LMK5B12204/LMK05318B EEPROM Programming

Dec. 9, 2022

Clock and Timing Solutions, Apps

Contents

- Memory Overview Diagram
- EEPROM programming method
  - REGCOMMIT
  - SRAM Direct Write
- Using the .epr to I2C writes excel tool
- Register description of REGCOMMIT method

Memory Overview

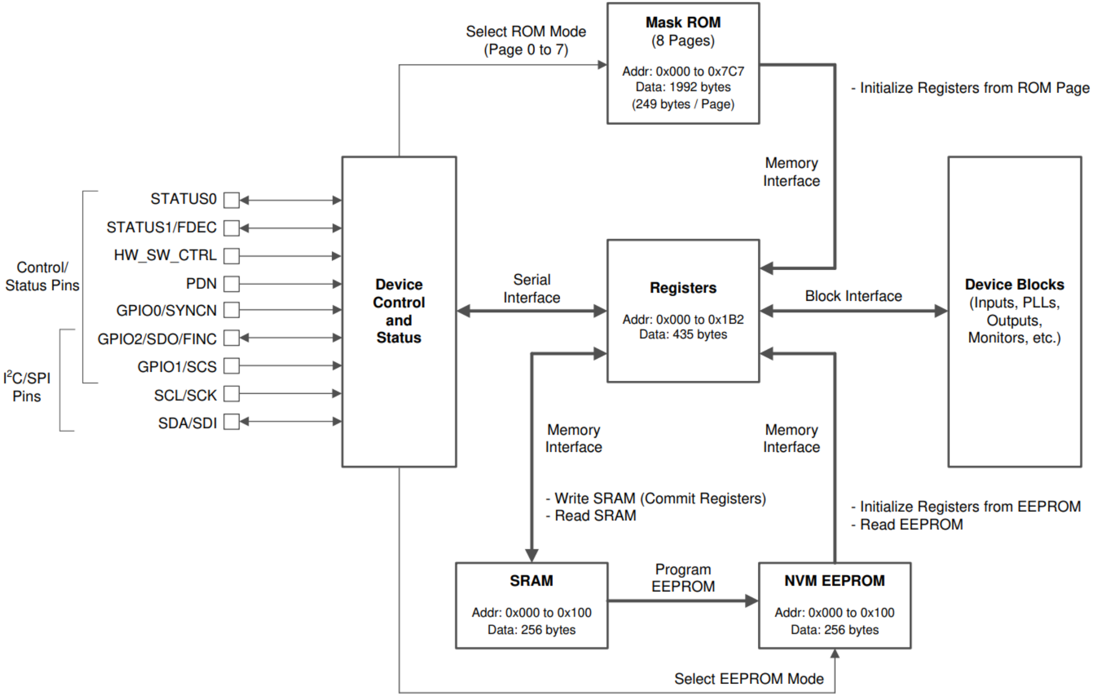


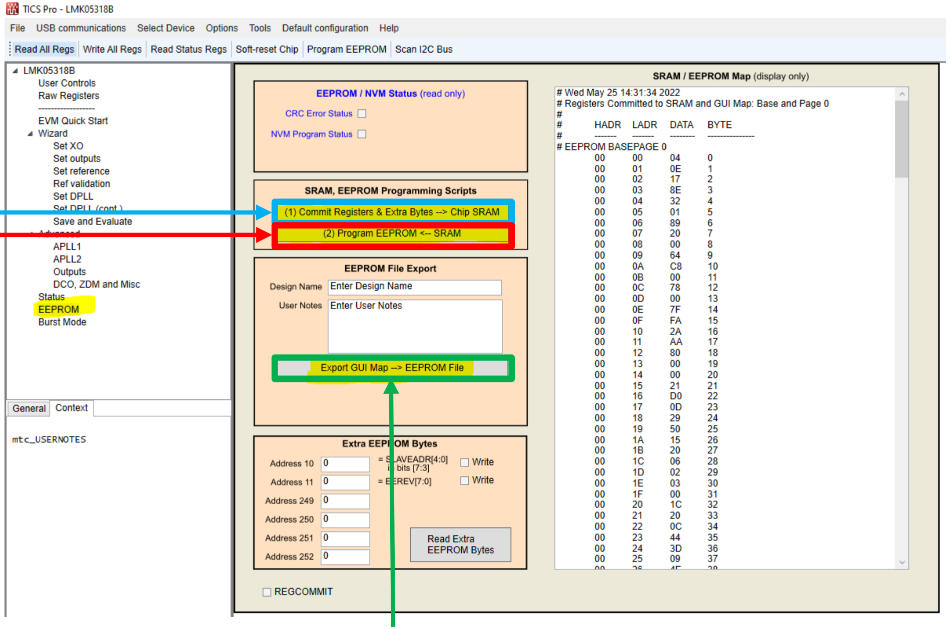
Figure 9-36. Device Control, Register, and Memory Interfaces

EEPROM Programming Methods

- Two methods available:
  - REGCOMMIT (recommended)
    - Stores current configuration (all active registers) to EEPROM.
    - TICS Pro uses this method to program when you press **Program EEPROM** button
  - SRAM Direct Write
    - Programs EEPROM one register/address at a time.
    - Use if you want to do an in-system update without disrupting clocks. On next restart new config takes effect.
    - Use to change the 5 MSBs of the I2C slave address, EEPROM Revision Number, and NVM Spare Bytes.
    - Does require you to utilize the saved ".EPR" file.

EEPROM Programming: REGCOMMIT Method (recommended)

1. Program active registers and confirm the current configuration outputs as desired
2. Commit active registers to SRAM
   a) Set REGCOMMIT, R157[6] = 1
3. Unlock EEPROM
   a) Set NVMUNLK, R164 = 0xEA
4. Erase EEPROM and initiate EEPROM programming
   a) Set NVMERASE and NVMPROG, R157[1:0] = 0x03
5. Wait for EEPROM programming to finish
   a) Poll NVMBUSY, R157[2], until cleared
   or wait ~ 500 ms
6. Lock EEPROM
   a) Set NVMUNLK, R164 = 0x00
7. Power cycle and check outputs to confirm EEPROM programming was successful



mtc_USERNOTES

CRC Error Status □

NVM Program Status □

Design Name: Enter Design Name

User Notes: Enter User Notes

Address 10: 0 = SLAVEADR[4:0] in bits [7:3] □ Write

Address 11: 0 = EEREV[7:0] □ Write

Address 249: 0

Address 250: 0

Address 251: 0

Address 252: 0

□ REGCOMMIT

```
# Wed May 25 14:31:34 2022
# Registers Committed to SRAM and GUI Map: Base and Page 0
#
#       HADR    LADR    DATA    BYTE
#       -------  -------  --------  ---------------
# EEPROM BASEPAGE 0
        00      00      04      0
        00      01      0E      1
        00      02      17      2
        00      03      8E      3
        00      04      32      4
        00      05      01      5
        00      06      89      6
        00      07      20      7
        00      08      00      8
        00      09      64      9
        00      0A      C8      10
        00      0B      00      11
        00      0C      78      12
        00      0D      00      13
        00      0E      7F      14
        00      0F      FA      15
        00      10      2A      16
        00      11      AA      17
        00      12      80      18
        00      13      00      19
        00      14      00      20
        00      15      21      21
        00      16      D0      22
        00      17      0D      23
        00      18      29      24
        00      19      50      25
        00      1A      15      26
        00      1B      20      27
        00      1C      06      28
        00      1D      02      29
        00      1E      03      30
        00      1F      00      31
        00      20      1C      32
        00      21      20      33
        00      22      0C      34
        00      23      44      35
        00      24      3D      36
        00      25      09      37
```

Export the GUI SRAM/EEPROM Map to an EEPROM file (.epr)

EEPROM Programming: SRAM Direct Write Method

1. Write the most significant five bits of the SRAM address to R159 (MEMADR byte 1).
2. Write the least significant eight bits of SRAM address to R160 (MEMADR byte 0).
3. Write the SRAM data byte to R162 (RAMDAT byte).
4. Unlock EEPROM
5. Erase EEPROM and initiate EEPROM programming.
6. Wait for EEPROM programming to finish
7. Lock EEPROM
8. Changes are available on next power cycle.

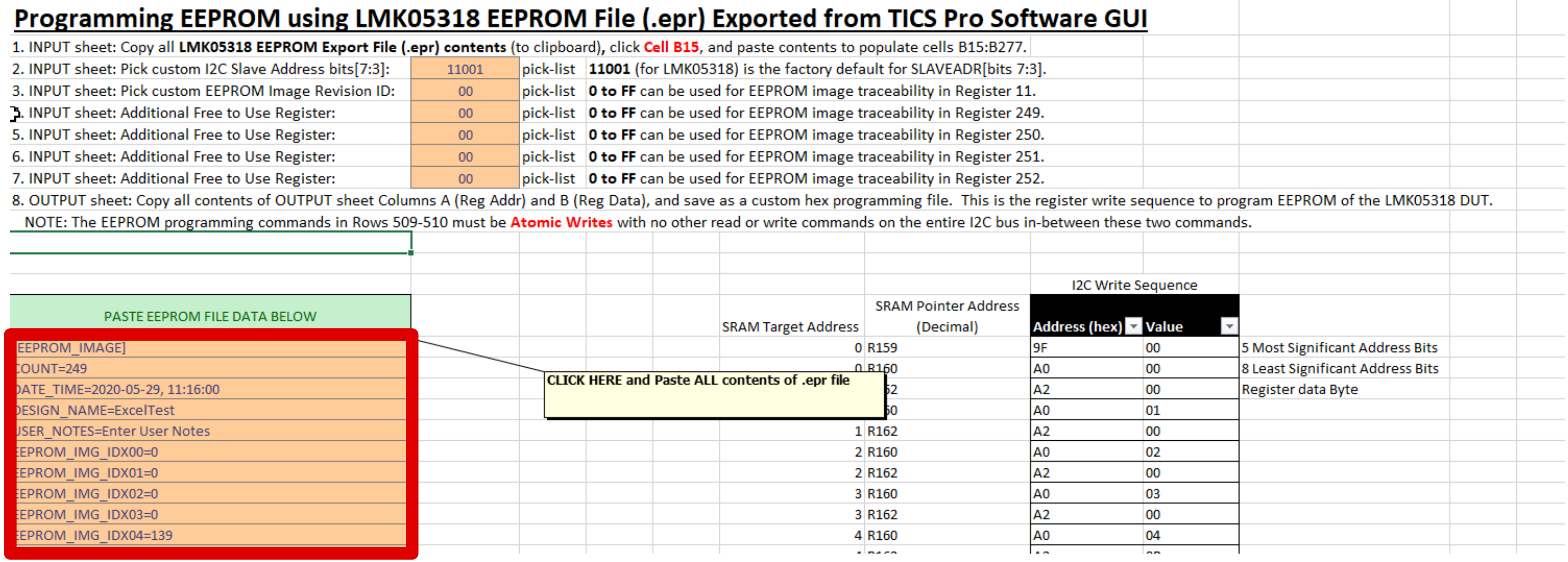
<u>Example for updating the EEREV:</u>

```
R159 (0x9F) = 0x00          #EEREV addrH is 0x00
R160 (0xA0) = 0x0B          # EEREV addrL is 0x0B
R162 (0xA2) = 0x02          # Set EEPROM Rev ID to 2
```

.epr to I2C Writes for EEPROM Programming

Using I2C tool (LMK05318B-EEPROM-I2C-Writes.xlsx)—pt 1

- Copy the .epr contents from the EEPROM map export (see slide 4)
- Paste them into the excel sheet



Programming EEPROM using LMK05318 EEPROM File (.epr) Exported from TICS Pro Software GUI

1. INPUT sheet: Copy all **LMK05318 EEPROM Export File (.epr) contents** (to clipboard), click **Cell B15**, and paste contents to populate cells B15:B277.

2. INPUT sheet: Pick custom I2C Slave Address bits[7:3]:	11001	pick-list	**11001** (for LMK05318) is the factory default for SLAVEADR[bits 7:3].
3. INPUT sheet: Pick custom EEPROM Image Revision ID:	00	pick-list	**0 to FF** can be used for EEPROM image traceability in Register 11.
4. INPUT sheet: Additional Free to Use Register:	00	pick-list	**0 to FF** can be used for EEPROM image traceability in Register 249.
5. INPUT sheet: Additional Free to Use Register:	00	pick-list	**0 to FF** can be used for EEPROM image traceability in Register 250.
6. INPUT sheet: Additional Free to Use Register:	00	pick-list	**0 to FF** can be used for EEPROM image traceability in Register 251.
7. INPUT sheet: Additional Free to Use Register:	00	pick-list	**0 to FF** can be used for EEPROM image traceability in Register 252.

8. OUTPUT sheet: Copy all contents of OUTPUT sheet Columns A (Reg Addr) and B (Reg Data), and save as a custom hex programming file. This is the register write sequence to program EEPROM of the LMK05318 DUT.

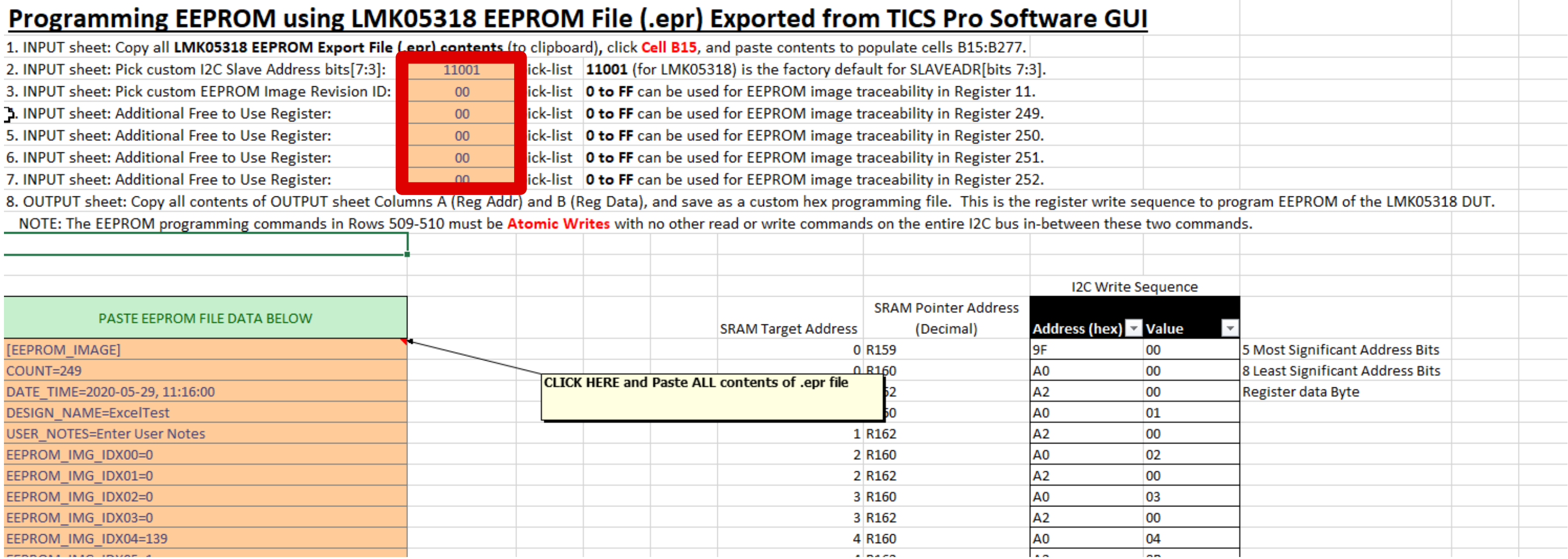
NOTE: The EEPROM programming commands in Rows 509-510 must be **Atomic Writes** with no other read or write commands on the entire I2C bus in-between these two commands.

CLICK HERE and Paste ALL contents of .epr file

PASTE EEPROM FILE DATA BELOW	SRAM Target Address	SRAM Pointer Address (Decimal)	I2C Write Sequence Address (hex)	 Value	
EEPROM_IMAGE]	0	R159	9F	00	5 Most Significant Address Bits
COUNT=249	0	R160	A0	00	8 Least Significant Address Bits
DATE_TIME=2020-05-29, 11:16:00		52	A2	00	Register data Byte
DESIGN_NAME=ExcelTest		50	A0	01	
USER_NOTES=Enter User Notes	1	R162	A2	00	
EEPROM_IMG_IDX00=0	2	R160	A0	02	
EEPROM_IMG_IDX01=0	2	R162	A2	00	
EEPROM_IMG_IDX02=0	3	R160	A0	03	
EEPROM_IMG_IDX03=0	3	R162	A2	00	
EEPROM_IMG_IDX04=139	4	R160	A0	04	

Using I2C tool (LMK05318B-EEPROM-I2C-Writes.xlsx)—pt 2

- Modify I2C address, EEREV, or NVM spare bytes as desired.



Programming EEPROM using LMK05318 EEPROM File (.epr) Exported from TICS Pro Software GUI

1. INPUT sheet: Copy all **LMK05318 EEPROM Export File (.epr) contents** (to clipboard), click **Cell B15**, and paste contents to populate cells B15:B277.

2. INPUT sheet: Pick custom I2C Slave Address bits[7:3]: 11001 ick-list **11001** (for LMK05318) is the factory default for SLAVEADR[bits 7:3].

3. INPUT sheet: Pick custom EEPROM Image Revision ID: 00 ick-list **0 to FF** can be used for EEPROM image traceability in Register 11.

4. INPUT sheet: Additional Free to Use Register: 00 ick-list **0 to FF** can be used for EEPROM image traceability in Register 249.

5. INPUT sheet: Additional Free to Use Register: 00 ick-list **0 to FF** can be used for EEPROM image traceability in Register 250.

6. INPUT sheet: Additional Free to Use Register: 00 ick-list **0 to FF** can be used for EEPROM image traceability in Register 251.

7. INPUT sheet: Additional Free to Use Register: 00 ick-list **0 to FF** can be used for EEPROM image traceability in Register 252.

8. OUTPUT sheet: Copy all contents of OUTPUT sheet Columns A (Reg Addr) and B (Reg Data), and save as a custom hex programming file. This is the register write sequence to program EEPROM of the LMK05318 DUT.

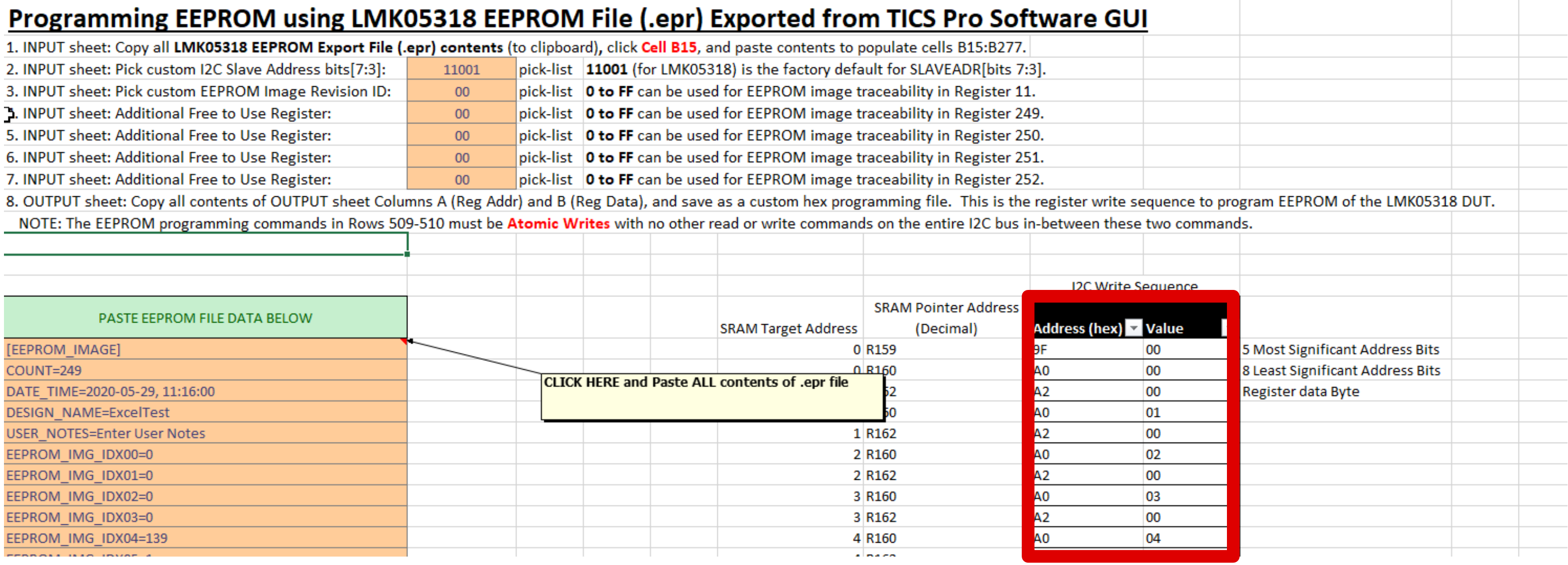
NOTE: The EEPROM programming commands in Rows 509-510 must be **Atomic Writes** with no other read or write commands on the entire I2C bus in-between these two commands.

CLICK HERE and Paste ALL contents of .epr file

			I2C Write Sequence		
PASTE EEPROM FILE DATA BELOW	SRAM Target Address	SRAM Pointer Address (Decimal)	Address (hex)	Value	
[EEPROM_IMAGE]	0	R159	9F	00	5 Most Significant Address Bits
COUNT=249	0	R160	A0	00	8 Least Significant Address Bits
DATE_TIME=2020-05-29, 11:16:00		62	A2	00	Register data Byte
DESIGN_NAME=ExcelTest		60	A0	01	
USER_NOTES=Enter User Notes	1	R162	A2	00	
EEPROM_IMG_IDX00=0	2	R160	A0	02	
EEPROM_IMG_IDX01=0	2	R162	A2	00	
EEPROM_IMG_IDX02=0	3	R160	A0	03	
EEPROM_IMG_IDX03=0	3	R162	A2	00	
EEPROM_IMG_IDX04=139	4	R160	A0	04	

Using I2C tool (LMK05318B-EEPROM-I2C-Writes.xlsx)—pt 3

- Attain I2C write sequence. Note that this uses the SRAM direct write method to write the entire register map to EEPROM.



Programming EEPROM using LMK05318 EEPROM File (.epr) Exported from TICS Pro Software GUI

1. INPUT sheet: Copy all **LMK05318 EEPROM Export File (.epr) contents** (to clipboard), click **Cell B15**, and paste contents to populate cells B15:B277.

2. INPUT sheet: Pick custom I2C Slave Address bits[7:3]:	11001	pick-list	**11001** (for LMK05318) is the factory default for SLAVEADR[bits 7:3].
3. INPUT sheet: Pick custom EEPROM Image Revision ID:	00	pick-list	**0 to FF** can be used for EEPROM image traceability in Register 11.
4. INPUT sheet: Additional Free to Use Register:	00	pick-list	**0 to FF** can be used for EEPROM image traceability in Register 249.
5. INPUT sheet: Additional Free to Use Register:	00	pick-list	**0 to FF** can be used for EEPROM image traceability in Register 250.
6. INPUT sheet: Additional Free to Use Register:	00	pick-list	**0 to FF** can be used for EEPROM image traceability in Register 251.
7. INPUT sheet: Additional Free to Use Register:	00	pick-list	**0 to FF** can be used for EEPROM image traceability in Register 252.

8. OUTPUT sheet: Copy all contents of OUTPUT sheet Columns A (Reg Addr) and B (Reg Data), and save as a custom hex programming file. This is the register write sequence to program EEPROM of the LMK05318 DUT.

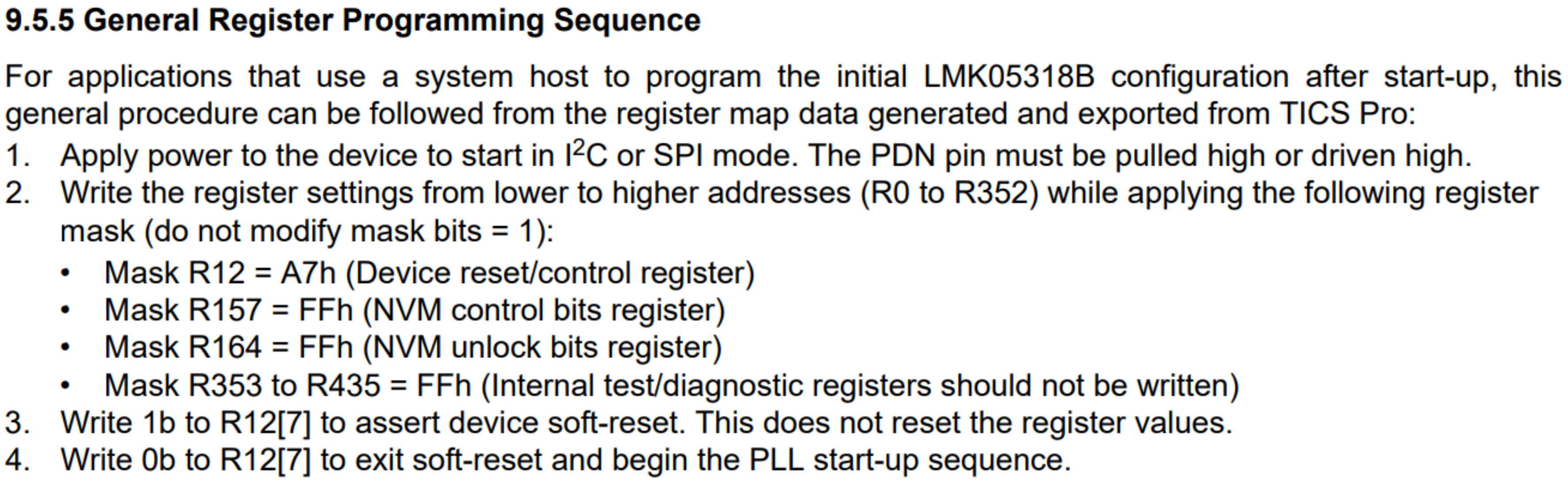
NOTE: The EEPROM programming commands in Rows 509-510 must be **Atomic Writes** with no other read or write commands on the entire I2C bus in-between these two commands.

CLICK HERE and Paste ALL contents of .epr file

PASTE EEPROM FILE DATA BELOW	SRAM Target Address	SRAM Pointer Address (Decimal)	I2C Write Sequence Address (hex)	Value	
[EEPROM_IMAGE]	0	R159	9F	00	5 Most Significant Address Bits
COUNT=249	0	R160	A0	00	8 Least Significant Address Bits
DATE_TIME=2020-05-29, 11:16:00		62	A2	00	Register data Byte
DESIGN_NAME=ExcelTest		60	A0	01	
USER_NOTES=Enter User Notes	1	R162	A2	00	
EEPROM_IMG_IDX00=0	2	R160	A0	02	
EEPROM_IMG_IDX01=0	2	R162	A2	00	
EEPROM_IMG_IDX02=0	3	R160	A0	03	
EEPROM_IMG_IDX03=0	3	R162	A2	00	
EEPROM_IMG_IDX04=139	4	R160	A0	04	

REGCOMMIT Register Details

Step 1: Program the registers with desired configuration.

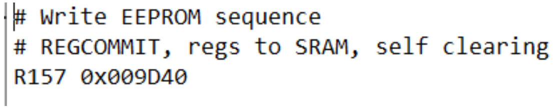


9.5.5 General Register Programming Sequence

For applications that use a system host to program the initial LMK05318B configuration after start-up, this general procedure can be followed from the register map data generated and exported from TICS Pro:

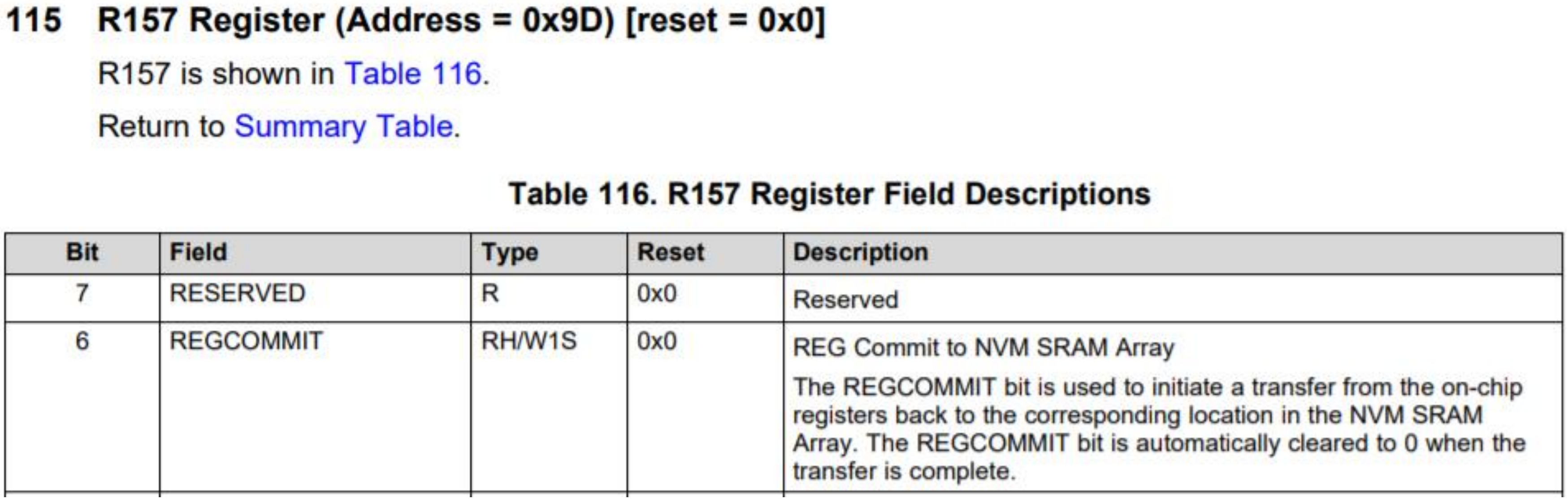
1. Apply power to the device to start in I^{2}C or SPI mode. The PDN pin must be pulled high or driven high.
2. Write the register settings from lower to higher addresses (R0 to R352) while applying the following register mask (do not modify mask bits = 1):
   - Mask R12 = A7h (Device reset/control register)
   - Mask R157 = FFh (NVM control bits register)
   - Mask R164 = FFh (NVM unlock bits register)
   - Mask R353 to R435 = FFh (Internal test/diagnostic registers should not be written)
3. Write 1b to R12[7] to assert device soft-reset. This does not reset the register values.
4. Write 0b to R12[7] to exit soft-reset and begin the PLL start-up sequence.

Step 2: Commit registers to SRAM



```
# Write EEPROM sequence
# REGCOMMIT, regs to SRAM, self clearing
R157 0x009D40
```

→ REGCOMMIT = 1



115 R157 Register (Address = 0x9D) [reset = 0x0]

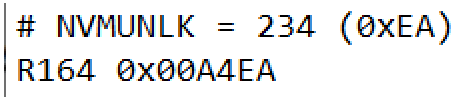
R157 is shown in Table 116.

Return to Summary Table.

Table 116. R157 Register Field Descriptions

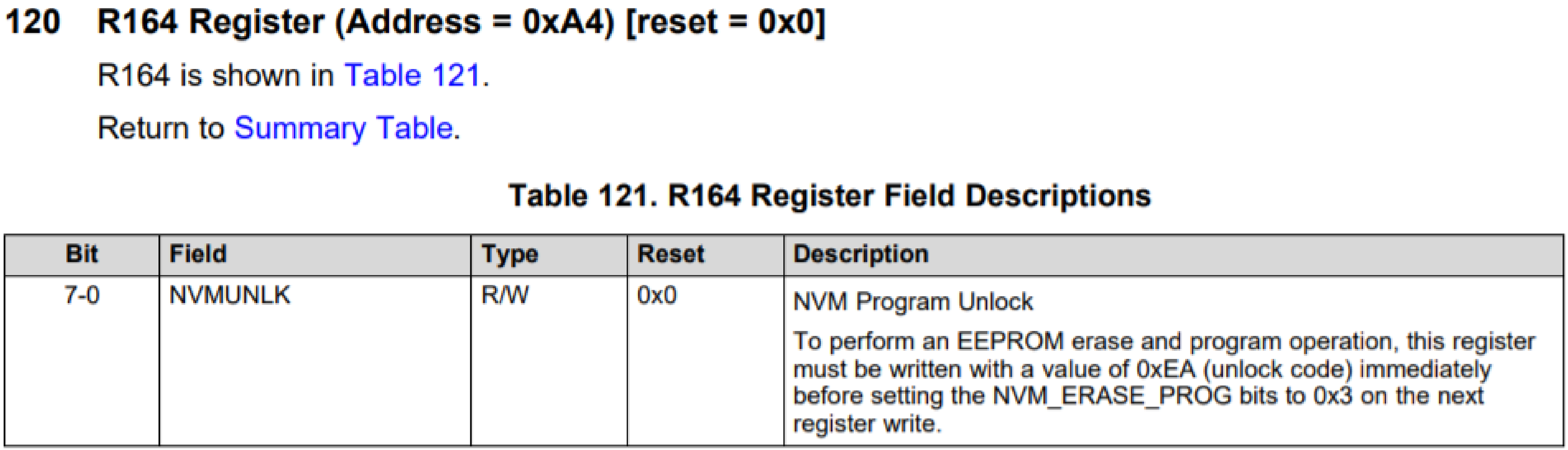
Bit	Field	Type	Reset	Description
7	RESERVED	R	0x0	Reserved
6	REGCOMMIT	RH/W1S	0x0	REG Commit to NVM SRAM Array The REGCOMMIT bit is used to initiate a transfer from the on-chip registers back to the corresponding location in the NVM SRAM Array. The REGCOMMIT bit is automatically cleared to 0 when the transfer is complete.

Step 3a. Unlock EEPROM



```
# NVMUNLK = 234 (0xEA)
R164 0x00A4EA
```

NVMUNLK = 0xEA



120 R164 Register (Address = 0xA4) [reset = 0x0]

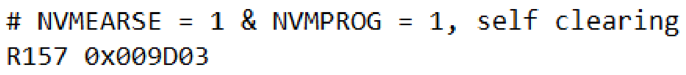
R164 is shown in Table 121.

Return to Summary Table.

Table 121. R164 Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	NVMUNLK	R/W	0x0	NVM Program Unlock To perform an EEPROM erase and program operation, this register must be written with a value of 0xEA (unlock code) immediately before setting the NVM_ERASE_PROG bits to 0x3 on the next register write.

Step 3b. Initiate EEPROM programming



```
# NVMEARSE = 1 & NVMPROG = 1, self clearing
R157 0x009D03
```

NVMERASE = 1
NVMPROG = 1

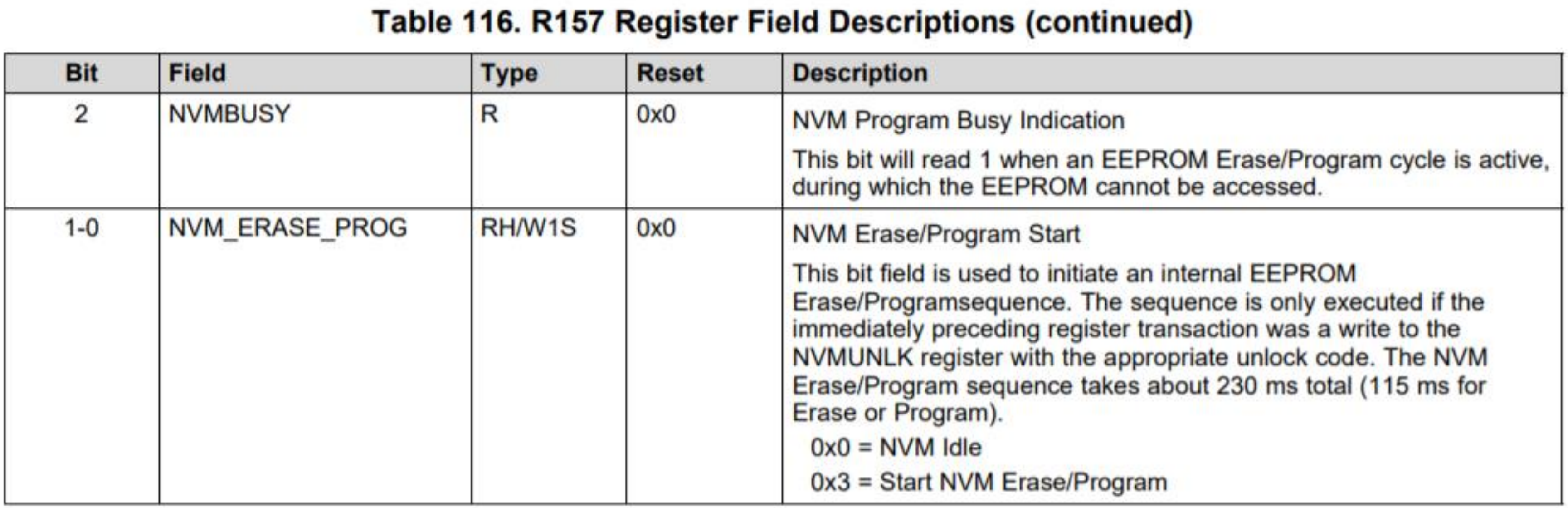


Table 116. R157 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	NVMBUSY	R	0x0	NVM Program Busy Indication This bit will read 1 when an EEPROM Erase/Program cycle is active, during which the EEPROM cannot be accessed.
1-0	NVM_ERASE_PROG	RH/W1S	0x0	NVM Erase/Program Start This bit field is used to initiate an internal EEPROM Erase/Programsequence. The sequence is only executed if the immediately preceding register transaction was a write to the NVMUNLK register with the appropriate unlock code. The NVM Erase/Program sequence takes about 230 ms total (115 ms for Erase or Program). 0x0 = NVM Idle 0x3 = Start NVM Erase/Program

Step 3c. Wait for EEPROM programming to finish

```
# Poll until NVMBUSY = 0
R157 0x009D04
```

Wait until NVMBUSY = 0

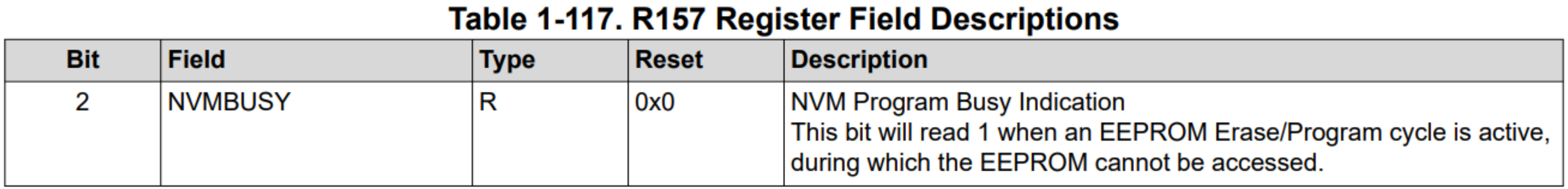
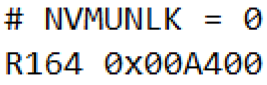


Table 1-117. R157 Register Field Descriptions

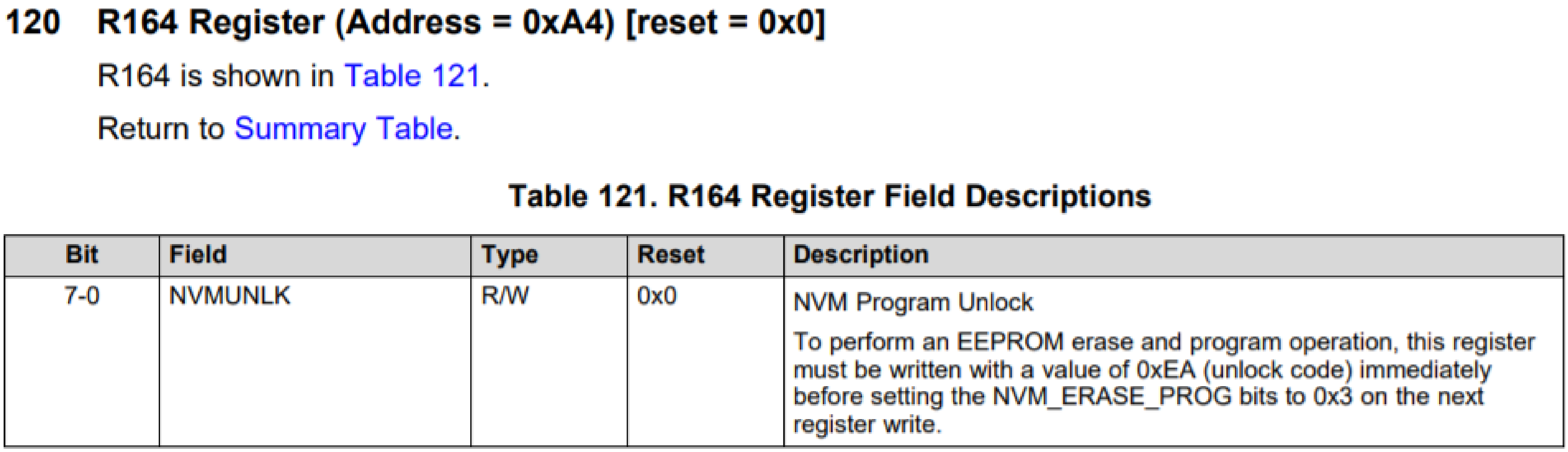
Bit	Field	Type	Reset	Description
2	NVMBUSY	R	0x0	NVM Program Busy Indication This bit will read 1 when an EEPROM Erase/Program cycle is active, during which the EEPROM cannot be accessed.

Step 3d. Lock EEPROM



```
# NVMUNLK = 0
R164 0x00A400
```

→ NVMUNLK = 0



120 R164 Register (Address = 0xA4) [reset = 0x0]

R164 is shown in Table 121.

Return to Summary Table.

Table 121. R164 Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	NVMUNLK	R/W	0x0	NVM Program Unlock To perform an EEPROM erase and program operation, this register must be written with a value of 0xEA (unlock code) immediately before setting the NVM_ERASE_PROG bits to 0x3 on the next register write.