STATUSES. SUBSEQUENT WRITE STATUS QUERIES WILL RETURN RFID_NO_STATUS UNTIL A NEW COMMAND IS STARTED.

RFW000

Query the last write operation to get status.

Example:

This example attempts to write the bytes ("0123456789") to the EPC memory bank.

```
<esc>RFW017,E,0,10,0123456789
```


Cancel current write (C)

This command is used to cancel the last write command. For this command, the command length is always "000", and the response length is always '001'.

Packet format, host to printer:

```
<esc>RFC000
```

Packet format, printer to host:

```
<esc>rfc001,<status byte>
```

Parameters

<esc>

The ASCII Escape character, 0x1B.

<status byte>

RFID stock status as defined.

Status	Value	Description
RFID_CANCELLING	'3'	A cancel command is in process
RFID_CANCELLED	'4'	A cancel command is complete

Read tag data (R)

This command is used to read data from the tag. For this command, the length is always "000" when requesting status, and the response length is always '001' while the operation is in progress.

Packet format, host to printer:

```
<esc>RFR<length in ASCII Digits>,<memory bank>,<offset>,<length to read>

<esc>RFR000
```

Packet format, printer to host:

```
<esc>rfr001,<status byte>

<esc>rfr<length in ASCII Digits>,<status byte>,<memory bank>,<offset>,<length
data returned>,<data returned>
```

Parameters

<esc>

The ASCII Escape character, 0x1B.

<length in ASCII Digits>

The length of parameter data provided with this command. This length must always be formatted as 3 ASCII digits. So a length of 0 would be encoded as "000". If this value is greater than 0, then there must be a comma "," separating this field from the actual data itself.

<memory bank>

The specific memory bank to read from.

Value	Bank	Description
E	01	EPC (Electronic Product Code) Memory bank
T	10	TID (Tag Identifier) Memory bank
U	11	User Memory bank

<status byte>

The status of the read operation.

Status	Value	Description
RFID_SUCCESS	'0'	Operation occurred successfully
RFID_BUSY	'1'	Operation in progress
RFID_FAILURE	'2'	Failed
RFID_NO_STATUS	'5'	No operation is in process for this query
RFID_INVALID_COMMAND	'6'	Command and/or parameters invalid

RFR000

Query the last read operation to get status. Normally used if RFID_BUSY was returned.

<offset>
Offset for data to read in the memory bank.

<length to read>
The maximum length of data to read.

<length data returned>
Length of the data returned in <data returned>.

<data returned>
Data bytes returned. Will be <length to read> in length, or length of data read, whichever is shorter.

Examples:

This example attempts to read up to 128 bytes from the EPC memory bank.

```
<esc>RFR007,E,0,128
```

This example attempts to read 4 bytes from the TID memory bank.

```
<esc>RFR005,t,0,4
```

Special Considerations for TID:

The TID bank is read only and factory programmed by the tag chip manufacturer. An attempt to read more data than the chip has will fail, either with an error or with blank data. The Gen2 specification only guarantees that a minimum of 4 bytes be readable. These include the object identifier (8 bits), the Vendor ID (12 bits), and the device ID (12 bits). This information identifies the tag manufacturer and type. If the user application knows that, and knows how much can be read and needs that additional information, it can safely request it.

Alternately, the user could use Gen2 specification information to decode the tag per the specification. For example:

A 4 byte read returns (in hex) E2 80 11 30. This breaks down as:
E2 – Class identifier, usually E2 or E0.
801 – Vendor ID for Impinj, with MSB set indicating Extended ID information is available
130 – Device ID indicating the chip is a Monza 5 model.

With this information, if the application knows of, and expects a Monza 5, it knows it can read a total of 12 bytes. Let us follow the example a bit further.

If the MSB of the VID is set, as in the above example, there must be 2 more bytes available, known as the XTID segment, which describe the extended TID data. A read of 6 bytes provided the above plus (hex) 20 00.
Using the information in the Gen 2 specification we find that this means there is a 48-bit (6byte) unique identifier, a chip serial number if you will, available. Therefore we can read 12 bytes, the last 6 being a unique serial number when use in conjunction with the Vendor ID and Device ID.

The reference document for Gen 2 tag memory layout is:
https://www.gs1.org/sites/default/files/docs/epc/uhfc1g2_2_0_0_standard_20131101.pdf

Move tag (M)

The tag will be moved forward (+) or backward (-) unless such a move would cause the tag to lose contact with the drive rollers. A move request with a length of zero ("000") is a move status request. This is legal whenever the printer is idle and ready.

Packet format, host to printer:

```
<esc>RFM<length in ASCII Digits>,<distance to move>

<esc>RFM000
```

Packet format, printer to host:

```
<esc>rfm001,<status byte>
```

Parameters

<esc>

The ASCII Escape character, 0x1B.

<length in ASCII Digits>

The length of ASCII data provided with this command. This length must always be formatted as 3 ASCII digits. So a length of 0 would be encoded as "000". If this value is greater than 0, then there must be a comma "," separating this field from the actual data itself.

<status byte>

The status of the operation.

Status	Value	Description
RFID_SUCCESS	'0'	Operation occurred successfully
RFID_BUSY	'1'	Operation in progress
RFID_FAILURE	'2'	Failed

Parameter set request (P)

Set the parameter data. If busy is returned because the response is not immediately available, a request with a length of "000" can be sent to request the response.

Packet format, host to printer:

```
<esc>RFP<length in ASCII Digits>,<parameter identifier>,<parameter data>
```

Packet format, printer to host:

```
<esc>rfp001,<status byte>
```

Parameters

<esc>

The ASCII Escape character, 0x1B.

<length in ASCII Digits>

The length of ASCII data provided with this command. This length must always be formatted as 3 ASCII digits. So a length of 0 would be encoded as "000". If this value is greater than 0, then there must be a comma "," separating this field from the actual data itself.

NOTE

THIS COMMAND SHOULD NOT BE SENT IF STOCK STATUS IS READER_NOT_DETERMINED OR NO_READER, OR IF ANOTHER RFID FUNCTION IS IN PROGRESS.

<parameter identifier>

The status of the read operation.

ID	Description
R	Configure the region
P	Configure Read Power
W	Configure Write Power
F	Set a new EPC Application Family Identifier
X	Reboot the RFID reader

<parameter data>

Regions available in reference section at "Definition of region parameters".
Power is a 1 to 3 digit number specifying the power level in centi-dBm. For reference, 0dBm is 1mw, the minimum setting. The default is 300, or 3.00dBm (2mw) The maximum is 999, or 9.99dBm, about 10mw.

<status byte>

The status of the operation.

Status	Value	Description
RFID_SUCCESS	'0'	Operation occurred successfully. The set parameter is the current parameter.
RFID_BUSY	'1'	Operation in progress. The command was forwarded to the reader for processing. Poll with the parameter set query for the operations status.
RFID_FAILURE	'2'	Failed
RFID_INVALID_COMMAND	'6'	Command failed parsing. This can be caused by an invalid region selector string. The supplied region selector string must be supported by the installed reader.

Parameter query request (Q)

Request the data from a specific parameter. If busy is returned because the response is not immediately available, a request with a length of "000" can be sent to request the response.

Packet format, host to printer:

```
<esc>RFQ<length in ASCII Digits>,<parameter identifier>

<esc>RFQ000
```

Packet format, printer to host:

```
<esc>rfq<length in ASCII Digits>,<status byte>,<parameter
identifier>,<parameter data>
```

Parameters

<esc>

The ASCII Escape character, 0x1B.

<length in ASCII Digits>

The length of ASCII data provided with this command. This length must always be formatted as 3 ASCII digits. So a length of 0 would be encoded as "000". If this value is greater than 0, then there must be a comma "," separating this field from the actual data itself.

<parameter identifier>

The status of the read operation.

ID	Description
R	Report the current region setting.
A	Report the RFID Middleware API version
N	Report the RFID Reader firmware version
P	Report the RFID Read power level, centi-dBm
W	Report the RFID Write power level, centi-dBm
F	Report the current EPC Application Family Identifier

<parameter data>

Regions available in reference section at "Definition of region parameters".

<status byte>

The status of the operation.

Status	Value	Description
RFID_SUCCESS	'0'	Query operation completed successfully
RFID_BUSY	'1'	Query operation in progress
RFID_FAILURE	'2'	Query failed

REFERENCE

Definition of return codes returned from API functions

STATUS BYTES

Status	Value	Description
RFID_SUCCESS	'0'	Read operation occurred successfully
RFID_BUSY	'1'	Read operation in progress
RFID_FAILURE	'2'	Read failed
RFID_CANCELLING	'3'	A cancel command is in process
RFID_CANCELLED	'4'	A cancel command is complete
RFID_NO_STATUS	'5'	No operation is in process for this query
RFID_INVALID_COMMAND	'6'	Command and/or parameters invalid
RFID_WRITE_OVERFLOW	'7'	Attempt to write more data than bank can hold. No data written to tag.

Definition of region parameters

REGIONS

Region	Value	Lowest Channel Limit (kHz)	Highest Channel Limit (kHz)
NA2 (Reduced FCC)	NA2	917,400 kHz	927,200 kHz
NA3 (Very Reduced FCC)	NA3	917,500 kHz	922,500 kHz
EU3 (ETSI)	EU3	865,600 kHz	867,600 kHz
IN (India)	IN	865,000 kHz	867,000 kHz
KR2 (Korea)	KR2	917,000 kHz	923,500 kHz
PRC (Peoples Republic of China)	PRC	920,125 kHz	924,875 kHz
AU (Australia)	AU	920,000 kHz	926,000 kHz
NZ (New Zealand)	NZ	922,000 kHz	927,000 kHz
JP (Japan)	JP	916,800 kHz	923,400 kHz
MY (Malaysia)	MY	919,000 kHz	923,000 kHz
ID (Indonesia)	ID	923,000kHz	925,000 kHz
PH (Phillipines)	PH	918,000 kHz	920,000 kHz
TW (Taiwan)	TW	922,000 kHz	928,000 kHz
MO (Macao)	MO	920,000 kHz	925,000 kHz
RU (Russia)	RU	866,000 kHz	868,000 kHz
SG (Singapore)	SG	920,000 kHz	925,000 kHz

Definition of return codes returned from DLL functions

API RETURN CODES

PA_ERR_OK
DLL function call completed without error. Corresponding data returned (generally status) is valid.

PA_ERR_PARAM1
First parameter supplied has failed validation.

PA_ERR_PARAM2
Second parameter supplied has failed validation.

PA_ERR_PARAM3
Third parameter supplied has failed validation.

PA_ERR_PARAM4
Fourth parameter supplied has failed validation.

PA_ERR_PIPE_OPEN_BUSY
Error occurs when the function paLMOpenPipe has explicitly opened the communication pipe to the printer language monitor but has not called paLMClosePipe. This error will NOT occur when using the PADLL status functions as these functions manage the proper opening and closing of the communications pipe.

PA_ERR_PIPE_OPEN_FAILED
Error occurs when a status request is directed to a printer\language monitor that has not opened a status pipe. Status pipes are opened by Practical Automation printer drivers version 2.0 and above.
This error indicates one of the following:
1 – Printer driver not installed
2 – Printer driver installed but is not set to default.
Set PracticalAutomation printer to default or use paSetNewPrinter(PrinterName) to point status function to proper printer

PA_ERR_SYSTEM
Error reading or writing to the pipe/language monitor.

PA_ERR_UNKNOWN_PRINTER
Printers language monitor does not support this function. Check with PA for a new driver version.

PA_ERR_PIPE_TIMEOUT
Timeout occurred reading from the printer language monitor.

PA_ERR_PRINTER_NOT_RESPONDING
The requested command has not been sent because the printer isn't responding to requests.

