iCLASS Reader Writer Migration

Application Note

PLT-03363, Rev. A.0

June 2017

hidglobal.com

Copyright

Trademarks

Revision History

Date	Description	Version
June 2017	Initial Release	A.0

Contacts

For additional offices around the world, see www.hidglobal.com corporate offices.

North America	Asia Pacific
611 Center Ridge Drive Austin, TX 78753 USA Phone: 866-607-7339 Fax: 949-732-2120	19/F 625 King's Road North Point, Island East Hong Kong Phone: 852 3160 9833 Fax: 852 3160 4809
Europe, Middle East and Africa	**Brazil**
Haverhill Business Park Phoenix Road Haverhill, Suffolk CB9 7AE England Phone: 44 (0) 1440 711 822 Fax: 44 (0) 1440 714 840	Condomínio Business Center Av. Ermano Marchetti, 1435 Galpão A2 CEP 05038-001 Lapa - São Paulo/SP Brazil Phone: 55 11 5514-7100
HID Global Customer Support: www.hidglobal.com/support	

Contents

1 Scope

The iCLASS Reader Writer Reader has been phased out by HID Global. This Application Note details the migration options for iCLASS Reader Writer Readers to iCLASS SE Platform Readers. The following is detailed for each migration option (where applicable); capabilities & limitations, testing by HID Global, documentation of serial output differences and how to move forward with migration options (part numbers or where to get developer documentation).

Specific alternative products are documented for the most popular primary reader use case - autonomous serial output of card PACS bits. Some alternatives utilize 3rd party protocol converting devices and in these cases HID Global has completed limited tested, however HID Global does not sell or provide technical support for these devices.

General guidance for alternative products is provided for the less popular and application specific secondary reader use case - custom card data read / write.

1.1 Primary Use Case - Autonomous Serial Output of Card PACS Bits

1.1.1 Alternative products

Depending on host physical interface and iCLASS Reader Writer Model, select the alternative iCLASS SE reader model from Table 1. If required, the part number for iCLASS SE Expansion module, iCLASS SE configuration card or 3rd party converter is listed. The software interface and physical wiring for the alternative product will be different; a system integrator must refer to the iCLASS Reader Writer software interface and iCLASS SE software interface to determine if the alternative product is compatible and to configure the system for compatibility.

Physical Interface	iCLASS Reader Writer Model/BasePN	iCLASS SE Reader Model/PN starts with	iCLASS SE Expansion Module Required	Configuration Card Required	3rd Party Converter	iCLASS Reader Writer Software Interface	iCLASS SE Software Interface
RS232	RW100/6101> RW150/6141 RW300/6111 RW400/6121 RWK400/6131 RWKL550/6171 RWKLB575/6181	R10/900NTNTEK R15/910NTNTEK R40/920NTNTEK R40/920NTNTEK RK40/921NTNTEK RKL40/929NTNTEK RKLB40/929NBNTEK	Not Required	Not Required	RF Ideas Wiegand to RS232 converter PN: OEM-W2RS232-V3	iCLASS Serial Protocol ASCII 144 bit card data string	RF Ideas ASCII data string
USB	RW100/6101 RW150/6141 RW300/6111 RW400/6121 RWK400/6131	R10/900NTNTEK R15/910NTNTEK R40/920NTNTEK R40/920NTNTEK RK40/921NTNTEK	PN: 6700-306-05	PN: SEX9X-CRD-0-Pxxx (Standard Key) or SEX9X-CRD-E-Pxxx (Elite Key)	FTDI 5V UART to USB PN: TTL-232RG-VSW5V-WE	iCLASS Serial Protocol ASCII 144 bit card data string – USB Host conformal to Human Interface Device (HID) Specification	iCLASS SE ASN.1 byte array FTDI - USB Driver
	RWKL550/6171 RWKLB575/6181	RKL40/929NTPTEK RKLB40/929NBPTEK	Not Required	PN: SEX9X-CRD-0-Pxxx (Standard Key) or SEX9X-CRD-E-Pxxx (Elite Key)	FTDI R485HDX to USB PN: USB-RS485-WE-1800-BT		iCLASS SE ASN.1 byte array FTDI - USB Driver
RS485HDX	RW100/6101 RW150/6141 RW300/6111 RW400/6121 RWK400/6131 RWKL550/6171 RWKLB575/6181	R10/900NTPTEK R15/910NTPTEK R40/920NTPTEK R40/920NTPTEK RK40/921NTPTEK RKL40/929NTPTEK RKLB40/929NBPTEK	Not Required	OSDP: Not Required iCLASS SE ASN.1 byte array: PN: SEX9X-CRD-0-Pxxx (Standard Key) or SEX9X-CRD-E-Pxxx (Elite Key)	Not Required	iCLASS Serial Protocol ASCII 144 bit card data string	OSDP 2.1.6 or iCLASS SE ASN.1 byte array

Physical Interface	iCLASS Reader Writer Model/BasePN	iCLASS SE Reader Model/PN starts with	iCLASS SE Expansion Module Required	Configuration Card Required	3rd Party Converter	iCLASS Reader Writer Software Interface	iCLASS SE Software Interface
RS485FDX	RW100/6101 RW150/6141 RW300/6111 RW400/6121 RWK400/6131 RWKL550/6171 RWKLB575/6181	R10/900NTNTEK R15/910NTNTEK R40/920NTNTEK R40/920NTNTEK RK40/921NTNTEK RKL40/929NTNTEK RKLB40/929NBNTEK	Not Required	Not Required	RF Ideas Wiegand to RS485 converter PN: OEM-W2RS485-V3	iCLASS Serial Protocol ASCII 144 bit card data string	RF Ideas ASCII data string
UART	RW100/6101 RW150/6141 RW300/6111 RW400/6121 RWK400/6131	R10/900NTNTEK R15/910NTNTEK R40/920NTNTEK R40/920NTNTEK RK40/921NTNTEK	PN: 6700-306-05	PN: SEX9X-CRD-0-Pxxx (Standard Key) or SEX9X-CRD-E-Pxxx (Elite Key)	Not Required	iCLASS Serial Protocol ASCII 144 bit card data string	iCLASS SE ASN.1 byte array
	RWKL550/6171 RWKLB575/6181	RKL40/929NTNTEK RKLB40/929NBNTEK	Not Required	Not Required	Wiegand to UART-No specific converter tested	iCLASS Serial Protocol ASCII 144 bit card data string	Dependent on 3rd party converter

Table 1: Autonomous Serial Output of Card PACS Bits Alternative Product Summary

HID Global has completed limited testing for 3rd party converters. HID Global does not sell or provide technical support for these devices.

Alternative readers utilizing 6700-306-05 expansion modules installed in the field may also be ordered with module installed in the factory via custom art number request.

1.1.2 iCLASS SE Software Interfaces

1.1.2.1 iCLASS SE ASN.1 Byte Array

Baud = 9600

Stop Bits = 1

Data Bits = 8

Data Flow = none

Example for a card programmed with H10301 format:

0011 0A4400000000 BD09 9E07 8105 06 80800E00 0712 (hex)

iCLASS SE Output: 0011 0A4400000000 BD09 9E07 8105 06 80800E00 0712

Message Length [2 bytes]: 0011

Message: 0A4400000000 BD09 9E07 810506 80800E00

CRC-16 [2 bytes]: 0712

Header [6 bytes]: 0A4400000000

Identifier [2 bytes]: BD09

Sub Identifier [2 bytes]: 9E07

iCLASS Media Tag [2 bytes]: 8105

Padding Bits number [1 byte]: 06

Card data with Padding: 80800E00

2. Extract Padding bits from the card data

80800E00 (hex) = 10000000100000000000111000000000 (bin)

Padding bits number = 06 hex = 6 (Dec)

1000 0000 1000 0000 0000 111000 000000 <- Padding bits = 6

CARD DATA = 1 0000 0001 0000 0000 0001 1100 0 (bin) = 02020038 (hex)

Step 4 – Interpreting Card Data

1. Select the format. In this case is 26 bits Stda (H10301).
2. Extract the parity bits. For H10301 format the parity bits are the first and the last bits.

1 0000 0001 0000 0000 0001 1100 0 (bin)

3. Separate FC and Card Number. FC bits are from 1 to 8 and for the Card Number are from 9 to 24

0000 0001 0000 0000 0001 1100

Facility Code = 0000 0001 (bin) = 1 (Dec)

Card Number = 0000 0000 0001 1100 (bin) = 28 (Dec)

1.1.2.2 FTDI - USB Driver

Host must be compatible with Virtual COM Port (VCP) FTID drivers and driver must be installed on the host (windows based hosts may automatically install driver upon first installation of device). Refer to FTID website for further information or for driver download
http://www.ftdichip.com/Drivers/D2XX.htm

1.1.2.3 RF Ideas ASCII Data String

Refer to user manual for device
https://www.rfideas.com/files/rfideas/files/support/doc/manuals/WiegandBoard-Manual.pdf

1.1.2.4 OSDP 2.1.6

Refer to SIA Specification available here:
http://services.securityindustry.org/eBusiness/Index.aspx

1.1.3 iCLASS Reader Writer Software Interfaces

1.1.3.1 iCLASS Serial Protocol ASCII 144 Bit Card Data String

Baud = 9600

Stop Bits = 1

Data Bits = 8

Data Flow = none

All standard iCLASS readers can be command card configured to echo the Wiegand data over the serial port. When a reader is configured to run the HID Access Control Application, after the reader authenticates with a programmed card it will echo 144 bits of data out the serial line. When the combined, the length of the start sentinel and card format do not equal 144 bits of data, zero padding will be attached to the beginning of the data. Data will be output as read below (left to right). The first occurrence of a 1 is the start sentinel. The start sentinel can be thrown away; the rest of the data is the card format. Card format data can be translated as first byte in to last byte in as most significant bit of the card format to least significant bit of the card format.

Zero Padding	Start Sentinel Bit	Card Format

An example output seen on the serial port for a 26-bit standard card format is as follows:

(hex) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 06 2C 00 2C

Where 18 bytes of data = 144 bits of data. The data (excluding most leading zeros) can be written as follows:

06 2C 00 2C (hex)

0000 0110 0010 1100 0000 0000 0010 1100 (binary) – (red = leading 0s + start sentinel)

10 0010 1100 0000 0000 0010 1100 (removal of start sentinel)

26-bit standard format is defined as follows:

EAAAAAAAABBBBBBBBBBBBBBBO where E is even parity for the first 1210 bits

(AAAAAAAABBBB), O is odd parity for the last 1210 bits (BBBBBBBBBBBB), A is the facility code and B is the card number.

0001 0110 0000 0000 0001 0110 (removal of parity)

Facility Code = 0x16 (22), Card Number = 0x16 (22)

1.1.3.2 iCLASS Serial Protocol ASCII 144 Bit Card Data String - USB

Host implements the iCLASS reader interface by conforming to the Human Interface Device (HID) specification.

The reader uses:

Vendor ID = 0x12D0

Product ID = 1

The data is bundled into a 65 byte packet. The first byte (byte 0) is an arbitrary report byte used to identify the packet. From the second byte the iCLASS Serial Protocol ASCII 144 bit card data string from Section 1.1.3.1 will be reported.

1.1.4 Alternative Product Physical Wiring

1.1.4.1 6700-306-05 iCLASS SE Module

Refer to *6700-603-05 iCLASS SE Module Installation Guide* (PLT-03284).

1.1.4.2 FTID R485HDX to USB or 5V UART to USB

iCLASS SE	(FTDI-USB)
Tx [RED/GREEN]	[4 ORANGE]
Rx [TAN]	[5 YELLOW]
GND [BLACK]	[1 BLACK]

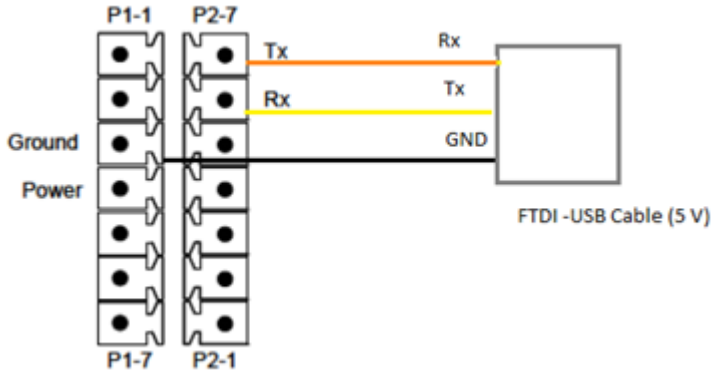

Figure 1: Terminal Connections in iCLASS SE Reader

1.1.4.3 RFideas Wiegand to RS485FDX or Wiegand to RS232

Refer to user manual for device:

https://www.rfideas.com/files/rfideas/files/support/doc/manuals/WiegandBoard-Manual.pdf

2 Secondary Use Case - Custom Card Data Read/Write

2.1 Alternative Product Guidance

Depending on physical interface and iCLASS Reader Writer Model, refer to the following table for guidance on alternative products:

Physical Interface	iCLASS Reader Writer Model/BasePN	Alternative Product: iCLASS SE Reader Model with Part Number	3rd Party Converter	Developer Information including Software Integration Documentation
RS232	RW100/6101 RW150/6141 RW300/6111 RW400/6121	900PTBTEK0036G 910PTBTEK0036G 920PTBTEK0036G 920PTBTEK0036G	UART to RS232	• Requires qualification as HID Global Embedded Solution Partner • For more information on this process contact your OEM Embedded Solutions sales representative in your country: https://www.hidglobal.com/worldwide-sales-tool
	RWK400/6131 RWKL550/6171 RWKLB575/6181	No Alternative		
USB	RW100/6101 RW150/6141 RW300/6111 RW400/6121	900PTBTEK0036G 910PTBTEK0036G 920PTBTEK0036G 920PTBTEK0036G	UART to USB	• Requires qualification as HID Global Embedded Solution Partner • For more information on this process contact your OEM Embedded Solutions sales representative in your country: https://www.hidglobal.com/worldwide-sales-tool
	RWK400/6131 RWKL550/6171 RWKLB575/6181	No Alternative		
RS485HDX	RW100/6101 RW150/6141 RW300/6111 RW400/6121	iCLASS SE Reader Module 900PTBTEK0036G 910PTBTEK0036G 920PTBTEK0036G 920PTBTEK0036G	UART to RS485HDX	• Requires qualification as HID Global Embedded Solution Partner • For more information on this process contact your OEM Embedded Solutions sales representative in your country: https://www.hidglobal.com/worldwide-sales-tool
	RWK400/6131 RWKL550/6171 RWKLB575/6181	No Alternative		
RS485FDX	RW100/6101 RW150/6141 RW300/6111 RW400/6121	900PTBTEK0036G 910PTBTEK0036G 920PTBTEK0036G 920PTBTEK0036G	UART to RS485FDX	• Requires qualification as HID Global Embedded Solution Partner • For more information on this process contact your OEM Embedded Solutions sales representative in your country: https://www.hidglobal.com/worldwide-sales-tool
	RWK400/6131 RWKL550/6171 RWKLB575/6181	No Alternative		
UART	RW100/6101 RW150/6141 RW300/6111 RW400/6121	900PTBTEK0036G 910PTBTEK0036G 920PTBTEK0036G 920PTBTEK0036G	Not required	• Requires qualification as HID Global Embedded Solution Partner • For more information on this process contact your OEM Embedded Solutions sales representative in your country: https://www.hidglobal.com/worldwide-sales-tool
	RWK400/6131 RWKL550/6171 RWKLB575/6181	No Alternative		

Table 2: Custom Card Read/Write Alternative Product Guidance

Appendix: A Sample Code to Capture Data From Win PC Serial Port

Simple Terminal SOFTWARE - C# Source Code used to capture data from the Serial Port on Windows PC – code is provided "as is" with no technical support.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO.Ports;

namespace SimpleTerminal
{
    class Program
    {
        static void Main(string[] args)
        {



            Console.WriteLine("******** SIMPLE TERMINAL ********" +
System.Environment.NewLine);
            //Get available COM Ports
            Console.WriteLine("AVAILABLE PORTS: ");
            string[] ports = SerialPort.GetPortNames();

            for (byte i = 0; i < ports.Length; i++)
                Console.WriteLine("{0}) {1}",i+1,ports[i]);

            Console.WriteLine();
            Console.Write("Select a port: ");
            string strPort = Console.ReadLine();

            SerialPort serialPort = new SerialPort(strPort);

            Console.WriteLine();
            Console.WriteLine("CONNECTION PARAMETERS:");
            serialPort.BaudRate = 9600;
            Console.WriteLine("BAUD RATE: 9600");
            serialPort.Parity = Parity.None;
            Console.WriteLine("PARITY: None");
            serialPort.StopBits = StopBits.One;
            Console.WriteLine("STOP BITS: 1");
```

```
            serialPort.DataBits = 8;
            Console.WriteLine("DATA BITS: 8");
            serialPort.Handshake = Handshake.None;
            Console.WriteLine("DATA FLOW: None");
            serialPort.ReadTimeout = 250;

            serialPort.DataReceived += new
SerialDataReceivedEventHandler(DataReceivedHandler);

            serialPort.Open();
            Console.WriteLine();
            Console.WriteLine("Present a Card or press any key to continue...");
            Console.WriteLine();
            Console.ReadKey();
            serialPort.Close();
        }

        private static void DataReceivedHandler(
                    object sender,
                    SerialDataReceivedEventArgs e)
        {

            SerialPort sp = (SerialPort)sender;
            int numbytes = 64;
            int j = 0;
            try
            {
                Console.Write("<-");
                do
                {
                    Console.Write(sp.ReadByte().ToString("X2"));
                    j++;

                } while (j < numbytes) ;


            }
            catch (System.TimeoutException te)
            {

               // Console.WriteLine(te.Message);
```

```
            }

            Console.WriteLine( " ({0} bytes)",j);

        }
    }
}
```