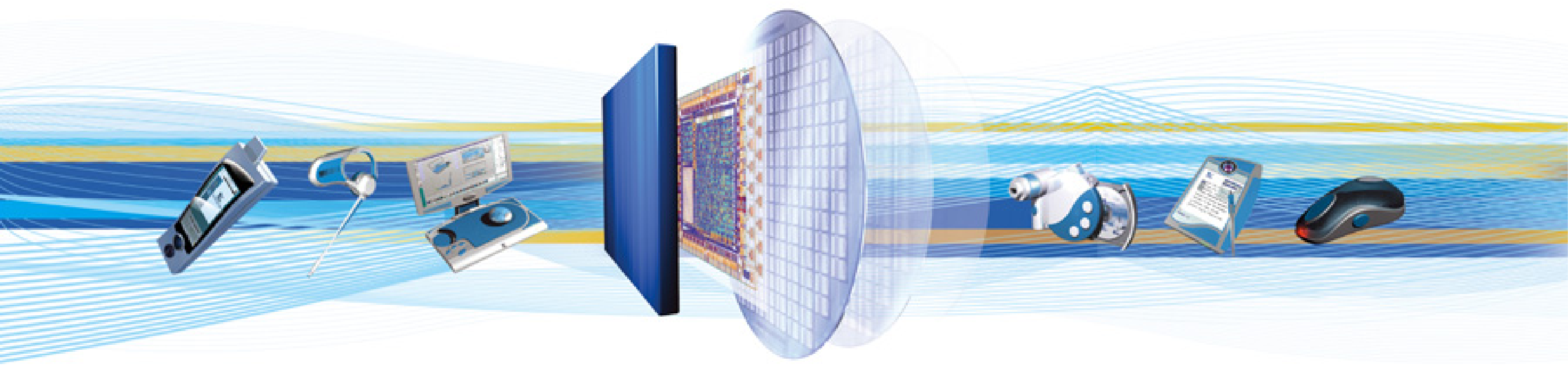
CSR Synergy®

CSR Synergy Framework 3.1.0

VM – Virtual Machine

API Description

August 2011



Cambridge Silicon Radio Limited

Churchill House
Cambridge Business Park
Cowley Road
Cambridge CB4 0WZ
United Kingdom

Registered in England and Wales 3665875

Tel: +44 (0)1223 692000
Fax: +44 (0)1223 692001
www.csr.com

Contents

List of Figures

List of Tables

1 Introduction

1.1 Introduction and Scope

This document describes the functionality and message interface used for receiving and sending Virtual Machine (VM) commands from and to the Virtual Machine within the BlueCore® chip.

Before reading this document, it is recommended to read [1] and [2].

2 Description

2.1 Introduction

The Virtual Machine (VM) module is used for sending and receiving messages from/to the Virtual Machine within the BlueCore® chip. The Virtual Machine embedded on the Bluecore is capable of running special made applications like the AV-router, echo cancelation etc.

The VM protocol is not part of the Bluetooth standard.

The VM module is used for carrying message between the host application and a VM application running on the virtual machine and the VM module does not know anything about the protocol defined between the two applications and is only a mean of transportation of messages and data. For a description of the commonly used message format between the host application and the VM application, please see Appendix A.

An application on top of the VM module can use the module to send and receive VM commands to the VM applications.

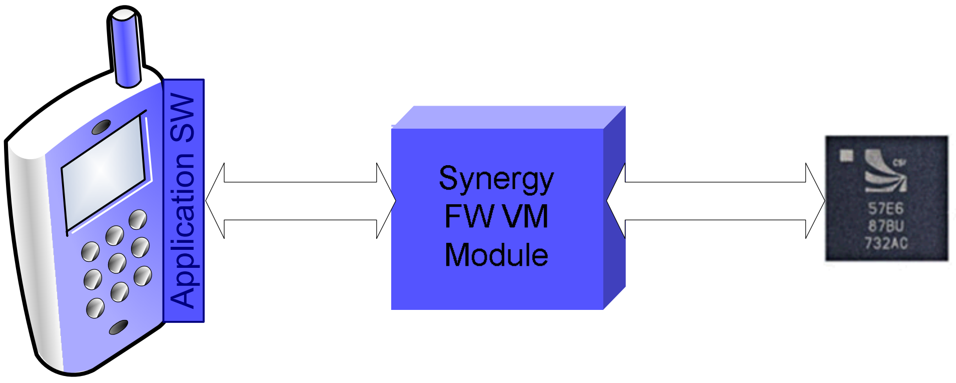


Figure 1: Typical scenario

Before using the VM module the host application must register with the VM module in order for the VM module to know where received VM messages are to be sent to. After the registration the host application can send and receive VM messages by means of a set of functions and primitive provided by the VM module These are described in chapter 3 of this document.

The VM module does not impose any restrictions on the protocol between the host and VM application

If no application is registered in the VM module – VM messages from the VM on the Bluecore will be discarded in the VM module.

A common way of using the CSR Synergy Framework VM module is depicted in Figure 2. Here the registration and message sending and receiving is shown.

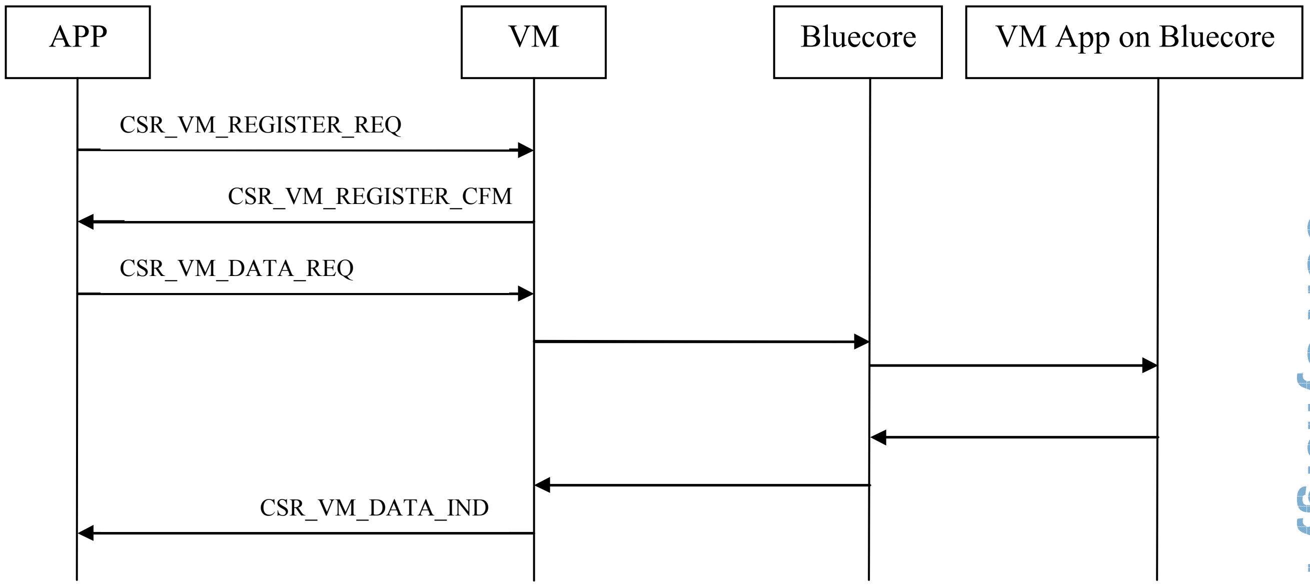


Figure 2: Example of how the CSR Synergy Framework VM module may be used

2.2 Sequence Overview

The VM module will always (after it has been initialised by the scheduler) be able to do the following:

- Register which application VM messages received from the Bluecore is to be send to
- Send VM messages from the host to the VM on the Bluecore.
- Receive VM messages from the VM on the Bluecore to the host

For further details on how this is done, please refer to section 3.

3 Interface Description

In this section a series of MSCs will be shown to explain the usage of the VM module. The primitives and the functions available to the application are also described in the subsections of this chapter.

3.1 Registration

If an application wants to “talk” with the VM application on the Bluecore it must register with the VM module first. A registration is initiated using the `CsrVmRegisterReqSend` function, which sends a CSR_VM_REGISTER_REQ primitive to the VM module.

The prototype for the `CsrVmRegisterReqSend` function is:

```
    CsrVmRegisterReqSend(CsrQid pHandle)
```

The arguments to the function are described in Table 1.

Type	Argument	Description
CsrQid	phandle	Handle to application, i.e. the queue on which the CSR_VM_REGISTER_CFM and CSR_VM_DATA_INDs are put by the VM module

Table 1: Arguments to `CsrVmRegisterReqSend` function

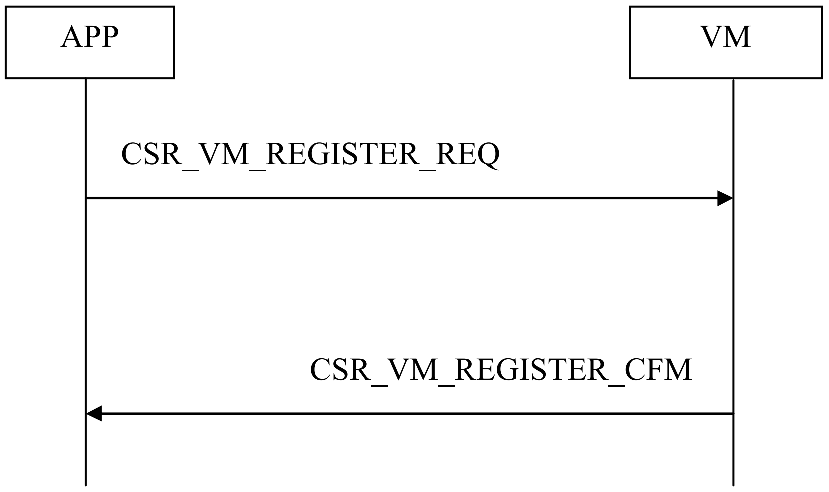


Figure 3: VM registration

When the VM module has finished handling the CSR_VM_REGISTER_REQ, it will confirm the request by sending a CSR_VM_REGISTER_CFM primitive, which contains the parameters found in Table 2 below. Note that the confirm signal is generated locally, as the registration is made in the CSR Synergy Framework VM module.

Currently the *result* parameter will always equal CSR_VM_OK (0x0000) as no error can occur in the registration process. All other result codes are reserved for future use and will indicate that an error has occurred.

Type	Parameter	Description
CsrVmPrim	type	Signal identity – always CSR_VM_REGISTER_CFM.
CsrResult	result	Result code. Currently this will always be CSR_RESULT_SUCCESS (0x0000) indicating a successful registration. All other result codes are reserved for future use and will indicate that an error has occurred.

Table 2: Parameters in a CSR_VM_REGISTER_CFM primitive

3.2 Sending and receiving VM messages

Once the VM module has been initialised, it is possible to send and receive VM messages. To send a message the VM_DATA_REQ is used and VM messages from the VM on the Bluecore is delivered in a CSR_VM_DATA_IND primitive. On Figure 4 and Figure 5 the sending and receiving is shown.

The prototype for the `CsrVmDataReqSend` function is:

```
CsrVmDataReqSend(CsrUint16 payloadLength, unit8_t *payload)
```

The arguments to the function are described in Table 3.

Type	Argument	Description
CsrUint16	payloadLength	Length of the payload data and is given in number of bytes
CsrUint8	*payload	The VM Message that is to be send

Table 3: Arguments to `CsrVmDataReqSend` function

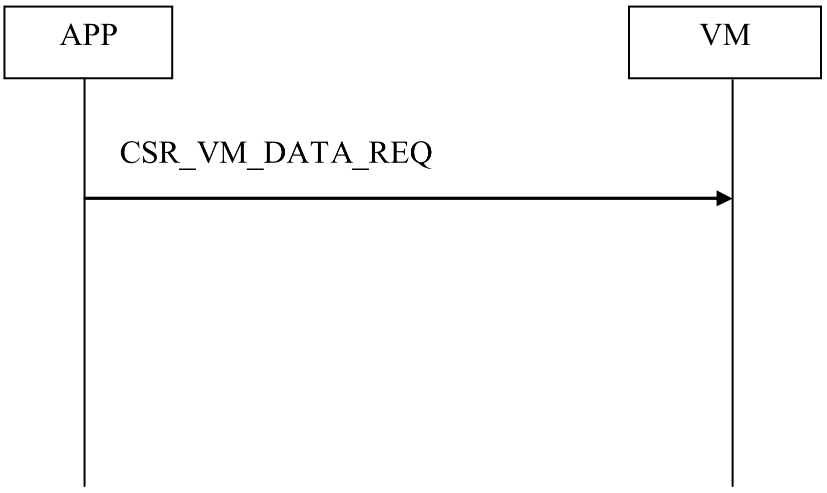


Figure 4: HQ message forwarding

The parameters of the CSR_VM_DATA_REQ primitive are found below.

Type	Parameter	Description
VmPrim	Type	Signal identity – always CSR_VM_DATA_REQ
CsrUint16	payloadLength	Length of the VM message
CsrUint8	*payload	The VM message.

Table 4: Parameters in a CSR_VM_DATA_REQ primitive

If CSR_BLUECORE_ONOFF is defined, the CSR_VM_DATA_REQ primitives will be discarded if the BlueCore is in the process of deactivating. If the application attempts to send a CSR_VM_DATA_REQ before the BlueCore has been activated, the request may be discarded by the lower layers . Consequently, the application should not send the CSR_VM_DATA_REQ until it is certain that the BlueCore has been activated. This can be achieved either by communicating with the controlling application which performs the BlueCore activation or by registering as a delegate using the Transport Manager API.

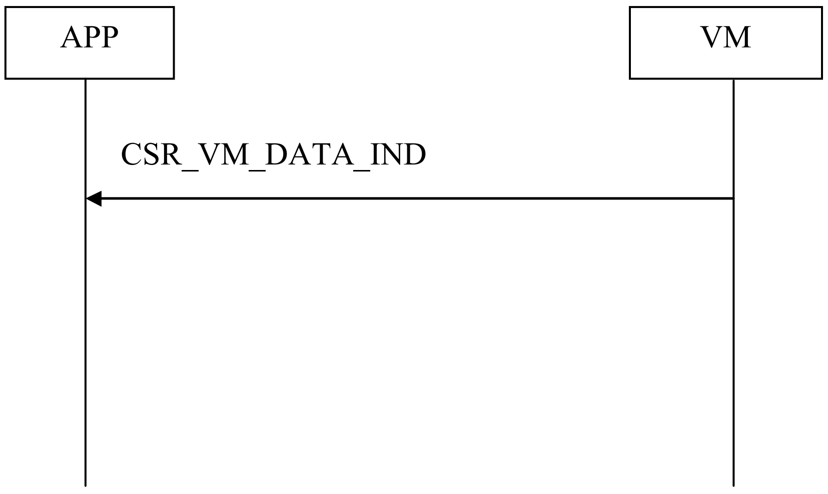


Figure 5: VM message from Bluecore

The parameters of the CSR_VM_DATA_IND primitive are found below.

Type	Parameter	Description
VmPrim	Type	Signal identity – always CSR_VM_DATA_IND
CsrUint16	payloadLength	Length of the VM message
CsrUint8	*payload	The VM message.

Table 5: Parameters in a CSR_VM_DATA_IND primitive

If CSR_BLUECORE_ONOFF is defined, the CSR_VM_DATA_IND will not be sent to the registered task if the BlueCore is in the process of deactivating. The data received from the chip will be discarded.

Appendix A VM Message format

The format within the VM messages is as such not part of the VM interface as the VM module does not know anything of the protocol format, but most VM applications uses the following format for sending and receiving messages:

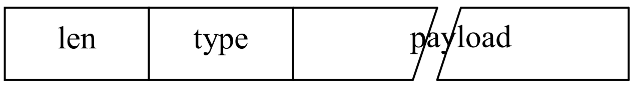


Figure 6 VM Message format

where

len	The length of the whole VM message including the length field itself, the type and the payload. The length is specified in number of words (16 bit).
type	Is the type of the VM message. Message type 0x0080 – 0x0100 is reserved for use internally and may not be used in any VM applications.
payload	Is the VM message payload

4 Document References

Terms and Definitions

BlueCore®	Group term for CSR's range of Bluetooth wireless technology chips
Bluetooth®	Set of technologies providing audio and data transfer over short-range radio connections
CSR	Cambridge Silicon Radio
UniFi™	Group term for CSR's range of chips designed to meet IEEE 802.11 standards

Document History

Revision	Date	History
1	19 DEC 08	Ready for first Engineering Release
2	16 JAN 09	Ready for second Engineering Release
3	09 FEB 09	Ready for release 1.0.0
4	26 MAY 09	Ready for release 1.1.0
5	30 NOV 09	Ready for release 2.0.0
6	20 APR 10	Ready for release 2.1.0
7	OCT 10	Ready for release 2.2.0
8	DEC 10	Ready for release 3.0.0
9	Aug 11	Ready for release 3.1.0

TradeMarks, Patents and Licences

Life Support Policy and Use in Safety-critical Compliance

Performance and Conformance

Refer to www.csrsupport.com for compliance and conformance to standards information.