

TB9M003FG

Case Study on Improving Motor Control Accuracy Using an Internal Oscillator

Description

Purpose and Scope of this Document

This document introduces, using a motor control example, a method for compensating for the effects of the internal oscillator accuracy of the SmartMCD™ TB9M003FG on system control by means of application software (SW).

We hope this document will serve as a useful reference when designing systems in which motor speed is a critical requirement.

In addition, this document assumes that the reader is already making use of the reference software (Note) provided by our company, and the explanations are given based on this premise.

Intended Audience of This Document

This document is intended for users who develop the above motor control system by utilizing the reference software provided by our company in combination with the TB9M003FG.

(Note) For further inquiries, please contact our sales department.

Toshiba Electronic Devices & Storage Corporation

Table of Contents

Description 1

Purpose and Scope of this Document 1

Intended Audience of This Document 1

Table of Contents 2

1. Purpose and Intention of This Document..... 4

2. Effect of Internal Oscillator Frequency Accuracy on Motor Control..... 4

 2.1. Evaluation of Motor Control Operation under Internal Oscillator Frequency Error..... 4

 2.1.1. Circuit for Generating Internal Oscillator Frequency Error ($\pm 5\%$) 4

 2.1.2. Evaluation of Motor Control under $\pm 5\%$ Frequency Variation 4

 2.1.3. Results of Evaluation 1-a (Effect on Motor Speed Relative to the Speed Command) 5

 2.1.4. Evaluation 1-b Results (Effects on PWM Period, VE Processing/Interrupt Time, and Dead Time)..... 6

3. Case Study of Motor Control Compensation for Internal Oscillator Frequency Error 8

 3.1. Method for Identifying Internal Oscillator Frequency Error (SW1 Case Study)..... 8

 3.1.1. Method for Measuring Internal Oscillator Frequency (Using LIN Communication) 9

 3.1.2. Notes and Considerations for Error Measurement (LIN Communication / Slew Rate) 10

 3.1.3. Proposed Countermeasures for the LIN Communication Measurement Environment (Slew Rate)..... 11

 3.1.4. Experimental Evaluation of Error Measurement Using a Pulse Generator 12

 3.1.5. Evaluation Results of Error Measurement Using a Pulse Generator 13

 3.1.6. LIN Communication Waveforms and Analysis of Experimental Results 14

 3.1.7. Example Code for SW1 15

 3.2. Data Logging of Information Related to Internal Oscillator Frequency Error (SW2 Case Study) 16

 3.2.1. Example Program for SW2 17

 3.3. Correction of Reference Software Based on Internal Oscillator Frequency Error Information (SW3 Example) 18

 3.3.1. State Before Oscillator Frequency Error Compensation 20

 3.3.2. State After Oscillator Frequency Error Compensation 21

 3.3.3. Example Program for SW3 22

4. Evaluation Example of Motor Control with Error Compensation Software 23

 4.1. Validation of Motor Control Software for Internal Oscillator Frequency Error Compensation 23

 4.1.1. Evaluation of Motor Control with Error Compensation Using a Frequency Error Generation Circuit 23

 4.1.2. Experimental Results of Evaluation 3a 24

 4.1.3. Experimental Results of Evaluation 3-b 26

 4.1.4. Summary of Evaluation 3-b Results..... 27

5. Summary 29

Notes on Contents..... 30

IC Usage Considerations..... 30

Notes on Handling of Ics 30

Points to Remember on Handling of ICs..... 31

RESTRICTIONS ON PRODUCT USE..... 32

List of Figures

Figure 2.1 Frequency Error ($\pm 5\%$) Generation Circuit..... 4

Figure 2.2 Experimental Concept of Evaluation 1 4

Figure 2.3 Experimental results of evaluation 1-a (1)..... 5

Figure 2.4 Experimental results of evaluation 1-a (2)..... 5

Figure 2.5 Experimental results of evaluation 1-b (1)..... 6

Figure 2.6 Experimental results of evaluation 1-b (2)..... 7

Figure 3.1 Concept of motor control software with oscillator frequency error correction..... 8

Figure 3.2 Relationship Between Response Reception and Interrupt Assertion 9

Figure 3.3 Operation Description of the 23-bit Counter for Automatic Baud Rate Mode 9

Figure 3.4 Counter value 23bit 10

Figure 3.5 Conceptual Diagram of LIN Communication Measurement Environment.....11

Figure 3.6 LIN Communication Signal with Slew Rate..... 12

Figure 3.7 Countermeasures for the Slew Rate 12

Figure 3.8 Experimental Concept of Evaluation 2 12

Figure 3.9 LIN Communication Waveforms in the Evaluation 2 Experimental Environment..... 14

Figure 3.10 Timing Chart of CPU, PMD, and MADC Operations When Using VE (Excerpt from HLD User Manual)..... 19

Figure 3.11 Effect of Internal Oscillator Frequency Error on Individual Timing element..... 20

Figure 3.12 Each Timing Element After Internal Oscillator Frequency Error Compensation..... 21

Figure 4.1 Experimental Concept of Evaluation 3 23

Figure 4.2 Experimental Results (1) of Evaluation 3a 24

Figure 4.3 Results of Evaluation 3-a (2)..... 25

Figure 4.4 Experimental Results of Evaluation 3-b (1)..... 26

Figure 4.5 Experimental Results of Evaluation 3-b (2)..... 27

Figure 4.6 Relationship Between PWM Period and Hardware Execution Time 28

Figure 4.7 MADC Single-Command Trigger Mode Measurement Timing..... 28

List of Tables

Table 3.1 Evaluation 2 Results (Commander Baud Rate: 19.2 kbps)..... 13

Table 3.2 Evaluation 2 Results (Commander Baud Rate: 4.8 kbps)..... 13

Table 4.1 Experimental Results of Evaluation 3-b..... 27

1. Purpose and Intention of This Document

In this application note, we describe a method for developing a motor control system using the TB9M003FG without implementing an external crystal oscillator, by utilizing the internal oscillator instead. When realizing motor control based on the internal oscillator, the accuracy of the oscillation frequency becomes a key challenge. This application note explains methods for improving and compensating for oscillation frequency accuracy, using reference examples, from the following perspectives:

- A method for measuring errors caused by oscillation frequency accuracy ($\pm 5\%$) during the manufacturing process of the user's end product
- A method for compensating the motor control program by utilizing the measured results

We hope that this application note will contribute to the development of improved motor control systems.

2. Effect of Internal Oscillator Frequency Accuracy on Motor Control

In this section, we discuss the effect of internal oscillator frequency accuracy ($\pm 5\%$) on motor control, based on experimental results.

2.1. Evaluation of Motor Control Operation under Internal Oscillator Frequency Error

2.1.1. Circuit for Generating Internal Oscillator Frequency Error ($\pm 5\%$)

In this experiment, an external clock is selected as shown in Fig. 2.1, and a signal from an external pulse generator is applied in order to pseudo-generate internal oscillator frequency error.

A 16 MHz external clock is input, and the clock configuration register (CLKSCCR1) is set so that the frequencies of CPUCLK, SYSCLK, and VECLK become identical to those obtained when the internal oscillator is selected.

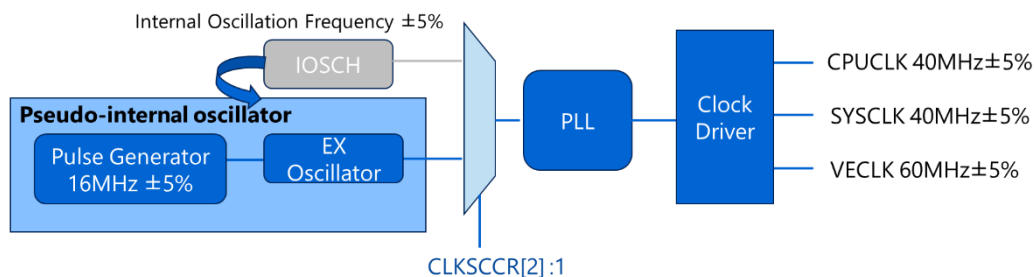


Figure 2.1 Frequency Error ($\pm 5\%$) Generation Circuit

2.1.2. Evaluation of Motor Control under $\pm 5\%$ Frequency Variation

Figure 2.2 illustrates the conceptual diagram of Evaluation 1.

In Evaluation 1, the reference software (motor_control_foc_ver1.0.0 / LLD ver1.0.0) is used as the motor control software to examine the effects that occur when the frequency of the internal oscillator varies by $\pm 5\%$.

Evaluation 1-a:

Using the reference software, the effect on actual motor speed relative to the commanded speed is examined.

Evaluation 1-b:

Using the reference software, the effect on the PWM period, VE (Vector Engine) processing time including VE interrupt handling, and dead time is examined.

Evaluation 1 (Reference SW)

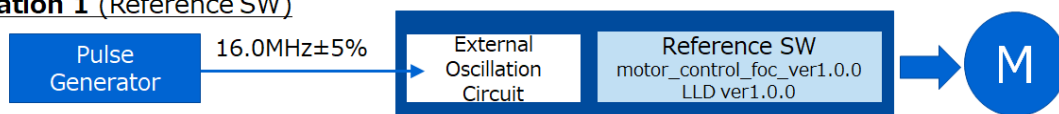
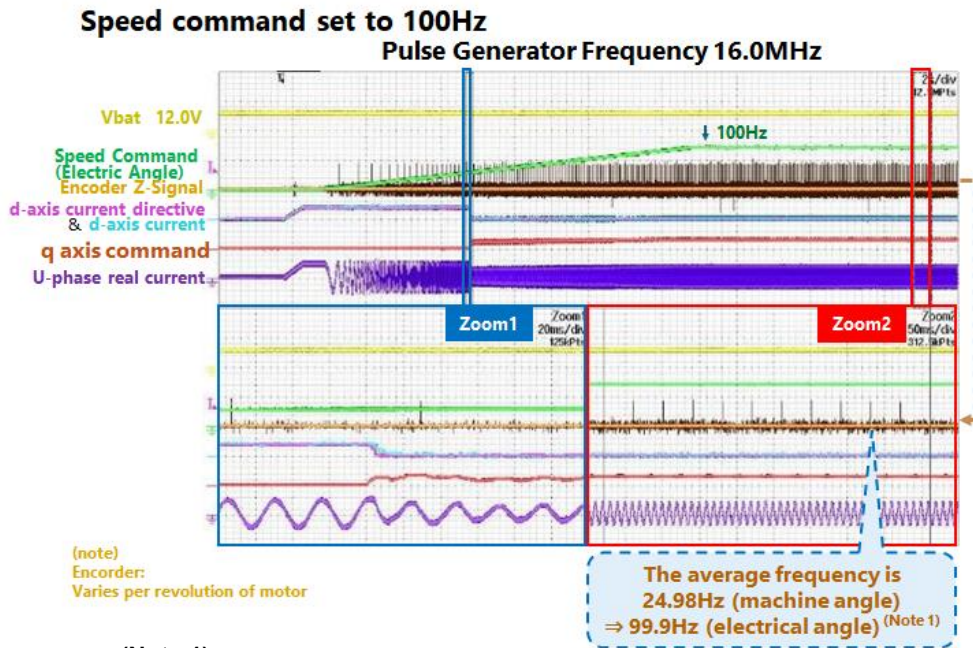


Figure 2.2 Experimental Concept of Evaluation 1

2.1.3. Results of Evaluation 1-a (Effect on Motor Speed Relative to the Speed Command)

1. Standard Condition: No Frequency Error



(Note 1):

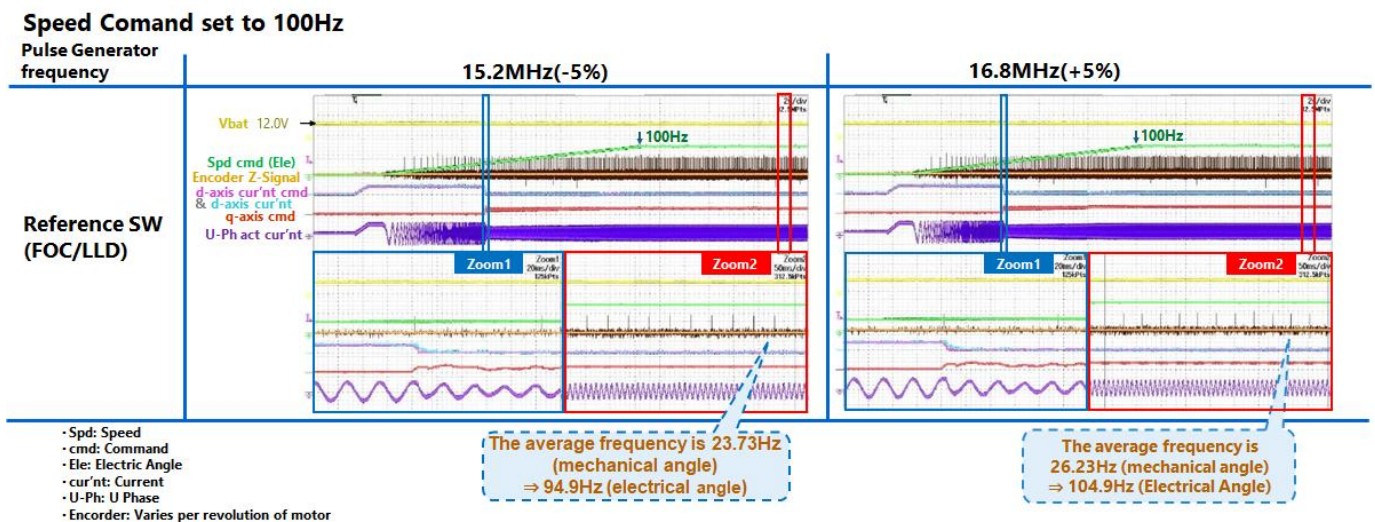
The motor used in this evaluation has four pole pairs; therefore, the electrical angle is four times the mechanical angle (electrical angle = 4 × mechanical angle).

Figure 2.3 Experimental results of evaluation 1-a (1)

Results:

Under the standard condition, with a speed command of 100 Hz, the motor rotational speed calculated from the encoder output (average electrical angular velocity) was 99.9 Hz, indicating that the motor speed closely matched the commanded speed.

2. Error Condition: ±5% Frequency Error



Results:

Under the condition simulating an oscillator frequency error of ±5%, the motor rotational speed calculated from the encoder output (average electrical angular velocity) was 94.9 Hz (-5%) and 104.9 Hz (+5%) for a speed command of 100 Hz.

These results indicate that the motor speed deviated by ±5% in accordance with the oscillator frequency error.

2.1.4. Evaluation 1-b Results (Effects on PWM Period, VE Processing/Interrupt Time, and Dead Time)

1. Standard Condition: No Frequency Error

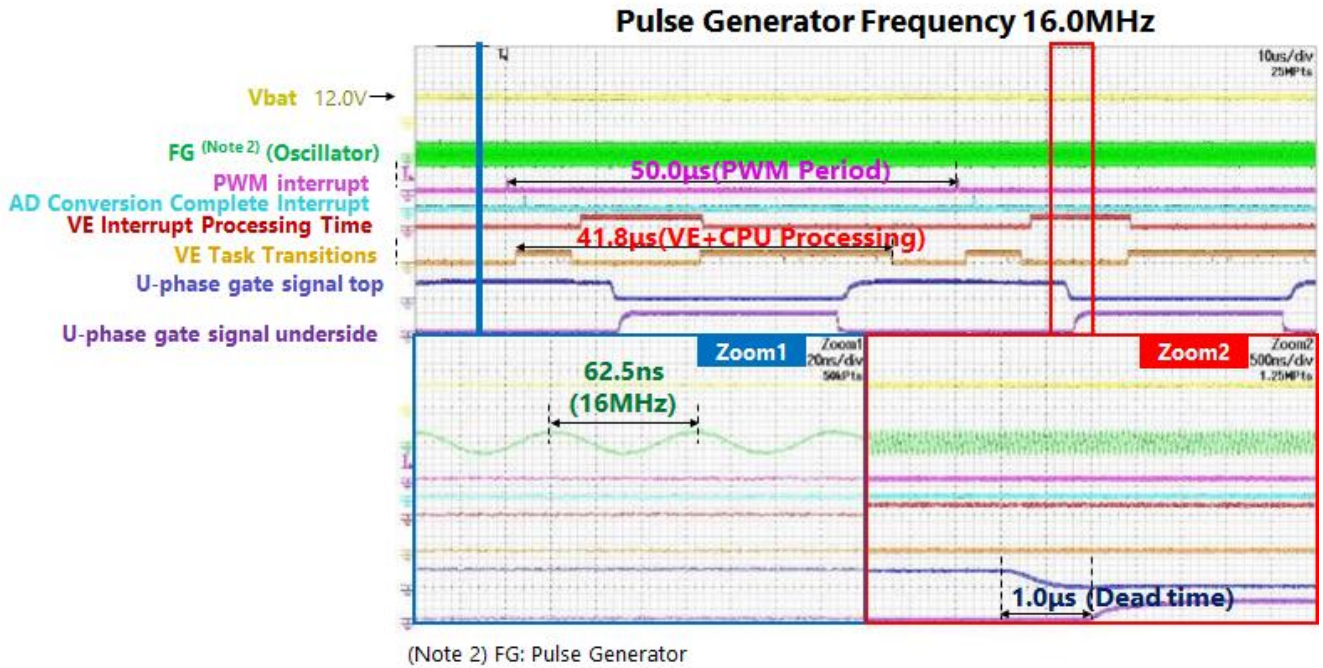


Figure 2.5 Experimental results of evaluation 1-b (1)

Results:

Under the standard condition, the PWM period, control processing time, and dead time were obtained as follows:

- | | |
|--|-----------|
| 1. PWM period | : 50.0 µs |
| 2. VE processing time + VE interrupt handling time | : 41.8 µs |
| 3. Dead time | : 1.0 µs |

2. Error Condition: $\pm 5\%$ Frequency Error

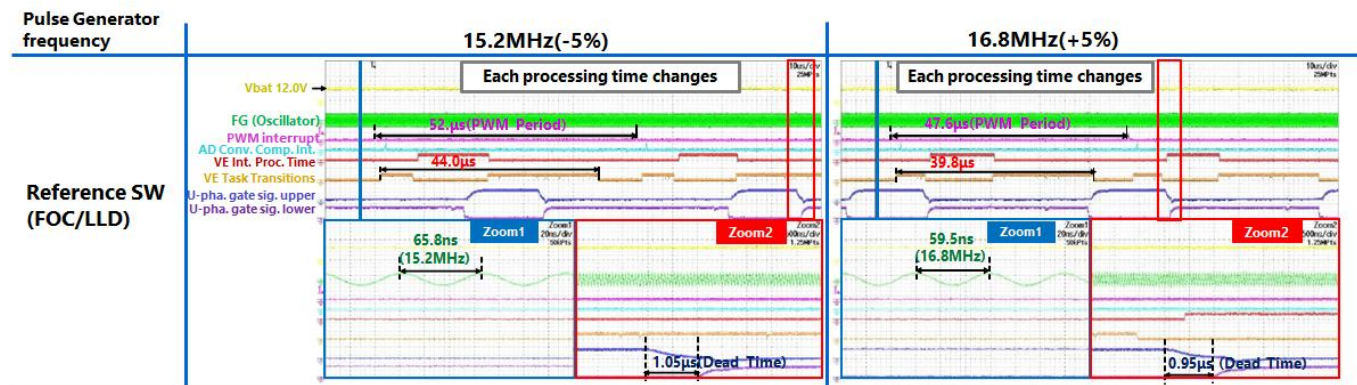


Figure 2.6 Experimental results of evaluation 1-b (2)

Results:

Under the condition where the oscillator frequency error was set to $\pm 5\%$, the PWM period, control processing time, and dead time showed deviations of approximately $\pm 5\%$ in accordance with the frequency error, as shown in Table 2.1.

Oscillator Frequency	15.2MHz (-5%)	16MHz	16.8MHz (+5%)
PWM periods	52.6 μ s (+5.2%)	50 μ s	47.6 μ s (-4.8%)
VE processing time + VE interrupt handling time	44 μ s (+5.3%)	41.8 μ s	39.8 μ s (-4.8%)
Dead Tome	1.05 μ s (+5%)	1.0 μ s	0.95 μ s (-5%)

3. Case Study of Motor Control Compensation for Internal Oscillator Frequency Error

In this chapter, we present an example of a method for controlling the motor rotational speed in accordance with the commanded value, even in the presence of errors caused by internal oscillator frequency accuracy ($\pm 5\%$).

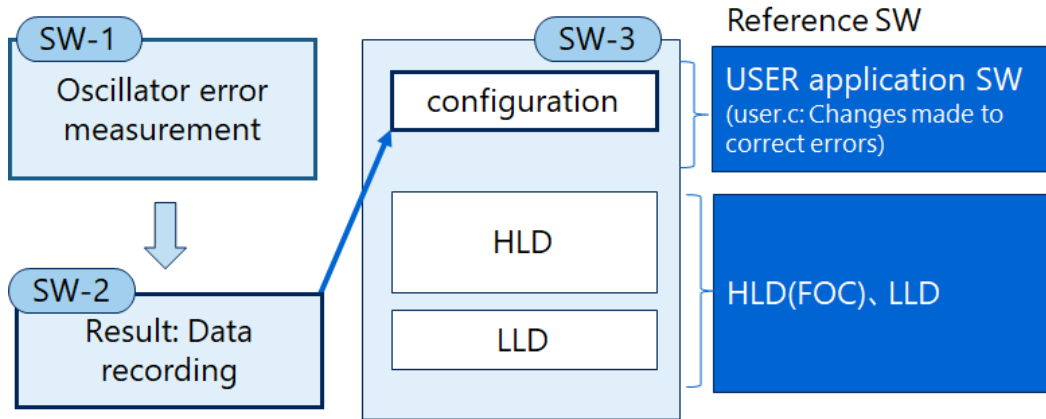


Figure 3.1 Concept of motor control software with oscillator frequency error correction

- **SW1:** Software for identifying the internal oscillator frequency error
- **SW2:** Software for storing the error information identified by SW1 into memory
- **SW3:** Software that performs error compensation based on the error information stored by SW2 and executes motor control (utilizing the reference software)

3.1. Method for Identifying Internal Oscillator Frequency Error (SW1 Case Study)

In this section, we describe a method for identifying internal oscillator frequency error.

To identify the internal oscillator frequency error, it is necessary to input an externally supplied signal with a known frequency, measure it as a time-based parameter using the device’s internal functions, and calculate the deviation by comparing the measured value with the expected value (known frequency).

In applications targeted by this product, communication functions used for information exchange between modules—such as LIN communication, UART communication, and PWM communication—can be used as such “signals with a known frequency.”

Therefore, in this section, we present an example that uses LIN communication among these communication functions.

3.1.1. Method for Measuring Internal Oscillator Frequency (Using LIN Communication)

Figure 3.2 shows a figure excerpted from Chapter 17 of the TB9M003FG User Manual (TB9M003FG_UM_J_Rev1.00) with auxiliary lines added, providing an overview of LIN communication. To measure the internal oscillator frequency using LIN communication, the “LIN automatic baud rate mode” is used by leveraging the TB9M003FG hardware functions that support this mode.

In principle, the measurement can be performed using a single communication frame from the commander that includes the Sync Field. However, this requires that the commander’s baud rate be stable. The TB9M003FG provides the registers shown in Fig. 3.4 for use with the LIN automatic baud rate mode. The internal oscillator frequency error can be calculated using the values stored in these registers. In automatic baud rate mode, the pulse width of the received Sync Field bits is measured with a 23-bit counter in order to detect the commander’s baud rate at the time of Sync Field reception. For this measurement, the internal LINCLK (Note 3) is used.

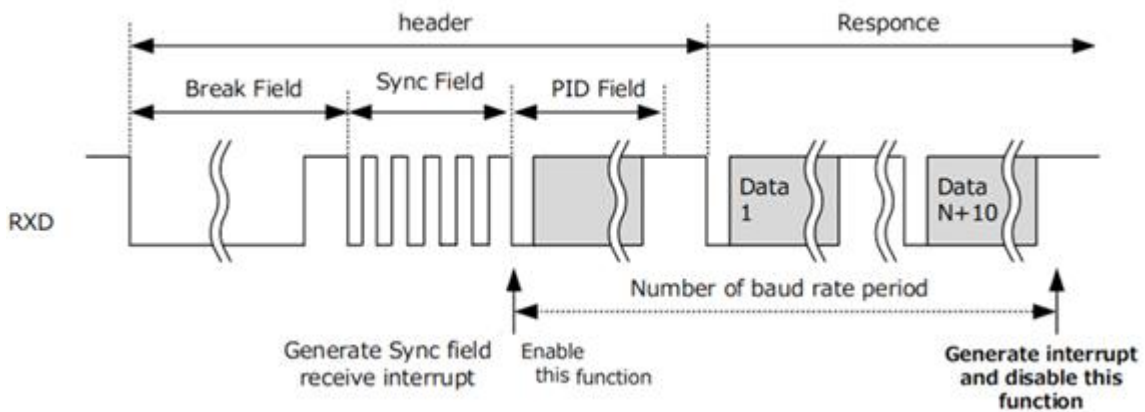


Figure 3.2 Relationship Between Response Reception and Interrupt Assertion (From Chapter 17 of the TB9M003FG User Manual)

Figure 3.3 illustrates an overview of this counter operation.

Since the measured count value is stored in a register, the accumulated count value corresponding to 8 bits of the received data can be obtained from this value.

(Note 3):LINCLK: System clock for the LIN control circuit

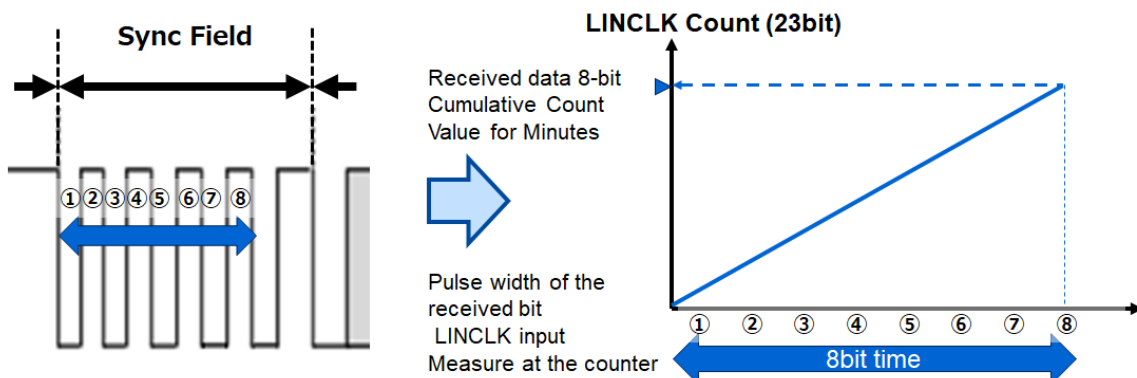


Figure 3.3 Operation Description of the 23-bit Counter for Automatic Baud Rate Mode

The 23-bit data shown in Fig. 3.4 represents the number of LINCLK counts corresponding to the received data (8 bits) within the Sync Field.

Therefore, if a predefined expected value (23-bit data) and a measured value (23-bit data) are available, the frequency error can be calculated by comparing these values.

However, due to the hardware configuration, the least significant bit of the count value is rounded off, and thus the effective number of valid bits is 22. This 22-bit value can be read from the register.

In addition, this register value theoretically includes an error of $\pm 2 \times \text{LINCLK}$ periods. For example, when LINCLK is 40 MHz, the following measurement error is included in the count value.

$$\pm 2 \times 25\text{ns} = \pm 50\text{ns} \quad \text{: Measurement Error}$$

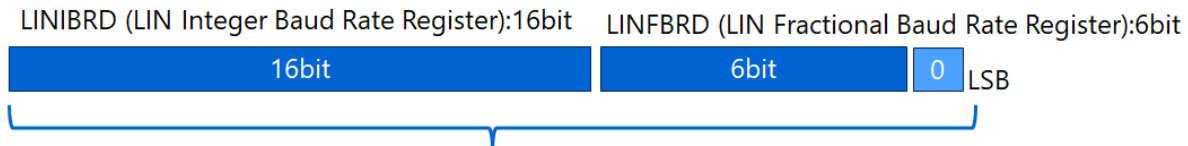


Figure 3.4 Counter value 23bit

3.1.2. Notes and Considerations for Error Measurement (LIN Communication / Slew Rate)

In this section, notes and considerations are summarized for measuring the internal oscillator frequency by applying LIN communication.

1. Effect of Slew Rate

In LIN communication, the slew rate varies depending on environmental conditions of the physical communication layer. As a result, when the bit length of received data is counted using a clock, the measurement may be affected by this slew rate variation.

Section 3.1.3 describes proposed countermeasures addressing this issue.

2. Baud Rate Setting at Initial Communication

In LIN communication, a frame timeout error is defined. To avoid this error, when using the LIN automatic baud rate mode, the baud rate for the initial communication (when the baud rate has not yet been determined) must be set sufficiently low.

A frame timeout error occurs if the frame header is not received within 1.4 times the frame time (34T (Note 4)), based on the configured baud rate.

For example, if the initial baud rate in automatic baud rate mode is set to 10 kbps, an error will occur when attempting to detect communication at 2 kbps.

3. Method for Obtaining Commander-Side Clock Accuracy and Expected Values

In the measurement used to identify frequency error described in this section, the accuracy of the signal period and the stability of the commander signal (external signal) generated by tools such as LIN bus simulation tools are critical.

As a method for obtaining the expected values from this signal, it is recommended to acquire and utilize data under a LIN communication measurement environment using a module in which the TB9M003FG equipped with a crystal (X'tal) oscillator is implemented on the signal reception side.

The reason for this recommendation is that the data obtained in this environment can be expected to provide highly accurate count values corresponding to the external signal under the given measurement conditions.

3.1.3. Proposed Countermeasures for the LIN Communication Measurement Environment (Slew Rate)

In this section, we explain slew rate variations that occur depending on environmental conditions during LIN communication, as well as methods for mitigating the effect of slew rate on frequency measurement.

Figure 3.5 conceptually illustrates the environment in which LIN communication is performed, as described in this section.

The LIN bus simulation tool is equipped with a stable, high-accuracy system clock (1), and generates a high-precision LIN communication reference signal (2) based on this clock. The TB9M003FG board receives this reference signal (2) as a LIN communication signal (3) via the external environment. The TB9M003FG mounted on the board then buffers this LIN communication signal (3) and subsequently receives signal (4) through the LIN communication control circuit.

For frequency measurement, it is necessary for the internal circuitry to receive signal (4) while maintaining the period of the reference signal (2).

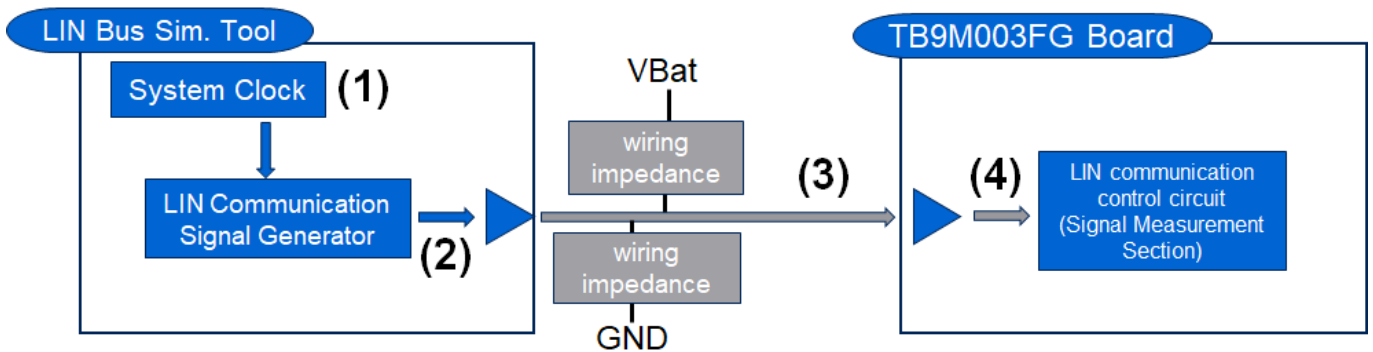


Figure 3.6 conceptually illustrates the LIN communication reference signal (2), the LIN communication signal (3) in which the reference signal (2) is affected by the external environment, and signal (4), which is obtained after signal (3) is buffered by the internal circuitry of the device.

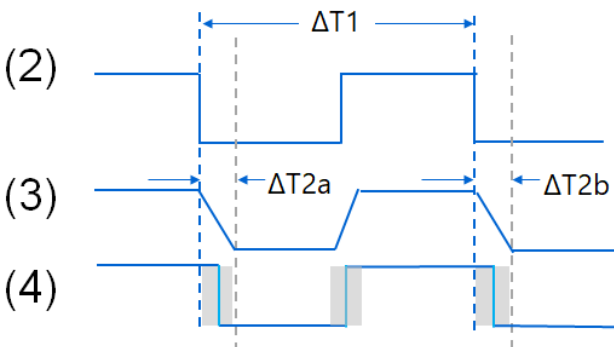
The reference signal (2) is a stable signal without slew rate effects, with $\Delta T1$ representing one bit period. The LIN communication signal (3) is the reference signal (2) affected by the external environment (wiring impedance), resulting in the occurrence of a slew rate $\Delta T2$.

Since the LIN communication control circuit inside the device uses the falling edge of the signal, the explanation here focuses on the slew rate on the falling-edge side.

Signal (4), which is received by the internal circuitry of the device, is the buffered version of the signal and therefore no longer exhibits slew rate effects. However, the timing of the falling edge is affected by the slew rate of the LIN communication signal (3) by up to $\Delta T2$.

The slew-rate duration $\Delta T2a$ of the first falling edge affects the period of signal (4) in the direction of shortening it.

In contrast, the slew-rate duration $\Delta T2b$ of the subsequent falling edge affects the period of signal (4) in the direction of lengthening it.



Since the 1?bit period of signal (4) is influenced by the combination of these two slew-rate effects, their effects tend to cancel each other out.

For example, when both effects are equivalent, they cancel out and no effect on the period occurs. However, in order to consider the worst-case scenario, the analysis is carried out assuming no cancellation (i.e., a condition in which the slew rate on one side is zero).

The conceptual relationships of $\Delta T1$ and $\Delta T2$ are expressed as follows:

Figure 3.6 LIN Communication Signal with Slew Rate

- $\Delta T1 = f$ (baud rate, system clock accuracy)
- $\Delta T2 = g$ (load, LIN physical layer drive capability)

The function for $\Delta T1$ does not include the factors that cause slew rate, which are represented in the function for $\Delta T2$. Likewise, the function for $\Delta T2$ does not include the factors that determine the 1?bit period (baud rate), which are represented in the function for $\Delta T1$.

Therefore, even if the baud rate is set so that $\Delta T1$ becomes larger, the value of $\Delta T2$ does not change. Figure 3.7 conceptually illustrates each signal when the baud rate is reduced to one-third.

As shown, by increasing $\Delta T1$ through a lower baud rate during measurement, it is possible to reduce the influence of slew rate caused by the measurement environment.

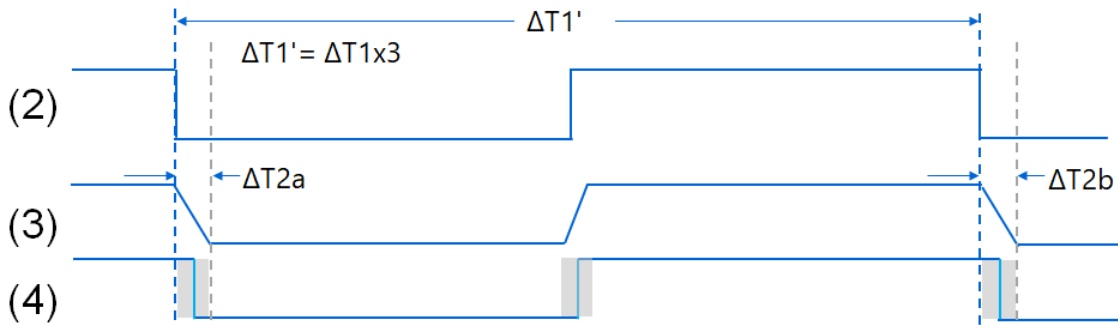


Figure 3.7 Countermeasures for the Slew Rate

3.1.4. Experimental Evaluation of Error Measurement Using a Pulse Generator

In Evaluation 2, an experiment is conducted to verify whether error detection using LIN communication can be correctly performed when the oscillator frequency is pseudo-varied by $\pm 5\%$ using a pulse generator. Figure 3.8 illustrates the concept of Evaluation 2.

Evaluation 2 (LIN sample program)

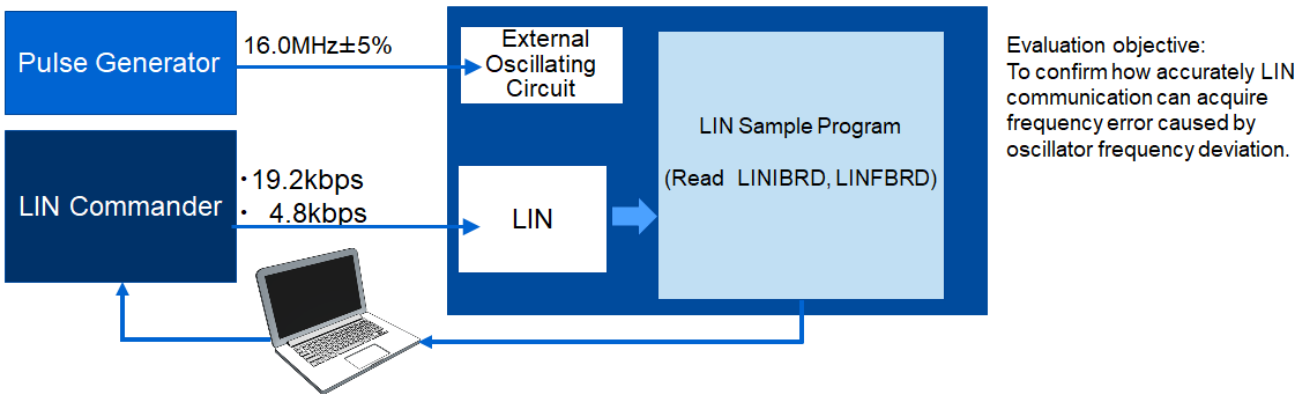


Figure 3.8 Experimental Concept of Evaluation 2

3.1.5. Evaluation Results of Error Measurement Using a Pulse Generator

Table 3.1 Evaluation 2 Results (Commander Baud Rate: 19.2 kbps)

1) Pulse Generator vs. Reference Period	Count Value (Decimal) (Note5)	LINIBRD (Hexa)	LINFBRD (Hexa)	2) Measured Value / Reference Calculation of Measured Value	Measurement Error 1) - 2)
105.00% : 16.80MHz	17516	88	36	104.99%	0.01%
103.13% : 16.50MHz	17202	86	19	103.10%	0.02%
Reference:16.00MHz	16684	82	16	100.00%	0.00%
96.88% : 15.50MHz	16122	7D	3D	96.63%	0.24%
95.00% : 15.20MHz	15848	7B	34	94.99%	0.01%

Table 3.2 Evaluation 2 Results (Commander Baud Rate: 4.8 kbps)

1) Pulse Generator vs. Reference Period	Count Value (Decimal) (Note5)	LINIBRD (Hexa)	LINFBRD (Hexa)	2) Measured Value / Reference Calculation of Measured Value	Measurement Error 1) - 2)
105.00% : 16.80MHz	70068	223	1A	104.99%	0.01%
103.13% : 16.50MHz	68810	219	25	103.11%	0.02%
Reference:16.00MHz	66736	209	18	100.00%	0.00%
96.88% : 15.50MHz	64480	1F7	30	96.62%	0.26%
95.00% : 15.20MHz	63392	1EF	10	94.99%	0.01%

(Note5) : 23-bit data

3.1.6. LIN Communication Waveforms and Analysis of Experimental Results

From the observed communication waveform, it was found that the fall time (T_f) is approximately $2\ \mu\text{s}$ (Fig. 3.9). The ratio relative to the time required for an 8-bit data length at 19.2 kbps is: $2\ \mu\text{s} / 417\ \mu\text{s} = 0.0048$ ($\approx 0.5\%$). Similarly, at 4.8 kbps, the ratio is: $2\ \mu\text{s} / 1667\ \mu\text{s} = 0.0012$ ($\approx 0.1\%$).

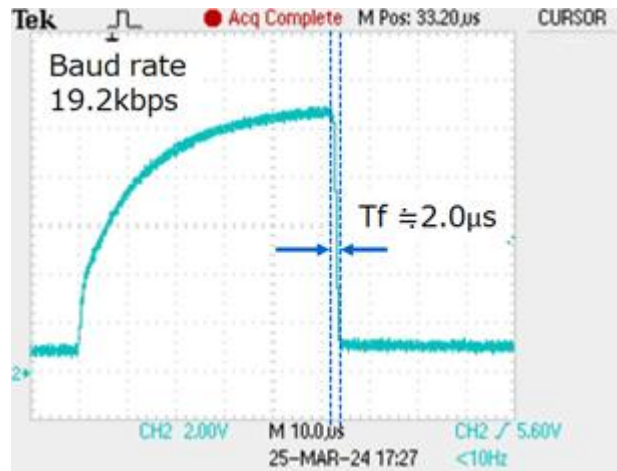


Figure 3.9 LIN Communication Waveforms in the Evaluation 2 Experimental Environment

In the discussion in Section 3.1.3, it was shown that these values can contribute to measurement error. However, in the evaluation results in Section 3.1.5, this error does not appear to be observed. This is because, in the error calculation, there was no significant difference between the two $\Delta T2$ conditions. If $\Delta T2a$ and $\Delta T2b$ described in Section 3.1.3 are equivalent, their effects may cancel each other out, resulting in the error not being detected.

In the evaluation results presented in Section 3.1.5, the measured count value obtained when a 16 MHz input was applied was used as the reference (expected) data.

In this case, it is important that the commander's clock period is stable in the user's environment, while its absolute accuracy is not critical (see Sections 3.1.2–3.1.4).

It is also possible to use a theoretical value calculated from the baud rate as the expected value. However, in this case, the frequency error of the commander's clock would be reflected as measurement error. In other words, there is a possibility that compensation may be applied even to components that do not require adjustment.

Based on the evaluation results in Section 3.1.5, it was demonstrated that, under this experimental environment, the internal oscillator frequency error can be detected with an accuracy of better than 0.3% using LIN communication.

In addition, except for the case where the pulse generator output was 15.5 MHz, detection results with high accuracy of better than 0.03% were obtained.

Section 3.1.7 introduces a method for measuring internal oscillator frequency error using LIN communication, along with example programs.

3.1.7. Example Code for SW1

(Interrupt handler configuration for transmission/reception and interrupt callback settings are omitted.)

```
uint32_t measure_lincount(void)
{
    lld_slin_config_t config = {
        40000000U, /* clock_info */
        1000U, /* baudrate */
        LLD_SLIN_BAUDRATE_MODE_AUTO, /* mode */
        LLD_SLIN_BREAK_WIDTH_10, /* break_width */
        LLD_SLIN_BIT_ERROR_AT_3_4TH, /* bit_error */
        LLD_SLIN_SLEW_RATE_1_5, /* slew rate mode */
        3U, /* inter_byte_space */
        3U /* response_space */
    };
    lld_slin_config(&config);
    lld_slin_start(lld_true);

    /* Execute communication here to measure baudrate */
    wait_for_lin_communication();
    lincount = (SMCD_LINC->LINIBRD << 7) | (SMCD_LINC->LINFBRD << 1);

    lld_slin_stop();
    return lincount;
}
```

LINCLK = 40 MHz; Automatic Baud Rate Mode; Initial baud rate set to the lowest value of 1 kbps
(to ensure a sufficiently long frame timeout period during header communication before automatic baud rate calculation)

The LIN count value is calculated from the LINIBRD and LINFBRD register values after LIN communication.

3.2. Data Logging of Information Related to Internal Oscillator Frequency Error (SW2 Case Study)

In Section 3.2, we describe a method for storing the measured internal oscillator frequency data and reference data in memory.

In order to utilize the internal oscillator frequency measurement data obtained after LIN communication for motor control, it is necessary to store the data in memory. The memory to be used for storage must be appropriately selected depending on the scenario in which error compensation is performed.

The candidate memory options are shown below.

- SRAM
- Code Flash
- External EEPROM

In this application note, the following scenario is assumed:

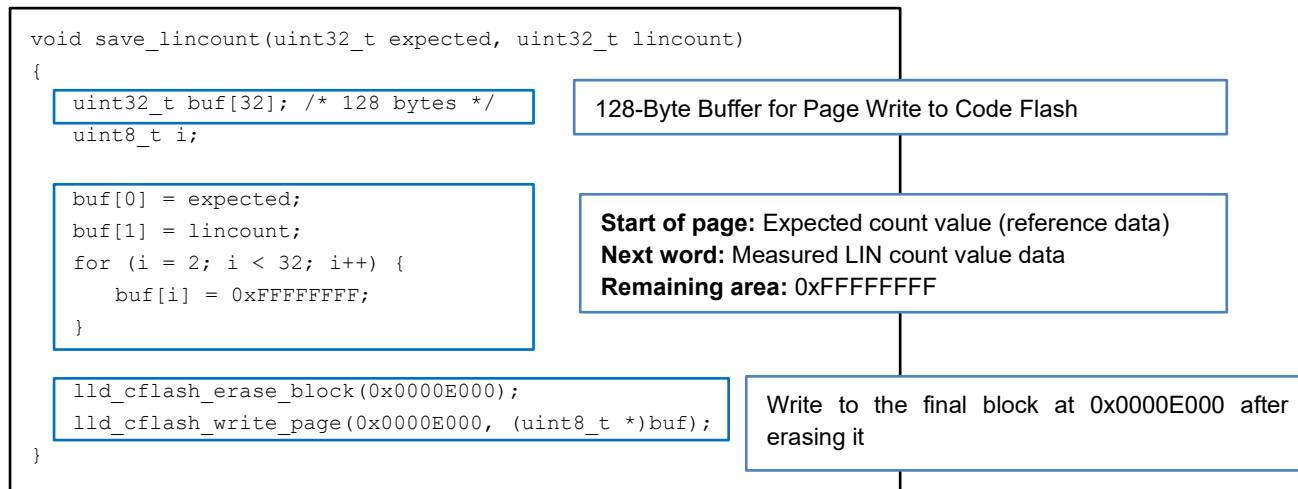
1. Measurement of the internal oscillator frequency using LIN communication and acquisition of reference data are performed during the manufacturing process of the user's product.
2. The measured internal oscillator frequency data and the reference data are stored in non-volatile memory, and no further updates are made thereafter.
 - Measured data: Count values obtained from measurements performed on the control board to be shipped (using the internal oscillator)
 - Reference data: Count values obtained from measurements performed on a control board equipped with a crystal (X'tal) oscillator
3. The two stored data sets are referenced during operation of the motor control system in the field and are used for error compensation.

Based on the above scenario, Code Flash was selected as the memory for data storage.

Section 3.2.1 introduces a method for writing data to Code Flash, along with an example program.

3.2.1. Example Program for SW2

Of the 64 KB Code Flash (0x00000000 – 0x0000FFFF), it is assumed that the final block (8 KB starting from 0x0000E000), which corresponds to an erase unit size of 8 KB, is an unused area reserved by the system. Based on this assumption, in the following example, data is stored at the beginning of the final block.



Example Program for Reading Recorded Data

```

uint32_t expected = *(uint32_t *)0x0000E000;
uint32_t lincount = *(uint32_t *)0x0000E004;
    
```

The recorded data can be read as shown on the left.

3.3. Correction of Reference Software Based on Internal Oscillator Frequency Error Information (SW3 Example)

In this section, we describe a method for maintaining the motor control algorithm appropriately even when errors are inherent in the oscillator frequency.

In the reference software, the following three-time axes are critical for correctly executing the motor control algorithm.

1. MAINLOOP_USER_PERIOD
2. SPD_CTRL_PERIOD
3. PWM_PERIOD

Before applying oscillator frequency error compensation, all of the above three timing elements operate with deviations corresponding to the frequency error.

From an internal perspective, the system appears to be operating correctly; however, since it is actually operating under timing conditions different from those used during motor control parameter tuning, it is important to note that this may affect tuning accuracy.

3. The effect of frequency error on PWM_PERIOD directly appears in the calculation of motor rotational speed. This is because this period is used as a parameter in the function that determines the motor rotational speed ω .

$$\omega = f(\text{PWM period, ...})$$

Therefore, frequency error compensation is required for the PWM period.

In addition, 1) MAINLOOP_USER_PERIOD and 2) SPD_CTRL_PERIOD define the processing cycle of the overall control system, and frequency error compensation is required to maintain appropriate contributions, such as gain control. Since these timing parameters are based on DTIMER, it is necessary to perform error compensation for this function as well.

Figure 3.10 is an excerpt from the “Timing” chapter of the Reference Software Manual: HLD User Manual. It illustrates the relationship between each function and the following timing elements. For details, refer to the same manual.

- 1) MAINLOOP_USER_PERIOD
- 2) SPD_CTRL_PERIOD
- 3) PWM_PERIOD

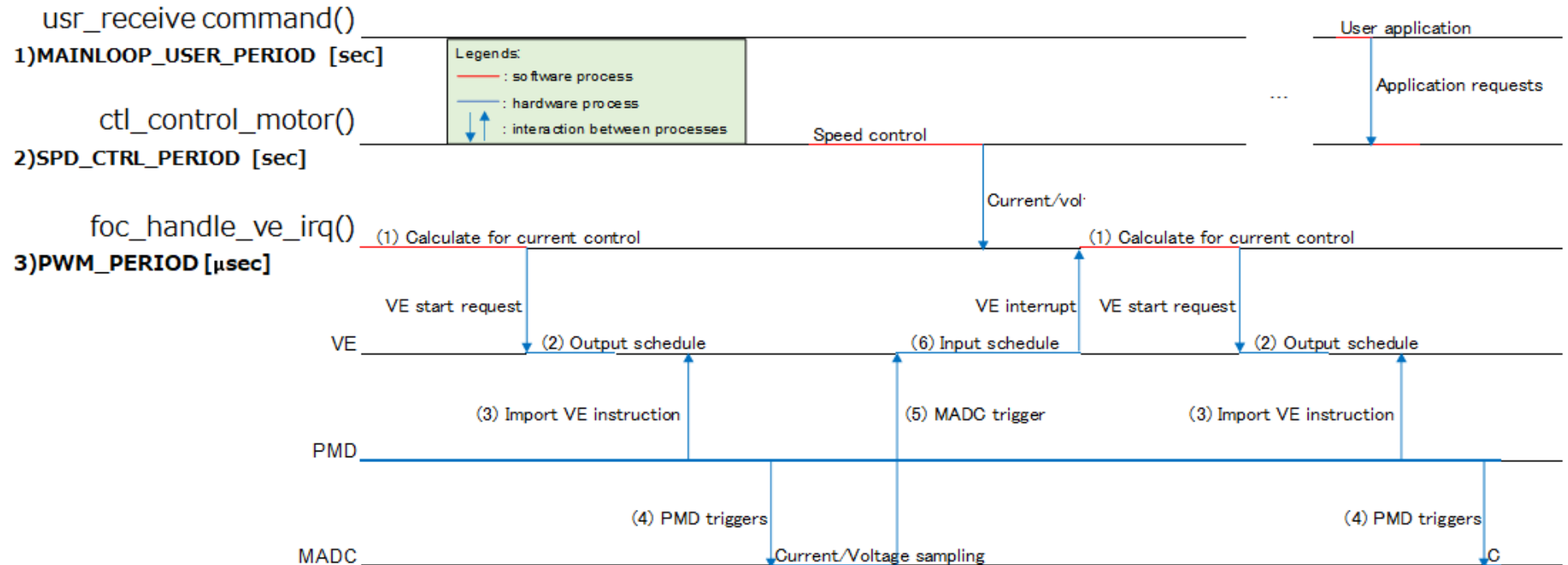


Figure 3.10 Timing Chart of CPU, PMD, and MADC Operations When Using VE (Excerpt from HLD User Manual)

3.3.1. State Before Oscillator Frequency Error Compensation

Each timing element varies uniformly in proportion to the oscillator frequency error. Figure 3.11 illustrates this condition conceptually.

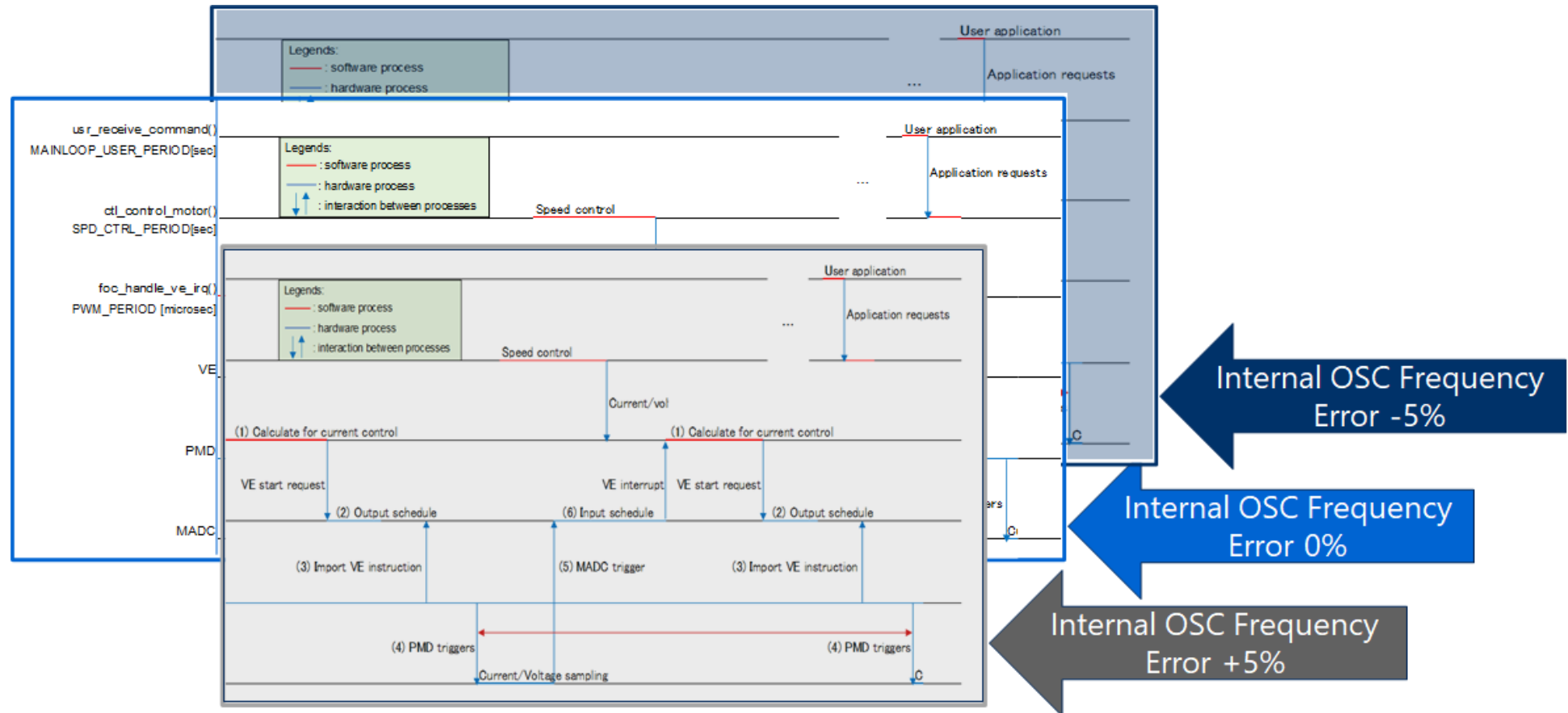


Figure 3.11 Effect of Internal Oscillator Frequency Error on Individual Timing element

3.3.2. State After Oscillator Frequency Error Compensation

Figure 3.12 shows how each timing element behaves when oscillator frequency error compensation is applied.

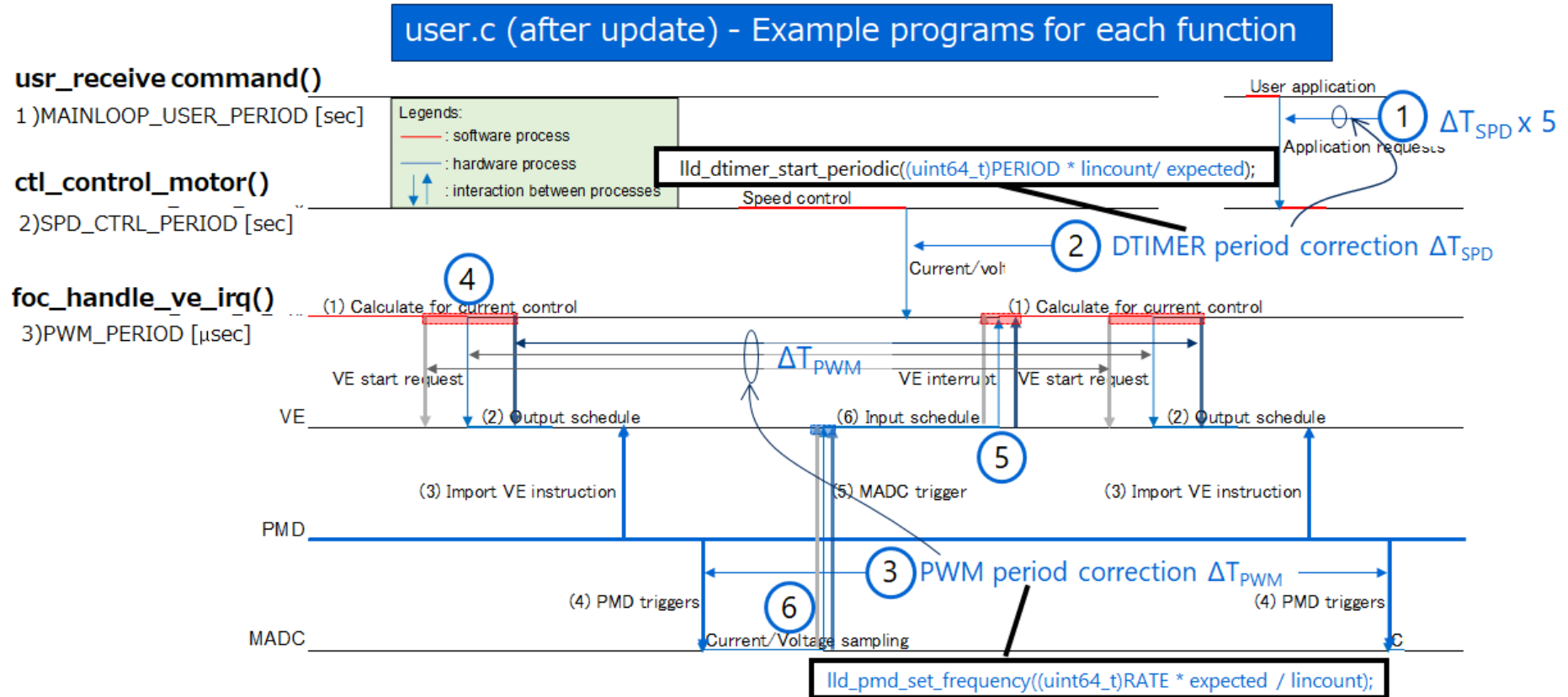


Figure 3.12 Each Timing Element After Internal Oscillator Frequency Error Compensation

The `foc_handle_ve_irq()` function is invoked as a VE interrupt handler at every PWM period (`PWM_PERIOD`), which is set within the PMD. Key operations—such as triggering VE input tasks and configuring PWM output based on VE output tasks—are executed with the PWM signal as the reference timing.

Therefore, frequency error compensation must be applied to the PWM signal. By doing so, ΔT_{PWM} shown as (3) in Fig. 3.12 is restored to its intended period, and the timing of control signals originating from the PMD (such as trigger signals to the VE and MADC) is also properly aligned.

Although the timing of signals dependent on other CPU processing (4), VE processing time (5), and ADC conversion time (6) is not directly adjusted, their periods are corrected to the intended ΔT_{PWM} . As a result, the effect of frequency error on the overall control system is significantly reduced.

In addition, since the PWM period is used in the calculation of the motor rotational speed ω , the motor speed is also properly regulated.

The `ctl_control_motor()` function corresponds to the fundamental control executed at every `SPD_CTRL_PERIOD` (1 ms), and performs the following operations:

- Speed control
- Target speed setting
- Field weakening control
- State control (positioning ~ steadystate operation, and fault handling)

The `usr_receive_command()` function is executed at every `MAINLOOP_USR_PERIOD` (`SPD_CTRL_PERIOD` × 5) and performs user-implemented processing. In the reference software, it executes command-check processing for starting and stopping motor control.

Since `SPD_CTRL_PERIOD` is defined by the DTIMER setting, compensating this value corrects ΔT_{SPD} shown as (2) in Fig. 3.12 to its intended period (1 ms).

As a result, $\Delta T_{SPD} \times n$ ($n = 5$), shown as (1), is also corrected to its intended period.

In Section 3.3.3, we introduce an example program for compensating the reference software based on internal oscillator frequency error information.

3.3.3. Example Program for SW3

Reference SW (motor_control_foc_ver1.0.0)
user.c

```
void usr_init(void)
{
  /* 1. Initialize motor control peripherals */
  init_ip_XXX();
  /* 2. Configure interrupts */
  arm_NVIC_XXX();
  /* 3. Initialize HLD */
  ctl_init_motor();
  /* 4. Start a timer for the control */
  lld_dtimer_start_periodic(ideal_period);
}
```

Current sample of user application



Reference SW (motor_control_foc_ver1.0.0)
user.c (example after update)

```
void usr_init(void)
{
  /* 1. Initialize motor control peripherals */
  init_ip_XXX();
  /* 2. Configure interrupts */
  arm_NVIC_XXX();
  /* 3. Initialize HLD */
  ctl_init_motor();
  /* Correct HLD timings by stored data */
  lld_pmd_set_frequency(
    (uint64_t)RATE * expected / lincount);
  /* 4. Start a timer for the control */
  lld_dtimer_start_periodic(
    (uint64_t)PERIOD * lincount / expected);
}
```

Using the ratio of the actual period deviation, defined as "actual LIN counter value" / "ideal LIN counter value" (hereinafter referred to as the correction ratio), the interval is adjusted.

Since it is a frequency value, correction is performed by dividing by the correction ratio.

Since it is a period value, correction is performed by multiplying by the correction ratio.

4. Evaluation Example of Motor Control with Error Compensation Software

In this chapter, we evaluate, based on experimental results, how motor control behavior changes when the internal oscillator frequency error ($\pm 5\%$) is detected and compensated by software.

4.1. Validation of Motor Control Software for Internal Oscillator Frequency Error Compensation

In this section, based on experimental results, we describe to what extent the motor control software that compensates for internal oscillator frequency error improves the control performance confirmed in Chapter 2. In addition, the experimental method is also explained.

4.1.1. Evaluation of Motor Control with Error Compensation Using a Frequency Error Generation Circuit

In Evaluation 3, the following SW1 to SW3 are used to verify how motor control operation is affected when the internal oscillator frequency varies by $\pm 5\%$.

In this experiment, the focus is placed on verifying the operation of the motor control software with error compensation (SW3). Therefore, SW1 and SW2 are implemented using methods different from those previously described (SW1: LIN communication, SW2: Code Flash), as summarized below.

1. **SW1 (Note 6):** A 20 kHz square wave generated by a pulse generator (channel 2) is applied as a reference signal from an external source to the capture input pin, and its period is measured using the capture timer function.
2. **SW2 (Note 6):** The expected value of the period (20 kHz) and the measured value obtained by SW1 are stored in SRAM.
3. **SW3 (Note 6):** By modifying part of the user application software (user.c) in the reference software (motor_control_foc_ver1.0.0 / LLD ver1.0.0), motor control with internal oscillator frequency error compensation is performed using the data stored by SW2.

(Note 6): Refer to Figure 3.1.

Figure 4.1 illustrates the experimental concept of Evaluation 3.

- **Evaluation 3-a:**
Using the reference software with frequency error compensation, the effect of the speed command on the motor rotational speed is verified.
- **Evaluation 3-b:**
Using the reference software with frequency error compensation, the effect on the PWM period, VE processing and VE interrupt handling time, and dead time is verified.

Evaluation 3 (Reference SW with Frequency Error Compensation)

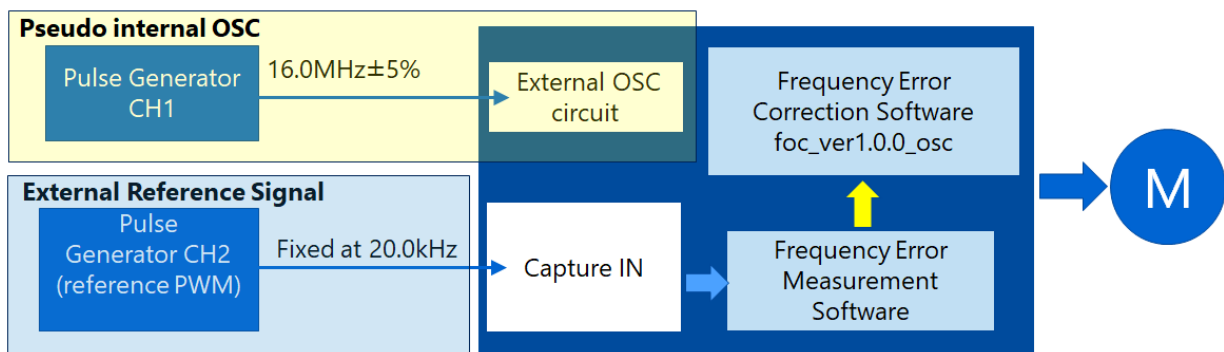


Figure 4.1 Experimental Concept of Evaluation 3

Objective:

To verify the improvement in operation under oscillator frequency variations using frequency-error-compensated reference software.

4.1.2. Experimental Results of Evaluation 3a

1. Nominal Condition: No Frequency Error

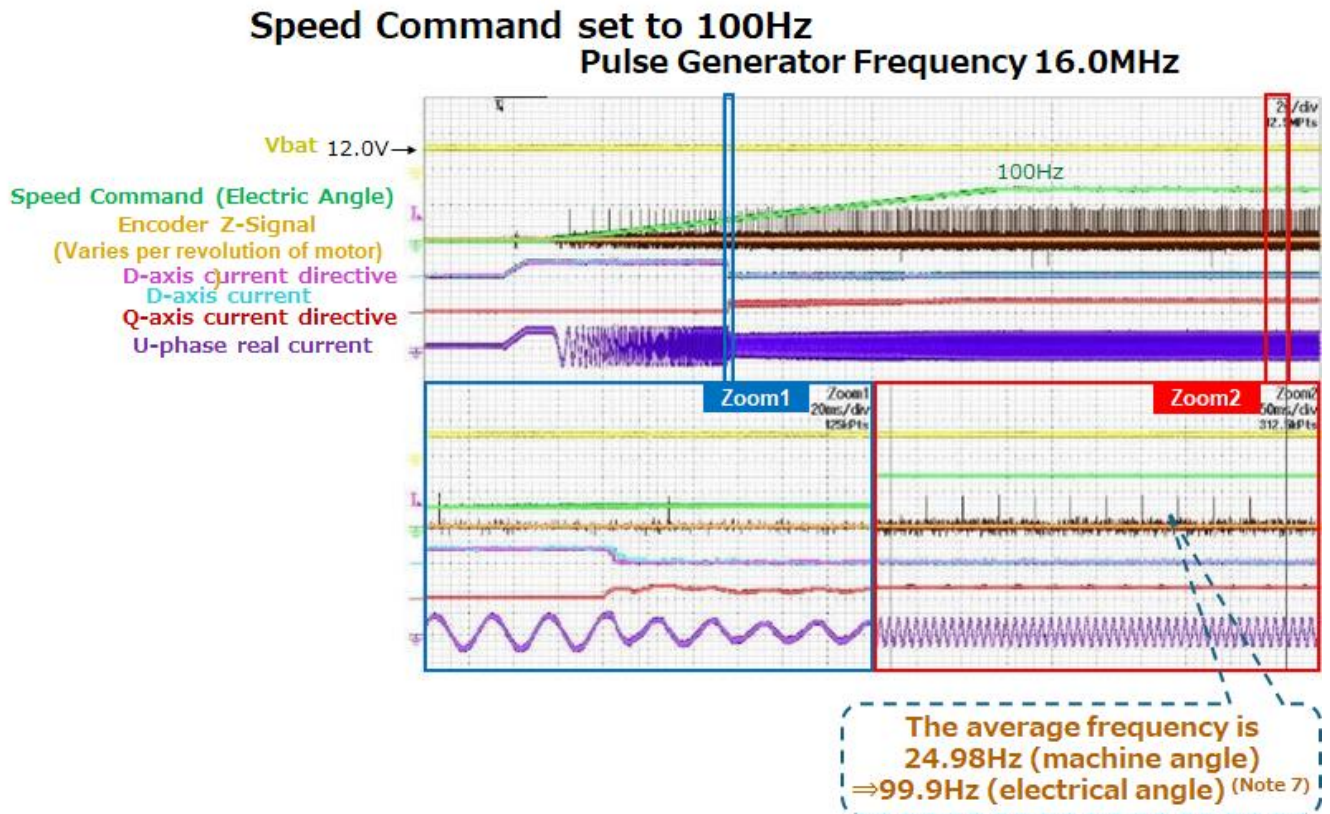


Figure 4.2 Experimental Results (1) of Evaluation 3a

(Note 7) As the motor used in this evaluation has four pole pairs, the relationship is:
electrical angle = 4 × mechanical angle.

Result:

Under standard conditions (i.e., no oscillator frequency error), when the speed command is set to 100 Hz, the motor rotational speed calculated from the encoder output (average electrical angular velocity) is 99.9 Hz, showing a value nearly identical to the command.

2. Error Condition: $\pm 5\%$ Frequency Deviation

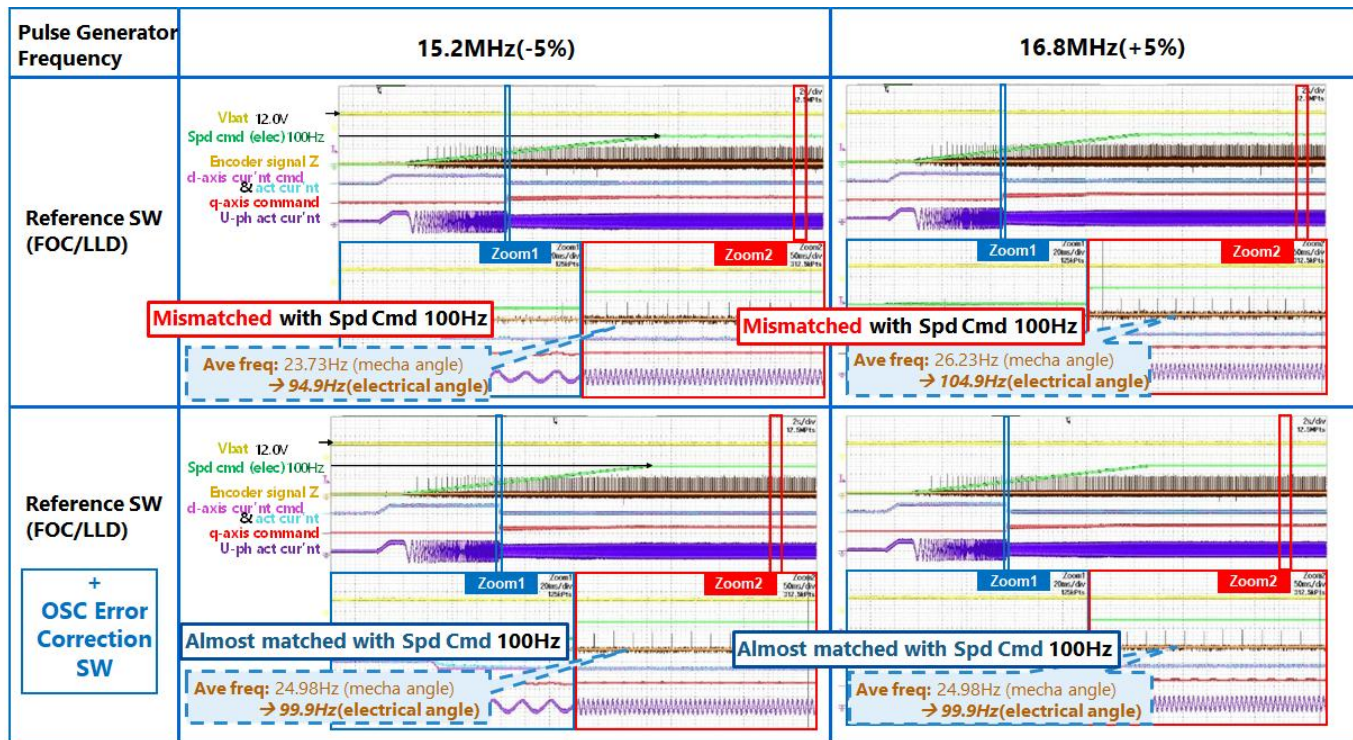


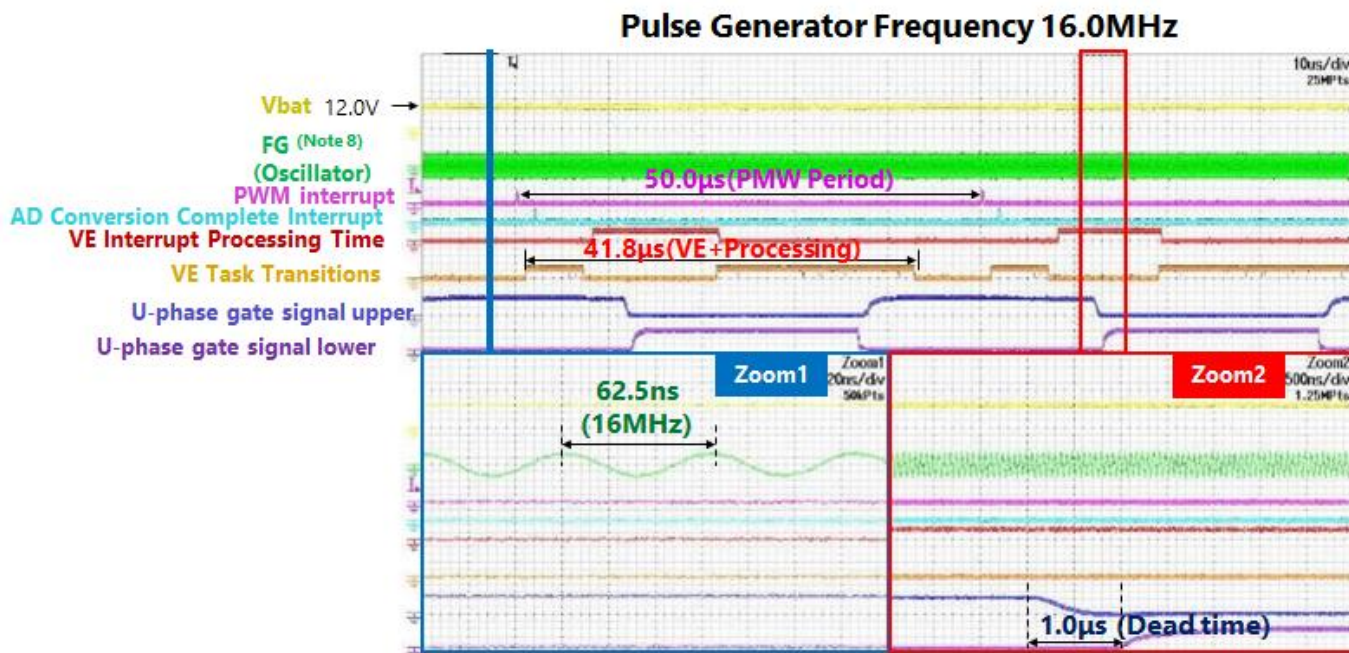
Figure 4.3 Results of Evaluation 3-a (2)

Results:

Under the condition simulating an oscillator frequency error of $\pm 5\%$, when the value of the speed command was 100 Hz, the motor rotational speed calculated from the encoder output (average electrical angular velocity) was 99.9 Hz, showing a value almost identical to that under the standard condition. Compared with the result of Evaluation 1-a, which showed a trend of $\pm 5\%$ deviation corresponding to the oscillator frequency error, a clear improvement can be observed.

4.1.3. Experimental Results of Evaluation 3-b

1. Nominal Condition: No Frequency Error



(Note8) FG: Pulse Generator

Figure 4.4 Experimental Results of Evaluation 3-b (1)

Results:

Under the standard condition (i.e., no oscillator frequency error), the PWM period, control processing time, and dead time were as follows:

1. PWM period: 50.0 µs
2. VE processing time + VE interrupt handling time: 41.8 µs
3. Dead time: 1.0 µs

2. Error Condition: $\pm 5\%$ Frequency Deviation

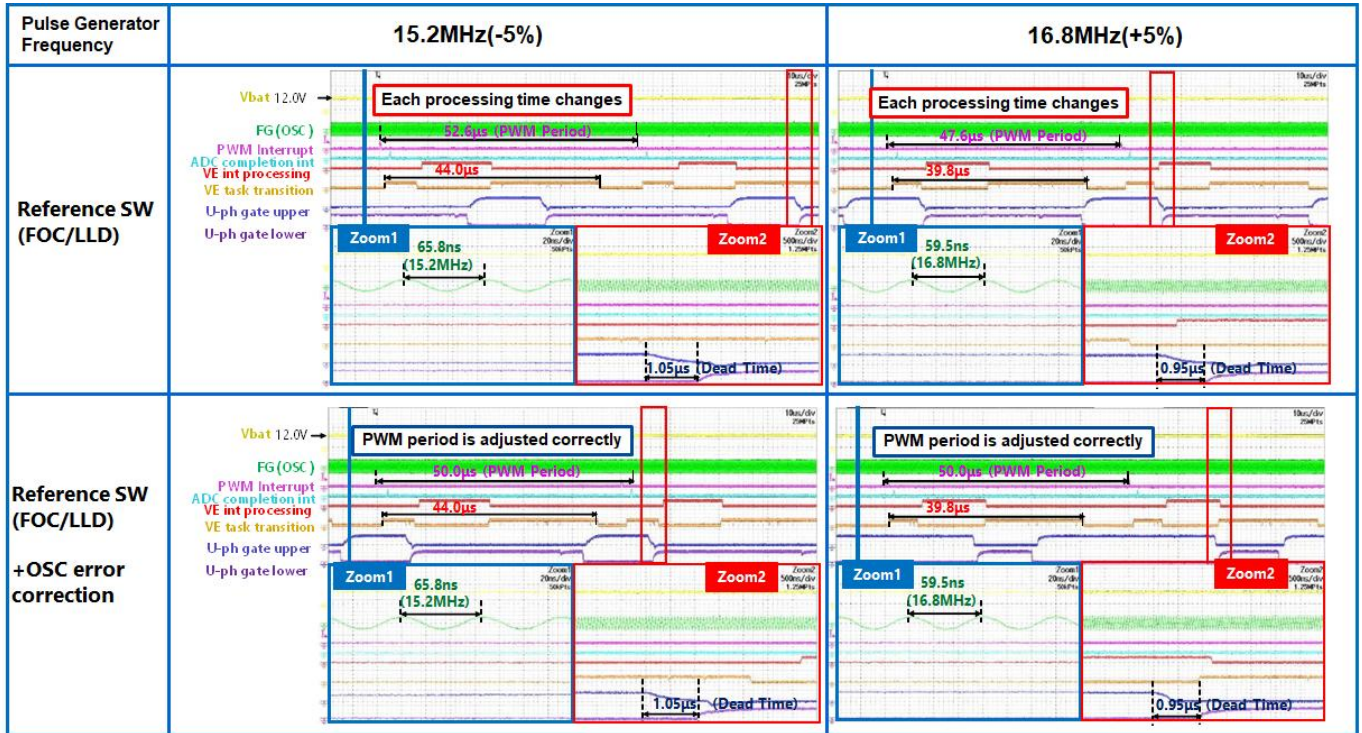


Figure 4.5 Experimental Results of Evaluation 3-b (2)

Results:

Under the condition simulating an oscillator frequency error of $\pm 5\%$, the PWM period can be compensated; however, the CPU and VE processing times cannot be compensated.

In addition, although the dead time can be compensated, it cannot be dynamically adjusted during motor control operation; therefore, it was excluded from the scope of this adjustment.

Furthermore, the achievable resolution for compensation is limited to $4/VECLK$ (Note 9), which does not allow for fine adjustment. When further tuning this parameter, these constraints must be taken into consideration.

(Note 9): $4/VECLK$: 66.7 ns at 60 MHz

4.1.4. Summary of Evaluation 3-b Results

Table 4.1 Experimental Results of Evaluation 3-b

Control SW Used	Oscillator Frequency	15.2MHz (-5%)	16MHz	16.8MHz (+5%)
Reference SW	PWM Period	52.6µs	50.0µs	47.6µs
	VE processing and VE interrupt processing time	44µs	41.8µs	39.8µs
	Dead Time	1.05µs	1.0µs	0.95µs
Reference SW + Frequency Error Compensation SW	PWM Period	50.0µs	50.0µs	50.0µs
	VE processing and VE interrupt processing time	44µs	41.8µs	39.8µs
	Dead Time	1.05µs	1.0µs	0.95µs

● **Supplementary Explanation for Evaluation 3-b (1)**

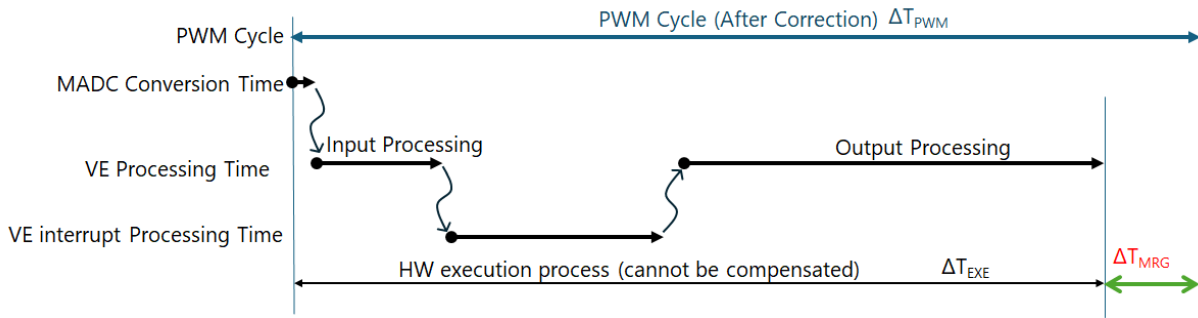


Figure 4.6 Relationship Between PWM Period and Hardware Execution Time

Figure 4.6 shows the relationship between the PWM period and hardware execution time (MADC conversion, VE processing, and VE interrupt processing).

In this error compensation method, the configuration is adjusted so that the PWM period remains constant even in the presence of oscillator frequency error.

On the other hand, the hardware execution time ΔT_{EXE} depends on the hardware and cannot be compensated.

The following describes the relationship between the above timing parameters.

$$\Delta T_{PWM} = \Delta T_{EXE} + \Delta T_{MRG}$$

At this point, for the control to operate correctly, the condition ΔT_{MRG} > 0 must be satisfied.

In other words, the hardware execution time must be shorter than the PWM period.

Therefore, when the oscillator frequency error is -5%, it is necessary to ensure that the following relationship holds.

$$\Delta T_{PWM} > \Delta T_{EXE} \times 1.05$$

(where each timing value is assumed to be the value under 0% frequency error)

● **Supplementary Explanation for Evaluation 3-b (2)**

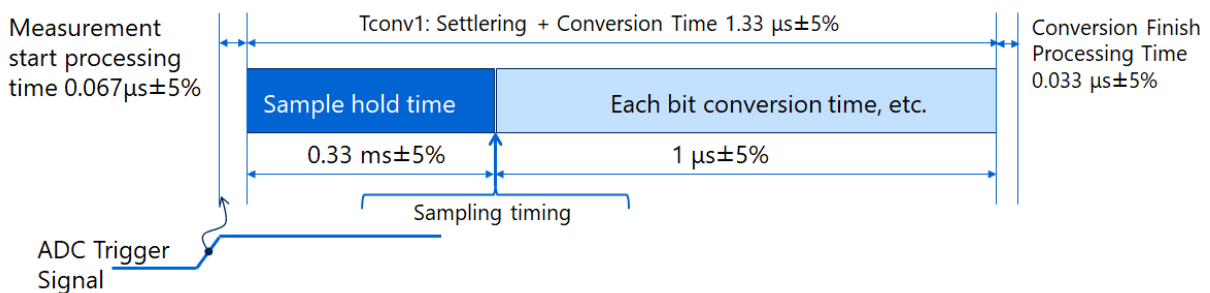


Figure 4.7 MADC Single-Command Trigger Mode Measurement Timing

Figure 4.7 is an excerpt from Chapter 22 of the User Manual (TB9M003FG_UM_J_Rev1.00), with supplementary explanations added.

In this figure, the key point is how the time from the ADC trigger signal to the falling edge of the sample-and-hold signal—that is, the sampling timing—is affected.

The sampling timing (measurement start processing time + sample-and-hold duration) is 0.40 μs ±5%.

A ±5% variation corresponds to approximately ±20 ns; therefore, this variation is considered to have no significant effect on practical use.

5. Summary

In this application note, we have presented, as a reference example, a method for achieving proper motor control by compensating in software for system errors caused by the internal oscillator frequency accuracy ($\pm 5\%$) of the TB9M003FG.

1. A method for obtaining measurement data of internal oscillator frequency error using hardware for LIN communication was presented as an example. Key considerations include ensuring stable communication conditions of the commander signal generation module, as well as selecting an appropriate baud rate that takes into account the effect of slew rate in the communication waveform.
 2. A method for writing the acquired frequency error data and reference data to Code Flash was presented as an example. In this scenario, the frequency error data and reference data are measured during the manufacturing process of the user's product, and the results are stored in Code Flash.
 3. A method was presented for compensating errors in motor control using the recorded frequency error data and reference data within the motor control program.
 4. Experimental Results
 - 1) Error measurement using LIN:

Error measurement using LIN communication is sufficiently practical when the effects of variations in the LIN communication environment are taken into account.
 - 2) Effects of frequency error ($\pm 5\%$):

A pseudo internal oscillator frequency error generation circuit was constructed, and operation was verified using the reference software. It was confirmed that the motor rotational speed is affected by the frequency error.
 - 3) Effect of error compensation software (Note 10):

The motor rotational speed is properly regulated.
However, it is necessary to consider dead time, as well as CPU and VE processing times, which cannot be compensated. In particular, it is important to ensure that the total processing time does not exceed the PWM period.
- (Note 10):
Error compensation software: Software with an error compensation function added to the reference software.

*SmartMCD™ is a trademark of Toshiba Electronic Devices & Storage Corporation.

*Other company names, product names, and service names may be trademarks of their respective companies.

Notes on Contents

1. Block Diagrams

Some of the functional blocks, circuits, or constants in the block diagram may be omitted or simplified for explanatory purposes.

2. Equivalent Circuits

The equivalent circuit diagrams may be simplified or some parts of them may be omitted for explanatory purposes.

IC Usage Considerations

Notes on Handling of Ics

- (1) The absolute maximum ratings of a semiconductor device are a set of ratings that must not be exceeded, even for a moment.
- (2) Use an appropriate power supply fuse to ensure that a large current does not continuously flow in case of over current and/or IC failure.

Points to Remember on Handling of ICs

- (1) Over current Protection Circuit
Over current protection circuits (referred to as current limiter circuits) do not necessarily protect ICs under all circumstances. If the Over current protection circuits operate against the over current, clear the over current status immediately.

- (2) Thermal Shutdown Circuit
Thermal shutdown circuits do not necessarily protect ICs under all circumstances. If the thermal shutdown circuits operate against the over temperature clears the heat generation status immediately.

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as "TOSHIBA".

Hardware, software and systems described in this document are collectively referred to as "Product".

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, Class 3 medical devices, and military vehicles and munitions. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**

Toshiba Electronic Devices & Storage Corporation

<https://toshiba.semicon-storage.com/>