

Using the SENT Communications Output Protocol with A1341 and A1343 Devices

By Nevenka Kozomora
Allegro MicroSystems

Allegro™ MicroSystems supports the Single-Edge Nibble Transmission (SENT) protocol in certain advanced digital output sensor ICs. The SENT protocol is a commonly accepted automotive protocol for highly efficient transfer of sensor data along intravehicular communications networks, and is standardized by the Society of Automotive Engineering in publication SAEJ2716.

This application note provides a description of the Allegro implementation of the SENT protocol, which includes extensions developed by Allegro to enhance the information carrying dimensions of the output from the Allegro sensor IC to the vehicle electronic control units (ECU).

System Requirements

The Allegro devices comply with the SENT 3-wire standard: providing power along the 5 V wire, a logic-level signal output, and a ground reference. Specific devices may provide additional capabilities with other pin configurations.

The system host controller must be capable of handling at least 20 bits of data, including data, cyclic redundancy checking (CRC), system status, and communication status.

Table of Contents

System Requirements	1	Asynchronous Transfer with SENT Messages of Equal Duration Maximum Response Time	16
SENT Protocol Overview	2	Synchronous Transfer Mode Minimum Response Time	17
SENT Output Mode	3	Synchronous Transfer Maximum Response Time	18
Message Structure	4	Device Response Time with Continuous Field Change	19
Data Nibble Format	6	Fast SENT Feature	20
Output Message Transfer	7	Minimum Message Length	20
Optional Serial Output Protocol	12	Device Response Time Example Calculation	22
Device Response Time	13	Electrical Specifications	26
Propagation Delay and Output Update Rate	13	SENT Data Programming Parameters	27
Asynchronous Transfer Minimum Response Time	14		
Asynchronous Transfer Maximum Response Time	15		

SENT Protocol Overview

The Allegro implementation of the SENT protocol complies with the J2716 Rev. 2010 SENT standard. The Allegro sensor IC takes the role of Slave in the SENT serial communications. In this role, the Allegro device sends information about the magnetic field applied to the device and about the internal status of the device. The Allegro device sends both types of information from the device output pin.

Two communications states are supported (figure 1):

- Default state: Slave sends messages to Master continuously.
- Programmable State: Slave sends one message to the Master after receiving a trigger signal from the Master.

The Allegro implementation of the SENT protocol has various programmable options:

- Clock Rates from 0.25 to 31.75 μ s
- Type and quantity of Data nibbles
- Output Frame Rate
- Duration of nibble low state
- Polarity on SENT output (to invert the signal)
- Status and Communication nibble format (error and serial protocol)
- Adjustable SENT nibble fall time

The Allegro implementation of the SENT protocol enables the user to speed-up communication by using minimum tick time, minimum fixed time in the nibble, and minimum quantity of SENT nibbles in a message.

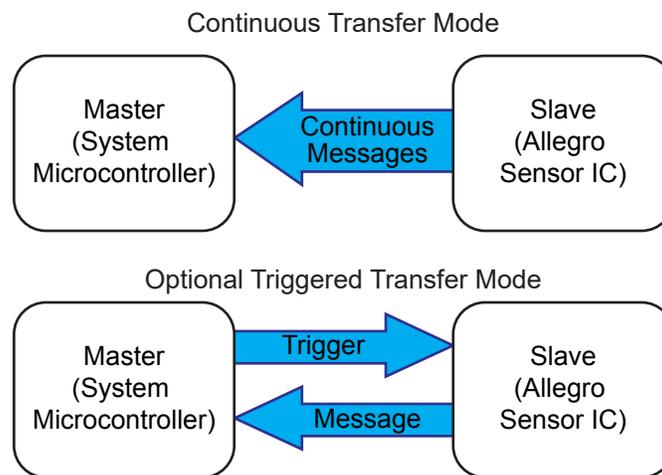


Figure 1. Message communication from the Allegro IC can be either: continuous (upper panel) or individual messages can be in response to a trigger signal from the Master (lower panel).

SENT Output Mode

The SENT output mode converts the input magnetic signal to a binary value digitally preprocessed and mapped to a Full-Scale Output (FSO) range as shown in figure 2. This data is inserted into a binary pulse message, referred to as a *frame*, that conforms to the SENT data transmission specification (SAEJ2716 JAN2010).

Certain parameters for configuration of the SENT messages can be set in EEPROM.

Nibble fall time is changed by changing the drive current to the output pin.

The SENT output mode is configured by setting the following parameters in EEPROM:

- PWM_MODE parameter set to 0 (default) to select the SENT option
- SENT_x programming parameters (see EEPROM Structure section)

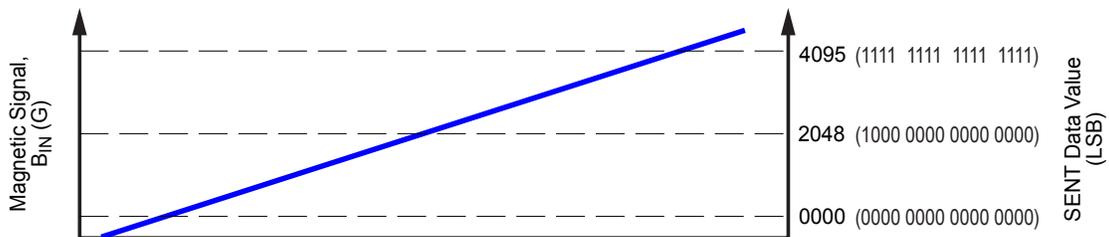


Figure 2. SENT mode outputs a digital value that can be read by the external controller.

Message Structure

A SENT message is a series of *nibbles*, with the following characteristics:

- Each nibble is a pair of voltage intervals: a low-voltage interval and a high-voltage interval (figure 3).
- The time duration of the nibble depends on the total duration, determined by the total quantity of time units, referred to as *ticks*, and the information contained by the nibble. The duration of a tick is set by dividing a 4 MHz clock by the value of the SENT_TICK parameter. The duration of the nibble is the sum of the low-voltage interval plus the high-voltage interval.
- The low-voltage interval is by default the *delimiting state*, which only sets a boundary for the nibble; to assign the delimiting state, select a fixed number of ticks for the inter-

val (the SENT_LOVAR parameter selects the interval, and SENT_FIXED sets the duration).

- The other interval in the pair, high-voltage, becomes the *information state* and is variable in duration, depending on the nibble data value. See table 1.

The nibbles of a SENT message are arranged in the following required sequence (see figure 4 and table 2):

1. Synchronization and Calibration: flags the start of the SENT message
2. Status and Communication: provides the device status and the format of the data
3. Data: magnetic field and optional data
4. CRC: error checking
5. Pause Pulse (optional): sets timing relative to device updates

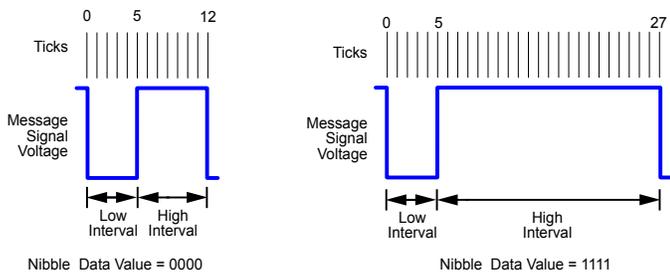


Figure 3. General value formulation for SENT nibble: (left) 0000, (right) 1111 (see table 1 for correspondence)

Table 1. Nibble Composition and Value

Quantity of Ticks per Nibble			Binary (4-Bit) Value	Decimal Equivalent Value
Low-Voltage Interval	High-Voltage Interval	Total		
5	7	12	0000	0
5	8	13	0001	1
5	9	14	0002	2
⋮	⋮	⋮	⋮	⋮
5	21	26	1110	14
5	22	27	1111	15

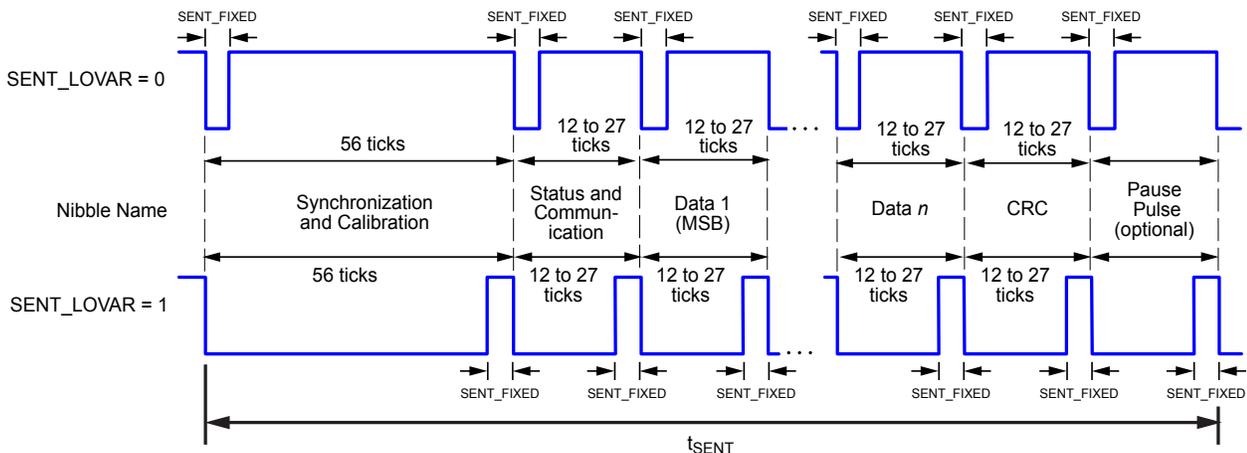


Figure 4. General format for SENT message frame: (upper panel) low state fixed, (lower panel) high state fixed

Table 2. SENT Message Frame Section Definitions

Section	Description
Synchronization and Calibration	
Function	Provide the external controller with a detectable start of the message frame. The large quantity of ticks distinguishes this section, for ease of distinction by the external controller.
Syntax	Nibbles: 1 Quantity of ticks: 56 Quantity of bits: 1
Status and Communication	
Function	Provides the external controller with the status of the device and indicates the format and contents of the Data section.
Syntax	Nibbles: 1 Quantity of ticks: 12 to 27 Quantity of bits: 4 1:0 Device status (set by SENT_STATUS parameter) 3:2 Message serial data protocol (set by SENT_SERIAL parameter)
Data	
Function	Provides the external controller with data selected by the SENT_SERIAL parameter.
Syntax	Nibbles: 3 to 6 Quantity of ticks: 12 to 27 (each nibble) Quantity of bits: 4 (each nibble)
CRC	
Function	Provides the external controller with cyclic redundancy check (CRC) data for certain error detection routines applied to the Data nibbles.
Syntax	Nibbles: 1 Quantity of ticks: 12 to 27 (each nibble) Quantity of bits: 4
Pause Pulse	
Function	(Optional) Additional time can be added at the end of a SENT message frame to ensure all message frames are of appropriate length. The SENT_UPDATE parameter sets format.
Syntax	Nibbles: 1 Quantity of ticks: 12 minimum (length determined by SENT_UPDATE option and by the individual structure of each SENT message) Quantity of bits: n.a.

Data Nibble Format

When transmitting normal operation data, information about the magnetic field is embedded in the first three Data nibbles (see figure 5). Each Data nibble consists of 4 bits with values ranging from 0 to 15. In order to present an output with the resolution of 12 bits, 3 Data nibbles are required. The Data nibble containing the MSB of the whole Data section is sent first.

Three additional optional Data nibbles can be associated with other parameters, by setting the parameter SENT_DATA:

- Counter – Each message frame has a serial number in each Counter nibble.

- Temperature – Temperature data from the device internal temperature sensor, in two's complement format, with MSB first:
 - All zeros = 25°C.
 - Temperature slope is always 0.8 LSB/°C, except for serial output protocol.
 - For serial output protocol, temperature slope = 0.5 LSB/°C.
- Inverted – The last nibble in the message frame is the first nibble, inverted (as an additional error check).

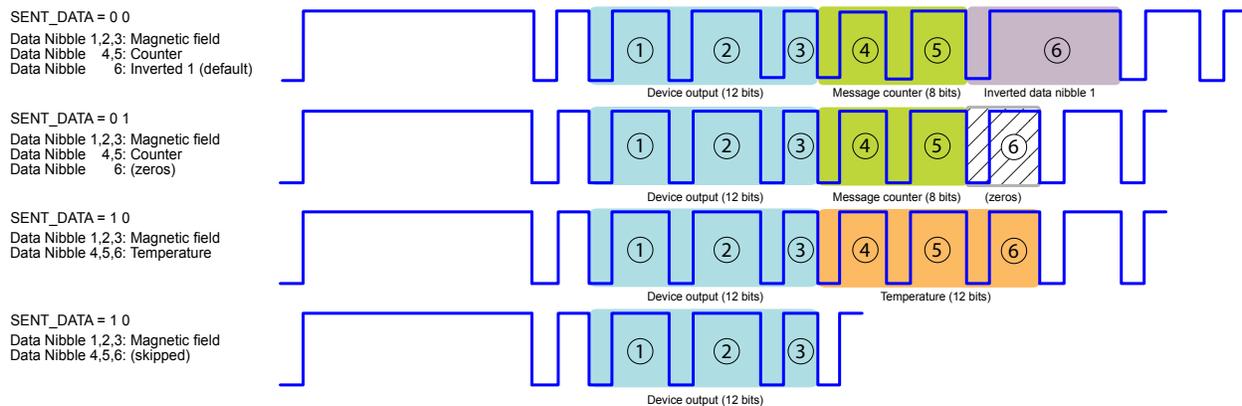


Figure 5. Options for SENT messages from the device (Slave), determined by the SENT_DATA field programmed value

Output Message Transfer

In the output stage of the sensor IC, signal samples proportional to the magnetic information are latched into the SENT converter and transferred to the user. The timing relationship, between the moment when magnetic information is latched into the SENT converter and the moment when the SENT message is transmitted to the user, falls into two types of SENT message transfers:

- Synchronous message transfer. Each SENT message is transmitted after new magnetic information (or multiples of magnetic information) reaches the SENT converter.
- Asynchronous message transfer. SENT messages are transmitted continuously, one after the other, not waiting for new magnetic information.

The SENT_UPDATE parameter determines the message transfer state:

- Asynchronous message transfer with variable SENT message duration—the device default state (SENT_UPDATE = 0). The output stage transmits the SENT messages independently of device internal output update rate (see figures 6 and 7). Allows

message frame duration to vary according to the contents; no Pause pulse is applied.

- Asynchronous data transfer with constant SENT message duration (SENT_UPDATE = 1). The output stage transmits the SENT messages independently of device internal output update rate (see figures 6 and 8). The Pause pulse is always inserted with a minimum nibble length of 12 ticks, but the nibble length is increased if the message is shorter than the maximum message length.
- Synchronous data transfer (SENT_UPDATE = 2) where the SENT message frame transmission rate is synchronized with the device internal output update rate (set by BW value) (see figures 6 and 9). If a particular message is shorter, a Pause pulse is inserted with a length that completes the message period.
- Asynchronous triggered message transfer (SENT_UPDATE = 3 or 4).

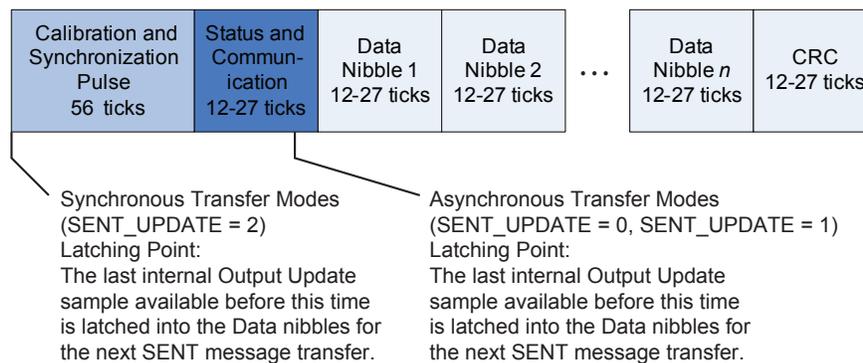
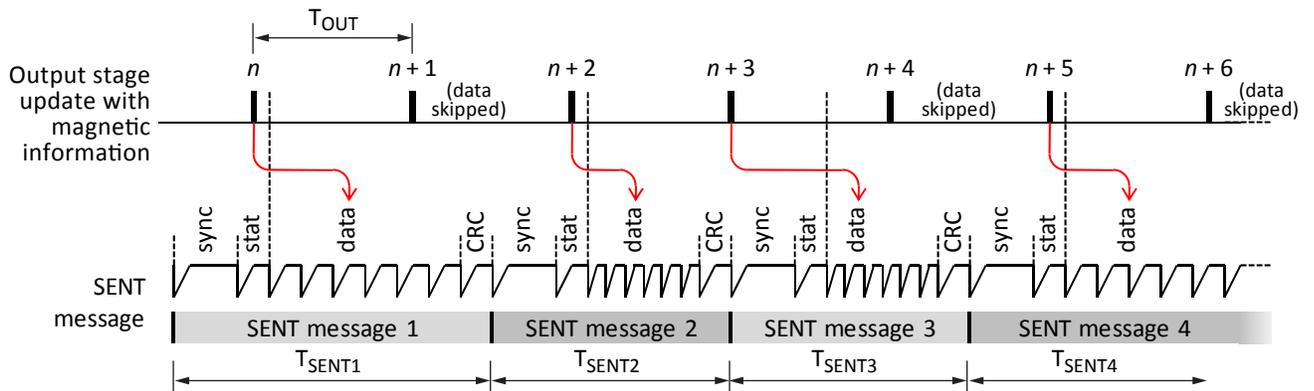
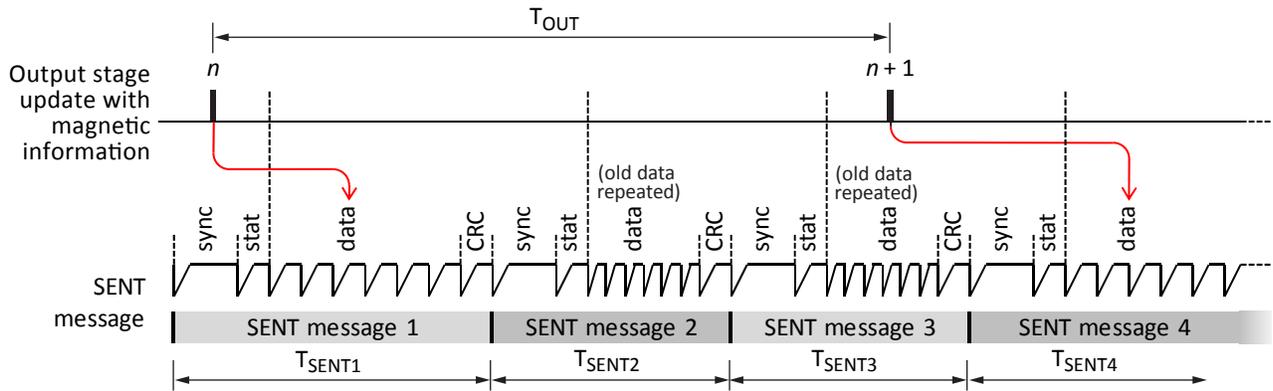


Figure 6. Latching Points for Available Data for SENT Message Data Nibbles

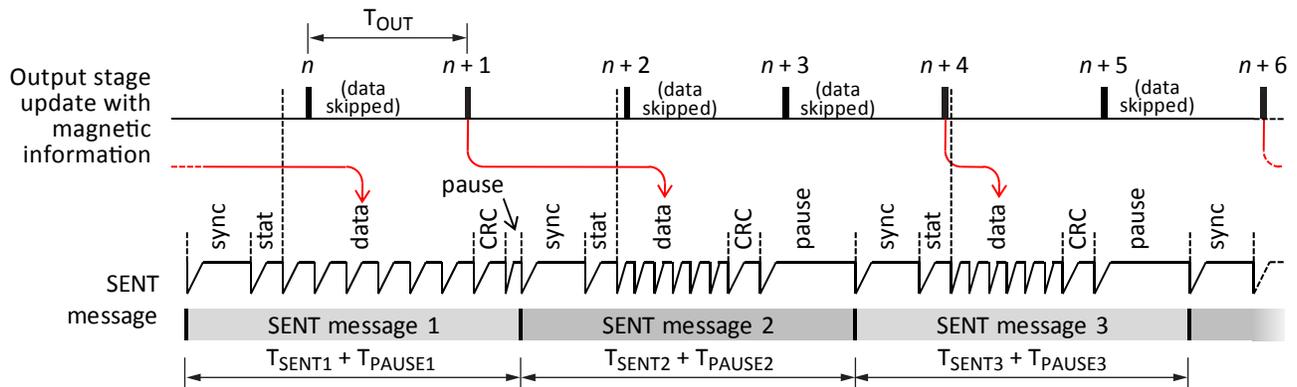


Panel 7(a). $T_{OUT} < T_{SENTx}$ (some data is not transmitted)

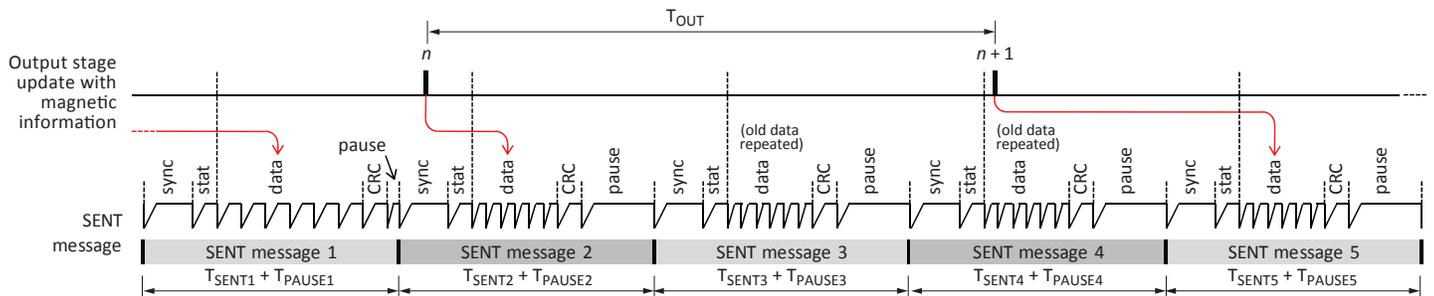


Panel 7(b). $T_{OUT} > T_{SENTx}$ (some data is repeated)

Figure 7. Messages do not contain a Pause pulse (SENT_UPDATE = 0), so the SENT message frame rate is not constant. The value transmitted in a message is taken from the last internal update ready before the first Data nibble of the message is composed. Therefore, individual internal updates may be skipped (panel a) or repeated (panel b), depending on the BW bandwidth and the message length defined by the SENT_TICK parameter setting.



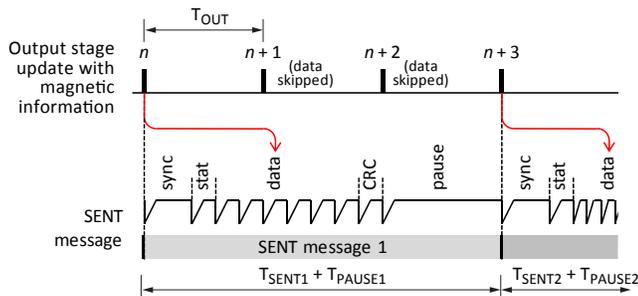
Panel 8(a). $T_{OUT} < T_{SENT} + T_{PAUSE}$ (some data is not transmitted)



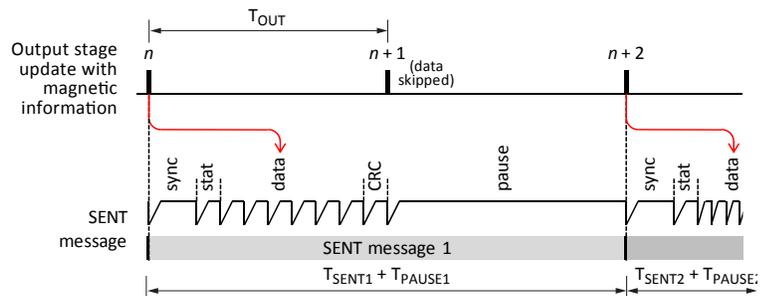
Panel 8(b). $T_{OUT} > T_{SENT} + T_{PAUSE}$ (some data is repeated)

Figure 8. A constant message frame rate is used, and for each message, a Pause pulse is used to extend the message to match the frame rate ($SENT_UPDATE = 1$). Internal updates may be skipped or repeated depending on the BW bandwidth and SENT message time settings. The quantity of skipped (panel a) or repeated (panel b) internal updates can vary from message to message.

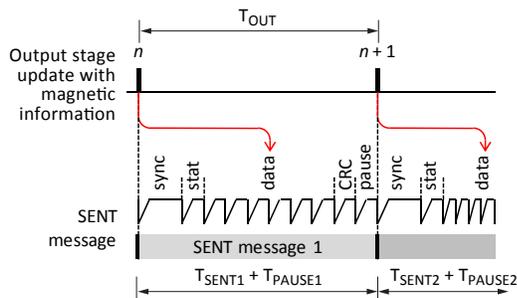
Note: Although the frame transmission rate is constant, discrete SENT messages do not represent equal time interval sampling of the magnetic field.



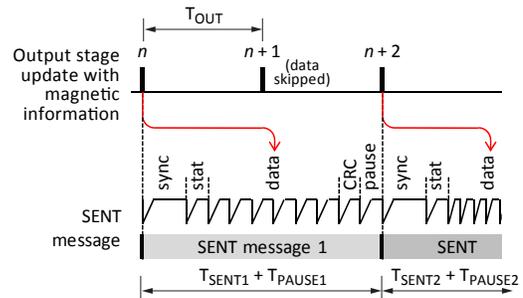
Panel 9(a). ($T_{OUT} < T_{SENT} + T_{PAUSE}$) The longest possible SENT message is synchronized at three times the internal update rate. The first update is ready before the Synchronization nibble is composed, and is transmitted. Three more updates occur before the next SENT message, so only the third update data is included, and the two intervening updates are skipped.



Panel 9(b). ($T_{OUT} < T_{SENT} + T_{PAUSE}$) The filter bandwidth is reduced by twice relative to the bandwidth in panel (a), which doubles the internal update interval. The longest possible SENT message is now synchronized at two times the internal update rate. The first update is ready before the Synchronization nibble is composed, and is transmitted. Two more updates occur before the next SENT message, so only the second update data is included, and the one intervening update is skipped.



Panel 9(c). ($T_{OUT} < T_{SENT} + T_{PAUSE}$) The internal update rate is the same as in panel (b), but the tick duration is reduced slightly. The longest possible SENT message is now synchronized at the internal update rate. Each update is ready before the Synchronization nibble is composed, and is transmitted. No updates are skipped.



Panel 9(d). ($T_{OUT} < T_{SENT} + T_{PAUSE}$) The faster update rate of panel (a) and the shorter tick duration of panel (c) are applied. Because the panel (d) higher bandwidth setting also applies, the overall device response time is faster than that shown in panel (c). However, the panel (c) settings reduce front-end noise better than those of panel (d), because of the lower bandwidth.

Figure 9. The SENT message rate is synchronized with the internal device internal update rate. For each message, a Pause pulse is used to extend the message to match the internal update rate ($SENT_UPDATE = 2$). A consistent number of updates are skipped (panels a, b, and d) from message to message. The internal update value transmitted is from the last update ready before the Synchronization and Calibration nibble of the message is composed.

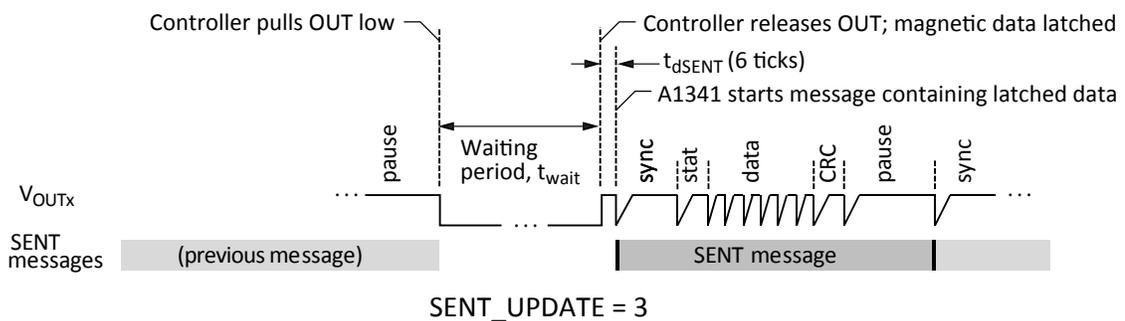
The SENT_UPDATE parameter has two other options which allow direct control of when magnetic field data is sent to the external controller:

- Tandem data latching and sending (SENT_UPDATE = 3)
- Immediate data latching with a controllable delay before sending (SENT_UPDATE = 4)

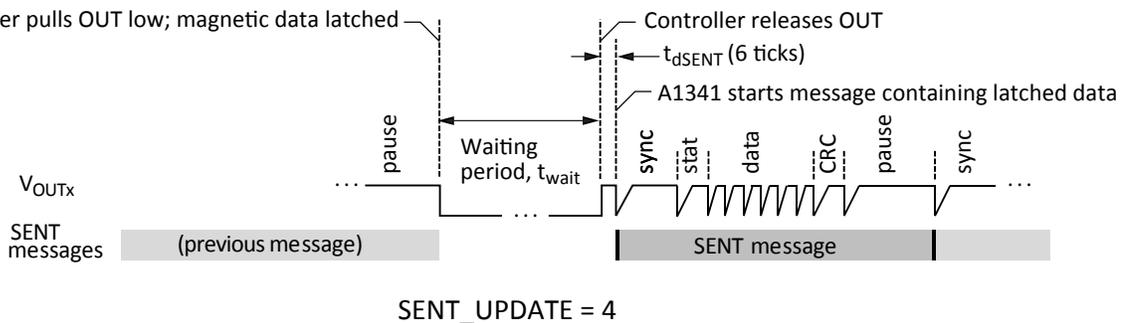
When SENT_UPDATE = 3 (upper panel in figure 10), while the sensor IC has a Pause pulse on the device output, the controller triggers a latch-and-send sequence by pulling the sensor IC output low. When the controller releases the output, the last processed signal (proportional to the magnetic field) is latched into the SENT converter, and after a delay of t_{dSENT} , the latched data is sent to the controller. This option is useful when the controller

requires a prompt response on the current magnetic field.

When SENT_UPDATE = 4 (lower panel in figure 10), while the sensor IC has a Pause pulse on the device output, the controller triggers a latch-and-send sequence by pulling the output low, which immediately latches the last processed signal (proportional to the magnetic field) into the SENT converter. This option allows the controller to postpone receiving the data. When the output is eventually released, the data is sent to the controller after a delay of t_{dSENT} . This option is useful where multiple sensor ICs are connected to the controller. All the sensor ICs can be instructed at the same time to latch magnetic field data, and the controller can then retrieve the data from each sensor IC individually.



Panel 10(a). SENT_UPDATE = 3



Panel 10(b). SENT_UPDATE = 4

Figure 10. Device output behavior where normal operation magnetic field data is latched at a defined time: (panel a) if SENT_UPDATE = 3, latched and sent at end of a low pulse, or (panel b) if SENT_UPDATE = 4, latched at the beginning of a low pulse, but not sent until the end of the pulse. The total delay from the beginning of the low pulse until the data message begins is: $t_{wait} + t_{dSENT}$.

Optional Serial Output Protocol

In the Status and Communication section, the data format selection can be:

- Normal device output (voltage proportional to applied magnetic field) in SENT protocol (SENT_SERIAL = 0).
- Augmented data on the magnetic parameters and device settings, in an optional Serial Output protocol (SENT_SERIAL = 1, 2, or 3). Any of these three protocols enables transmission of values from the following EEPROM parameters, in the following order:

Message ID (4 or 8 bits)	Data (8, 12, or 16 bits)
0	Corrected temperature
1	SENS_COARSE
2	SIG_OFFSET
3	QOUT_FINE
4	SENS_MULT
5	CLAMP_HIGH
6	CLAMP_LOW
7	DEVICE_ID (1341 ₁₀ or 1343 ₁₀ , per device)

- Additional Short serial protocol (SENT_SERIAL = 1). Has a message payload of 12 bits: 8 bits are for value data, and 4 bits for the message ID (identification). A total of 16 separate SENT messages are required to transmit the entire data group.
- Additional Enhanced 16-bit serial protocol (SENT_SERIAL = 2). Has 12 bits for value data, and 4 bits for the message ID. A total of 18 SENT messages are required to transmit the entire data group.
- Additional Enhanced 24-bit serial protocol (SENT_SERIAL = 3). Has 16 bits for value data, and 8 bits for the message ID. A total of 18 SENT messages are required to transmit the entire data group.

Device Response Time

The Device Response Time depends on three factors:

- Propagation Delay – This is the traveling time of the signal from the Input Stage of the Hall device to the Output Stage.
- Output Message Transfer – Synchronous and asynchronous.
- SENT Message Length – The various choices for the SENT message configuration give different SENT message lengths.

These three factors are applied sequentially, as illustrated in figure 11.

Propagation Delay and Output Update Rate

Propagation Delay and the Output Update Rate depend greatly on the device internal filter bandwidth. The bandwidth is set by programming the BW field in EEPROM. The correspondence of programmed value with Propagation Delay and Output Update Rate is given in table 3.

Table 3. Bandwidth Settings and Outcomes

Programming Code, BW	3-dB Bandwidth (kHz)	Maximum Propagation Delay (ms)	Output Stage Update Frequency (kHz)
0	1.5	0.63	8
1	3	0.37	16
2	1.5	0.63	8
3	0.750	1.26	4
4	0.375	2.52	2
5	0.188	5.04	1
6	0.094	10.08	0.500
7	0.047	20.16	0.250

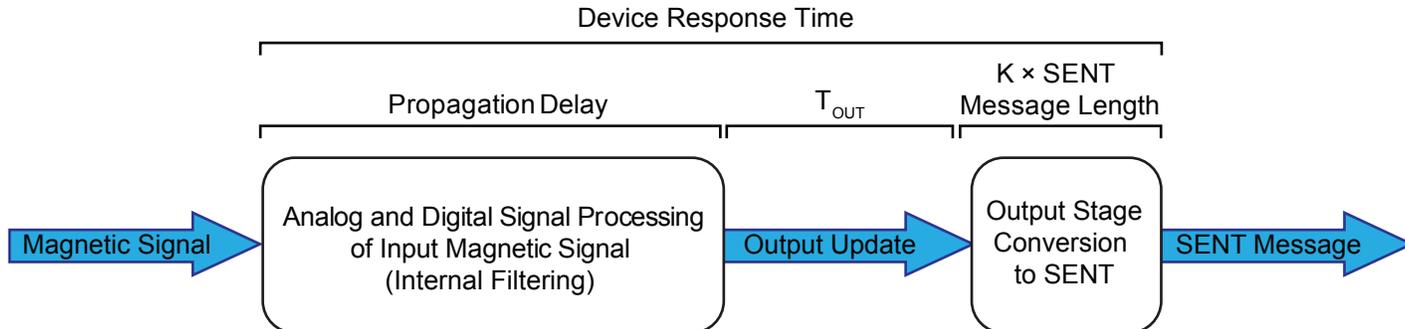


Figure 11. Model of Overall SENT Response Time. The summation of the significant processes is expressed in the following equation:

$$\text{Device Response Time} = \text{Propagation Delay} + T_{\text{OUT}} + K \times \text{SENT Message Length}$$

where

T_{OUT} is the period of the Output Update rate, and

K is the coefficient determined by the moment of the Output Stage update

Asynchronous Transfer Minimum Response Time

The shortest device response time is realized when the Output Update sample appears immediately before a new SENT message is configured. In asynchronous mode, this can occur later in the SENT message period, up to the Status and Communication bit, as shown in figure 12.

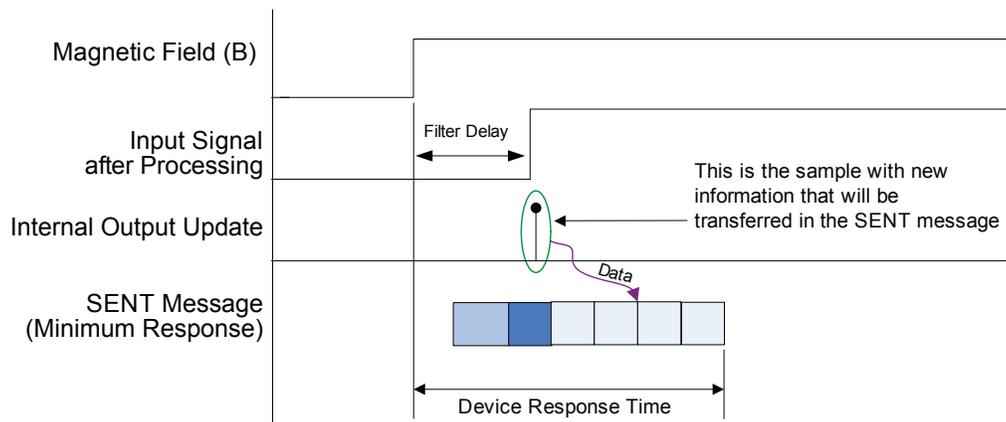
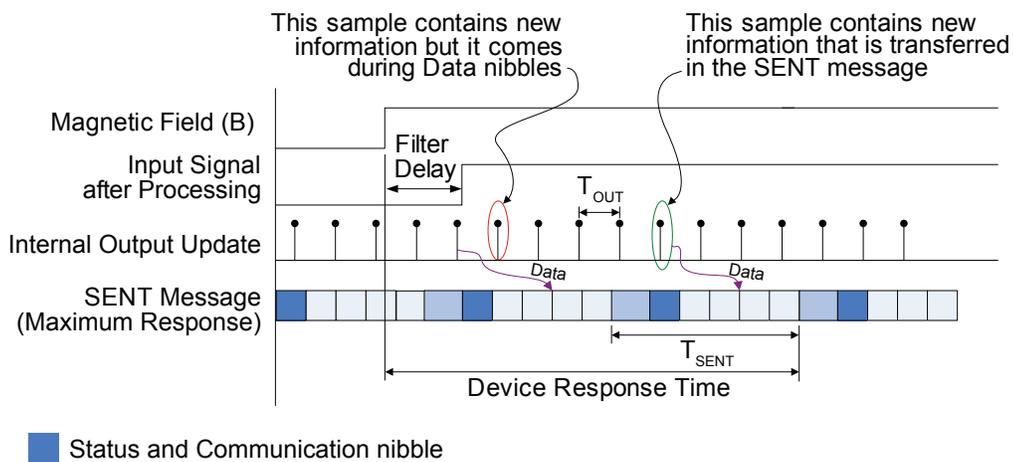


Figure 12. Minimum Device Response Time, Asynchronous Transfer mode

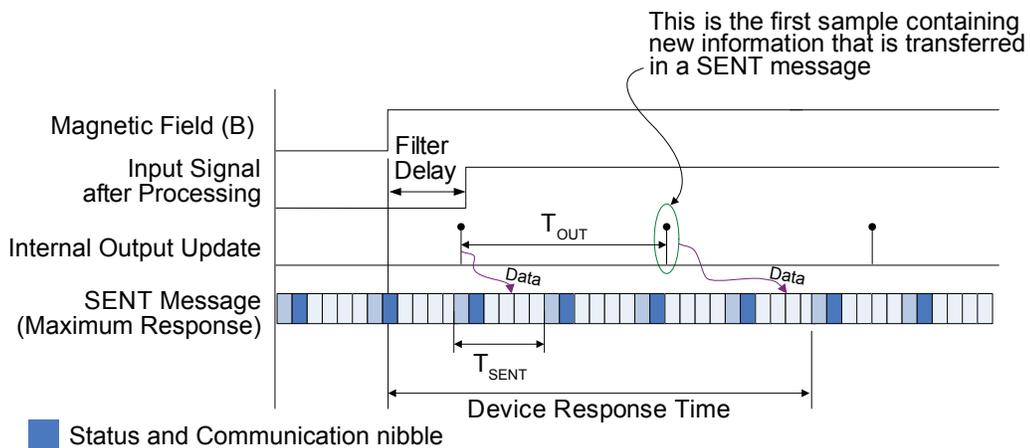
$$\text{Minimum Device Response Time} = \text{Filter Delay} + 3 \times \text{Data Nibble} + \text{CRC}$$

Asynchronous Transfer Maximum Response Time

With Asynchronous Transfer selected, the longest device response time is realized when an internal Output Update sample appears immediately before the Filter Delay period ends, and the next Status and Communication nibble ends before the next sample occurs, as shown in figure 13. Latching of the sample occurs near the end of the Status and Communication nibble.



Panel 13(a). $T_{OUT} < T_{SENT}$



Panel 13(b). $T_{OUT} > T_{SENT}$

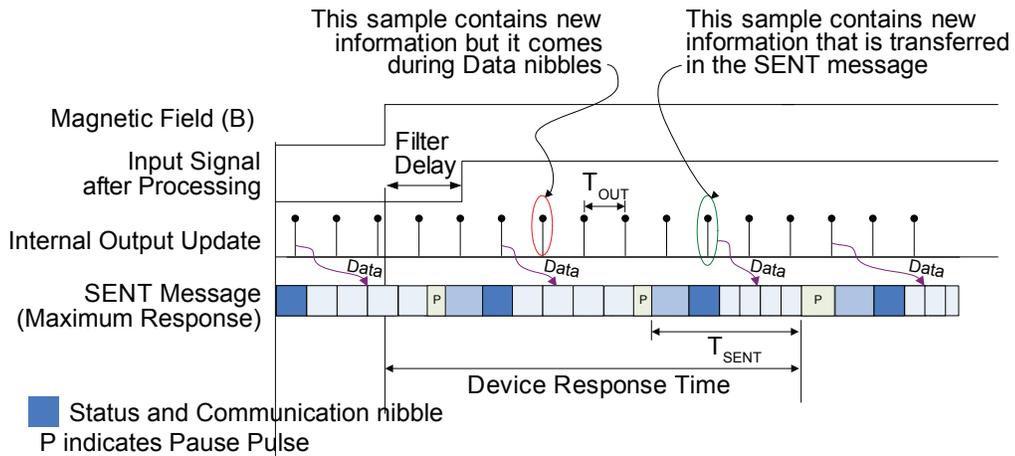
Figure 13. Maximum Device Response Times Compared for $T_{OUT} < T_{SENT}$ (panel a) and $T_{OUT} > T_{SENT}$ (panel b)

$$\text{Maximum Device Response Time} = \text{Filter Delay} + T_{OUT} + 3 \times \text{Data Nibble} + \text{CRC} + T_{SENT}$$

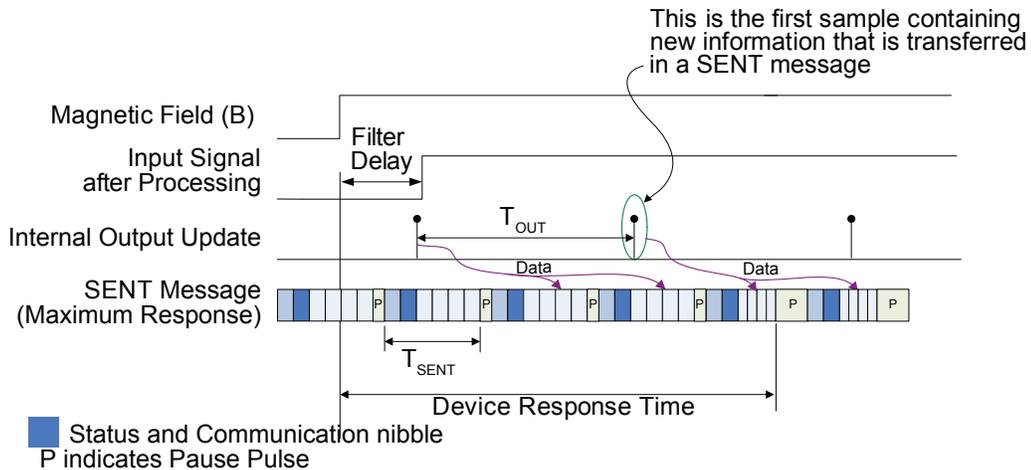
Asynchronous Transfer with SENT Messages of Equal Duration Maximum Response Time

With Asynchronous Transfer selected, and use of a Pause Pulse is enabled to ensure all SENT messages are of the same duration, the longest device response time is realized when an internal Output Update sample appears immediately before the Filter Delay

period ends, and the next Status and Communication nibble ends before the next sample occurs, as shown in figure 14. Latching of the sample occurs near the end of the Status and Communication nibble.



Panel 14(a). $T_{OUT} < T_{SENT} + T_{PAUSE}$



Panel 14(b). $T_{OUT} > T_{SENT} + T_{PAUSE}$

Figure 14. Maximum Device Response Times Compared for $T_{OUT} < T_{SENT} + T_{PAUSE}$ (panel a) and $T_{OUT} > T_{SENT} + T_{PAUSE}$ (panel b). Note: For purposes of comparison, the total length of the equal length SENT messages is for messages having the maximum number of ticks in each section and a minimum Pause Pulse of 12 ticks.

$$\text{Maximum Device Response Time} = \text{Filter Delay} + T_{OUT} + 3 \times \text{Data Nibble} + \text{CRC} + T_{PAUSE} + T_{SENT}$$

Synchronous Transfer Mode Minimum Response Time

The shortest device response time is realized when the Output Update sample appears immediately before a new SENT message is configured. In synchronous mode, this must occur simultaneously with the start of the Synchronization bit, as shown in figure 15.

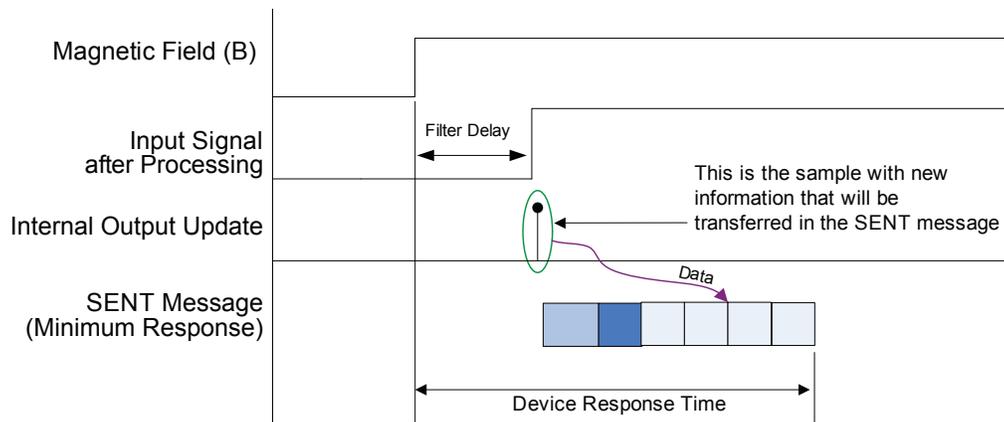
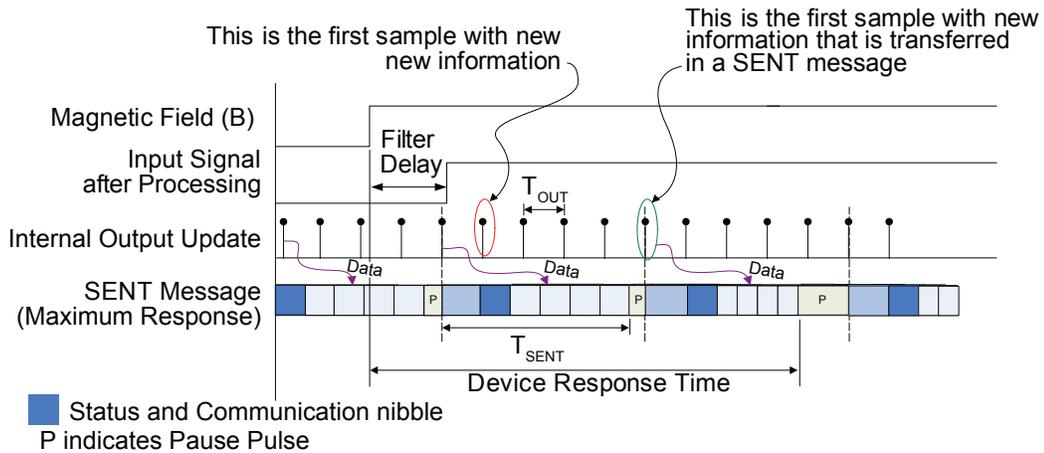


Figure 15. Minimum Device Response Time, Synchronous Transfer mode

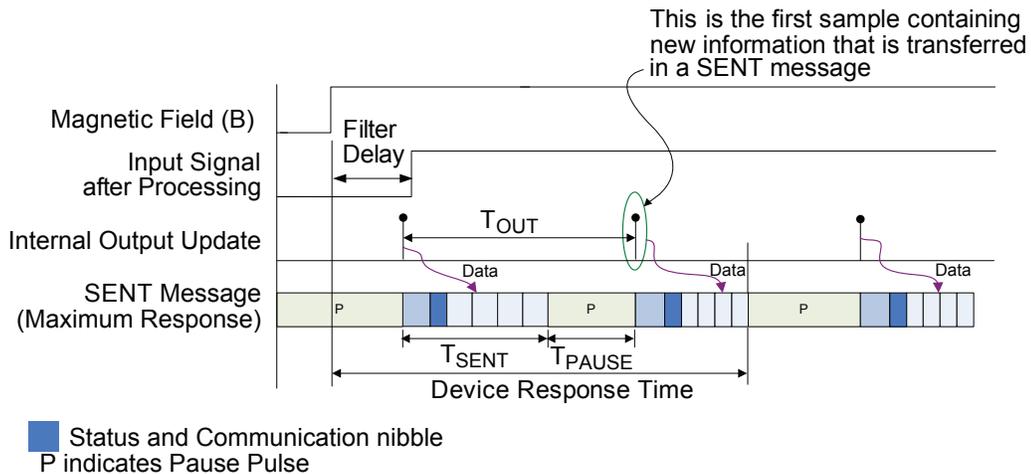
$$\text{Minimum Device Response Time} = \text{Filter Delay} + T_{\text{SENT}}$$

Synchronous Transfer Maximum Response Time

With Synchronous Transfer selected, the longest device response time is realized when an internal Output Update sample appears immediately before the Filter Delay period ends, and the next Status and Communication nibble ends before the next sample occurs, as shown in figure 16. Latching of the sample occurs near the end of the Status and Communication nibble.



Panel 16(a). $T_{OUT} < T_{SENT} + T_{PAUSE}$; Maximum Device Response Time = Filter Delay + $T_{SENT} + T_{PAUSE} + T_{OUT}$



Panel 16(b). $T_{OUT} = T_{SENT} + T_{PAUSE}$; Maximum Device Response Time = Filter Delay + $T_{SENT} + T_{OUT}$

Figure 16. Maximum Device Response Times Compared for $T_{OUT} < T_{SENT} + T_{PAUSE}$ (panel a) and $T_{OUT} > T_{SENT} + T_{PAUSE}$ (panel b). Note: For purposes of comparison, the total duration of the equal SENT messages is for messages having the maximum number of ticks in each section and a Pause Pulse of 12 ticks or more, satisfying the equation: $T_{SENT} + T_{PAUSE} = n \times T_{OUT}$, where n is an integer number.

Device Response Time with Continuous Field Change

In the case where the applied magnetic field is continuously changing and the application requires the device output to track the magnetic field closely:

- The Initial Response Delay can be treated the same as a device response to a magnetic step function. The Initial Response Delay can be as long as the Maximum Response Time.
- After the Initial Response Delay, updates reflecting the continuous change are transferred with every SENT message.

These considerations are represented in figure 17. The Response Delay for field A represents the minimum step response.

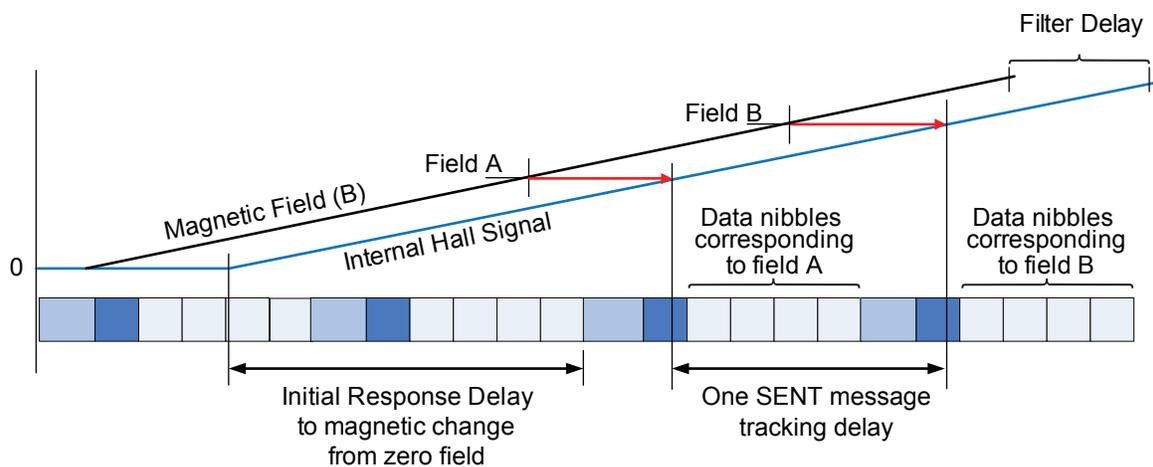


Figure 17. Device Response Time Characteristics for Device in Continuously Changing Magnetic Field

Fast SENT Feature

The Allegro proprietary programmable Fast SENT feature includes:

- Minimum clock rate: 0.25 μ s. This can be achieved by programming parameter SENT_TICK, register 7, bits 17:11, to code 1.
- Minimum quantity of fixed ticks for low-voltage interval: 4 ticks. This can be achieved by programming SENT_FIXED, register 7, bits 10:9, to code 1.
- Number of Data nibbles: 3. This can be achieved by programming SENT_DATA, register 7, bits 4:3.
- Default update rate: One message after another, with no pause pulse. Accept the default for SENT_UPDATE, register 7, bits 7:5, default 0.
- Serial data: No serial data. Accept the default for SENT_SERIAL, register 7, bits 1:0, default 0.

Minimum Message Length

The shortest SENT message contains 6 sections, as illustrated in figure 18. The shortest duration of a SENT message can be calculated using the following equation:

$$\begin{aligned} \text{Minimum Message Length} = & (\text{Synchronization and} \\ & \text{Calibration Pulse} \\ & + \text{Status and Communication} \\ & + \text{Data Nibble} \times 3 \\ & + \text{CRC}) \\ & \times \text{Tick Time} \end{aligned}$$

The Tick Time is set by programming the SENT_TICK field in EEPROM. Tick Time is the internal 4 MHz count divided by the SENT_TICK setting. The correspondence of programmed value with Tick Time is given in table 4.

The shortest Tick Time is 0.25 μ s. Given the Minimum Message Length, as defined above, and the maximum ticks as shown in figure 9, the shortest SENT message duration is:

$$\begin{aligned} \text{Minimum Message Length} = & (56 \text{ ticks} \\ & + 27 \text{ ticks} \\ & + 27 \text{ ticks} \times 3 \\ & + 27 \text{ ticks}) \\ & \times 0.25 \\ = & 191 \times 0.25 = 47.75 \mu\text{s} \end{aligned}$$

Table 4. Tick Settings and Outcomes

Programming Code SENT_TICK	Tick Time (4 MHz / SENT_TICK) (μ s)
0 (default)	3
1	0.25
2	0.5
⋮	⋮
12	3
⋮	⋮
127	31.75



Figure 18. Model of Shortest Valid SENT Message

A comparison of the default SENT message transmission rate and the Fast SENT rate is shown in figure 19. Approximately 12 Fast SENT messages can be sent in the same time period as one message at the default rate.

An expanded view of one Fast SENT message is provided in figure 20. Including a payload of three data nibbles, the total elapsed time is approximately 33 μ s.



Figure 19. Comparison of time required to output (top) a default SENT mode message, and (bottom) a Fast SENT mode message

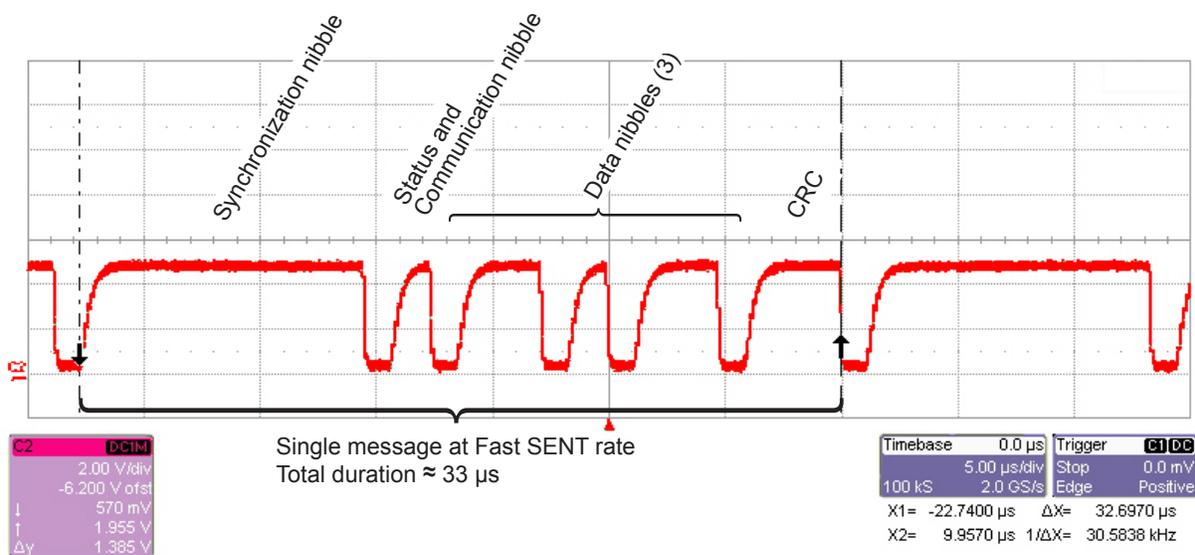


Figure 20. Expanded view of a single Fast SENT mode message

Device Response Time Example Calculation

A comparison of typical minimum and maximum device response times is presented in table 5. The results are based on the following assumptions:

- Tick Time = 0.25 μs
- Bandwidth = 3 kHz
- $T_{\text{OUT}} = 1 / 16 \text{ kHz} = 62.5 \mu\text{s}$
- Internal Filter Delay = 350 μs

- Length of 4 data nibbles = 27 ticks (every nibble has the maximum number of ticks)
- Maximum Message Length $T_{\text{SENT}} = 47.75 \mu\text{s}$ (every SENT message has the maximum message length, and each message section has the maximum number of ticks)
- Maximum Device Response Time formula applied is for condition where $T_{\text{OUT}} > T_{\text{SENT}} (+ T_{\text{PAUSE}})$

Table 5. Comparative Response Times

	Asynchronous Transfer (μs)	Asynchronous Transfer with equal SENT Duration (μs)	Synchronous Transfer (μs)
Minimum Response Times	Internal Filter Delay + 4 data nibbles = 350 + 27 = 377	Internal Filter Delay + 4 data nibbles = 350 + 27 = 377	Internal Filter Delay + SENT message = 350 + 47.75 = 397.75
Maximum Response Times	Internal Filter Delay + SENT message + Synch pulse + Status/Communication = 350 + 47.75 + 20.75 = 418.5	Internal Filter Delay + SENT message + Synch pulse + Status/Communication = 350 + 47.75 + 20.75 = 418.5	Internal Filter Delay + T_{OUT} = 350 + 62.5 = 412.5

Trigger Mode Fast SENT Feature

Trigger mode can be applied to the Allegro Fast SENT feature by setting the SENT_UPDATE field to 3 or 4. When the message should be transmitted, the device output must be pulled low for a minimum interval of 2 ticks, and then pulled high. After 6 ticks have expired at the high level, the SENT message is transmitted.

In preparation for transmission, the sample data is latched at the end of the Status and Communication nibble. It is then sent at the beginning of a Trigger pulse (SENT_UPDATE set to 3) or at the

end of the pulse (SENT_UPDATE set to 4).

The actual response time depends on the relative timing of the internal output update and the latching (figure 21):

- The minimum response time occurs when the output data was latched immediately after a new internal sample emerged.
- The maximum response time occurs when the output data was latched immediately before a new internal sample emerged.

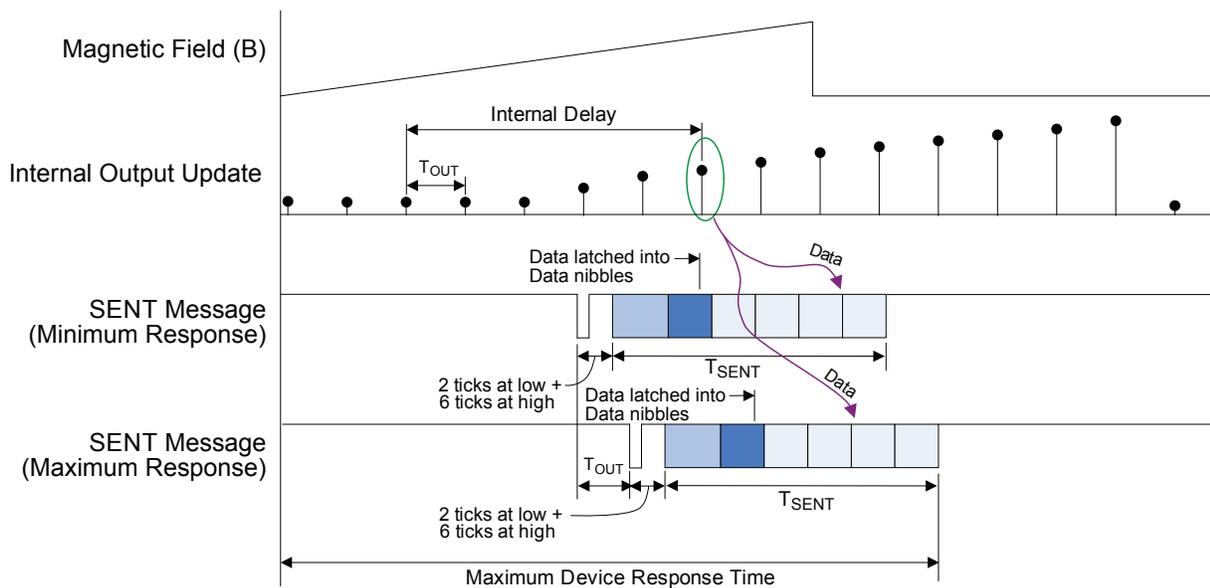


Figure 21. Sensor IC response characteristics using triggered Fast SENT mode

Calculation assuming following parameters:

- Tick time of 0.25 μ s
- Message format of 3 data nibbles with maximum length of 27 ticks
- Device internal bandwidth of 3000 Hz
- Propagation delay of 350 μ s

Minimum Response Time \approx Propagation Delay + Length of 4 Data Nibbles = 350 + 27 = 377 μ s

Maximum Response Time \approx Propagation Delay + T_{OUT} + Length of 4 Data Nibbles (3 Data and 1 CRC) = 350 + 62.5 + 27 = 439.5 μ s

Trigger Mode with Two Sensor ICs

Trigger mode can be applied to compare the simultaneous output of two Allegro devices. This feature allows dual sources to be used without any requirement to synchronize the clocking of the Allegro devices.

The actual response time of each of the devices depends on the independent relative timing of the internal sampling cycle and

the latching (see the Trigger Mode Fast SENT feature section). If the two devices receive the Trigger pulse at the same time, the internal timing can lead to a maximum difference defined by the period of the output update signal between the actual sample acquisitions of the two devices. As shown in figure 22, the effect is that different sample points can be used for the data output.

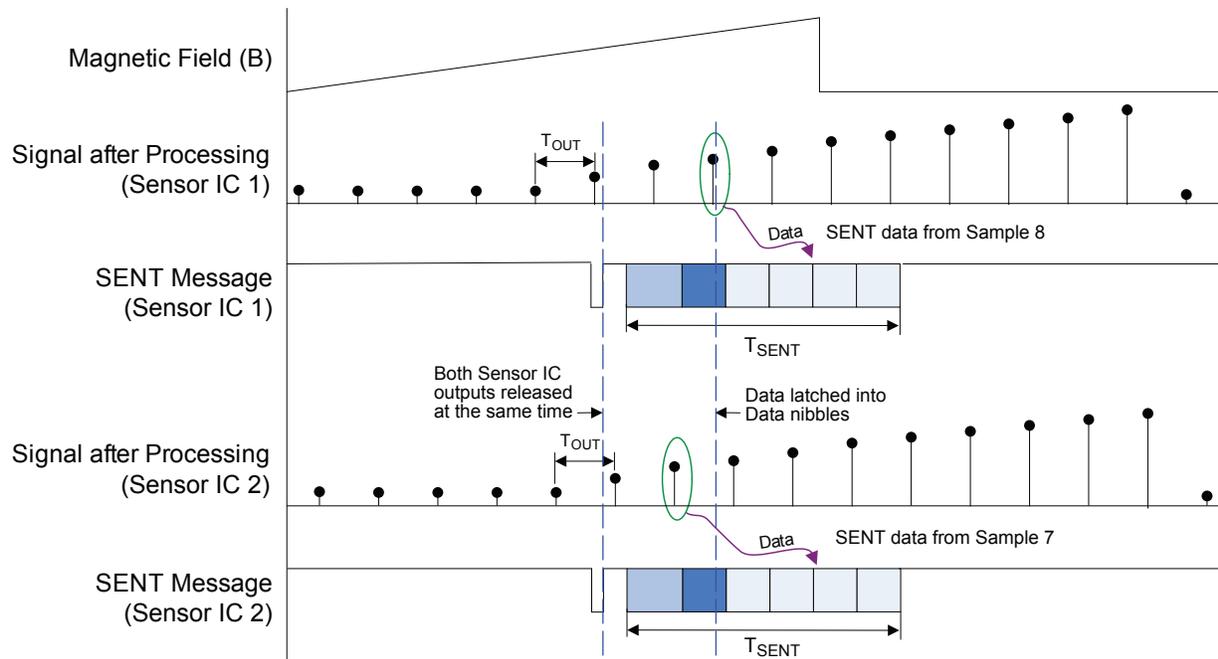


Figure 22. Sensor IC differential response characteristics using two Sensor ICs in Trigger mode

Adjustable Nibble Fall Time

The timing of the nibble fall time can be adjusted by a combination of an external capacitor and the value programmed for the OUTDRV_CFG parameter. The value of an external capacitor, C_{LOAD} , on the the OUT pin sets the coarse range for the fall time. Within that range, a fine setting is determined by the OUTDRV_DFG programmed code, according to table 6.

Table 6. Nibble Fall Time Values (OUTDRV_CFG)

Values	Code	Fall Time (Typical) (μ s)		
		$C_{LOAD} = 100$ pF	$C_{LOAD} = 1$ nF	$C_{LOAD} = 10$ nF
	000 (Default)	0.048	0.149	1.324
	001	0.114	0.217	1.323
	010	0.202	0.309	1.404
	011	0.290	0.400	1.492
	100	0.760	0.854	1.948
	101	1.539	1.555	2.669
	110	3.161	2.978	4.118
	111	4.819	4.442	5.557

NOTE: Values are based on design simulations. Lower values have been obtained in actual benchtop tests.

Electrical Specifications

Typical Allegro device specifications are given in table 7.

Table 7. OPERATING CHARACTERISTICS Valid through full operating temperature range, T_A , and supply voltage, V_{CC} , $C_{BYPASS} = 10 \text{ nF}$, unless otherwise specified

Characteristics	Symbol	Test Conditions	Min.	Typ.	Max.	Unit
General Electrical Characteristics¹						
SENT Message Duration	t_{SENT}	Tick time = 3 μs , 3 data nibbles of information, nibble length = 27 ticks	–	573	–	μs
Minimum Programmable SENT Message Duration	$t_{SENTMIN}$	Tick time = 0.25 μs , 3 data nibbles of information, nibble length = 27 ticks	–	47	–	μs
SENT Programmable Characteristics¹						
SENT Output Signal ^{2,3}	$V_{SENT(L)}$	$10 \text{ k}\Omega \leq R_{pullup} \leq 50 \text{ k}\Omega$	–	–	0.05	V
	$V_{SENT(H)}$	Minimum $R_{pullup} = 10 \text{ k}\Omega$	$0.9 \times V_{CC}$	–	–	V
		Maximum $R_{pullup} = 50 \text{ k}\Omega$	$0.7 \times V_{CC}$	–	–	V
SENT Output Trigger Signal	$V_{SENTtrig(L)}$		–	–	1.2	V
	$V_{SENTtrig(H)}$		2.8	–	–	V

¹ Determined by design.

² For pull-up values lower than 10 k Ω , $V_{SENT(L)}$ will be higher and can be calculated as: $V_{SENT(L)} = V_{PULL-UP} \times [60 (\Omega) / (60 (\Omega) + R_{PULL-UP})]$. Therefore, for $R_{PULL-UP} = 500 \Omega$, and $V_{PULL-UP} = 5 \text{ V}$, low voltage will be a minimum of 535 mV.

³ For pull-up values lower than 10 k Ω , $V_{SENT(H)}$ will be higher than $0.9 \times V_{CC}$.

SENT Data Programming Parameters

OUTDRV_CFG (Register Address: 0x07, bits 20:18)

Function	Output Signal Configuration Sets configuration of the output signal slew-rate control. Sets the ramp rate on the gate of the output driver, thereby changing slew rate at the output.			
Syntax	Quantity of bits: 3			
Related Commands	–			
Values	Code	Fall Time (Typical) (μ s)		
		$C_{LOAD} = 100$ pF	$C_{LOAD} = 1$ nF	$C_{LOAD} = 10$ nF
	000 (Default)	0.048	0.149	1.324
	001	0.114	0.217	1.323
	010	0.202	0.309	1.404
	011	0.290	0.400	1.492
	100	0.760	0.854	1.948
	101	1.539	1.555	2.669
110	3.161	2.978	4.118	
111	4.819	4.442	5.557	
Options	NOTE: Fall Time values are based on design simulations. Lower values have been obtained in actual benchtop tests.			
Examples	–			

SENT_DATA (Register Address: 0x07, bits 4:3)

Function	Data Nibble Format Quantity and contents of Data nibbles in message. (Does not relate to data contained in the Status and Communication nibble.)
Syntax	Quantity of Bits: 2
Related Commands	–
Values	<p>0 0: Nibbles 1,2,3: magnetic field data; nibbles 4,5: counter data; nibble 6: inverted nibble 1 (Default)</p> <p>0 1: Nibbles 1,2,3: magnetic field data; nibbles 4,5: counter data; nibble 6: all zeros</p> <p>1 0: Nibbles 1,2,3: magnetic field data; nibbles 4,5,6: current temperature data</p> <p>1 1: Nibbles 1,2,3: magnetic field data (nibbles 4,5,6 skipped)</p>
Options	–
Examples	–

SENT_FIXED (Register Address: 0x07, bits 10:9)

Function	Fixed Interval Duration Indicates the quantity of ticks in fixed-duration intervals.
Syntax	Quantity of Bits: 2
Related Commands	SENT_LOVAR
Values	0 0: 5 ticks (Default) 0 1: 4 ticks 1 0: 7 ticks 1 1: 8 ticks
Options	SENT_FIXED = 1 (4 ticks) does not meet the SENT spec, but is provided for custom fast or improved-EMI communication.
Examples	–

SENT_LOVAR (Register Address: 0x07, bit 8)

Function	State Assignments Assigns fixed duration state (becomes delimiting state; other interval becomes the information state)
Syntax	Quantity of Bits: 1
Related Commands	SENT_FIXED
Values	0: Low interval of every nibble is fixed in duration, and the high interval becomes the information state (Default). 1: High interval of every nibble is fixed in duration, and the low interval becomes the information state.
Options	SENT_LOVAR = 0 meets the SENT specification. SENT_LOVAR = 1 does not meet the SENT spec, but is provided for custom improved-EMI communication. For SENT_UPDATE = 3 or 4, the Pause pulse has a fixed low time regardless of the SENT_LOVAR setting.
Examples	–

SENT_SERIAL (Register Address: 0x07, bits 1:0)

Function	Status and Communication Nibble Format Defines values of bits 2 and 3 inside the Status and Communication nibble.
Syntax	Quantity of Bits: 2
Related Commands	–
Values	<p>0 0: Bits 2 and 3 are 0 (Default).</p> <p>0 1: Bits 2 and 3 are 0 part of the Short Serial protocol: 8-bit value data, 4-bit message ID, 16 SENT frames are required to send an entire serial message.</p> <p>1 0: Bits 2 and 3 are part of the Enhanced 16-bit Serial protocol: 12-bit value data, 4-bit message ID, 18 SENT frames are required to send an entire serial message.</p> <p>1 1: Bits 2 and 3 are part of the Enhanced 24-bit Serial protocol: 16-bit value data, 8-bit message ID, 18 SENT frames are required to send an entire serial message.</p>
Options	–
Examples	–

SENT_STATUS (Register Address: 0x07, bit 2)

Function	Error Condition Status Defines values of bits 0 and 1 inside the Status and Communication nibble. Defines data inside the Status and Communication nibble on device error status.
Syntax	Quantity of Bits: 1
Related Commands	SENT_SERIAL
Values	<p>(SENT_STATUS = 0)</p> <p>0 0: No error (Default)</p> <p>0 1: Not used</p> <p>1 0: Overvoltage condition</p> <p>1 1: Nonrecoverable EEPROM error, bad Linearization table or other error</p> <p>(SENT_STATUS = 1)</p> <p>0 0: No error (Default)</p> <p>0 1: Error condition</p>
Options	–
Examples	A Status and Communication nibble value of 0010 indicates an overvoltage condition.

SENT_TICK (Register Address: 0x07, bits 17:11)

Function	Tick Duration Sets the SENT tick rate coefficient: $4 \text{ MHz} / \text{SENT_TICK} = \text{tick} (\mu\text{s})$		
Syntax	Quantity of Bits: 7 Any value from 0 to 127 can be used		
Related Commands	–		
Values	Code	PWM Frequency (Typical) (μs)	Coefficient (MHz/SENT_TICK)
	000 0000	3.0 (Default)	4/12
	000 0001	0.25	4/1
	000 0010	0.5	4/2
	000 0111	0.75	4/3
	111 1111	32	4/125
	111 1110	31.5	4/126
Options	SENT_TICK = 1 through 11 do not meet the SENT spec, but are provided for custom fast communication.		
Examples	–		

SENT_UPDATE (Register Address: 0x07, bits 7:5)

Function	Pause Pulse and Frame Rate Pause pulse usage and message frame rate.
Syntax	Quantity of Bits: 3
Related Commands	SENT_LOVAR
Values	<p>000: No Pause pulse; new frame immediately follows previous frame (Default).</p> <p>001: Pause pulse used for minimum constant frame rate (Length of other message sections, plus length of Pause Pulse nibble, is constant. For the maximum message length, Pause pulse information state is the minimum size of 12 ticks.)</p> <p>010: Pause pulse used for constant frame rate, synchronized with device internal update rate. (Handshaking occurs such that the Synchronization and Calibration nibble starts immediately after the next new data word is ready.)</p> <p>011: Pause pulse held indefinitely until receipt of trigger pulse (OUT pulled low) from the controller, data latched after output released and message is sent.</p> <p>100: Pause pulse held indefinitely until receipt of trigger pulse (OUT pulled low) from the controller, data latched immediately and sent when output is released.</p> <p>101, 110, 111: Same function as 000.</p>
Options	–
Examples	–

Revision History

Number	Date	Description
–	November 2, 2015	Initial Release
1	May 25, 2017	Updated Table 2: SENT Message Frame Section Definitions, CRC Function description (page 5).
2	September 27, 2018	Minor editorial updates
3	October 18, 2019	Minor editorial updates

Copyright 2019, Allegro MicroSystems.

The information contained in this document does not constitute any representation, warranty, assurance, guaranty, or inducement by Allegro to the customer with respect to the subject matter of this document. The information being provided does not guarantee that a process based on this information will be reliable, or that Allegro has explored all of the possible failure modes. It is the customer's responsibility to do sufficient qualification testing of the final product to ensure that it is reliable and meets all design requirements.

Copies of this document are considered uncontrolled documents.

For the latest version of this document, visit our website:

www.allegromicro.com