AN5023
Application note

L9680 airbag system IC

Introduction

The L9680 is an advanced airbag system chip solution targeted for airbag high-end market.

This device is family compatible with the L9678, L9679 and L9679E devices.

This document is an integration of the application note (AN4437) developed for L9678 and available in STMicroelectronics web site.

Safety system integration is enabled through higher power supply currents and integrated active wheel speed sensor interface.

System designs can be completed using L9680, SPC560Px microcontroller and eventually companion chip i.e. L9679E.

The active wheel speed interface is shared with the PSI-5 satellite interface to create a generic remote safety sensor interface compliant to both systems.

Contents

List of tables

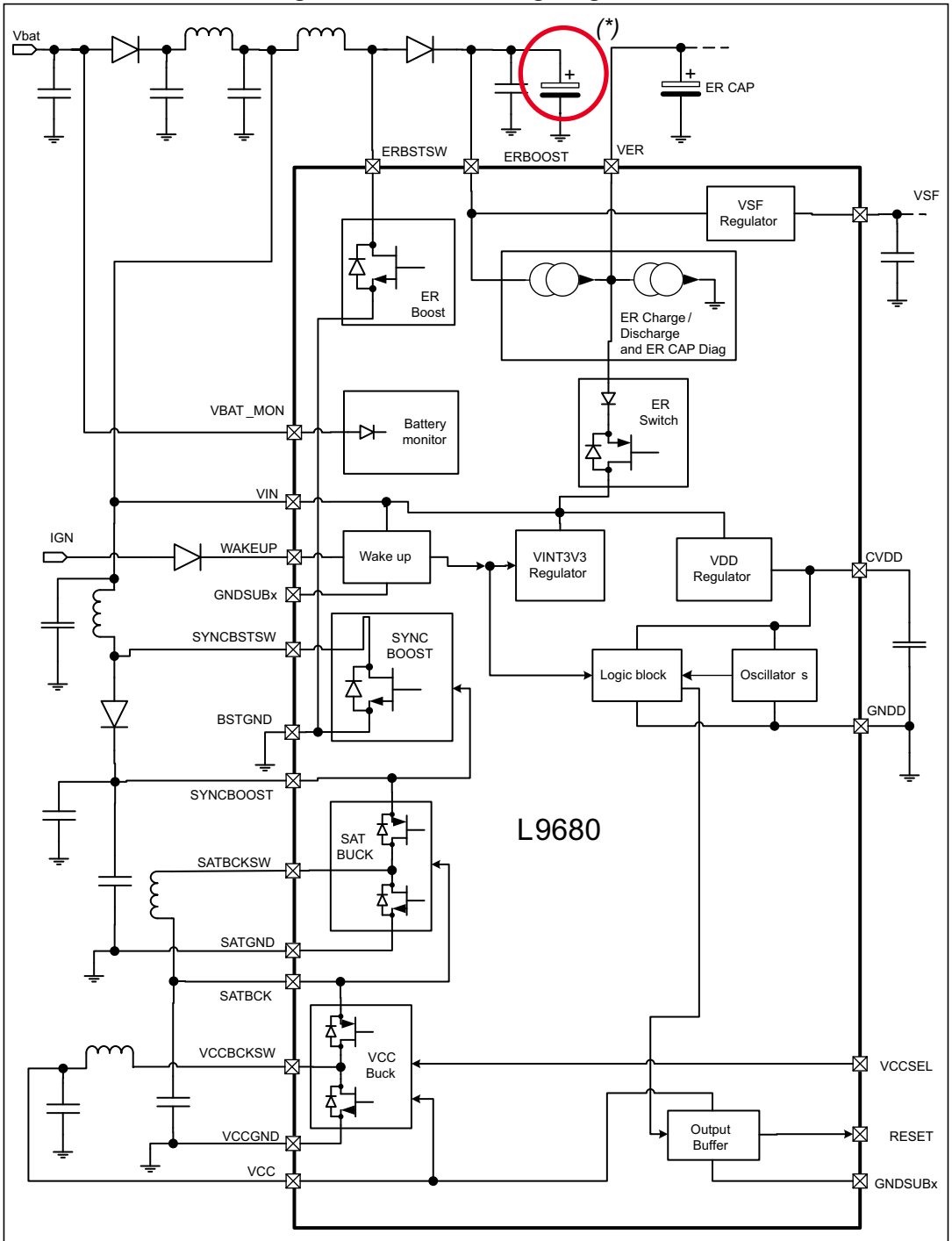
List of figures

1 Detail features

- Energy reserve voltage power supply - ERBOOST
  - High frequency boost regulator, 1.875 MHz
  - Output voltage user selectable, 23 V or 33 V ±5%
  - Measurement of reserve capacitor value & ESR diagnostics
- Output voltage regulator power supply for PSI-5 SYNC pulse - SYNCBOOST
  - High frequency boost regulator, 1.875 MHz
  - Output voltage user selectable, 12 V/14.75 V ±5%
- Output voltage regulator power supply for remote sensor - SATBCK
  - High frequency buck converter, 1.875 MHz
  - Output voltage user selectable, 7.2 V/9 V ±4%
- Output voltage regulator power supply for linear and logic - VCC
  - High frequency buck converter, 1.875 MHz
  - Output voltage user selectable, 3.3 V or 5.0 V ±4%
- Integrated crossover switch
  - Crossover performance 3 Ω - 980 mA max
  - Switch active output indicator (COVRACT pin)
- Battery voltage monitor & shutdown control with Wake-up control
- System voltage diagnostics with integrated ADC
- Squib deployment drivers
  - 12 channel HSD/LSD
  - 25 V max deployment voltage
  - Various deployment profiles, 1.2 A/1.75 A, x*0.064 ms up to 4.032 ms
  - Current monitoring
  - Rmeasure, STB, STG & Leakage diagnostics
  - High & low side driver FET tests
  - Safing FET test
- High side safing switch regulator and enable control
- Four channels remote sensor interface for PSI-5 (synchronous mode) or active wheel speed sensors
- Three channels GPO, HSD or LSD configurable, with PWM 0-100% control
- Nine channels Hall-effect, resistive or switch sensor interface
- User customizable safing logic
- Specific disarm signal for passenger airbag
- Temporal and Algorithmic Watchdog timers
- End of life disposal interface
- Temperature sensor
- 32 bit SPI communications
- Minimum operating voltage = 5.5V at device battery pin
- Operating temperature, -40 °C to 95 °C
- Packaging - 100 pin

2 Device mechanization overview

Figure 1. Device's voltage regulators



(*) Capacitor on ERBOOST can be selected smaller in case ERBOOST is set to remain ON also in Autarchy mode (ER STATE). This selection is chosen setting b[12] = Keep_ERBoost_ON in $01 SYS_CFG (INIT). Recommended value is 2.2µF (1µF min) low ESR, otherwise its value is higher, 100µF-200uF low ESR.

GAPGPS03089

2.1 External components summary

The ASD551C has been designed in order to be compatible with all the ASIC of the family, ie L9678, L9679 and L9680. It is possible either to solder the ASIC or mount a socket adapter.

The board has been designed in order to provide the means to host an external microcontroller daughter-board.

Most external components are standard design practice devices, such as filter capacitors. The user selectable components typically are determined by system requirements or operating profiles. The following sections outline analysis to derive some key system components.

See *Bill of materials on page 96*.

2.1.1 Component layout considerations

Placement of the component's regulators on the layout has to be evaluated with care, focusing on the regulation loops that have to be optimized from area point of view. Loops that have to be managed with care are highlighted in different colors, one per loop, the other components, left in black, can be located on the board with no particular restrictions.

Two loops related to BSTGND:

1. Loop 1: D1 and C7 must be placed in order to minimize the loop area between ERBSTSW (pin 84) and BSTGND (pin 85)
2. Loop 2: D12 and C85_1 must be placed in order to minimize the loop area between SYNCBOOSTSW (pin 86) and BSTGND (pin 85)

Two loops related to SATGND:

1. Loop 3: C81 must be placed in order to minimize the loop area between SYNCBOOST (pin 87) and SATGND (pin 89)
2. Loop 4: C2_1 must be placed in order to minimize the loop area between SATBCK (pin 90) and SATGND (pin89); loop area minimization is relevant but less relevant respect to loop 3

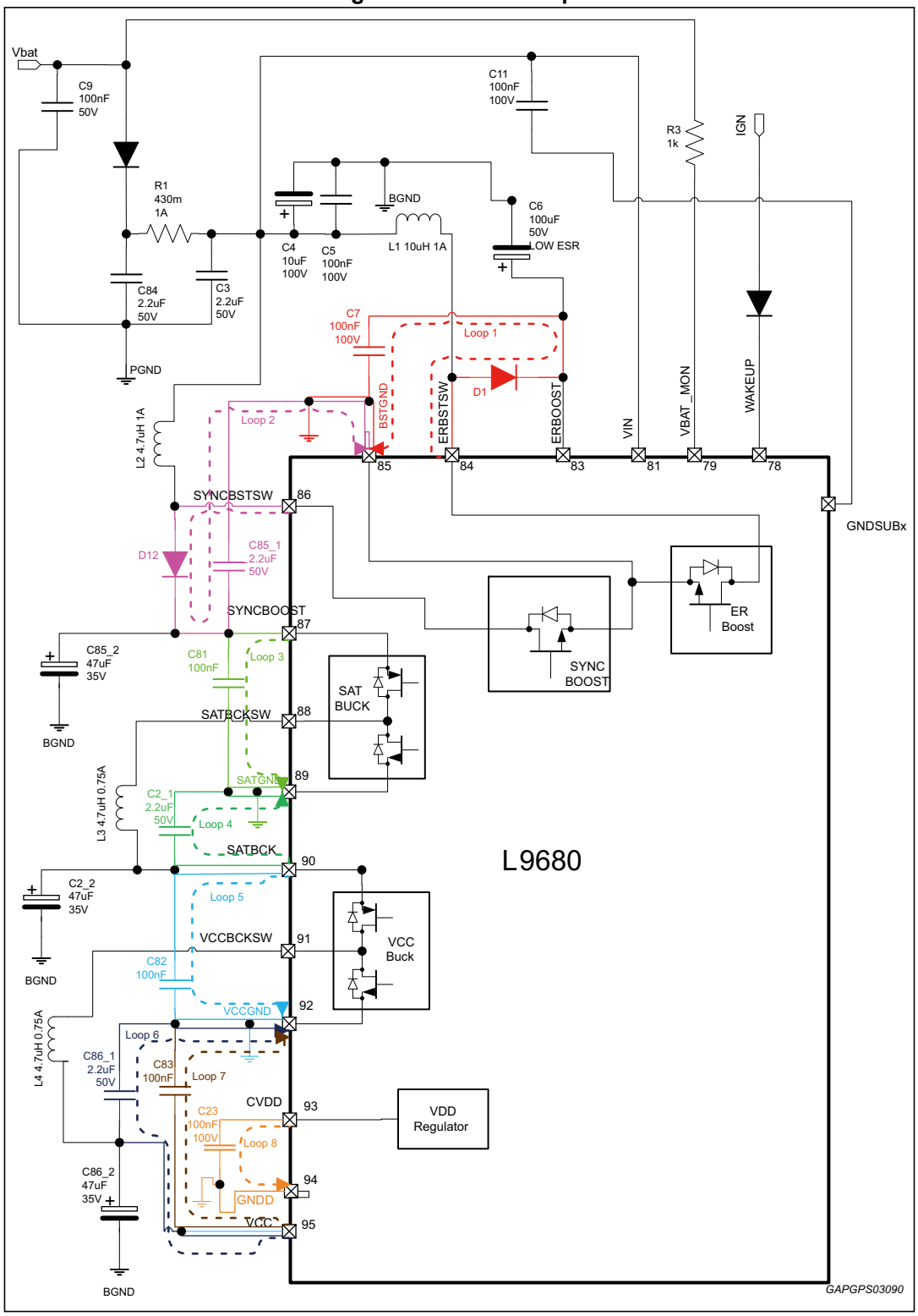
Three loops related to VCCGND:

1. Loop 5: C82 must be placed in order to minimize the loop area between SATBCK (pin 90) and VCCGND (pin 92)
2. Loop 6: C86_1 must be placed in order to minimize the loop area between VCC (pin 95) and VCCGND (pin 92); loop area minimization is relevant but less relevant respect to loops 5 and 7
3. Loop 7: C83 must be placed in order to minimize the loop area between VCC (pin 95) and VCCGND (pin 92)

One loop related to GNDD:

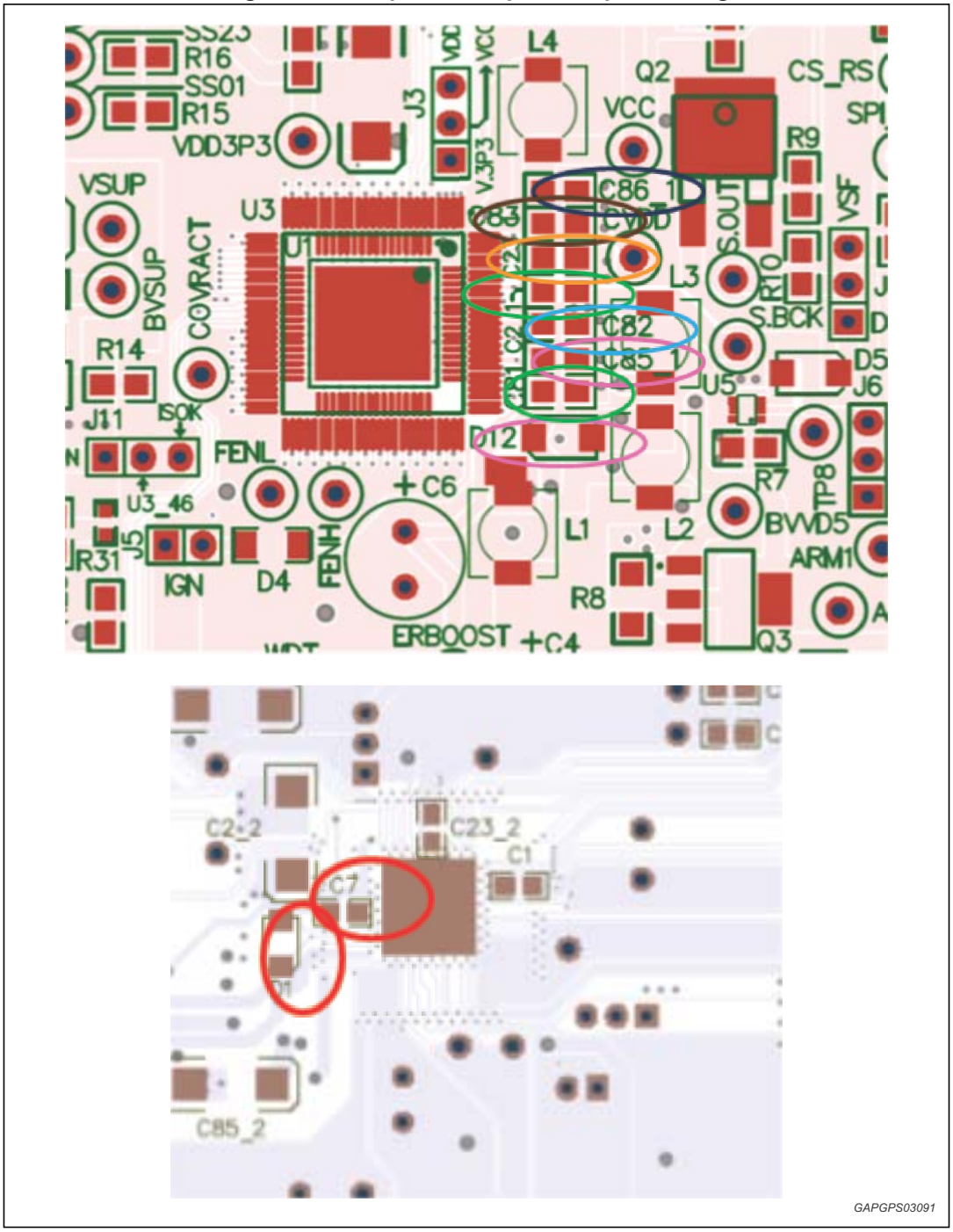
1. Loop 8: C23 must be placed in order to minimize the loop area between CVDD (pin 93) and GNDD (pin 94)

Figure 2. Device's loops



Referring to the regulation loops, an example of the components' disposition is:

Figure 3. Example of components positioning



2.1.2 ESD and SPIKE protection

To protect against ESD stress, a low impedance path to ground must exist through the device.

An effective approach for protecting any electronic system against ESD is to mechanically minimize the path ways by which high voltages enter the system from the outside environment.

Using external components, a common method to protect the IC from ESD events is to add a small serial resistor in-line between the ESD energy source and the device pin to be protected.

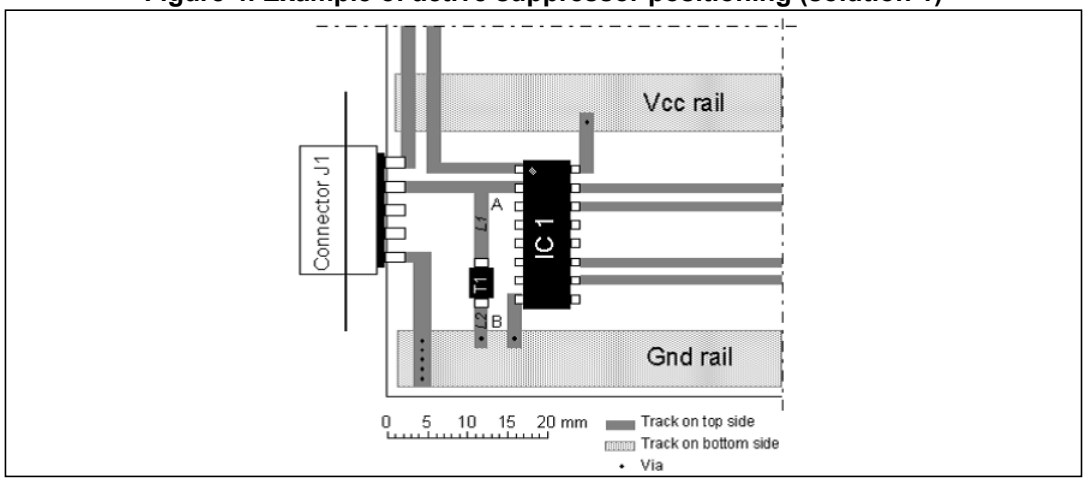
This method improves the ESD system robustness for two reasons:

1. The resistor's parasitic capacitor (typically in the range of 5 pF to 10 pF) slows the events
2. The resistor, in combination with the internal pin impedance, forms a voltage divider so that only a part of the high ESD voltage reaches the pin itself

In case of input/output lines that are susceptible to ESD it is suggested to add an external active component, TVS (transient voltage suppression) or Zener diode, which shunts the high energy of the ESD stress before it can reach the device pin.

The following examples are aimed at optimizing ESD protection device placement and the PCB layout to reach the best ESD performance possible.

Figure 4. Example of active suppressor positioning (solution 1)



It is here highlighted the parasitic inductance influence of the PCB track with respect to the insertion of an active suppressor, T1. An ESD event shows a rise time within 1 ns, attaching importance to the track parasitic inductances named L1 and L2 in the picture.

As an example, let's consider L1 and L2 the parasitic inductance associated to the two tracks named L1 and L2 in *Figure 4*, estimate 1nH/mm their specific inductance and di/dt = 20 A/ns. Voltage at pin is so determined:

$$V_{CL} + (L_1 + L_2) * \frac{di}{dt} =$$

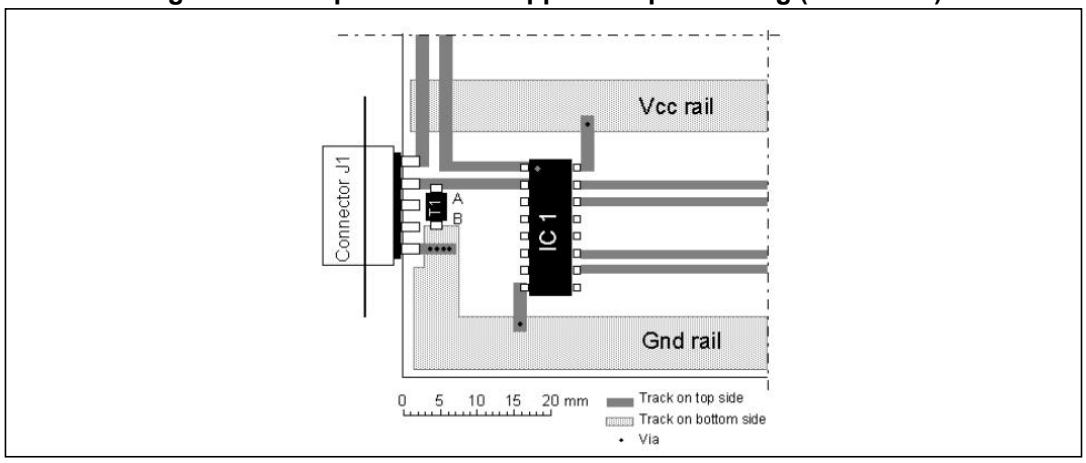
$$= V_{CL} + (10 + 5)nH * 20 \frac{A}{ns} =$$

$$= V_{CL} + 300\,V$$

GAPGPS03093

Where V_{CL} is the clamp voltage of the T1.

Figure 5. Example of active suppressor positioning (solution 2)



This solution reduces considerably the parasitic inductance of the track interposed between the ESD source and the point to be protected.

In this case the value of L1 and L2 is strongly reduced and the resulting spike at the pin is closer to V_{CL}.

2.1.3 Battery input diode

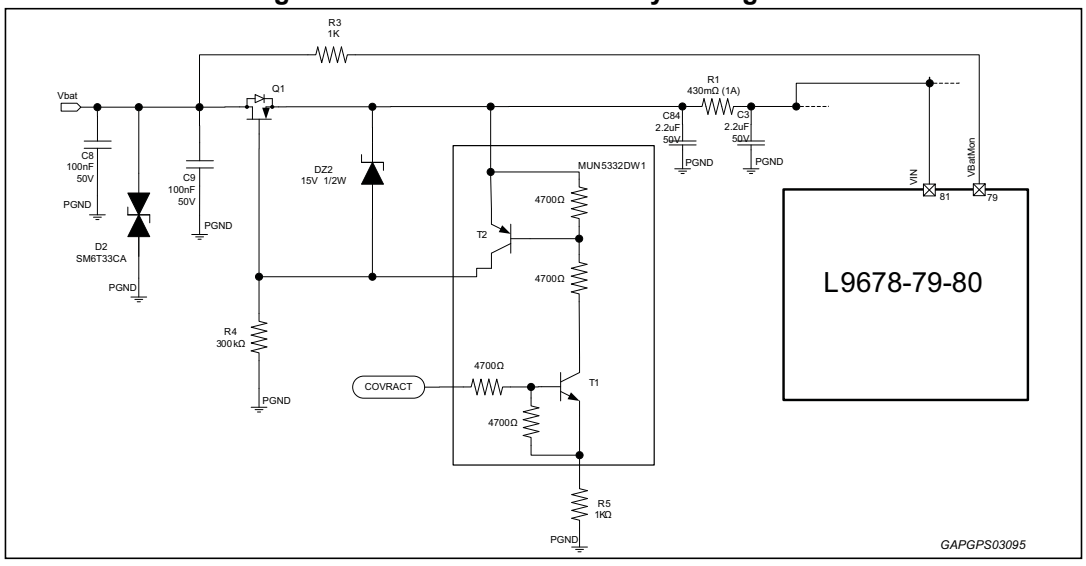
The system has been designed to support a reverse diode protection.

In case of very low voltage operation, in order to reduce the drop across this diode, the device provides the means to implement an active reverse protection, as showed below.

During normal operation, being COVRACT signal not asserted, T1 and T2 are kept OFF and PMOS Q1 is ON (VSG(Q1)~VBAT), showing at its terminals its Rdson which minimizes the voltage drop from battery.

When the IC moves in autarchy mode, COVRACT signal is asserted, Q1 is switched OFF and its intrinsic diode decouples VBAT from VIN;
Vin is connected through crossover switch at the external reserve capacitor.

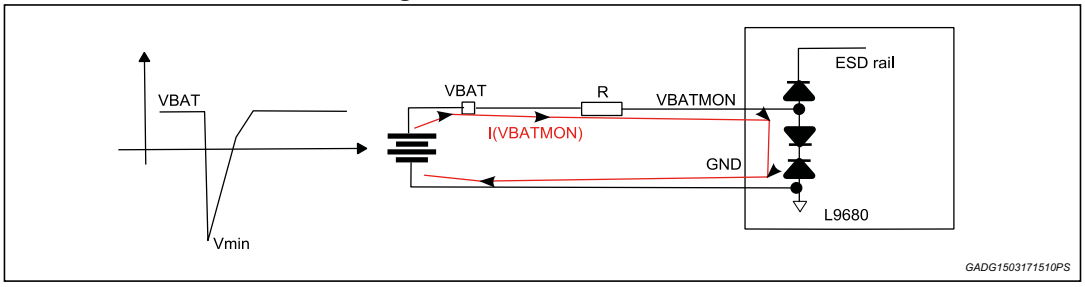
Figure 6. COVRACT - low battery management



VBATMON pin is designed to be connected before the reverse protection diode, so far it is able to sustain down to -18 V.

For the applications which require to survive at a deeper under-voltage, the pin is designed to sustain 20 mA max reverse current. To limit the current in VBATMON pin it is possible to insert a series resistor. Its value is defined as shown below:

Figure 7. Protection circuit



$$I(VABATMON) = \frac{|Vmin - (-18V)|}{R} \leq 20mA$$

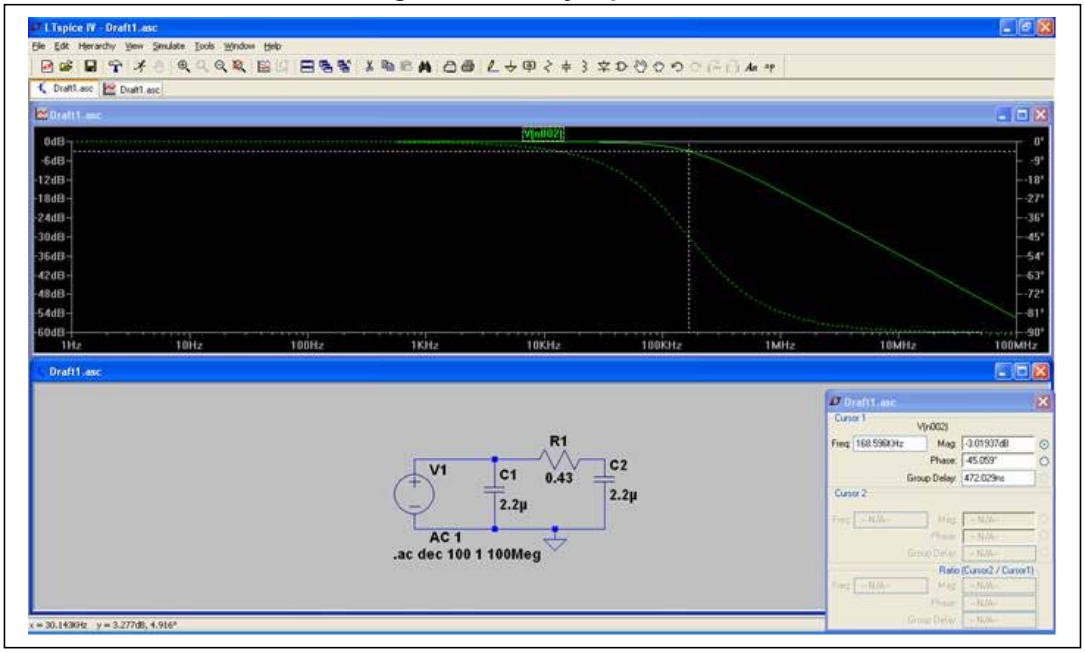
$$R \geq \frac{|Vmin + 18V|}{20mA}$$

GADG1503171530PS

2.1.4 Battery input PI filter (R1, C2, C3)

Battery current is influenced primarily by the energy reserve boost regulator operation. The boost clock frequency is nominally 1.875 MHz. The filter 3 dB corner frequency is set to 200 kHz. To reduce R1 voltage drop and power dissipation, the value is kept low while also allowing reasonable C2 & C3 component values. All components can be adjusted as necessary for the application.

Figure 8. Battery input filter



2.1.5 Energy reserve capacitor (C6)

Energy reserve capacitor stores the necessary energy to operate the Airbag ECU during loss of battery. System operating requirements influence the device selection and calculations. The following example makes general operating assumptions and changing the assumptions may effect calculations and results.

During loss of battery operation, energy reserve operation can be mechanized as shown. To continue the analysis, system functional operating assumptions must be determined. These assumptions are shown in the drawing below. Operation assumes three states, sensing, deploy and shutdown. During sensing state, all functions operate normally. In deploy state, all functions remain operational and all squib deployment channels are fired. The final state, shutdown, reduces operation to the microcontroller only.

Figure 9. Blocks supplied through energy reserve capacitor

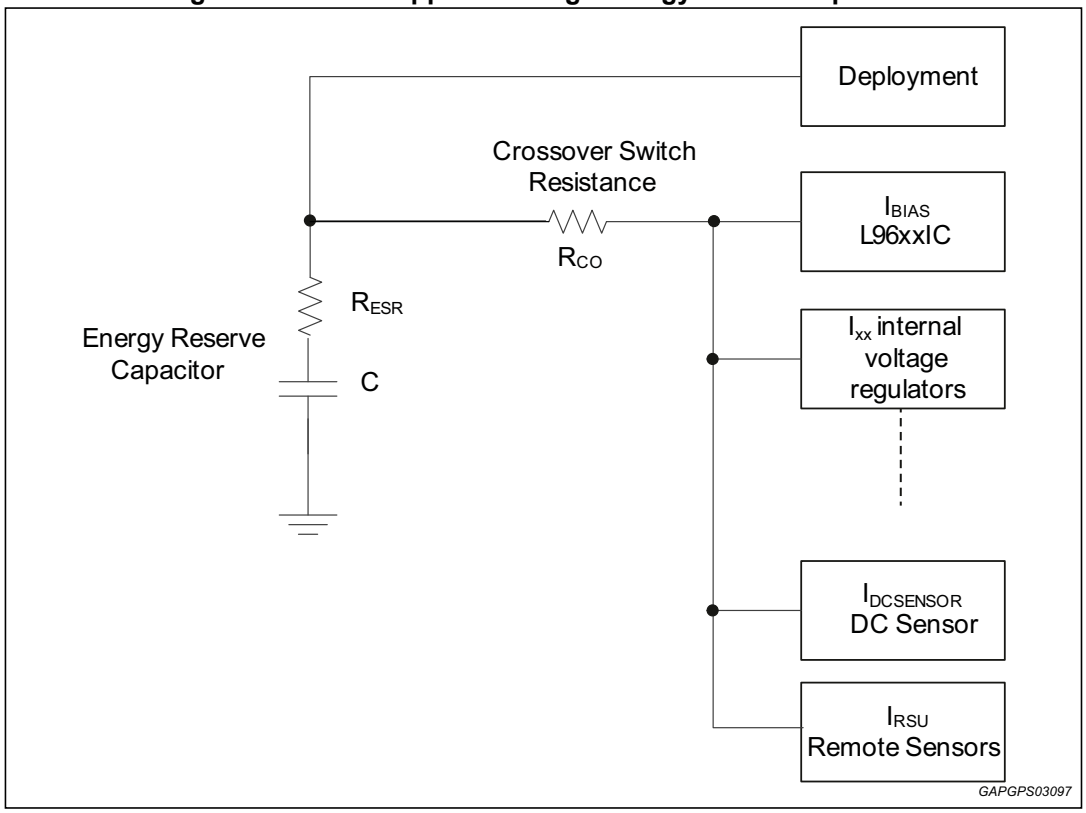
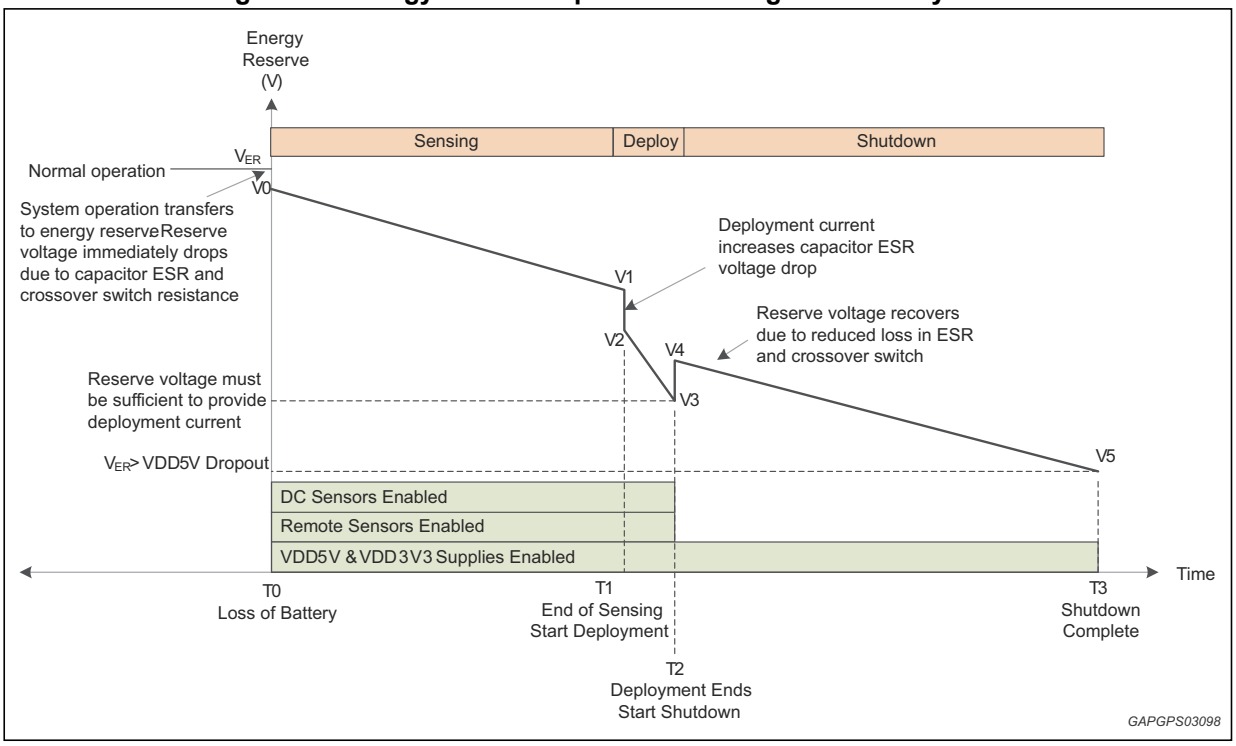


Figure 10. Energy reserve capacitor discharge in autarchy mode



At t = T0 = 0, $V_0 = V_{ER} - I_{SYS} \bullet (R_{ESR} + R_{CO})$

Where:

V_{ER} = Energy reserve voltage just prior to loss of battery detection and crossover operation

I_{SYS} = System current consumption, L96xx bias, Voltage regulators, RSUs, DC Sensors

R_{ESR} = Energy capacitor's ESR

R_{CO} = Crossover Switch Resistance

$$I_{SYS} = C * \left(\frac{V_0 - V_1}{T_1 - T_0} \right)$$

GAPGPS03005

$$V_1 = V_0 - \frac{I_{SYS} * (T_1 - T_0)}{C}$$

GAPGPS03006

Equation 1

$$V1 = V_{ER} - I_{SYS} * (R_{ESR} + R_{CO}) - \frac{I_{SYS} * T_1}{C}$$

GAPGPS03007

Deployment begins at T1, thus increasing energy reserve current and effects due to ESR

Equation 2 $V2 = V1 - I_{DEPLOY} \bullet R_{ESR}$

Where I_{DEPLOY} = total deployment current controlled by L96xx

Substituting (1) into (2):

Equation 3

$$V2 = V_{ER} - I_{SYS} * (R_{ESR} + R_{CO}) - \frac{I_{SYS} * T_1}{C} - I_{DEPLOY} * R_{ESR}$$

GAPGPS03008

During deployment phase, reserve voltage behavior is characterized as:

$$I_{SYS} + I_{DEPLOY} = C * \frac{V2 - V3}{T2 - T1}$$

GAPGPS03009

Equation 4

$$V3 = V2 - \frac{T_{DEPLOY} * (I_{SYS} + I_{DEPLOY})}{C}$$

GAPGPS03010

Substituting (3) into (4):

$$V3 = (V_{ER} - I_{SYS} * (R_{ESR} + R_{CO}) - \frac{I_{SYS} * T_1}{C} - I_{DEPLOY} * R_{ESR}) - \frac{T_{DEPLOY} * (I_{SYS} + I_{DEPLOY})}{C}$$

GAPGPS03011

Once deployment is complete, the airbag module enters its final state, shutdown. Reserve current is reduced causing reserve voltage to increase due to a minor loss in capacitor CER and Cross over switch. The change in reserve voltage is calculated as:

Equation 5

$$V4 = V3 + I_{DEPLOY} * R_{ESR} + (I_{DCSENSOR} + I_{RSU}) * (R_{ESR} + R_{CO})$$

GAPGPS03012

Equation 6

$$V4 = (V_{ER} - I_{SYS} * (R_{ESR} + R_{CO}) - \frac{(I_{SYS} * T_1 + I_{SYS} * T_{DEPLOY} + I_{DEPLOY} * T_{DEPLOY})}{C} +$$
$$- I_{DEPLOY} * R_{ESR} + I_{DEPLOY} * R_{ESR} + (I_{DCSENSOR} + I_{RSU}) * (R_{ESR} + R_{CO})$$

GAPGPS03013

In the equation (5) above, the system disables current to all deployment drivers, DC sensor and Remote Sensor Interfaces.

During shutdown phase, only Voltage regulator and device (L96xx) bias current are needed from reserve.

To complete energy reserve capacitor estimate, the analysis must assume a final reserve voltage requirement. In the study, energy reserve must be higher than VDDx dropout voltage where VDDx is the supply of the microcontroller.

By assuming this requirement, the system is designed to operate for the desired reserve time.

Equation 7 $V5 > VDDx_{DROPOUT}$

Reserve voltage behavior is as follows:

Equation 8

$$I_{SHUTDOWN} = C * \frac{V4 - V5}{T3 - T2}$$

GAPGPS03014

Where

$I_{SHUTDOWN} = I_{VDD} + I_{BIAS}$

Re-arranging (8)

$I_{SHUTDOWN} \bullet (T_3 - T_2)\ C \bullet V4 - V5)$

Substituting (6) and (7):

$$I_{SHUTDOWN} * (T_3 - T_2) = C * ((V_{ER} - I_{SYS} * (R_{ESR} + R_{CO}) - \frac{(I_{SYS} * T_1 + I_{SYS} * T_{DEPLOY} + I_{DEPLOY} * T_{DEPLOY})}{C} +$$

$$- I_{DEPLOY} * R_{ESR} + I_{DEPLOY} * R_{ESR} + (I_{DCSENSOR} + I_{RSU}) * (R_{ESR} + R_{CO}) - VDDx_{DROPOUT})$$

GAPGPS03016

Simplify and arrange

$$C = \frac{I_{SHUTDOWN} * (T_3 - T_2) + (I_{SYS} * T_1 + I_{SYS} * T_{DEPLOY} + I_{DEPLOY} * T_{DEPLOY})}{V_{ER} - I_{SYS} * (R_{ESR} + R_{CO}) + (I_{DCSENSOR} + I_{RSU}) * (R_{ESR} + R_{CO}) - VDDx_{DROPOUT}}$$

GAPGPS03017

2.1.6 Safing switch MOSFET

Safing switch provides two functions, fail-safe strategy and reducing energy absorbed by the high side deployment driver.

As fail-safe strategy, common practice is to have two independent and independently controlled components within the deployment path. This is often achieved by using a safing switch controlled by the microcontroller with an independent deployment driver IC.

For energy reduction, the L9680 is capable of deploying at a maximum voltage of 25 V. This can be done using the safing switch to absorb the excessive power or by setting VER to 23 V.

It is possible to measure via ADC the VSF and SSxy voltage, paying attention to the fact that as VSF is turned ON (SPI command) the external safing FET turns ON too.

The schematic proposed here represents a possible solution to perform SAFING FET test, which requires to have previously charged the external reserve capacitor, CER.

The solution requires an external network, two commands from the microcontroller, UC_DEPEN and UC_DIAGEN and the SSxy voltage measurement via ADC.

Depending on the status of VSF (ON or OFF) and on the commands from the microcontroller, three cases can occur as below described.

Figure 11. Schematic example to perform SAFING FET test

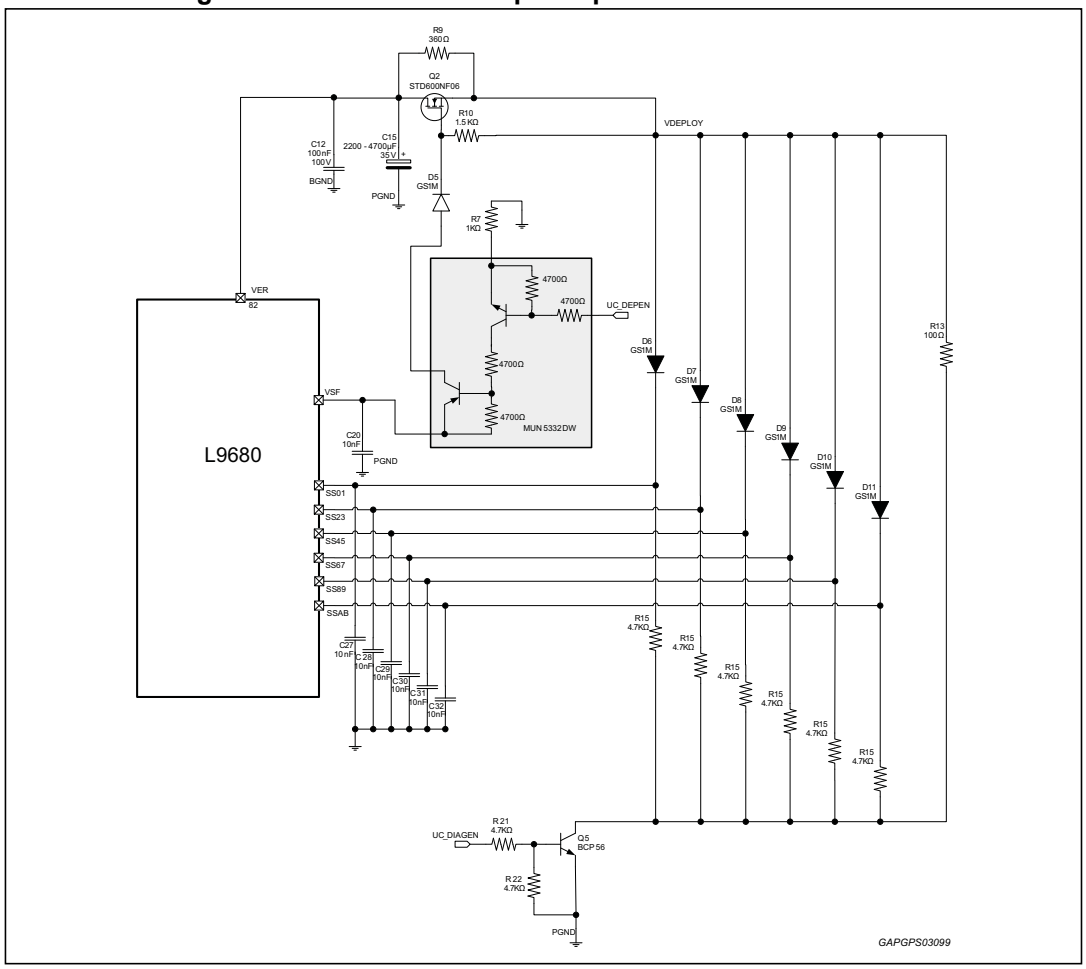


Figure 12. Range values for VSF safing fet test

VSF	UC_DEPEN	UC_DIAGEN	=SSxy pin voltage range = VDEPLOY – V diode	Normal operating
ON	1	1	=VSF-Vcesat-Vdiode-Vgs-Vdiode 10V 22V	FET reg and diagnostic enable
X	0	1	=(VER - Vcesat)*100/(100+360)+Vcesat - Vdiode 4V 7V	FET reg disable, Diagnostic disable
X	X	0	=VER-Vdiode 22.6V	FET reg and diagnostic disable

GAPGPS03100

In the **first case** of this *Figure 12*, external FET is working in voltage regulator mode (VSF ON, UC_DEPEN, UC_DIAGEN set);

voltage on SSxy pin is:

$$V_{SS_{xy}} = VSF - V_{CE_{SAT}} - V_{DIODE} - V_{GS} - V_{DIODE}$$

GAPGPS02928

The expected value read on ADC, depending on all the parameter variations, is in the range of 10 V to 22 V.

In the **second case** the low side command of the diagnostic is enabled and voltage on SSxy pin is:

$$V_{SS_{xy}} = 100 \cdot \frac{VER - V_{CE_{SAT}}}{100 + 360} + V_{CE_{SAT}} - V_{DIODE}$$

GAPGPS02929

The expected value read on ADC, depending on all the parameter variations, is in the range of 4V to 7V.

In the **third case** everything is disabled so the voltage on SSxy is expected to be close to VER:

$$V_{SS_{xy}} \approx VER - V_{DIODE}$$

GAPGPS02930

Any SSxy read via ADC, out of the expected range has to be considered as a faulty condition.

Note: Once UC_DIAGEN is active, capacitors on SSxy pins are discharged through 4.7 kΩ which requires about 1 ms to reach a steady state, so a proper time should elapse before running the ADC conversion.

Besides, in order to guarantee better safety, it is possible to read the voltage on VDEPLOY net through a voltage divider which is sensed by the ADC of the microcontroller.

The solution presented can be used when the ARMING is based on internal safing engine in order to guarantee redundancy; parallel path IC independent is needed to enable safing FET.

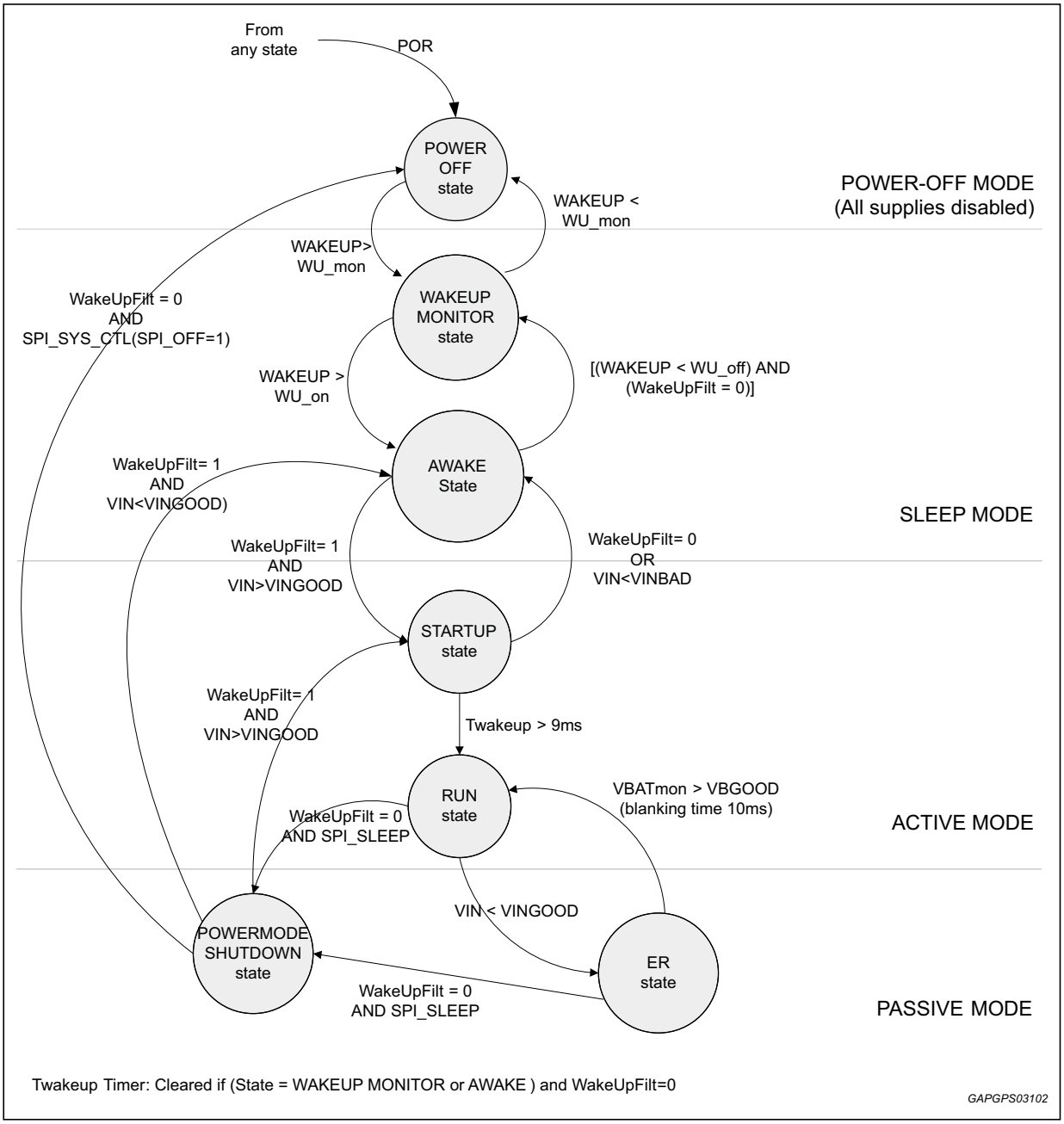
In case the ARMING algorithm is run by the microcontroller, the circuit which turns ON the safing FET can be removed (both MUN5332DW1 and D5 diode): VSF can be connected directly to the FET gate and UC_DEPEN can be used to drive FENH and FENL or their equivalent.

3 Power modes

The state diagram shown below defines L9680 power mode progression.

There are four specific power modes: Power-Off, Sleep, Active and Passive.

Figure 13. Operating modes



Power mode status can be read through SYS_STATE register.

Table 1. $04 POWER_CTL_STATE

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$04 POWER_CTL_STATE	-	R										OPER_CTL_STATE								POWER_CTL_STATE			b[2:0]] 000 = AWAKE 001 = START UP 010 = RUN 011 = ER 100 = POWER SHUT-DOWN 101, 110, 111 unused

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

3.1 Power off mode

In Power Off mode, the device is in reset, all power supplies are disabled.

As soon as the voltage on WAKEUP pin rises over WU_mon, the device starts the power-up process, passing in Sleep Mode.

It will turn back in POWER OFF mode from PASSIVE due to a specific SPI command combined with adequate WAKEUP condition or power on reset, POR, due to battery lost.

3.2 Sleep mode

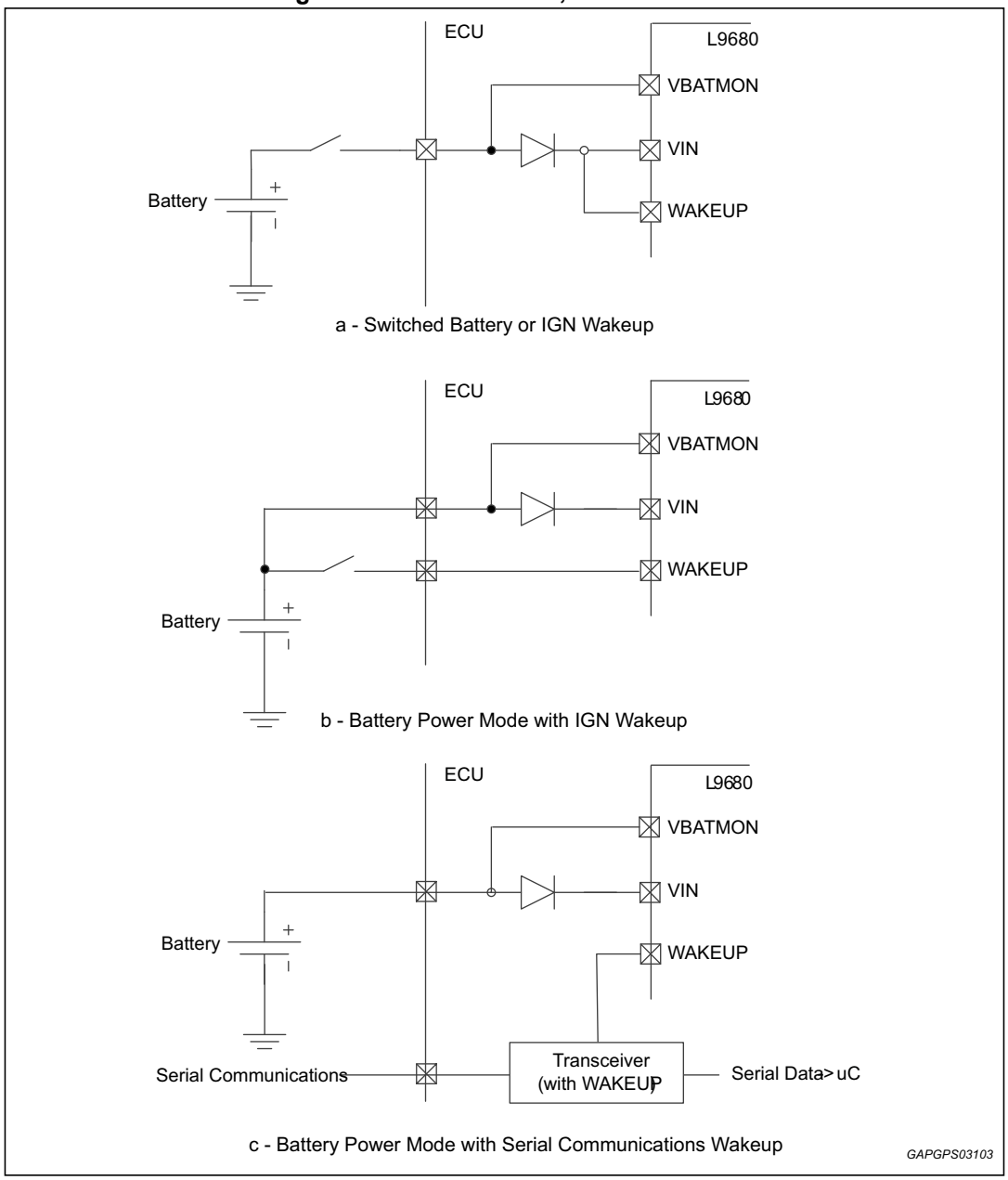
The IC enters in SLEEP MODE

1. during the device's turn on as WAKEUP>WU_mon
2. from ACTIVE MODE (before 9 ms filter time has expired on WAKEUP pin), as WAKEUP turns low or VIN<VINBAD
3. during its turn off from POWER SHUTDOWN mode as VIN<VINGOOD

Typical mechanizations for electrically configuring wake-up are shown in *Figure 14*.

These are typical mechanizations and do not show detailed descriptions for all connections and external components.

Figure 14. L9680 turn on, SLEEP MODE



Once passed in SLEEP MODE, the internal regulators, 3V3INT & CVDD (only CVDD is accessible on pin 93), are turned on and the IC is ready for the activation of all the other supplies.

Supposing the IC supplied, see as example *Figure 15*, through VIN and VBAT_MON (that is the supply monitor) the IC's turn on depends on WAKE_UP, that can be connected to the external power supply (cases "a" and "b") or can be driven by the microcontroller (case "c").

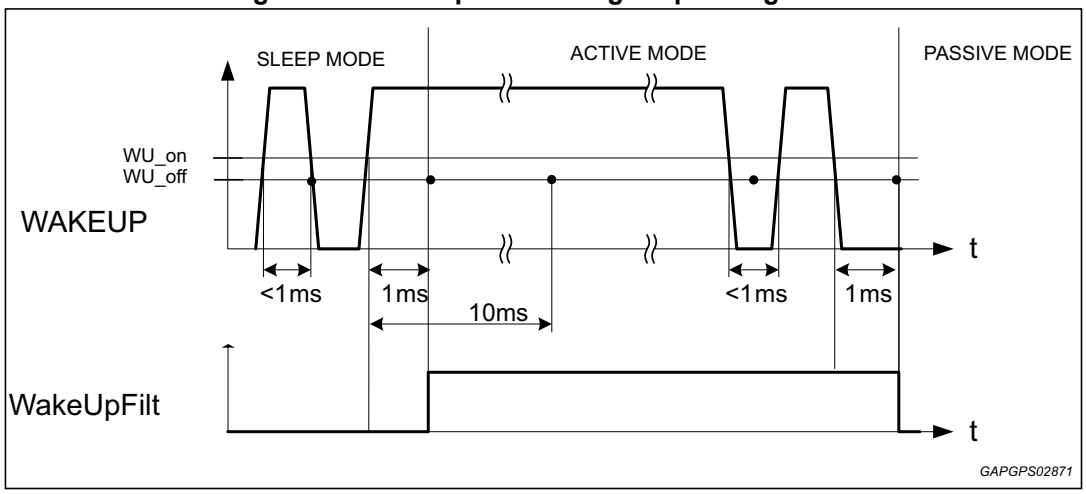
As WAKE_UP is high, it is necessary that its value should be constantly high for at least WakeUpFil time otherwise the turn on procedure restarts from the case of WAKE_UP pin low and the IC remains in SLEEP MODE. See *Figure 14* for WAKE_UP timing management.

Once:

(WakeUpFil=1) AND (VIN > VINGOOD)

the device moves in ACTIVE MODE.

Figure 15. Wake up filter timing & operating mode



For value of relevant parameters, refer to the device specification.

VBGOOD, VBBAD, VINGOOD & VINBAD thresholds are programmable using the SYS_CTL register. However, this cannot be done until the device reaches ACTIVE mode, therefore the device must be initially powered at the default values.

3.3 Active mode

As stated in the previous section, the device enters Active Mode once the following conditions are met:

- WakeUpFil=1; WAKEUP high for filter time
- VIN > VINGOOD

While in ACTIVE Mode and Twakeup timing has not expired, the device is in the STARTUP state. In STARTUP, it is possible to return back to SLEEP Mode should any of the conditions shown above fail.

Once WAKEUP is asserted for longer than Twakeup, the device starts normal operation, RUN State. In RUN state, all supplies are enabled and RESET is released upon reaching regulation. A possible topology of serial communications wake-up is shown in *Figure 16* and *17*.

In RUN state, the IC cannot directly return to the previous states, either the STARTUP or SLEEP mode. The device can only move to PASSIVE mode or remain in RUN state.

While in RUN state, normal ECU operation starts. The microcontroller can start programming, controlling and monitoring system functions through the L9680 device. Details on programming, controlling and monitoring system functions will be covered in the System Operating States section.

Figure 16. Normal power-up sequence with VCOREMON function disabled

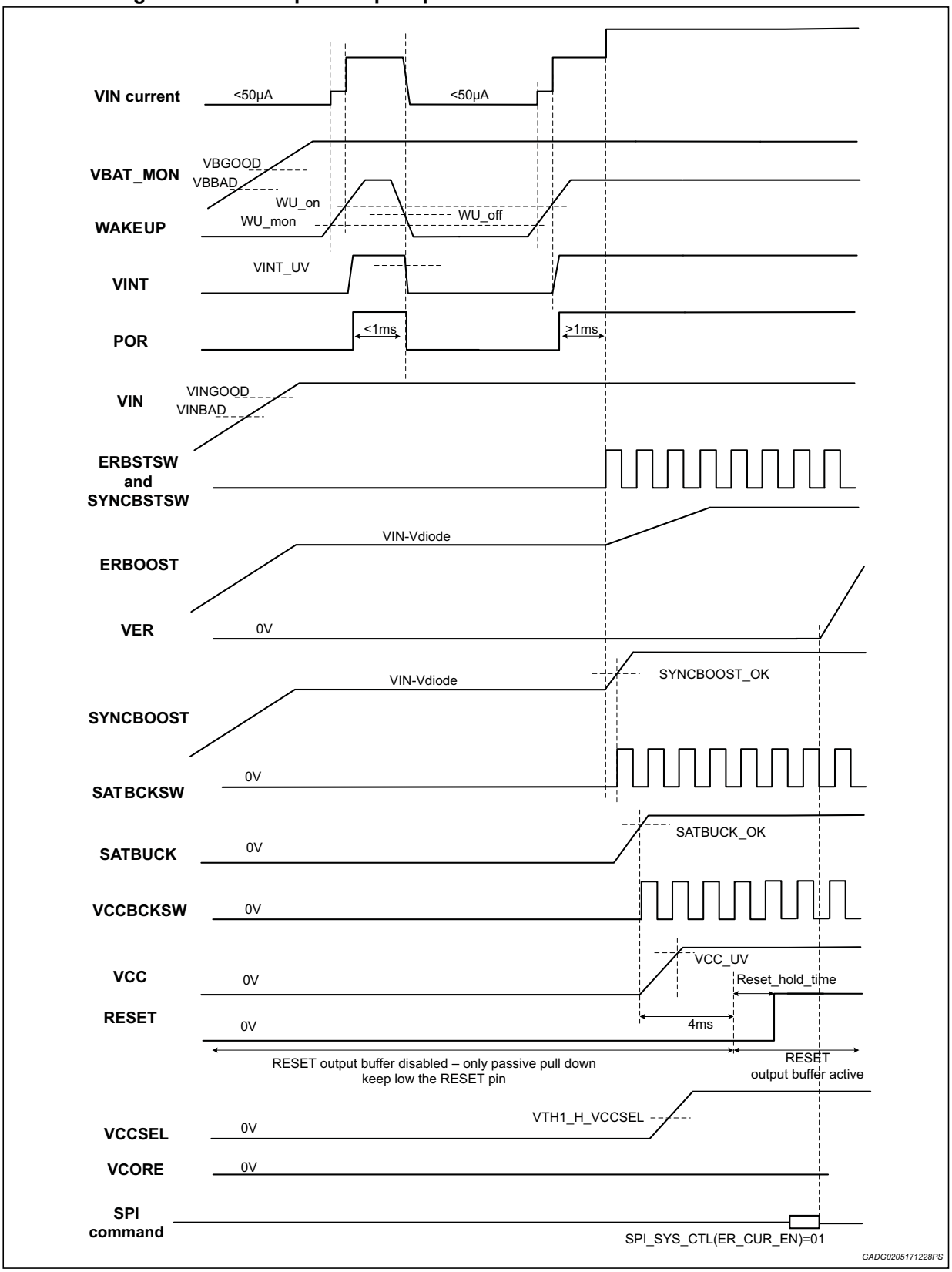
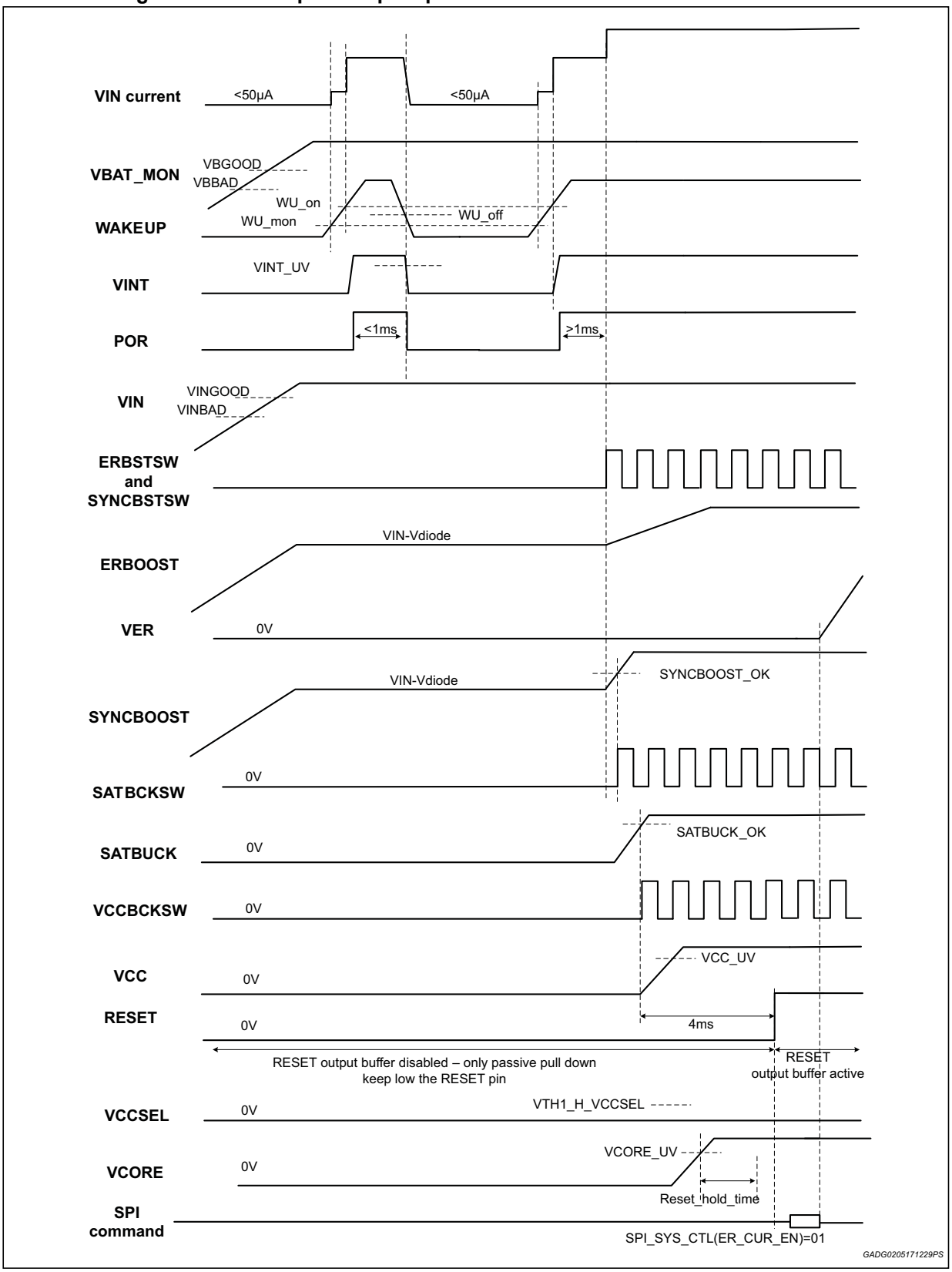


Figure 17. Normal power-up sequence with VCOREMON function enabled



Note: In case of VCCSEL open pin, an internal pull down current would force VCCSEL to ground and then the VCORE monitor will be enabled function.

If the VCORE voltage is low and the VCCSEL pin is higher than VTH1_L_VCCSEL, after the 4ms delay from power-up, a latched VCOREMON fault will cause RESET output to drive low, even though VCCSEL pin is high enough to satisfy the disabling of VCOREMON function. This occurs only once at power-up, and is then appropriately disabled. For this reason the RESET is released 500us (namely the reset_hold_time) after the 4ms delay from power-up.

Different RESET behavior depending on VCORE monitor status are reported in Figure 16 and 17.

Note: There are two thresholds for VIN GOOD and VINBAD, VINGOOD0 / VINBAD0 & VINGOOD1 / VINBAD1.

At power up, the default values are VINGOOD0 / VINBAD0.

Once WAKEUP has been correctly managed up (see *Section 3.2: Sleep mode*) as VIN>VINGOOD0 the IC moves in ACTIVE MODE.

Let's consider the situation where in STARTUP the microcontroller sets via SPI the thresholds VINGOOD1 / VINBAD1.

Let's also consider to be in a condition where

VINGOOD0 < VIN < VINBAD1

VIN < VINBAD1 in STARTUP state determines the IC to turn back in SLEEP mode (AWAKE state) and the SPI configurations to be reset at the default value.

So far, once back in SLEEP mode (AWAKE state) VIN > VINGOOD0 determines the IC to move again in STARTUP (active MODE). Microcontroller re-starts the programming and sets again VINGOOD1 / VINBAD1 threshold.

Then, being VIN < VINBAD1, the IC turns back in SLEEP mode (AWAKE state).

On the other hand, let's consider the case where the IC is already in RUN state the microcontroller sets via SPI the thresholds VINGOOD1 / VINBAD1.

Let's also consider to be in a condition where

VINGOOD0 < VIN < VINBAD1

VIN < VINBAD1 in RUN state determines the IC to move into autarchy mode (PASSIVE MODE, ER state).

3.4 Passive mode

PASSIVE mode can only be entered from ACTIVE mode, RUN state. There are two states within PASSIVE mode, ER state (autarchy) and SHUTDOWN state.

There are two scenarios for transitioning to PASSIVE mode:

- Scenario 1: WakeUpFlt = 0 & SPI_SLEEP command → SHUTDOWN state
- Scenario 2: VIN < VINGOOD → ER state

Scenario 1 corresponds to a driven shutdown via microcontroller which drives the WAKEUP pin low for at least 1 msec and sends a specific SPI sleep command:

Table 2. $03 SPI_SLEEP

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$03 SPI_SLEEP	-	W	-	-	-	-	$3C95															

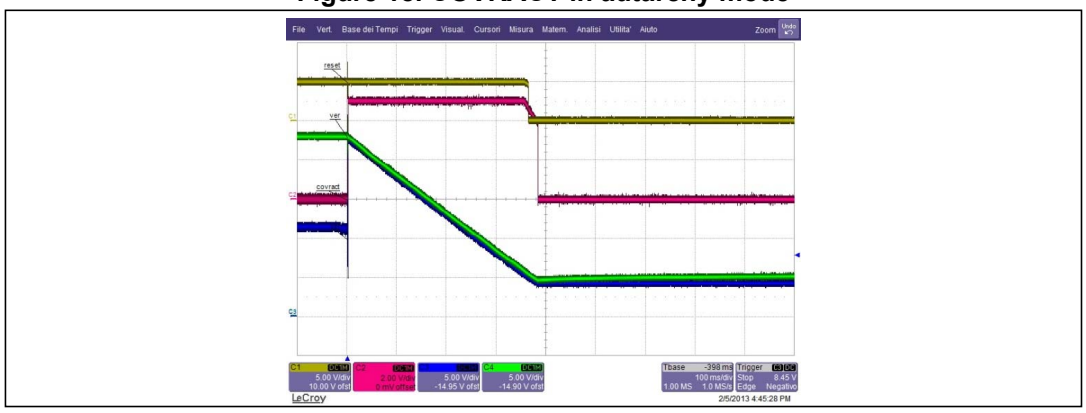
1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Scenario 2 represents the case low or loss of battery voltage - autarchy mode.

When VIN < VINGOOD the IC moves in PASSIVE MODE - ER state and COVRACT signal is asserted.

An example of COVRACT behavior in autarchy mode is reported in *Figure 18*, where ch1 = RESET, ch2 = COVRACT, ch3 = VIN, ch4 = VER.

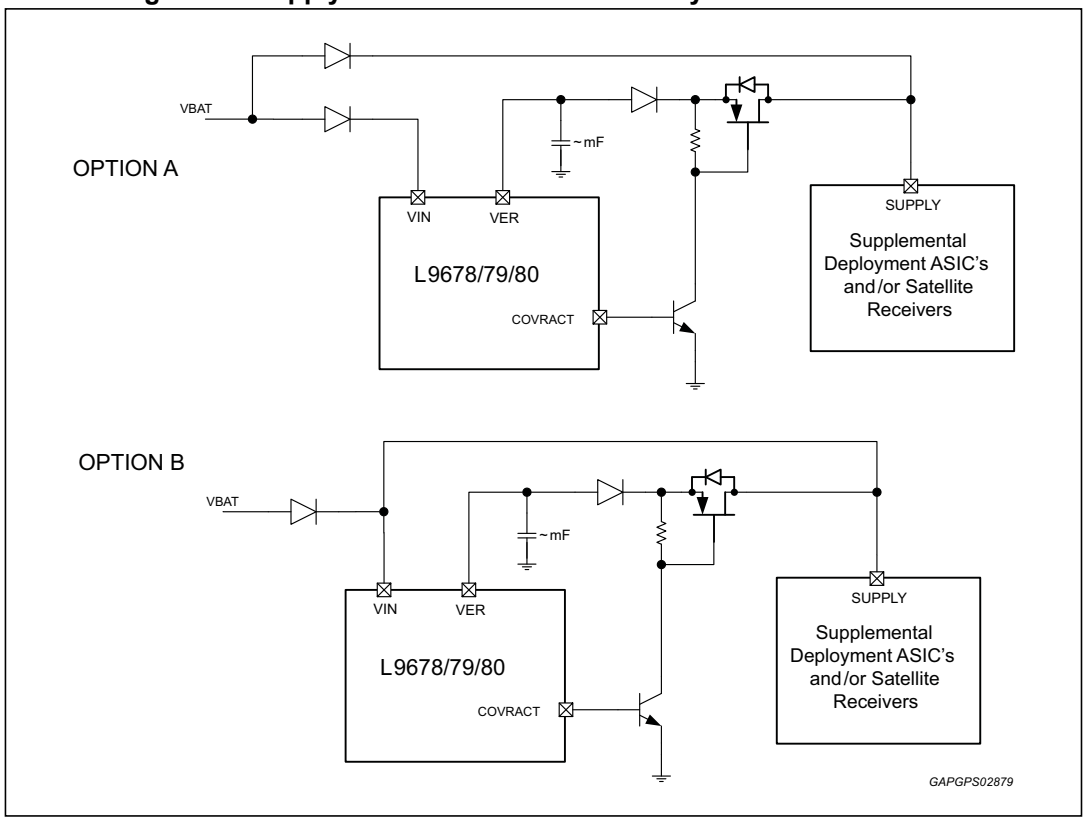
Figure 18. COVRACT in autarchy mode



COVRACT signal can be employed in case other ICs and/or satellite receivers have to be supplied during autarchy mode.

Being ER SWITCH current limited, a supplementary cross over switch may be needed and the simplified external circuitry to be used is shown in *Figure 19*.

Figure 19. Supply of other devices in autarchy mode via COVRACT



The difference between options A and B is just the current that flows through the diode (active or passive) placed between battery line (VBAT) and VIN pin.

If the current consumption of additional ASICs is too high to permit the IC to work at low battery level, the option A is mandatory, otherwise option B is suited too.

In autarchy mode the energy requirement, which may include the firing too, is taken from the external reserve capacitor, up to the capacitor is depleted and reset happens. In this condition, current from the external reserve capacitor is limited by "ER switch" (refer to the spec for the values), regardless VER value (22 V or 33 V). The external reserve capacitor has to be chosen taking into account the highest energy requirement.

If the capacitor on VER pin is not fully charged and VIN goes low, the external reserve capacitor could not be able to supply the system with the energy required.

In autarchy mode "ER charge" is disabled to decouple ERBOOST from VER.

ERBOOST in autarchy mode can remain enabled or can be disabled basing on the configuration of bit[12] KEEP_ERBST_ON in $01 SYS_CFG register:

Table 3. $01 SYS_CFG ERBOOST in ER state

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$01 SYS_CFG	I	W	-	-	-	-				KEEP_ ERBST _ON													12: 0=ERBOOST disabled in ER state 1= ERBOOST stay enabled in ER state

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Once in autarchy mode, the IC turns back in RUN state if VBATmon > VBGOOD condition is satisfied that implies the battery voltage recovered from a low voltage or dropout condition and hence normal operation can resume.

If the condition is not satisfied, shutdown can occur for the depletion of energy reserve (reset) or through a driven shutdown via SPI command. In this case the microcontroller completes all necessary "housekeeping" tasks, disables the WAKEUP input and sends the SPI_SLEEP command reaching SHUTDOWN MODE.

Once in SHUTDOWN MODE, the IC can:

1. turn back in ACTIVE MODE if VIN turns at its correct level and WAKEUP is correctly managed
2. move to POWER OFF if microcontroller sends SPI_OFF command and VIN is enough to keep the IC out of RESET, otherwise a POR happens.

3.5 Power mode functions status

Various functions are enabled and disabled within the power mode operating state machine. A summary of functions DISABLED in each operating mode is shown in *Table 4*.

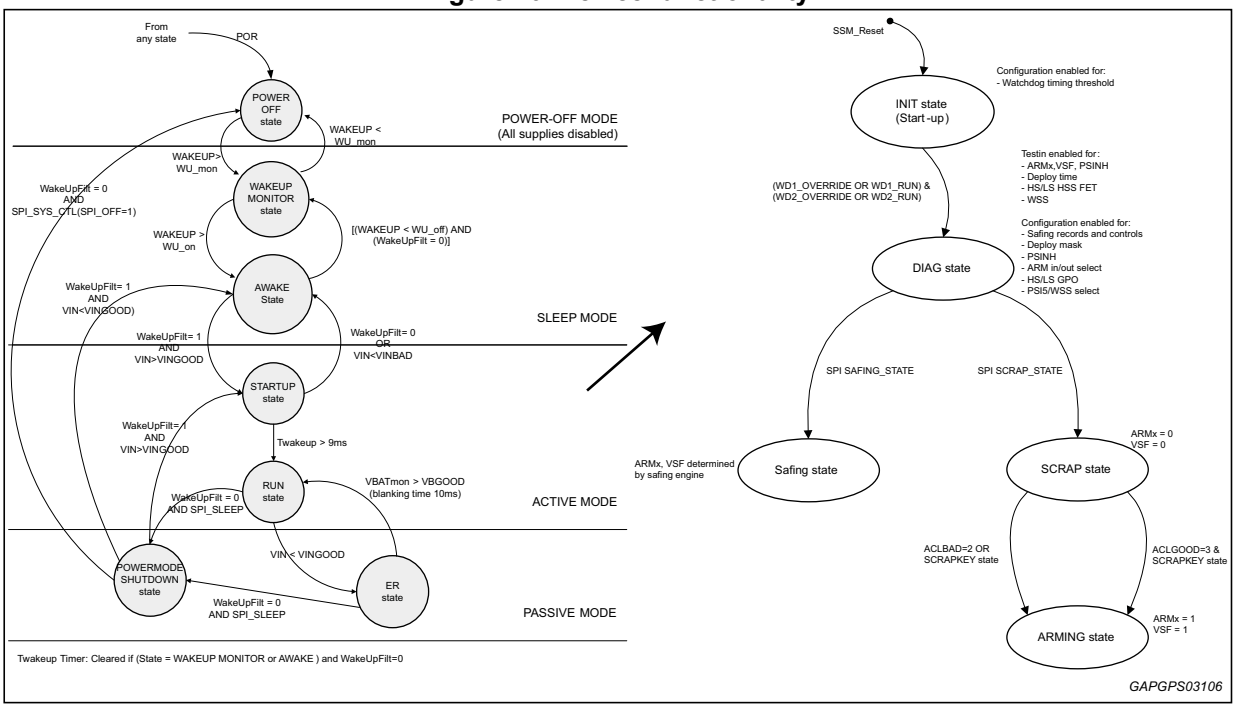
Table 4. Functions DISABLED by Power Mode

	Power Off	Wakeup Monitor	Awake	Startup	Run	Power mode Shutdown	ER
Wakeup Detector	√						
Internal Regulators	√	√					
ERBOOST Regulator	√	√	√			√	√
SYNCBOOST Regulator	√	√	√				
ER CAP Charge Current	√	√	√			√	√
ER switch	√	√	√	√	√		
SATBUCK Regulator	√	√	√				
VCC Regulator	√	√	√				
Deployment Drivers	√	√	√				
VSF Safing FET regulator	√	√	√				
Remote Sensor Interfaces	√	√	√				
Watchdog	√	√	√				
Diagnostics	√	√	√				
DC Sensor Interface	√	√	√				
GPO drivers	√	√	√				
Safing Logic	√	√	√				

4 Operation, control & diagnostics

Normal operation occurs when the device is in the ACTIVE power mode, particularly in the RUN state.

Figure 20. Device functionality



During power up phase, as RESET is de-asserted (IC in ACTIVE MODE), microcontroller starts working. In ACTIVE MODE are distinguishable five states, see *Figure 20*:

INITIALIZATION / DIAGNOSTIC / SAFING / SCRAP / ARMING

Each IC' state is readable via SPI:

Table 5. $04 POWER_CTL_STATE

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$04 POWER_CTL_STATE	-	R										OPER_CTL_STATE								POWER_CTL_STATE			b[10:8] 000 = INIT 001 = DIAG 010 = SAFING 011 = SCRAP 100 = ARMING 101, 110, 111 unused

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

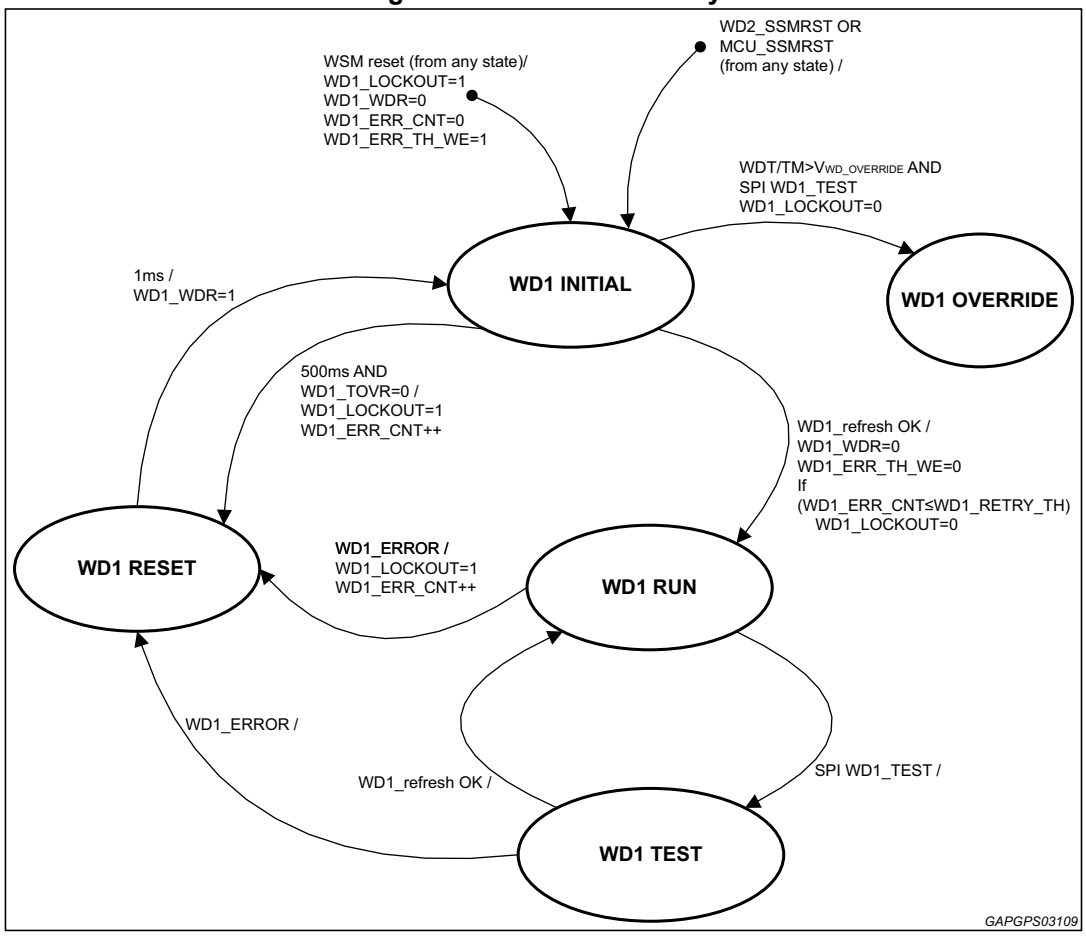
4.1 WDOG

In INITIALIZATION state, several configuration parameters are managed, including watchdog settings.

The IC is equipped with two kinds of watchdog, time window (WD1) and algorithmic seed/key (WD2).

4.1.1 Temporal watchdog initial - INIT STATE

Figure 21. WD1 functionality



Watchdog status is readable via SPI:

Table 6. $2C WD_STATE

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$2C WD_STATE	-	R						WD1_ERR_CNT				WD1_STATE			WD2_ERR_CNT				WD2_STATE				b[10:8]	000 = INITIAL 001 = RUN 010 = TEST 011 = RESET 100 = OVERRIDE
																							b[3:0]	0000 = INITIAL 0001 = OVERRIDE 0010 = INITSEED 0011 = RUN 0100 = TEST 0101= QUAL 0110=LOCK 0111= STOPPING 1000=STOP 1001=RESET

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Watchdog faults are readable via SPI in $00 FLTSR register:

Table 7. $00 FAULT STATE

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$00 FLTSR	-	R					WD2 retry cnt [3:0]					WD2_LO	WD2_TM	WD2_WDR	WD1_LO	WD1_TM	WD1_WDR						b[10:8] 000 = INIT 001 = DIAG 010 = SAFING 011 = SCRAP 100 = ARMING 101, 110, 111 unused

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

As WSM is released, watchdog is in its initial state, WD1 INITIAL.

In this phase it is possible to define all WD1 parameters, through $2A WDTCR and $28 WD_RETRY_CONF; as soon as the IC leaves WD1_INITIAL and reaches DIAGNOSTIC state (that means WD1, WD2 run or OVERRIDE, see *Figure 21*), WD1 parameters can't be modified any more.

Table 8. Watchdog WD1

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$2A WDTCR	I	W	-	-	-	-	X	WD1_MODE	WDTMIN[6:0]							WDTDELTA[6:0]							14:	0= fast WD1 8us resolute, 2ms max 1== slow WD1 64µs resolute, 16.3ms
																							13:7	WD1 min time window. $32 = 400µs; in fast mode, see WD1_MODE
																							6:0	WDTMAX=WDTMIN +WDTDELTA; $19 = 200µs in fast mode, see WD1_MODE
$28 WD_RETRY_CONF	I	W	-	-	-	-			WD2_ERR_TH			WD2_RETRY_TH								WD1_RETRY_TH			2:0	number of permitted error before latching WD1_LOCKOUT=1
$01 SYS_CFG	I	W	-	-	-	-																WD1_TOVR	WD1_TOVR: 0= 500ms timeout active 1= 500ms timeout disabled	

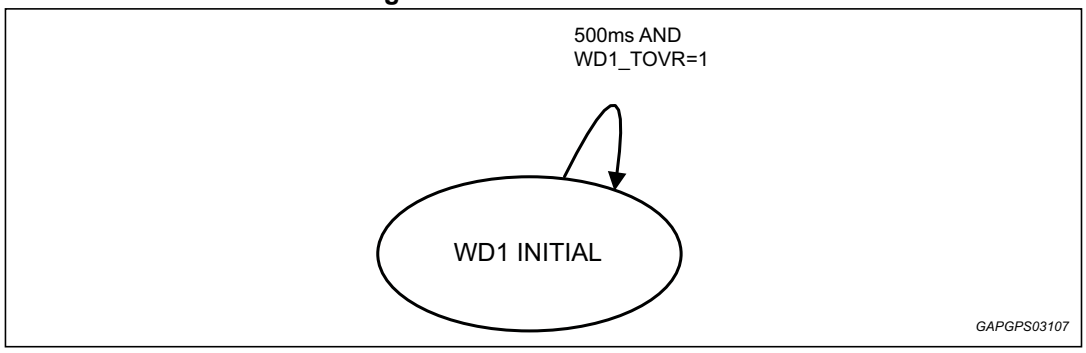
1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

In WD1 INITIAL state all arming signals are disabled to prevent deployment. As soon as entered in WD1 INITIAL, a 500 ms time window counter is started.

Before such a 500 ms time counter expires, all watchdog parameters should be fixed and the WD1_INITIAL should be left, a reset of the microcontroller will occur otherwise.

In order to disable the 500 ms counter and avoid reset generation, it is possible to set the bit WD1_TOVR (see *Table 8*).

Figure 22. WATCHDOG initial



Note: Disabling initial RESET toggling does not depend on the state of WDT/TM pin: SPI command is effective even if WDT/TM is grounded.

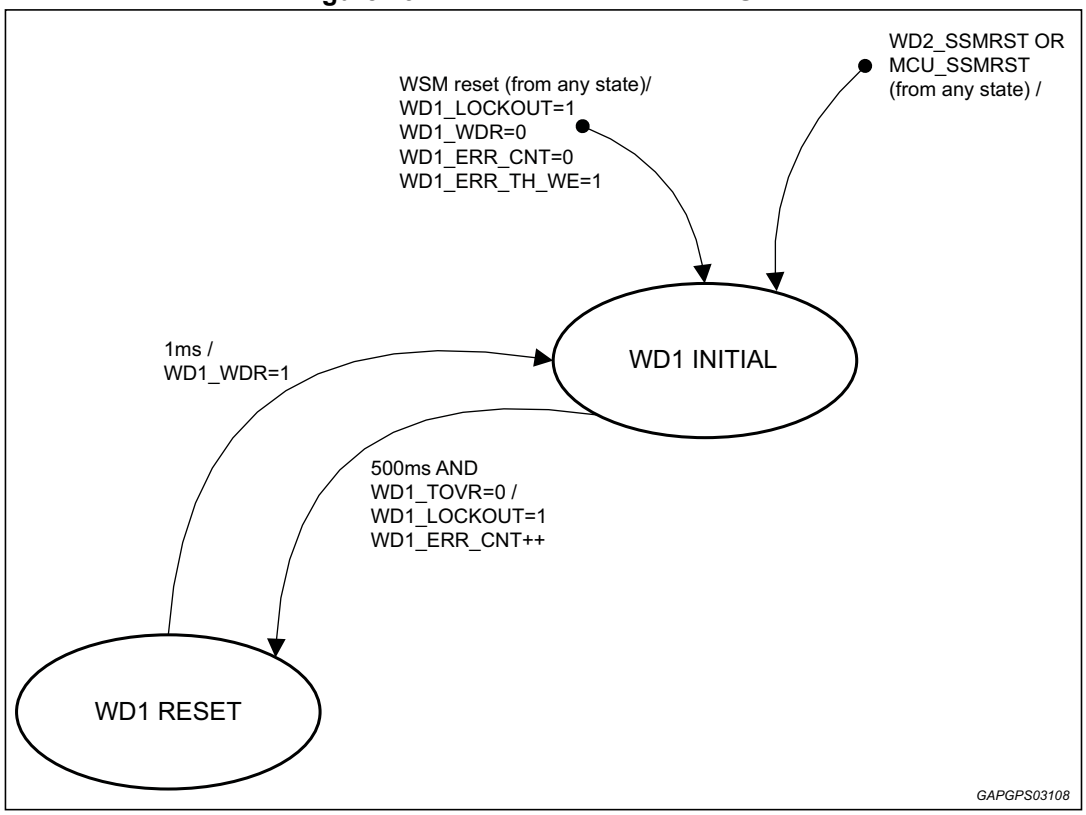
Each time the watchdog service routine fails, a counter is incremented; if the error counter reaches a programmable threshold (WD1_RETRY_TH) deployment is no more allowed (wd1_lockout = 1).

It is also possible by means of a combination of a high voltage level on pin WDT_DIS/TM and a proper SPI frame to completely disable watchdog and join DIAG state without any watchdog servicing.

In the following paragraphs a detailed description of the different watchdog phases is shown.

4.1.2 WD1 INITIAL- WD1 RESET

Figure 23. WD1 INITIAL ↔ WD1 RESET



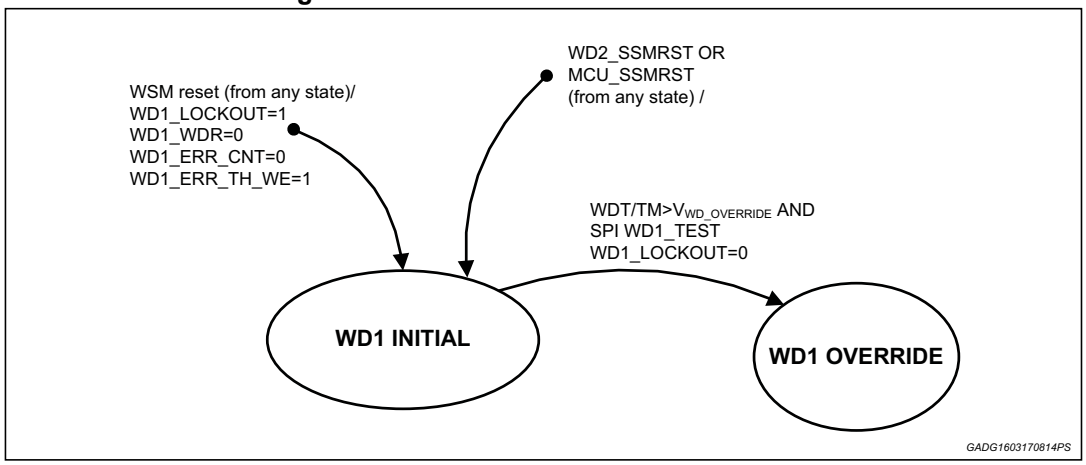
If neither WD1_TOVR is set in the first 500 ms nor WDOG service occurs, WD1_LOCKOUT bit is set and an error counter, WD1_ERR_CNT, is incremented. WD1_ERR_CNT value is readable via SPI, see *Table 6*.

WATCHDOG block moves in reset, WD1 RESET.

When WD1_LOCKOUT is set, all arming signals are disabled and deployment is inhibited.

4.1.3 WD1 INITIAL - WD1 OVERRIDE

Figure 24. WD1 INITIAL ↔ WD1 OVERRIDE



The watchdog transition from WD1_INITIAL state to WD1_OVERRIDE state corresponds to the transition from INIT to DIAGNOSTIC mode of the IC, see *Figure 20*.

In order to enter this state, the pin WDT/TM must be biased to the voltage VWD_OVERRIDE (refer to the spec for the value) and a proper SPI frame must be sent, accessing to the register $35 WD_TEST

Table 9. WD_TEST

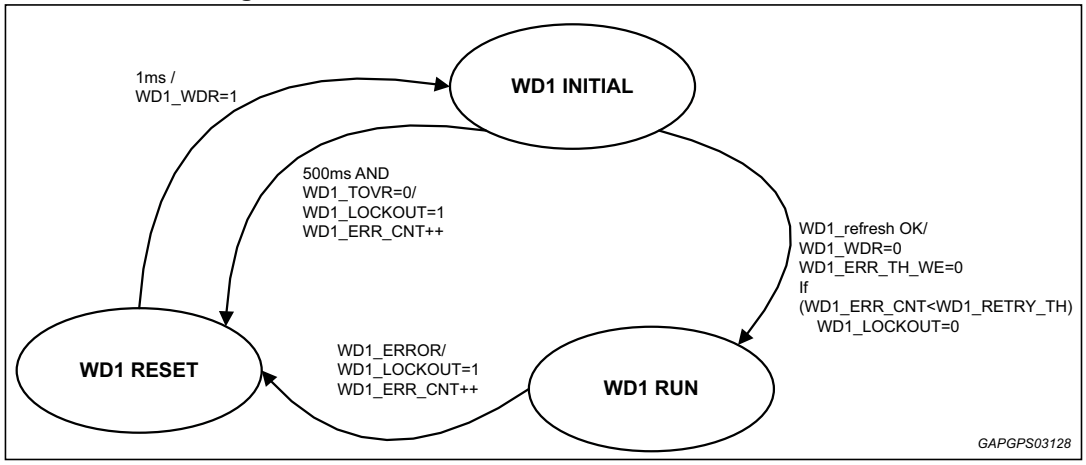
	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$35 WD_TEST	-	W	-	-	-	-	WD1_TEST								WD2_TEST								b[15:8] = $3C b[7:0] = $C3

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

4.1.4 WD1 INITIAL - WD1 RUN

When the watchdog routine is correctly serviced, the watchdog block enters in RUN MODE (*Figure 25*).

Figure 25. WD1 INITIAL ↔ WD1 RUN ↔ WD1 RESET



The counter WD1_ERR_CNT holds its value during the transition WD1_INITIAL WD1_RUN.

If WD1_ERR_CNT has reached its threshold (that is WD1_RETRY_TH defined in $28 WD_RETRY_CONF 28 see *Table 8*), WD1_LOCKOUT bit remains set; otherwise lockout is automatically removed.

The number of allowed error cycles before permanently asserting the lockout is defined via SPI into the WD1_RETRY_TH.

The watchdog transition from WD1_INITIAL state to WD1_RUN state corresponds to the transition from INIT to DIAGNOSTIC mode of the IC, see *Figure 20*.

Watchdog service is defined into register WD1T:

Table 10. $2B WD1T

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$2B WD1T	-	W	-	-	-	-															WD1CTL [1:0]		b[1:0] 00, 11 NOP 01 = A 10 = B
$2B WD1T	-	R	WD1_TIMER																		WD1CTL [1:0]		b[1:0] 00, 11 NOP / 01 = A / 10 = B WD1_TIMER b[16:8] WD1 timer value, step 8 µs or 64 µs in WD_RUN or WD1_TEST

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

WD1_TIMER counter controls the correct watchdog service. Its parameters are programmed in $2A WDTCR register in INIT IC state.

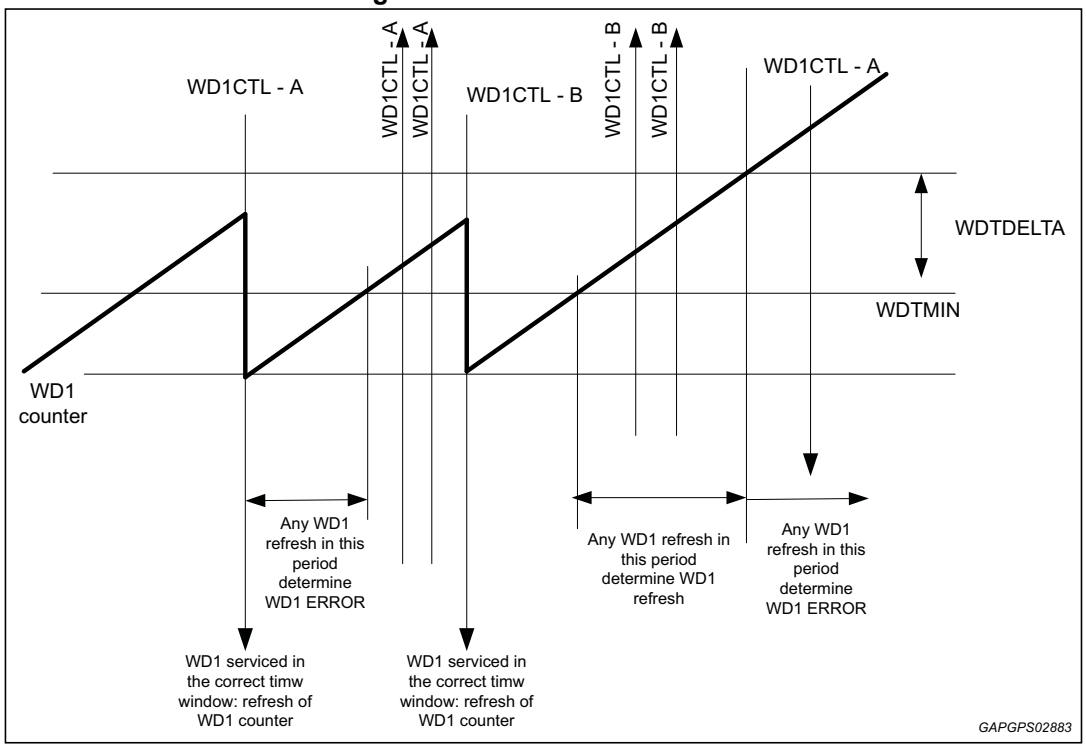
The counter WD1_TIMER, is reset every time the watchdog is service (WD1CTL is valid).

If the watchdog is serviced too early or too late with respect to the time window, the watchdog error counter, WD1_ERR_CNT, is incremented. WD1_LOCKOUT is set, watchdog status passes in WD1 RESET state for the management; the watchdog error WD1_WDR is set and readable via SPI in $00 FLTST register, see *Table 7*.

RESET pin is asserted with a time defined in the data sheet (TWD1_RST).

In *Figure 26* is sketched the WD1 service.

Figure 26. WATCHDOG service



If more than one WD1 with the same key value is received (for example A instead of B, see *Figure 26*), the WD1 counter is not refreshed until the correct key value is received in the defined time window (WDTMIN, WDTDELTA) and no error signals are asserted.

If more than one WD1 with the same key value (….A A A instead of …A and then B) is received no error signals are asserted even if the WD1 refresh command arrives before the counter has reached WDTMIN programmed value.

4.1.5 WD1 RUN - WD1 test

WD1_TEST state can be reached only from WD1_RUN state through the same SPI command used to move in WD1 OVERRIDE:

Table 11. $35 WD_TEST

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$35 WD_TEST	-	W	-	-	-	-	WD1_TEST								WD2_TEST								b[15:8] = $3C b[7:0] = $C3

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

This state generates a WD1_ERROR, without asserting WD1_LOCKOUT=1.

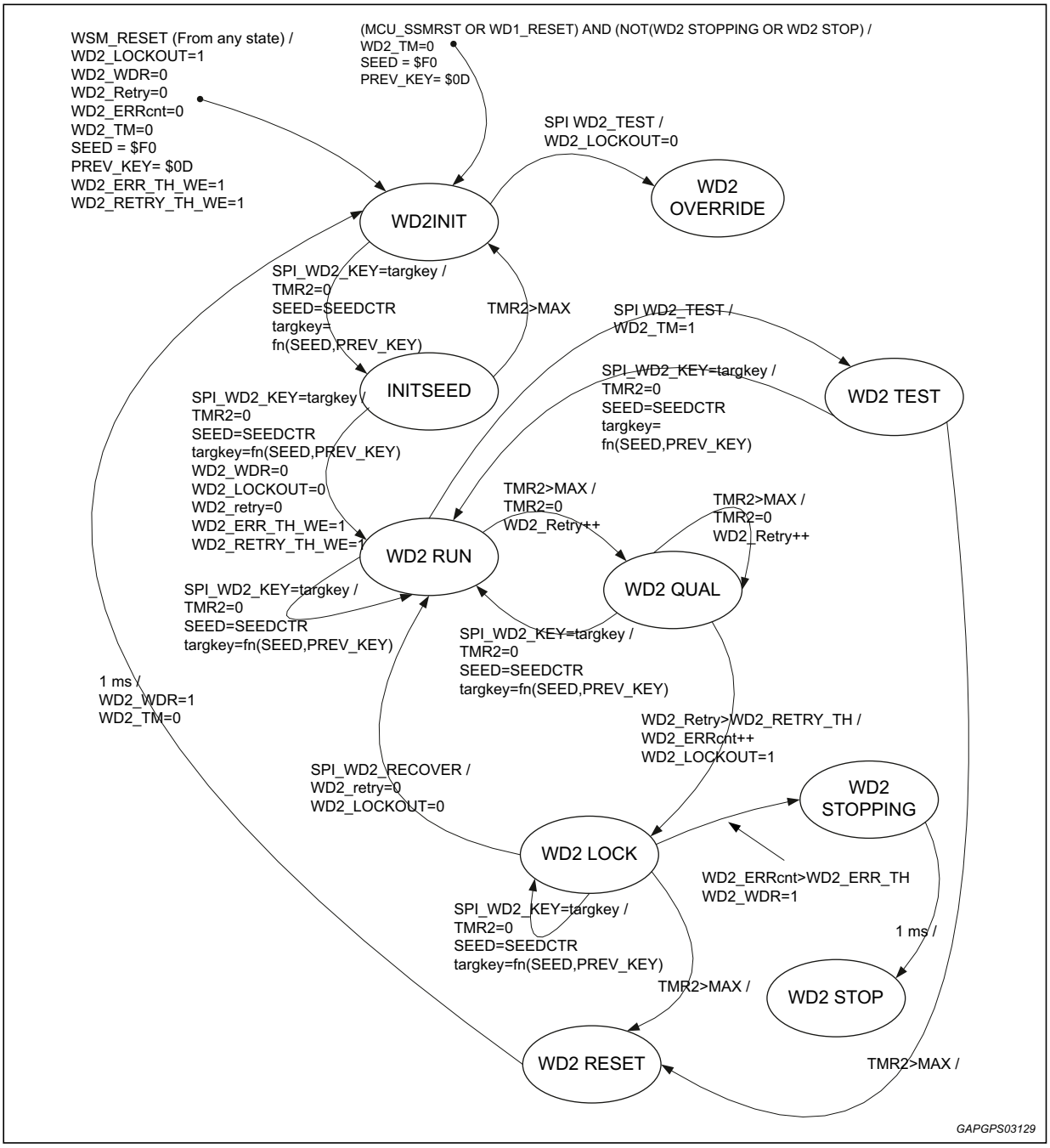
This is used to test the WD refresh. Typically it is implemented once per power up, even if there are no restrictions to access this mode in other moments.
In this state, deployments are not enabled.
Servicing WD1, watchdog turns back into WD1 RUN.

4.1.6 Algorithmic watchdog WD2

WD2 is an algorithmic watchdog managed starting from two default parameters, WD2_PREV_KEY and WD2_SEED, on a base time TMR2 and the number of error is counted in two counters, WD2_RETRY and WD2_ERR.

The overview of WD2 functionality, with signals and counters involved in WD2 management is summarized in *Figure 27*:

Figure 27. WD2 functionality



4.1.7 Algorithmic watchdog WD2 - WD2 INIT

Watchdog status is readable via SPI:

Table 12. $2C WD_STATE

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$2C WD_STATE	-	R						WD1_ERR_CNT				WD1_STATE			WD2_ERR_CNT				WD2_STATE				b[10:8]	000 = INITIAL 001 = RUN 010 = TEST 011 = RESET 100 = OVERRIDE
																							b[3:0]	0000 = INITIAL 0001 = OVERRIDE 0010 = INITSEED 0011 = RUN 0100 = TEST 0101= QUAL 0110=LOCK 0111= STOPPING 1000=STOP 1001=RESET

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Watchdog faults are readable via SPI in $00 FLTSR register:

Table 13. $00 FAULT STATE

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$00 FLTSR	-	R						WD2 retry cnt [3:0]				WD2_LO	WD2_TM	WD2_WDR	WD1_LO	WD1_TM	WD1_WDR						

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

WD2INIT is the default state (from start up or SSM reset, if WD2 is not in WD2 STOP state) that means a specific default value for two parameters, WD2_PREV_KEY[7:0] and WD2_SEED[7:0]. These parameters are available on SPI:

Table 14. $33 WD2_SEED

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$33 WD2_SEED	-	R					WD2_PREV_KEY								WD2_SEED								WD2_PREV_KEY [15:8] $0D default value
																							WD2_SEED [7:0] $F0 default value

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Based on WD2_PREV_KEY and WD2_SEED default values readable in $33 WD2_SEED register, the IC calculates the new key value as bitwise exclusive OR between the randomly generated seed and the key valid at the previous watchdog service step:

KEY = (SEED^WD2_PREV_KEY) + $01

This value is the expected value of WD2_KEY from the microcontroller written in $34 WD2_KEY register, indeed at the first step the microcontroller is supposed to read WD2_PREV_KEY and WD2_SEED default values and elaborates WD2_KEY.

In case the elaboration of the key in the microcontroller provides the same code as the internal calculation, WD2 block passes from WD2INIT to INITSEED.

4.1.8 Algorithmic watchdog WD2 - INITSEED

Once in INITSEED, TMR2 timer starts running. If TMR2 reaches its maximum value without any correct WD2_KEY, WD2 moves back to WD2INIT, otherwise a correct WD2_KEY determines the transition into WD2RUN.

4.1.9 Algorithmic watchdog WD2 - WD2RUN

In WD2RUN, if WD2KEY is continuously recalculated and serviced by the microcontroller, WD2 remains in RUN; provided that also the WD1 is overridden or correctly serviced, the IC consequently moves from INIT to DIAG state, see *Figure 20*.

4.1.10 Algorithmic watchdog WD2 - WD2QUAL

From WD2RUN, if WD2 is not serviced in TMR2 MAX, WD2 moves in WD2QUAL and counter WD2_RETRY is incremented.

If threshold of WD2_RETRY counter is not reached, as WD2 is correctly serviced (correct WD2_KEY), WD2 turns back to WD2RUN, recording the event of "WD2 not serviced inside TMR2 MAX time" in the counter WD2_RETRY.

If in WD2QUAL WD2 is NOT correctly serviced (correct WD2_KEY) in TMR2 MAX time, the counter WD2_RETRY is incremented.

In WD2QUAL the procedure of waiting for a correct WD2_KEY / increment of counter WD2_RETRY proceeds until the counter WD2_RETRY reaches its threshold, WD2_RETRY_TH. In correspondence of threshold reached event, WD2 moves into

WD2LOCK, WD2_LOCKOUT signal is asserted and another counter, WD2_ERRcnt, is incremented

WD2_ERRcnt holds the number of time that the first counter WD2_RETRY has reached its threshold, which is considered as WD2 not serviced correctly.

4.1.11 Algorithmic watchdog WD2 - WD2LOCK

In WD2LOCK, WD2_LOCKOUT signal is asserted, which means that any arming and deployment events are inhibited.

Once in WD2LOCK and the threshold of the second counter WD2_ERRcnt is not reached (WD2_ERR_TH),

1. if WD2 is correctly serviced, WD2 remains in WD2LOCK.
2. if WD2 is NOT correctly serviced, WD2 moves into WD2RESET
3. if the microcontroller sends a specific SPI word, WD2_RECOVER, WD2 moves back to WD2RUN, WD2_RETRY counter restarts and WD2LOCKOUT signal is reset.

Table 15. $32 WD2_RECOVER

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$32 WD2_RECOVER	-	W					-								$AA								

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

4.1.12 Algorithmic watchdog WD2 - WD2STOPPING - WD2STOP

Once in WD2LOCK and WD2_ERR counter has reached its threshold, WD2_ERR_TH, WD2 moves into WD2STOPPING and, with 1ms low pulse on RESET pin, into WD2STOP.

The only way to change WD2 state is a POWER ON RESET.

4.1.13 Algorithmic watchdog WD2 - WD2RESET - WD2INIT

Once in WD2LOCK and the threshold of the second counter WD2_ERR is not reached (WD2_ERR_TH) if WD2 is NOT correctly serviced, WD2 moves into WD2RESET.

From WD2RESET, WD2 turns into WD2INIT with 1ms low pulse on RESET pin, setting WD2_WDR and resetting WD2_TM.

4.1.14 Algorithmic watchdog WD2 - WD2OVERRIDE

From the first WD2 state, WD2INIT, WD2 can move in WD2OVERRIDE state through the same SPI command used to move WD1 in WD1OVERRIDE state:

Table 16. $35 WD_TEST

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$35 WD_TEST	-	W	-	-	-	-	WD1_TEST								WD2_TEST								b[15:8] = $3C b[7:0] = $C3

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

In correspondence WD2_LOCKOUT signal is reset.

Note that in order to override WD2 it is not required to have any high voltage on pin WDT/TM, as required for WD1.

4.1.15 Algorithmic watchdog WD2 - WD2TEST

From the WD2RUN state, WD2 can move in WD2TEST state through the same SPI command used to move WDx in WDxOVERRIDE state:

In correspondence WD2_TM signal is set, and it is generated a 1ms low pulse on RESET signal without setting WD2_LOCKOUT signal.

In WD2TEST as the WD2 is serviced, WD2 turns in WD2RUN state;

If in WD2TEST WD2 is NOT serviced in TMR2 time, WD2 moves into WD2RESET.

4.2 ER CAP features

The energy reserve capacitor connected to VER pin can be charged in an efficient way by means of a current generator.

Its capability is IER_CHARGE. The current generator is activated or deactivated by SPI command only while in ACTIVE mode. Out of ACTIVE mode, the current generator is switched off in order to decouple ERBOOST node voltage from VER pin.

In order to discharge the energy reserve capacitor, there is an internal current generator controlled via SPI, IER_DISCHARGE, high or low, when the IC is not in SLEEP mode.

Charge and discharge circuits are mutually exclusive.

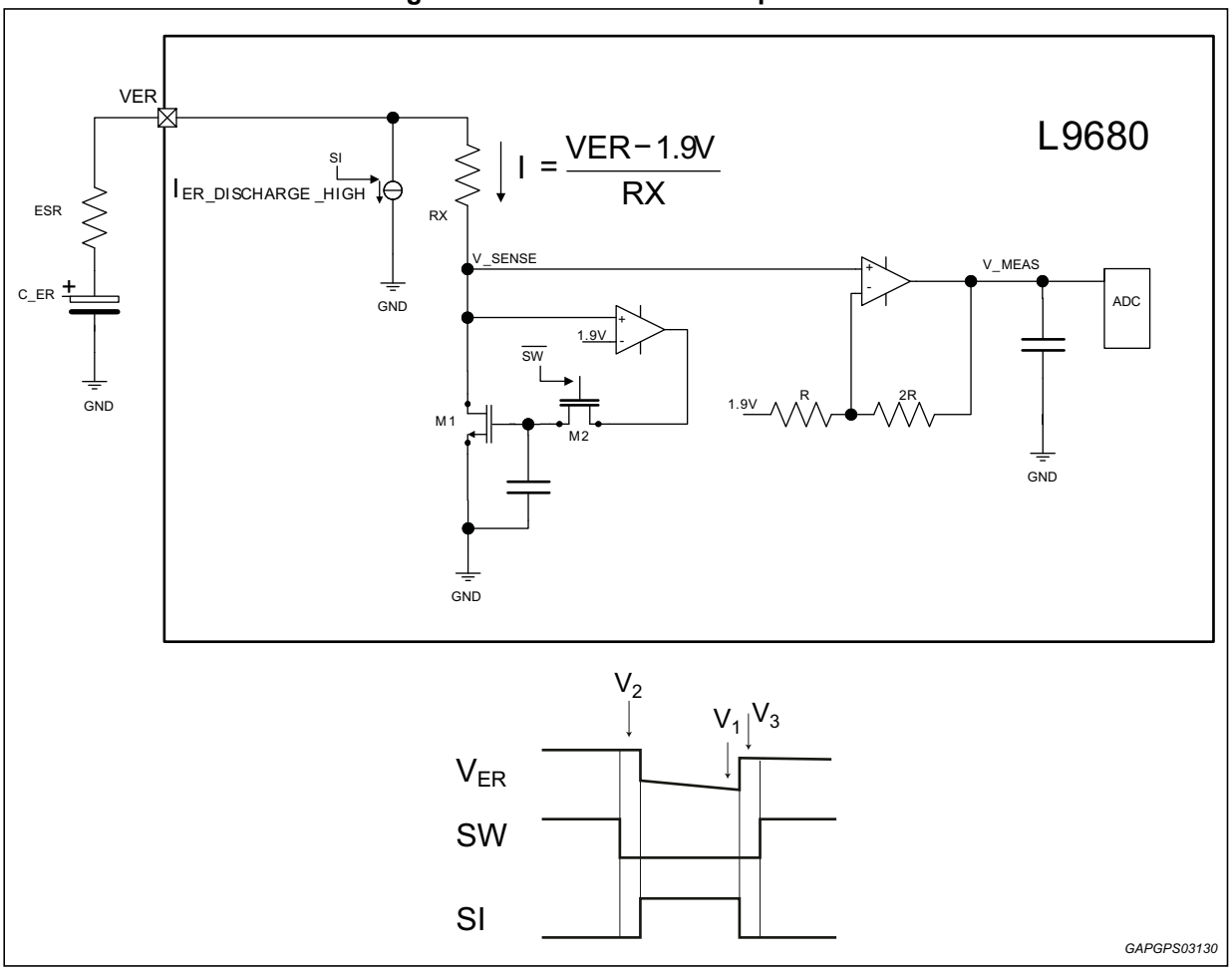
Diagnostic available on energy reserve capacitor regards the ESR capacitor measurement and the capacitor value measurement.

4.2.1 ESR measurement

Here below is shown the procedure to be used in order to measure value of ESR related to C_ER.

Refer to *Figure 28* for the measurement phases:

Figure 28. ESR measurement phases



Set up: the test can be performed in DIAGNOSTIC state, readable on register $04:

Table 17. $04 SYS_STATE

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$04 SYS_STATE	-	R	0	0	0	0	0	0	0	0	0	OPER_CTL_STATE			0	0	0	0	0	POWER_CTL_STATE			b[10:9]	001: DIAG
																							b[2:0]	010: RUN

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.

2. R = READ; W = WRITE.

C_ER has to be previously charged at VER, through ER_CHARGE ON:

Table 18. $02 SYS_CTL (1)

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$02 SYS_CTL	-	W	0	0	0	0	0	0	0	0	0	0	ER_BST_V	ER_CUR_EN									9:	0=23V 1=33V
																							8-7	01=ER_CHARGE ON

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Once C_ER has been charged, ER_CHARGE has to be switched OFF:

Table 19. $02 SYS_CTL (2)

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$02 SYS_CTL	-	W	0	0	0	0	0	0	0	0	0	0	ER_BST_V	ER_CUR_EN									9	0=23V 1=33V
																							8-7	00=ER_CHARGE OFF

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Test procedure

Test can be performed only in HIGH LEVEL DIAGNOSTIC mode, setting $38 LPDIAGREQ:

Table 20. $38 LPDIAGREQ

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$38 LPDIAGREQ	(I)	W	0	0	0	High /I ow level	0	0	0	0	0	X	X	X	HIGH_LEVEL_DIAG_SELECTOR			X	1	1	1	1	16	1= high level
																							b[7:5]	100= ER cap ESR meas
																							b[3:0]	1100÷1111 no Chanel selection

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.

2. R = READ
W = WRITE.

1. SW signal drives the MOS M2 ON and then M1 is driven ON too. Current I flows as indicated in the figure. Current generator IER_DISCHARGE_HIGH is OFF (through SI signal).
2. SW signal drives the correspondent MOS in OFF state; the capacitor between Gate and Source of M1 behaves as a memory and holds the voltage stored then M1 remains ON. Current generator IER_DISCHARGE_HIGH is still OFF, thorough SI signal. V2 voltage measurement of V_SENSE signal is taken (V2~1.9V+Voffset)
3. Current generator IER_DISCHARGE_HIGH is driven ON through SI signal for the whole time of the measurement. IER_DISCHARGE_HIGH current is sunk from C_ER determining a voltage drop across R_ESR. This voltage drop is the step down of VER signal that is reported on V_SENSE signal.
4. Current generator IER_DISCHARGE_HIGH ON discharges C_ER with a fixed current. At the end of the measurement, V1 voltage measurement of V_SENSE that takes care of the C_ER discharge is taken.
5. Current generator IER_DISCHARGE_HIGH is switched OFF and the last V_SENSE measurement (V3) is then taken.

The three voltage values, V1, V2 and V3, are automatically stored in this way:

DIAGCTRLA ADCREQ $15 ADCRES=V2
DIAGCTRLB ADCREQ $14 ADCRES=V1
DIAGCTRLC ADCREQ $16 ADCRES=V3

The three values (V1, V2 and V3) are converted from binary to decimal and then in voltage value considering that each bit corresponds to 2.5/1024 = 2.4 mV.

R_ESR is so calculated:

$$V_2 = V_1 + R \cdot I + (V_2 - V_3)$$

$$\rightarrow R = \frac{V_3 - V_1}{I}$$

GAPGPS03131

where I= $I_{ER\_DISCHARGE\_HIGH}$, reported in the spec.

Considering the gain of the second op amp before the ADC (see *Figure 28*,) and considering the residual offset introduced by the measurement circuit, the formula to be finally used becomes:

$$ESR_{ERCAP} = \frac{V_3 - V_1}{G_{ER\_ESR} \cdot I_{ER\_DISCHARGE\_HIGH}} + OFF_{ER\_ESR}$$

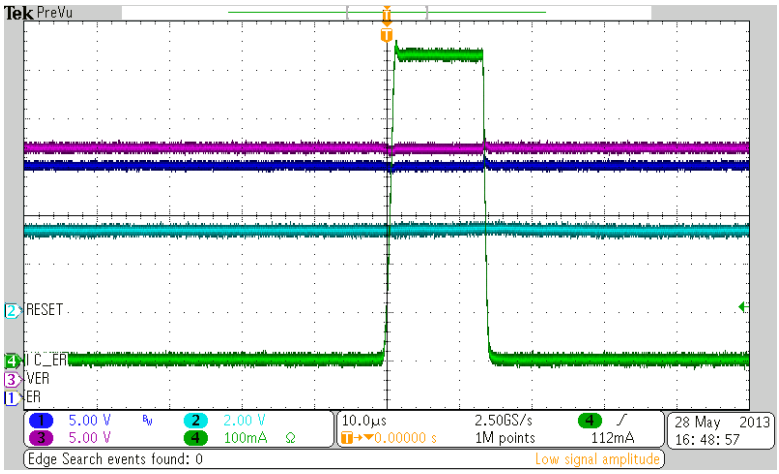
GAPGPS03132

Example

Figure 29 shows an example of a measurement performed.

The relevant signals are IER_DISCHARGE_HIGH (green) and VER (pink)

Figure 29. ESR measurement



blu=ER; pink=VER (23V left); light blu=RESET
Green= IER_DISCHARGE_HIGH during the measurement

GAPGPS03133

Table 21. ESR measurement

	(ADCREQ)16	(ADCRES)10	2.5/1024=2.4mV	
DAIGCTRL_3A	$15 (V2)	809	809 * 2.4 mV = 1.98 V	R =(2.02-1)V/(3*0.65A)=523mΩ
DAIGCTRL_3B	$14 (V1)	411	411 * 2.4 mV = 1.00 V	
DAIGCTRL_3C	$16 (V3)	829	829 * 2.4 mV = 2.02 V	

Supposing I= IER_DISCHARGE_HIGH = 640 mA (typ by spec)

4.3 ER CAP measurement

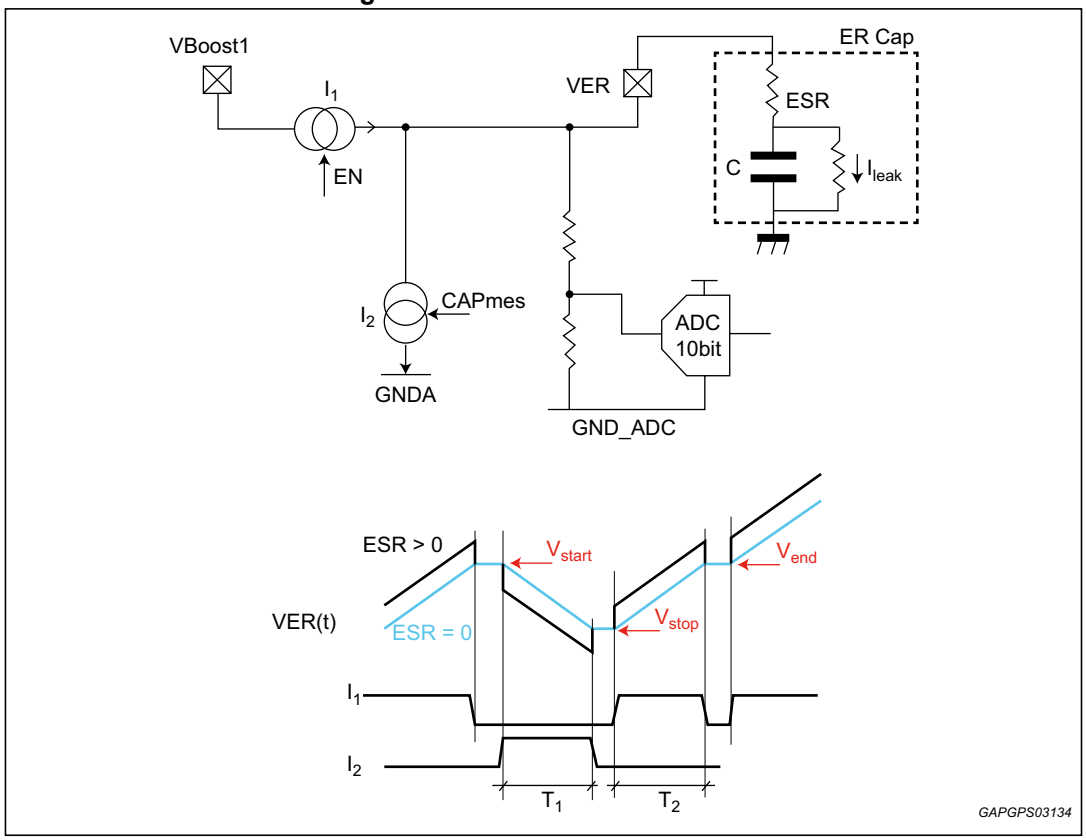
The ER CAP measurement is performed through three VER measurements via ADC.

ER_CAP measurement requires first ER_CAP charged, then discharged through a fixed current for a fixed chosen time (T1) and then re-charged for the same discharge time (T2 being T2=T1=T):

1. Definition of discharge (T1) and charge time (T2) with T1=T2=T
2. C_ER has been previously charged through I1 at roughly at half of ER_BOOST
3. switch off the ER_CHARGE via SPI
4. read the first VER value, named Vstart
5. switch on ER_DISCHARGE to discharge the capacitor through I2 in T1 time
6. switch off ER_DISCHARGE LOW
7. read the second VER value, named Vstop
8. switch on ER_CHARGE and re-charge the capacitor through I1 in T2 time
9. switch off ER_CHARGE
10. read the third VER value, named Vend

A sketch of the procedure is described in *Figure 30*.

Figure 30. ER CAP measurement



In order to avoid ESR error contribution, each VER measurement is performed when no current flows through ER cap (except the C_ER leakage ILEAK).

T_1 = predefined discharge time

T_2 = predefined charge time, same as charge time

$T_1 = T_2 = T$

I_1 = charge current

I_2 = discharge current, same as charge current

$I_1 = I_2 = I$

I_{LEAK} = leakage current

During C_ER discharge

$$Vstop - Vstart = -\frac{(I_2 + I_{LEAK}) * T}{C} \quad \text{a)}$$

GAPGPS03135

During C_ER charge

$$Vend - Vstopt = \frac{(I_1 - I_{LEAK}) * T}{C} \quad \text{b)}$$

GAPGPS03136

Once measured Vstart, Vstop and Vend, assuming the current which charges the capacitor as positive, equation (C) can be obtained by linear combination of equations (a) and (b)

(b) – (a):

$$Vend - Vstop - (Vstop - Vstart) = \frac{I_1 + I_{LEAK}}{C} T_1 + \frac{I_2 - I_{LEAK}}{C} T_2$$

$$C = \frac{2 \cdot I \cdot T}{Vstart + Vend - 2 \cdot Vstop}$$

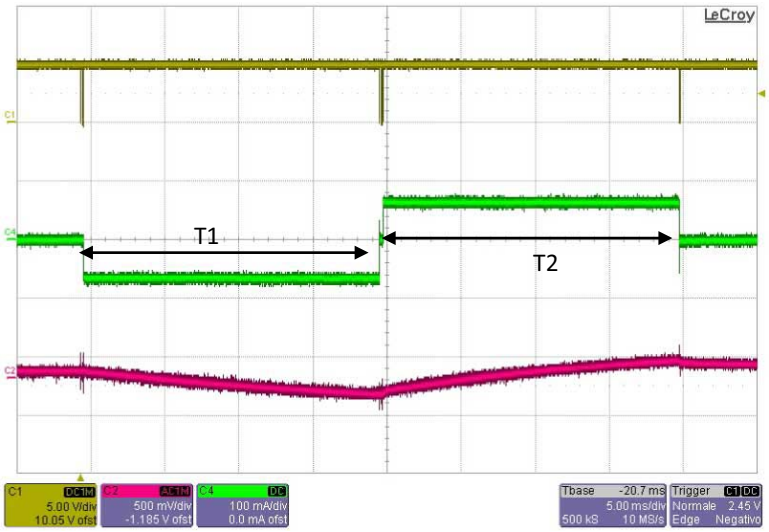
GAPGPS03137

I: refers to the datasheet for the value.

Example

T= 20 ms
VER~15 V

Figure 31. C_ER measurement



yellow = CS (trigger)ER; pink = VER (23V in AC);
Green = IER_DISCHARGE_LOW/ IER_CHARGE

GAPGPS03158

$3A	Vstart	434 → 15.894 V
$3B	Vstop	427 → 15.637 V
$3C	Vend	435 → 15.930 V

$$C = \frac{2 \cdot I \cdot T}{Vstart + Vend - 2 \cdot Vstop} = \frac{2 * 65 * 10^{-3} * 20 * 10^{-3}}{(434 + 435 - 2 * 427) * 15 * \frac{2.5}{1024}} = 4.7mF$$

GAPGPS03159

The reason why the measurement procedure has been implemented starting from ~15 V is to have the possibility to show a constant IER_DISCHARGE_LOW/ IER_CHARGE.

The user can decide the charge and discharge time based on the ER CAP used in application, in order to maximize the differential voltage and then improve the accuracy.

Anyway, a timeout on ER Discharge current has been implemented to prevent thermal issue, so the discharge time cannot be longer than 350 ms.

5 Arming

The device offers the possibility to select either an internal or external safing engine.

In case of internal safing engine, data from the satellites are elaborated based on a configurable algorithm in order to assert the internal arming signals; in case of external safing engine, the microcontroller is responsible for the arming algorithm.

5.1 Internal safing engine

SAFESEL bit[3], $01 SYS_CFG:

0 = internal safe engine

In case of internal safing engine, the assignment of loops matrix (LOOPx) to ARM signal is defined by LOOP_MATRIX_ARMx registers:

$6E LOOP_MATRIX_ARM1

$6F LOOP_MATRIX_ARM2

$70 LOOP_MATRIX_ARM3

$71 LOOP_MATRIX_ARM4

Table 22. $6E LOOP_MATRIX_ARM1

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$6E LOOP_MATRIX_ARM1	D	W	0	0	0	0	X	X	X	X	ARM1_LB	ARM1_LA	ARM1_L9	ARM1_L8	ARM1_L7	ARM1_L6	ARM1_L5	ARM1_L4	ARM1_L3	ARM1_L2	ARM1_L1	ARM1_L0	0	ARMx signal not associated with LOOPy
																							1	ARMx signal associated with LOOPy
																							Reset SSM_RESET	

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Once the sensor data has been captured, it is elaborated by the internal safing algorithm.

Each safing record (or their elaboration based on combine function - COMB = 1) is compared with the correspondent threshold.

Thresholds are always

- user defined
- programmed in absolute value (unsigned, no matter if applied to a single or a combination of safing record data)
- 16 bit length maximum

The safing records data or their combination is compared with the SAF_TH.

$DF SAF_THRESHOLD_1 $EE SAF_ THRESHOLD_16

Table 23. $DF SAF_THRESHOLD_1

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$DF SAF_THRESHOLD_1	D	W	x	x	x	x	SAF_THRESHOLD_1																cleared by SSM_RESET

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Inside the IC there are two dedicated counters (POS_COUNT, NEG_COUNT) with features defined in $66 SAF_ALGO_CONF:

Table 24. $66 SAF_ALGO_CONF

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$66 SAF_ALGO_CONF	D	W	0	0	0	0	NO DATA	X	ARMN_TH				ARMP_TH				SUB_VAL			ADD_VAL			cleared by SSM_RESET

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

If safing record data > +|SAF_TH| → positive counter, POS_COUNT, is incremented of ADD_VAL the counter is decremented of SUB_VAL otherwise;

If safing record data < -|SAF_TH| → negative counter, NEG_COUNT is incremented of ADD_VAL; the counter is decremented of SUB_VAL otherwise.

The value of the two counters, POS_COUNT and NEG_COUNT, are respectively compared with the threshold ARMP_TH and ARMN_TH to define ARMN and ARMP signals.

If POS_COUNT ≥ ARMP_TH → ARMP set

If NEG_COUNT ≥ ARMN_TH → ARMN set

Based on ARMSEL bit (register $EF-$FE, see *Section 5.3*) and on ARMP/ARMN result, the internal arming flags will be asserted or not.

In case the sample cycle time is elapsed (microcontroller has read the register bit $FF SAF_CC, typically every 500us but there is no implemented restrictions inside the IC about the value of this time), there are two possibilities.

1. sensor data have been received (CC_x=1), so the arming processing has been run successfully
2. no data has been received (CC_x=0)

In this last case (no data received) it is possible to configure how the IC behaves:

NO DATA = 1 no data received event is considered in a similar way of data below threshold (arming counters decremented);

NO DATA = 0 no data received events reset the arming counters (more severe alternative)

During a sampling cycle time, each safing record is able to elaborate only one data coming from the related sensor;

once a data has been elaborated by the safing record logic, the correspondent flag is set in the register $FF SAF_CC (CC_i bit=1). Should a new data come without the SAF_CC having been read from the microcontroller (that means CC_i still set), the data is ignored.

- CC (Compare Complete) - feedback from safing state machine indicating data received and processed.
  This internal flag is cleared by the microcontroller on each sensor sampling period.
  – CCi bit[i-1], i = 1÷16:
  0 = compare not completed for record i
  1 = compare completed for record i

Table 25. $FF SAF_CC

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$FF SAF_CC	-	R	0	0	0	0	CC_16	CC_15	CC_14	CC_13	CC_12	CC_11	CC_10	CC_9	CC_8	CC_7	CC_6	CC_5	CC_4	CC_3	CC_2	CC_1	cleared by SSM_RESET or SPI readout

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
   (I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
   W = WRITE.

SAF_CC register ($FF), together with ARM_state ($6A) ARMINT_x, b[5:2], see *Table 26* can be read both through Global and RS SPIs.

Table 26. $6A ARM STATE

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$6A ARM_STATE	-	R	0	0	0	0	0	0	0	0	0	0	PSIINHINT	PSINH_EXP_TIME	ACL_PIN_STATE	ACL_VALID	ARMINT_4	ARMINT_3	ARMINT_2	ARMINT_1	0	0	cleared by SSM_RESET or SPI readout

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
   (I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
   W = WRITE.

5.2 External safing engine

SAFESEL bit[3], $01 SYS_CFG:

1 = external safe engine

In case of external safing engine, the assignment of loops matrix (LOOPx) to ARM signal is defined as below described:

ARM1, ARM2, ARM3 are associated to the High Sides (FENH, active high) based on the configuration of the registers:

$6E LOOP_MATRIX_ARM1

$6F LOOP_MATRIX_ARM2

$70 LOOP_MATRIX_ARM3

Table 27. $6E LOOP_MATRIX_ARM1

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$6E LOOP_MATRIX_ARM1	D	W	0	0	0	0	X	X	X	X	ARM1_LB	ARM1_LA	ARM1_L9	ARM1_L8	ARM1_L7	ARM1_L6	ARM1_L5	ARM1_L4	ARM1_L3	ARM1_L2	ARM1_L1	ARM1_L0	0	ARMx signal not associated with LOOPy
																							1	ARMx signal associated with LOOPy
																							Reset SSM_RESET	

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

ARM4 is associated to all the Low Side enable (FENL, active low)

In this case the meaning of the register $71 LOOP_MATRIX_ARM4 is no more relevant being all the Low Side assigned to ARM4; in case of external safing engine the user should take care to distribute all the 12 loops available over the registers $6E-70.

5.3 Safing records

The IC is able to manage data from sensors either on board or remotely located, connected to the system via SPI (SPI_RS, SAF_CSx) or via RSUx interface.

In case of sensors' SPI connection, there is a regular communication between the main microcontroller and the sensors through multiple SPI messages. Since not all communications between sensors and the microcontroller contain data, it is important for the decoder to properly sort the communications and extract only the targeted data.

The solution involves defining specific masking functions, programmed by the user; the IC is able to manage either 16 or 32 bit length masks.

Sensors data are processed in order to engage the ARMING internal procedure through the safing engine.

The IC is equipped with the possibility to manage two kinds of mask.

1. 16 bit length masks:
   Applied to SAFING_RECORD_1÷SAFING_RECORD_13, defined with the correspondent SAF_REQ_MASK, SAF_REQ_TARGET, SAF_RESP_MASK, SAF_RESP_TARGET.
2. 32 bit length masks:
   Applied to SAFING_RECORD_14÷SAFING_RECORD_16, defined with the correspondent SAF_REQ_MASK, SAF_REQ_TARGET, SAF_RESP_MASK, SAF_RESP_TARGET. Each mask is composed by 2x16bit words, defined in the registers SAFXXX_pt1 and SAFXXX _pt2.

Despite the length of the masks, all the safing records are designed to manage 16 bit length max sensor data.

Inside the safing record, the sensor data can be located in a different position.

If the safing record is 16 bit length, data fill the safing record itself according to the procedure described in 6.4 SPI SENSOR DATA DECODING (16 bit SAFING RECORD MASK);

If the safing record is 32 bit length, data are still 16bit length, but data can be distributed over the first MSB 16 bit, the last LSB 16 bit or across them:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Acceleration data within the 16 MSB																/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	Acceleration data within the 16 LSB															
/	/	/	/	/	/	/	/	/	/	/	Acceleration					data across the 1st and 2nd 16 bit words											/	/	/	/	/

Each safing record has SPI accessible registers defined in the SPI command tables and summarized below:

- Request Mask and Request Target - to understand which sensor the microcontroller is addressing
- Response Mask and Response Target - to identify the sensor response
- Data Mask - to extract relevant sensor data from the response
- Safing Threshold - specific value that sets the comparator limit for successful arming

- Control, $EF÷$FE SAF_CONTROL_x
  - IF, In Frame - to indicate serial data response is "in frame". There are two types of potential serial data response, "in-frame" and "out of frame".
    IF bit[0]:

    0 = out of frame response

    1 = in frame response

    CS - to align safing record with a specific SPI CS. The device contains 5 SPI CS inputs for the safing function (CS_RS, SAF_CSx)
    CSx bit [2:0]:

    000, 110, 111 no selection for record x

    001= SAF_CS0

    010= SAF_CS1

    011= SAF_CS2

    100= SAF_CS3

    101= CS_RS
  - ARM - there are four internal arming signals, each active record is assigned or mapped to any arming signal. Several safing records can be mapped to a single arming output. ARMx outputs can be enabled also simultaneously.
    ARM4x bit[7]:

    0= safing record not assigned to ARM4INT

    1= safing record assigned to ARM4INT

    The same as the other 3 bits, ARM3x[6], ARM2x[5], ARM1x[4]
  - Dwell - Once an arming condition is detected, the IC remains armed for the specified dwell time.
    DWELLx bit[9:8]:

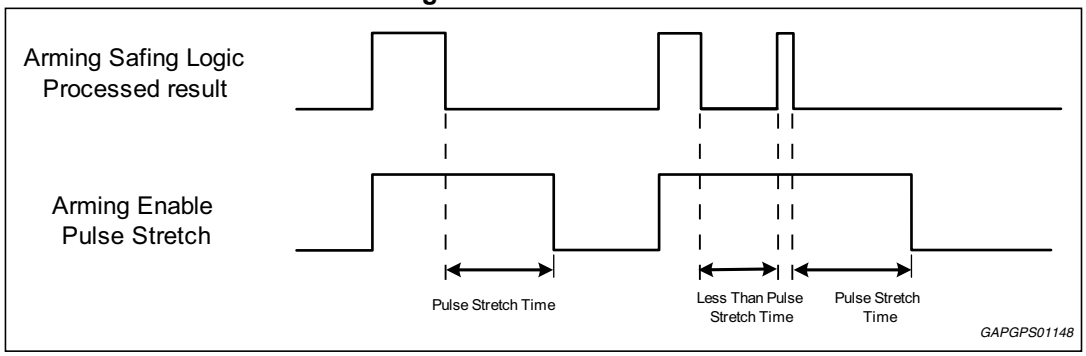
    00 = 2048 ms

    01 = 256 ms

    10 = 32 ms

    11 = 0 ms

Figure 32. DWELL time



For each safing record a specific Dwell value is programmed. In case multiple safing records with different Dwell values are assigned to a specific ARMx signal, the longer value is used

– Comb (Combined Data) - specific solution for dual axis high-g sensors specifically oriented off-axis (see 6.6).
COMBx bit[10]:

0 = combine function disabled

1 = combine function enabled

– Lim En (Limit Enable) - to enable PSI5 out-of-range control.
LIM Enx bit[11]:

0 = data range limit disabled

1 = data range limit enabled

– Lim Sel (Limit Select) - to select PSI5 out-of-range thresholds between 8-bit and 10-bit protocol.
LIM SELx bit[12]:

0 = 8 bit data range limit, |data|>120d not recognized as valid data

1 = 10 bit data range limit, |data|>480d not recognized as valid data

– SPIFLDSELx bit[13]: - only for 16bit masks safing records:

0 = sensor data are located in the first 16 bit of SPI MISO

1 = sensor data are located in the second 16 bit of SPI MISO

If the SPI message is shorter than 32 bit, SPIFLDSELx is don't care

- ARMSELx - to select positive or negative acceleration
ARMSELx bit[15, 14]:

00, 11 = ARMP or ARMN

01 = ARMP

10 = ARMN

Table 28. $EF÷FE SAF_CONTROL_x

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$EF÷$FE SAF_CONTROL_x	D	W	0	0	0	0	ARMSELx		SPIFLDSE	LIM_SELx	LIM_ENx	COMBx	DWELLx		ARM4x	ARM3x	ARM2x	ARM1x	CSx[2:0]			IFx	Updated by SSM_RESET or SPI write in DIAG state

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

The En (Record Enable) bit for any record is programmable as ON or OFF at any time and will enable/disable the record itself upon the following SATSYNC.

- En (Record Enable) - determines when a safing record is active or disabled
EN_SAFi bit[i-1], 1=1÷16:

0 = disabled

1 = enabled

Table 29. $7F SAF_ENABLE

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$EF÷$FE SAF_CONTROL_x	(I)	W	0	0	0	0	EN_SAF16	EN_SAF15	EN_SAF14	EN_SAF13	EN_SAF12	EN_SAF11	EN_SAF10	EN_SAF9	EN_SAF8	EN_SAF7	EN_SAF6	EN_SAF5	EN_SAF4	EN_SAF3	EN_SAF2	EN_SAF1	Updated by SSM_RESET or SPI write

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Safing record can only be evaluated on the first matching input packet. Any further data packet matches are ignored (i.e. once CC is set, record can't be processed until next SATSYNC)

If input packet matches multiple safing records, the safing engine will process all of them and treat them independently.

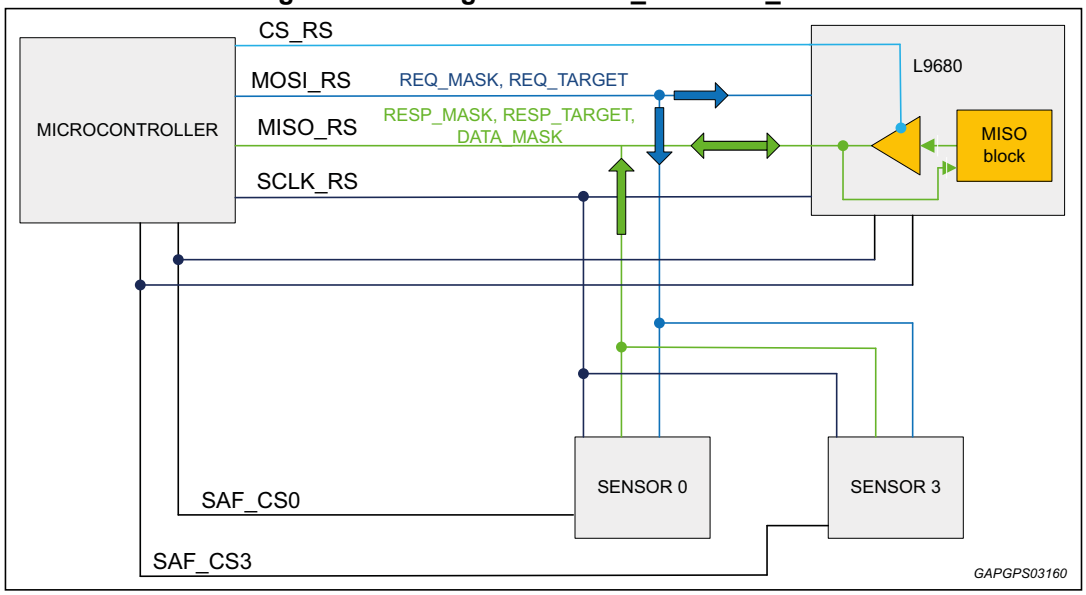
Safing Engine must not process sensor data in any state but in Safing state.

All safing records are cleared on SSM RESET.

Following the sketch below, see *Figure 33*, when one of SAF_CS0 ÷ SAF_CS3 is asserted, IC is able both to capture data on MOSI_RS and sniff traffic from MISO_RS line so the communication between microcontroller and sensor is monitored. In this case the IC does not apply any integrity check on SPI information sniffed (parity, CRC or whatever) but it simply verifies that a minimum of 16bit has been transmitted for each SPI frame.

The microcontroller has to guarantee that SAF_CSx, and CS_RS are not asserted simultaneously; should it happen, the SPI frame is ignored. The CS_G is not involved in this check.

Figure 33. SPI signal and SAF_CS0÷SAF_CS3



5.4 SPI sensor data decoding (16 bit Safing record mask)

Supposing, for example, the sensors are so organized:

b[14,13] = channel selection, x or y

b[9:0] = data sensor (10 bit data length)

b12 = parity bit

b[11,10] = type of data in sensor data field (acceleration, self-test, error)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO	0	CH1	CH0	P	ST1	ST0	ACCELERATION DATA									

If the microcontroller request is (MOSI):

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SI	0	0	1	X	1	0	X						1	0	1	0

b[14:13] = 01 this codification can be for example the sensor for x acceleration

b12 = don't care

b[11,10] = 10 this codification can indicate for example that in the data field are reported acceleration data

b[9:4] = don't care. These are the data the sensor sends. Microcontroller reads them, but doesn't write them

b[3:0] = address of the sensor, for example 1010 that is sensor n.3

5.4.1 Request Mask (MOSI line)

Setting SAF_REQ_MASK[15:0] bit in SAF_REQ_MASK_x, it is defined, inside the frame, the position of the bit to be considered in the data received. This is done setting bit in SAF_REQ_MASK_x located in the position to consider.

Extraction of data is done putting in AND (bit per bit) the SAF_REQ_MASK with the data received.

SAF_REQ_MASK is applied on messages recorded from MOSI pin; useful information is considered to be transmitted always on first 16 bit block.

$80 SAF_REQ_MASK_1 ÷ $8C SAF_REQ_MASK_13

Table 30. $80 SAF_REQ_MASK_1

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$80 SAF_REQ_MASK_1	D	W	x	x	x	x	SAF_REQ_MASK_x																cleared by SSM_RESET

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

In the example SAF_REQ_MASKx[15:0]:

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1	1	1	0	1	1	0						1	1	1	1

b[15:13], b[11:10], b[3:0] set

b12, b[9:4] correspond to the position of the bit not to be considered; they are left at 0.

5.4.2 Request Target (MOSI line)

Considering the bit selected through SAF_REQ_MASK, the value these bit have to assume is indicated REQUEST TARGET.

So:

in REQUEST MASK it is defined which bit position to look at;

The expected values of the bit, which are in the positions indicated by SAF_REQ_MASK, are defined in SAF_REQ_TARGET.

This procedure is the definition of the filter of all the data exchanged: which bit to consider and which value they should have.

$93 SAF_REQ_TARGET_1 ÷ $9F SAF_REQ_TARGET_13

Table 31. $93 SAF_REQ_TARGET_1

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$93 SAF_REQ_TARGET_1	D	W	x	x	x	x	SAF_REQ_TARGET_x																cleared by SSM_RESET

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

In the example SAF_REQ_TARGETx[15:0]:

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	1	X	1	0	X						1	0	1	0

b15 is expected 0
b[14:13] is expected to be 01 being x axis data
b12 is the parity bit, so a priori is not defined
b[11:10] is expected to be 10 because it is supposed that the data received from sensor will be the acceleration
b[3:0] is expected to be 1010 being selected the sensor number 3
b[9:4] are the data, so their value is not known a priori.
SI[15:0] & SAF_REQ_MASKx[15:0] = SAF_REQ_TARGETx[15:0]

5.4.3 Response Mask (MISO line)

To identify the sensor response, it is defined, inside the frame received from the sensor (MISO), the position of those bit to be considered.

This is done setting the bit, in SAF_RESP_MASK_x, located in the position of the bit to be considered.

$A6 SAF_RESP_MASK_1 ÷ $B2 SAF_RESP_MASK_13

Table 32. $A6 SAF_RESP_MASK_1

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$A6 SAF_RESP_MASK_1	D	W	x	x	x	x	SAF_RESP_MASK_x																cleared by SSM_RESET

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

In the example SAF_RESP_MASKx[15:0]:

The SAF_RESP_MASK allows choosing one message with respect to all that runs on SPI bus.

In this case, the IC sniffs the frame on MISO line. It considers the position of bit at 1 in the frame.

b15 is expected at 0, so b15 is in a relevant position and b15 in the SAF_RESP_MASK is set.

b[14:13] expected 01 so b14 and b13 are set

b12 don't care so b12 is left at 0

b[11:10] expected 10so b11, b10 are set

b[9:0] are the data, not useful to identify which message has to be chosen, so they are left 0

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1	1	1	0	1	1	0									

5.4.4 Response Target (MISO line)

Considering all the data that can pass on MISO line, the only one that has to be considered is the data that in the position defined in RESP MASK has the value defined in the RESP TARGET.

So:

in RESP MASK it is defined which bit to look at;

in RESP TARGET it is defined the expected value they have, coming from the sensors (MISO line)

This procedure is the definition of the filter of all the data exchanged: which bit to consider and which value they should have, sensor side.

$B9 SAF_RESP_TARGET_1 ÷ $C5 SAF_RESP_TARGET_13

Table 33. $B9 SAF_RESP_TARGET_1

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$B9 SAF_RESP_TARGET_1	D	W	x	x	x	x	SAF_RESP_ TARGET_x																cleared by SSM_RESET

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

In the example SAF_RESP_TARGETx[15:0]:

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	1	X	1	0	X									

The SAF_RESP_TARGET indicates which values are expected for the bit selected in the SAF_RESP_MASK:

b15 = 0

b[14:13] = 01

b12 don't care → x

b[11:10] = 10

b[9:0] this field correspond to the data that are not known a priori ' x

The SAF_DATA_MASK indicates where the data bit are localized, in the frame received from the sensor and identified through SAF_RESP_MASK, SAF_RESP_TARGET

In this example data are supposed to be localized in position 9:0, so the correspondent bit b[9:0] =1

SO[15:0] & SAF_RESP_MASKx[15:0] = SAF_RESP_TARGETx[15:0]

5.4.5 Data Mask (MISO line)

It is defined the position of the bit to be considered, inside the frame received from the sensor (MISO) to extract relevant sensor data from the response:

So:

$CC SAF_DATA_MASK_1 ÷ $D8 SAF_DATA_MASK_13

Table 34. $CC SAF_DATA_MASK_1

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$CC SAF_DATA_MASK_1	D	W	x	x	x	x	SAF_DATA_MASK_1																cleared by SSM_RESET

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

In the example SAF_DATA_MASKx[15:0]:

b[9:0] this field correspond to the position of the data

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	1									

Safing Threshold - specific value that sets the comparator limit for successful arming.

The integrated safing logic uses data from on-board and remote locations by decoding the various SPI communications between the interfaces and the main microcontroller.

Arming condition, required to be able to deploy, can be processed by an internal logic or external logic, according to SAFESEL bit[3], $01 SYS_CFG:

0 = internal safe engine

1 = external safe engine

Table 35. $01 SYS_CFG

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$01 SYS_CFG	-	W	0	0	0	0													SAFESEL				Updated by SSM_RESET or SPI write

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

5.5 SPI sensor data decoding (32 bit safing record mask)

Let's suppose to have the case where the 16bit sensor data are located across the first and second 16 bit words of the SPI sensor frame. In this case it is not possible to use the safing records equipped with 16 bit masks but one of the safing records 14-16 has to be used.

Despite the masks length, the sensor data is supposed to be always 16 bit maximum.

Supposing, for example, the sensor SPI frame is so organized:

14 bit acceleration data distributed between bit 10 and bit 23 of the SPI frame;

inside the 32 bit frame, data should be distributed as here indicated:

b[28:27] = channel selection, x or y; should 01 the codification for x channel

b[26]= parity bit

b[25:24] = codification data sensor field information (acceleration, self-test, error, etc.); should in the example b[25:24] =10 the codification for the acceleration data of sensor data field

b[23:6]= sensor data, split between 14 bit acceleration data [23:10] and 4 bit address b[9:6] (= 1010), in this example.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO			0	CH1	CH0	P	ST1	ST0	ACCELERATION DATA														ADD3	ADD2	ADD1	ADD0						

If the microcontroller request is (MOSI):

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SI	X	X	0	0	1	X	1	0	0	0	0	0	0	0	X	X	X	X	X	0	0	0	1	0	1	0	X	X	X	X	X	X

Acceleration data (14 bit, b[13:0]) are located in the sensor SO frame in the position b[23:10].

5.5.1 Request Mask (MOSI line)

Setting SAF_REQ_MASK[15:0] bit in SAF_REQ_MASK_x, it is defined, inside the frame, the position of the bit to be considered in the data received.

Extraction of data is done putting in bitwise AND (&) the SAF_REQ_MASK with the data received.

SAF_REQ_MASK is applied on messages recorded from MOSI pin;

The device splits the first 16 MSB bit of the 32 bit frame received in the SAF_REQ_MASKx_pt1[15:0] the other 16 LSB of the receiver frame in SAF_REQ_MASKx_pt2[15:0]

$8D SAF_REQ_MASK_14_pt1 ÷ $92 SAF_REQ_MASK_16_pt2

Table 36. $8D SAF_REQ_MASK_14_pt1

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$8D SAF_REQ_MASK_14_pt1	D	W	x	x	x	x	SAF_REQ_MASK_x_pti																cleared by SSM_RESET

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

In the example SAF_REQ_MASKx_pt1 [15:0], SAF_REQ_MASKx_pt2 [15:0]:

SAF_REQ_MASKx_pt1 [15:0]																
16MSB of 32 bit frame	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAF_REQ_MASKx_pt1 bit [15:0]	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_REQ_MASKx_pt1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	0	0
SAF_REQ_MASKx_pt2 [15:0]																
16LSB of 32 bit frame	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_REQ_MASKx_pt2 bit [15:0]	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_REQ_MASKx_pt2	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0

SAF_REQ_MASKx_pt1 [15:0]:

b[13] = 1 correspondent bit is expected to be 0

b[12:11] = 1, in correspondence of these bit positions, the bit identifying channel are localized

b[9:8] = 1, in correspondence of these bit positions, the bit identifying sensor signal are localized (acceleration, self-test..)

b[7:2] = 1 correspondent bit is expected to be 0

b[10], b[1:0] = 0 in correspondence of these bit positions, the bit identifying unknown bit a priori are localized.

SAF_REQ_MASKx_pt2 [15:0]:

b[15:13] = 0 in correspondence of these bits positions, the bit identifying unknown bit a priori are localized.

b[12:10] = 1 correspondent bit is expected to be 0

b[9:6] = 1, in correspondence of these bits positions, the bit identifying sensor address are localized.

b[5:0] = 0 in correspondence of these bit positions, the bit identifying unknown bit a priori are localized.

5.5.2 Request Target (MOSI line)

Considering bit selected through SAF_REQ_MASKx_pti, the value they have to assume is indicated REQUEST TARGET.

So:

in SAF_REQ_MASK_pti it is defined which bit position to look at;

the expected values of the bit, which are in the positions indicated by SAF_REQ_MASK_pti, are defined in SAF_REQ_TARGET_pti.

This procedure is the definition of the filter of all the data exchanged: which bit to consider and which value they should have.

$A0 SAF_REQ_TARGET_14_pt1 ÷ $A5 SAF_REQ_TARGET_16_pt2

Table 37. $A0 SAF_REQ_TARGET_14_pt1

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$A0 SAF_REQ_TARGET_14_pt1	D	W	x	x	x	x	SAF_REQ_TARGET_x																cleared by SSM_RESET

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

In the example, SAF_REQ_TARGETx_pt1[15:0], SAF_REQ_MASKx_pt2[15:0]:

SAF_REQ_TARGETx_pt1 [15:0]																
16MSB of 32 bit frame	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAF_REQ_TARGETx_pt1 bit [15:0]	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_REQ_TARGETx_pt1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
SAF_REQ_TARGETx_pt2 [15:0]																
16LSB of 32 bit frame	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_REQ_TARGETx_pt2 bit [15:0]	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_REQ_TARGETx_pt2	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0

SAF_REQ_TARGETx_pt1 [15:0]:

b[13] = 0, this bit is expected to be 0

b[12:11] = 01 they are expected to be 01 identifying in the example x axis data

b[9:8] = 10 having supposed this combination identifies that data received from sensor are the acceleration

b[7:2] these bits are expected to be 0

SAF_REQ_TARGETx_pt2 [15:0]:

b[12:10] these bits are expected to be 0

b[9:6] = 1010 being selected the sensor with this address.

SI[31:16] & SAF_REQ_MASKx_pt1[15:0] = SAF_REQ_TARGETx_pt1[15:0]

SI[15:0] & SAF_REQ_MASKx_pt2[15:0] = SAF_REQ_TARGETx_pt2[15:0]

5.5.3 Response Mask (MISO line)

To identify the sensor response, it is defined, inside the frame received from the sensor (MISO), the position of those bit to be considered.

This is done setting the bit, in SAF_RESP_MASK_x_pti, located in the position of the bit to be considered.

$B3 SAF_RESP_MASK_14_pt1 ÷ $B8 SAF_RESP_MASK_16_pt2

Table 38. $B3 SAF_RESP_MASK_14_pt1

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$B3 SAF_RESP_MASK_14_pt1	D	W	x	x	x	x	SAF_RESP_MASK_x																cleared by SSM_RESET

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

In the example SAF_RESP_MASKx_pt1[15:0], SAF_RESP_MASKx_pt2[15:0]

SAF_RESP_MASKx_pt1 [15:0]																
16MSB of 32 bit frame	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAF_RESP_MASKx_pt1 bit [15:0] - 16MSB frame	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_RESP_MASKx_pt1	0	0	1	1	1	0	1	1	0							
SAF_RESP_MASKx_pt2 [15:0]																
16LSB of 32 bit frame	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_RESP_MASKx_pt2 bit [15:0] - 16LSB frame	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_REQ_TARGETx_pt2	0						1	1	1	1	0	0	0	0	0	0

SAF_RESP_MASKx_pti allows choosing one message with respect to all that runs on SPI bus.

In this case, the IC sniffs the frame on MISO line and considers, in the 32 bit frame, the position of bit at 1.

b29 in the sensor data frame corresponds to b13 in SAF_RESP_MASKx_pt1. It is expected to be 0. Its position is relevant so it is set in SAF_RESP_MASKx_pt1.

b[28:27] in the sensor data frame correspond to b[12:11] in SAF_RESP_MASKx_pt1. They are expected to be 01. Their position is relevant so they are set in SAF_RESP_MASKx_pt1.

b26 in the sensor data frame corresponds to b10 in SAF_RESP_MASKx_pt1. It is the sensor parity bit. It is not relevant so it is left 0.

b[25:24] in the sensor data frame correspond to b[9:8] in SAF_RESP_MASKx_pt1. They are expected to be 10. Their position is relevant so they are set in SAF_RESP_MASKx_pt1.

b[23:16] in the sensor data frame correspond to b[7:0] in SAF_RESP_MASKx_pt1. This is data field; data value is not known a priori, so these bits are left 0.

b[15:10] in the sensor data frame correspond to b[15:10] in SAF_RESP_MASKx_pt2. This is data field; data value is not known a priori, so these bits are left 0.

b[9:6] in the sensor data frame correspond to b[9:6] in SAF_RESP_MASKx_pt2. These bits represent the sensor's address so they are set.

5.5.4 Response Target (MISO line)

Considering all the data that can pass on MISO line, the only one that has to be considered is the data that in the position defined in RESP MASK has the value defined in the RESP TARGET.

So:

in RESP MASK it is defined which bit to look at;

in RESP TARGET it is defined the expected value they have, coming from the sensors (MISO line)

This procedure defines a filter for all the data exchanged: which bit to consider and which value they correspond, sensor side.

$C6 SAF_RESP_TARGET_14_pt1 ÷ $CB SAF_RESP_TARGET_16_pt2

Table 39. $C6 SAF_RESP_TARGET_14_pt1

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$C6 SAF_RESP_TARGET_14_pt1	D	W	x	x	x	x	SAF_RESP_ TARGET_x																cleared by SSM_RESET

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

The SAF_RESP_TARGETx_pti indicates which values are expected for the bit selected in the SAF_RESP_MASKx_pti.

Following the example, SAF_RESP_TARGETx_pt1[15:0], SAF_TARGET_MASKx_pt2[15:0]

SAF_RESP_TARGETx_pt1 [15:0]																
16MSB of 32 bit frame	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAF_RESP_TARGETx_pt1 bit [15:0] - 16MSB frame	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_RESP_TARGETx_pt1	0	0	0	0	1	0	1	0	0							
SAF_RESP_TARGETx_pt2 [15:0]																
16LSB of 32 bit frame	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_RESP_TARGETx_pt2 bit [15:0] - 16LSB frame	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_RESP_TARGETx_pt2	0						1	0	1	0	0	0	0	0	0	0

b[13] in SAF_RESP_TARGETx_pt1 (that is b[29] in the sensor data frame) is expected to be 0.

b[12:11] in SAF_RESP_TARGETx_pt1 (that are b[28:27] in the sensor data frame) are expected to be 01

b[9:8] in SAF_RESP_TARGETx_pt1 (that are b[25:24] in the sensor data frame) are expected to be 10.

b[9:6] in SAF_RESP_TARGETx_pt2 (that are b[9:6] in the sensor data frame) depend on the sensor's address expected to be 1010.

SAF_RESP_MASKx_pti indicates where sensor data bit are localized, in the frame received from the sensor;

in SAF_RESP_TARGETx_pti it is defined the expected value for the bit with the aim to select the data inside all the frame exchanged between the device and the microcontroller.

SO[31:16] & SAF_RESP_MASKx_pt1[15:0] = SAF_RESP_TARGETx_pt1[15:0]

SO[15:0] & SAF_RESP_MASKx_pt2[15:0] = SAF_RESP_TARGETx_pt2[15:0]

5.5.5 Data Mask (MISO line)

Once identified the channel sensor direction (x or y), the kind of the data (acceleration, self-test, error) through SAF_REQ_MASKx_pti, SAF_REQ_TARGETx_pti, SAF_RESP_MASKx_pti and SAF_RESP_TARGETx_pti, it is required to extract acceleration data, through SAF_DATA_MASKx_pti.

SAF_DATA_MASKx_pti define the position of the bit to be considered, inside the frame received from the sensor (MISO) to extract the sensor data from the response:

$D9 SAF_DATA_MASK_14_pt1 ÷ $DE SAF_DATA_MASK_16_pt2

Table 40. $D9 SAF_DATA_MASK_14_pt1

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$D9 SAF_DATA_MASK_14_pt1	D	W	x	x	x	x	SAF_DATA_MASK_1																cleared by SSM_RESET

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Following the example, SAF_DATA_MASKx_pt1[15:0], SAF_DATA_MASKx _pt2[15:0].

SAF_DATA_MASKx_pt1 [15:0]																
16MSB of 32 bit frame	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAF_DATA_MASKx_pt1 bit [15:0] - 16MSB frame	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_ DATA_MASKx_pt1	0	0	0	0	0	0	0	0	1							
SAF_DATA_MASKx_pt2 [15:0]																
16LSB of 32 bit frame	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_DATA_MASKx_pt2 bit [15:0] - 16LSB frame	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAF_DATA_MASKTx_pt2	1						0	0	0	0	0	0	0	0	0	0

b[7:0] in SAF_DATA_MASKx_pt1 (that are b[23:16] in the sensor data frame) localize the field of sensor data - MSB part.

b[15:10] in SAF_DATA_MASKx_pt2 (that are b[15:10] in the sensor data frame) localize the field of sensor data - LSB part.
Safing Threshold - specific value that sets the comparator limit for successful arming.

5.6 Trigonometry for crash sensor orientation

5.6.1 Comb (Combined Data)

This command allows, performing mathematic elaboration as sum or difference between accelerations read from different sensors, to elaborate the direction of the acceleration detected from x-y axis to have an on-axis response. The direction depends on the sensor orientation inside the system.

Combine function (COMB) is enabled through COMB bit, bit 10 in SAF_CONTROL_x registers, x=1÷12 (address EF÷FA).

When the combine function is enabled, the elaboration, being x=1, 3, 5, 7, 9, 11, is:

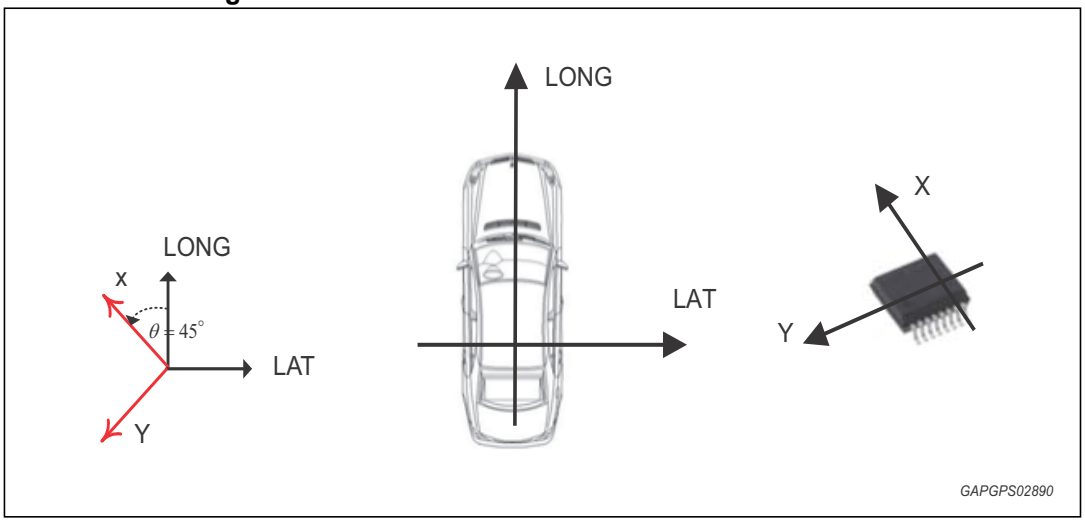
- Safing record pairs x, x+1 (1, 2 / 3, 4 / 5, 6 / 7, 8 / 9, 10 / 11, 12)
- Safing record (x) = data(x) + data (x+1)
- Safing record (x+1) = data(x) - data (x+1)

Since this elaboration is performed using two safing records, the elaboration itself is performed only after that the two safing records have been captured (CC_x=1).

As an example, consider the case of an on-board dual axis sensor which is offset 45 degrees from the center, the IC allows vector addition/subtraction to refer the sensor data to the vehicle axis.

Here below it is shown how to refer the data caught by the sensors, referred to their axis orientation, to the axis LONG and LAT of the vehicle, through simple sum or difference of sensor data, with the aim of managing the ARMING process.

Figure 34. Sensor's axis and vehicle's axis correlation



$$LONG = X * \cos(\theta) + Y * \cos(\theta + 90°)$$
$$LAT = X * \cos(\theta + 90) + Y\cos(\theta + 180)$$

if

$\theta = 45°$

$$LONG = X * \frac{\sqrt{2}}{2} - Y * \frac{\sqrt{2}}{2} = \frac{\sqrt{2}}{2}(X - Y)$$
$$LAT = -X * \frac{\sqrt{2}}{2} - Y\frac{\sqrt{2}}{2} = -\frac{\sqrt{2}}{2}(X + Y)$$

GAPGPS02889

Thresholds are referred to LONG and LAT reference system;

Data caught from sensor are referred to X and Y axis. These axis are rotated with respect to LONG and LAT axis, see *Figure 34*.

The thresholds (TH_LONG and TH_LAT thresholds) must consider the sensor direction with respect to the vehicle axis, so the trigonometric parameters (θ) are known and included in the thresholds:

$$|X - Y| > TH * \frac{2}{\sqrt{2}} = TH\_LONG$$
$$|X + Y| > TH * \frac{2}{\sqrt{2}} = TH\_LAT$$

GAPGPS02891

6 Deployment

L9680 is equipped with 12 independent loops, each of them composed by a low and high side, which can deploy at the same time or according to a given sequence.

As for the low end, dedicated ground connection for a couple of loops, SGxy, is connected to GNDSUB through a diode to guarantee the firing in case of SGxy lost.

In case the low side SRx is shorted to ground, the deployment, if requested, is guaranteed to succeed.

In any case, SSxy voltage has to be lower than 25V.

Both high side and low side are equipped with a passive turn-off to guarantee that they are always in off state except when the deployment has to take place.

6.1 Deployment requirement

Deployment features are deploy current, deploy time and deployment expiration time;

deployment expiration time is the duration time in which the deploy command remains valid, once it is received, waiting for the ARMING signal asserted; it is the correspondent of DWELL time;

These parameters are set in DCR_x registers:

Table 41. $06 DCR_0: $11 DCR_B deploy parameters

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$06 DCR_0: $11 DCR_B	(I)	W	0	0	0	0	X	X	X	X	Deploy_time						Deploy_current		Depl_expire_time		X	X	b[11:6]	=0.064 ms/count*depl time ≤ 4.032 ms
																							b[5:4]	00, 11 not used 01 1.75 A min 10 1.2 A min
																							b[3:2]	00 500 ms 01 250 ms 10 125 ms 11 0 ms

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

The Deploy Time field allows the device deploying for a configurable time 4.032 ms max (64 µs step), whereas in the low end the deploy time can be chosen between three fixed values, 0.5 ms, 0.7 ms, 2 ms.

In the low end, L9678, there is the inhibition of the combination 1.75 A, 2 ms; in the L9680 device, this constraint is no more effective, so it is under user's responsibility to prevent excessive thermal heating in the squib driver section by setting the deploy parameters carefully.

In case the deployment minimum current is set at 1.75 A, it is recommended

- for deployment times between 0.7ms and 2ms, the voltage drop across the pins is limited to 17 V max
- for deployment times up to 3.2 ms, the voltage drop across the pins is limited to 15 V max.

In case the deployment minimum current is set at 1.2 A, it is recommended

- for deployment times between 2 ms and 3.2 ms, the voltage drop across the pins is limited to 22 V max.

The voltage values above indicated (15V, 17V, 22V), relevant for long deploy time, depend on the squib resistance value being the voltage across the high side power (SSxy - SFx(y)) roughly VSF-RSQUIB*I.

Being the L9680 flexible in deploy parameters configuration, the load in the loop can be a squib -RSQUIB- or can be another component, as for example a pre-tender, whose parameters are slightly different from the squib itself, but it can be anyway driven by the L9680 itself.

The parameters in each DCR_x registers have to be confirmed at least the first time the device has to deploy, even in case they are left at their default value; the deployment does not occur otherwise.

Status of each loop is monitored in deployment status register, one for each channel:

Table 42. $13 DSR_0: $1E DSR_B

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$13 DSR_0: $1E DSR_B		R	0	0	0	0	CHxDSX	CHxSTAT	0	DCRxERR	0	0	0	0	0	0	DEP_CHx_EXP_TIME						b[15]	0 depl not successful 1 dep successful
																							b[14]	0 depl not in progress 1 depl in progress
																							b[12]	0 depl conf accepted and stored 1 depl conf change not accepted because deploy is in progress.
																							b[5:0]	8 ms/count

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Once deploy parameters have been set, it is required to assign the channels to deployment loops. This allows deploying different channels basing on the arming result. Arming results are in turn correlated to the acceleration detected and then to the collision happened.

The combination channels-deployment loop is fixed via SPI considering that external and internal arming are different.

In case of internal safing engine, four deployment loops are available and defined in registers $6E: $71;

In case of the external safing engine, three deployment loops are available, each of them associated to ARM1, ARM2, ARM3 pin (acting as FENH) value being ARM4 common for all (acting as FENL).

For deployment loop it is meant a group of different deployment channels related to the same ARMx signal.

Table 43. $6E LOOP_MATR_ARM1: $71 LOOP_MATR_ARM4

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$6E LOOP_MATR_ARM1 $6F LOOP_MATR_ARM2 $70 LOOP_MATR_ARM3 ((3)) $71 LOOP_MATR_ARM4	D	W	-	-	-	-	X	X	X	X	ARMx_LB	ARMx_LA	ARMx_L9	ARMx_L8	ARMx_L7	ARMx_L6	ARMx_L5	ARMx_L4	ARMx_L3	ARMx_L2	ARMx_L1	ARMx_L0	b[i] i=11:0 x=1:4	0 ARMx not associate to Li 1 ARMx associate to Li

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.
3. This register is meaningless in case of external safing engine.

Once fixed the deploy parameters, in order to satisfy a deploy request, the IC has to move in SAFING state or SCRAP state.

Both states are driven by specific SPI commands; SAFING state corresponds to the normal IC operation whereas SCRAP state corresponds to the operation at final disposal of the IC.

Once sent the command to move into SAFING or SCRAP state, the verification of the IC's status is readable into $04 SYS_STATE register.

The SPI commands just mentioned are:

Table 44. $30 SCRAP state, $31 SAFING state, $04 SYS_STATE

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$30 SCRAP STATE	D	W	0	0	0	0	$3535																SCRAP state command	
$31 SAFING STATE	D	W	0	0	0	0	$ACAC																SAFING state command	
$04 SYS_STATE	-	R										OPER_CTL_STATE								POWER_CTL_STATE			b[10:8]	010=SAFING 011=SCRAP
																							b[2:0]	010=RUN

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

In order to be able to deploy, the arming signals have to be serviced;

their state is readable in $6A ARM_STATE register:

Table 45. $6A ARM_STATE

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$6A ARM_STATE	-	R	0	0	0	0	0	0	0	0	0	0	PSINHINT	PSINH_EXP_TIME	ACL_PIN_STATE	ACL_VALID	ARMINT_4	ARMINT_3	ARMINT_2	ARMINT_1	0	0	b[5], b[4], b[3], b[2] ARMINT_4..ARMINT_1	Internal safing engine: Result of safing engine External safing engine: Eco of ARMx pin

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

If the internal safing machine has been selected, the four relevant bit ARMINT_4 … ARMINT_1 indicate the internal safing engine result, if the external safing engine has been selected the same four bits are the echo of the four pins ARMx.

Deployment has to be enabled via SPI, $25 SPIDEPEN because the default value is disabled:

Table 46. $25 SPIDEPEN

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$25 SPIDEPEN	S, A	W/R	-	-	-	-	DEPEN_WR[15:0] / DEPEN_STATE[15:0]																$0FF0 – DEP DISABLED $F00F – DEP ENABLED

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Deployment command request has to be received by the IC, $12 DEPCOM.

Once the deployment command has been received, the deploy time is elapsed, and deploy success bit is set (CHxDSX see below), deployment enable toggles into DEP DISABLED;

The next deploy requires the DEPEN reconfigured again as ENABLED; this feature has to be considered in case of multiple deployment, after each of them, before the next deployment the correspondent bit DEPEN has to be set again:

Table 47. $12 DEPDOM, $25 SPIDEPEN

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$12 DEPCOM	S, A	W	-	-	-		0	0	0	0	CHBDEPREQ	CHADEPREQ	CH9DEPREQ	CH8DEPREQ	CH7DEPREQ	CH6DEPREQ	CH5DEPREQ	CH4DEPREQ	CH3DEPREQ	CH2DEPREQ	CH1DEPREQ	CH0DEPREQ	b[11:0] CHxDEPREQ	0 no change to dep ch x 1 if in ARMING or SAFING, clear and start expiration time and DEPLOY_ENAB
$12 DEPCOM	-	R	-	-	-		0	0	0	0	CHBDEP	CHADEP	CH9DEP	CH8DEP	CH7DEP	CH6DEP	CH5DEP	CH4DEP	CH3DEP	CH2DEP	CH1DEP	CH0DEP	b[11:0] CHxDEP	1 expiration timer enabled, DEPCOM still valid 0 expiration timer disabled, DEPCOM no more valid
$25 SPIDEPEN	S, A	W/ R	-	-	-	-	DEPEN_STATE[15:0]																$0FF0 – DEP DISABLED	

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Deployment status of each channel is readable in $13 DSR_0 ÷ $1E DSR_B registers:

Table 48. $13 DSR_0: $1E DSR_B

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
$13 DSR_0: $1E DSR_B		R	0	0	0	0	CHxDS	CHxSTAT	0	DCRxERR	0	0	0	0	0	0	DEP_CHx_EXP_TIME						b[15]	0 depl not successful 1 dep successful
																							b[14]	0 depl not in progress 1 depl in progress
																							b[12]	0 depl conf accepted and stored 1 depl conf change not accepted because deploy is in progress.
																							b[5:0]	8 ms/count

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Each High Side (SFx) has a current comparator to indicate when the current flowing through is greater than the deployment current threshold (ITHDEPL=90%IDEPLx) and for each channel there is a timer (Current_Mon_Timer) that measures, with 16us resolution, how long the current is at high level. This parameter is considered useful for the microcontroller to identify if the deployment has been effective or not.

During a deploy event, if the current falls momentarily below the threshold, the timer stops (timer pause), and continues to count as the current turns high.

Current_Mon_Timer is refreshed upon read or when a new DEPCOM command on the channel is received . For this reason, the microcontroller reads the data after the deployment event and before a new deployment command. The current measurement stops at the end of the deployment time.

Figure 35. Current measurement during deploy

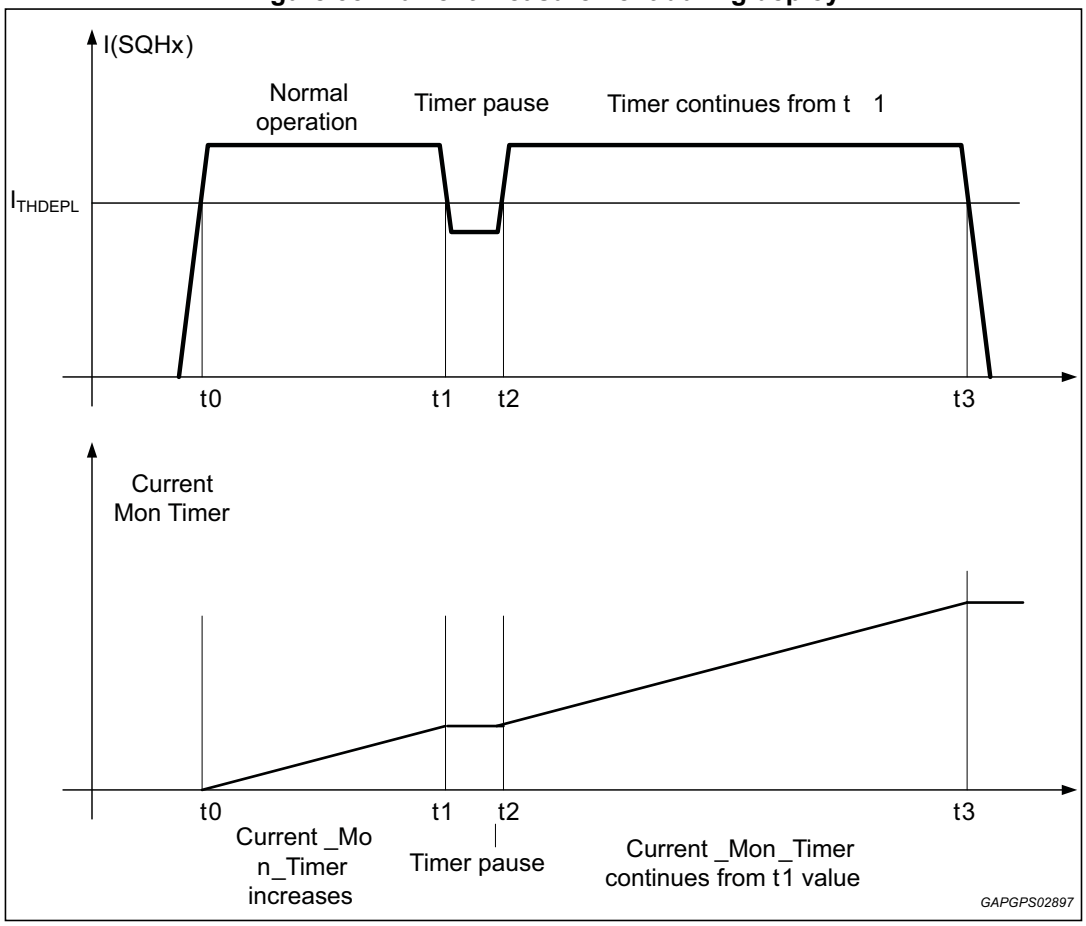


Table 49. $1F DCMTS_01 ÷ $24 DCMTS_AB

	(1)	(2)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$1F DCMTS01: $24 DCMTSAB		R	0	0	0	0	CURRENT MONITOR TIME CH 1, 3, 5, 7, 9, B								CURRENT MONITOR TIME CH 0, 2, 4, 6, 8, A								b[15:8] current monitor time y = 1, 3, 5, 7, 9, B b[7:0] current monitor time x = 0, 2, 4, 6, 8, A 16µs increment while Deploy_curr> monitor threshold channel per channel

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES
(I)=no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING.
2. R = READ
W = WRITE.

Once the deployment has started, it can be interrupted by

- over-current in the low side
- GND loss
- SSM reset
- End of deployment time

Successful deploy event is reported in GSW, in DEPOK bit. Such a flag is the "OR" of the eventual deployment success of all the twelve channels.

MISO BIT	31	30	29	28	27	26	25	24	23	22	21
MISO	SPIFLT	DEPOK	RSFLT	WDTDIS_S	ERSTATE	POWERFLT	FLT	CONVRDY2	CONVRDY1	ERR_WID	ERR_RID
GSW BIT	10	9	8	7	6	5	4	3	2	1	0

Table 50. DEPOK in GSW

GSW	MISO b[30] DEPOK = GSW b[9]
	0 = all DSR_x / CHxDS bit are = 0 (no deployment success on all channel)
	1 = at least a deployment successful on the channels.

In case the deploy success = 1, this doesn't mean that the current is really passed through the squib for the programmed time. This bit means only that no inhibition of deployment has happened. The real evaluation about the deployment is through the channel current monitor time, $1F DCMTS01 ÷ $24 DCMTSAB registers.

In case of a short to ground of the Low Side during the deployment, the current is limited by the High Side avoiding the device's damage. The same protection is available if an open load condition happens, followed by a short to ground of the Low Side.

6.2 Deployment driver protection

In order to avoid damage the IC due to eventual free-wheeling, two protections are implemented.

1. after a deployment, once the High Side is switched off, the low side is kept on for $t_{DEL\_SD\_LS}$ (50 µs min) in order to allow fly-back.
2. once low side is switched off, a protection against the overvoltage through a clamp structure is implemented.

On the Low Side there is a current limitation and overcurrent protection circuit that attends limiting the current at ILIM_SRx avoiding, in case of pin short to battery, any damage. If the malfunction lasts over $t_{FLT\_ILIM\_LS}$ (100 µs typ) the whole channel (High and Low Side) is switched off until a new deployment command, via SPI_DEPEN occurs.

The squib driver can stand the short to ground of the pins during the deployment, because the high side current is limited by the high side itself.

Squib driver can also manage the case of SR short to ground after an open circuit, being it able to detect the open circuit condition and then limiting the current overshoot as the open circuit disappears.

In case of squib's intermittence during deployment phase, current limitation is ensured by the Low Side current limitation, I_{LIM_SRx}. If the condition lasts longer than $T_{FLT\_OS\_LS}$ the High Side is switched off for $t_{OFF\_OS\_HS}$ and then on again.

This allows distinguish Open Load and Low Side short to battery cases and then properly manage them.

6.3 Deployment driver example

It is reported in this section an example of the set-up and SPI sequence implemented to deploy.

It has been chosen the external safing engine (an example of internal safing engine is described in the low level device's application note) so organized:

6.3.1 LOOP_MATRIX_ARMx:

ARM1	deployment loops 0, 1, 2, 3
ARM2	deployment loops 4, 5, 6, 7
ARM3	deployment loops 8, 9, A, B
ARM4	referred to all the low sides
ARM1=high	deployment loops 0, 1, 2, 3 armed
ARM2=ARM3=low	deployment loops 4:B NOT armed
ARM4=low	all the low side armed

6.3.2 Deployment configuration channel

deployment loops 0, 2, 4, 6, 8, A I_{depl} = 1.75 A, deply time max 500ms, deploy time = 7 ↔ 448 µs

deployment loops 1, 3, 5, 7, 9, B I_{depl} = 1.2 A, deply time max 500ms, deploy time = 7 ↔ 448 µs

Table 51. Example of SPI message to deploy

Register	(1)	(2)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Description
$01 SYS_CFG	I	W	0	X	0	0	1	0	0	0	0	0	0	0	1	0	0	1	b[15]: 0 "auto off" off ISRC&DCS reg after meas def. b[13]: 0 hi curr overcurrent detect for syncbst satbuck vccbuck def b[12]: 0 ERBST disabled in ER state def b[11]: 1 PSINH external eng mode def. b[10]: 0 short time high lev squib diag def. b[9]:0 short time def. b[8,7]: 8 sample DC-squib-temp measure def. b[6,5]: 00 4 sample def b[4]=0 passenger inhibit current def b[3]=1 external safing eng b[2]=0 VSF=20V def. b[1]=0 syncbst=12V, satbuck=7.2v rsux=6.1 def. b[0]=1 timeout disabled
$04 SYS_STATE	-	R						0	0	0						0	1	0	b[10:8]: 000 =INIT b[2:0]: 010=RUN
2C WD_STATE	-	R		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b[14:11] WD1 ERR CNT b[10:8] WD1STATE 000 init 001 run 010 test 011 reset 100 override b[7:4] WD2 ERR CNT

Table 51. Example of SPI message to deploy (continued)

Register	(1)	(2)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Description
2C WD_STATE	-	R		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b[3:0] WD2STATE 0000 init 0001 override 0010 initseed 0011 run 0100 test 0101 qual 0110 lock 01111 stopping 1000 stop 1001 reset
$35 WD_TEST	-	W	WD1_TEST								WD2_TEST								b[15:8]: $3C b[7:0]: $C3
$2C WD_STATE	-	R	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	b[14:11] WD1 ERR CNT
																			b[10:8] WD1STATE 100 override
																			b[7:4] WD2 ERR CNT
																			b[3:0] WD2STATE 0001 override
$04 SYS_STATE	-	R						0	0	1						0	1	0	b[10:8]: 001 =DIAG b[2:0]: 010=RUN
$6E LOOP_MATRIX_ARM1	D	W	X	X	X	X	0	0	0	0	0	0	0	0	1	1	1	1	b[3:0]: ARM1 assigned to L3:L0
$6F LOOP_MATRIX_ARM2	D	W	X	X	X	X	D	W	X	X	1	1	1	1	0	0	0	0	b[7:4]: ARM2 assigned to L7:L4
$70 LOOP_MATRIX_ARM3	D	W	X	X	X	X	1	1	1	1	0	0	0	0	0	0	0	0	b[11:8]: ARM3 assigned to LB:L8

Table 51. Example of SPI message to deploy (continued)

Register					(1)	(2)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Description
$02 SYS_CTL					-	W	X	X	0	0	0	0	0	0	1	1	1	0	X	X	X	X	b[13]: 0 syncbst disabled in ER state def b[12]: 0 VIN comp thresh =5.5V def b[11:10]: 00 VIBATMON comp thresh =6V def b[9]:0 ERBST 23V def. b[8,7]: 01 ER charge enabled b[6]: 1 ER BST on def. b[5]: 1 SYNC BST on def. b[4]: 0 no SPI transition to Power OFF def.
$05 POWER STATE					--	R	0	0	0	0	0	0	0	1	1	X	X	0	1	1	1	1	b[19]: 1 WAKEUP>WU_on - expected b[18]: 0 VBATMON>VBBAD - expected b[17]: 0 VBATMON>VBGOOD - expected b[16]: 0 VIN>VINBAD - expected b[15]: 0 VIN>VINGOOD - expected b[14]: 0 SYNCBST>SYNCBST_OK - expected b[13]: 0 SATSUCK>SATBUCK_OK - expected b[12]: 0 ER_BST>ER_BST_OK - expected b[11]: 0 VCC>VCC_UV expected b[10]: 0=VCC<VCC_OV expected b[9]: 0 don't care b[8]: 1 ER_BST ON - expected b[7]: 1 ER_CHARGE ON - expected b[6]: 0 ER_LCDIS_ON ER low curr disch disab b[5]: 0 ER_HCDIS_ON ER high curr disch disab b[4]: 0 ER_SWITCH state OFF b[3]: 1 SYCN_BST_ACT - ON state b[2]: 1 STABUCK_ACT - ON state b[1]: 1 VCC_ACT - ON state b[0]: 0 VSF_ACT - OFF
(3)	19	18	17	16																			
	1	0	0	0																			

Table 51. Example of SPI message to deploy (continued)

Register	(1)	(2)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Description
$06 DCR_0 $0A DCR_4 $0E DCR_8	D	W	X	X	X	X	0	0	0	1	1	1	0	1	0	0	X	X	b[11:6]: DEPLOY_TIME * 0.064ms/count up to 4.032ms max, $[111]_2$ 7*0.064ms=448µs b[5:4]: 00, 11 not used 01=1.75A 10=1.2A b[3:2]: 00 = 500ms 01 = 250ms 10 = 125ms 11 = 0ms
$07 DCR_1 $0B DCR_5 $0F DCR_9	D	W	X	X	X	X	0	0	0	1	1	1	1	0	0	0	X	X	
$08 DCR_2 $0C DCR_6 $10 DCR_A	D	W	X	X	X	X	0	0	0	1	1	1	0	1	0	0	X	X	
$09 DCR_3 $0D DCR_7 $11 DCR_B	D	W	X	X	X	X	0	0	0	1	1	1	1	0	0	0	X	X	
$13 DSR_0. $1E DSR_B	-	R			0														b[15]: CHiDS 0 = deployment not successful 1 = deployment successful b[14]: CHiSTAT 0 = deployment not in progress 1 = deployment in progress b[12]: DCRiERR 0 = depl. conf changes accepted and stored 1 = depl. conf changes rejected because deployment in progress
$31 SAFING_STATE	D	W	1	0	1	0	1	1	0	0	1	0	1	0	1	1	0	0	b[15:0]: ACAC from DIAG to SAFING state
$04 SYS_STATE (4)	-	R						0	1	0						0	1	0	b[10:8]: 010 =SAFING b[2:0]: 010=RUN
$6A ARM_STATE	-	R						0	1	0						0	1	0	b[5]: ARM4 b[4]: ARM3 b[3]: ARM2 b[2]: ARM1 Echo at ARMx pin

Table 51. Example of SPI message to deploy (continued)

Register	(1)	(2)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Description
$25 SPIDEPEN	S, A	W	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	$F00F=UNLOCK
$12 DEPCOM	S, A	W	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	b[i]: 3: 1= deploy request 0 = no change in deploy
$25 SPIDEPEN	-	R	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	$0FF0=LOCK
$13 DSR_0	-	R	1																b[15]: 1=CH0DS deployment successful
$14 DSR_1	-	R	1																b[15]: 1=CH0DS deployment successful
$15 DSR_2	-	R	1																b[15]: 1=CH0DS deployment successful
$16 DSR_3	-	R	1																b[15]: 1=CH0DS deployment successful
$17 DSR_4 - $1E DSR_B	-	R	0																
$1F DCMTS01	-	R	0	0	0	1	1	0	1	0	0	0	0	1	1	0	1	0	curr monit timer, 16µs time resol b[15:8] ch0 [11010]2 26*16µs=416µs b[7:0] ch1 [11010]2 26*16µs=416µs
$20 DCMTS23	-	R	0	0	0	1	1	0	1	0	0	0	0	1	1	0	1	0	curr monit timer, 16µs time resol b[15:8] ch0 [11010]2 26*16µs=416µs b[7:0] ch1 [11010]2 26*16µs=416µs
$21 DCMTS46 - $24 DCMTS23	-	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	curr monit timer, 16µs time resol b[15:8] ch[i+1] [00000000]2 0*16µs=0µs b[7:0]] ch[i] [00000000]2 0*16µs=0µs

1. I=INIT / D=DIAG / S=SAFING / C=SCRAP / A=ARMING / - = ALL STATES (I) = no in INIT / (D)= no in DIAG / (S)= no in SAFING / (C)= no in SCRAP / (A)= no in ARMING
2. R = READ
W = WRITE
3. Further bit over the 16 standard.
4. Once a deployment has run successfully, restart from point *5 if no parameters have to be changed.

$1F DCMTS01:

b[15:8] ch0 [11010]2 26*16 µs = 416 µs

b[7:0] ch1 [11010]2 26*16us=416us

$20 DCMTS23

b[15:8] ch3 [11010]2 26*16 µs = 416 µs

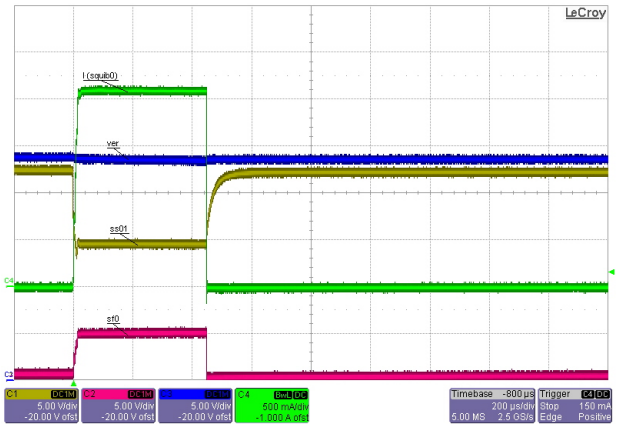
b[7:0] ch2 [11010]2 26*16 µs = 416 µs

$21 DCMTS46 - $24 DCMTS23

b[15:8] ch[i+1] [00000000]2 0*16 µs = 0 µs

b[7:0] ch[i] [00000000]2 0*16 µs = 0 µs

Figure 36. Deployment loop example



Yellow = SS01, pink = SF0, blu = VER, green = current through deployment loop 0

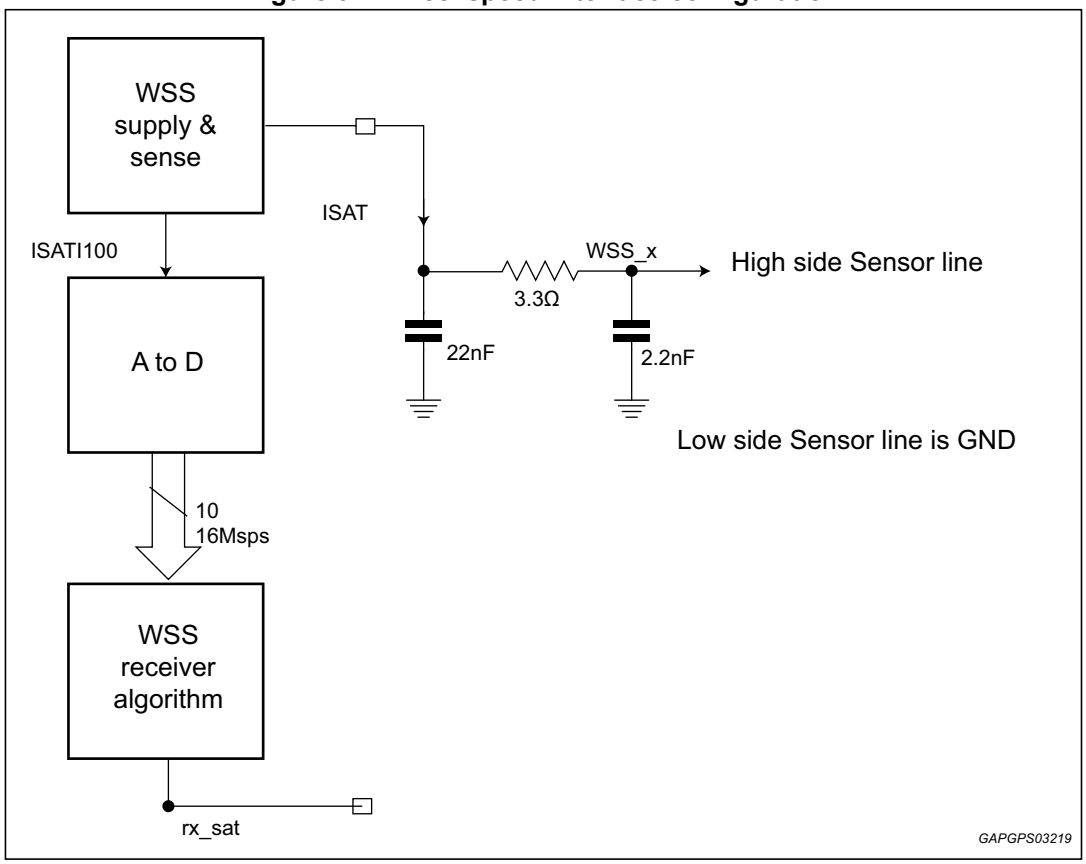
7 Wheel speed sensor interface (WSS)

Aim of this section is to define a guideline and allow the user working with WSS through L9680.

L9680 is able to decode the signals coming from wheel speed sensor by means of an analog current sensing which works together with a fast ADC and a logic control unit, which allows modulating the speed information on the WSx pin.

A simplified block diagram is shown in *Figure 37*.

Figure 37. Wheel speed interface configuration



The IC is able to operate with several sensors: in particular the following ones have been considered

- Standard active 2-level sensor (7/14 mA): Allegro ATS682LSH
- PWM encoded 2-level sensor (7/14 mA, 1 edge per tooth): Allegro ATS651LSH
- PWM encoded 2-level sensor (7/14 mA, 2 edges per tooth): Infineon TLE4942, BOSCH DF11
- VDA compliant sensor 3-level sensor (7/14/28 mA): Philips KMI22/1

Each sensor is either able to provide simple information related to wheel speed by means of frequency variation of the output current signal (STD) or is also able to provide further diagnostic by means of duty cycle modulation (PWMs) or Manchester encoder (VDA).

7.1 SPI configuration

In order to enable the Wheel speed interface, L9680 requires both software configurations, as described below.

1. Enter in DIAG mode
2. Configure sensor type (RSCRx, registers $4A-$4D)
3. Enable the WSS channels: access RSCTRL ($4E) and enable channels

7.2 Bench results - Allegro ATS682LSH STD sensor

Here are summarized the results obtained at bench working with different sensors. The blue waveform is the modulated current from the sensor, while the pink one is the signal on WSx pin. All the ASIC parameters, except where specified, are set at their default state.

Figure 38. STD operation @ different speeds

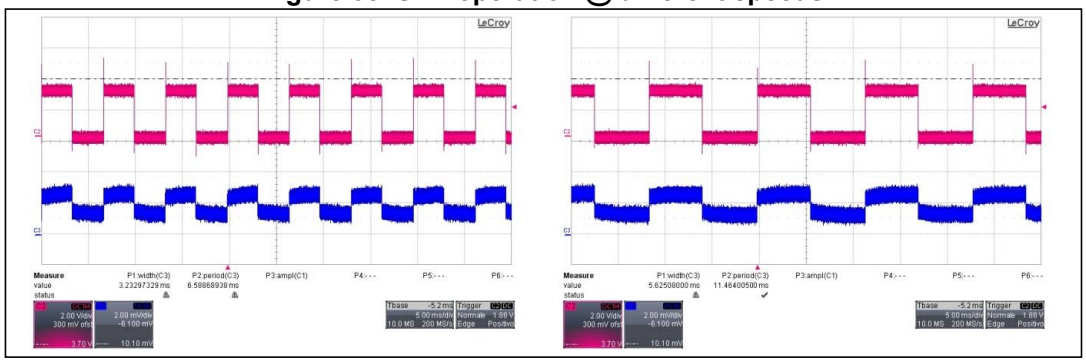


Figure 39. Possible STD operation in standstill conditions

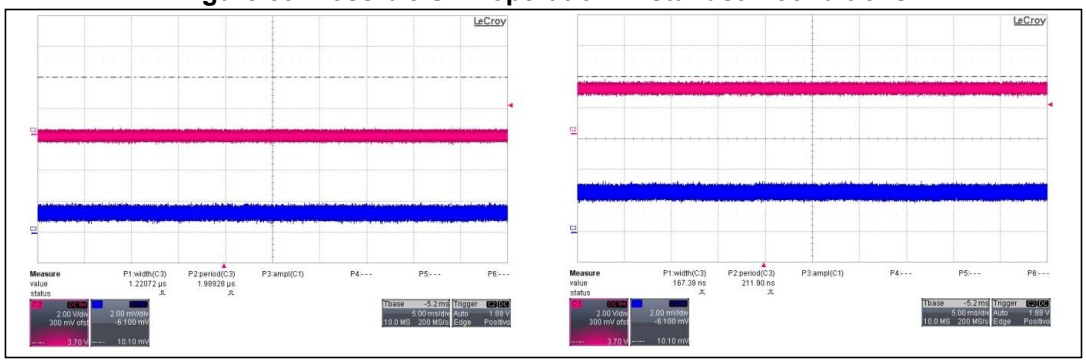
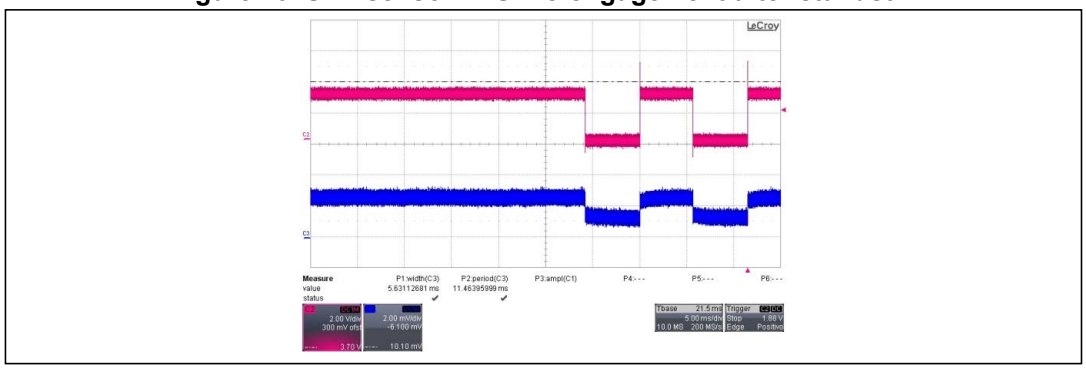


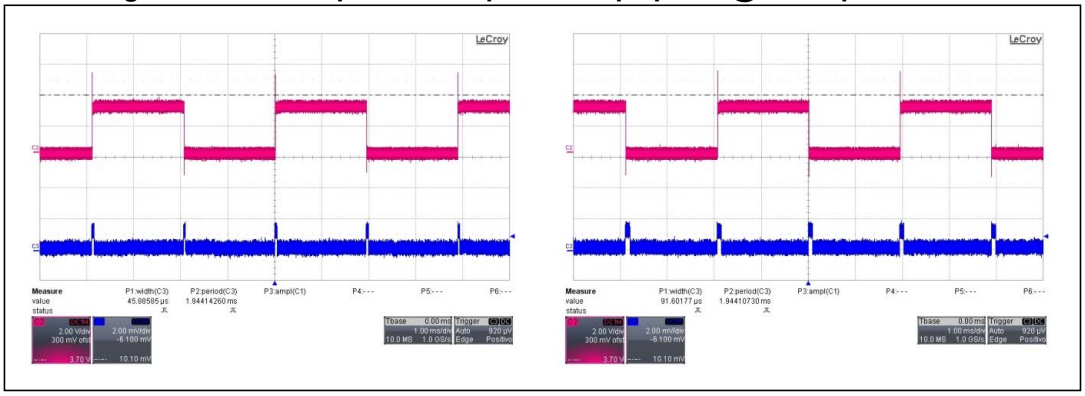
Figure 40. STD sensor: WSx re-engagement after standstill



7.3 Bench results - Allegro ATS651LSH PWM sensor

For this kind of sensors, L9680 allows reading the pulse length of the WSS signal by means of a 9 bit counter, with a resolution of 5 µs. The value of this counter can be read via SPI in the registers RSDRx in the bit [8:0]

Figure 41. PWM1 operation 45 µs and 90 µs pulse @fixed speed v = v1



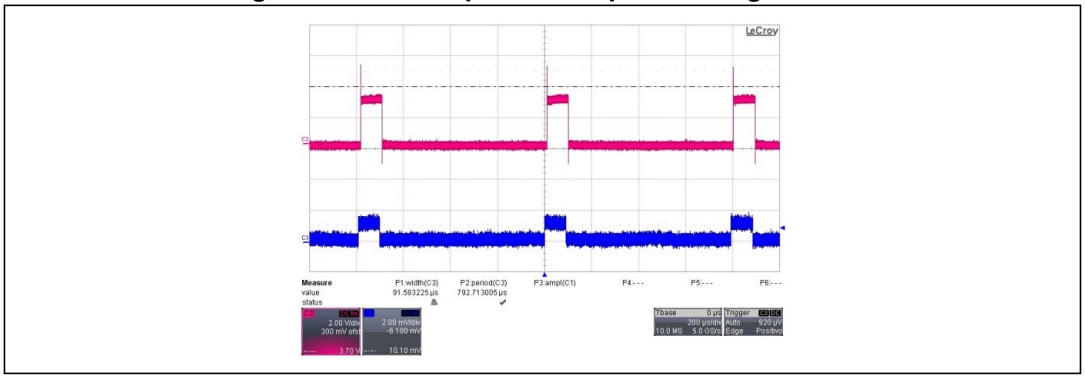
The results of the reading of the bit [8:0] of the registers RSDRx are reported in the table below

Table 52. PWM1 pulse speed length measurement in RSDRx

Pulse length [µs]	Actual Pulse length [µs]	RSDRx bit [8:0]	Measured Pulse length [µs]	Error [%]
45	45.88	000001001	45	1.9
90	91.6	000010010	90	1.7

L9680 provides the means to let the microcontroller to decode directly the waveform coming from the sensor, in pass-through mode. Such a feature can be enabled in the register (RSCRx, registers $4A-$4D)

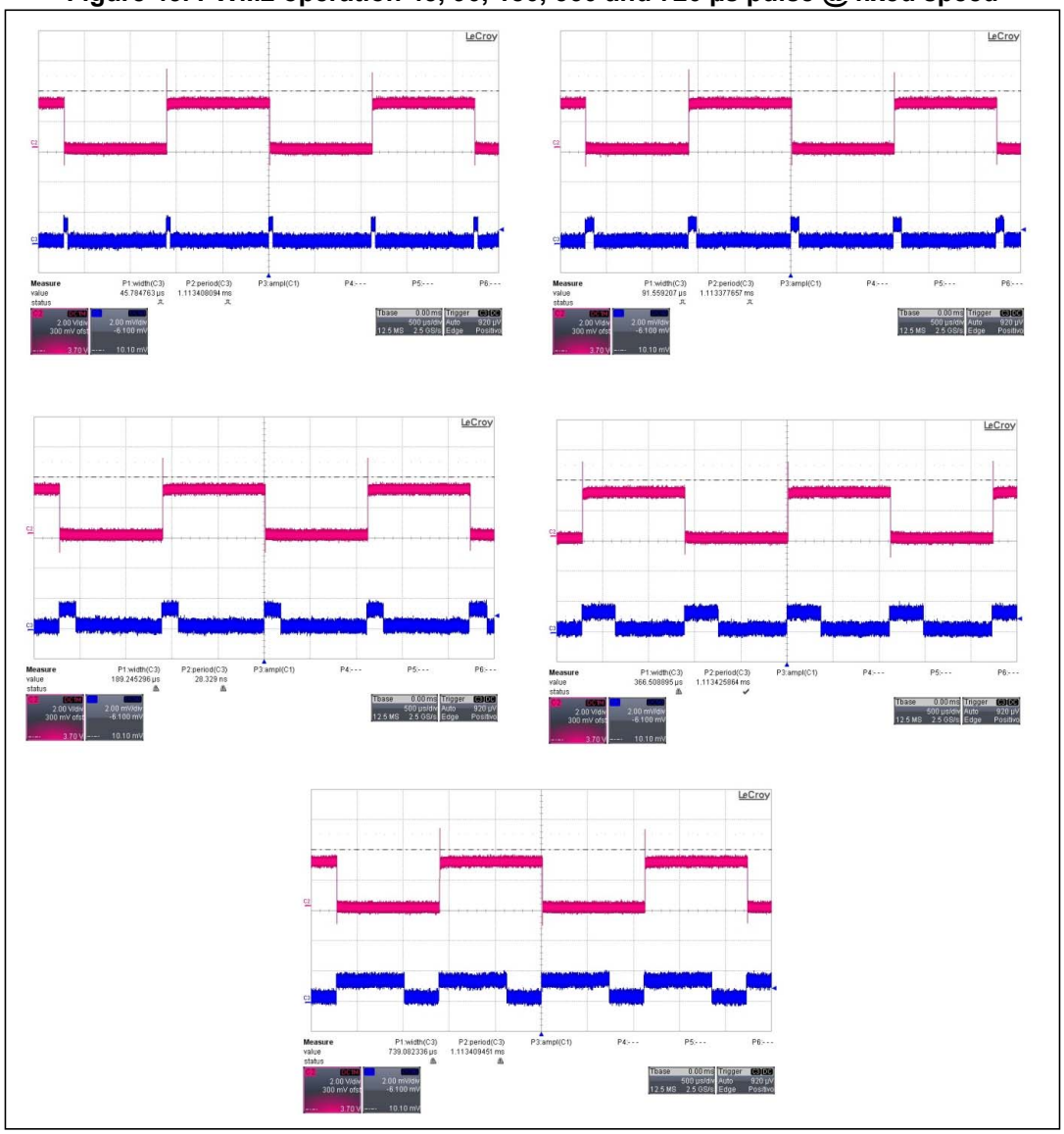
Figure 42. PWM1 operation in pass-through mode



7.4 Bench results Infineon TLE4942, BOSCH DF11 PWM sensors

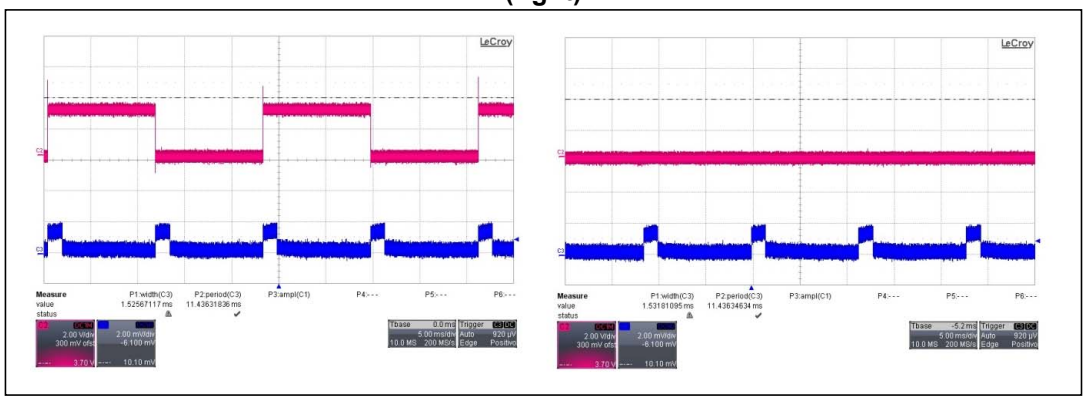
Also for this kind of sensors, L9680 allows reading the pulse length of the WSS signal by means of a 9-bit counter, with a resolution of 5 µs. The value of this counter can be read via SPI in the registers RSDRx in the bit [8:0]. In addition, the standstill condition can be detected and properly decoded.

Figure 43. PWM2 operation 45, 90, 180, 360 and 720 µs pulse @ fixed speed



Specific test has been addressed in order to verify how the L9680 behaves in standstill condition, and to verify the suppression of the WSx commutation during standstill (feature selectable via SPI through the RSCRx registers). In both situations it has been verified that the standstill bit is set in the RSDRx registers in the bit [16].

Figure 44. PWM2 operation in standstill with SSDIS bit disabled (left) and enabled (right)



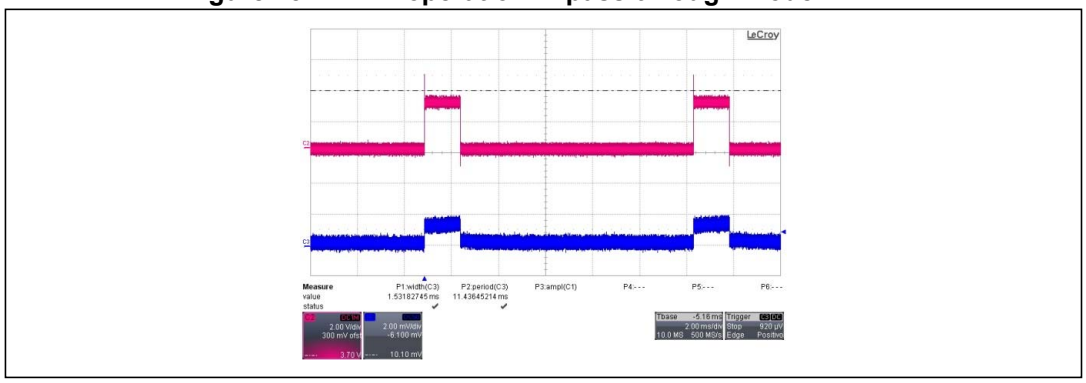
The results of the reading of the bit [8:0] of the registers RSDRx are reported in the table below:

Table 53. PWM2 pulse speed length measurement in RSDRx

Pulse length [µs]	Actual Pulse length [µs]	RSDRx bit [8:0]	Measured Pulse length [µs]	Error [%]
45	45.7	000001001	45	1.5
90	91.55	000010010	90	1.7
180	189,24	000100101	185	2.2
360	366,5	001001010	370	0.9
720	739,08	010010011	735	0.5
1440	1525	100101111	1515	0.6

L9680 provides the means to let the microcontroller decode the waveform coming from the sensor directly, in pass-through mode. Such a feature can be enabled in the register (RSCRx, registers $4A-$4D).

Figure 45. PWM2 operation in pass-through mode PWM2

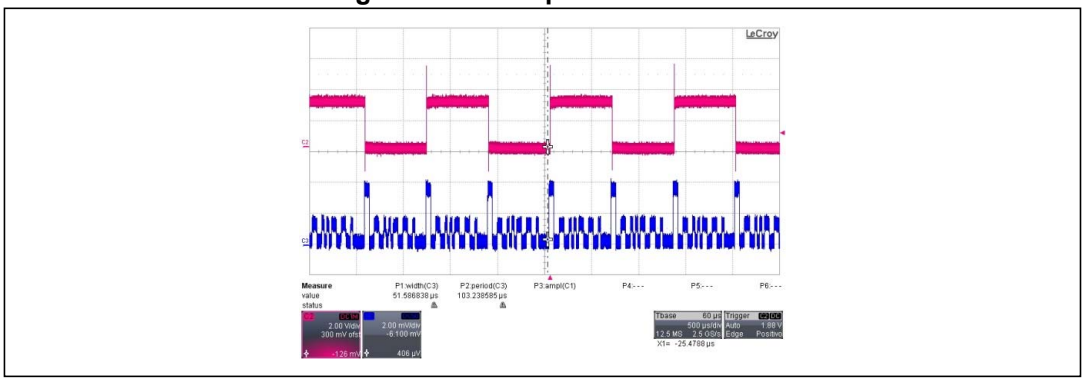


7.5 Bench results Philips KMI22/1 VDA sensor

For this sensor, L9680 allows reading the diagnostic bit transmitted by means of a Manchester decoder. Due to the fact that the number of transmitted bit depends on the speed of the wheel, L9680 also provides a counter of the decoded bit. All this info can be read in the registers RSDRx in the bit [11:8] (counter) and bit [7:0] (data).

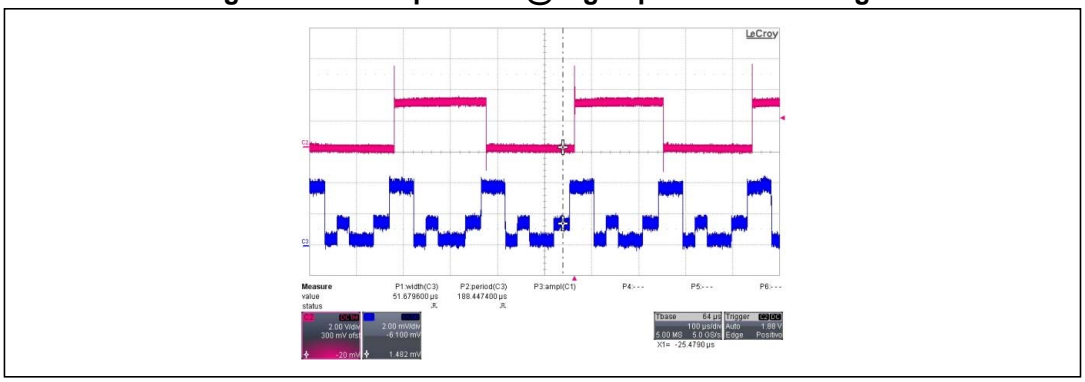
In addition, the standstill condition can be detected and properly decoded.

Figure 46. VDA operation 8bit + P



A specific sensor mode @high speed operation has been tested, where only two bits are sent and the new speed pulse overcomes the incoming frame: in this case both the data decoding and WSx commutation have been verified.

Figure 47. VDA operation @ high speed: 2bit message



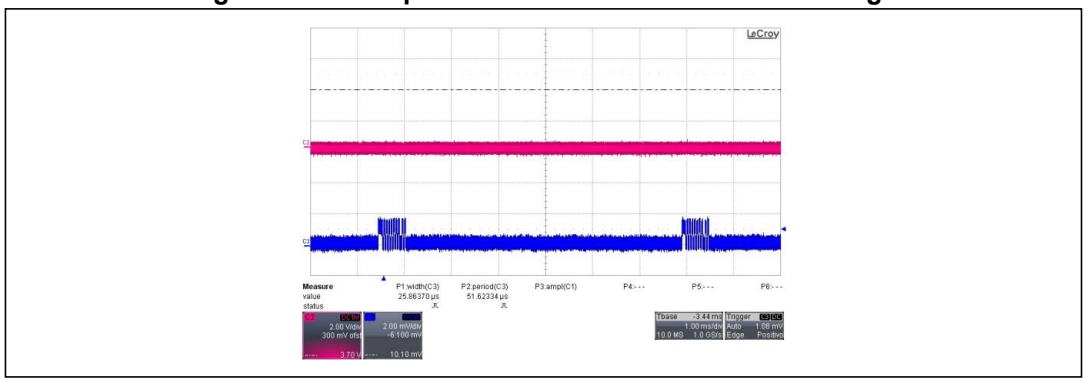
The results of the reading of the bit [11:0] of the registers RSDRx are shown in the table below.

Table 54. VDA message reading WSIRSDR [3:0]

Message sent	WSIRSDRx bit [11:8]	WSIRSDRx bit [7:0]
101110101	1001	10111010
01	0010	01

Finally, the standstill operation has been verified, as shown in the next figure.

Figure 48. VDA operation in standstill: 8bit + P message



It has been verified that the correct message is decoded in the bit [16] (standstill) and [11:0] (counter + data) of the registers RSDRx.

Appendix A Bill of materials

Table 55. Bill of materials

Description	Value	Tol.	Rating	Specific.	Package	Manufacturer	Part Number	Distr.	Order Code	UM	Qty	Designator	Notes
Printed Circuit Board	ASD551C									NR	1	ASD551C	
SMD RESISTOR	0 Ohm				0603					NR	11	R31, R32, R33, R34, R35, R36, R37, R38, R38, R40,R41	
SMD RESISTOR	0,430 Ohm	1%		1A	2512	PANASONIC	ERJ1TRQ FR43U	FN	1893025	NR	1	R1	
SMD RESISTOR	1K Ohm	5%			0805					NR	4	R2, R3, R5, R29	
SMD RESISTOR	22K Ohm	5%			0805					NR	1	R2	
SMD RESISTOR	300K Ohm	1%	0,125W		0805	MULTICOMP	MCMR08X 3003FTL	FN	2073733	NR	1	R4	
SMD RESISTOR	510Ohm	1%			0805					NR	1	R6	
SMD RESISTOR	3K Ohm	1%	0,125W		1206	MULTICOMP	MC 0.125W 1206 1% 3K	FN	9336311	NR	2	R8, R12	
SMD RESISTOR	360 Ohm	5%	0,125W		0805					NR	1	R9	
SMD RESISTOR	1,5K Ohm	5%			0805					NR	4	R10, R11, R14, R28	
SMD RESISTOR	100 Ohm	5%			0805					NR	1	R13	

Table 55. Bill of materials (continued)

Description	Value	Tol.	Rating	Specific.	Package	Manufacturer	Part Number	Distr.	Order Code	UM	Qty	Designator	Notes
SMD RESISTOR	4,7K Ohm	5%			0805					NR	8	R15, R16, R17, R18, R19, R20, R21, R22	
SMD RESISTOR	1,5 Ohm	5%			0805					NR	4	R23, R24, R25, R26	
SMD RESISTOR	1,8K Ohm	5%			0805					NR	1	R27	
SMD CAPACITOR	100nF / 100V	20 %	100V		0805					NR	22	C1, C5, C7, C8, C9, C11, C12, C14, C21, C23_1, C23_2, C24, C25, C41, C42, C43, C44, C45, C46, C81, C82, C83	
SMD CAPACITOR	10nF / 25V	20 %	25V		0805					NR	8	C10, C20, C27, C28, C29, C30, C31, C32	

Table 55. Bill of materials (continued)

Description	Value	Tol.	Rating	Specific.	Package	Manufacturer	Part Number	Distr.	Order Code	UM	Qty	Designator	Notes
SMD CAPACITOR	22nF / 25V	20 %	25V		0805					NR	37	C37, C38, C39, C40, C47, C48, C49, C50, C51, C52, C53, C54, C55, C56, C57, C58, C59, C60, C61, C62, C63, C64, C65, C66, C67, C68, C69, C70, C71, C72, C73, C74, C75, C76, C77, C78, C79	
SMD CAPACITOR	3,3nF / 25V	20 %	25V		0805					NR	4	C33, C34, C35, C36	
SMD CAPACITOR	4,7nF / 25V	20 %	25V		0805					NR	1	C80	
SMD CAPACITOR	2,2uF / 50V	20 %	50V		0805					NR	5	C2_1, C3, C84, C85_1, C86_1	
SMD TAMTALIUM CAPACITOR	10uF / 35V	20 %	35V		D					NR	3	C13, C22, C26	
SMD TAMTALIUM CAPACITOR	47uF / 35V	20 %	35V		X	KEMET	T495X476 K035ATE3 00	FN	1794757	NR	3	C2_2, C85_2, C86_2	Do not populate

Table 55. Bill of materials (continued)

Description	Value	Tol.	Rating	Specific.	Package	Manufacturer	Part Number	Distr.	Order Code	UM	Qty	Designator	Notes
ELECTROLITIC CAPACITOR	10uF / 100V	20 %	100V		RB.1/.2					NR	1	C4	
ELECTROLITIC CAPACITOR	100uF / 50V	20 %	50V		RB.15/.32					NR	1	C6	
ELECTROLITIC CAPACITOR	2200-4700uF / 50V	20 %	50V		RB.4/1.2	PANASONIC	ECOS1HA472DA	FN	1198550	NR	1	C15	
SMD INDUCTOR	4,7uH/1A	20 %	1A		B82462A4	EPCOS	B82462A4472M	FN	7429819	NR	3	L2, L3, L4	
SMD INDUCTOR	10uH / 1A	20 %	1A		B82462A4	EPCOS	B82462A4103W	FN	7429835	NR	1	L1	
DIODE SCHOTTKY 1A 40V	SS14		1A	40V	SMA	FAIRCHILD SEMICONDUCTOR	SS14	FN	1467537	NR	1	D1	
DIODE HIGH SPEED 1A 100V	LL4148		1A	100V	MINIMEL3					NR	1	D4	
SMD DIODE 1A 1000V	GS1M		1A	1000V	SMA					NR	8	D3, D5, D6, D7, D8, D9, D10, D11	
DIODE SCHOTTKY 1A 60V	SS16		1A	60V	SMA	MULTICOMP	SS16	FN	4085167	NR	1	D12	
DIODE ZENER 15V	15V / 0,500W		15V	0,500W	MINIMELF2					NR	2	DZ1, DZ2	
TVS BIDIRECTIONAL DIODE	SM6T33CA				SMB	ST Microelectronics	SM6T33CA	FN	9802681	NR	1	D2	

Table 55. Bill of materials (continued)

Description	Value	Tol.	Rating	Specific.	Package	Manufacturer	Part Number	Distr.	Order Code	UM	Qty	Designator	Notes
DUAL TRANSISTOR NPN/PNP	PUMD15				SOT-363	NXP	PUMD15	FN	8738335	NR	2	U4, U5	
MOSFET P-Channel	STD10PF06				DPAK	ST Microelectronics	STD10PF06T4	FN	1468000	NR	1	Q1	
MOSFET N-Channel	STD60NF06				DPAK	ST Microelectronics	STD60NF06T4	FN	9935436	NR	1	Q2	
PNP BIPOLAR TRANSISTOR	BCP53-10				SOT223	NXP	BCP53-10	FN	8734810	NR	2	Q3, Q4	
NPN TRANSISTOR	BCP56				SOT223	NXP	BCP56	FN	1081257	NR	1	Q5	
L9678	L9678				TQFP64_L	ST Microelectronics	L9678			NR	1	U1	
L9679	L9679				TQFP100_EXPOSED PAD	ST Microelectronics	L9679			NR	1	U2	
L9680	L9680				TQFP100_EXPOSED PAD	ST Microelectronics	L9680			NR	1	U3	

Table 55. Bill of materials (continued)

Description	Value	Tol.	Rating	Specific.	Package	Manufacturer	Part Number	Distr.	Order Code	UM	Qty	Designator	Notes
TEST POINT					TEST POINT					NR	36	TP1, TP2, TP3, TP4, TP5, TP6, TP7, TP8, TP9, TP10, TP11, TP12, TP13, TP14, TP15, TP16, TP17, TP18, TP19, TP20, TP21, TP22, TP23, TP24, TP25, TP26, TP27, TP28, TP29, TP30, TP31, TP32, 33, TP34, TP35, TP36	
AUTOMOTIVE CONNECTOR 80 WAY	502225-0801		80 POLES	MALE	SD-502225-001	MOLEX	502225-0801	FN MO	1874028538-502225-0801	NR	1	J30	
CONNECTOR CART	p. 7.62mm				GMKDS 3/2-7.62					NR	1	J4	Do not populate

Table 55. Bill of materials (continued)

Description	Value	Tol.	Rating	Specific.	Package	Manufacturer	Part Number	Distr.	Order Code	UM	Qty	Designator	Notes
HEADER 3X1	p. 2.54mm		3 POLES	MALE	SIP3					NR	6	J3, J6, J7, J11, J12, J28	
GOUTTE CONTACT					JUMP A SALDARE					NR	3	J1, J13, J20	
HEADER "T" TIPE	p. 2.54mm		4 POLES	MALE	SIP4-T					NR	2	J15, J16	
HEADER 2X1	p. 2.54mm		2 POLES	MALE	SIP2					NR	6	J2, J5, J8, J9, J17, J29	
HEADER 7X1	p. 2.54mm		7 POLES	MALE	SIP7					NR	2	J18, J19	
HEADER 34X4	p. 2.54mm		136 POLES	MALE/FEMALE	IDC34X4	SAMTEC	ESQ-134-14-G-D	FN	1769251	NR	1	J10	Each connector is composed by 2 ESQ-134-14-G-D
INFORMATION ABOUT MECHANICAL PARTS													
JUMPER CONFIGURATION			2 POLES	FEMALE						NR	14		Provide in envelope
INFORMATION ABOUT SPARE PARTS													
FEMALE AUTOMOTIVE CONNECTOR 48WAY			48 POLES	FEMALE		MOLEX	64320-1319	FN	2060665	NR	1		

Table 55. Bill of materials (continued)

Description	Value	Tol.	Rating	Specific.	Package	Manufacturer	Part Number	Distr.	Order Code	UM	Qty	Designator	Notes
FEMALE AUTOMOTIVE CONNECTOR 32WAY			32 POLES	FEMALE		MOLEX	64319-3211	FN	1874144	NR	1		
TERMINAL 14-16 AWG				FEMALE		MOLEX	64323-1039	FN	2060654	NR	16		
TERMINAL 0,5MM2				FEMALE		MOLEX	64322-1039	FN RS	1830393 723-9362	NR	64		
JACK BANANA WHITE	4mm			WHITE				FN	110-1105	NR	6		
JACK BANANA YELLOW	4mm			YELLOW				FN	110-1104	NR	2		
JACK BANANA GREEN	4mm			GREEN				FN	110-1102	NR	4		
JACK BANANA BLUE	4mm			BLUE				FN	110-1103	NR	9		
JACK BANANA RED	4mm			RED				FN	110-1098	NR	17		
JACK BANANA BLACK	4mm			BLACK				FN	110-1101	NR	16		
RED CABLE 1mmq - LEGHT= 70cm	1mmq			RED						NR	4		Lenght = 70cm

Table 55. Bill of materials (continued)

Description	Value	Tol.	Rating	Specific.	Package	Manufacturer	Part Number	Distr.	Order Code	UM	Qty	Designator	Notes
RED CABLE 0.35mmq - LEGHT= 70cm	0.35mmq			RED						NR	13		Lenght = 70cm
BLACK CABLE 0.35mmq - LEGHT= 70cm	0.35mmq			BLACK						NR	16		Lenght = 70cm
WHITE CABLE 0.35mmq - LEGHT= 70cm	0.35mmq			WHITE						NR	6		Lenght = 70cm
YELLOW CABLE 0.35mmq - LEGHT= 70cm	0.35mmq			YELLOW						NR	2		Lenght = 70cm
GREEN CABLE 0.35mmq - LEGHT= 70cm	0.35mmq			GREEN						NR	4		Lenght = 70cm
BLUE CABLE 0.35mmq - LEGHT= 70cm	0.35mmq			BLUE						NR	9		Lenght = 70cm

Revision history

Table 56. Document revision history

Date	Revision	Changes
21-Dec-2017	1	Initial release.