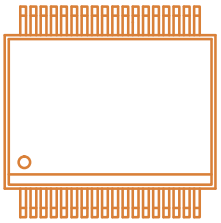
i i Pi
C f A m i
(F) Li i
NC 78343

MARKING DIAGRAM



SAFETY DESIGN ASIL B

ASIL B Product developed in compliance with ISO 26262 for which a complete safety package is available.

ORDERING INFORMATION

Device	Package	Shipping†
NCV78343DQ0R2G	SSOP36 EP (P Free)	1500 / Tape & Reel

†For information on tape and reel specifications, including part orientation and tape sizes, please refer to our Tape and Reel Packaging Specification Brochure, BRD8011/D.

Introduction

The NCV78343 is a single−chip pixel controller with embedded switches to control individual LEDs in a series LED string, designed for automotive dynamic lighting applications and in particular for high current LEDs. In order to create a pixel lighting solution, the LEDs need to be powered by current sources such as NCV78763 or NCV78723. The NCV78343 pixel controller devices receive the pixel control parameters from the pixel light ECU which translates the required light pattern or light image into individual pixel dimming data.

One pixel controller device can control up−to 12 pixels of 1× or 2× 1.4 A LEDs per pixel. The maximum LED string voltage has to be limited to 60 V.

When more than 12 pixels are to be controlled, multiple pixel controllers can be combined in a single system.

The NCV78343 uses two communication interfaces for connection with a microcontroller. A universal asynchronous receiver transmitter (UARTv bp

NV78343 = Specific Device Code
F = Fab Indicator
A = Assembly Location
WL = Wafer Lot
YYWW = Year / Work Week
G = Pb Free Designator

Typical Applications

- Dynamic Adaptive Driving Beam Functions
  - Glare−free High Beam
  - Static Swiveling
  - Beam Shaping
  - Light Power Adjustment
- Animated Welcome Functions on Signal Lights
- Wiping Blinker

PACKAGE AND PIN DESCRIPTION

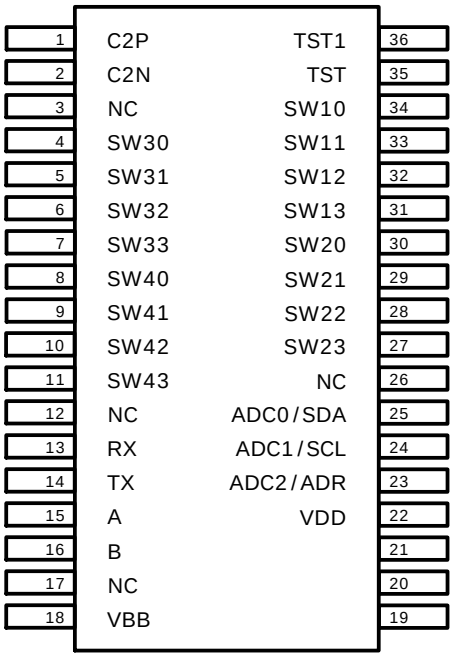


Figure 1. Pin Connections – SSOP36 EP (Top View)

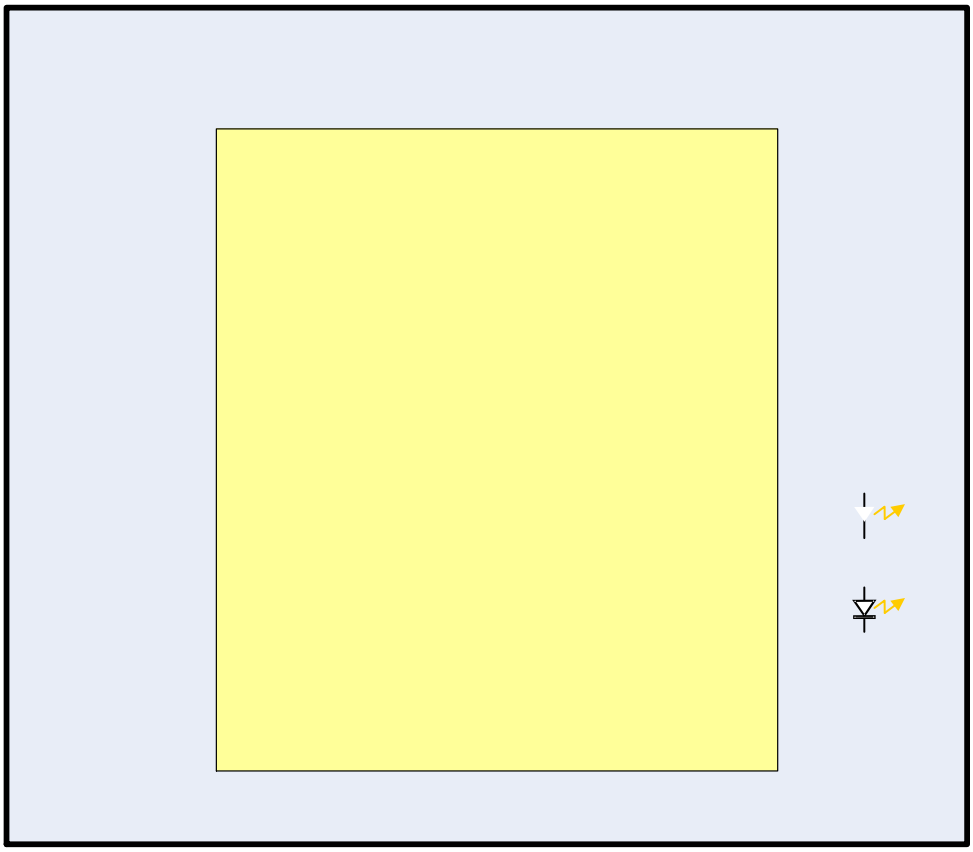


Figure 2. Application Diagram

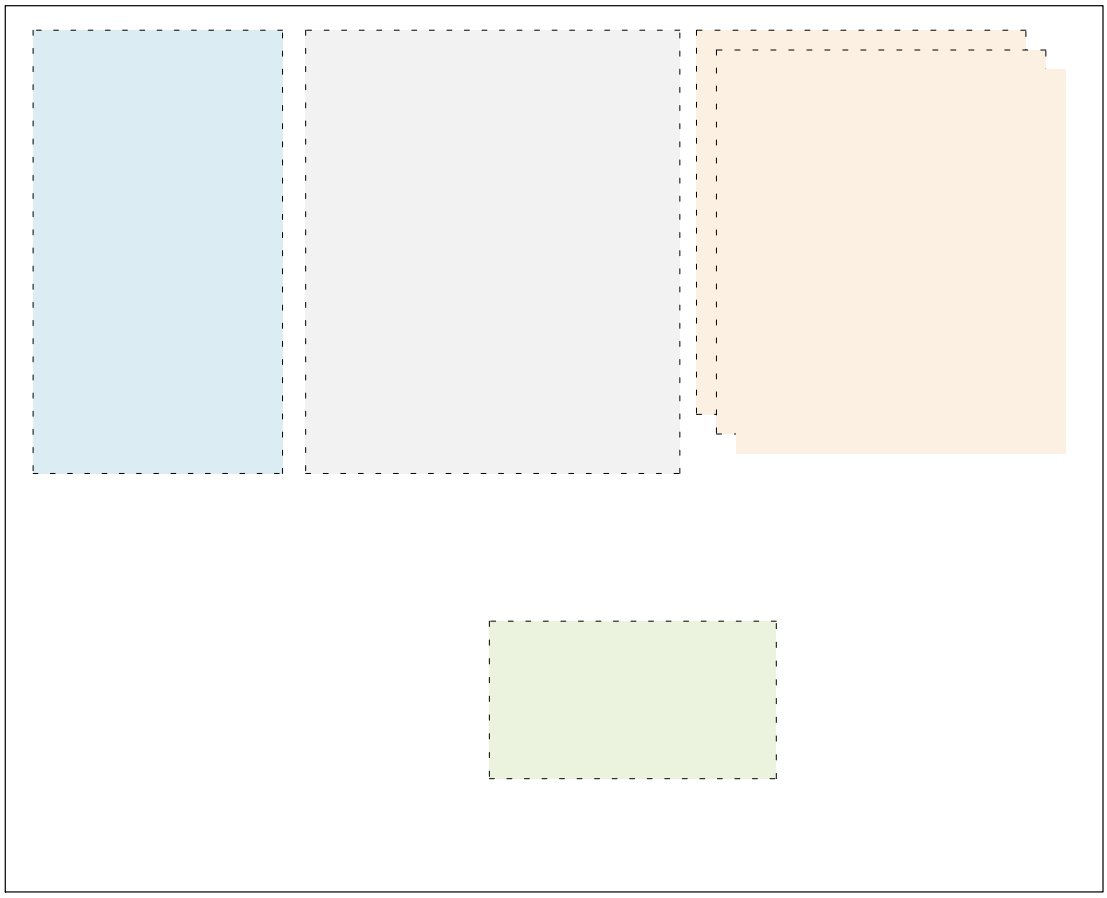


Figure 3. Block Diagram

The advantage of sharing common heatsink for higher currents can be reached by placement of the NCV78343 together with the LEDs on same PCB (IMS type of board supported). This is not necessary for lower currents or application where the LED string is connected over two NCV78343 devices.

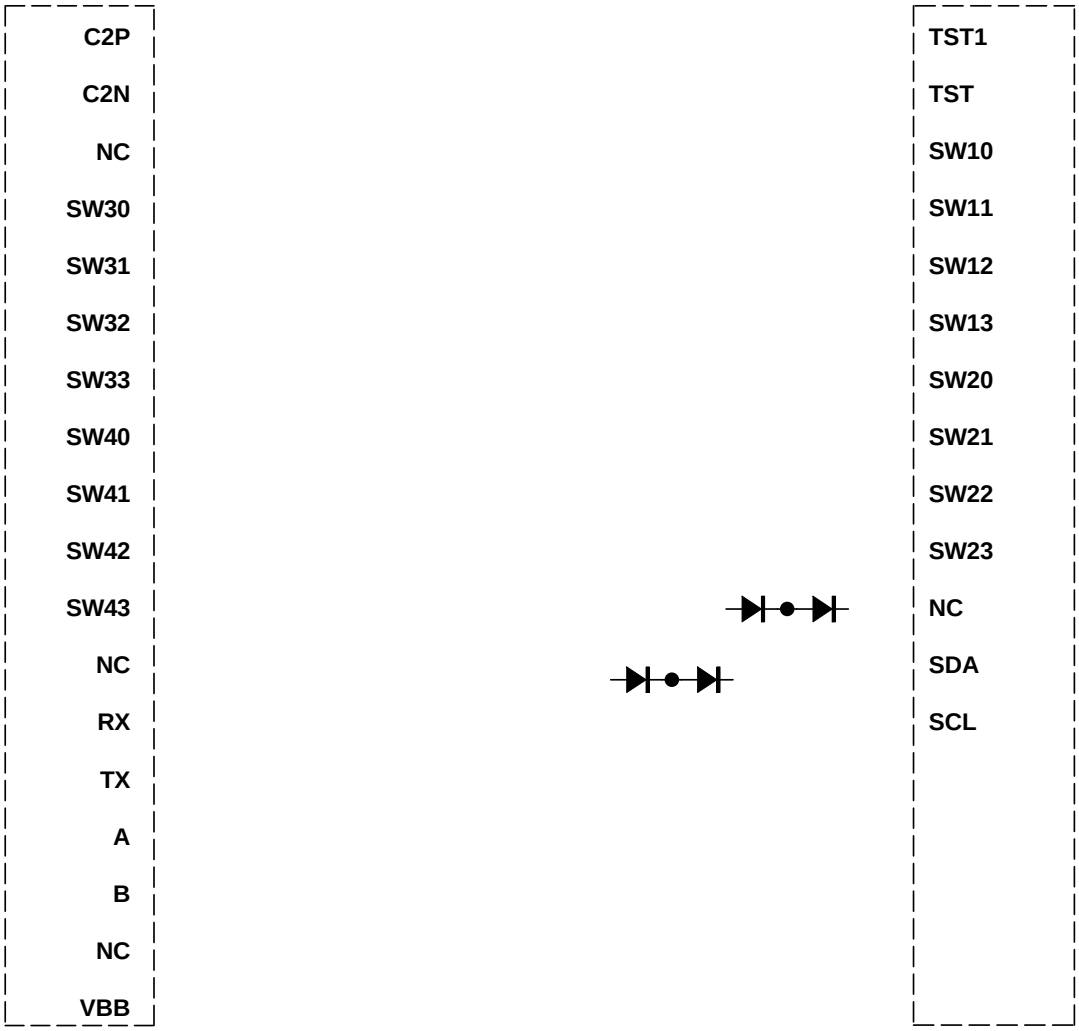


Figure 6. ESD Protection Schematic

Typical Switch Resistance

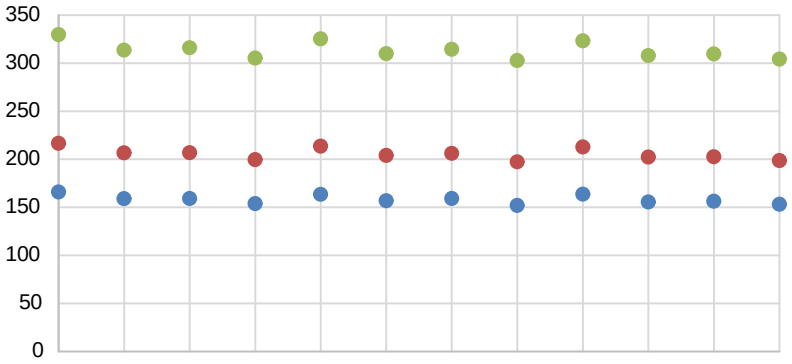


Figure 7. Typical Switch Resistance

Table 3. ABSOLUTE MAXIMUM RATINGS

Characteristic	Symbol	Min	Max	Unit
Battery Supply voltage (Note 1)	V_{BB}	0.3	60	V
Low voltage supply (Note 2)	V_{DD}	0.3	3.6	V
High voltage control IO pins (Note 3)	I_{OHV60}	0.3	60	V
High voltage IO pins (Note 4)	I_{OHV}	0.3	68	V
Medium voltage IO pins (Note 5)	I_{OMV}	0.3	6.5	

ELECTRICAL CHARACTERISTICS

NOTE: All Min and Max parameters are guaranteed over full junction temperature (T_{JP}) range (−40 °C; 150 °C), unless otherwise specified.

Table 6. CURRENT CONSUMPTION

Characteristic	Symbol	Conditions	Min	Typ	Max	Unit
The VBB current consumption	I_VBB			19	25	mA
The VBB current consumption UART only device	I_VBB_M LVDS_ OFF	M LVDS off; OTP bi82 0 8 .236 76.762 S(19)TjET453.146 643.181 .90707 1 Tm(.899[(L)76.3(VD				

Table 10. PIXEL SWITCHES

Characteristic	Symbol	Conditions	Min	Typ	Max	Unit
RON from SWx3 to SWx0 pin (3 switches)						

Table 14. M LVDS INTERFACE: A, B

Characteristic	Symbol	Conditions	Min	Typ	Max	Unit
Differential output voltage magnitude	M LVDS_TX_VAB	Rload_A B = 49.9 Ω ±1% Vtest = from 1 V to 3.4 V	480			

DETAILED OPERATING AND PIN DESCRIPTION

SUPPLY CONCEPT IN GENERAL

Low operating voltages become more and more required due to the growing use of start stop systems. In order to respond to this necessity, the NCV78343 is designed to support power−up starting from VBB = 4.5 V.

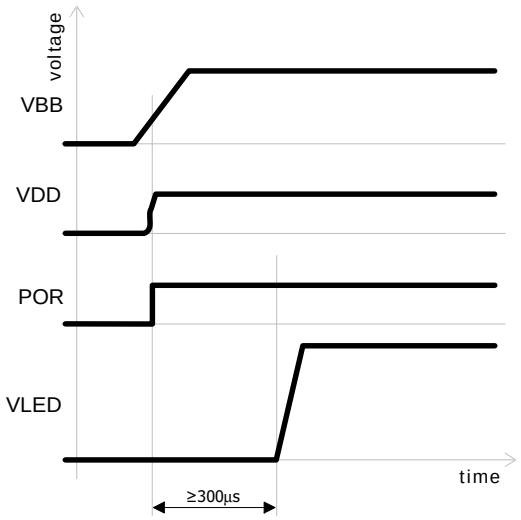


Figure 10. Power up Sequence

A specific power−up and power−down sequences are shown in the Figure 20 and Figure 21.

There is no special circuit to disable switches in case of VBB power supply disconnection. The gate of the switch is discharged by SW−OFF circuit in case the VBB−LOW threshold is crossed. The gate of the switch is discharged by leakage currents when the supply is suddenly lost. Because of low leakage currents, the switch may stay enabled for a few seconds after power lost. Possible temperature rise speeds up opening the switch by higher leakage current.

VDD Supply

The VDD supply is the low voltage digital and analog supply for the chip, which is powered from VBB. VDD is supplying the internal analog and digital circuits as well as external components like I2C EEPROM and resistor divider on ADC inputs. The POR−circuit is monitoring both the VBB and VDD voltages.

VLED Supply

If the device is running but the LED current source is disconnected, the LEDs can light up because of the bias currents flowing through pins of the switches. Up to 180 µA (typical) from switch current source may cause the bottom−most LED to shine. If needed, resistors can be connected in parallel to the switches to avoid undesired LED lighting (typically 10 kΩ).

INTERNAL CLOCK GENERATION

The clocks are fully internally generated without the need for any trimming by the user. The accuracy is guaranteed under all operating conditions and independent of external component selection.

OSC20M Clock

The OSC20M clock is the system clock. All the internal timings as well as the internal PWM unit depend on OSC20M accuracy.

Communication Clock

The internal clock is also used for oversampling of UART incoming frame and I2C EEPROM, so there is no need for any external clock.

DIMMING CONTROLLER

Internal (built−in) dimming controller allows change of light intensity of individual LEDs in LED string by means of digital (PWM) dimming.

Dimming Control Parameters

The dimming for all switches is controlled from 1 common 10−bit counter. The ON and OFF events are programmable per channel, each with a 10 bit counter value.

100% duty cycle is generated when ON time is set to min. value (0) and OFF time is set to max value (1023).

0% duty cycle is generated when ON time is equal to OFF time. When more than one 0% or 100% duty cycle is required, the TR (transition) slots must be used.

The dimming frequency is the DIMCLK frequency divided by 1024. The T_{DIMCLK} is the duration of one PWM tick. The duration of one PWM period is T_{PWM}. The required time for one switch ON sequence is T_{SW_SEQ}. The ratio of T_{SW_SEQ} and T_{DIMCLK} results in number of PWM ticks required for one switch ON sequence. The number of slots available for each DIMCLK is 1024 divided by the ratio. The recommended time for TR slots and recommended step between each switch ON request is shown in Table 43. When the TR slot technique is used, the ON values should not be set within this period.

Dimming Mode

The NCV78343 incorporates two modes of operation – ON/OFF dimming mode and direct mode.

- – the NCV78343 controls the dimming duty cycle and phase shift for each switch individually. The time of ON event is set by means of <ONx[9:0]> register and the time of OFF event is set by means of <OFFx[9:0]> register.
- – in addition to ON/OFF dimming mode, the state of the switches can be controlled directly by means of <SWx> register.

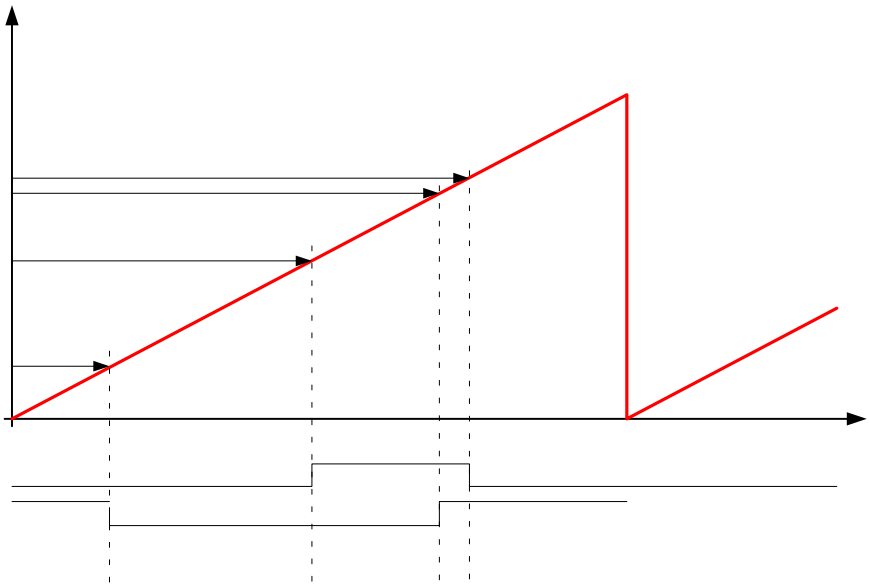


Figure 11. Dimming Operation (dimming ON/OFF event)

OPEN, SHORT and FAIL Status Detection

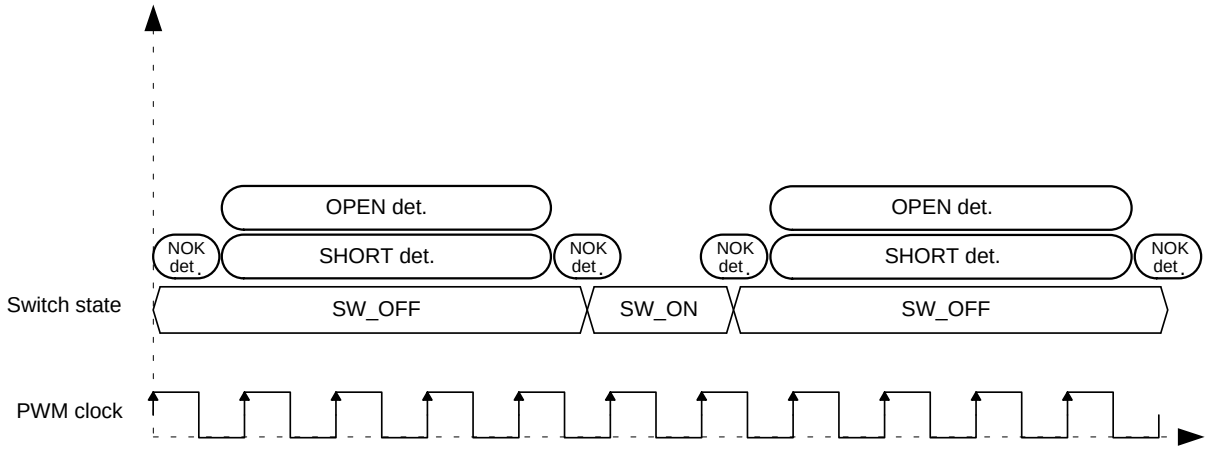


Figure 13. OPEN, SHORT and FAIL Status Detection Timing

Following the figure above, the OPEN and SHORT flags are detected only during the switch OFF state. The On/Off Failed flag detection is triggered by the transition between the switch ON and the switch OFF event. The SHORT and On/Off Failed status flags are cleared upon a successful read out of register 0x0F. Due to this behavior and the diagram

communication speed 250 kbps). The BREAK field stop bit (BREAK field delimiter) is minimum 1 Tbit and maximum according to the selected watchdog time. If the device is not responding through the repeater−slave, the extended break (26 Tbits) can be required to recover communication to slave devices. Such case can occur when Read frame is addressed non assigned address. In case of only M−LVDS slave cluster, the DE pin on the M−LVDS transceiver (e.g. NBA3N206S) must be set LOW within 1 Tbit after the Header part to allow device response.

The PXN protocol supports two frame types:

− configuration frame
− register bank frame

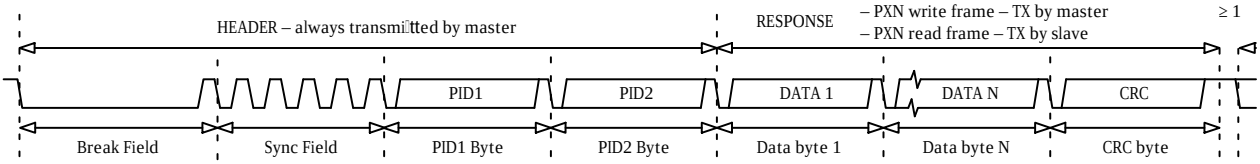


Figure 16. PXN Frame

Multi−level addressing procedure with long time delay recommendation in Application notes. If the Multi−level addressing is successful, a device stays in the OTP config mode (see Figure 23).

Auto addressing Procedure

Regardless the node address assigned after the measurement of ADC2/ADR multi−level input, the auto−addressing procedure can still be invoked using the auto−address enable PXN frame. The auto−addressing is enabled/disabled on the PXN node upon receiving CF4 PXN configuration frame. The CF4 frame is accepted in the OTP_CONFIG mode only (see Table 53).

The PXN node address, when auto−addressing is enabled is assigned upon receiving CF5 PXN configuration frame. The CF5 frame is accepted in AUTO_ADDR mode only (see Table 54).

The auto−addressing procedure is described in the APPLICATION RELATED INFORMATION section.

OTP Node Address

The OTP node address can be zapped by customer at EoL (End of Line) after the PXN node address was determined either by means of multi−level address pin measurement or by means of auto−addressing procedure. The value of OTP node address and OTP bank lock bit is obtained each time the PXN node is powered up and the custom OTP bank is read out. Loading of other device settings from OTP memory speeds up device setup after power−on. OTP memory zapping is necessary to fulfil ASIL B safety requirements.

PXN Communication Speed

The PXN node can communicate at following speed:

− 125 kb/s
− 250 kb/s (default)
− 500 kb/s
− 1 Mb/s

Communication speed is changed upon receiving CF12 PXN communication frame in OTP_CONFIG, AUTO_ADDR and NORMAL modes only (see Table 61).

OTP Bank – Custom Data

The custom OTP bank is typically zapped by customer at EOL and stored values are used for system operation customization.

Table 24. OTP BANK

OTP#	OTP Name
0	**OTP lock bit**
1	OTP node address lock bit
2	OTP node address bit 0
3	OTP node address bit 1
4	OTP node address bit 2
5	OTP node address bit 3
6	OTP node address bit 4
7	**Fail safe state lock bit**
8	Fail safe state of LEDs in string 1
9	Fail safe state of LEDs in string 2
10	Fail safe state of LEDs in string 3
11	Fail safe state of LEDs in string 4
12	**PXN lock bit**
13	Mode (slave/repeater slave)
14	Communication speed bit 0
15	Communication speed bit 1
16	Global bit error detection disable
17	M LVDS OFF
18	UART OFF
19	**EEPROM lock bit (write protect)**
20	CRC bit0
21	CRC bit1
22	CRC bit2
23	CRC bit3
24	CRC bit4
25	CRC bit5
26	CRC bit6

<OTP_LOCK_BIT> − custom OTP bank general lock bit. When zapped, any further zapping attempt of custom OTP bank is declined.

<OTP_NODE_ADDR_LOCK_BIT> − PXN node address lock bit. When zapped, any further zapping attempt of <OTP_NODE_ADDR> bits of custom OTP bank is declined.

<OTP_NODE_ADDR [4:0]> − 5−bit PXN node address. This address is taken into account only when the <OTP_NODE_ADDR_LOCK_BIT> is zapped.

<FAIL_SAFE_STATE_LOCK_BIT> − fail safe state of LED string lock bit. When zapped, any further zapping attempt of <FAIL_SAFE_STATE_LED_STRINGx> bits of custom OTP bank is declined.

<FAIL_SAFE_STATE_LED_STRINGx> − state of the LED string x, x={1,2,3,4}, in cas.026 13.663 ref359.94 192.926 .6803 fail safail safail s ÜÎ fa@V −68ÈÀ 1 wk0ts zapped.

<PXN_COMMUNICATION_SPEED [1:0]> − 2−bit PXN communication speed selection bit. The communication speed selection is taken into account only when the <PXN_LOCK_BIT> is set and the CRC is correct.

Table 25. PXN COMMUNICATION SPEED

COMMUNICATION_SPEED[1:0] OTP Setting

and PWM modes is same as Normal Direct and Normal PWM modes. NO_CRC prefix means that the device detected invalid CRC in OTP memory bank 2 during power on, most likely because OTP memory is not written.
Please note that only device with written OTP memory achieves ASIL B safety rating.

Fail−safe OTP mode – when this fail−safe state is entered after watchdog timeout, OTP fail−safe data are loaded and

REGISTER DESCRIPTION

Table 27. REGISTER 0x00

Table 30. REGISTER 0x03

Register 0x03

Table 39. REGISTER 0x0C

Register 0x0C													
Address		Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0C	Name	OFF12[11:4]								TR12[3:0]			
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Address		Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12
0x0C	Name	ON12[23:14]										OFF12[13:12]	
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ON12[9:0] – 10−bit switch ON threshold.

OFF12[9:0] – 10−bit switch OFF threshold.

TR12[3:0] – Transition vector duration, it is prolonging the duration of ON resp. OFF value at the end of PWM period by <TRx[3:0]> × Time slot between switch activations.

Table 40. REGISTER 0x0D

Register 0x0D													
Address		Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0D	Name	TW_CODE[11:8]								ADC_SEL[3:2]		CRC_CLR	MAPENA
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Address		Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12
0x0D	Name									TW_CODE[15:12]			
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

MAPENA – Register bank map enable request. When MAPENA request is written ‘1’ the internal mapena register bit is set. It remains set until PWM counter overflows. The internal mapena is cleared upon the PWM counter overflows. The <MAPENA> is always read as ‘0’.

CRC_CLR – <PXN_CRC_ERR_CNT[3:0]> clear request. When <CRC_CLR> bit is set to ‘1’ the <PXN_CRC_ERR_CNT[3:0]> bits are cleared immediately. The <CRC_CLR> bit is always read as ‘0’.

ADC_SEL[1:0] – 2−bit ADC measurement channel selection for ADCx A/D conversion (see Table 11. ADC for measuring VBB, VDD, VLED, TEMP, ADCx). The measurement channel is selected according to the following table:

Table 41. ADC MEASUREMENT CHANNEL SELECTION

ADC_SEL[1:0]	Measurement Channel
0x0	ADC0
0x1	ADC1
0x2	ADC2
0x3	Reserved

NOTE: The ADCx measurement result can be obtained by reading <ADCx_RES[7:0]> status bits.
In case the <ADC_SEL[1:0]> bits are set to “11”, the returned measured value is always “00000000”.

TW_CODE – 8−bit thermal warning threshold. The default value is calculated as follows:
TW_CODE[7:0] = TSD_CODE[7:0] − 9

Table 42. REGISTER 0x0E

Register 0x0E													
Address		Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0E	Name					DIMFREQ[7:3]					CONF_SEL[2:0]		
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Address		Bit 23	Bit 22	Bit 21	Bit 20	Bit 19							

Bit 1

Table 44. REGISTER 0x0F

Register 0x0F													
Address		Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0F	Name	SW6.STATUS [11:10]		SW5.STATUS [9:8]		SW4.STATUS [7:6]		SW3.STATUS [5:4]		SW2.STATUS [3:2]		SW1.STATUS [1:0]	
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	Access	R	R	R	R	R	R	R	R	R	R	R	R
Address		Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12
0x0F	Name	SW12.STATUS [23:22]		SW11.STATUS [21:20]		SW10.STATUS [19:18]		SW9.STATUS [17:16]		SW1.STATUS [1:0]		SW7.STATUS [13:12]	
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	Access	R	R	R	R	R	R	R	R	R	R	R	R

SWx.STATUS[2] − Reflects status of internal SWx flags:

“00” – On/Off OK

“01” – On/Off Failed

“10” – Open

“11” – Short

The bit is cleared upon a successful readout over PXN (clear by read bit).

Table 45. REGISTER 0x10

Register 0x10													
Address		Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
										DIMWARN	GSWERR	TSD	TW
										0	0	0	0
										R	R	R	R
										Bit 15	Bit 14	Bit 13	Bit 12

PXN_F

RAME825 44.957 .90707 ref.85 7 0 0 7 527.470820.605 .9070 0 0 7 52439

VBB_LOW − The bit is set if the battery voltage is lower than 4.5 V. The bit is cleared upon a successful readout over PXN (clear by read bit).

GND_LOSS − The GND loss comparator detects Ground connection loss. The TST1 pin is used as reference ground. The TST1 pin is connected to ground on application PCB level (see Table 12. GND Loss Detection). The bit is cleared upon a successful readout over PXN (clear by read bit).

PWM_CNT_OVF −

Table 49. CF0

Table 50. CF1 EEPROM WRITE DATA

Byte	Name	Contents				
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3

Table 52. CF3 EEPROM READ DATA

Byte	Name	Contents							
		Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	PID1	P	1	1	SA[4:0]				
1	PID2	P	0	0	CSID[4:0] = 0x03				
2	DATA1	WP	EES[1:0]		0	0	EESA[2:0]		
3	DATA2	EEBA[7:0]							
4	[redacted]								

Table 54. CF5 ASSIGN ADDRESS

Byte	Name	Contents							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	PID1	P	1	1	SA[4:0]				
1	PID2	P	0	0	CSID[4:0] = 0x05				
2	DATA1	B	0	0	AA_ADR[4:0]				
3	DATA2	AA_THR[7:0]							
4	CRC	CRC[7:0]							

DATA1

B broadcast bit:
1 – broadcast frame
0 – addressed frame

AA_ADR[4:0] 5 bit address to assign

DATA2

AA_THR[7:0] 8 bit auto address threshold value

The CF5 frame is accepted in AUTO_ADDR mode only.

Table 55. CF6 OP MODE STATUS

Byte	Name	Contents							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	PID1	P	1	1	SA[4:0]				
1	PID2	P	0	0	CSID[4:0] = 0x06				
2	DATA1	PXN_FRAME_CNT[3:0]				OPMODE[3:0]			
3	CRC	CRC[7:0]							

DATA1

PXN_FRAME_CNT[4:0] PXN frame counter – 4 bit counter which is incremented each time any valid PXN frame is processed. The counter overflows to 0 upon the increment.

OPMODE[3:0] OP mode status:

0x0	not valid	0x7	normal fail safe open mode
0x1	OTP config mode	0xC	NO_CRC direct mode
0x2	auto addressing mode	0xD	NO_CRC pwm mode
0x4	normal direct mode	0xE	fail safe OTP mode
0x5	normal pwm mode	0xF	fail safe OPEN mode
0x6	normal fail safe otp mode		

Table 56. CF7 SLAVE/REPEATER SLAVE PXN MODE SELECTION

Byte	Name	Contents							
		Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	PID1	P	1	1	SA[4:0]				
1	PID2	P	0	0	CSID[4:0] = 0x07				
2	DATA1	B	0	0	0	0	0	0	PMC
3	CRC	CRC[7:0]							

DATA1

B broadcast bit:
1 – broadcast frame
0 – addressed frame

PMC PXN mode control bit:
1 repeater slave mode
0 slave mode

Overwrites device mode loaded from the OTP memory.

Table 57. CF8 READ PXN MODE STATUS

Byte	Name	Contents
		Bit 7 repeater slave mode 0 slave mode

Table 58. CF9 WRITE DATA TO OTP

Byte	Name	Contents							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	PID1	P	1	1	SA[4:0]				
1	PID2	P	0	0	CSID[4:0] = 0x09				
2	DATA1	B	0	0	0	0	0	OTPBS[1:0]	
3	DATA2	BYTE0 @(OTPBA+0)							
4	DATA3	BYTE1 @(OTPBA+1)							
5	DATA4	BYTE2 @(OTPBA+2)							
6	DATA5	BYTE3 @(OTPBA+3)							
7	DATA6	0x00							
8	DATA7	CRC[7:0]							

DATA1

B — broadcast bit:
1 – broadcast frame
0 – addressed frame

OTPBS[1:0] — 2 bit OTP bank selection
0x2 – custom OTP bank
0x0, 0x1, 0x3 – no bank selected

DATA2 – DATA6

BYTEx[7:0] — bytes to be written

The CF9 frame is accepted in OTP_CONFIG mode only.

Table 59. CF10 REQUEST DATA FROM OTP

Byte	Name	Contents							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	PID1	P	1	1	SA[4:0]				

Table 60. CF11 READ DATA FROM OTP

Byte	Name	Contents							
		Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	PID1	P	1	1	SA[4:0]				
1	PID2	P	0	0	CSID[4:0] = 0x0B				
2	DATA1	LOCKB	OTPS[1:0]		0	0	OTPBS[2:0]		
3	DATA2	BYTE0 @(OTPBA+0)							
4	DATA3	BYTE1 @(OTPBA+1)							
5	DATA4	BYTE2 @(OTPBA+2)							
6	DATA5	BYTE3 @(OTPBA+3)							
7	DATA6	BYTE4 @(OTPBA+4)							
8	CRC	CRC							

DATA1

Table 61. CF12 COMMUNICATION SPEED

Byte	Name	Contents							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	PID1	P	1	1	SA[4:0]				
1	PID2	P	0	0	CSID[4:0] = 0x0C				
2	DATA1	B	0	0	0	0	0	CSPEED[1:0]	
3	CRC	CRC[7:0]							

DATA1

B broadcast bit:
1 broadcast frame
0 addressed frame

CSPEED[1:0] communication speed:
0x0 – 125 kbps
0x1 – 250 kbps (default)
0x2 – 500 kbps
0x3 – 1000 kbps

Overwrites device communication speed loaded from the OTP memory.

Table 62. CF13 SWITCH TO NORMAL MODE

Byte	Name	Contents							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	PID1	P	1	1	SA[4:0]				
1	PID2	P	0	0	CSID[4:0] = 0x0D				
2	DATA1	B	0	0	0	0	0	0	NMD
3	CRC	CRC[7:0]							

DATA1

B broadcast bit:
1 broadcast frame
0 – addressed frame

NMD request to enter NORMAL mode from OTP_CONFIG mode:
1 – go to NO_CRC or normal mode; according to the OTP bank 2 lock bit
0 – no effect

The CF13 frame is accepted in OTP_CONFIG mode only.

Table 63. CF14 RESET SYSTEM

Byte	Name	Contents							
		Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	PID1	P	1	1	SA[4:0]				
1	PID2	P	0	0	CSID[4:0] = 0x0E				
2	DATA1	B	0	0	0	0	0	0	SWRST
3	CRC	CRC							

DATA1

B broadcast bit:
1 broadcast frame
0 – addressed frame

SWRST software reset bit:
1 perform software reset
0 – no effect

The CF14 frame is accepted in NORMAL mode only.

Table 64. CF15 TRIGGER MAPENA

Byte	Name	Contents							
		Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	PID1	P	1	1	SA[4:0]				
1	PID2	P	0	0	CSID[4:0] = 0x0F				
2	DATA1	B	0	0	0	WDT_SEL[1:0]		0	MAPENA
3	CRC	CRC[7:0]							

DATA1

B broadcast bit:
1 broadcast frame
0 – addressed frame

MAPENA trigger MAPENA
1 – perform MAPENA
0 – no effect

WDT_SEL[1:0] watchdog8ee[d MAPENA0 – noech

THERMAL WARNING, ERROR DETECTION AND DIAGNOSTICS FEEDBACK

The NCV78343 offers a wide range of device−integrated diagnostic features. Their description follows.

Thermal Warning and Shutdown

The junction temperature can be calculated from ADC code as follows:

$$T_j = \frac{ADC_{CODE} + 184}{TSD_{CODE} + 184} \cdot TSD_{TEMPERATURE} + 273 - 273 \quad \text{(eq. 2)}$$

The <TSD_CODE> is trimmed in production to 170°C ($TSD_{TEMPERATURE}$). Typical value of <TSD_CODE> is 186 and exact trimmed value can be read from Register 0x12. The <TW> status bit is set high in case the measurement result is greater than or equal to TW_CODE[7:0] value. The <TSD> status bit is set high in case the measurement result is greater than or equal to TSD_CODE[7:0].

The TW and TSD status bits are cleared in case the

PXN CRC Code Example

```
// Byte reverse
uint8_t l343_byte_reverse(uint8_t  b)
{
   b = (b & 0xF0) >> 4 | (b & 0x0F) << 4;
   b = (b & 0xCC) >> 2 | (b & 0x33) << 2;
   b = (b & 0xAA) >> 1 | (b & 0x55) << 1;
   return b;
}

// Calculate the PXN CRC
uint8_t l343_pxn_crc(uint8_t *data_bits, uint8_t length)
{
    uint8_t CRC8 = 0;

    for (uint8_t a=0; a<length; ++a)
    {
        CRC8 = CRC8 ^ data_bits[a];

        for (uint8_t i=0; i<8; ++i)
        {
            if (CRC8 & 0x80)
            {
                CRC8 = ((0x7F & CRC8) * 2) ^ 0x07;
            }
            else
            {
                CRC8 = CRC8 * 2;
            }
         }
    }

    return CRC8;
}

Function call example:
int main(void)
{
    // MAPENA as broadcast; SEED, PID1, PID2, DATA0; CRC = 0x28
    uint8_t data_bits[] = {0xFF, 0x61, 0x8F, 0x81};

    // Write to REG00; SEED, PID1, PID2, DATA0, DATA1, DATA2; CRC = 0x14
    // uint8_t data_bits[] = {0xFF, 0xA1, 0x80, 0x55, 0x05, 0x00};

    // Get the number of bytes
    int length = sizeof(data_bits)/sizeof(data_bits[0]);

    // Invert the input
    for (uint8_t a = 0; a<length; ++a) data_bits[a] = l343_byte_reverse(data_bits[a]);

    // Get the result
    uint8_t result = l343_pxn_crc(data_bits, length);
}
```

Parity Bit Calculation

```
uint8_t l343_parity(uint8_t val)
{
    bool Par;

    Par = ((val>>0)&1) ^ ((val>>1)&1) ^ ((val>>2)&1) ^ ((val>>3)&1) ^ ((val>>4)&1) ^
    ((val>>5)&1) ^ ((val>>6)&1);
    Par = (Par ^ 1) & 1;

    return (uint8_t)Par;
}
```

Go to NMD Frame (CF13)

```
int32_t l343_normal_mode(uint8_t addr)
{
    uint8_t p;
    uint8_t pdata[5];

    // SYNC
    pdata[0] = 0x55;

    // PID1
    uint8_t PID1 = 0;
    PID1 = 3<<5 | addr;
    p = l343_parity(PID1);
    pdata[1] = (p << 7) | PID1;

    // PID2
    uint8_t PID2 = 0;
    PID2 = 0x0D;
    p = l343_parity(PID2);
    pdata[2] = (p << 7) | PID2;

    // DATA bytes
    pdata[3] = 1;        // NMD


    uint8_t pdata_crc[4];

    // Invert the input
    for (uint8_t a = 0; a<4; ++a) pdata_crc[a] = l343_byte_reverse(pdata[a]);

    // Calculate the CRC
    pdata_crc[0] = 0xFF;
    uint8_t crc = l343_pxn_crc(pdata_crc, 4);

    pdata[4] = crc;

    // Send data
    return serial_pxn_set_data(pdata, 5);
}
```

OTP Write Code Example

```
typedef struct OTP_t
{
            unsigned lb: 1;           // Lock bit
            unsigned na_lb: 1;        // Node Address Lock bit
            unsigned na: 5;           // Note Address
            unsigned fss_lb: 1;       // Fail Safe State Lock bit
            unsigned fss: 4;          // Fail Safe State of LEDs
            unsigned pxn_lb: 1;       // PXN Lock bit
            unsigned mode: 1;         // Mode
            unsigned cs: 2;           // Communication speed
            unsigned gbed: 1;         // Global bit error detection
            unsigned m_lvds_off: 1;   // M−LVDS off
            unsigned uart_off: 1;     // UART off
            unsigned ee_lb: 1;        // EEPROM lock bit
            unsigned crc: 7;          // CRC
} OTP_t;

// Calculate the OTP CRC
uint8_t l343_otp_crc(uint8_t *data_bits, char length)
{
            uint8_t CRC7 = 0;

            for (uint8_t a = 0; a<length; ++a)
            {
                        CRC7 = CRC7 ^ data_bits[a];

                        for (uint8_t i = 0; i < 8; ++i)
                        {
                                    if (CRC7 & 0x80)
                                    {
                                                CRC7 = ((0x7F & CRC7) * 2) ^ (0x37 * 2);
                                    }
                                    else
                                    {
                                                CRC7 = (CRC7 * 2);
                                    }
                        }
            }

            return (CRC7 / 2);
}

int32_t l343_otp_zapping(OTP_t otp)
{
            uint8_t data_bits[4];
            data_bits[0] = 0x07;
            data_bits[1] = 0x0F<<4 | otp.ee_lb<<3 | otp.uart_off<<2 | otp.m_lvds_off<<1 | otp.gbed<<0;
            data_bits[2] = otp.cs<<6 | otp.mode<<5 | otp.pxn_lb<<4 | otp.fss<<0;
            data_bits[3] = otp.fss_lb<<7 | otp.na<<2 | otp.na_lb<<1 | otp.lb<<0;

            // Get the number of bytes
            int length = sizeof(data_bits)/sizeof(data_bits[0]);

            // Get the result
            otp.crc = l343_otp_crc(data_bits, length);

            // PXN OTP write frame
```

```
            uint8_t p;
            uint8_t pdata[10];

            // SYNC
            pdata[0] = 0x55;

            // PID1
            uint8_t PID1 = 0;
            PID1 = 3<<5 | otp.na;
            p = l343_parity(PID1);
            pdata[1] = (p << 7) | PID1;

            // PID2
            uint8_t PID2 = 0;
            PID2 = 0x09;
            p = l343_parity(PID2);
            pdata[2] = (p << 7) | PID2;


            // DATA bytes
            const uint32_t data =  otp.lb |
                                   otp.na_lb << 1 |
                                   otp.na << 2 |
                                   otp.fss_lb << 7 |
                                   otp.fss << 8 |
                                   otp.pxn_lb << 12 |
                                   otp.mode << 13 |
                                   otp.cs << 14 |
                                   otp.gbed << 16 |
                                   otp.m_lvds_off << 17 |
                                   otp.uart_off << 18 |
                                   otp.ee_lb << 19 |
                                   otp.crc << 20;

            pdata[3] = 0x02;
            pdata[4] = ((data >> 0 ) & 0xFF);
            pdata[5] = ((data >> 8 ) & 0xFF);
            pdata[6] = ((data >> 16) & 0xFF);
            pdata[7] = ((data >> 24) & 0xFF);
            pdata[8] = 0x00;


            uint8_t pdata_crc[9];

            // Invert the input
            for (uint8_t a = 0; a<9; ++a) pdata_crc[a] = l343_byte_reverse(pdata[a]);

            // Calculate the CRC
            pdata_crc[0] = 0xFF;
            uint8_t crc = l343_pxn_crc(pdata_crc, 9);

            pdata[9] = crc;

            // Send data
            return serial_pxn_set_data(pdata, 10);
}
```

```
Function call example:
int main(void)
{
     // Zap the OTP
     //            lb   nalb   na   fss_lb fss  pxn_lb  mode  CS   GBED  lOff  uOff  eeLb  crc
     OTP_t otp = {0x01, 0x01, 0x01, 0x01, 0x0F, 0x01,  0x00, 0x01, 0x00, 0x00, 0x01, 0x00, 0x00};
     l343_otp_zapping(otp);
}
```

Auto addressing (AA) Process

This example is valid for two devices, where the first one is in repeater−slave mode and the second one is in slave mode. The first device is connected to the MCU via UART and the second device is connected to the first device via M−LVDS.

The slave address (SA) for the first device will be set to ‘1’ and the SA for the second device will be set to ‘7’.

The AA process combines benefits of current source and connected LED string. The application does not need any additional wires. When the device is connected to the LED string and the current source for this LED string is enabled, the voltage drop across the LED string will occur. The LED string voltage VLED is measured by the device. Thus the address may be assigned to specific device.

In general, the MCU sends a broadcast frame CF4 (see Table 53) to all node devices and the second broadcast frame CF5 (see Table 54) with the VLED threshold and new device address as parameters. Doing this, all devices on the node will be in AA mode and only the device with VLED higher than set threshold will assign new address.

For this example, the LED string voltage is 33 V (127 ADC code). The auto−addressing threshold will be set to 80.

1. Disable all buck outputs, thus the LEDs are not shining.
2. Enable buck output 1, so the LED string for the device 1 is shining.
3. Enable AA mode by sending CF4 as broadcast (B=1; AAC=1); see Table 53.
4. Assign the address for the device 1 by sending CF5 as broadcast (B=1; ADR=1; THR=80); see Table 54.
5. Disable buck output 1, so the LED string for the device 1 is not shining.
6. Disable the AA mode for the first device SA=1 by sending CF4 (B=0; AAC=0); see Table 53.
7. Force the normal mode for the first device SA=1 by sending CF13 (NMD=1); see Table 62.
8. Set the first device as repeater−slave SA=1 by sending CF7 (PMC=1); see Table 56.
9. Enable buck output 2, so the LED string for the device 2 is shining.
10. Enable AA mode by sending CF4 as broadcast (B=1; AAC=1); see Table 53.
11. Assign the address for the device 2 by sending CF5 as broadcast (B=1; ADR=7; THR=80); see Table 54.
12. Disable buck output 2, so the LED string for the device 2 is not shining.
13. Disable the AA mode for the second device SA=7 by sending CF4 (B=0; AAC=0); see Table 53.
14. Force the normal mode for the second device SA=7 by sending CF13 (NMD=1); see Table 62.

For multiple devices connected to the first one via M−LVDS, please repeat steps 9−14 with different ADR, THR, SA values.

```
// LED brightness; the length should be DEVICES*REGISTERS; or two-dimensional array might be used
uint16_t DC[DEVICES*REGISTERS];         // values are in a range of [0; 1023]
// uint16_t DC[DEVICES][REGISTERS];

// Drimfreq for each device
uint8_t dimfreq[DEVICES]

// Function call example:
// Send ON, OFF, TR values to all devices
void l343_send(uint16_t *DC)
{
        for (uint8_t dev = 0; dev<DEVICES; ++dev)
        {
            uint16_t ON[REGISTERS];
            uint16_t OFF[REGISTERS];
            uint16_t TR[REGISTERS];

            // Calculate the dimming values
            l343_dimming(dimfreq[dev], DC+REGISTERS*dev, ON, OFF, TR);
//          l343_dimming(dimfreq[dev], DC[dev], ON, OFF, TR);        // when two-dimensional array is used

            // Fill the registers values
            uint32_t reg[REGISTERS];

            for (uint8_t r = 0; r<REGISTERS; ++r)
            {
                reg[r] = (ON[r]<<14) | (OFF[r]<<4) | (TR[r]&0xF);
            }

            // 36 bytes need to be sent in 3 frames
            uint8_t RBA[3] = {1, 5, 9};

            for (uint8_t r = 0; r<3; ++r)
            {
                uint8_t p;
                uint8_t pdata[16];

                // SYNC
```

```
pdata[2] = p<<7 | PID2;

// DATA bytes
pdata[3] = reg[r*4+0]&0xFF;
pdata[4] = (reg[r*4+0]>>8)&0xFF;
pdata[5] = (reg[r*4+0]>>16)&0xFF;

pdata[6] = reg[r*4+1]&0xFF;
pdata[7] = (reg[r*4+1]>>8)&0xFF;
pdata[8] = (reg[r*4+1]>>16)&0xFF;

pdata[9] = reg[r*4+2]&0xFF;
pdata[10] = (reg[r*4+2]>>8)&0xFF;
pdata[11] = (reg[r*4+2]>>16)&0xFF;

pdata[12] = reg[r*4+3]&0xFF;
pdata[13] = (reg[r*4+3]>>8)&0xFF;
pdata[14] = (reg[r*4+3]>>16)&0xFF;

uint8_t pdata_crc[16];

// Invert the input
for (unsigned cI q6fa.754 618[; F;
```

Return to Normal Operation after Fail Safe Mode

Once a device detects one of the following status bits: TSD or CAP_UV or VBB_LOW or DIMERR or TIMEOUT, the device enters the fail−safe mode (see Operating Modes section). To leave this mode, the superior system shall read out the REG 0x10 (see Table 45) and/or handle the error if required.

The device enters FAIL−SAFE OTP mode when DIMERR or TIMEOUT appears. When TIMEOUT is set and this mode is entered, switches are set according to the OTP memory values. If the OTP memory is not zapped, the switches are switched OFF. Once the error status bit is cleared, the switches remain unchanged. When DIMERR is set and this mode is entered, switches operation is unaffected.

The device enters FAIL−SAFE OPEN mode when TSD or CAP_UV or VBB_LOW appears. In this mode, the switches are automatically switched OFF. Once the error status bit is cleared, the switches are set according to the values in REG 0x00 (see Table 27).

The TSD/CAP_UV/VBB_LOW group of bits (hardware fail) have higher priority to the DIMERR/TIMEOUT group of bits (application fail). When these two groups appear at the same time, the device enters the FAIL−SAFE OPEN mode.

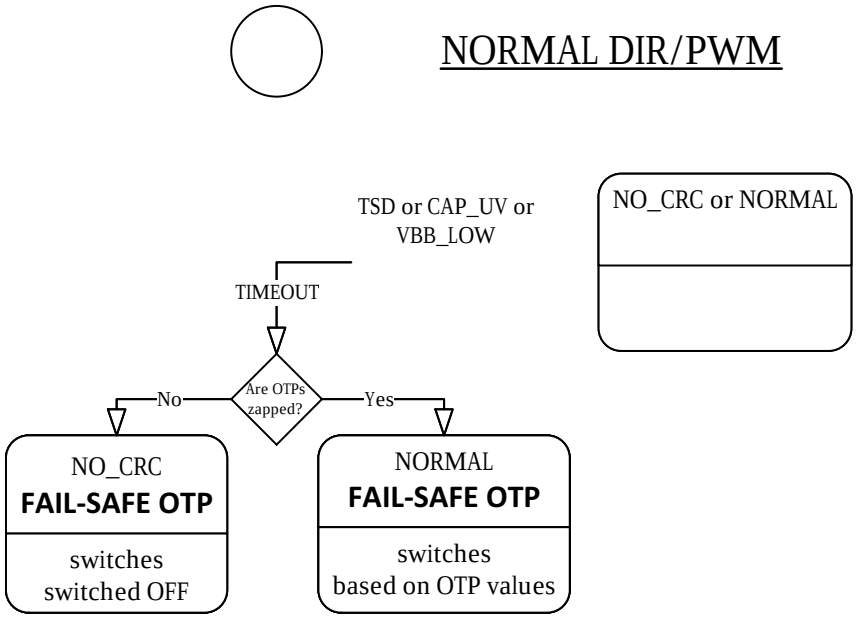


Figure 19. FAIL SAFE Modes

Power Up and Down Sequences

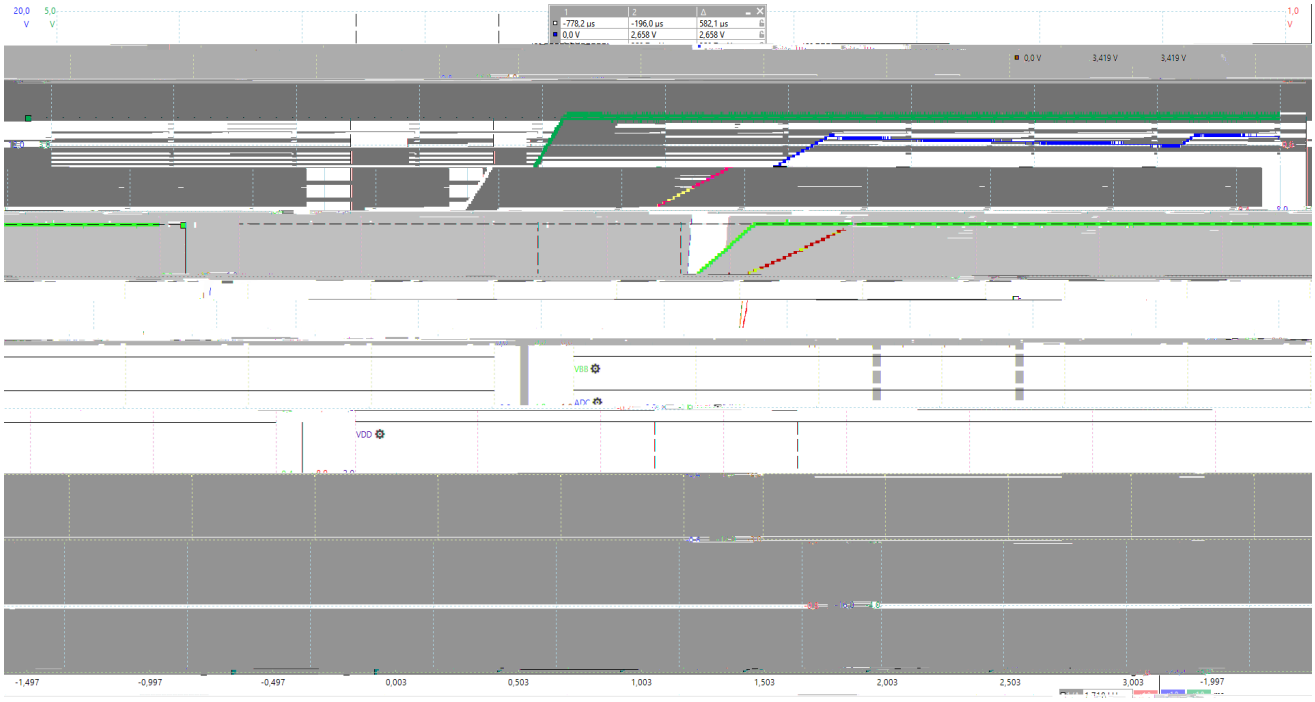


Figure 20. Power up Sequence with 10 nF at ADC2/ADR Pin

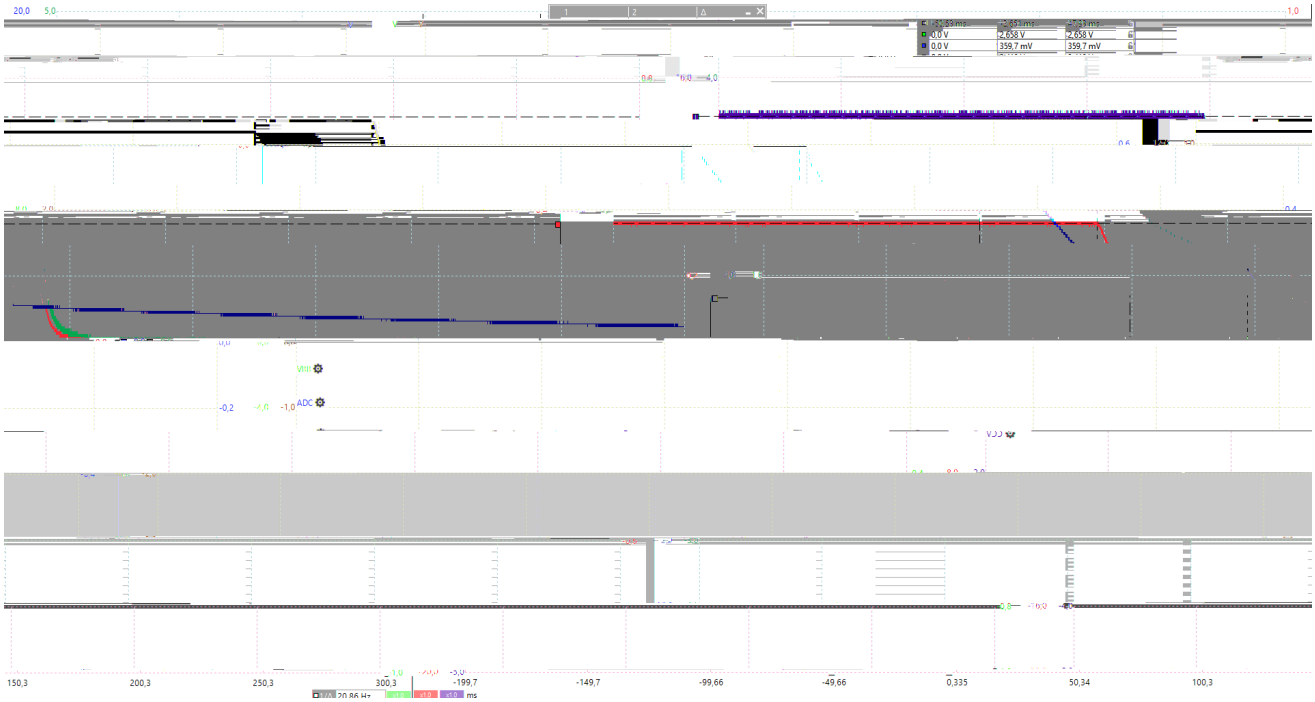


Figure 21. Power down Sequence with 10 nF at ADC2/ADR Pin

Example Flow Chart Diagram for the Normal Operational Mode

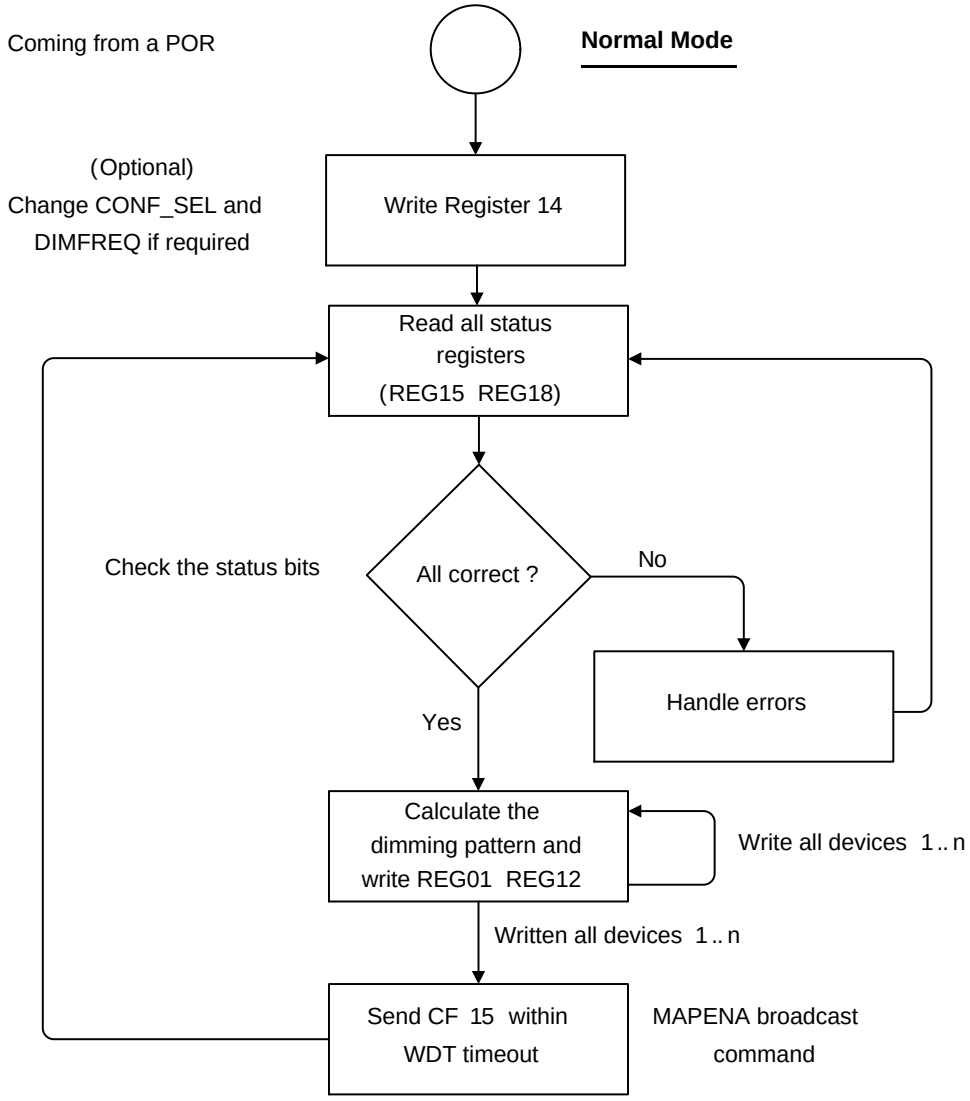


Figure 22. Normal Operation Mode Flow Chart Diagram

The main loop should consist of checking all status bits and handling them if necessary. Set a refresh rate for common headlamp lighting functions (e.g. High beam) as well as fulfill watchdog timeout. The MCU should calculate

Flow Chart after POR

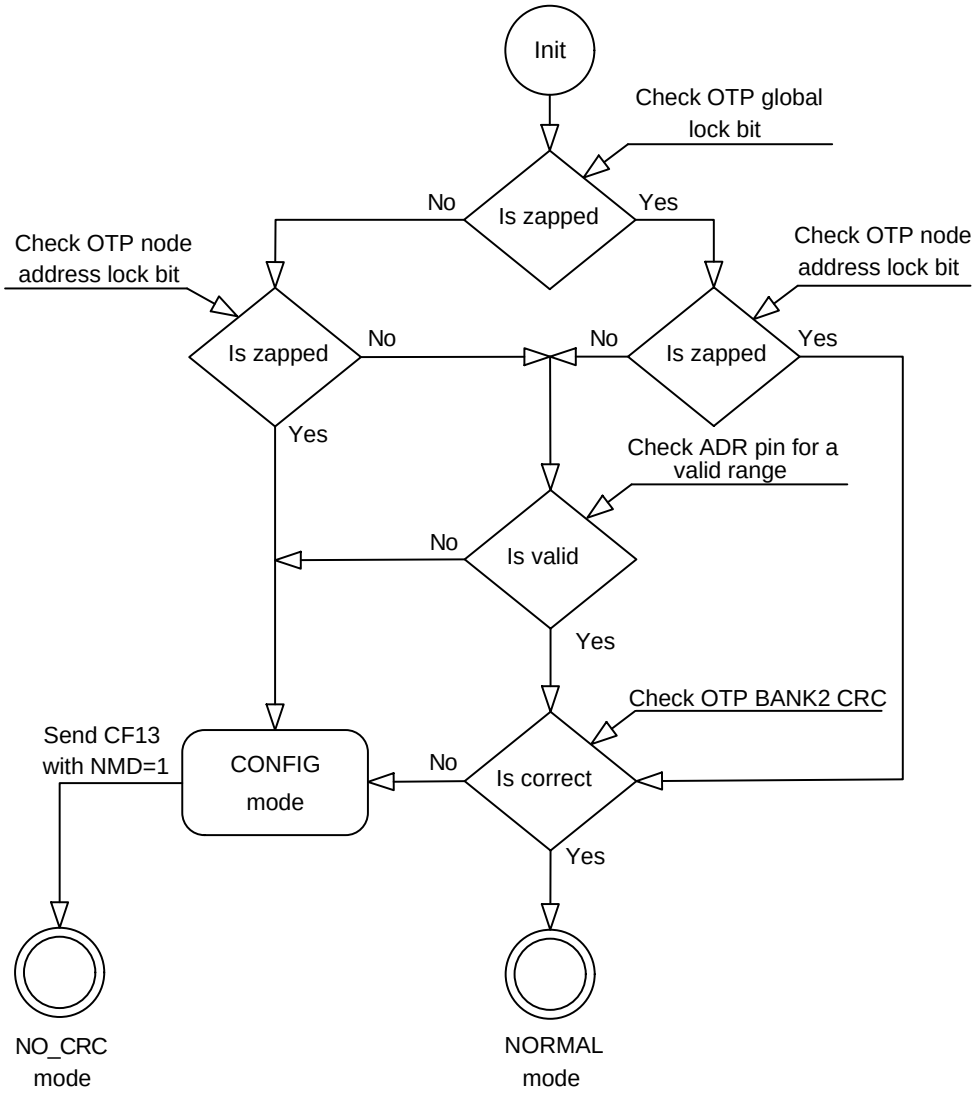


Figure 23. Flow Chart Diagram after POR

The diagram above is an automatic flow after each POR. A device might end up in either CONFIG or NORMAL or NO_CRC mode according to zapped OTP bits.

A new device will end up in CONFIG mode, because OTP bits are not zapped. An address is set by either Auto−addressing process or by Multi−level addressing using a voltage divider or zapped in the OTP memory.

Multi level Addressing Procedure with Long Time Delay at ADC2/ADR Pin

The following flow chart is valid for the repeater−slave and slaves cluster, where the repeater−slave communicates through CAN−PHY layer and slaves are connected via local

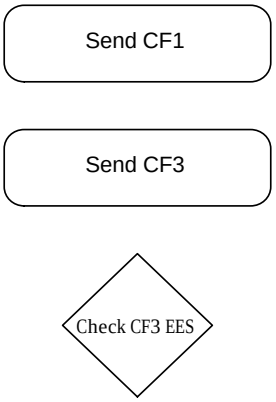


Figure 25. EEPROM Write Operation

F