

Contents

| | | |
|-------|---|----|
| 1 | Control Logic | 4 |
| 1.1 | General Description | 4 |
| 1.2 | CMC Description..... | 4 |
| 1.3 | General Working Modes..... | 4 |
| 1.3.1 | Normal Operation Mode (NOM)..... | 5 |
| 1.3.2 | Command Mode (CM)..... | 5 |
| 1.3.3 | Diagnostic Mode (DM) | 6 |
| 1.3.4 | Failsafe Tasks and Error Codes | 7 |
| 1.3.5 | Temperature Out-Of-Range Check..... | 8 |
| 2 | Signal Conditioning | 9 |
| 2.1 | A/D Conversion..... | 9 |
| 2.2 | Bridge Sensor Signal Conditioning Formula | 10 |
| 2.3 | Digital Bridge Sensor Signal Filter Function | 11 |
| 2.4 | Temperature Conditioning Formula..... | 11 |
| 3 | Digital PWM Output..... | 12 |
| 4 | Digital LIN Interface..... | 14 |
| 4.1 | General Description | 14 |
| 4.2 | LIN Protocol | 14 |
| 4.2.1 | Frame | 14 |
| 4.2.2 | Bit Rate..... | 15 |
| 4.2.3 | Synchronization..... | 15 |
| 4.2.4 | Bit Sampling | 15 |
| 4.2.5 | Protected Identifier (PID)..... | 15 |
| 4.2.6 | Checksum | 15 |
| 4.3 | LIN Publisher Frame – Normal Operation Mode (NOM) | 16 |
| 4.4 | LIN Slave Status Information..... | 17 |
| 4.5 | LIN Command Mode | 18 |
| 4.6 | LIN Transport Layer..... | 19 |
| 4.6.1 | Assign-NAD..... | 20 |
| 4.6.2 | Conditional-Change-NAD..... | 20 |
| 4.6.3 | Read-By-Identifier | 21 |
| 4.6.4 | Assign-Frame-Identifier | 24 |
| 4.6.5 | Assign-Frame-Identifier-Range | 25 |
| 4.6.6 | Save-Configuration | 26 |

| | | |
|-------|--|----|
| 4.6.7 | Go-To-Sleep..... | 27 |
| 4.6.8 | Data-Dump..... | 27 |
| 4.7 | LIN Sleep Mode..... | 28 |
| 4.8 | Differences between LIN Mode 1.3 and LIN Mode 2.0/2.1 – Summary..... | 28 |
| 5 | Serial Digital I ² C Interface..... | 29 |
| 5.1 | General Description..... | 29 |
| 5.2 | Digital I ² C Output..... | 30 |
| 5.3 | I ² C Protocol..... | 31 |
| 6 | Interface Commands..... | 34 |
| 6.1 | Command Set..... | 34 |
| 6.2 | Command Processing..... | 38 |
| 6.3 | Output Data in Command and Diagnostic Mode..... | 38 |
| 6.4 | Detailed Description of Oscillator Frequency Adjustment..... | 38 |
| 7 | EEPROM and RAM..... | 40 |
| 7.1 | Programming the EEPROM..... | 40 |
| 7.2 | EEPROM and RAM Contents..... | 40 |
| 7.3 | Configuration Words..... | 43 |
| 7.4 | EEPROM Signature..... | 48 |
| 7.5 | EEPROM Write Locking..... | 48 |
| 8 | Related Documents..... | 49 |
| 9 | Glossary..... | 49 |
| 10 | Document Revision History..... | 50 |

List of Figures

| | | |
|------------|---|----|
| Figure 1.1 | Modes of Digital Serial Communication..... | 5 |
| Figure 3.1 | PWM Output Characteristics..... | 12 |
| Figure 4.1 | LIN Publisher Frame..... | 16 |
| Figure 5.1 | I ² C Read Request during NOM, Temporary DM..... | 30 |
| Figure 5.2 | I ² C Read Request after Detecting an Error (Steady DM)..... | 30 |
| Figure 5.3 | I ² C Read Request Answering a Command (CM)..... | 30 |
| Figure 5.4 | Principles of I ² C Protocol..... | 31 |
| Figure 5.5 | Write Operation I ² C..... | 32 |
| Figure 5.6 | Read Operation I ² C – (Data Request)..... | 32 |
| Figure 5.7 | Timing I ² C Protocol..... | 33 |
| Figure 7.1 | Source Code Signature Generation..... | 48 |

List of Tables

| | | |
|------------|---|----|
| Table 1.1 | Error Detection Functionality and Error Codes | 7 |
| Table 3.1 | PWM Configuration..... | 12 |
| Table 4.1 | LIN Status Bits Transmitted in Normal Operation Mode (NOM)..... | 17 |
| Table 4.2 | Assign-NAD Request..... | 20 |
| Table 4.3 | Assign-NAD Positive Response | 20 |
| Table 4.4 | Conditional-Change-NAD Request..... | 21 |
| Table 4.5 | Conditional-Change-NAD Positive Response | 21 |
| Table 4.6 | Read-By-Identifier Request | 21 |
| Table 4.7 | Read-By-Identifier - Supported Identifiers | 22 |
| Table 4.8 | Read-By-Identifier Positive Response | 23 |
| Table 4.9 | Read-By-Identifier Negative Response | 23 |
| Table 4.10 | Assign-Frame-Identifier Request | 24 |
| Table 4.11 | Assign-Frame-Identifier Positive Response | 24 |
| Table 4.12 | Assign-Frame-Identifier-Range Request | 25 |
| Table 4.13 | Assign-Frame-Identifier-Range Positive Response..... | 25 |
| Table 4.14 | Save-Configuration Request..... | 26 |
| Table 4.15 | Save-Configuration Positive Response | 26 |
| Table 4.16 | Save-Configuration Negative Response | 26 |
| Table 4.17 | Go-To-Sleep Request..... | 27 |
| Table 4.18 | Data-Dump Request | 27 |
| Table 4.19 | Data-Dump Positive Response..... | 27 |
| Table 4.20 | LIN modes..... | 28 |
| Table 5.1 | Timing I ² C Protocol | 33 |
| Table 6.1 | Command Set | 34 |
| Table 6.2 | Serial Digital Interface Output Registers | 38 |
| Table 6.3 | Oscillator Frequency Adjustment Sequence | 39 |
| Table 7.1 | EEPROM and RAM Contents..... | 41 |
| Table 7.2 | Configuration Word CFGLIN..... | 43 |
| Table 7.3 | Configuration Word CFGAFE | 44 |
| Table 7.4 | Configuration Word CFGTS..... | 45 |
| Table 7.5 | Configuration Word CFGAPP | 46 |
| Table 7.6 | Configuration Word CFGSF..... | 47 |

1 Control Logic

1.1 General Description

The control logic of the ZSSC3170 consists of the calibration microcontroller (CMC), the module control logic of the analog-to-digital converter (ADC), and serial digital Interface. The configuration of the various modes of the device is done by programming an EEPROM.

The CMC controls the measurement cycle and performs the calculations for sensor signal conditioning. This eliminates the gain deviation, the offset, the temperature deviation, and the non-linearity of the pre-amplified and A/D-converted sensor signal. The controller of the A/D conversion is started by the CMC and executed as a continuous measurement cycle. The conditioning calculation by the CMC is performed in parallel with the A/D conversion.

An external microcontroller can read the sensor signal conditioning results from the ZSSC3170 via the LIN interface. PWM output is also available.

Communication between an external microcontroller and the sensor system consisting of the transducer and the ZSSC3170, especially for calibration purposes, is done via serial digital interfaces. Communication protocols according to the LIN and I²C standards are supported. During calibration, the CMC performs internal processing of received interface commands. As a result, the measurement cycle is interrupted if a command is received.

1.2 CMC Description

The calibration microcontroller (CMC) is especially adapted to the tasks connected with the signal conditioning. The main features are as follows:

- The microcontroller uses 16-bit processing width, and it is programmed via ROM.
- A watchdog timer controls the proper operation of the microcontroller.
- Constants/coefficients for the conditioning calculation are stored in the EEPROM. The EEPROM is mirrored to the RAM after power-on or after re-initialization from EEPROM by sending a specific command to the serial interface.
- Parity is checked continuously during every read from RAM. If incorrect data is detected, the Diagnostic Mode is activated (error code is written to the serial digital output).

1.3 General Working Modes

ZSSC3170 supports three separate working modes:

- Normal Operation Mode (NOM)
- Command Mode (CM)
- Diagnostic Mode (DM)

1.3.1 Normal Operation Mode (NOM)

After power-on, the ZSSC3170 completes an initialization routine during which the EEPROM is mirrored to RAM and the content is checked against a stored signature. If enabled, a ROM signature check is processed (see Table 7.6). If any error is detected, the Diagnostic Mode is activated. Otherwise the configuration of the ZSSC3170 is set, serial digital interfaces are enabled, and Normal Operation Mode is started.

In LIN mode, LIN communication is always available. This is used for reading the sensor signal using a publisher frame or for end-of-line configuration and calibration using transport layer services. For details, see section 4.

In PWM mode, a start window of 30ms (nominal) is opened. During the start window, both PWM pins are set to the recessive level and the device can receive LIN frames via both pins. To activate the Command Mode for end-of-line configuration and calibration, use the transport layer service *Data-Dump* (see section 4.6.8) to transmit the START_CM command. If this command is received, NOM is stopped and the ZSSC3170 waits for further commands. If no valid START_CM command is received during the PWM start window, the ZSSC3170 continues normal operation (NOM).

In NOM, the continuous measurement cycle and conditioning calculations are processed. Bridge sensor and temperature signal conditioning results are frequently refreshed. The conditioning results can be read via the serial digital interfaces (LIN or I²C), or they can generate a PWM output. Read out of the conditioning result via LIN or I²C does not interrupt the continuous processing of the signal measurement and conditioning routine.

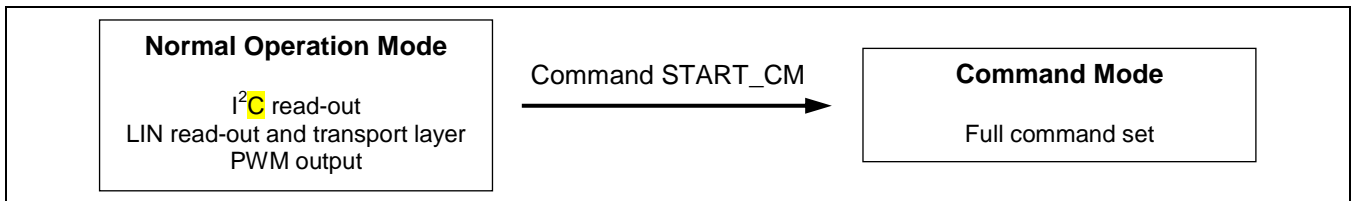
1.3.2 Command Mode (CM)

The CM start command START_CM generates an interrupt to the NOM, which stops the measurement cycle. The ZSSC3170 changes to CM only after receiving the START_CM command via the digital serial interface (LIN or I²C). This protects the ZSSC3170 against interruption of processing during NOM (continuous signal conditioning mode) and/or unintentional changes of configuration. In CM, the full set of commands is supported (see section 6.1).

If the ZSSC3170 receives a command other than START_CM in NOM, it is not valid. In this case, the invalid command is ignored and no interrupt to the continuous measurement cycle is generated.

In CM, the full command set is enabled for processing. During processing of a received command, the serial interfaces are disabled; no further commands are recognized. After finishing the processing, the CMC waits for further commands or processes loops continuously (e.g. after measurement commands).

Figure 1.1 Modes of Digital Serial Communication



EEPROM programming is only enabled after receiving the EEP_WRITE_EN command (see section 6.1).

In LIN mode, it is always possible to change to Command Mode via transport layer communication (see section 4.5), but this is not intended to use in a running cluster during the application.

In PWM mode, the START_CM command must be transmitted during the start window (nominal 30ms). After the start window has expired, it is not possible to communicate with the ZSSC3170 via its PWM pins.

In I²C mode, starting the CM via I²C communication (pins SCL, SDA) is possible at any time.

1.3.3 Diagnostic Mode (DM)

The ZSSC3170 detects various failures. When a failure is detected, Diagnostic Mode (DM) is activated.

In LIN mode, the DM is indicated by error flags contained in the LIN signal that is transmitted when responding with a publisher frame in NOM. Thus every read-out of the bridge sensor signal and temperature is paired with failure status information. See section 4.4 for a detailed description of the LIN status information.

In PWM mode, the DM is indicated by output of the recessive state (i.e., PWM output is in the high impedance state). Note that the recessive level depends on the selected PWM output driver. The low-side switch (LSS, which is selected by setting CFGAPP:PWMMODE to 0) requires an external pull-up resistor for generating high levels, so the LOUT pin remains HIGH in DM. The high-side switch (HSS, which is selected by setting CFGAPP:PWMMODE to 1) requires an external pull-down resistor for generating low levels so the HOUT pin remains LOW in DM.*

During DM, LIN communication is possible via all output pins (LIN, HOUT, LOUT). This ensures that a non-configured device is accessible via LIN for end-of-line configuration. Transport layer service *Read-by-Identifier-32* (see section 4.6.3 and Table 4.7) returns an error code specifying the reason for DM activation. Error codes are listed in Table 1.1. Error codes can also be read out via I²C during DM using the command GET_ERR_STATUS.

Note that error detection functionality can be partly enabled/disabled by configuration word CFGSF (e.g. sensor connection check, sensor short check, sensor aging check, ROM check, etc.; see Table 7.6).

There are three options for Diagnostic Mode:

- Steady Diagnostic Mode

In steady DM, the measurement cycle is stopped and failure notification is activated. If enabled, a reset after the time-out of a watchdog is executed.

- Temporary Diagnostic Mode

There is a failure counting sequence that can result in a temporary DM. DM is activated after two consecutively detected failure events and is deactivated after a failure counter counts down if the failure condition is no longer detected. The measurement cycle is continuously processed.

- Power and Ground Loss

Power and ground loss cases are signaled by interrupting the communication/data stream at the output. Output pins are set to high-impedance states. The output level is determined by the external load.

* This function is valid for ZSSC3170 silicon revisions \geq F. Note that the previous revision "E" does not provide the Sensor Connection Check (SCC) or Sensor Short Check (SSC) diagnostics functionality.

1.3.4 Failsafe Tasks and Error Codes

Note: Error codes can be bitwise ORed. MSB is even parity. The reset after the watchdog timeout overwrites previously detected errors.

Table 1.1 Error Detection Functionality and Error Codes

| Failsafe Task | Description | Messaging Time | Error Code | Activation | Action |
|--------------------------------|--|---------------------------------|--------------|------------------|--|
| Oscillator Fail | Oscillator is observed generating clock pulses by an asynchronous timing logic | < 200µs | - | - | Temporary DM |
| ROM Signature | CMC ROM signature check LIN controller ROM signature check | Start-up | 6500 | CFGSF: CHKROM | Steady DM |
| EEPROM Multiple-Bit Error | Detection of non-correctable multiple-bit error per 16-bit word | Start-up | 6440 | - | |
| EEPROM Signature | Signature check for RAM mirror of EEPROM content | Start-up | 6600 | - | |
| Watchdog | Watchdog time-out during start-up routine (number of clock cycles after power-on: 262144 if <i>CFGAPP:ADCSLOW</i> is disabled (see Table 7.5); 524288 if <i>ADCSLOW</i> is enabled) or measurement cycle (2 x conversion cycle time) | Start-up or 2 measurement times | 6402 6420 | - | Steady DM or reset after watchdog time-out (enabled by <i>CFGAPP:DMRES</i>) |
| RAM Parity | Parity check at every RAM access | Immediately | 6404 6408 | - | |
| Register Parity | Permanent parity check of configuration registers | Immediately | 6410 | - | |
| Arithmetic Check | Functional check of arithmetic unit | One measurement cycle | 6480 | - | |
| SSC ¹⁾ | Sensor short check | Two measurement cycles | A820 | CFGSF: CHKSSC | Temporary DM |
| SCC ¹⁾ | Sensor connection check | | A840 | CFGSF: CHKSCC | |
| SAC | Sensor aging check | | A880 | CFGSF: CHKSAC | |
| BCC | Broken chip check | | AA00 | CFGSF: CHKBCC | |
| Temperature Out-of-Range Check | Detection of ADC range overflow for temperature measurement | | C900 | CFGSF: CHKOOR | |
| Power & Ground Loss | Power and ground loss detection | < 5ms | - | - | Reset |

1) Note: For ZSSC3170 revisions ≤ E, there is no diagnostic functionality in PWM Mode with the low-side or high-side switch enabled.

1.3.5 Temperature Out-Of-Range Check

The temperature out-of-range check detects whether the ADC dynamic range has been exceeded during the temperature measurement. The signal raw value is checked if it is equal to 0 or $(2^{r_{ADC}} - 1)$. This can result from various causes: the external temperature sensor is unconnected; the analog temperature input channel is not sufficiently adjusted; or the input signal is out of the ADC range.

2 Signal Conditioning

2.1 A/D Conversion

During NOM, the analog preconditioned sensor signal is continuously converted from analog to digital. The A/D conversion has a resolution r_{ADC} of 13-bit or 14-bit, as set by configuration word CFGAFE:ADCRES, and it is performed in the two-step conversion mode. It is configurable for the inherent range shift rs_{ADC} by the configuration word CFGAFE:ADCRS. The resolution for the A/D conversion is equal for all measurements in the measurement cycle (e.g., input voltage, temperature, auto-zero, etc.). The measured digital raw values (e.g., bridge sensor signal, temperature) are determined by the following equations:

⇒ **Analog differential input voltage to A/D conversion**

Measured value V_{IN_DIFF} to be conditioned:

$$V_{ADC_DIFF} = a_{IN} \cdot V_{IN_DIFF} + a_{XZC} \cdot V_{XZC}$$

⇒ **Digital raw A/D conversion results**

$$Z_{ADC} = 2^{r_{ADC}} \cdot \left(\frac{V_{ADC_DIFF} + V_{OFF}}{V_{ADC_REF}} + rs_{ADC} \right)$$

⇒ **Auto-zero value**

$$Z_{AZ} = 2^{r_{ADC}} \cdot \left(\frac{V_{OFF}}{V_{ADC_REF}} + rs_{ADC} \right)$$

⇒ **Auto-zero corrected raw A/D conversion result**

$$Z_{CORR} = Z_{ADC} - Z_{AZ} = 2^{r_{ADC}} \cdot \frac{V_{ADC_DIFF}}{V_{ADC_REF}}$$

| | |
|-----------------|---|
| V_{IN_DIFF} | Differential input voltage to analog front end |
| V_{OFF} | Residual offset voltage of analog front end (which is eliminated by the $Z_{ADC} - Z_{AZ}$ difference calculation) |
| V_{XZC} | Extended zero compensation voltage (programmable via CFGAFE:PXZC) |
| a_{IN} | Gain of analog front end for differential input voltage |
| a_{XZC} | Gain for extended zero compensation voltage |
| V_{ADC_DIFF} | Differential input voltage to ADC |
| V_{ADC_REF} | ADC reference voltage (ratiometric reference for measurement) |
| r_{ADC} | Resolution of A/D conversion (13/14-bit) |
| rs_{ADC} | Range shift of A/D conversion Bridge Sensor Measurement: $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$ Temperature Measurement: $\frac{1}{2}$ |

2.2 Bridge Sensor Signal Conditioning Formula

The digital raw value $Z_{P,CORR}$ for the measured bridge sensor signal is further processed with the conditioning formula to remove offset and temperature dependency and to compensate non-linearity up to 3rd order. The signal conditioning equation is processed by the CMC and is defined as follows:

⇒ Range definition of inputs

$$Z_{P,CORR} \in \left[-2^{r_{ADC}}; 2^{r_{ADC}} \right)$$

$$Z_{CT,CORR} \in \left[-2^{r_{ADC}-1}; 2^{r_{ADC}-1} \right)$$

⇒ Conditioning Equations

$$Y = \frac{Z_{P,CORR} + c_0 + 2^{-(r_{ADC}-1)} \cdot c_4 \cdot Z_{CT,CORR} + 2^{-2(r_{ADC}-1)} \cdot c_5 \cdot Z_{CT,CORR}^2}{c_1 + 2^{-(r_{ADC}-1)} \cdot c_6 \cdot Z_{CT,CORR} + 2^{-2(r_{ADC}-1)} \cdot c_7 \cdot Z_{CT,CORR}^2}$$

$$Y \in [0; 1)$$

$$P = Y \cdot \left(1 - 2^{-15} \cdot c_2 - 2^{-15} \cdot c_3 \right) + 2^{-30} \cdot c_2 \cdot Y^2 + 2^{-45} \cdot c_3 \cdot Y^3$$

$$P \in [0; 1)$$

| | |
|---------------|---|
| r_{ADC} | Resolution of A/D conversion (13 or 14-bit) |
| $Z_{P,CORR}$ | Raw A/D conversion result for bridge sensor signal (auto-zero compensated) |
| $Z_{CT,CORR}$ | Raw A/D conversion result for calibration temperature (auto-zero compensated) |
| P | Conditioned bridge sensor signal result |

Conditioning coefficients stored in EEPROM registers 0 to 7:

$c_i \in [-2^{15}; 2^{15})$, two's complement

| | |
|-------|--|
| c_0 | Bridge offset |
| c_1 | Gain |
| c_2 | Non-linearity correction 2 nd order |
| c_3 | Non-linearity correction 3 rd order |
| c_4 | 1 st order temperature coefficient correcting bridge offset |
| c_5 | 2 nd order temperature coefficient correcting bridge offset |
| c_6 | 1 st order temperature coefficient correcting bridge gain |
| c_7 | 2 nd order temperature coefficient correcting bridge gain |

The first equation above compensates the offset and fits the gain including its temperature dependence. The non-linearity is then corrected for the intermediate result Y. The result of these equations is a non-negative value P for the measured bridge sensor signal in the range [0; 1).

Note that the conditioning coefficients c_i are positive or negative values in two's complement.

2.3 Digital Bridge Sensor Signal Filter Function

The ZSSC3170 offers a digital (averaging) filter function for the bridge sensor signal output in NOM.

The filter can be parameterized using two coefficients – the integrating coefficient CFGSF:PAVRG and the differential coefficient CFGSF:PDIFF (see Table 7.6). The filter function is implemented as follows:

⇒ **Digital Filter Function**

$$P_{OUT,i} = P_{OUT,i-1} + (P_i - P_{OUT,i-1}) \cdot \frac{PDIFF + 1}{2^{PAVRG}} \quad i > 0$$

with PAVRG, PDIFF ∈ [0; 7]
 $P_{OUT,i} \in [0; 1)$

| | |
|-------------|---|
| P_i | Conditioned bridge sensor signal result (see section 2.2) |
| $P_{OUT,i}$ | Filtered output result |
| PAVRG | Averaging filter coefficient |
| PDIFF | Differential filter coefficient |

The result of the filter function is a non-negative value P_{OUT} for the measured bridge sensor signal in the range [0; 1). This value P_{OUT} is used for generating the continuously written output value during the measurement cycle.

Note that the first output value $P_{OUT,0}$ is set equal to P_0 .

Note that setting both of the coefficients CFGSF:PAVRG and CFGSF:PDIFF to 0 disables the filter function.

2.4 Temperature Conditioning Formula

Output of the temperature value is only available in LIN mode. The digital raw value $Z_{MT,CORR}$ for the measured temperature is processed with the conditioning formula to remove offset and to compensate non-linearity up to 2nd order. The signal conditioning equation is processed by the CMC and is defined as follows:

⇒ **Range definition of inputs**

$$Z_{MT,CORR} \in [-2^{r_{ADC}-1}; 2^{r_{ADC}-1})$$

⇒ **Conditioning Equations**

$$Y_T = \frac{Z_{MT,CORR} + t_0}{t_1} \quad Y_T \in [0; 1)$$

$$T = Y_T \cdot (1 - 2^{-15} \cdot t_2) + 2^{-15} \cdot t_2 \cdot Y_T^2 \quad T \in [0; 1)$$

| | |
|--|---|
| r_{ADC} | Resolution of A/D conversion (13/14-bit) |
| $Z_{MT,CORR}$ | Raw A/D conversion result for temperature (auto-zero compensated) |
| Conditioning coefficients stored in EEPROM registers 8 to 10 | |
| $t_i \in [-2^{15}; 2^{15})$, | two's complement. |
| t_0 | Temperature offset |
| t_1 | Temperature gain |
| t_2 | Temperature non-linearity correction 2 nd order |

The first equation above compensates the offset and fits the gain. The non-linearity is then corrected for the intermediate result Y_T . The result of these equations is a non-negative value T for the measured temperature in the range [0; 1). This value T is used for generating the continuously written output value during the measurement cycle.

Note that the conditioning coefficients t_i are positive or negative values in two's complement.

3 Digital PWM Output

Digital output via the PWM interface is processed with a bridge sensor signal value that is at least 11 bits. Temperature information is not available in PWM mode.

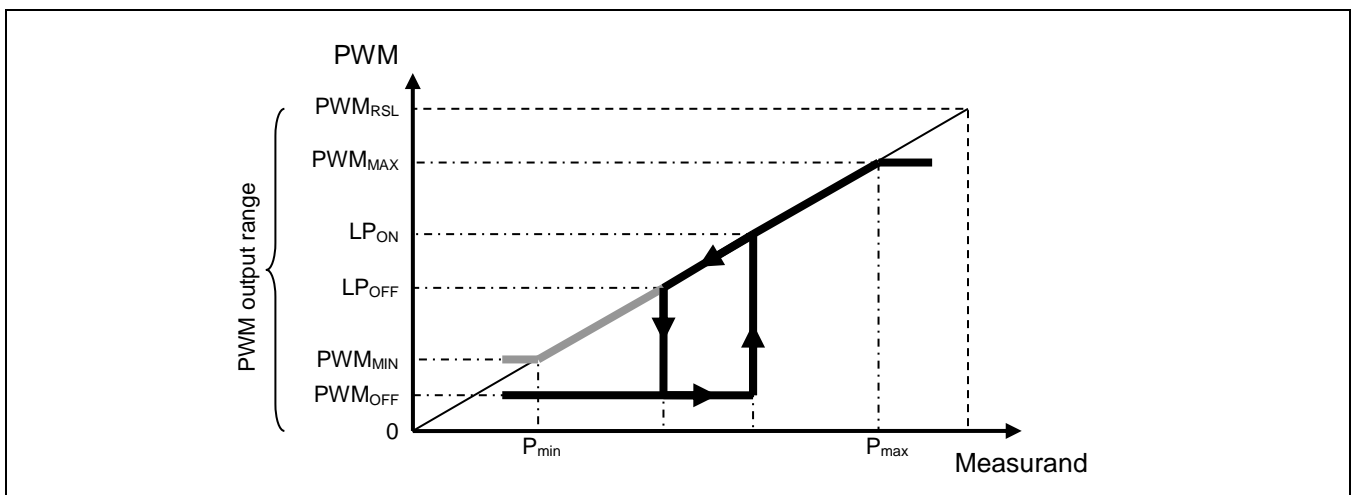
The PWM output is synchronized to the measurement cycle. The PWM period is an even-numbered multiple of the A/D conversion time. Consequently the PWM resolution depends on the selected A/D conversion. PWM_{RSL} is the maximum PWM output value. The PWM period depends on the resolution and LSB length and can be balanced by adjusting the frequency of internal oscillator f_{OSC} (see section 6.4).

Table 3.1 PWM Configuration

| A/D conversion | | | PWM | | |
|-----------------------------|------------------------|------------------------|---------------------------------------|--------------------------|---|
| Resolution CFGAFE:ADCRES | Order CFGAFE:ADCORD | Mode CFGAFE:ADCMODE | Resolution PWM_{RSL} (in LSB) | LSB (in $1/f_{osc}$) | Period @ $f_{osc} = 1.8MHz$ (in ms) |
| 14 | 2 step | 2 | 4480 | 8 | 19.9 |
| | | 3 | 4864 | 4 | 10.8 |
| | | 4 | 5632 | 2 | 6.3 |
| | | 5 | 7168 | 1 | 4.0 |
| | 1 step | - | 8448 | 8 | 37.5 |
| 13 | 2 step | 2 | 2432 | 8 | 10.8 |
| | | 3 | 2816 | 4 | 6.3 |
| | | 4 | 3584 | 2 | 4.0 |
| | | 5 | 5120 | 1 | 2.8 |
| | 1 step | - | 4352 | 8 | 19.3 |

The bridge sensor signal output value $P_{OUT} \in [0; 1)$ is normalized to the PWM period.

Figure 3.1 PWM Output Characteristics



The following sequence of normalization, limitation, hysteresis, and noise suppression is processed (see Table 7.1 for settings for PWMMIN, PWMMAX, LPOFF, LPON, PWMOFF, and ZMIN).

⇒ **PWM Output Function**

$$P_{PWM} = P_{OUT} \cdot PWM_{RSL} \quad \text{with } P_{OUT,i} \in [0; 1)$$

⇒ **Limitation (clipping)**

$$PWM_0(P_{PWM} \in [PWM_{MIN}; PWM_{MAX}]) = P_{PWM}$$

$$PWM_0(P_{PWM} > PWM_{MAX}) = PWM_{MAX}$$

$$PWM_0(P_{PWM} < PWM_{MIN}) = PWM_{MIN}$$

⇒ **Hysteresis**

$$PWM(PWM_0 < LP_{OFF}) = PWM_{OFF}$$

$$PWM(PWM = PWM_{OFF} \ \& \ PWM_0 > LP_{ON}) = PWM_0$$

⇒ **Noise Suppression**

The switch between the characteristic curve and PWMOFF (in both directions) is processed only after a number of discrete result values PWM_0 complying with hysteresis conditions.

| | |
|-------------|--|
| P_{OUT} | Bridge sensor signal output value (see section 2.2) |
| PWM_{RSL} | PWM resolution (see Table 3.1) depending on A/D conversion setup |
| P_{PWM} | Normalized PWM bridge sensor signal output |
| PWM_{MIN} | Lower PWM output value (PWMMIN) |
| PWM_{MAX} | Upper PWM output value (PWMMAX) |
| PWM_0 | Limited PWM bridge sensor signal output |
| LP_{OFF} | Low bridge sensor signal off value (LPOFF) |
| LP_{ON} | Low bridge sensor signal on value (LPON) |
| PWM_{OFF} | PWM off output value (PWMOFF) |
| PWM | PWM bridge sensor signal output |
| Z_{MIN} | Number of bridge sensor signal measurements for switch on/off noise suppression (ZMIN) |

Note that limitation can be disabled by setting PWMMIN to 0 and setting PWMMAX to greater than or equal to the PWM tick count for the application.

Note that hysteresis can be disabled by setting LP_{OFF} to a value less than or equal to PWMMIN or by setting ZMIN to 0.

Note that noise suppression can be disabled by setting Z_{MIN} to 1.

Note: LIN Sleep Mode must be disabled for proper PWM operation.

4 Digital LIN Interface

4.1 General Description

The ZSSC3170 includes a serial digital LIN interface. It allows the programming of the EEPROM to configure the application mode and to calibrate the sensor signal conditioning. During normal operation, it provides the read-out of the conditioned sensor signal and the temperature measurement.

The LIN interface implemented in the ZSSC3170 is based on the LIN Specification Package 2.1 (2006-11-24), Package 2.0 (2003-09-23), and LIN 2.1 Specification Errata Sheet (Revision 1.3, (2009-04-02).

For compatibility reasons, it includes a mode based on LIN Specification Package 1.3 (2002-12-13).

The ZSSC3170 always works as a LIN slave node.

The LIN interface is conceptually divided into two main parts: the LIN Protocol Controller and the LIN Physical Transceiver. This section describes the functionality of the LIN Protocol Controller. For the LIN Physical Transceiver, see the *ZSSC3170 LIN Interface Description*.

The ZSSC3170 LIN interface supports the following features:

- Single-wire LIN transceiver implementation
- Compatibility with LIN specification package 2.1, 2.0 and 1.3
- Bit rates: 1kbit/s up to 20kbit/s
- Fast mode with bit rates up to 80kbit/s
- Signal-based application interaction
- Re-configurability
- Transport layer and diagnostic support
- Sleep mode
- Protection against short circuits on the supply and ground
- LIN pin load dump protection (40V)
- LIN pin ESD protection 8kV

4.2 LIN Protocol

4.2.1 Frame

The entities that are transferred on the LIN bus are referred to as frames. The ZSSC3170 LIN publisher frame consists of a break field, a sync byte field, a protected identifier, 4 data bytes, and a checksum. This results in a publisher frame with a nominal length of 84 bit times (t_{Bit}). The break field, sync byte field, and protected identifier are also called the header. The data bytes and checksum are called the response.

4.2.2 Bit Rate

The supported LIN bit rate is specified in the range of 1kbit/s to 20kbit/s. The two preferred bit rates are 9.6kbit/s and 19.2kbit/s, especially for starting communication with a non-configured device. A fast mode is also supported with a bit rate up to 80kbit/s. This can be enabled by the command LIN_FAST.

4.2.3 Synchronization

Synchronization is adjusted with the sync byte field of every LIN frame. The full bit rate in LIN mode is ensured for internal oscillator frequencies adjusted to be in the range of 1.5 to 3MHz.

Oscillator frequency adjustment can be performed using the synchronization result of the LIN interface (see section 6.4).

The precision of the internal oscillator guarantees synchronization between master and slave better than or equal to $\pm 1\%$. The slave synchronizes to the sync byte field and triggers every new byte field with the falling edge of the start bit. As a result, the maximum deviation between master and slave within each byte field is less than or equal to $\pm 10\%$ according to the master clock time.

4.2.4 Bit Sampling

A byte field is synchronized at the falling edge of the start bit. There are two possible bit sampling modes, selected via CFGLIN:LINSMPL. A bit is evaluated either with 3 samples within a window between 7/16 and 9/16 of bit time or with 5 samples within a window between 6/16 and 10/16 of bit time. The preferred sampling mode is 3 samples per bit. The bit data is determined by the bit sample majority.

4.2.5 Protected Identifier (PID)

The protected identifier byte field consists of an identifier (6LSB) and parity bits (2MSB). The protected identifiers are used for signal-carrying publisher frames during NOM. These transmit 4 data byte fields carrying the transmitted signal containing bridge sensor signal and temperature result values and status information.

The publisher PID is programmed in EEPROM with 8 bits and is valid if the EEPROM signature is valid. Note: PID parity information is not checked. It can be changed by transport layer communication even in normal operation.

The ZSSC3170 supports transport layer communication using reserved identifiers 60 (PID 3C_{HEX}) and 61 (PID 7D_{HEX}). A non-configured device due to an invalid EEPROM signature subscribes and publishes only to transport layer frames. This must be used to define a valid configuration. See section 4.5 for details. Transport layer frames contain 8 data byte fields.

4.2.6 Checksum

The checksum is defined as the inverted 8-bit sum with carry. It is calculated including all data bytes (classic checksum) or including all data bytes and the protected identifier (enhanced checksum). The classic checksum is used for publisher frames according to LIN 1.3 and for transport layer frames. The enhanced checksum is used for publisher frames according to LIN 2.1 and LIN 2.0. The checksum type is selected via CFGLIN:LINMODE.

Commands received via master request frame are only processed if a valid checksum is detected.

4.3 LIN Publisher Frame – Normal Operation Mode (NOM)

During NOM, LIN communication is used to read out the conditioned bridge sensor and temperature sensor signal. Therefore the LIN master initiates a publisher frame by generating the header. The response is published by the ZSSC3170 slave node. The publisher PIDs are programmed in EEPROM register 18_{HEX}. NOM requires that the ZSSC3170 is configured (i.e., the EEPROM signature is valid).

There are several signal formats for publisher frames according to PID1 configured by CFGLIN:LINCFGFRM. The publisher frame according to PID2 has a fixed format and can be used specifically to read 15-bit raw measurement values during calibration.

Figure 4.1 LIN Publisher Frame

| Byte Field | Bit 0 LSB | | Signal Carrying Data Bytes | | | | Bit 31 MSB | | Checksum |
|----------------|----------------------------|---|---------------------------------|---|-------------------------|---------------------------|---------------------------|--|----------|
| | Header | Data Byte 0 | Data Byte 1 | Data Byte 2 | Data Byte 3 | | | | |
| Frame 0 NOM | Includes Publisher PID1 | Bridge sensor signal LSB (12-bit) MSB | Temperature LSB (12-bit) MSB | | Error status (4-bit) | Variant ID (4-bit) | Enhanced or classic | | |
| Frame 1 NOM | Includes Publisher PID1 | Bridge sensor signal LSB (12-bit) MSB | Temperature LSB (10-bit) MSB | | Error status (2) | Variant ID (8-bit) | Enhanced or classic | | |
| Frame 2 NOM | Includes Publisher PID1 | Bridge sensor signal LSB (12-bit) MSB | Temperature LSB (8-bit) MSB | Error status (4-bit) | Variant ID (8-bit) | Enhanced or classic | | | |
| Frame 3 NOM | Includes Publisher PID1 | Bridge sensor signal LSB (11-bit) MSB | Temperature LSB (9-bit) MSB | Error status (4-bit) | Variant ID (8-bit) | Enhanced or classic | | | |
| Frame 4 NOM | Includes Publisher PID2 | Bridge sensor signal LSB (14-bit) MSB | P error status (2) | Temperature LSB (13-bit) MSB | T error status (2) | Response error (1) | Enhanced or classic | | |
| Frame 5 CM | Includes Publisher PID2 | Raw value (two's complement) LSB (15-bit) MSB | 1 (1) | 0000 _{HEX} LSB (14-bit) MSB | 1 (1) | Response error (1) | Enhanced or classic | | |
| | Master | Slave | | | | | | | |

Signals, including the bridge sensor and temperature values, are sent with LSB first.

Bridge sensor and temperature values are limited to the values stored in EEPROM address 0B_{HEX} to 0D_{HEX}. Limitation is always based on the 12-bit internal values independent of the actual resolution of the values output in the frame.

Error notification is transmitted with up to 4 status bits. See section 4.4 for a description. Bridge sensor and temperature values are transmitted even if an error status bit is set. Error status must be evaluated by the LIN master to assess the validity of bridge sensor and temperature values.

The variant ID is stored in EEPROM register 16_{HEX} (low byte).

The enhanced checksum is used in LIN mode 2.0/2.1, and the classic checksum is used in LIN mode 1.3.

4.4 LIN Slave Status Information

There are up to 4 status bits transferred in publisher frames during NOM to indicate the status of the LIN slave. The set of status bits to be used is configured by CFGLIN:LINCFGFRM (see section 7.3).

Table 4.1 LIN Status Bits Transmitted in Normal Operation Mode (NOM)

| LIN Status Bit | Description |
|--------------------------------------|--|
| Bridge Sensor Error | Bridge sensor value is out of defined limits. Limits are defined in EEPROM registers B _{HEX} and D _{HEX} . <i>OR</i> Any of the following internal error detections has indicated a failure: sensor connection or short check (SCC/SSC); sensor-aging check (SAC); calibration temperature out-of-range check; watchdog; arithmetic check; RAM parity; register parity; EEPROM error; ROM signature. |
| Bridge Sensor Signal Out-of-Limits | Bridge sensor value is out of defined limits. Limits are defined in EEPROM registers B _{HEX} and D _{HEX} . |
| Temperature Error | Temperature value is out of defined limits. Limits are defined in EEPROM registers C _{HEX} and D _{HEX} . <i>OR</i> Any of following internal error detections has indicated a failure: temperature out-of-range check; watchdog; arithmetic check; RAM parity; register parity; EEPROM error; ROM signature. |
| Temperature Out-of-Limits | Temperature value is out of defined limits. Limits are defined in EEPROM registers C _{HEX} and D _{HEX} . |
| Bridge sensor or Temperature Invalid | Any of following internal error detections has indicated a failure: sensor connection or short check (SCC/SSC); sensor aging check (SAC); calibration temperature out-of-range check; temperature out of range check; watchdog; arithmetic check; RAM parity; register parity; EEPROM error; ROM signature. |
| Response Error | The response error is announced if a checksum error in the subscriber frame is detected; a frame error in the subscriber frame is detected (byte field start and stop bit); or a bit error in the publisher frame is detected. |

| LIN Status Bit | Description |
|-----------------|--|
| Checksum Error | A checksum error in a subscriber frame has been detected. Error notification is low-pass filtered: the failure counter is incremented by 8 up to 63 and decremented by 1; the status bit is set at 63 and reset at 0. |
| Bit Error | A bit error in a publisher frame has been detected. Error notification is low-pass filtered: the failure counter is incremented by 8 up to 63 and decremented by 1; the status bit is set at 63 and reset at 0. |
| Value Refreshed | Bridge sensor or temperature value has not been sent since last update. |

Also see section 1.3.3 for a detailed description of behavior in the Diagnostic Mode.

4.5 LIN Command Mode

The ZSSC3170 allows end-of-line configuration and calibration via one-wire LIN communication. The ZSSC3170 functions as a LIN slave.

When using LIN communication, the Command Mode starts after the ZSSC3170 receives the command `START_CM` from the master via the transport layer service *Data-dump* (see section 4.6.8). The ZSSC3170's internal measurement cycle is stopped, and it waits for further commands. The full command set (section 6.1) is available.

If LIN mode is configured (`CFGAPP:PWMENA = 0`), the master request is received via the LIN pin. Starting Command Mode is always available in LIN mode. Alternatively, in NOM, the bridge sensor and temperature values can be read using the publisher PIDs.

If PWM mode is configured (`CFGAPP:PWMENA = 1`), the master request must be received during the start window via the PWM output pins HOUT or LOUT. During the start window, the recessive level is applied to both pads. The duration of the start window is 30ms (nominal). If the start window expires without receiving the `START_CM`, the PWM output (NOM) begins. After the start window, LIN communication via the HOUT or LOUT pins is no longer possible.

4.6 LIN Transport Layer

LIN transport layer is used for diagnostics and for configuration of the ZSSC3170. The ZSSC3170 supports Diagnostic Class I according to the LIN Specification Package 2.1.

The transport layer has fixed frame IDs. Messages issued by a master are called master requests and use the ID 60 (PID 3C_{HEX}). Messages issued by the slave are called slave requests and use the frame ID 61 (PID 7D_{HEX}).

The ZSSC3170 transport layer supports only single frames containing the node address byte (NAD); protocol control information byte (PCI); service identifier byte (SID) or response service identifier byte (RSID); 5 additional data bytes; and classic checksum byte. Unused bytes must be filled with the recessive level (FF_{HEX}).

NAD: The node address (NAD) uniquely identifies a slave node. NAD values are in the range of 0 to 127.

NAD 127 is reserved as the broadcast NAD addressing all connected nodes.

NAD 126 is reserved for functional requests in LIN 2.1.

NAD 0 is reserved for the go-to-sleep master request.

ZSSC3170 handles two NADs – the initial NAD and a configured NAD. The initial NAD and the LIN Product Identification form the node identity and are not changeable via LIN diagnostic services. The configured NAD must be identical to the initial NAD after manufacturing but can be changed using LIN master requests *Assign-NAD* (see section 4.6.1) or *Conditional-Change-NAD* (see section 4.6.2).

A non-configured ZSSC3170 must be addressed using broadcast NAD 127. If the EEPROM signature is valid, the NADs programmed in EEPROM register 17_{HEX} are used.

PCI: The ZSSC3170 transport layer supports only single frames. Therefore the protocol control information (PCI) is equal to the number of data bytes used plus one (for SID or RSID). Frames with inconsistent length information are ignored.

SID: The service identifier (SID) specifies the request that will be performed by the slave node addressed. See the following sections for a detailed description of supported services.

RSID: The response service identifier (RSID) specifies the content of the response. A positive response is indicated by SID + 40_{HEX}. A negative response is indicated by 7F_{HEX} and is followed by the error code.

4.6.1 Assign-NAD

The *Assign-NAD* service is used to set a configured NAD to resolve conflicting NADs in a LIN cluster.

The initial NAD is stored in the EEPROM register 17_{HEX} high byte; the configured NAD is stored in the EEPROM register 17_{HEX} low byte.

The master request frame contains the initial NAD, Supplier ID and Function ID. For identification, each of these can be replaced by wildcard values.

Table 4.2 Assign-NAD Request

| NAD | PCI | SID | D1 | D2 | D3 | D4 | D5 |
|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|--------------------------|
| Initial NAD | 06 _{HEX} | B0 _{HEX} | Supplier ID | | Function ID | | New configured NAD |
| Broadcast | | | LSB | MSB | LSB | MSB | |
| 7F _{HEX} | | | Wildcards | | | | |
| | | | FF _{HEX} | 7F _{HEX} | FF _{HEX} | FF _{HEX} | |

A positive response is generated if transferred IDs match internally stored IDs.

Table 4.3 Assign-NAD Positive Response

| NAD | PCI | RSID | D1 | D2 | D3 | D4 | D5 |
|-------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Initial NAD | 01 _{HEX} | F0 _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} |

Note that the positive response uses the initial NAD.

If the initial NAD or transferred IDs do not match the stored values, no response is sent.

Note that NADs and IDs are internally set to wildcard values if the EEPROM signature is not valid.

Note that new configured NAD is not permanent initially. Use the LIN master request *Save-Configuration* (see section 4.6.6) to store the configured NAD to EEPROM.

4.6.2 Conditional-Change-NAD

Conditional-Assign-NAD service is used to resolve conflicting NADs in a LIN cluster. The configured NAD is stored in EEPROM register 17_{HEX} low byte.

The master request frame contains an identifier that defines which internal identification number the condition is related to. Supported identifiers are listed in Table 4.8; e.g., identifier 0 means Supplier ID, Function ID and Variant ID and identifier 1 means the Serial Number.

The master request frame also contains a byte identifier that defines which byte of the chosen identification number the condition is related to.

The master request frame contains a mask and an invert byte at the end. It defines the condition. The selected byte of the internal identification number is first bitwise XORed with the invert byte and then bitwise ANDed with the mask. If the result is zero, then the configured NAD is changed.

Table 4.4 Conditional-Change-NAD Request

| NAD | PCI | SID | D1 | D2 | D3 | D4 | D5 |
|-------------------|-------------------|-------------------|------------|------|------|--------|--------------------|
| Configured NAD | 06 _{HEX} | B3 _{HEX} | Identifier | Byte | Mask | Invert | New configured NAD |
| Broadcast | | | | | | | |
| 7F _{HEX} | | | | | | | |

A positive response is generated if the condition is successfully evaluated as zero.

Table 4.5 Conditional-Change-NAD Positive Response

| NAD | PCI | RSID | D1 | D2 | D3 | D4 | D5 |
|--------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| New configured NAD | 01 _{HEX} | F3 _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} |

Note that the positive response uses the new configured NAD.

If initial NAD does not match the stored value or if the condition is not successfully evaluated, no response is sent.

Note that NADs and IDs are internally set to wildcard values if the EEPROM signature is not valid, in which case, the Serial Number is set to FFFF_{HEX}.

Note that new configured NAD is not permanent initially. Use the LIN master request *Save-Configuration* to store the configured NAD to EEPROM.

4.6.3 Read-By-Identifier

The *Read-by-Identifier* service is used to read out the LIN slave node properties.

The master request frame contains the Supplier ID and Function ID. For identification, both of these can be replaced by wildcards.

The LIN Product Identification is stored in EEPROM registers 14_{HEX} (Supplier ID) and 15_{HEX} (Function ID).

Table 4.6 Read-By-Identifier Request

| NAD | PCI | SID | D1 | D2 | D3 | D4 | D5 |
|-------------------|-------------------|-------------------|------------|-------------------|-------------------|-------------------|-------------------|
| Configured NAD | 06 _{HEX} | B2 _{HEX} | Identifier | Supplier ID | | Function ID | |
| Broadcast | | | | LSB | MSB | LSB | MSB |
| 7F _{HEX} | | | | Wildcards | | | |
| | | | | FF _{HEX} | 7F _{HEX} | FF _{HEX} | FF _{HEX} |

The requested property is specified by an identifier.

Table 4.7 Read-By-Identifier - Supported Identifiers

| Identifier | Service | Response Data | | | |
|--|--|---|---|---|-----------|
| 0 | <i>Read-by-Identifier-0:</i> LIN Product Identification | Supplier ID, Function ID, Variant ID (EEPROM register 14 _{HEX} to 16 _{HEX}) | | | |
| 1 | <i>Read-by-Identifier-1:</i> Serial Number | Serial Number (EEPROM registers 1C _{HEX} to 1D _{HEX}) | | | |
| 16 | <i>Read-by-Identifier-16:</i> Message ID 1 and PID1 | Message ID 1 (EEPROM register 19 _{HEX}), Publisher ID 1 (EEPROM register 18 _{HEX} , low byte) (Service is specified in LIN 2.0. It is also supported in LIN 2.1.) | | | |
| 17 | <i>Read-by-Identifier-17:</i> Message ID 2 and PID2 | Message ID 2 (EEPROM register 1A _{HEX}), Publisher ID 2 (EEPROM register 18 _{HEX} , high byte) (Service is specified in LIN 2.0. It is also supported in LIN 2.1.) | | | |
| 32 | <i>Read-by-Identifier-32:</i> Diagnostic Mode Error Code | Internal 16-bit error code (see Table 1.1) (If no error code is available, data bytes are set to 0000 _{HEX} .) | | | |
| 33 | <i>Read-by-Identifier-33:</i> Bit Rate | Frequency ratio = $2 f_{OSC} / f_{LIN} - 2$ where f_{OSC} is the internal oscillator frequency, and f_{LIN} is LIN frequency | | | |
| 34 | <i>Read-by-Identifier-34:</i> LIN Output Buffer (2MSB of high bytes are error status!) | Internal 32-bit Response Buffer | | | |
| | | D1 | D2 | D3 | D4 |
| | Normal Operation Mode | 2MSB.. T value invalid T value out-of-limits 14 bit .. Temperature value LSB MSB | | 2MSB.. P value invalid P value out-of-limits 14 bit .. Bridge sensor value LSB MSB | |
| Command Mode STRT_AD_X Read raw measurement values. | 8000 _{HEX} LSB MSB | | MSB .. set to 1'b1 15 bit .. measured raw value (two's complement) LSB MSB | | |
| 35 | <i>Read-by-Identifier-35:</i> Command Response Buffer | Internal 32-bit Slave Response Buffer | | | |
| | | D1 | D2 | D3 | D4 |
| | Normal Operation Mode | n/a | n/a | n/a | n/a |
| | Command Mode | Data high byte | Data low byte | Check sum | cmd |
| | Command Mode STRT_AD_X | cmd | C3 _{HEX} | cmd | Check sum |
| Command Mode READ_EEP_RAW | 00 _{HEX} | 6-bit parity | Data high byte | Data low byte | |

A positive response is generated if transferred IDs match internally stored IDs. If EEPROM signature is not valid, the ZSSC3170 only responds to ID wildcard values.

Table 4.8 Read-By-Identifier Positive Response

| ID | NAD | PCI | RSID | D1 | D2 | D3 | D4 | D5 |
|----|-----|-------------------|-------------------|----------------------|---------------|--------------------|-------------------|-----------------------|
| 0 | NAD | 06 _{HEX} | F2 _{HEX} | Supplier ID LSB | MSB | Function ID LSB | MSB | Variant ID LSB MSB |
| 1 | NAD | 05 _{HEX} | F2 _{HEX} | Serial Number LSB | MSB | | | FF _{HEX} |
| 16 | NAD | 04 _{HEX} | F2 _{HEX} | Message ID 1 LSB | MSB | PID1 LSB MSB | FF _{HEX} | FF _{HEX} |
| 17 | NAD | 04 _{HEX} | F2 _{HEX} | Message ID 2 LSB | MSB | PID2 LSB MSB | FF _{HEX} | FF _{HEX} |
| 32 | NAD | 03 _{HEX} | F2 _{HEX} | Error code LSB | MSB | FF _{HEX} | FF _{HEX} | FF _{HEX} |
| 33 | NAD | 03 _{HEX} | F2 _{HEX} | Bit Rate LSB | MSB | FF _{HEX} | FF _{HEX} | FF _{HEX} |
| 34 | NAD | 05 _{HEX} | F2 _{HEX} | D1 LSB MSB | D2 LSB MSB | D3 LSB MSB | D4 LSB MSB | FF _{HEX} |
| 35 | NAD | 05 _{HEX} | F2 _{HEX} | D1 LSB MSB | D2 LSB MSB | D3 LSB MSB | D4 LSB MSB | FF _{HEX} |

A negative response is generated if an unknown identifier is requested.

Table 4.9 Read-By-Identifier Negative Response

| NAD | PCI | RSID | D1 | D2 | D3 | D4 | D5 |
|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Configured NAD | 03 _{HEX} | 7F _{HEX} | B2 _{HEX} | 12 _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} |

If the NAD or transferred IDs do not match the stored values, no response is sent.

4.6.4 Assign-Frame-Identifier

Assign-Frame-Identifier service is used to set or disable one PID.

Note that *Assign-Frame-Identifier* service is specified in LIN 2.0 only. For setting PIDs according to LIN 2.1 see *Assign-Frame-Identifier-Range*.

Two publisher PIDs are stored in EEPROM register 18_{HEX}. The message ID of publisher PID1 (low byte) is stored in EEPROM register 19_{HEX}. The message ID of publisher PID2 (high byte) is stored in EEPROM register 1A_{HEX}.

Master request frame contains the Supplier ID and Message ID. For identification, both of these can be replaced by wildcards.

Table 4.10 Assign-Frame-Identifier Request

| NAD | PCI | SID | D1 | D2 | D3 | D4 | D5 |
|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|--------------------|
| Configured NAD | 06 _{HEX} | B1 _{HEX} | Supplier ID | | Message ID | | New PID LSB MSB |
| Broadcast | | | LSB | MSB | LSB | MSB | |
| 7F _{HEX} | | | Wildcards | | Wildcards | | |
| | | | FF _{HEX} | 7F _{HEX} | FF _{HEX} | FF _{HEX} | |

PID 00_{HEX} disables the connected frame.

A positive response is generated if transferred IDs match internally stored IDs. The new PID is assigned.

Table 4.11 Assign-Frame-Identifier Positive Response

| NAD | PCI | RSID | D1 | D2 | D3 | D4 | D5 |
|----------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Configured NAD | 01 _{HEX} | F1 _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} |

If the NAD or transferred IDs do not match the stored values, no response is sent.

Note that new PID is not permanent initially. Use the LIN master request *Save-Configuration* to store the new PID to EEPROM.

4.6.5 Assign-Frame-Identifier-Range

The *Assign-Frame-Identifier-Range* service is used to set or disable PIDs.

Note that the *Assign-Frame-Identifier-Range* service is specified in LIN 2.1. For setting PIDs according to LIN 2.0, see *Assign-Frame-Identifier*.

Two publisher PIDs are stored in EEPROM register 18_{HEX}.

The master request frame contains a start index (set to 00_{HEX}) and 4 PIDs. The first and second PID are assigned to publisher frame PID1 and PID2; the next 2 PIDs are not supported and must be set to “do not care” (FF_{HEX}).

Table 4.12 Assign-Frame-Identifier-Range Request

| NAD | PCI | SID | D1 | D2 | D3 | D4 | D5 |
|-------------------|-------------------|-------------------|---------------------------|---------------------|---------------------|-------------------|-------------------|
| Configured NAD | 06 _{HEX} | B7 _{HEX} | index = 00 _{HEX} | New PID1 LSB MSB | New PID2 LSB MSB | FF _{HEX} | FF _{HEX} |
| Broadcast | | | | Do not care | Do not care | | |
| 7F _{HEX} | | | | FF _{HEX} | FF _{HEX} | | |

PID 00_{HEX} disables the connected frame. PID FF_{HEX} does not change the stored PID.

A positive response is generated if all transferred PIDs can be assigned.

Table 4.13 Assign-Frame-Identifier-Range Positive Response

| NAD | PCI | RSID | D1 | D2 | D3 | D4 | D5 |
|----------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Configured NAD | 01 _{HEX} | F7 _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} |

If the NAD does not match the stored value or if a transferred PID could not be assigned, no response is sent.

Note that new PIDs are not permanent initially. Use LIN master request *Save-Configuration* to store new PIDs to EEPROM.

4.6.6 Save-Configuration

Save-Configuration service is used to initiate the slave node to save its configuration into EEPROM.

Upon receiving the master request *Save-Configuration*, the ZSSC3170 stores the LIN configuration (configured NAD, PIDs) into EEPROM registers 17_{HEX} and 18_{HEX}. The new signature is also evaluated and stored in EEPROM register 1B_{HEX}.

Important Note: *Save-Configuration* interrupts the normal operation measurement cycle and initiates 3 EEPROM programming cycles. Including restarting the measurement cycle, it takes a processing time of 50ms.

The *Save-Configuration* service must be enabled by CFGLIN:LINSVCFG.

ZSSC3170 does *not* check the EEPROM lock bit (CFGSF:EEPLOCK).

Table 4.14 Save-Configuration Request

| NAD | PCI | SID | D1 | D2 | D3 | D4 | D5 |
|----------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Configured NAD | 01 _{HEX} | B6 _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} |

A positive response is generated if saving the configuration has started. It does not wait until programming is finished but is sent immediately if the slave response frame is recognized.

Table 4.15 Save-Configuration Positive Response

| NAD | PCI | RSID | D1 | D2 | D3 | D4 | D5 |
|----------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Configured NAD | 01 _{HEX} | F6 _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} |

A negative response is generated if *Save-Configuration* service is disabled by CFGLIN:LINSVCFG.

Table 4.16 Save-Configuration Negative Response

| NAD | PCI | RSID | D1 | D2 | D3 | D4 | D5 |
|----------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Configured NAD | 03 _{HEX} | 7F _{HEX} | B6 _{HEX} | 11 _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} |

4.6.7 Go-To-Sleep

The *Go-To-Sleep* service is used to set the LIN cluster into sleep mode.

NAD 00_{HEX} is reserved for this service. No further relevant data is sent. The slave node ignores the subsequent byte fields but evaluates the transferred checksum.

Table 4.17 Go-To-Sleep Request

| NAD | PCI | SID | D1 | D2 | D3 | D4 | D5 |
|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| 00 _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} |

No response is generated because slave node is set to sleep mode.

4.6.8 Data-Dump

The *Data-Dump* service is used to configure and calibrate the ZSSC3170 slave node. This service must be used only by supplier diagnostics and not in a running cluster during the application.

The signal format is ZSSC3170-specific. It makes available the complete command set defined in section 6. The master request contains the command byte and up to 2 optional data bytes. Data bytes that are not used must be filled with FF_{HEX}. Note that PCI is always 05_{HEX}.

Commands are only processed if the Command Mode has been entered previously. Therefore the command START_CM is sent first even using *Data-Dump* service. After changing to Command Mode all other commands are available.

Table 4.18 Data-Dump Request

| NAD | PCI | SID | D1 | D2 | D3 | D4 | D5 |
|-------------------|-------------------|-------------------|--------------------|-------------------|-------------------|-------------------|-------------------|
| NAD | 05 _{HEX} | B4 _{HEX} | Command LSB MSB | FF _{HEX} | FF _{HEX} | FF _{HEX} | FF _{HEX} |
| Broadcast | | | Command LSB MSB | Data1 LSB MSB | Data2 LSB MSB | FF _{HEX} | FF _{HEX} |
| 7F _{HEX} | | | | | | | |

A positive response is generated if the Command Mode is set. The response always contains 4 data bytes. See section 6.3 for the content of response data.

Table 4.19 Data-Dump Positive Response

| NAD | PCI | RSID | D1 | D2 | D3 | D4 | D5 |
|-----|-------------------|-------------------|------------------|------------------|------------------|------------------|-------------------|
| NAD | 05 _{HEX} | F4 _{HEX} | Data1 LSB MSB | Data2 LSB MSB | Data3 LSB MSB | Data4 LSB MSB | FF _{HEX} |

If ZSSC3170 is not in Command Mode, no response is sent.

4.7 LIN Sleep Mode

The ZSSC3170 supports LIN sleep mode functionality if enabled by configuration bit CFGLIN:LINSLP.

There are two conditions that initiate the ZSSC3170 switching to sleep mode:

- LIN bus inactivity (recessive or dominant level) for more than 4s. Sleep mode is achieved within 10s.
- Receipt of the master request frame *Go-To-Sleep*.

In sleep mode, the LIN node retains its configuration but does not response to LIN communication. The measurement cycle is stopped.

A wake up signal is issued by forcing the bus to a dominant state for at least 250 μ s. ZSSC3170 detects a dominant state longer than 150 μ s. It starts initialization from RAM and then the measurement cycle. Wake up time depends on configuration and is less than 50ms.

ZSSC3170 slave node is not able to generate the wake up signal by itself.

Note: LIN Sleep Mode must be disabled for proper PWM operation.

4.8 Differences between LIN Mode 1.3 and LIN Mode 2.0/2.1 – Summary

Active LIN mode is selected via CFGLIN:LINMODE. It can be set to LIN mode 1.3, 2.0, or 2.1.

The differences between available LIN modes are shown in Table 4.20.

Table 4.20 LIN modes

| CFGLIN: LINMODE | LIN Mode | Publisher Frame Checksum type | NAD 0x7E | Slave Response Frame Timeout |
|--------------------|----------|----------------------------------|--------------------------|---------------------------------|
| 00 _{BIN} | LIN 1.3 | Classic check sum | Normal NAD | Not available |
| 01 _{BIN} | LIN 2.0 | Enhanced check sum | Normal NAD | Not available |
| 10 _{BIN} | LIN 2.1 | Enhanced check sum | Functional NAD (ignored) | 1000ms |
| 11 _{BIN} | LIN 2.1 | Enhanced check sum | Functional NAD (ignored) | 1000ms |

5 Serial Digital I²C Interface

5.1 General Description

The ZSSC3170 includes a serial digital I²C interface. The I²C allows programming the EEPROM to configure the application mode for the ZSSC3170 and to calibrate the conditioning equation. It also provides the read out of the conditioning results as a digital value. The ZSSC3170 always works as the slave.

The I²C interface is always enabled after power-on and a short initialization phase independent of programmed configuration. In Normal Operation Mode (NOM), the result values for the bridge sensor signal and for the temperature can be read out. In Command Mode (CM), all calibration commands are available. Note that Command Mode requires transmitting a command START_CM to enable processing calibration commands. EEPROM access via I²C is always available after this command. The EEPROM lock bit only affects EEPROM access via LIN. Therefore EEPROM is always programmable via I²C. In Diagnostic Mode (DM), an error code is available to identify the error source.

A command consists of a device address byte and a command byte. Some commands (e.g., writing data into EEPROM) also include two data bytes. Refer to section 1.3 for details of working modes and section 6 for command descriptions.

There are two general methods/requests for reading data via the I²C from the ZSSC3170:

- Digital read out
 - ➔ (Continuously) reading the conditioning result in Normal Operation Mode (NOM)

During the measurement cycle, the ZSSC3170 transfers the conditioning results for the bridge sensor signal and temperature into the output register of the I²C. These data will be sent if the master generates a read-request. The active measurement cycle is not interrupted by this.

- Calibration and/or configuration tasks
 - ➔ Reading internal data (e.g. EEPROM content) or acquired measurement data in Command Mode (CM)

To read internal and/or measurement data from the ZSSC3170 in CM, normally a specific command must be sent to transfer this data into the output register of the I²C. Thereafter the data will be sent if the master generates a read-request.

After generating the start condition the master sends the device address byte containing a 7-bit address followed by a data direction bit (R/W). A '0' indicates a transmission from master to slave (WRITE); a '1' indicates a data read-request (READ). During a read-request, the data bytes are repeatedly transmitted until the master sends a stop condition.

The ZSSC3170 general slave address is 78_{HEX} (7 bit). The addressed slave answers with an acknowledge; all other slaves connected to the I²C bus ignore this communication. Via EEPROM configuration (address 18_{HEX}), it is possible to activate an additional slave address with an available range of 40_{HEX} to 7F_{HEX} to a single device. In this case, the ZSSC3170 recognizes communication on both device addresses, on the general one and on the additional one.

Both available protocols, I²C and LIN, are active in parallel, but only one of them can be used at time.

5.2 Digital I²C Output

A read request is answered by transmitting data from the I²C output registers.

During the continuous measurement cycle (NOM, Temporary DM), digital output via I²C interface is done with 13-bit bridge sensor and 13-bit temperature values. The MSB carries an even-parity (PAR) bit. 2 bits for the Diagnostic Status (ERR) are included.

Figure 5.1 I²C Read Request during NOM, Temporary DM

| Byte | Device Address | | Bridge Sensor Signal | | | | Temperature | | | |
|-------|-------------------|-------------------------|---------------------------|--------------------------|--|--|---------------------------|--------------------------|--|--|
| | | | High Byte | | Low Byte | | High Byte | | Low Byte | |
| Value | 78 _{HEX} | R/W 1 _{BIN} | PAR 0/1 _{BIN} | ERR 00 _{BIN} | Bridge sensor (13-bit) MSB LSB | | PAR 0/1 _{BIN} | ERR 00 _{BIN} | Temperature (13-bit) MSB LSB | |

During Steady Diagnostic Mode (DM), i.e., when an error has been detected, the ERR status bits of each I²C 16-bit output word are used as the error identifier.

Figure 5.2 I²C Read Request after Detecting an Error (Steady DM)

| Byte | Device Address | | Error Code | | | | Error Code | | | |
|-------|-------------------|-------------------------|---------------------------|------------|--|--|---------------------------|------------|--|--|
| | | | High Byte | | Low Byte | | High Byte | | Low Byte | |
| Value | 78 _{HEX} | R/W 1 _{BIN} | PAR 0/1 _{BIN} | ERR err | Error code MSB LSB | | PAR 0/1 _{BIN} | ERR err | Error code MSB LSB | |

In Command Mode (CM) a 16-bit answer is sent (2 bytes). A check sum is added (1 byte) followed by the command which is answered. The check sum and the returned command allow verification of received data by the master. For details and exceptions, see section 6.2.

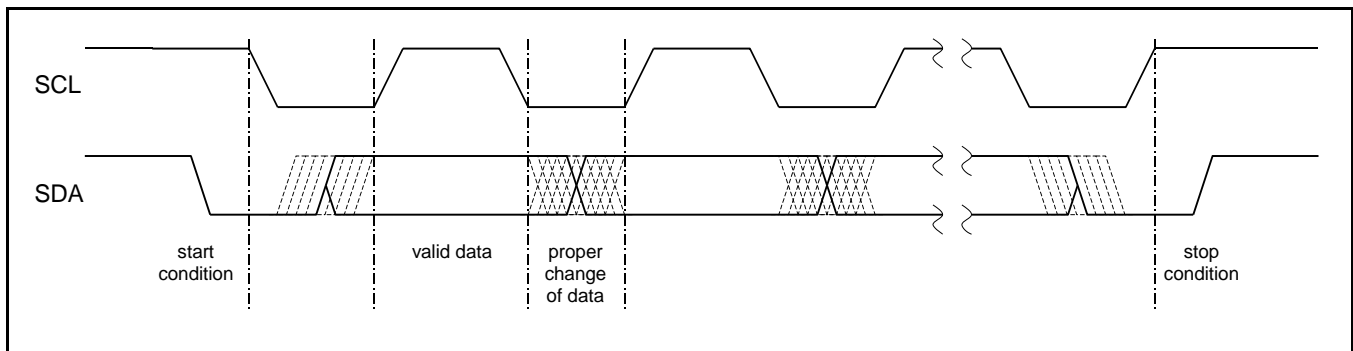
Figure 5.3 I²C Read Request Answering a Command (CM)

| Byte | Device Address | | Answer | | | | Verification | | | |
|-------|-------------------|-------------------------|---|--|----------|--|---|--|---|--|
| | | | High Byte | | Low Byte | | High Byte | | Low Byte | |
| Value | 78 _{HEX} | R/W 1 _{BIN} | Response (16-bit) MSB LSB | | | | Check sum (8-bit) MSB LSB | | Responded command (8-bit) MSB LSB | |

5.3 I²C Protocol

For I²C communication, a data line (SDA) and a clock line (SCL) are required.

Figure 5.4 Principles of I²C Protocol



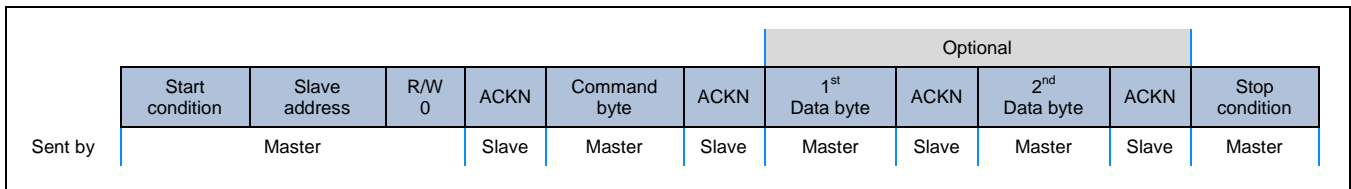
The I²C communication and protocol used is defined as follows:

- Idle period
When the bus is inactive, SDA and SCL are pulled-up to supply voltage VDDA.
- Start condition
A high-to-low transition on SDA while SCL is at the high level indicates a start condition. Every command must be initiated by a start condition sent by a master. A master can always generate a start condition.
- Stop condition
A low-to-high transition on SDA while SCL is at the high level indicates a stop condition. A command must be closed by a stop condition for the ZSSC3170 to start processing the command routine. The ZSSC3170 changes to inactive interface mode during processing of internal command routines started by a previously sent command.
- Valid data
Data is transmitted in bytes (8 bits) starting with the most significant bit (MSB). Each byte transmitted is followed by an acknowledge bit. Transmitted bits are valid if after a start condition SDA maintains a constant level during a high period of SCL. The SDA level must change only when the clock signal at SCL is low.
- Acknowledge
An acknowledge after a transmitted byte is obligatory. The master must generate an acknowledge-related clock pulse. The receiver (slave or master) pulls-down the SDA line during the acknowledge clock pulse. If no acknowledge is generated by the receiver, a transmitting slave will become inactive. A transmitting master can abort the transmission by generating a stop condition and can then repeat the command.
A receiving master must signal the end of transfer to the transmitting slave by not generating an acknowledge and afterwards transmitting a stop condition.

- Write operation

During transmission from master to slave (WRITE), the device address byte, which includes a data direction bit set to '0' (WRITE), is followed by a command byte and up to 2 optional data bytes, depending on the transmitted command. The internal microcontroller evaluates the received command and processes the related routine.

Figure 5.5 Write Operation I²C



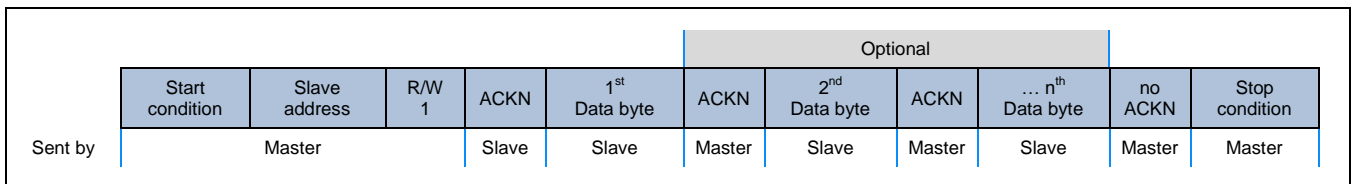
- Read operation

After the master data requests data from the slave by sending a device address byte including a data-direction bit set to '1' (READ), the slave answers by sending data from the interface output registers. The master must generate the transmission clock on SCL, acknowledges after each data byte (except after the last one), and then the stop condition.

A data request is answered by the interface module itself and consequently does not interrupt the current process of the internal microcontroller.

The data in the output registers is sent continuously until an acknowledge is not received or a stop condition is detected. After transmitting all available data, the slave starts repeating the data.

Figure 5.6 Read Operation I²C – (Data Request)



During operation, measurement cycle data is constantly updated with conditioning results. To get other data from the slave (e.g. EEPROM content), a specific command must be sent before the data request to initiate the transfer of this data to the interface output registers. This command does interrupt the current process of the internal microcontroller; e.g., the active measurement cycle.

Figure 5.7 Timing I²C Protocol

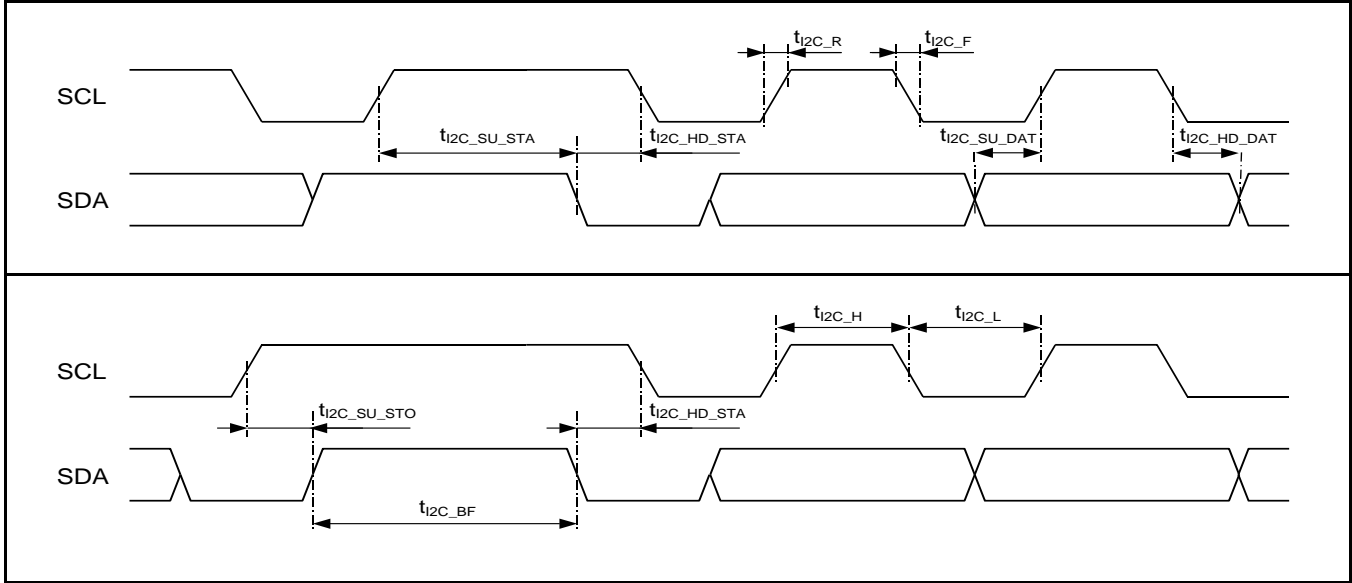


Table 5.1 Timing I²C Protocol

| Nr. | Parameter | Symbol | min | typ | Max | Unit | Conditions |
|-----|--|--------------------|-----|-----|-----|---------------|----------------------------|
| 1 | SCL Clock frequency | f_{SCL} | | | 400 | kHz | $f_{osc} \geq 2\text{MHz}$ |
| 2 | Bus free time between start and stop condition | t_{12C_BF} | 1.3 | | | μs | |
| 3 | Hold time start condition | $t_{12C_HD_STA}$ | 0.6 | | | μs | |
| 4 | Setup time repeated start condition | $t_{12C_SU_STA}$ | 0.6 | | | μs | |
| 5 | Low period SCL/SDA | t_{12C_L} | 1.3 | | | μs | |
| 6 | High period SCL/SDA | t_{12C_H} | 0.6 | | | μs | |
| 7 | Data hold time | $t_{12C_HD_DAT}$ | 0 | | | μs | |
| 8 | Data setup time | $t_{12C_SU_DAT}$ | 0.1 | | | μs | |
| 9 | Rise time SCL/SDA | t_{12C_R} | | | 0.3 | μs | |
| 10 | Fall time SCL/SDA | t_{12C_F} | | | 0.3 | μs | |
| 11 | Setup time stop condition | $t_{12C_SU_STO}$ | 0.6 | | | μs | |
| 12 | Noise interception SDA/SCL | t_{12C_NI} | | | 50 | ns | Spike suppression |

6 Interface Commands

6.1 Command Set

All commands are available for I²C and LIN communication but only in Command Mode (CM). CM is initiated by sending the command START_CM [72 74] (see section 1.3.1). Every received command is answered. The response consists of the 2 bytes for the requested data or a validation code, 1 byte for the check sum, and a 1-byte command reply. See the following table for exceptions (also refer to section 6.2).

Important: EEPROM programming must be enabled by sending the command EEP_WRITE_EN [6C F7 42_{HEX}].

A read command can be sent during an active measurement cycle (i.e., the processing time has not yet elapsed after sending one of the STRT_CYC_x or START_AD_x commands). If any of the other commands is to be sent during an active measurement cycle, the measurement command must first be aborted. Typically an active measurement cycle is aborted if a non-read command is received, but in special cases, the command might not be received correctly and the active measurement is not aborted. Therefore, for safe communication during an active measurement cycle, IDT recommends sending the START_CM command [72 74] first for non-read commands.

Table 6.1 Command Set

| Command (HEX) | Data | Command | Notes | Processing Time @ f _{OSC} =2MHz |
|---------------|------|------------------------------|--|--|
| 01 / 02 | | STRT_CYC_EEP STRT_CYC_RAM | Start measurement cycle including initialization from EEPROM or RAM. Output mode as configured. | 500µs + A/D conversion time |
| 10 to 2A | | READ_RAM | Read data from RAM addresses 00 _{HEX} through 1A _{HEX} . | 100µs |
| 30 to 4F | | READ_EEP | Read data from EEPROM addresses 00 _{HEX} through 1F _{HEX} . | 100µs |
| 50 | | ADJ_OSC_ACQ | Use this command with LIN communication only! Acquire frequency ratio ($2 f_{OSC} / f_{LIN} - 2$) where f _{OSC} is the frequency of the internal oscillator, and f _{LIN} is LIN communication frequency. Use this for adjusting the internal oscillator frequency with CFGAPP:OSCADJ. | 100µs |
| 58 | | GET_ERR_STATUS | Read and reset error code. | 100µs |
| 59 | | LIN_FAST | Enable LIN fast communication mode. Allows communication frequency up to 80kbit/s. Disables slope control and current limitation of LIN Transceiver. | 100µs |

| Command (HEX) | Data | Command | Notes | Processing Time @ f _{osc} =2MHz |
|---------------|---------|----------------|---|--|
| 5B | | GET_BCC_STATUS | Read status information from Broken Chip Check Returns C35B _{HEX} if there is no failure. Returns CF5B _{HEX} if there is a failure. | 100μs |
| 60 | 2 bytes | PWM_OUT | Set PWM output value and start PWM output. PWM must be configured in EEPROM/RAM! | 100μs |
| 65 | 2 bytes | ADJ_OSC_WRI | Write and activate oscillator adjust value CFGAPP:OSCADJ to RAM. Returns complete configuration word CFGAPP. | 100μs |
| 6C | 2 bytes | EEP_WRITE_EN | Enable data write to EEPROM. To be sent with data F742 _{HEX} ; sending any other data disables EEPROM writing. | 100μs |
| 6D | 2 bytes | CHECK_EEP | Calculate and return EEPROM signature. Low byte is start address; high byte is end address of evaluated area. Use [6D 1A 00 _{HEX}] for getting EEPROM signature of configuration. | 250μs |

| Command (HEX) | Data | Command | Notes | Processing Time @ f _{osc} =2MHz |
|---------------|---------|----------------|---|--|
| 72 | 1 byte | START_CM | Start Command Mode. To be sent with data 74 _{HEX} . | 100μs |
| 80 to 9A | 2 bytes | WRITE_RAM | Write data to RAM addresses 00 _{HEX} through 1A _{HEX} respectively. | 100μs |
| A0 to BE | 2 bytes | WRITE_EEP | Write data to EEPROM addresses 00 _{HEX} through 1E _{HEX} respectively. | 12.5ms |
| C0 | | COPY_EEP2RAM | Copy content of EEPROM addresses 00 _{HEX} through 1A _{HEX} to RAM. Restores EEPROM configuration in RAM. | 200μs |
| C3 | | COPY_RAM2EEP | Copy contents of RAM addresses 00 _{HEX} through 1A _{HEX} to EEPROM. Generates EEPROM signature; writes it to address 1B _{HEX} . Returns EEPROM signature. | 350ms |
| C9 | | GEN_EEP_SIGN | Calculate and return EEPROM signature and write it to EEPROM address 1B _{HEX} . | 12.7ms |
| CA | | GET_RAM_SIGN | Calculate and return RAM signature. | 250μs |
| CE | | GET_ROM_STATUS | Check ROM Diagnostic Status. Returns C3CE _{HEX} if there is no failure. Returns CFCE _{HEX} if there is a failure. | 10ms |
| CF | | GET_ROM_REV | Get hardware and ROM revision. <ul style="list-style-type: none"> The ROM version is defined by the low byte of the response to the CF command. The design version is defined by the high byte of the response to the CF command. | 100μs |

Note: All Dx commands are used for the calibration process and return raw conversion results via LIN and I²C. The Dx commands do not generate PWM output. With $f_{OSC}=2\text{MHz}$, processing time for the D0 to D7, DB, and DC commands is $100\mu\text{s} + \text{A/D conversion time}$. The processing time is 2 times this value for the D8 to DA and DE commands.

Note: Enabling the A/D converter clock divider (i.e., bit CFGAPP:ADCSLOW is set to 1) doubles only the A/D conversion time.

| Command (HEX) | Command | Notes |
|---------------|--------------------|---|
| D0 | START_AD_P | Start cyclic A/D conversion at channel bridge sensor. |
| D1 | START_AD_CT | Start cyclic A/D conversion at channel calibration temperature. |
| D2 | START_AD_MT | Start cyclic A/D conversion at temperature channel. |
| D3 | START_AD_SSCP | Start cyclic A/D conversion for positive-biased Sensor Short Check. |
| D4 | START_AD_PAZ | Start cyclic AD conversion auto-zero at channel bridge sensor. |
| D5 | START_AD_CTAZ | Start cyclic A/D conversion auto-zero at channel calibration temperature. |
| D6 | START_AD_MTAZ | Start cyclic A/D conversion auto-zero at channel temperature. |
| D7 | START_AD_SSCN | Start cyclic A/D conversion for negative-biased Sensor Short Check. |
| D8 | START_AD_P_AZC | Start cyclic A/D conversion at channel bridge sensor including auto-zero. |
| D9 | START_AD_CT_AZC | Start cyclic A/D conversion at channel calibration temperature including auto-zero. |
| DA | START_AD_MT_AZC | Start cyclic A/D conversion at channel temperature including auto-zero. |
| DB | START_AD_SSCP-SSCN | Start cyclic A/D conversion for Sensor Short Check positive biased – negative biased. |
| DC | START_AD_CMV | Start cyclic A/D conversion for Sensor Aging Check (Common Mode Voltage Measurement). |
| DE | START_AD_CMV_AZC | Start cyclic A/D conversion for Sensor Aging Check (Common Mode Voltage Measurement) including auto-zero. |

6.2 Command Processing

All implemented commands are available for both protocols – I²C and LIN. If CM is active, receiving a valid command interrupts the internal microcontroller (CMC) and starts a routine processing the received command. During this processing time, the interfaces are disabled and transmitted commands are ignored. The processing time depends on the internal system clock frequency. A command always returns data (e.g., register contents, acquired measurements) to interface output registers, which can be read out by read request.

6.3 Output Data in Command and Diagnostic Mode

Output data consists of two 16-bit words (4 bytes) that can be read out by read request via the LIN (*Read-by-Identifier* 32 or 35, see 4.6.3) or I²C interface. Depending on the current operation mode of the ZSSC3170 (CM or DM – see section 1.3), different data word formats are used.

Table 6.2 Serial Digital Interface Output Registers

| | Output Data Word 1 | | Output Data Word 2 | |
|-----------------|---|----------|--------------------|-------------------|
| | High Byte | Low Byte | High Byte | Low Byte |
| Command Mode | Requested data depending on command: - Memory content - Conditioned value - Measured raw value - Success code {C3 _{HEX} , command} - Failure code {CF _{HEX} , command} | | Check sum | Processed command |
| Diagnostic Mode | Error code | | Error code | |

The check sum is calculated with following formula: $check\ sum = FF_{HEX} - (HighByte_{1st_word} + LowByte_{1st_word})$.

6.4 Detailed Description of Oscillator Frequency Adjustment

ADJ_OSC_x commands are used to adjust the frequency of the internal oscillator. This frequency is adjustable in the range of 1.5 to 3MHz and has a directly proportional effect on the A/D conversion time and the PWM period. The internal oscillator frequency can be adjusted by CFGAPP:OSCADJ (see sections 7.2 and Table 7.5). The frequency is adjusted by steps with one step equal to approximately -125kHz (frequency is decreased if CFGAPP:OSCADJ is increased).

The ADJ_OSC_ACQ command is sent first. This command works ONLY with the LIN Master Request *Data-Dump*. It returns a value F_RATIO, which is equal to $(2 f_{OSC}/f_{LIN} - 2)$, where f_{OSC} is the present frequency of the internal oscillator and f_{LIN} is the LIN communication frequency. f_{LIN} is known, so F_RATIO can be used to calculate the existing f_{OSC} and determine if adjustment is needed. Alternatively, this frequency ratio can be read out with the LIN Master Request *Read-by-Identifier-33*.

Note that the resolution of the frequency measurement is better when a lower communication frequency is used.

The required adjustment D_OSCADJ for determining the new setting OSCADJ_{NEW} for CFGAPP:OSCADJ that is needed to set the internal oscillator to the target frequency f_{OSC_NEW} can be calculated from F_RATIO and the adjustment increment of -125kHz/step:

$$f_{\text{RATIO}} = 2 * \frac{f_{\text{OSC}}}{f_{\text{LIN}}} - 2$$

$$D_OSCADJ = \frac{f_{\text{OSC_NEW}} - f_{\text{LIN}} * (f_{\text{RATIO}} + 2) * 0.5}{-125\text{kHz}} \quad \text{and} \quad \text{OSCADJ}_{\text{NEW}} = \text{OSCADJ} + D_OSCADJ$$

Where

D_OSCADJ is the difference between the present oscillator adjustment setting and the required setting.

The ADJ_OSC_WRI command is used to write CFGAPP:OSCADJ to RAM and to activate the new adjustment. The command returns the complete configuration word CFGAPP (all other configuration bits retain their value).

This sequence allows an easy and accurate adjustment of the internal frequency during end-of-line calibration.

Table 6.3 Oscillator Frequency Adjustment Sequence

| Command | Description | Comment |
|---|----------------|--|
| Task: Measure and adjust the internal oscillator frequency f_{OSC} | | |
| [72 74] | START_CM | Start command mode |
| [22] | READ_RAM 0x12 | Read RAM CFGAPP |
| [READ] | READ CFGAPP | Read CFGAPP; OSCADJ = CFGAPP[4:0] |
| LIN[50] | ADJ_OSC_ACQ | Acquire frequency ratio |
| [READ] | READ F_RATIO | Read F_RATIO |
| | | Manually calculate D_OSCADJ from F_RATIO |
| | | Manually calculate OSCADJ _{new} = OSCADJ + D_OSCADJ |
| [65 OSCADJ _{new}] | ADJ_OSC_WRI | Write CFGAPP:OSCADJ |
| [READ] | READ CFGAPP | Read CFGAPP _{new} |
| Task: Check the resulting internal oscillator frequency (optional) | | |
| LIN[50] | ADJ_OSC_ACQ | Acquire frequency ratio |
| [READ] | READ F_RATIO | Read F_RATIO |
| Task: Write the new oscillator frequency adjustment to EEPROM | | |
| [6C F7 42] | EEP_WRITE_EN | Enable data write to EEPROM |
| [B2 CFGAPP _{new}] | WRITE_EEP 0x12 | Write EEPROM CFGAPP _{new} |
| [C9] | GEN_EEP_SIGN | Generate and write EEPROM signature |

7 EEPROM and RAM

7.1 Programming the EEPROM

Programming the EEPROM is done using an internal charge pump to generate the required programming voltage. The timing of the programming pulses is controlled internally. The programming time for a write operation is typically 12.5ms independent of the programmed clock frequency (CFGAPP:OSCADJ). Recommendation: Wait a minimum of 15ms per write operation before starting the next communication.

To program the EEPROM, the ZSSC3170 must be set to Command Mode by the command START_CM [72 74] and EEPROM programming must be enabled by the command EEP_WRITE_EN [6C F7 42]. Writing data to the EEPROM is done via the serial digital interface by sending specific commands (refer to section 6). The WRITE_EEP command includes the address of the targeted EEPROM word and is followed by two data bytes. During EEPROM programming, the serial digital interface is disabled and no further commands can be recognized.

The COPY_RAM2EEP command [C3] writes the contents of the RAM mirror area to the EEPROM. This is to simplify the calibration process when the ZSSC3170 is configured iteratively. The EEPROM signature, which is not mirrored in RAM, is generated, written to EEPROM, and returned to the interface output register. This copy operation includes 28 EEPROM write operations and therefore typically requires 350ms (recommended wait time 420ms).

7.2 EEPROM and RAM Contents

The configuration of the ZSSC3170 is stored in 32 EEPROM 16-bit words.

Calibration coefficients for conditioning the sensor signal via conditioning calculations and output limits are stored in 15 words. There are 5 words for setting the configuration of the ZSSC3170 for the application. 7 words are used to configure the LIN interface. One register is used for storing the EEPROM signature, which is used in NOM to check the validity of the EEPROM contents after power-on. 4 additional 16-bit words are available for optional user data.

After every power-on, the EEPROM contents are mirrored to RAM. After this read out, the contents of the RAM mirror is checked by calculating the signature and comparing it to the one stored in EEPROM. If a signature error is detected, the ZSSC3170 changes to Diagnostic Mode (DM). In LIN mode, DM is indicated by error flags contained in the LIN signal. In PWM mode, the output is set to the recessive level. Subsequently the error code can be read out via LIN or I²C. The LIN interface is also enabled in PWM mode.

The configuration of the device is done from the mirrored area in RAM, so the configuration words are subsequently transferred to the internal registers. The calibration coefficients for the conditioning calculations are also read from RAM. As a result, every change to the RAM mirror area impacts the configuration and behavior of the device after the next start of the measurement cycle.

After power-on, the contents of the RAM mirror area are determined by the EEPROM contents and can then be changed by specific commands writing to RAM. This new configuration can be activated by the STRT_CYC_RAM command or START_AD_x commands.

Table 7.1 EEPROM and RAM Contents

| EEPROM/RAM Address in Hex | Write cmd RAM/EEPROM | Description |
|--|----------------------|---|
| Note: The MSB is given first if an address has more than one assignment. | | |
| Conditioning Coefficients – Correction Formula Bridge Sensor Signal (section 2.2) | | |
| 00 | 80/A0 | c0 - Offset |
| 01 | 81/A1 | c1 - Gain |
| 02 | 82/A2 | c2 - Non-linearity 2 nd order |
| 03 | 83/A3 | c3 - Non-linearity 3 rd order |
| 04 | 84/A4 | c4 - Temperature coefficient offset 1 st order |
| 05 | 85/A5 | c5 - Temperature coefficient offset 2 nd order |
| 06 | 86/A6 | c6 - Temperature coefficient gain 1 st order |
| 07 | 87/A7 | c7 - Temperature coefficient gain 2 nd order |
| LIN Mode Configuration (section 2.4 and section 4) | | |
| 08 | 88/A8 | t0 – Offset temperature |
| 09 | 89/A9 | t1 – Gain temperature |
| 0A | 8A/AA | t2 – Non-linearity 2 nd order temperature |
| 0B | 8B/AB | LINPMAX:LINPMIN – Maximum/minimum LIN bridge sensor output value |
| 0C | 8C/AC | LINTMAX:LINTMIN – Maximum/minimum LIN temperature output value |
| 0D | 8D/AD | 4 LSBs of values LINPMAX:LINPMIN:LINTMAX:LINTMIN |
| PWM Mode Configuration (section 3) | | |
| 08 | 88/A8 | PWMMIN – Minimum PWM output value |
| 09 | 89/A9 | PWMMAX – Maximum PWM output value |
| 0A | 8A/AA | LPON – Low bridge sensor signal on-value |
| 0B | 8B/AB | LPOFF – Low bridge sensor signal off-value |
| 0C | 8C/AC | PWMOFF – PWM off-value |
| 0D | 8D/AD | ZMIN – PWM off-filter parameter |
| Sensor Aging Check (SAC) limits | | |
| 0E | 8E/AE | CMVMAX:CMVMIN – Upper/lower limit common mode voltage |
| Configuration Words (section 7.3) | | |
| 0F | 8F/AF | CFGLIN – Configuration of LIN interface and frames |
| 10 | 90/B0 | CFGAFE – Configuration of analog front end |
| 11 | 91/B1 | CFGTS – Configuration of temperature measurement |
| 12 | 92/B2 | CFGAPP – Configuration of target application |
| 13 | 93/B3 | CFGSF – Configuration of safety functions (Diagnostic function and bridge sensor signal filter function) |

| EEPROM/RAM Address in Hex | Write cmd RAM/EEPROM | Description |
|--|----------------------|---|
| Note: The MSB is given first if an address has more than one assignment. | | |
| LIN Configuration (section 4) | | |
| 14 | 94/B4 | LIN Product Identification – Supplier ID |
| 15 | 95/B5 | LIN-Product Identification – Function ID |
| 16 | 96/B6 | LIN Product Identification – Variant ID (8 LSB) |
| 17 | 97/B7 | MSB: 0 (1 bit) LIN Initial NAD (7 bit) LSB: 0 (1 bit) LIN Configurable NAD (7 bit) |
| 18 | 98/B8 | MSB: LIN publisher PID2 for data transmission (8 bit) LSB: LIN publisher PID1 for data transmission (8 bit) 6 LSBs are also used as a singular device address for I ² C communication. Resulting I ² C device address range is 40 _{HEX} to 7F _{HEX} . |
| 19 | 99/B9 | LIN message ID for publisher PID1 (LIN 2.0). |
| 1A | 9A/BA | LIN message ID for publisher PID2 (LIN 2.0). |
| Calculated Signature Based on Register 00_{HEX} to 1A_{HEX} Data | | |
| 1B | - /BB | Signature |
| Free Memory Available for Optional Use by User Applications (not included in signature) | | |
| 1C | - /BC | Free user memory, not included in signature (e.g., serial number) |
| 1D | - /BD | Free user memory, not included in signature (e.g., serial number) |
| 1E | - /BE | Free user memory, not included in signature |
| Restricted | | |
| 1F | - /BF | No customer access - IDT restricted use |

Note: The contents of the EEPROM registers at delivery are not specified and can be subject to changes. Particularly with regard to traceability, the contents can be unique per die. Note that contents at delivery might not have a valid signature. Consequently the ZSSC3170 would start in the Diagnostic Mode.

All registers must be rewritten during the calibration procedure.

7.3 Configuration Words

The data stored in EEPROM at addresses 0F_{HEX} to 13_{HEX} determine the configuration of the ZSSC3170, as explained in the following tables.

Table 7.2 Configuration Word CFGLIN

| Bit | CFGLIN - Configuration of LIN Interface and Frames | EEPROM/RAM Address F _{HEX} | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|-------------------------------------|------------------------------------|---------------------------|--------------------------------------|------------------|---------|----|----------------------|--------------------|------------------|------------------|----|------------------------------------|---------------------------|--------------------------------------|------------------|----|----------------------|-------------------|----------------|------------------|----|----------------------|-------------------|----------------|------------------|
| 15:12 | Not used | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11:10 | LIN ConFiGuration FRaMe | LINCfGFRM | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1"> <thead> <tr> <th></th> <th colspan="2">Data Byte 0</th> <th colspan="2">Data Byte 3</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>12-bit Bridge Sensor</td> <td>12-bit Temperature</td> <td>4-bit Status</td> <td>4-bit Variant ID</td> </tr> <tr> <td>01</td> <td>12-bit Bridge Sensor</td> <td>10-bit Temperature</td> <td>2-bit Status</td> <td>8-bit Variant ID</td> </tr> <tr> <td>10</td> <td>12-bit Bridge Sensor</td> <td>8-bit Temperature</td> <td>4-bit Status</td> <td>8-bit Variant ID</td> </tr> <tr> <td>11</td> <td>11-bit Bridge Sensor</td> <td>9-bit Temperature</td> <td>4-bit Status</td> <td>8-bit Variant ID</td> </tr> </tbody> </table> | | | Data Byte 0 | | Data Byte 3 | | 00 | 12-bit Bridge Sensor | 12-bit Temperature | 4-bit Status | 4-bit Variant ID | 01 | 12-bit Bridge Sensor | 10-bit Temperature | 2-bit Status | 8-bit Variant ID | 10 | 12-bit Bridge Sensor | 8-bit Temperature | 4-bit Status | 8-bit Variant ID | 11 | 11-bit Bridge Sensor | 9-bit Temperature | 4-bit Status | 8-bit Variant ID |
| | | | Data Byte 0 | | Data Byte 3 | | | | | | | | | | | | | | | | | | | | | | |
| | 00 | | 12-bit Bridge Sensor | 12-bit Temperature | 4-bit Status | 4-bit Variant ID | | | | | | | | | | | | | | | | | | | | | |
| | 01 | | 12-bit Bridge Sensor | 10-bit Temperature | 2-bit Status | 8-bit Variant ID | | | | | | | | | | | | | | | | | | | | | |
| 10 | 12-bit Bridge Sensor | 8-bit Temperature | 4-bit Status | 8-bit Variant ID | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | 11-bit Bridge Sensor | 9-bit Temperature | 4-bit Status | 8-bit Variant ID | | | | | | | | | | | | | | | | | | | | | | | |
| <i>Status Bits Frame 01: LSB: Bridge Sensor or Temperature Error MSB: Response Error</i> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9:8 | LIN ConFiGuration STATus bits | LINCfGSTAT | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1"> <thead> <tr> <th></th> <th>LSB</th> <th>LSB + 1</th> <th>LSB + 2</th> <th>LSB + 3</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>Bridge Sensor Error</td> <td>Temperature Error</td> <td>Values Refreshed</td> <td>Response Error</td> </tr> <tr> <td>01</td> <td>Bridge Sensor Signal Out of Limits</td> <td>Temperature Out of Limits</td> <td>Bridge Sensor or Temperature Invalid</td> <td>Response Error</td> </tr> <tr> <td>10</td> <td>Bridge Sensor Error</td> <td>Response Error</td> <td>Checksum Error</td> <td>Bit Error</td> </tr> <tr> <td>11</td> <td>Bridge Sensor Error</td> <td>Temperature Error</td> <td>Checksum Error</td> <td>Bit Error</td> </tr> </tbody> </table> | | | LSB | LSB + 1 | LSB + 2 | LSB + 3 | 00 | Bridge Sensor Error | Temperature Error | Values Refreshed | Response Error | 01 | Bridge Sensor Signal Out of Limits | Temperature Out of Limits | Bridge Sensor or Temperature Invalid | Response Error | 10 | Bridge Sensor Error | Response Error | Checksum Error | Bit Error | 11 | Bridge Sensor Error | Temperature Error | Checksum Error | Bit Error |
| | | | LSB | LSB + 1 | LSB + 2 | LSB + 3 | | | | | | | | | | | | | | | | | | | | | |
| | 00 | | Bridge Sensor Error | Temperature Error | Values Refreshed | Response Error | | | | | | | | | | | | | | | | | | | | | |
| | 01 | | Bridge Sensor Signal Out of Limits | Temperature Out of Limits | Bridge Sensor or Temperature Invalid | Response Error | | | | | | | | | | | | | | | | | | | | | |
| 10 | Bridge Sensor Error | Response Error | Checksum Error | Bit Error | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | Bridge Sensor Error | Temperature Error | Checksum Error | Bit Error | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | LIN SaVe ConFiGuration Service enable 0 disabled 1 enabled | LINSVCFG | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | LIN SaMPLe mode 0 3 samples / bit 1 5 samples / bit | LINSMPL | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5:4 | LIN Low Pass 00 disabled 10 8 samples 01 4 samples 11 16 samples | LINLP | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | LIN SLeeP Mode 0 disabled 1 enabled | LINSLP | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Not used | - | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1:0 | LIN MODE | LINMODE | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 00 LIN Protocol Revision 1.3 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 01 LIN Protocol Revision 2.0 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 10 LIN Protocol Revision 2.1 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 11 LIN Protocol Revision 2.1 | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 7.3 Configuration Word CFGAFE

| Bit | CFGAFE - Configuration of analog front end | EEPROM/RAM Address 10 _{HEX} | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------|---|--------------------------------------|-------------------|------|-------------------|--------|-------|------|-------|---------|------|------|-----|------|-----|------|----|------|---|------|-----|------|-------|------|-----|------|-----|--|--|--|--|-------|
| 15 | <p>Bridge Sensor channel eXtended Zero Compensation POLarity (offset compensation by analog front end - refer to section 2.1)</p> <p>0 negative – compensates positive offsets 1 positive – compensates negative offsets</p> | PXZCPOL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 14:10 | <p>Bridge Sensor channel eXtended Zero Compensation value (offset compensation by analog front end - refer to section 2.1)</p> <p>Offset compensation is only active if PXZC ≠ 0. The value of one compensation step depends on the selected input span (refer to section 2.3 in the ZSSC3170 Data Sheet).</p> | PXZC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9:6 | <p>Bridge Sensor channel GAIN (a_{IN} - refer to section 2.1)</p> <table border="0"> <tr> <td>0000</td><td>420</td> <td>0101</td><td>70</td> <td>1001</td><td>14</td> </tr> <tr> <td>0001</td><td>280</td> <td>0110</td><td>52.5</td> <td>1010</td><td>9.3</td> </tr> <tr> <td>0010</td><td>210</td> <td>0111</td><td>35</td> <td>1011</td><td>7</td> </tr> <tr> <td>0011</td><td>140</td> <td>1000</td><td>26.25</td> <td>11dd</td><td>2.8</td> </tr> <tr> <td>0100</td><td>105</td> <td></td><td></td> <td></td><td></td> </tr> </table> | 0000 | 420 | 0101 | 70 | 1001 | 14 | 0001 | 280 | 0110 | 52.5 | 1010 | 9.3 | 0010 | 210 | 0111 | 35 | 1011 | 7 | 0011 | 140 | 1000 | 26.25 | 11dd | 2.8 | 0100 | 105 | | | | | PGAIN |
| 0000 | 420 | 0101 | 70 | 1001 | 14 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0001 | 280 | 0110 | 52.5 | 1010 | 9.3 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0010 | 210 | 0111 | 35 | 1011 | 7 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0011 | 140 | 1000 | 26.25 | 11dd | 2.8 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0100 | 105 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5:4 | <p>A/D Conversion input Range Shift regarding measured signal (r_{ADC} – refer to section 2.1)</p> <p>00 $\frac{1}{16} \rightarrow$ ADC range = $[(-1/16 V_{ADC_REF}) \text{ to } (+15/16 V_{ADC_REF})]$ 01 $\frac{1}{8} \rightarrow$ ADC range = $[(-1/8 V_{ADC_REF}) \text{ to } (+7/8 V_{ADC_REF})]$ 10 $\frac{1}{4} \rightarrow$ ADC range = $[(-1/4 V_{ADC_REF}) \text{ to } (+3/4 V_{ADC_REF})]$ 11 $\frac{1}{2} \rightarrow$ ADC range = $[(-1/2 V_{ADC_REF}) \text{ to } (+1/2 V_{ADC_REF})]$</p> | ADCRS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3:2 | <p>A/D Conversion MODE (resolution 2nd conversion step)</p> <p>Valid for bridge sensor signal as well as for temperature measurement.</p> <table border="0"> <tr> <td>00</td><td>2 bit</td> <td>10</td><td>4 bit</td> </tr> <tr> <td>01</td><td>3 bit</td> <td>11</td><td>5 bit</td> </tr> </table> | 00 | 2 bit | 10 | 4 bit | 01 | 3 bit | 11 | 5 bit | ADCMODE | | | | | | | | | | | | | | | | | | | | | | |
| 00 | 2 bit | 10 | 4 bit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 | 3 bit | 11 | 5 bit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | <p>A/D Conversion RESolution (r_{ADC} - refer to section 2.1)</p> <p>Valid for bridge sensor signal as well as for temperature measurement.</p> <table border="0"> <tr> <td>0</td><td>13 bit</td> <td>1</td><td>14 bit</td> </tr> </table> | 0 | 13 bit | 1 | 14 bit | ADCRES | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 13 bit | 1 | 14 bit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | <p>A/D Conversion ORDER</p> <table border="0"> <tr> <td>0</td><td>1-step conversion</td> <td>1</td><td>2-step conversion</td> </tr> </table> | 0 | 1-step conversion | 1 | 2-step conversion | ADCORD | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1-step conversion | 1 | 2-step conversion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 7.4 Configuration Word CFGTS

| Bit | CFGTS - Configuration of Temperature Measurement | EEPROM/RAM Address 11 _{HEX} | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------|--|--------------------------------------|-----------------|-----------------|-----------------|--|--|--|-------|-------|----------|--|--|-----------------------|------------|----------------|----|-----|-----|-----|------|----|-----|-----|------|------|----|-----|-----|------|------|--------|
| 15:13 | Temperature channel OFFSet value Offset by analog front end. | TOFFS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12:10 | Calibration Temperature channel OFFSet value Offset by analog front end. | CTOFFS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9:8 | Temperature channel GAIN <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2"></th> <th>Internal sensor</th> <th colspan="2">External sensor</th> </tr> <tr> <th colspan="2"></th> <th>Diode</th> <th>Diode</th> <th>Resistor</th> </tr> <tr> <th colspan="2"></th> <th>LSB₁₄ / K</th> <th>ppmFS / mV</th> <th>ppmFS / (mV/V)</th> </tr> </thead> <tbody> <tr> <td>0d</td> <td>GT2</td> <td>-34</td> <td>966</td> <td>3333</td> </tr> <tr> <td>10</td> <td>GT3</td> <td>-38</td> <td>1063</td> <td>3667</td> </tr> <tr> <td>11</td> <td>GT4</td> <td>-41</td> <td>1159</td> <td>4000</td> </tr> </tbody> </table> | | | Internal sensor | External sensor | | | | Diode | Diode | Resistor | | | LSB ₁₄ / K | ppmFS / mV | ppmFS / (mV/V) | 0d | GT2 | -34 | 966 | 3333 | 10 | GT3 | -38 | 1063 | 3667 | 11 | GT4 | -41 | 1159 | 4000 | TGAIN |
| | | Internal sensor | External sensor | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | Diode | Diode | Resistor | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | LSB ₁₄ / K | ppmFS / mV | ppmFS / (mV/V) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0d | GT2 | -34 | 966 | 3333 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | GT3 | -38 | 1063 | 3667 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | GT4 | -41 | 1159 | 4000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7:6 | Calibration Temperature channel GAIN <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2"></th> <th>Internal sensor</th> <th colspan="2">External sensor</th> </tr> <tr> <th colspan="2"></th> <th>Diode</th> <th>Diode</th> <th>Resistor</th> </tr> <tr> <th colspan="2"></th> <th>LSB₁₄ / K</th> <th>ppmFS / mV</th> <th>ppmFS / (mV/V)</th> </tr> </thead> <tbody> <tr> <td>0d</td> <td>GT2</td> <td>-34</td> <td>966</td> <td>3333</td> </tr> <tr> <td>10</td> <td>GT3</td> <td>-38</td> <td>1063</td> <td>3667</td> </tr> <tr> <td>11</td> <td>GT4</td> <td>-41</td> <td>1159</td> <td>4000</td> </tr> </tbody> </table> | | | Internal sensor | External sensor | | | | Diode | Diode | Resistor | | | LSB ₁₄ / K | ppmFS / mV | ppmFS / (mV/V) | 0d | GT2 | -34 | 966 | 3333 | 10 | GT3 | -38 | 1063 | 3667 | 11 | GT4 | -41 | 1159 | 4000 | CTGAIN |
| | | Internal sensor | External sensor | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | Diode | Diode | Resistor | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | LSB ₁₄ / K | ppmFS / mV | ppmFS / (mV/V) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0d | GT2 | -34 | 966 | 3333 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | GT3 | -38 | 1063 | 3667 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | GT4 | -41 | 1159 | 4000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5:3 | Temperature Sensor select : 00d on-chip diode d10 external resistor on pin VTN1 d11 external diode on pin VTN1 100 external resistor on pin VTN2 101 external diode on pin VTN2 | MTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2:0 | Calibration Temperature Sensor select : 00d on-chip diode d10 external resistor on pin VTN1 d11 external diode on pin VTN1 100 external resistor on pin VTN2 101 external diode on pin VTN2 | CTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 7.5 Configuration Word CFGAPP

| Bit | CFGAPP - Configuration of Target Application | EEPROM/RAM Address 12 _{HEX} |
|------|---|--------------------------------------|
| 15 | LIN Transceiver Conformity TEST 0 disabled 1 enabled | LINTEST |
| 14 | Enables triggering a RESet if Diagnostic Mode (DM) occurs 0 stop and DM 1 reset and start-up again Reset is executed after time-out of watchdog. | DMRES |
| 13 | A/D Conversion SLOW mode 0 disabled 1 enabled | ADCSLOW |
| 12 | Bridge Signal POLarity (Differential voltage at pins VBP, VBN) 0 positive ($V_{IN_DIFF} = V_{VBP} - V_{VBN}$) 1 negative ($V_{IN_DIFF} = V_{VBN} - V_{VBP}$) | BPOL |
| 11 | Not used | - |
| 10:9 | PWM SLow Rate adjust | PWMSLR |
| 8 | PWM SLOPE control 0 current control 1 voltage control | PWMSLOPE |
| 7 | PWM MODE 0 low-side switch (LOUT pin) 1 high-side switch (HOOUT pin) | PWMMODE |
| 6 | PWM ENable 0 disabled (output via LIN) 1 enabled (output via PWM) | PWMENA |
| 5 | Enables Bias current BOOST for analog front end 0 disabled 1 enabled Activation is recommended for clock frequency > 2.8MHz. | BBOOST |
| 4:0 | ADJ ust frequency f_{osc} of internal OSC illator Adjustment of f_{osc} in the range of 1.5 to 3MHz. | OSCADJ |

Table 7.6 Configuration Word CFGSF

| Bit | CFGSF - Configuration of Safety Functions | EEPROM/RAM Address 13 _{HEX} |
|--------------|--|--------------------------------------|
| 15:14 | Not used | - |
| 13 | Enables EEPROM lock for LIN communication 0 disabled 1 enabled | EEPLOCK |
| 12 | Enable ROM Signature Check at power-on. Start-up time is increased by approximately 10ms. 0 disabled 1 enabled | CHKROM |
| 11 | Enable Broken Chip Check 0 disabled 1 enabled | CHKBCC |
| 10 | Enable Sensor Short Check 0 disabled 1 enabled | CHKSSC |
| 9 | Enable Sensor Connection Check High Capacity Mode 0 disabled 1 enabled | CHKSCCHIC |
| 8 | Enable Sensor Connection Check 0 disabled 1 enabled | CHKSCC |
| 7 | Enable Sensor Aging Check 0 disabled 1 enabled | CHKSAC |
| 6 | Enable Temperature Out-Of-Range Check Applies to temperature and calibration temperature. 0 disabled 1 enabled | CHKOOR |
| 5:3 | Digital Low Pass Filter Bridge Sensor Signal Output - Differential coefficient Range [0; 7] Setting both PDIFF and PAVRG to 0 disables the filter. | PDIFF |
| 2:0 | Digital Low Pass Filter Bridge Sensor Signal Output - Averaging coefficient Range {1; 2; 4; 8; 16; 32; 64; 128} Setting both PDIFF and PAVRG to 0 disables the filter. | PAVRG |

7.4 EEPROM Signature

The EEPROM signature (address 1B_{HEX}) is used to check the validity of the EEPROM contents. The signature is built using a polynomial arithmetic modulo 2. The following source code generates the signature if the field `eeptest[]` is allocated by the EEPROM content (addresses 0_{HEX} to 1A_{HEX}). The parameter N is the count of applicable addresses and must be set N=27.

Figure 7.1 Source Code Signature Generation

```
#define POLYNOM 0xA005
unsigned short signature(eepcont, N)
    unsigned short eepcont[], N;
    {
        unsigned short sign, poly, p, x, i, j;
        sign = 0; poly = POLYNOM;
        for (i=0; i<N; i++) {
            sign^=eepcont[i];
            p=0; x=sign&poly;
            for (j=0; j<16; j++, p^=x, x>>=1);
            sign<<=1; sign+=(p&1);
        }
        return(~sign);
    }
```

7.5 EEPROM Write Locking

The ZSSC3170 supports EEPROM write locking (EEPLOCK). If the mode is active (CFGSF:EEPLOCK=1), it is not possible to overwrite the current EEPROM content using the LIN Master Request *Data-Dump*. The ZSSC3170 answers the command `EEP_WRITE_EN` with error code CF6C_{HEX} if the EEPROM is locked.

Note that the LIN Master Request *Save-Configuration* stores the configured NAD and PIDs to EEPROM even if `EEPLOCK` is activated.

An activated `EEPLOCK` does not affect writing to the EEPROM using I²C and can always be reset using I²C.

`EEPLOCK` is active only if programmed into EEPROM and activated due to

- New power-on or
- Receiving the `EEP_WRITE_EN` command or
- Starting the measurement cycle by receiving the `STRT_CYC_x` command

The following write sequence is possible:

- Write calibration data including `EEPLOCK` to RAM mirror
- Enable EEPROM writing by sending the command `EEP_WRITE_EN`
- Copy the RAM mirror to EEPROM
- Write the EEPROM signature directly to EEPROM

If an invalid EEPROM signature is detected, the EEPROM lock is always deactivated.

8 Related Documents

| Document |
|--|
| ZSSC3170 Data Sheet |
| ZSSC3170 AN – LIN and PWM Operation |
| SSC App Note - RBIC1 Calibration DLL Description |
| ZSSC3170 High Voltage Protection Description |
| ZSSC3170 LIN Interface Description |
| LIN Specification Package 2.1 (LIN Consortium, 2006-11-24) |

Visit the ZSSC3170 product page www.IDT.com/ZSSC3170 or contact your nearest sales office for the latest version of these documents.

9 Glossary

| Term | Description |
|------|---|
| ADC | Analog-to-Digital Converter |
| BCC | Broken Chip Check |
| CM | Command Mode |
| CMC | Calibration Microcontroller |
| CMV | Common Mode Voltage |
| DM | Diagnostic Mode |
| HSS | High-Side Switch |
| LSB | Least Significant Bit |
| LSS | Low-Side Switch |
| MSB | Most Significant Bit |
| NAD | Node Address Byte |
| NOM | Normal Operating Mode |
| PCI | Protocol Control Information Byte |
| PID | Protected Identifier |
| RSID | Response Service Identifier Byte |
| SAC | Sensor Aging Check |
| SCC | Sensor Connection Check |
| SID | Service Identifier Byte |
| SSC | Sensor Short Check or Sensor Signal Conditioner |

10 Document Revision History

| Revision | Date | Description |
|----------|--------------------|--|
| 1.10 | | First release of document. |
| 1.20 | May 27, 2010 | Completely revised Functional Description. |
| 1.30 | July 22, 2010 | Broken Chip Check (section 6.1, Table 7.6). LIN Fast Mode (section 6.1). Sensor Connection Check High Capacity Mode (Table 7.6) Updates for sales contact information. |
| 1.40 | August 31, 2010 | Error code Broken Chip Check (Table 1.1) LIN Publisher Frame regarding PID2 (section 4.3) Response LIN MRF Read-By-Identifier-34/35 (Table 4.7) Positive Response LIN 2.0 MRF Read-By-Identifier-16/17 (Table 4.8) Oscillator frequency adjustment sequence (Table 6.3) Gain Temperature Channel CFGTS:MTGAIN, CFGTS:CTGAIN (Table 7.4) |
| 1.50 | January 10, 2011 | Clarification of setting the LIN Protocol Revision (Table 4.20, Table 7.2) |
| 1.60 | 2011 | Broadcast added (Table 4.18) Response Data Format for <i>Read-By-Identifier-34</i> (Table 4.7) |
| 1.70 | September 10, 2013 | PWM operation and LIN Sleep mode incompatibility note added. Revision to conditioning equation in section 2.2. Recommendation regarding non-read commands added to section 6.1. Correction of write command EEPROM addresses for memory addresses 1B through 1F in Table 7.1. Minor edits for clarity. |
| 1.80 | November 25, 2015 | Update regarding Diagnostic Mode and PWM output mode in section 1.3.4. Updates for Dx command descriptions in Table 6.1. CRC changed to check sum. Update for contact information. Minor edits for minor corrections or clarity. |
| | March 29, 2016 | Changed to IDT branding. |
| | May 16, 2016 | Removed footnote 1 reference in Table 1.1 for the SAC and BCC checks. Removed asterisk on "Temperature Out of Range Check." |

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.