EM9305 – NVM PROGRAMMING

Product Family: **EM BLE System-On-Chip solution**

Part Number: EM9305
Keywords: NVM programing

PURPOSE

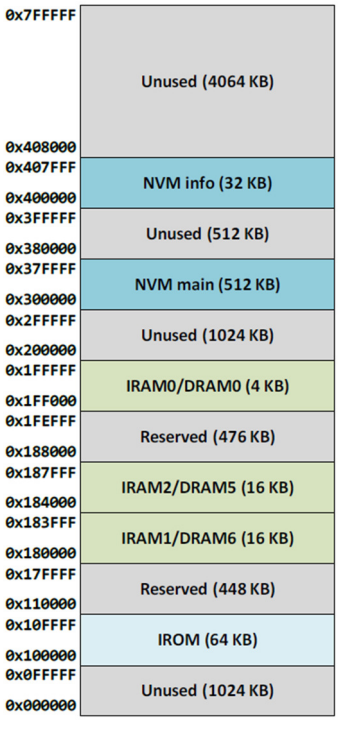
The purpose of this application note is to share information related to the programming of the NVM in the EM9305.

SCOPE

The EM9305 described in this application note has the ROM version 1.0.

INTRODUCTION

The NVM is one of the memories available on the EM9305 with some IROM, IRAMs and DRAMs. The NVM is split into two parts: the "NVM main" and the "NVM info".



Instruction address space

The "NVM main" has a size of 512kB and is split into 64 pages of 8kB.
The "NVM info" has a size of 32kB and is split into 4 pages of 8kB. It is organized as described in the following table and only Page 2 can be programmed by the user.

NVM INFO PAGE	PURPOSE
Page 3	EM data. This page is used by EM to store: • EM production information (wafer and lot information, device ID, MAC address) • EM trimming data • EM hardware lock bits • EM authentication public keys
Page 2	User data. This page is used to store User related date: • User flags • User trimming data • User hardware lock bits • User authentication public keys
Page 1	Reserved for future use.
Page 0	Key container. CPU does not have access to this page.

NVM PROGRAMMING FLOW

"NVM info Page 2" programming

A dedicated Application Note is available to address the programming of "NVM info page 2" and its particularities, but it mainly follows the flow described for the "NVM main" programming

"NVM main" programming

Programming the "NVM main" is a 4–step process:
1- Enter in "configuration mode"
2- Perform a "NVM main" mass erase
3- Write the NVM content, generally an ihex file
4-Calculate and compare the CRC to check the Write success

NVM PROGRAMMING HCI COMMANDS

The HCI commands listed below allow programming the EM9305. Details can be found in chapters "11.2.6. EMS MEMORY MANAGEMENT (EMSMM) COMMANDS" and "11.2.3. EMS GENERAL (EMSG) COMMANDS" of the EM9305 datasheet.

COMMAND	OPCODE
EMSMM Write At Address	0xFD03
EMSMM Write Continue	0xFD04
EMSMM NVM Erase Main	0xFD06
EMSG Calculate CRC32	0xFC4E

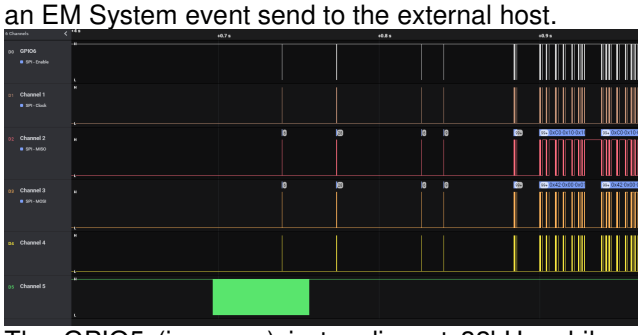
For details about the HCI commands format, please refer to chapter "11.1. PACKET DESCRIPTION" of the EM9305 datasheet.

PHYSICAL ACCESS

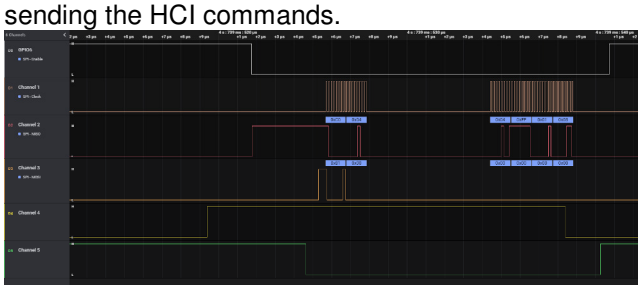
HCI commands can be used over UART or SPI. Per default the EM9305 is in SPI slave and can be addressed through:
GPIO0: CSN-S
GPIO1: SCK-S
GPIO2: MISO-S
GPIO3: MOSI-S
GPIO4: RDY

CONFIGURATION MODE

Configuration Mode is entered if GPIO5 is toggling at 30 KHz rate during the boot process. The CM request detection is based on edge detection of a signal in the range 25kHz – 50kHz.
Once Configuration Mode is entered, it is signalized by an EM System event send to the external host.



The GPIO5 (in green) is toggling at 32kHz while a reset is triggered to start a boot process, prior to sending the HCI commands.



CM Response is 0x04FF0103

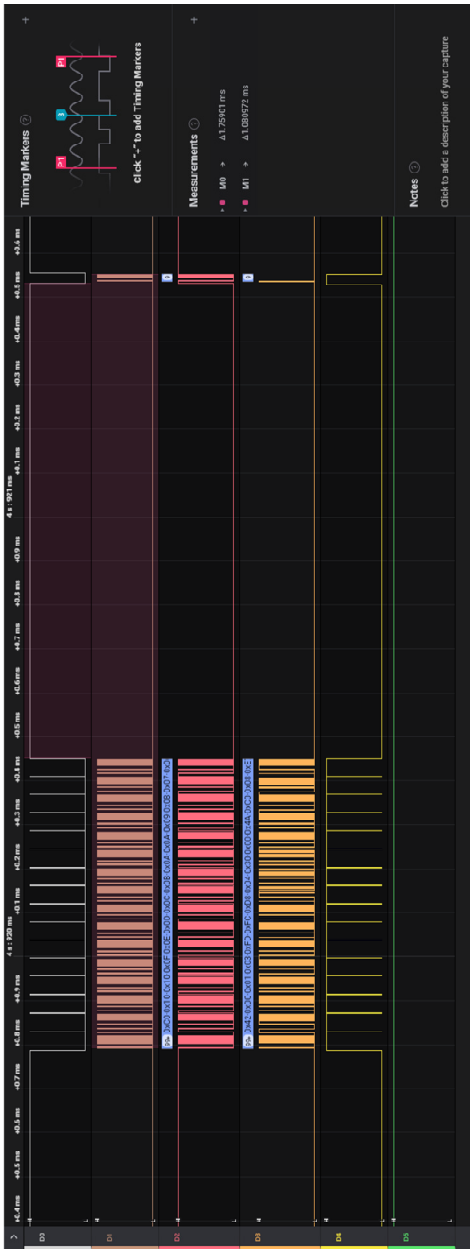
PERFORMANCE EXPECTATIONS

Successive write commands over the SPI bus are used to program the NVM. It should make the best possible usage of the SPI FIFO RX.
The times below are the times to program the full 512kB of NVM in different conditions. SPI is running at 7MHz on the DVK

Programming tool	Comments	time
Production tester	2.5MHz SPI	~15s
Production tester	Extrapolation at 8MHz	~5s
DVK	Fast mode	10s
DVK	Slow mode	84s
DVK	Extrapolation removing pause time between 248 useful Bytes writing bursts	**3.73s**

Below, the times to physically write 248 Bytes in the NVM.

Programming tool	Comments	time
Theoretical	Best case - Physical write time, no control software overhead	496us
Theoretical	Worst case- Physical write time, no control software overhead	992us
DVK	Time between the end of the SPI write and the SPI event	~1100us



RECOMMENDATIONS

To minimize the writing time, the maximum length for an HCI command of 256bytes should be used. This would allow 248 bytes to be written in the memory and reduces the protocol overhead to its minimum.

When performing multisite programming, wait for the worst case write timing before checking if all receive events are received as write time will differ slightly from parts to part

Programming “NVM info page 2” should be done at the latest possible stage