



ViSi SM130 RFID Module Interface to 4D Display

DOCUMENT DATE: **22nd April 2019**
DOCUMENT REVISION: **1.1**



Description

This Application note is intended to demonstrate to the user how to interface SM130 RFID Module to Diablo16 Displays. How to perform HALT, SEEK, AUTHENTICATE, and READ BLOCK commands of RFID using 4D Display will be discussed.

Before getting started, the following are required:

- The target screen can be any of the following Diablo16 touch display modules:

gen4-uLCD-24D Series	gen4-uLCD-28D Series	gen4-uLCD-32D Series
gen4-uLCD-35D Series	gen4-uLCD-43D Series	gen4-uLCD-50D Series
gen4-uLCD-70D Series		
uLCD-35DT	uLCD-43D Series	uLCD-70DT

Visit www.4dsystems.com.au to see the latest products using any of these graphics processors.

- [4D Programming Cable](#) / [µUSB-PA5](#)/[µUSB-PA5-II](#)
for non-gen4 displays (uLCD-xxx)
- [4D Programming Cable](#) & [gen4-IB](#) / [gen4-PA](#) / [4D-UPA](#),
for gen-4 displays (gen4-uLCD-xxx)

- [Workshop 4 IDE](#) (installed according to the installation document)
- [RFID Evaluation Shield](#)
- [RFID Module - SM130 Mifare](#)
- [RFID Tag - Adhesive Mifare 1K \(13.56 MHz\)](#)
- When downloading an application note, a list of recommended application notes is shown. It is assumed that the user has read or has a working knowledge of the topics presented in these recommended application notes.

Content

Description	2
Content	3
Application Overview	3
Setup Procedure	4
Create a New Project	4
Design the Project	5
<i>The ViSi - based application project</i>	<i>5</i>
<i>Halt Command</i>	<i>6</i>
<i>Select Command</i>	<i>7</i>
<i>Authenticate Command</i>	<i>8</i>
<i>Read Block Command</i>	<i>9</i>
Running the project	10
Proprietary Information	11
Disclaimer of Warranties & Limitation of Liability	11

Application Overview

This document shows how to make a 4D Display – based RFID evaluation software/reader that reads Mifare tags. The Graphical User Interface contains buttons for HALT, SEEK, AUTHENTICATE and READ BLOCK. The ViSi program controls the SM130 RFID Module via UART.

“Radio-frequency identification (RFID) is the wireless non-contact use of radio-frequency electromagnetic fields to transfer data, for the purposes of automatically identifying and tracking tags attached to objects. The tags contain electronically stored information. Some tags are powered by and read at short ranges (a few meters) via magnetic fields (electromagnetic induction). Others use a local power source such as a battery, or else have no battery but collect energy from the interrogating EM field, and then act as a passive transponder to emit microwaves or UHF radio waves (i.e., electromagnetic radiation at high frequencies). Battery powered tags may operate at hundreds of meters. Unlike a bar code, the tag does not necessarily need to be within line of sight of the reader, and may be embedded in the tracked object.

RFID tags are used in many industries. An RFID tag attached to an automobile during production can be used to track its progress through the assembly line. Pharmaceuticals can be tracked through warehouses.

[-Wikipedia](#)

SM130 RFID Module



RFID EVALUATION SHIELD



Setup Procedure

For instructions on how to launch Workshop 4, how to open a **ViSi** project, and how to change the target display, kindly refer to the section “**Setup Procedure**” of the application note

[ViSi Getting Started - First Project for Picaso and Diablo16](#)

Create a New Project

For instructions on how to create a new **ViSi** project, please refer to the section “**Create a New Project**” of the application note

[ViSi Getting Started - First Project for Picaso and Diablo16](#)

Design the Project

To create a simple program that will be able to control the SM130 RFID Module, we will need to use some commands enlisted in the [DIABLO 4DGL Internal Functions](#).

Serial (UART) Communication functions are the commands/functions used to be able to control the RFID module. This project used uLCD - 35DT which uses DIABLO processor. To use UART commands, first we need to specify the GPIO pin to use for the com ports receive and transmit line. Here is the list of available GPIO to use:

4D Pin Name (Predefined)	DIABLO16 Pin Number	Availability
PA0	61	Yes
PA1	62	Yes
PA2	63	Yes
PA3	64	Yes
PA4	46	Yes
PA5	49	Yes
PA6	50	Yes
PA7	51	Yes
PA8	52	Yes
PA9	53	Yes
PA10	43	Yes
PA11	44	Yes
PA12	31	Yes
PA13	32	Yes
PA14	37	No
PA15	36	No

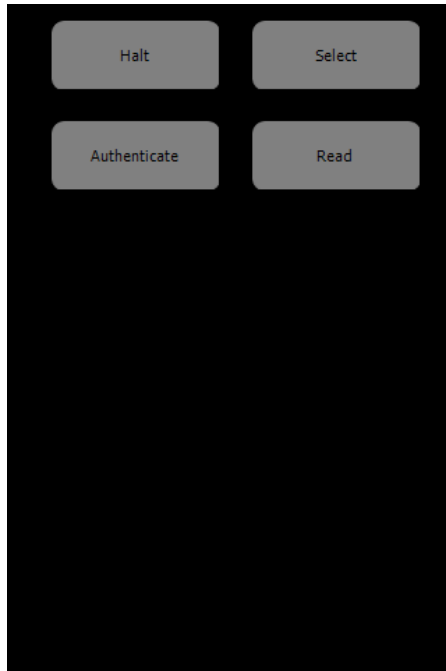
This project used PA12 GPIO for RX pin, and PA13 for TX pin which is connected to Serial Port of SM130 Module.

The ViSi - based application project

SM130 have two communication interfaces; UART and I2C. In this application note, UART was used. The communication between the host and the module can take place at 9600bps, 19200bps, 38400bps, 57600bps or 115200bps.

```
COM1_RX_pin(PA12);  
COM1_TX_pin(PA13);  
com_SetBaud(COM1, 960);  
com1_Init(combuf, 50, 0);
```

In this project, the communication between SM130 and the display is at 9600bps.



If one of the button is pressed. The display will output the following: Command, Response, UART frame sent by the display and Response sent by SM130.



The [datasheet of SM130 RFID Module](#) shows the different commands and responses. The datasheet also discusses the communication protocols. This project only discusses how to execute HALT, SELECT, AUTHENTICATE and READ BLOCK 0 commands.

Halt Command

To execute HALT command of the SM130, the display must transmit the necessary UART FRAME:

```
serout1(255);
serout1(0);
serout1(1);
serout1(147);
serout1(148);
pause(100);
```

This UART Frame is sent by the HALT() function which is executed if HALT button is pressed



Select Command

This command Selects a Tag if it is present in the field. On receiving the command, the module executes an Anti-collision and Select command sequences. If a tag was present, then it selects the tag and sends the serial

number of the tag as response. If a tag was not present, it sends an error code back.

Here are the following UART Frame sent by the display in Select() function:

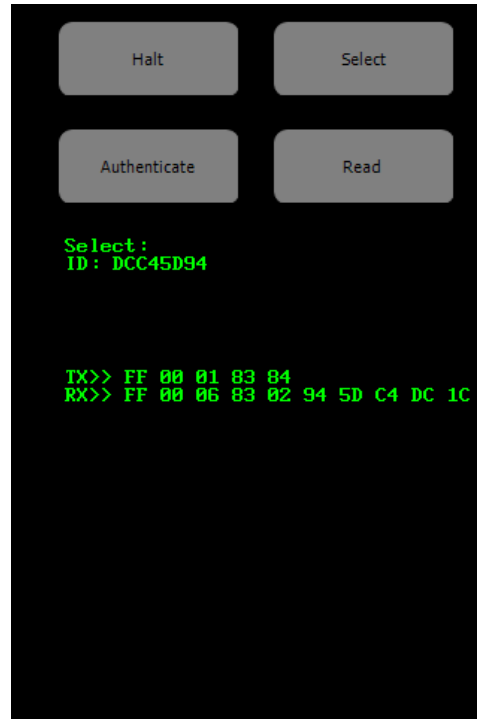
```
serout1(255);
serout1(0);
serout1(1);
serout1(131);
serout1(132);
pause(100);
```

If a tag was selected, the response data length is 6 or 9 bytes. The first byte is the command byte (0x83) and the next is the Tag Type and the next 4 bytes / 7 bytes are the tag serial number MSB first. The serial number is 4 bytes for Mifare 1K and 4K tags and 7 bytes for Mifare Ultralight tags.

If no tag was present, the data length is set to 1 and the Error code returned is 'N' If RF Field is OFF, the data length is set to 1 and the Error code returned is 'U'.



Output when no tag



Output when tag is present

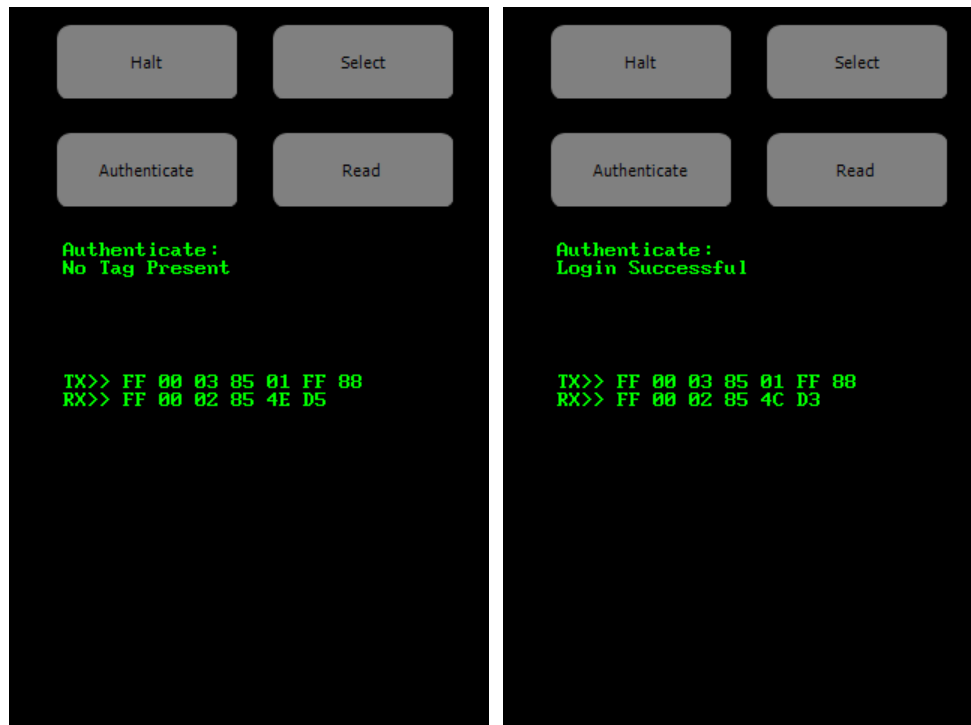
Authenticate Command

This command authenticates the specified block with the specified Key type and Key sequence. If Authentication fails then the Select Tag operation should be repeated to authenticate again.

Here are the following UART Frame sent by the display in Authenticate() function:

```
serout1(255);  
serout1(0);  
serout1(3);  
serout1(133);  
serout1(0);  
serout1(255);  
serout1(135);  
pause(100);
```

If a tag is previously selected the response would be login successful. Else if no tag is previously selected then the response would be no tag present.



Output with no tag

Output with tag

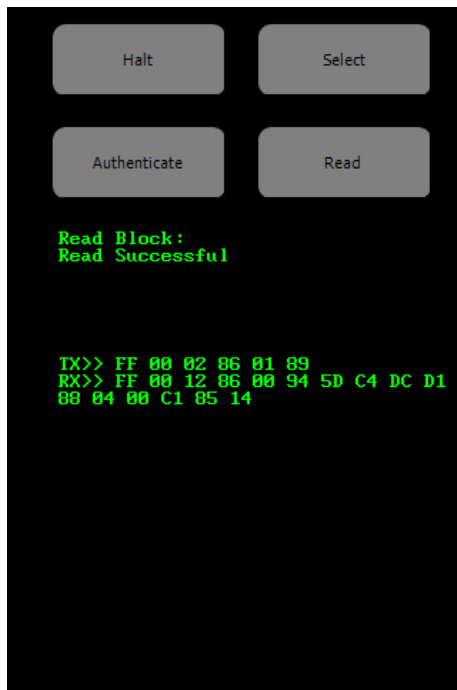
Read Block Command

This command reads 16 bytes from the specified block. Before executing this command, the particular block should be authenticated. If not authenticated, this command will fail.

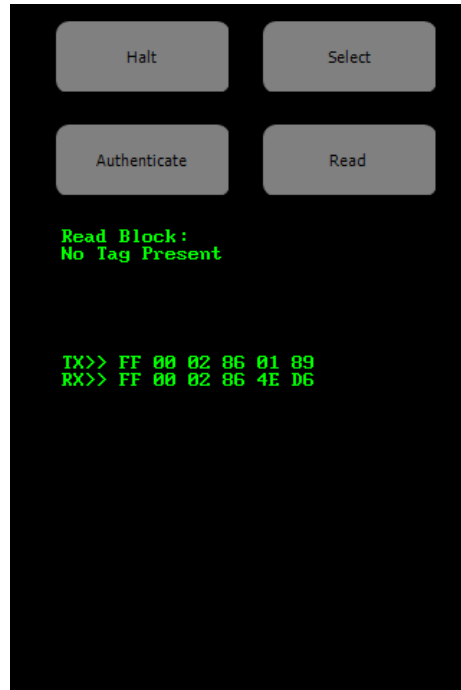
Here are the following UART Frame sent by the display in Authenticate() function:

```
serout1(255);  
serout1(0);  
serout1(2);  
serout1(134);  
serout1(0);  
serout1(136);  
pause(100);
```

When reading a Mifare UL tag, the first 4 bytes are from the block number specified. The next 12 bytes are from the consecutive blocks. In this project, BLOCK 0 of the tag is read.



Output if read successful



Output if read fail

Running the project

For instructions on how to save a **ViSi** project, how to connect the target display to the PC, how to select the program destination (this option is not available for Goldelox displays), and how to compile and download a program, please refer to the section “**Run the Program**” of the application note

[ViSi Getting Started - First Project for Picaso and Diablo16](#)

The uLCD-32PTU and uLCD-35DT display modules are commonly used as examples, but the procedure is the same for other displays.

Proprietary Information

The information contained in this document is the property of 4D Systems Pty. Ltd. and may be the subject of patents pending or granted, and must not be copied or disclosed without prior written permission.

4D Systems endeavours to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission. The development of 4D Systems products and services is continuous and published information may not be up to date. It is important to check the current position with 4D Systems.

All trademarks belong to their respective owners and are recognised and acknowledged.

Disclaimer of Warranties & Limitation of Liability

4D Systems makes no warranty, either expresses or implied with respect to any product, and specifically disclaims all other warranties, including, without limitation, warranties for merchantability, non-infringement and fitness for any particular purpose.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

In no event shall 4D Systems be liable to the buyer or to any third party for any indirect, incidental, special, consequential, punitive or exemplary damages (including without limitation lost profits, lost savings, or loss of business opportunity) arising out of or relating to any product or service provided or to be provided by 4D Systems, or the use or inability to use the same, even if 4D Systems has been advised of the possibility of such damages.

4D Systems products are not fault tolerant nor designed, manufactured or intended for use or resale as on line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of the product could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). 4D Systems and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

Use of 4D Systems' products and devices in 'High Risk Activities' and in any other application is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless 4D Systems from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any 4D Systems intellectual property rights.