life.augmented

AN6052

Application note

SPSB100 NVM configuration

Introduction

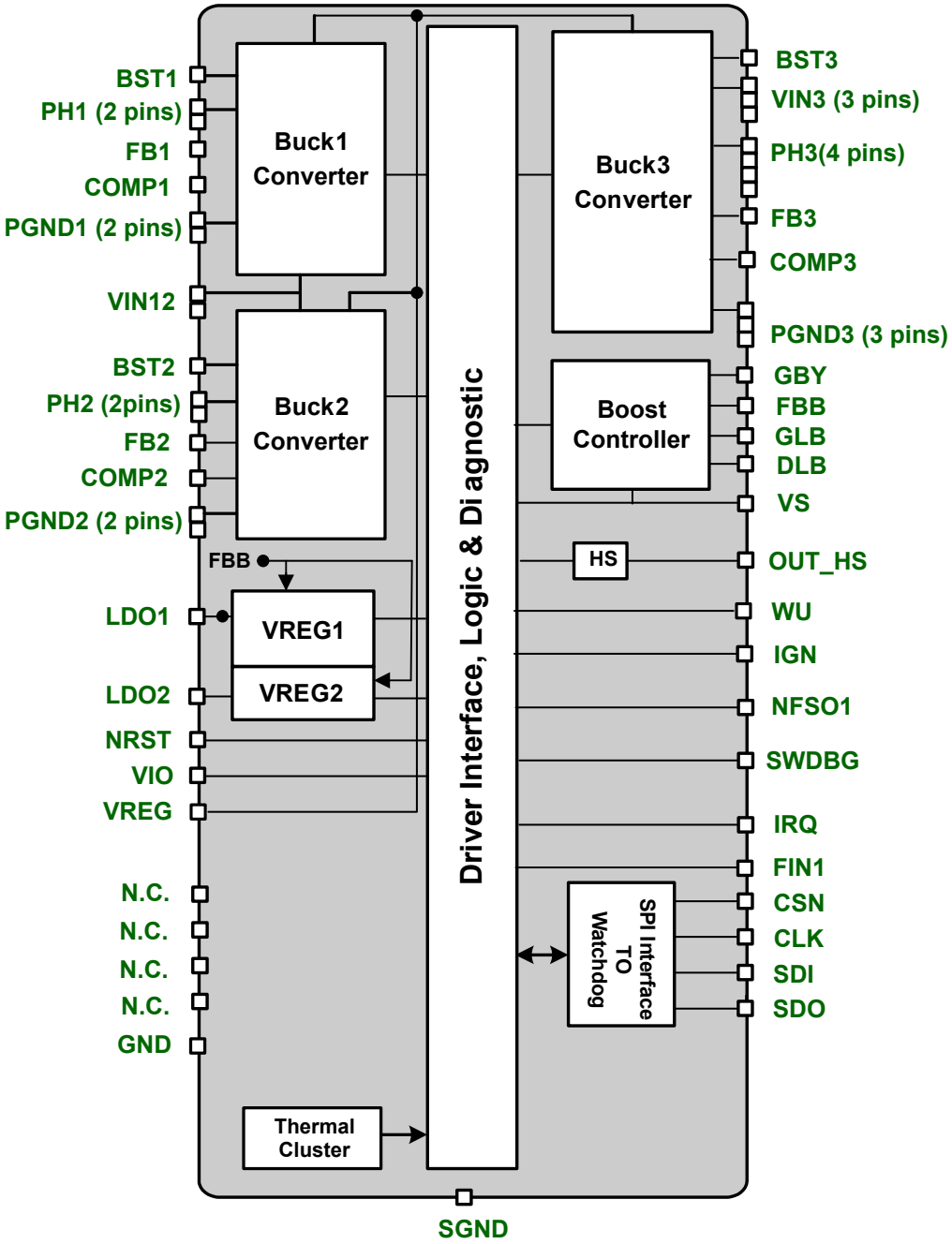
This document is intended to integrate the information provided in the SPSB100 product datasheet, to facilitate the correct procedure to define a customized NVM configuration. A specific focus is reserved to the needed steps to follow to properly write the new content into NVM.

1 Features

One of the main advantages in using a NVM solution is the possibility to reconfigure the different parameters, like output voltage, frequency and power-up power-down sequence just to consider the one with the highest impact on device functionality.

The next chapters will describe the single NVM registers and the possible configuration for each of them and the procedure to follow to properly write the desired content into the NVM.

Figure 1. Functional block diagram



2 Application use cases for programming

The USER-NVM programming requires the following setup:

- 9.4 - 20 V power supply at KL30, KL15
- 9.4 - 20 V signal on SWDBG (see the datasheet specification for allowed voltage range)
- Power supply for VIO (usually taken from SPI HOST side)
- SPI host tool/device

All power outputs of SPSB100 are inactive in programming phase and get active after programming procedure is over and power-up is completed.

Two use cases for USER-NVM programming can be considered.

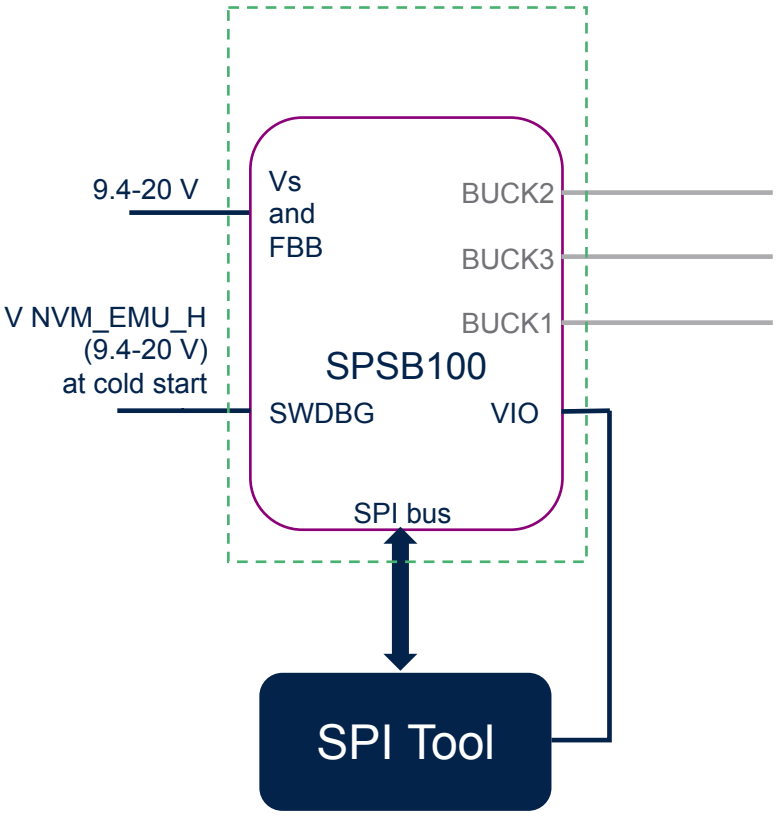
- Standalone programming
- In-application programming

Where standalone programming means that there is nothing connected to the SPSB100 power outputs during programming, that is, SPSB100 is programmed, for example, in a dedicated socketed tool.

2.1 Standalone device programming

Since there is no interference with around parts, standalone device programming does not introduce any specific limitation or “pay attention” points.

Figure 2. Block scheme for device standalone programming



2.2 In-application device programming

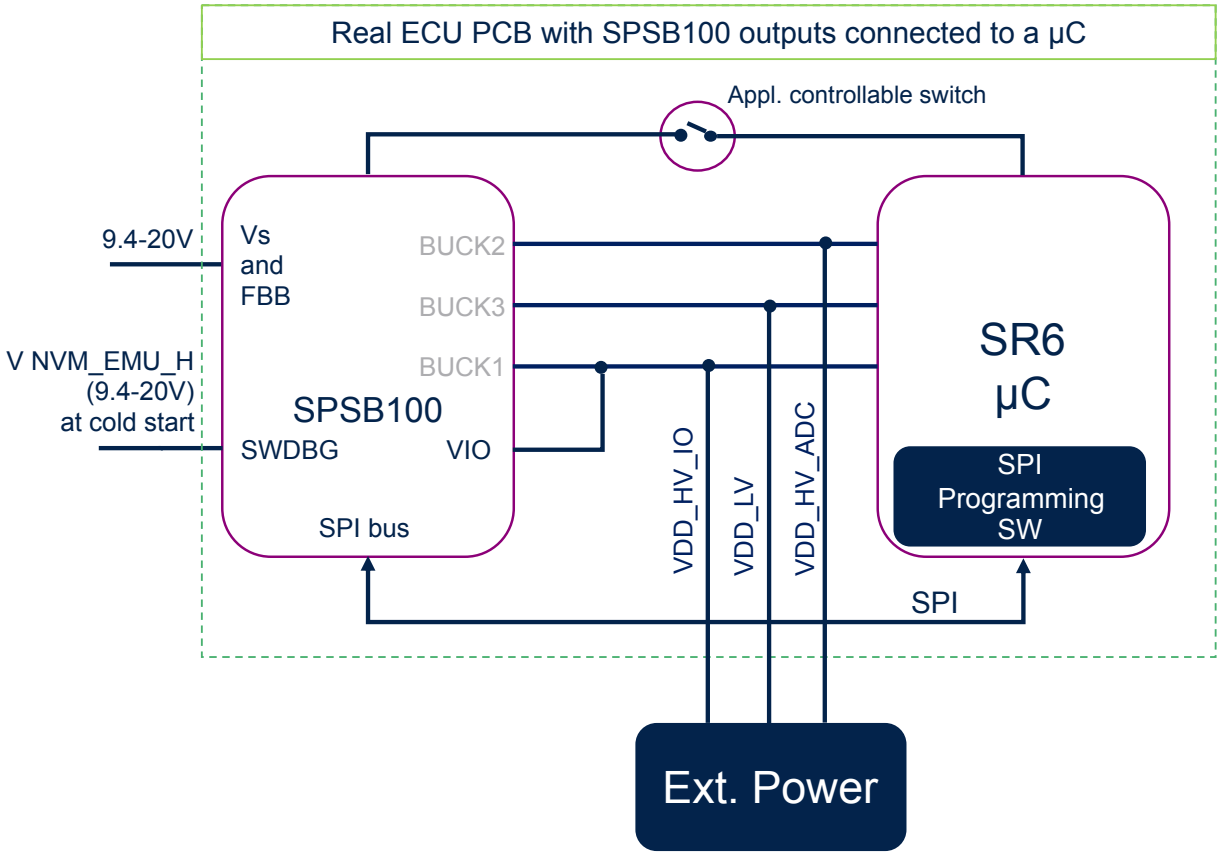
In principle, there are two possible ways to program SPSB100 in application.

2.2.1 By SPI host present in application µC

The main limitation of this way is that there is no power supply for µC present, so

- Power supply needs to be provided externally and
- Programming software needs to be flashed into the uC

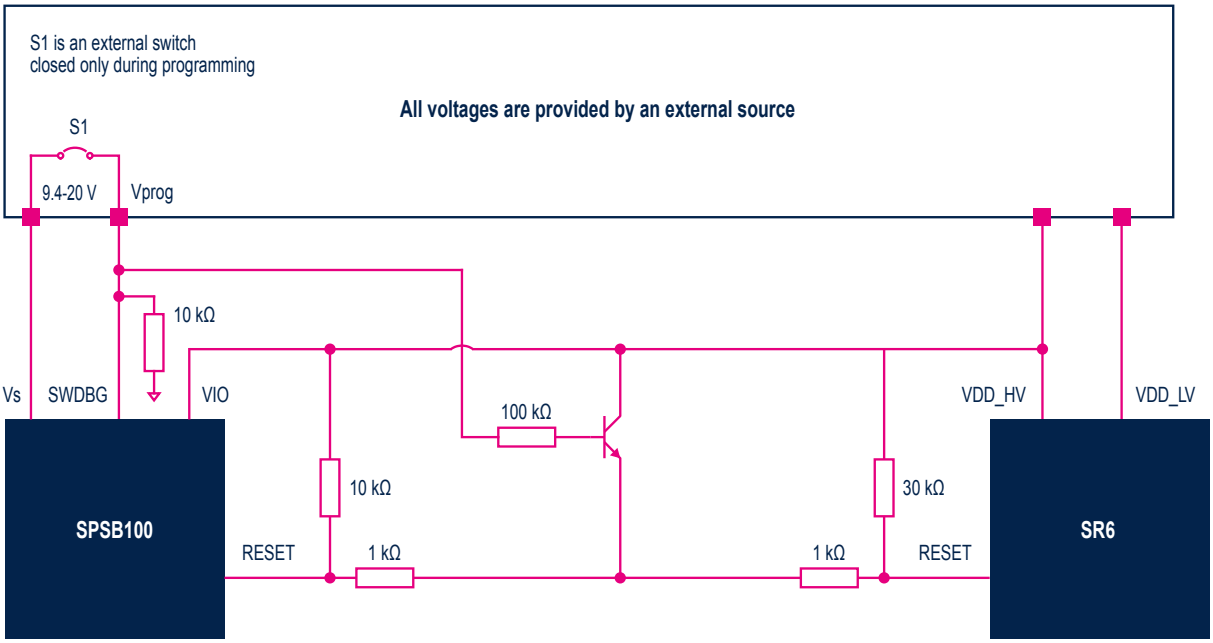
Figure 3. Block scheme for in-application programming by µC



Note:

- *Picture refers to default configuration of SPSB100 + SR6.*
- *SPSB100 reset line is held down in SWDBG mode. But µC cannot run any programming code while being under reset. It is needed to implement a hardware solution to disconnect the reset line between SPSB100 and µC during programming. An example of a hardware solution is depicted in the Figure 4.*

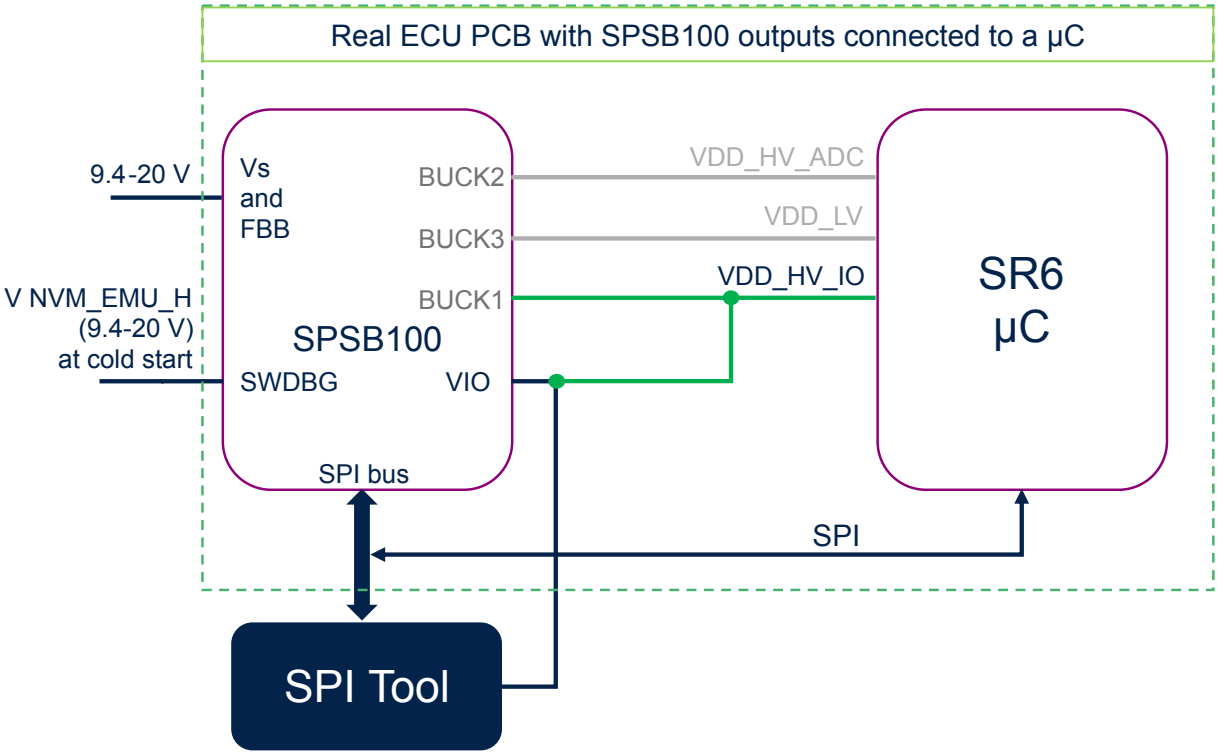
Figure 4. Hardware solution example to enable in-application programming via app. µC



2.2.2 By external SPI tool

Since SPSB100 is connected to µC in real application via SPI, this connection is present also in case the programming shall happen. It implies µC can get powered through internal ESD diodes on SPI IOs. Therefore the VDD_HV_IO power rail of µC shall be connected to VIO which eliminates the power injection via ESD diodes. It implies that SPI programming tool IOs should be connected only after VIO supply.

Figure 5. Block scheme for device in-application hardware programming



3 NVM configuration

In this chapter the USER-NVM modification procedure is explained.

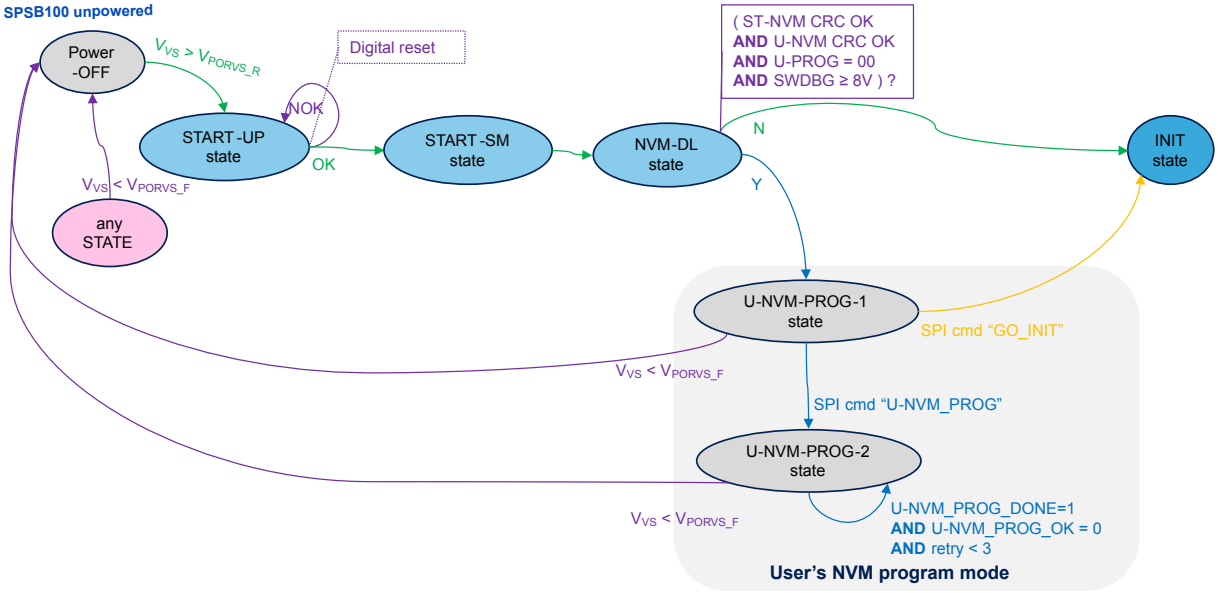
USER-NVM programmable control registers inside SPSB100 are mapped at SPI address from 0x0Ah to 0x19h. A detailed register description is part of the Section 6: NVM registers DATA field details.

SPSB100 integrates non-volatile memory (NVM) to store ST factory trim data and customer's personalization data. Trim data are stored in ST-NVM space while customer's data are stored in USER-NVM space. The Figure 6 describes the state diagram of the FSM controlling NVM operations.

The following color code is used to indicate the offered possibilities:

- Green: SPSB100 start up with valid data stored in ST-NVM and USER-NVM (USER-NVM previously programmed by ST or by the customer)
- Blue: SPSB100 enters in USER-NVM programming procedure
- Orange: USER-NVM emulation feature is used

Figure 6. NVM configuration state machine



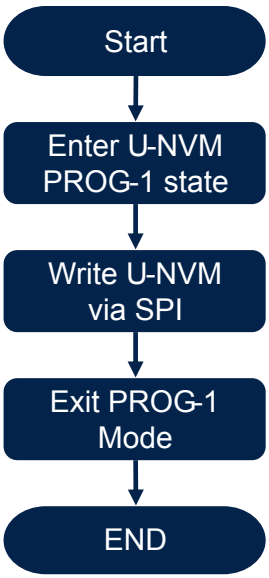
3.1 USER-NVM programming procedure

The programming procedure can be done just one time. Once programming is successfully completed, subsequent attempts to program the NVM will be blocked.

The steps to follow to successfully complete the programming procedure can be divided into 3 macro sections, as reported also in the Figure 7:

1. Enter USER-NVM PROG-1 state;
2. Modify the USER-NVM content through SPI access;
3. Exit the USER-NVM programming mode.

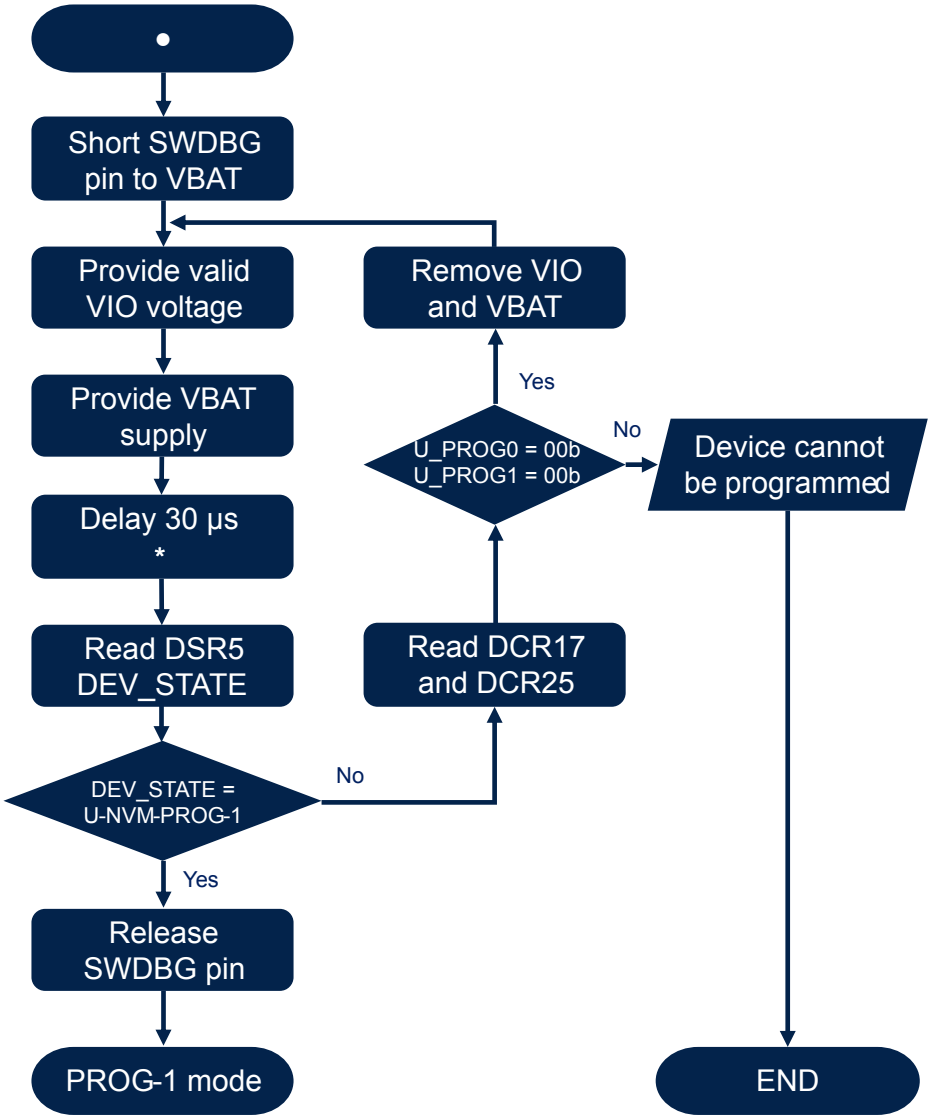
Figure 7. USER-NVM programming procedure



3.1.1 Entering USER-NVM PROG-1 state

The first step is to enter USER-NVM PROG-1.

Figure 8. Enter USER-NVM PROG-1 procedure



* Add more delay to guarantee minimally 30 µs after 9.4 V is present before DSR5 reading.

Steps to follow to enter USER-NVM PROG-1:

1. SWDBG pin forced to a voltage higher than VNVM_EMU_H (usually it can be shorted to VBAT but considering the 20 V as max);
2. VIO line needs to be supplied externally (regulator outputs are not available);
3. Apply 9.4 - 20 V (voltage should be present at VS and FBB pins);
4. SWDBG pin can be released (its value is ignored till the next power-up);

5. As a last step, it is needed to check the device status through the DSR5 reg.
   a. If the status readback is USER-NVM PROG-1, the procedure is correctly completed. Now, the device is prepared to receive USER-NVM content into RAM;
   b. Instead, if the status is different from USER-NVM PROG-1, there are two possibilities:
      i. The procedure to enter USER-NVM PROG-1 is not correct. In such case DCR17 U-PROG0[1:0] and the DCR25 U-PROG1[1:0] fields are ‘00’ (both). It is important to guarantee that the SWDBG pin is at a value higher than VNVM_EMU_H (8 V) prior to device logic startup.

Note: Due to excessive capacitive load, even if the SWDBG is shorted to VBAT, it may ramp up too slowly and the result could be not to enter the SWDBG mode. Device logic startup threshold and threshold to enter SWDBG mode are different (SWDBG mode threshold is higher).

      ii. The device is already programmed or faulty. In this case, the DCR17 U-PROG0[1:0] and the DCR25 U-PROG1[1:0] (both) fields are different from ‘00’.

Note: Be aware that if a device cannot get into the USER-NVM-PROG-1 state and goes into INIT state, it starts up regulators and provides voltages.

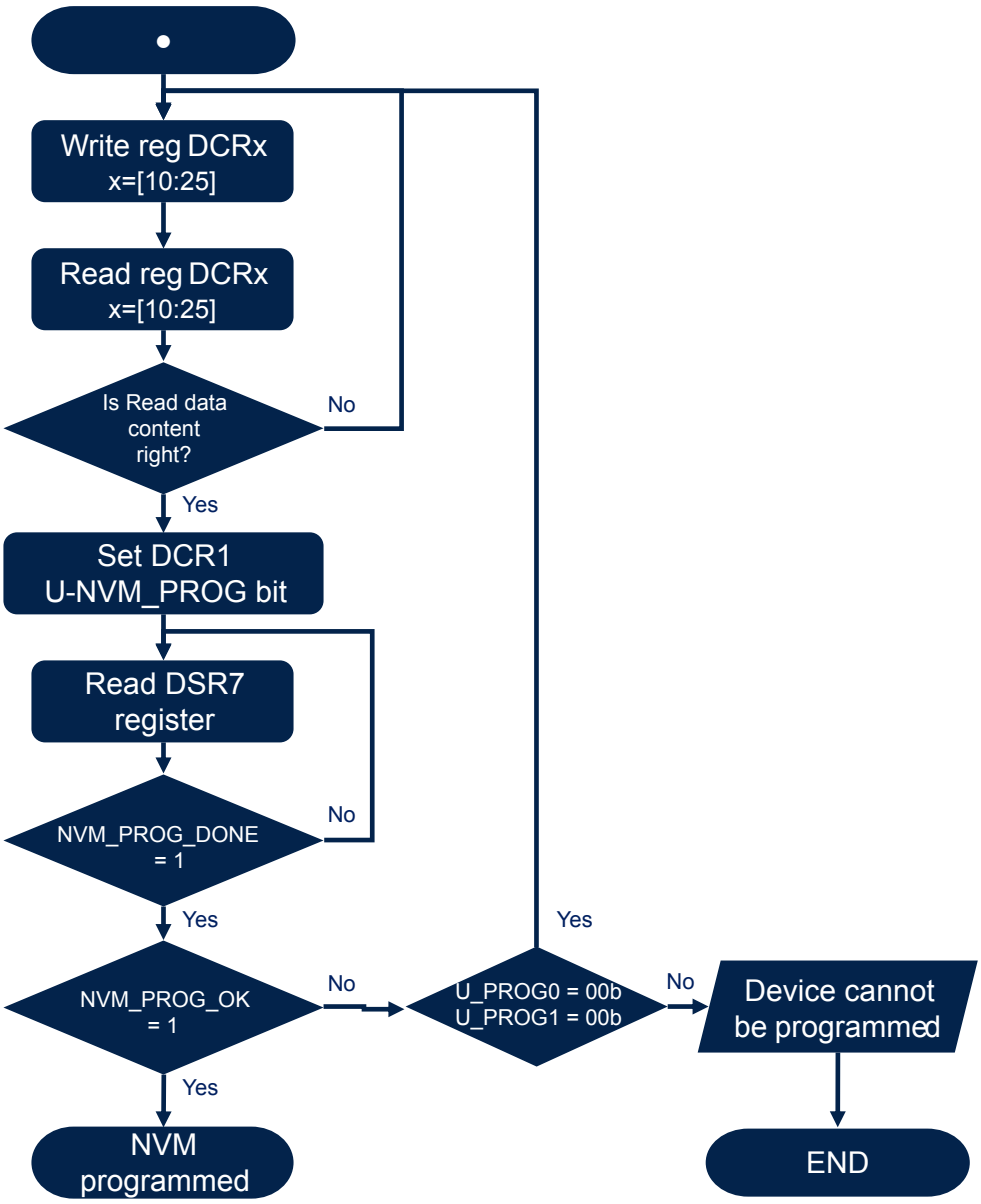
3.1.2 Write USER-NVM content

In this step the new USER-NVM configuration is:

- Transferred into RAM
- Written to NVM

The Figure 9 shows the actions to write and check that the NVM was properly written.

Figure 9. Write USER-NVM procedure



Steps to write USER-NVM:

1. Write device DCR registers from 10 to 25 through SPI access;
   a. Write '11b' into DCR17 U_PROG0 and into DCR25 U_PROG1:

Note: All USER-NVM registers have to be written, even the ones not intended to be modified.

2. Read DCR registers to double check the content;
3. Set the U-NVM_PROG bit in DCR1 using SPI access. This starts the USER-NVM programming from RAM. Device changes state into USER-NVM-PROG-2;
4. When the USER-NVM programming is completed, the status bit NVM_PROG_DONE in DSR7 is set;

5. Once NVM_PROG_DONE is set, read DSR7 NVM_PROG_OK to verify that the USER-NVM programming has been completed correctly;

Note: USER-NVM programming is followed by the USER-NVM verification (self-CRC check).

6. In the case NVM_PROG_OK is 0, read DCR17 U_PROG0[1:0] and DCR25 U_PROG1[1:0]:
   a. If both bitfields are '00', it is necessary to repeat the procedure.
   b. If at least one of the bits is '1', the device is faulty and should not be used.

3.1.3 Exit the USER-NVM programming mode

The last step to complete the USER-NVM modification procedure is to exit from the USER-NVM-PROG-2 mode. In this case, the only option available is to unplug the battery.

3.2 USER-NVM emulation

It is possible to emulate the USER-NVM content using the data stored in RAM image.

This can only be done when the USER-NVM has not been programmed yet.

Steps to follow for NVM emulation:

1. Enter programming mode. It is the same procedure as to enter USER-NVM-PROG1 described by the Section 3.1.1: Entering USER-NVM PROG-1 state;
2. Write device DCR registers from 10 to 24 through SPI access:
   a. No need to care about DCR17 U_PROG0 and DCR25 U_PROG1 fields.
3. Read DCR registers to double check the content;
4. Write DCR8 GO_INIT bit to 1 by SPI. The device transits into the INIT state.

The user can emulate the USER-NVM as many times as he wants as far as the USER-NVM is not programmed yet. Once USER-NVM is programmed, it cannot be emulated and the device proceeds with valid data stored in USER-NVM. For default USER-NVM content, see the Section 4: NVM configuration examples.

Note: The USER-NVM emulated values are lost at chip reset or power supply removal.

4 NVM configuration examples

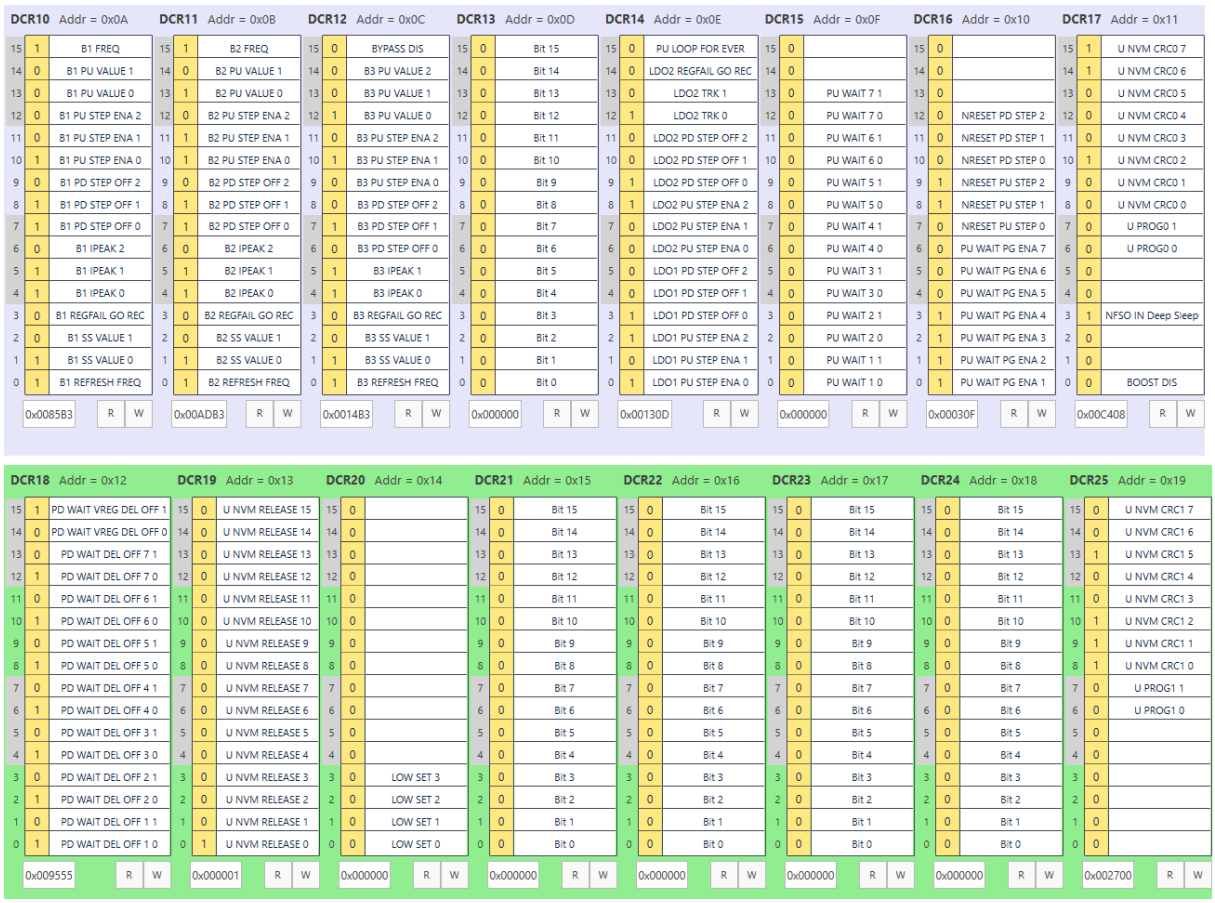
4.1 USER-NVM configuration

Content of the USER-NVM default configuration:

```
Register = 0x0A, data16 = 0x85B3
Register = 0x0B, data16 = 0xADB3
Register = 0x0C, data16 = 0x14B3
Register = 0x0D, data16 = 0x0000
Register = 0x0E, data16 = 0x130D
Register = 0x0F, data16 = 0x0000
Register = 0x10, data16 = 0x030F
Register = 0x11, data16 = 0xC408
Register = 0x12, data16 = 0x9555
Register = 0x13, data16 = 0x0001
Register = 0x14, data16 = 0x0000
Register = 0x15, data16 = 0x0000
Register = 0x16, data16 = 0x0000
Register = 0x17, data16 = 0x0000
Register = 0x18, data16 = 0x0000
Register = 0x19, data16 = 0x2700
```

Note: Register number and 16 bits data deconstructed from read SPI word.

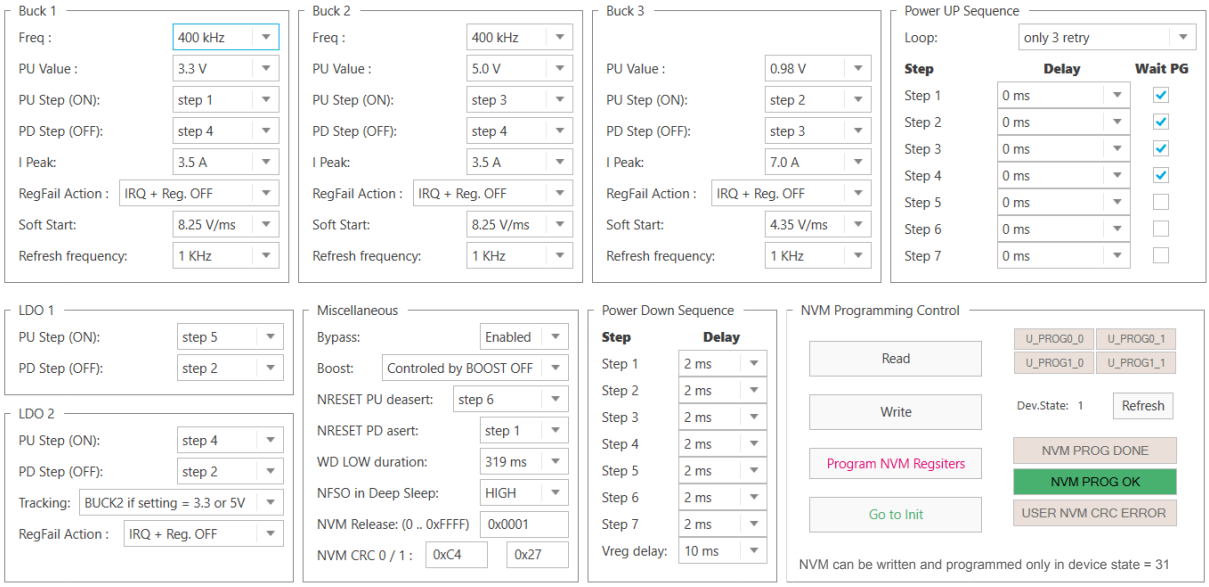
Figure 10. Default USER-NVM registers content



DCR10 Addr = 0x0A		DCR11 Addr = 0x0B		DCR12 Addr = 0x0C		DCR13 Addr = 0x0D		DCR14 Addr = 0x0E		DCR15 Addr = 0x0F		DCR16 Addr = 0x10		DCR17 Addr = 0x11	
15	1 B1 FREQ	15	1 B2 FREQ	15	0 BYPASS DIS	15	0 Bit 15	15	0 PU LOOP FOR EVER	15	0	15	0	15	1 U NVM CRC0 7
14	0 B1 PU VALUE 1	14	0 B2 PU VALUE 1	14	0 B3 PU VALUE 2	14	0 Bit 14	14	0 LDO2 REGFAIL GO REC	14	0	14	0	14	1 U NVM CRC0 6
13	0 B1 PU VALUE 0	13	1 B2 PU VALUE 0	13	0 B3 PU VALUE 1	13	0 Bit 13	13	0 LDO2 TRK 1	13	0 PU WAIT 7 1	13	0	13	0 U NVM CRC0 5
12	0 B1 PU STEP ENA 2	12	0 B2 PU STEP ENA 2	12	1 B3 PU VALUE 0	12	0 Bit 12	12	1 LDO2 TRK 0	12	0 PU WAIT 7 0	12	0 NRESET PD STEP 2	12	0 U NVM CRC0 4
11	0 B1 PU STEP ENA 1	11	1 B2 PU STEP ENA 1	11	0 B3 PU STEP ENA 2	11	0 Bit 11	11	0 LDO2 PD STEP OFF 2	11	0 PU WAIT 6 1	11	0 NRESET PD STEP 1	11	0 U NVM CRC0 3
10	1 B1 PU STEP ENA 0	10	1 B2 PU STEP ENA 0	10	1 B3 PU STEP ENA 1	10	0 Bit 10	10	0 LDO2 PD STEP OFF 1	10	0 PU WAIT 6 0	10	0 NRESET PD STEP 0	10	1 U NVM CRC0 2
9	0 B1 PD STEP OFF 2	9	0 B2 PD STEP OFF 2	9	0 B3 PU STEP ENA 0	9	0 Bit 9	9	1 LDO2 PD STEP OFF 0	9	0 PU WAIT 5 1	9	1 NRESET PU STEP 2	9	0 U NVM CRC0 1
8	1 B1 PD STEP OFF 1	8	1 B2 PD STEP OFF 1	8	0 B3 PD STEP OFF 2	8	0 Bit 8	8	1 LDO2 PU STEP ENA 2	8	0 PU WAIT 5 0	8	1 NRESET PU STEP 1	8	0 U NVM CRC0 0
7	1 B1 PD STEP OFF 0	7	1 B2 PD STEP OFF 0	7	1 B3 PD STEP OFF 1	7	0 Bit 7	7	0 LDO2 PU STEP ENA 1	7	0 PU WAIT 4 1	7	0 NRESET PU STEP 0	7	0 U PROG0 1
6	0 B1 IPEAK 2	6	0 B2 IPEAK 2	6	0 B3 PD STEP OFF 0	6	0 Bit 6	6	0 LDO2 PU STEP ENA 0	6	0 PU WAIT 4 0	6	0 PU WAIT PG ENA 7	6	0 U PROG0 0
5	1 B1 IPEAK 1	5	1 B2 IPEAK 1	5	1 B3 IPEAK 1	5	0 Bit 5	5	0 LDO1 PD STEP OFF 2	5	0 PU WAIT 3 1	5	0 PU WAIT PG ENA 6	5	0
4	1 B1 IPEAK 0	4	1 B2 IPEAK 0	4	1 B3 IPEAK 0	4	0 Bit 4	4	0 LDO1 PD STEP OFF 1	4	0 PU WAIT 3 0	4	0 PU WAIT PG ENA 5	4	0
3	0 B1 REGFAIL GO REC	3	0 B2 REGFAIL GO REC	3	0 B3 REGFAIL GO REC	3	0 Bit 3	3	1 LDO1 PD STEP OFF 0	3	0 PU WAIT 2 1	3	1 PU WAIT PG ENA 4	3	1 NFSO IN Deep Sleep
2	0 B1 SS VALUE 1	2	0 B2 SS VALUE 1	2	0 B3 SS VALUE 1	2	0 Bit 2	2	1 LDO1 PU STEP ENA 2	2	0 PU WAIT 2 0	2	1 PU WAIT PG ENA 3	2	0
1	1 B1 SS VALUE 0	1	1 B2 SS VALUE 0	1	1 B3 SS VALUE 0	1	0 Bit 1	1	0 LDO1 PU STEP ENA 1	1	0 PU WAIT 1 1	1	1 PU WAIT PG ENA 2	1	0
0	1 B1 REFRESH FREQ	0	1 B2 REFRESH FREQ	0	1 B3 REFRESH FREQ	0	0 Bit 0	0	1 LDO1 PU STEP ENA 0	0	0 PU WAIT 1 0	0	1 PU WAIT PG ENA 1	0	0 BOOST DIS
0x0085B3 R W		0x00ADB3 R W		0x0014B3 R W		0x000000 R W		0x00130D R W		0x000000 R W		0x00030F R W		0x00C408 R W	

DCR18 Addr = 0x12		DCR19 Addr = 0x13		DCR20 Addr = 0x14		DCR21 Addr = 0x15		DCR22 Addr = 0x16		DCR23 Addr = 0x17		DCR24 Addr = 0x18		DCR25 Addr = 0x19	
15	1 PD WAIT VREG DEL OFF 1	15	0 U NVM RELEASE 15	15	0	15	0 Bit 15	15	0 Bit 15	15	0 Bit 15	15	0 Bit 15	15	0 U NVM CRC1 7
14	0 PD WAIT VREG DEL OFF 0	14	0 U NVM RELEASE 14	14	0	14	0 Bit 14	14	0 Bit 14	14	0 Bit 14	14	0 Bit 14	14	0 U NVM CRC1 6
13	0 PD WAIT DEL OFF 7 1	13	0 U NVM RELEASE 13	13	0	13	0 Bit 13	13	0 Bit 13	13	0 Bit 13	13	0 Bit 13	13	1 U NVM CRC1 5
12	1 PD WAIT DEL OFF 7 0	12	0 U NVM RELEASE 12	12	0	12	0 Bit 12	12	0 Bit 12	12	0 Bit 12	12	0 Bit 12	12	0 U NVM CRC1 4
11	0 PD WAIT DEL OFF 6 1	11	0 U NVM RELEASE 11	11	0	11	0 Bit 11	11	0 Bit 11	11	0 Bit 11	11	0 Bit 11	11	0 U NVM CRC1 3
10	1 PD WAIT DEL OFF 6 0	10	0 U NVM RELEASE 10	10	0	10	0 Bit 10	10	0 Bit 10	10	0 Bit 10	10	0 Bit 10	10	1 U NVM CRC1 2
9	0 PD WAIT DEL OFF 5 1	9	0 U NVM RELEASE 9	9	0	9	0 Bit 9	9	0 Bit 9	9	0 Bit 9	9	0 Bit 9	9	1 U NVM CRC1 1
8	1 PD WAIT DEL OFF 5 0	8	0 U NVM RELEASE 8	8	0	8	0 Bit 8	8	0 Bit 8	8	0 Bit 8	8	0 Bit 8	8	1 U NVM CRC1 0
7	0 PD WAIT DEL OFF 4 1	7	0 U NVM RELEASE 7	7	0	7	0 Bit 7	7	0 Bit 7	7	0 Bit 7	7	0 Bit 7	7	0 U PROG1 1
6	1 PD WAIT DEL OFF 4 0	6	0 U NVM RELEASE 6	6	0	6	0 Bit 6	6	0 Bit 6	6	0 Bit 6	6	0 Bit 6	6	0 U PROG1 0
5	0 PD WAIT DEL OFF 3 1	5	0 U NVM RELEASE 5	5	0	5	0 Bit 5	5	0 Bit 5	5	0 Bit 5	5	0 Bit 5	5	0
4	1 PD WAIT DEL OFF 3 0	4	0 U NVM RELEASE 4	4	0	4	0 Bit 4	4	0 Bit 4	4	0 Bit 4	4	0 Bit 4	4	0
3	0 PD WAIT DEL OFF 2 1	3	0 U NVM RELEASE 3	3	0 LOW SET 3	3	0 Bit 3	3	0 Bit 3	3	0 Bit 3	3	0 Bit 3	3	0
2	1 PD WAIT DEL OFF 2 0	2	0 U NVM RELEASE 2	2	0 LOW SET 2	2	0 Bit 2	2	0 Bit 2	2	0 Bit 2	2	0 Bit 2	2	0
1	0 PD WAIT DEL OFF 1 1	1	0 U NVM RELEASE 1	1	0 LOW SET 1	1	0 Bit 1	1	0 Bit 1	1	0 Bit 1	1	0 Bit 1	1	0
0	1 PD WAIT DEL OFF 1 0	0	1 U NVM RELEASE 0	0	0 LOW SET 0	0	0 Bit 0	0	0 Bit 0	0	0 Bit 0	0	0 Bit 0	0	0
0x009555 R W		0x000001 R W		0x000000 R W		0x000000 R W		0x000000 R W		0x000000 R W		0x000000 R W		0x002700 R W	

Figure 11. Default USER-NVM registers interpretation

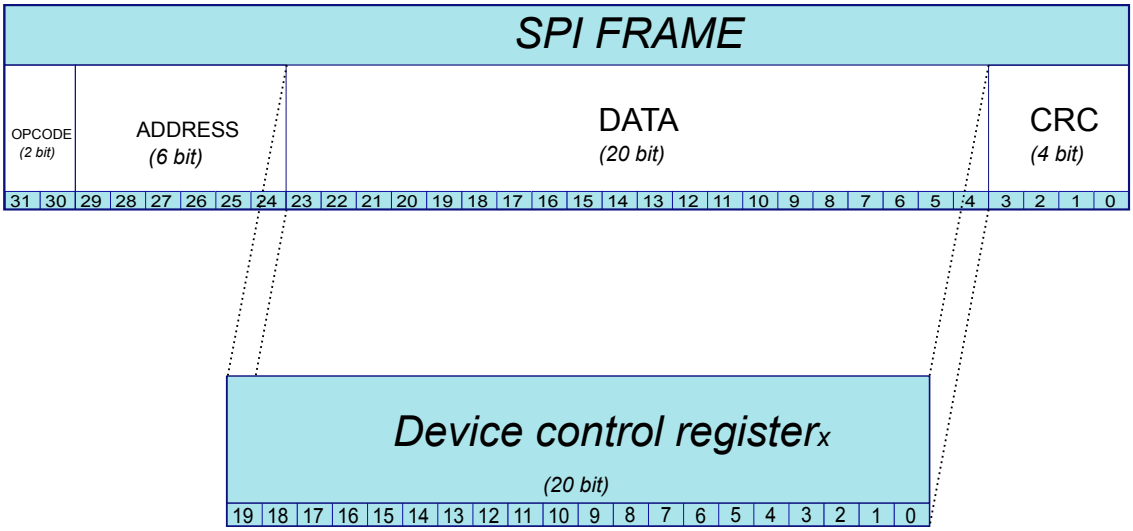


5 SPI frame and protocol details

The 32 bits SPI, defined for the device communication, has to be used to manage NVM operations. The 32 bits SPI format is reported in the Figure 12. The frame is divided into 4 different fields:

- OPCODE 2 bits [31:30]
- ADDRESS 6 bits [24:29]
- DATA field 20 bits [23:5], further aim of this field only with re-numbered bits [19:0]
- CRC 4 bits [3:0] This means the bit 4 will be identified as bit 0

Figure 12. SPI frame structure



6 NVM registers DATA field details

Write 0 to the "(RESERVED)" and "(NOT USED)" fields.

6.1 DCR10 (0X0A)

Table 1. DCR10 DATA field structure

DCR10																		
(RESERVED)				BUCK1_FREQ	BUCK1_PU_VALUE		BUCK1_PU_STEP_ENA			BUCK1_PD_STEP_OFF			BUCK1_IPEAK			BUCK1_REGFAIL_GO_REC	BUCK1_SS_VALUE		BUCK1_REFRESH_FREQ
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BUCK1_FREQ
Buck1 switching frequency:
0: 2.4 MHz
1: 0.4 MHz
BUCK1_PU_VALUE [1:0]
Buck1 voltage setting:
00: 3.3 V
01: 5.0 V
1x: 6.5 V
BUCK1_PU_STEP_ENA [2:0]
Buck1 turn on step:
000: buck is not turned ON
001: step 1
010: step 2
011: step 3
100: step 4
101: step 5
110: step 6
111: step 7
BUCK1_PD_STEP_OFF [2:0]
Buck1 turn off step:
000: step 1
001: step 2
010: step 3
011: step 4
100: step 5
101: step 6
11x: step 7
BUCK1_IPEAK [2:0]
Buck1 peak limitation current:
000: 2.0 Amp
001: 2.5 Amp
010: 3.0 Amp

011: 3.5 Amp
100: 4.0 Amp
101: 4.5 Amp
11x: 5.0 Amp

BUCK1_REGFAIL_GO_FAIL

Device behavior in case of Buck1 fault:

0: BUCK1 regulator fail generates IRQ (if unmasked) and deactivates regulator (if unmasked)
1: BUCK1 regulator fail initiates a power-down sequence and moves the state machine to REC-2 state

BUCK1_SS_VALUE [1:0]

Buck1 soft start slope setting:

00: 16.5 V/ms
01: 8.25 V/ms
10: 3.3 V/ms
11: 1.65 V/ms

BUCK1_REFRESH_FREQ

Buck1 bootstrap capacitor refresh choice:

1: refresh frequency 1 kHz
0: refresh frequency 25 kHz

6.2 DCR11 (0X0B)

Table 2. DCR11 DATA field structure

DCR10																			
(RESERVED)				BUCK2_FREQ	BUCK2_PU_VALUE		BUCK2_PU_STEP_ENA			BUCK2_PD_STEP_OFF			BUCK2_IPEAK			BUCK2_REGFAIL_GO_REC	BUCK2_SS_VALUE		BUCK2_REFRESH_FREQ
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BUCK2_FREQ

Buck2 switching frequency:

0: 2.4 MHz
1: 0.4 MHz

BUCK2_PU_VALUE [1:0]

Buck2 voltage setting:

00: 3.3 V
01: 5.0 V
1x: 6.5 V

BUCK2_PU_STEP_ENA [2:0]

Buck2 turn on step:

000: buck is not turned ON
001: step 1
010: step 2
011: step 3
100: step 4
101: step 5
110: step 6
111: step 7

BUCK2_PD_STEP_OFF [2:0]

Buck2 turn off step:

000: step 1

001: step 2

010: step 3

011: step 4

100: step 5

101: step 6

11x: step 7

BUCK2_IPEAK [2:0]

Buck2 peak limitation current:

000: 2.0 Amp

001: 2.5 Amp

010: 3.0 Amp

011: 3.5 Amp

100: 4.0 Amp

101: 4.5 Amp

11x: 5.0 Amp

BUCK2_REGFAIL_GO_FAIL

Device behavior in case of Buck2 fault:

0: BUCK2 regulator fail generates IRQ (if unmasked) and deactivates regulator (if unmasked)

1: BUCK2 regulator fail initiates a power-down sequence and moves the state machine to REC-2 state

BUCK2_SS_VALUE [1:0]

Buck2 soft start slope setting:

00: 16.5 V/ms

01: 8.25 V/ms

10: 3.3 V/ms

11: 1.65 V/ms

BUCK2_REFRESH_FREQ

Buck2 bootstrap capacitor refresh choice:

1: refresh frequency 1 kHz

0: refresh frequency 25 kHz

6.3 DCR12 (0X0C)

Table 3. DCR12 DATA field structure

DCR12																			
(RESERVED)				BYPASS_DIS	BUCK3_PU_VALUE		BUCK3_PU_STEP_ENA			BUCK3_PD_STEP_OFF			BUCK3_IPEAK			BUCK3_REGFAIL_GO_REC	BUCK3_SS_VALUE		BUCK3_REFRESH_FREQ
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BYPASS_DIS

Boost bypass option:

1: Disabled–Bypass disable

0: Enabled–Bypass enable (will be ON or OFF based on the sensed value on the VS pin)

BUCK3_PU_VALUE [2:0]

Buck3 voltage setting:

001: 0.98 V

010: 1.1 V

011: 1.2 V

100: 1.25 V

101: 3.3 V

11x: 6.5 V

BUCK3_PU_STEP_ENA [2:0]

Buck3 turn on step:

000: buck is not turned ON

001: step 1

010: step 2

011: step 3

100: step 4

101: step 5

110: step 6

111: step 7

BUCK3_PD_STEP_OFF [2:0]

Buck3 turn off step:

000: step 1

001: step 2

010: step 3

011: step 4

100: step 5

101: step 6

11x: step 7

BUCK3_IPEAK [2:0]

Buck3 peak limitation current:

00: 4.0 Amp

01: 5.0 Amp

10: 6.0 Amp

11: 7.0 Amp

BUCK3_REGFAIL_GO_FAIL

Device behavior in case of Buck3 fault:

0: BUCK3 regulator fail generates IRQ (if unmasked) and deactivates regulator (if unmasked)

1: BUCK3 regulator fail initiates a power-down sequence and moves the state machine to REC-2 state

BUCK3_SS_VALUE [1:0]

Buck3 soft start slope setting:

00: 8.7 V/ms

01: 4.35 V/ms

10: 1.75 V/ms

11: 0.87 V/ms

BUCK3_REFRESH_FREQ

Buck3 bootstrap capacitor refresh choice:

1: refresh frequency 1 kHz

0: refresh frequency 25 kHz

6.4 DCR13 (0X0D), DCR21-24 (0x15-0x18)

Table 4. DCR13, DCR21-24 DATA field structure

DCR13, DCR21-24																			
NOT USED – write 0x00000																			
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

6.5 DCR14 (0X0E)

Table 5. DCR14 DATA field structure

DCR14																			
(RESERVED)				PU_LOOP_EVE	LDO2_REGFAIL_GO_REC	LDO2_TRK		LDO2_PD_STEP_OFF			LDO2_PU_STEP_ENA			LDO1_PD_STEP_OFF			LDO1_PU_STEP_ENA		
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

LDO2_REGFAIL_GO_FAIL

Device behavior in case of LDO2 fault:

0: LDO2 regulator fail generates IRQ (if unmasked) and deactivates regulator (if unmasked)

1: LDO2 regulator fail initiates a power-down sequence and moves the state machine to REC-2 state

LDO2_TRK [1:0]

LDO2 tracked voltage setting:

00: BUCK1 (only if BUCK1 output voltage is 3.3 V or 5 V otherwise LDO2 OFF)

01: BUCK2 (only if BUCK2 output voltage is 3.3 V or 5 V otherwise LDO2 OFF)

1x: BUCK3 (only if BUCK3 output voltage is 3.3 V otherwise LDO2 OFF)

LDO2_PD_STEP_OFF [2:0]

LDO2 turn off step:

000: step 1

001: step 2

010: step 3

011: step 4

100: step 5
101: step 6
11X: step 7

LDO2_PU_STEP_ENA [2:0]

LDO2 turn on step:

000: LDO is not turned ON
001: step 1
010: step 2
011: step 3
100: step 4
101: step 5
110: step 6
111: step 7

LDO1_PD_STEP_OFF [2:0]

LDO1 turn off step:

000: step 1
001: step 2
010: step 3
011: step 4
100: step 5
101: step 6
11x: step 7

LDO1_PU_STEP_ENA [2:0]

LDO1 turn on step:

000: LDO is not turned ON
001: step 1
010: step 2
011: step 3
100: step 4
101: step 5
110: step 6
111: step 7

6.6 DCR15 (0X0F)

Table 6. DCR15 DATA field structure

DCR15																			
(RESERVED)				(NOT USED)		PU_WAIT_DEL_ENA_7		PU_WAIT_DEL_ENA_6		PU_WAIT_DEL_ENA_5		PU_WAIT_DEL_ENA_4		PU_WAIT_DEL_ENA_3		PU_WAIT_DEL_ENA_2		PU_WAIT_DEL_ENA_1	
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PU_WAIT_DEL_ENA_7 [1:0]
Power up delay at step 7:
00: 0 ms
01: 2 ms
10: 5 ms
11: 10 ms
PU_WAIT_DEL_ENA_6 [1:0]
Power up delay at step 6:
00: 0 ms
01: 2 ms
10: 5 ms
11: 10 ms
PU_WAIT_DEL_ENA_5 [1:0]
Power up delay at step 5:
00: 0 ms
01: 2 ms
10: 5 ms
11: 10 ms
PU_WAIT_DEL_ENA_4 [1:0]
Power up delay at step 4:
00: 0 ms
01: 2 ms
10: 5 ms
11: 10 ms
PU_WAIT_DEL_ENA_3 [1:0]
Power up delay at step 3:
00: 0 ms
01: 2 ms
10: 5 ms
11: 10 ms
PU_WAIT_DEL_ENA_2 [1:0]
Power up delay at step 2:
00: 0 ms
01: 2 ms
10: 5 ms
11: 10 ms

PU_WAIT_DEL_ENA_1 [1:0]
Power up delay at step 1:
00: 0 ms
01: 2 ms
10: 5 ms
11: 10 ms

6.7 DCR16 (0X10)

Table 7. DCR16 DATA field structure

DCR16																			
(RESERVED)				(NOT USED)			NRESET_PD_STEP_ASSERT			NRESET_PU_STEP_DEASSERT			PU_WAIT_PG_ENA_7	PU_WAIT_PG_ENA_6	PU_WAIT_PG_ENA_5	PU_WAIT_PG_ENA_4	PU_WAIT_PG_ENA_3	PU_WAIT_PG_ENA_2	PU_WAIT_PG_ENA_1
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

NRESET_PD_STEP_ASSERT [2:0]
To define NRESET assertion steps:
000: step 1
001: step 2
010: step 3
011: step 4
100: step 5
101: step 6
11x: step 7
NRESET_PU_STEP_DEASSERT [2:0]
To define NRESET de-assertion and WD start step:
000: end of step 7
001: step 1
010: step 2
011: step 3
100: step 4
101: step 5
110: step 6
111: step 7
PU_WAIT_PG_ENA_7
Power-good signal choice:
1: at power-up step 7 wait for power-good signal
0: at power-up step 7 does not wait for power-good signal
PU_WAIT_PG_ENA_6
Power-good signal choice:
1: at power-up step 6 wait for power-good signal
0: at power-up step 6 does not wait for power-good signal
PU_WAIT_PG_ENA_5
Power-good signal choice:
1: at power-up step 5 wait for power-good signal
0: at power-up step 5 does not wait for power-good signal

PU_WAIT_PG_ENA_4

Power-good signal choice:

1: at power-up step 4 wait for power-good signal

0: at power-up step 4 does not wait for power-good signal

PU_WAIT_PG_ENA_3

Power-good signal choice:

1: at power-up step 3 wait for power-good signal

0: at power-up step 3 does not wait for power-good signal

PU_WAIT_PG_ENA_2

Power-good signal choice:

1: at power-up step 2 wait for power-good signal

0: at power-up step 2 does not wait for power-good signal

PU_WAIT_PG_ENA_1

Power-good signal choice:

1: at power-up step 1 wait for power-good signal

0: at power-up step 1 does not wait for power-good signal

6.8 DCR17 (0X11)

Table 8. DCR17 DATA field structure

DCR17																			
(RESERVED)				U_NVM_CRC0								U_PROG0		(NOT USED)		NFSO_STATE_IN_DEEP_SLEEP	LOW_SET		BOOST_DIS
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

U_NVM_CRC0 [7:0]

Control CRC

Note: Written by SPSB100. Content is ignored during writing done by the user.

U_PROG0 [1:0]

Confirmation user NVM has been programmed:

00: not programmed

11: already programmed

NFSO_STATE_IN_DEEP_SLEEP

NFSO option:

1: NFSO pin is high when FSM in DEEP_SLEEP state

0: NFSO pin is low when FSM in DEEP_SLEEP state

BOOST_DIS

Boost option:

1: Disabled–boost function is not used

0: Enabled–boost function is used and depends on BOOST_OFF control bit

6.9 DCR18 (0X12)

Table 9. DCR18 DATA field structure

DCR18																			
(RESERVED)				PD_WAIT_VREG_DEL_OFF		PD_WAIT_DEL_OFF_7		PD_WAIT_DEL_OFF_6		PD_WAIT_DEL_OFF_5		PD_WAIT_DEL_OFF_4		PD_WAIT_DEL_OFF_3		PD_WAIT_DEL_OFF_2		PD_WAIT_DEL_OFF_1	
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PD_WAIT_VREG_DEL_OFF [1:0]
Power down delay for Vreg:
00: 2 ms
01: 5 ms
10: 10 ms
11: 20 ms
PU_WAIT_DEL_OFF_7 [1:0]
Power down delay at step 7:
00: 0 ms
01: 2 ms
10: 5 ms
11: 10 ms
PU_WAIT_DEL_OFF_6 [1:0]
Power down delay at step 6:
00: 0 ms
01: 2 ms
10: 5 ms
11: 10 ms
PU_WAIT_DEL_OFF_5 [1:0]
Power down delay at step 5:
00: 0 ms
01: 2 ms
10: 5 ms
11: 10 ms
PU_WAIT_DEL_OFF_4 [1:0]
Power down delay at step 4:
00: 0 ms
01: 2 ms
10: 5 ms
11: 10 ms
PU_WAIT_DEL_OFF_3 [1:0]
Power down delay at step 3:
00: 0 ms
01: 2 ms
10: 5 ms
11: 10 ms

PU_WAIT_DEL_OFF_2 [1:0]
Power down delay at step 2:
00: 0 ms
01: 2 ms
10: 5 ms
11: 10 ms
PU_WAIT_DEL_OFF_1 [1:0]
Power down delay at step 1:
00: 0 ms
01: 2 ms
10: 5 ms
11: 10 ms

6.10 DCR19 (0X13)

Table 10. DCR19 DATA field structure

DCR19																			
(RESERVED)				U_NVM_RELEASE (16 bit identification of USER_NVM release)															
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

6.11 DCR20 (0X14)

Table 11. DCR20 DATA field structure

DCR20																			
(RESERVED)				(NOT USED)												LOW_SET			
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

LOW_SET [3:0]
Long open window setting
0000: 319 ms max.
0001: 479 ms max.
0010: 638 ms max.
0011: 1025 ms max.
0100: 115 ms max.
0101: 159 ms max.
0110: 230 ms max.
0111: 460 ms max.
1000-1011: 2300 ms max.
1100: 4600 ms max.
1101: 6900 ms max.
1110: 9200 ms max.
1111: infinite

6.12 DCR25 (0X19)

Table 12. DCR25 DATA field structure

DCR25																			
(RESERVED)				U_NVM_CRC1								U_PROG1		(NOT USED)					
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

U_NVM_CRC1 [7:0]

Control CRC

Note: Written by SPSB100. Content is ignored during writing done by the user.

U_PROG1 [1:0]

Confirmation user NVM has been programmed:

00: not programmed

11: already programmed

Revision history

Table 13. **Document revision history**

Date	Version	Changes
05-Feb-2024	1	Initial release.
12-Jul-2024	2	Updated Figure 1. Functional block diagram, Figure 10. Default USER-NVM registers content, Figure 11. Default USER-NVM registers interpretation and Section 6.8: DCR17 (0X11). Removed section "Recommended configuration (BA silicon)" and "Write example of recommended configuration (BA silicon)".

Contents

List of tables

List of figures