



Lab Garage, Powered by Planeta Informática

EMV Reader Implementation Guide

May 2019

Version 1.0

Lab Garage - Confidential

(Visit our website: <http://www.planeta.inf.br>)

Public

The information in this document is proprietary to Lab Garage and is intended for use only by Lab Garage clients and other third parties that have a current nondisclosure agreement (NDA) that covers disclosure of the information contained herein.

No part of this document may be reproduced in any form by any means without prior written authorization of Lab Garage.

Changes are periodically added to the information herein. At any time, Lab Garage may make improvements and/or changes in in this document.

© Lab Garage 2019

Document history

Version	Date	Author	Summary of Changes
1.0	May 20 th , 2019	Bernardo Segal, Bruno Jesus	Initial version

Contents

1	About this document.....	5
1.1	Purpose.....	5
1.2	Audience.....	5
1.3	Scope.....	5
1.4	Document organization.....	6
2	Definitions	7
3	Introduction	8
4	VSAM	9
5	Reader Requirements	10
6	Source Code Provided	11
7	Implementation test.....	12
8	Remote firmware update.....	13

Tables

Table 1 - Document organization	6
Table 2 - Definitions.....	7

Figures

Figure 1 - Contactless reader block diagram.....	8
Figure 2 - VSAM format	9

1 About this document

1.1 Purpose

This document intends to describe the steps to add, adapt and compile the source code described ahead to a smart card EMV Level1 reader firmware that with a VSAM installed becomes EMV L2 compliant.

1.2 Audience

This document is intended for the following entities implementing contactless payment acceptance:

- Transport operators
- System integrators
- Software developers

It is meant for tech people with C language knowledge and that are responsible to comply with this document targets.

1.3 Scope

This document provides a technical guide of this implementation developed by Planeta Informatica (www.planeta.inf.br).

1.4 Document organization

This document has the following sections:

Table 1 - Document organization

Section	Description
1. About this document	Describes the purpose, audience, scope and organization of this guide.
2. Definitions	List of definitions used in this document
3. Introduction	Shows a contactless reader architecture
4. VSAM	Brief description of the SAM used in this implementation
5. Reader Requirements	Describes the minimum requirements that the reader must fulfill
6. Source Code Provided	Explanation about the C code here provided
7. Implementation Tests	Describes the test to check if the new adapt layer is working properly
8. Remote firmware update	Details about the reader update process using APDU commands

2 Definitions

This document uses the following terms and abbreviations:

Table 2 - Definitions

Term	Description
Contactless Card	A cardholder device carrying a contactless payment application. It may be implemented on a plastic card, a mobile phone, a wearable device, or an alternative form factor.
Contactless reader	Reader that accepts contactless cards (e.g. contactless debit or credit cards) and support contactless payment methods like Apple, Android and Samsun Pay.
EMV Level 1 Certification	Defined by EMV Co. It covers the electrical / physical interfaces and the transmission of data between the terminal and the card and ensures that the contactless reader meets the lower level electromagnetic and communication protocol requirements. There is an extensive EMV Co defined level 1 approval process, which requires every card reader to have completed laboratory type approval before they can be used to perform EMV transactions.
EMV Level 2 Certification	Certification defined by EMV Co, concerned with the validation of the software that implements the payment functionality and that runs on the EMV Level 1 certified contactless reader. Referred to as a payment kernel and it used to perform EMV transactions. There is an extensive EMV Co defined Contactless Level 2 approval process, which requires every reader to have completed laboratory type approval.
PCI	PCI Standard. It is mandated by the card brands and administered by the Payment Card Industry Security Standards Council. The standard was created to increase controls around cardholder data to reduce card fraud.
SAM	Secure Access Module is a smart card based on plug-in format plugged into a contact card slot. It enhances the security and cryptography performance in payment terminals, such as validator, against contactless cards.

3 Introduction

This document is intended for programmers that are responsible to add, compile and test the supplied source code for the EMV reader. If you did not download this document from BitBucket the first thing to do is asking Planeta Informática for BitBucket access to the source code and extra documentation repository (<https://bitbucket.org/pibrz/pi-reader/>).

The whole source code provided is programmed in C language. It was compiled and tested in a variety of compilers and we expect little to no problem when compiling for your reader hardware. Any bugs and problems can be reported directly at the BitBucket repository Issue tracker.

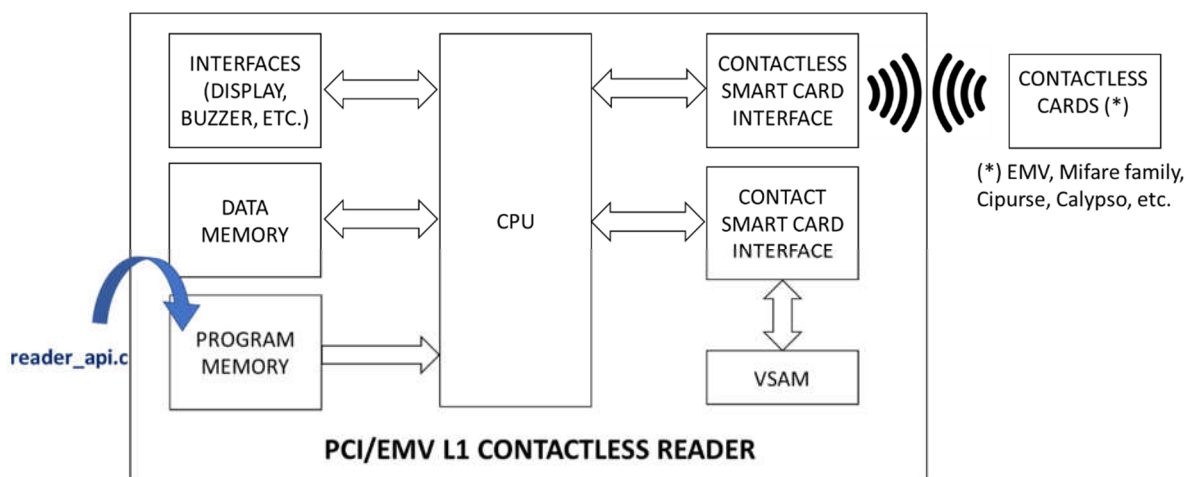
The source code is split in several files but only a single file must be modified, this file is the adapt layer for your hardware. All other files are internal files used to handle APDU commands received and dispatching them to the correct hardware devices, they must not be modified. In case bugs are found in the internal files an issue can be created at the tracker. More information about the adapt layer and its divisions in the chapter 6. The reader main loop may also be changed to fit your hardware needs.

The internal files may be updated from time to time to expand its capabilities with new commands for example. The adapt layer may evolve too and in such cases new functions may be required from the reader hardware. For more information about the minimum reader hardware requirements look at chapter 5.

After completed, the EMV reader will be then interchangeable with any other reader that implements the same specification.

The figure below shows the main blocks of a reader.

Figure 1 - Contactless reader block diagram



4 VSAM

This SAM (Secure Access Module), known as “VSAM”, is a smart card plug-in format, installed in the EMV Level 1 certified contactless reader. The main purpose is to guarantee security and cryptographic performance of contactless payments.

It contains level 2 EMV Kernel, it is optimized for Visa On-Line Deferred transaction, including: FDDA validation, card expiration date verification, EMV L2 validations, VCPS specification (Visa Contactless Payment System), specifications for Visa Ready MTT (mass transit transaction) and deny list.

Figure 2 - VSAM format



5 Reader Requirements

The reader must fulfill the following requirements for this implementation:

- Must have:
 - EMV Level 1 certified;
 - A communication interface that can be either USB device (used as CDC protocol) or RS232 serial port or Ethernet or others;
 - At least one contact card slot for SAM cards (including VSAM) in SIM card format (at least 2 slots being desirable) – ISO 7816 compliant;
 - Contactless interface with support for ISO-14443 type A and B compliant with level - 4 communication protocol;
 - Program memory space to support this implementation. The size depends on the compiler. Usually 256 Kbytes is more than enough;
 - Additional RAM memory: 32 Kbytes.
- Desirable:
 - Buzzer that rings in different frequencies;
 - 4 leds;
 - Thread support for better performance;
 - Support for remote firmware upgrade (see chapter 8);
 - PCI certification desired (it depends on the geographic region and local acquire rules).

6 Source Code Provided

The reader_api.c is divided in four sections:

- **LOW LEVEL FUNCTIONS:** Functions related to the reader CPU and external communication.
- **CONTACTLESS AND CONTACT FUNCTIONS:** Functions to power on/off and exchange commands to both contact and contactless cards.
- **MIFARE FUNCTIONS:** Mifare specific functions (Auth, Read, Update, Inc, Dec).
- **FILE FUNCTIONS:** Related to file system manipulation for creating, writing and reading files.

In order to understand what each function does it is possible to refer to the adapt implementations supplied for Windows and GNU/Linux. The Windows project can be opened in Visual Studio 2010 onwards. This virtual implementation uses a Planeta RC700 reader to dispatch the commands.

If your reader hardware runs on GNU/Linux, you can drastically reduce the implementation time by reusing the CPU and File sections of the functions supplied in the GNU/Linux implementation.

7 Implementation test

In order to test if the new adapt layer is working as expected, you can download and install the software PIDE following the instructions in the repository.

The tests consist of series of APDU commands to check if all hardware is working as expected. To run the complete test, a VSAM, a Mifare Classic card and an EMV card are required.

The VSAM test ensures that the contact card slot is working as expected and it will be possible to check if the speed is acceptable.

The Mifare card will be used to test the Mifare specific API and functions, a blank Mifare Classic 1K card is required, no keys will be changed.

The EMV card will be used to test the detection of intelligent cards and level 4 ISO-14443 APDU transmissions.

8 Remote firmware update

It is desirable that the reader firmware to be updated using APDU commands. In order to implement this feature, the file system functions must be implemented. The reader calls the function `OpenFileC` with file name `PROGRAM`, after that it will write the whole firmware data using calls to function `WriteDataFile`. And finally, a call to `ReadDataFile` with `data == NULL` and `addr == 0`, this is the trigger for the reader to use the contents of the file to self-update the firmware and reset.