ICM-42607x ICM-42670x FSYNC User Guide

TABLE OF CONTENTS

1 PURPOSE AND SCOPE

An application of TDK InvenSense motion sensors is image stabilization, which requires a synchronization pin to tag the image or video frame to the gyro data. This synchronization pin is known as FSYNC. The FSYNC pin has been implemented across numerous product lines and used by many customers. This document will go over the usage of FSYNC as it applies to Electronic Image Stabilization (EIS) for TDK InvenSense Motion Sensors in general, but also cover the benefits of EIS.

FSYNC is documented in detail in the component's product spec, but this document addresses questions on how exactly FSYNC is used and why it is implemented that way. This document serves as a beginner's guide and supplement to the FSYNC specs and aims to provide a base understanding.

2 GENERAL OVERVIEW OF FSYNC

The general idea of FSYNC is very simple.

- The FSYNC pin is an input meant to be connected to an image sensor's VSYNC (or Frame Sync) output pin. FSYNC can be used to synchronize camera image or video frames to the IMU for the purpose of electronic image stabilization.
- ICM-42607x and ICM-42670x devices incorporate an FSYNC ODR Delay Counter to accurately measure the time delta between the FSYNC signal and latest gyroscope sample event. This is especially important for EIS systems. The ICM-42607x and ICM-42670x uses registers and FIFO field for FSYNC ODR delay counter information storage (described in section 8).
- No cost added for additional performance.
- FSYNC is a simple hardware connection. The user just needs to run a trace from the image sensor's VSYNC output to the FSYNC pin. The FSYNC input pin on ICM-42607x and ICM-42670x is pin-7. If the user does not use the FSYNC function, please connect the pin-7 to GND to prevent floating input causing current leak.
- Software integration of FSYNC is very easy. It's essentially a set-it-and-forget-it operation.
- FSYNC reduces post-processing time and power.

3 ELECTRICAL CHARACTERISTICS

3.1 A.C. ELECTRICAL CHARACTERISTICS

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
DIGITAL INPUTS (FSYNC, SPC, SDI, CS)						
V_{IH}, High Level Input Voltage		0.7*VDDIO			V	1
V_{IL}, Low Level Input Voltage				0.3*VDDIO	V	
C_I, Input Capacitance			< 10		pF	

Table 1. A.C. Electrical Characteristics

3.2 ABSOLUTE MAXIMUM RATINGS

PARAMETER	RATING
Input Voltage Level (FSYNC, SCL, SDA)	-0.5V to VDDIO + 0.5V

Table 2. Absolute Maximum Ratings

4 WHY USE EIS?

When taking a still image or a video, any shaking of the hands can impact the quality of the image or video. Motion sensors can be used to detect this motion and apply a correction to improve the quality of the image or video. This can be done using optical image stabilization (OIS) or electronic image stabilization (EIS). OIS is a method in which the motion of the camera is detected, and a correction is applied to the camera or image sensor itself. EIS is a method in which the motion of the camera is detected, and a correction is applied digitally to the image or video frame by cropping to the appropriate portion of the image. EIS requires an FSYNC pin to synchronize image or video frame to camera motion. The motion vector calculation is independent of image processing. The downside to EIS is that the image or video frame must be cropped, which reduces the quality of the image or video. Figure 1 illustrates how EIS is applied.

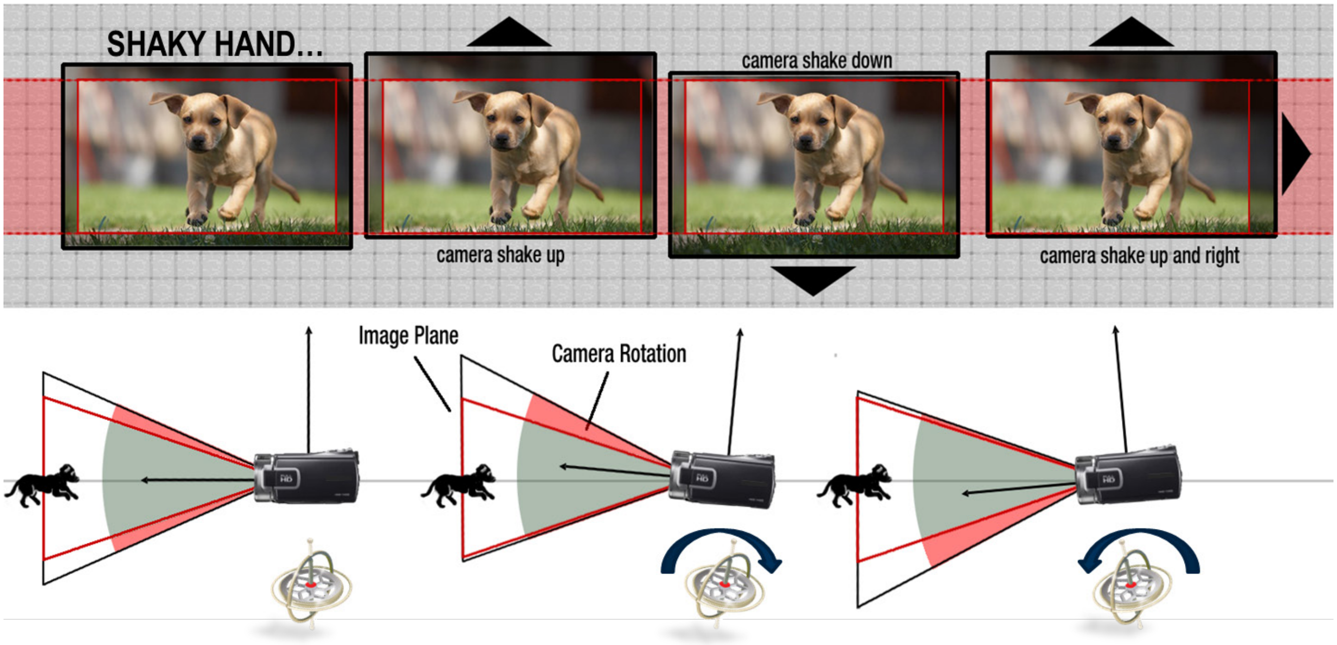

Figure 1. EIS Application

5 HOW FSYNC IS CONNECTED AND HOW IT WORKS

FSYNC is an input pin to the TDK InvenSense motion sensor that is meant to be connected to an image sensor's VSYNC pin for EIS applications. The block diagram in Figure 2 illustrates how FSYNC should be connected.

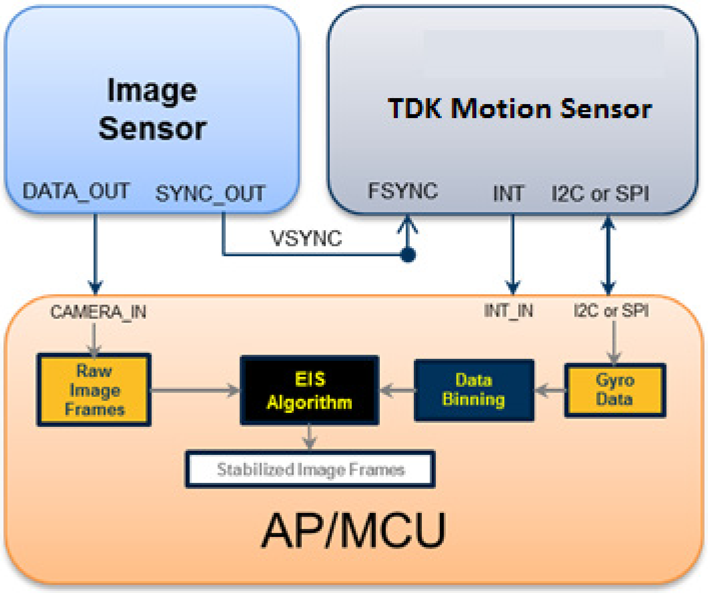

Figure 2. Block Diagram

The AP or MCU must associate or bin gyro samples with the correct frame. The image sensor sends the FSYNC pulse at the beginning of every frame to synchronize the gyro data to the image or video frame. The AP or MCU post processes the image by reading image sensor frame(s) and gyro data and associating the image frame with gyro samples. For EIS, the result is an image-stabilized frame. Inaccurate image-to-gyro data alignment can result in image blur and jitter.

6 FSYNC DETECTION

There are three ways for FSYNC to be detected, and there are pros and cons to each method of FSYNC detection.

1. An interrupt can be triggered on an FSYNC event.
2. The FSYNC bit can be polled on sensor data LSB (temperature, gyro, or accelerometer data).
3. The FSYNC information is present in corresponding FIFO data packet.

If the system is too busy, the host may not have time to read the interrupt status register on time. In that case, it is difficult to know when the FSYNC signal happened. However, if all the data is latched in the FIFO, it is easy to find out during which gyro data period the FSYNC happened, even if the system is busy. In the first method of FSYNC detection, the system must be able to read the gyroscope data every sample period.

The FSYNC pulse width does play a role in how well the gyro data output will be synchronized to the FSYNC event. In Figure 3, the FSYNC input is a short pulse. The FSYNC event will be associated to the data sampled after the FSYNC pulse.

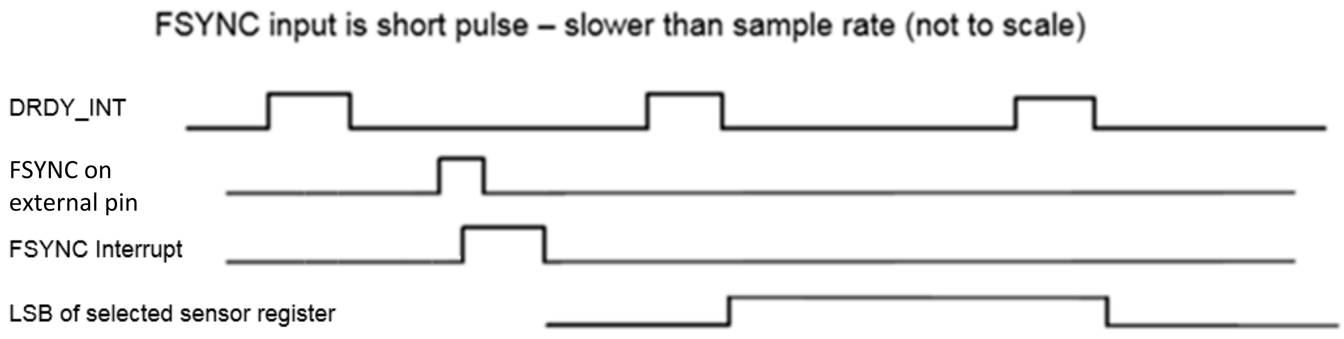
FSYNC input is short pulse – slower than sample rate (not to scale)
DRDY_INT
FSYNC on external pin
FSYNC Interrupt
LSB of selected sensor register

Figure 3. Example FSYNC Input Pulse

For ICM-42607x ICM-42670x the FSYNC detection circuit is edge-triggered with a minimum FSYNC pulse width of 100 ns.

7 ALIGNMENT OF FSYNC AND GYRO DATA

A time delta can exist between the FSYNC signal and the next gyro sample. This time delta results in less accurate results due to lack of alignment of FSYNC.

To improve alignment of the FSYNC signal and the gyro data, the time delta must be reduced as much as possible. Using high gyro data ODR can minimize the time delta between FSYNC and the gyro data. Figure 4 illustrates this phenomenon.

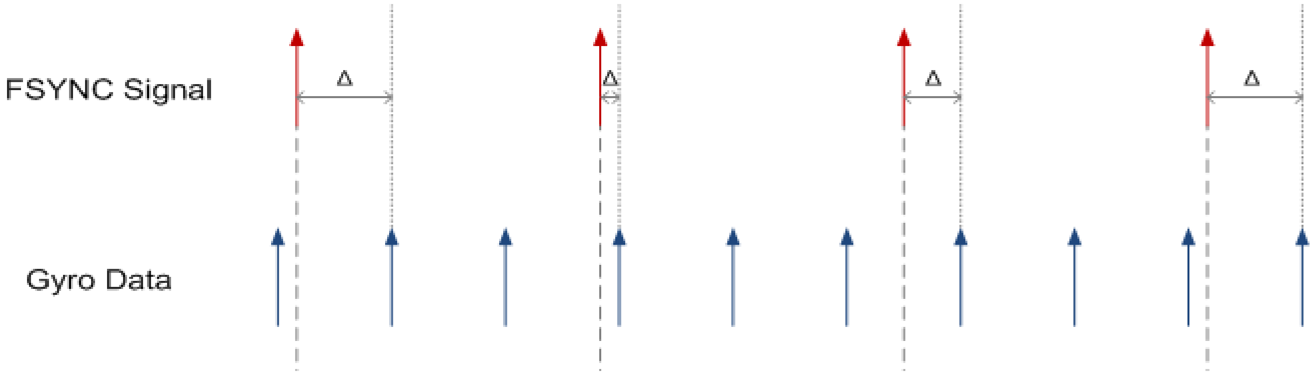

Figure 4. Timing Diagram to show time delta

To solve the alignment problem, ICM-42607x and ICM-42670x have an FSYNC ODR Delay Counter to store the time delta between FSYNC signal and the latest gyroscope data for providing interpolated gyroscope data for better EIS performance. The host can read the time delta from the FSYNC ODR Delay Counter and use this information to create an interpolated gyroscope for EIS video stabilization. Interpolated gyroscope data has better alignment with FSYNC than original data resulting in more accurate EIS performance.

The time delta can be read out from registers or from FIFO. Figure 5illustrates this phenomenon.

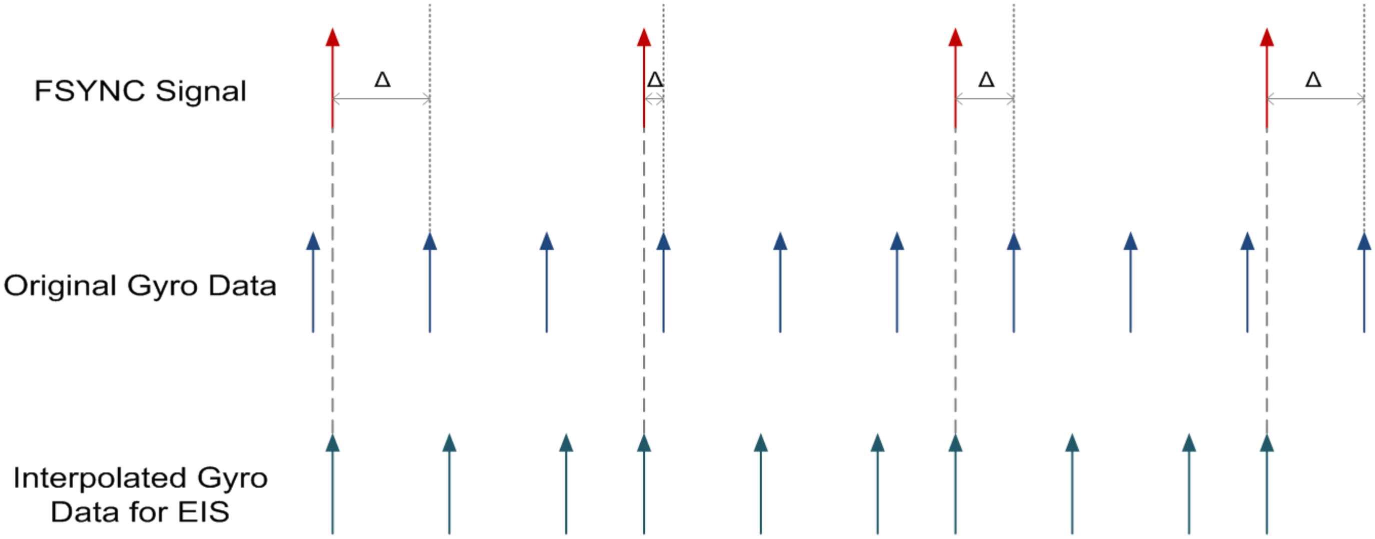

Figure 5. Timing Diagram to show interpolated gyro data

8 REGISTER INFORMATION

Please refer to the ICM-42607x and ICM-42670x register map for FSYNC related registers and bits.

8.1 ICM-42607X AND ICM-42670X CONFIGURATION

As mentioned above, ICM-42607x and ICM-42670x can save the time delta in register or in FIFO. The following sections will describe how to enable the FSYNC input, how to enable the time delta counter, and how to retrieve the time delta.

8.2 ENABLE FSYNC

To benefit from FSYNC feature in the ICM-42607x and ICM-42670x chips, configuring and enabling gyroscope in low-noise mode at appropriate FSR and ODR is recommended. An FSR of 500 dps or higher is required to capture the full range of the movement that need to be compensated. The recommended ODR is 200 Hz or faster.

- *(An Example) Set '8' (200Hz) in GYRO_ODR bitfield @bank0.0x20 (GYRO_CONFIG0)*
- *(An Example) Set '2' (500 dps) in GYRO_UI_FS_SEL bitfield @bank0.0x20 (GYRO_CONFIG0)*

- *Set '3' (Gyro in Low Noise Mode) in GYRO_MODE bitfield @bank0.0x1F* (PWR_MGMT0*)*

Then, we will configure which bit of the sensor data will be used to tag the FSYNC event. We recommend using the LSB of temperature to prevent any loss accuracy in gyro or accel resolution. The tag can then be read on the LSB of *TEMP_DATA0* @bank0.0x0A.

- *Set '1' (TEMP_OUT) in FSYNC_UI_SEL bitfield @bank1.0x03 (FSYNC_CONFIG)*

Finally, you need to enable the FSYNC feature on the time stamp register and on the FIFO. When an FSYNC event occurs, the timestamp will be replaced with the FSYNC delay counter.

- *Set '1' in TMST_FSYNC_EN bitfield @bank1.0x00 (TMST_CONFIG1)*
- *Set '1' in FIFO_TMST_FSYNC_EN bitfield @bank1.0x05 (FIFO_CONFIG5)*

8.3 FSYNC DETECTION AND TIME DELTA RETRIEVING

There are two methods to detect FSYNC and read out the time delta between FSYNC and the latest ODR. The two methods will be described below.

8.3.1 Through Registers

FSYNC can trigger interrupt output from ICM-42607x and ICM-42670x. The FSYNC interrupt can be assigned to INT1 or INT2.

FSYNC_INT1_EN bitfield in bank0.0x2B (INT_SOURCE0) and FSYNC_INT2_EN bitfield in bank0.0x2D (INT_SOURCE3) are the bits to assign the FSYNC interrupt output.

Status bit FSYNC_INT in bank0.0x3A (INT_STATUS) will be set when FSYNC is detected.

Register TMST_FSYNCH/L @bank0.0x17/8 store the time delta from the rising edge of FSYNC to the latest ODR. User can read out the 16bit data from the two registers.

8.3.2 Through FIFO

Once FSYNC is enabled, for each packet read from FIFO, FSYNC is signaled when bit timestamp_fsync in packet header is set b’11.

For that specific packet, regular timestamp field of the FIFO packet is replaced with the FSYNC delay counter.

FIFO Packet fields	Bitfield	Value
Header field	msg	b’0
	accel	b’1
	gyro	b’1
	20b	b’X
	timestamp_fsync	b’11
	odr_accel	b’X
	odr_gyro	b’X
Accel field		6 bytes
Gyro field		6 bytes
Temperature field		1 or 2 bytes
Timestamp field		16b FSYNC delay counter

Table 3. FIFO packet when FSYNC occurs

If FSYNC signal is triggered at less than 50 µs before the gyro sampling is done, the timestamp (FSYNC) field in FIFO will contain 0xFFFF value. It is up to EIS system to handle this specific case depending on platform requirement.

8.4 CALCULATION

To retrieve the time at which FSYNC occurred, you can use the following equation:

$$t_{FSYNC} = t_{last\ gyro} + ODR_{gyro} - fsync\_delay$$

8.5 DISABLE FSYNC

To disable FSYNC feature, just clear the following bits:

- *Set ‘0’ in FSYNC_UI_SEL bitfield @bank1.0x03 (FSYNC_CONFIG)*
- *Set ‘0’ in TMST_FSYNC_EN bitfield @bank1.0x00 (TMST_CONFIG1)*
- *Set ‘0’ in FIFO_TMST_FSYNC_EN bitfield @bank1.0x05 (FIFO_CONFIG5)*

9 REVISION HISTORY

REVISION DATE	REVISION	DESCRIPTION
03/24/2022	1.0	Initial Draft

This information furnished by InvenSense or its affiliates ("TDK InvenSense") is believed to be accurate and reliable. However, no responsibility is assumed by TDK InvenSense for its use, or for any infringements of patents or other rights of third parties that may result from its use. Specifications are subject to change without notice. TDK InvenSense reserves the right to make changes to this product, including its circuits and software, in order to improve its design and/or performance, without prior notice. TDK InvenSense makes no warranties, neither expressed nor implied, regarding the information and specifications contained in this document. TDK InvenSense assumes no responsibility for any claims or damages arising from information contained in this document, or from the use of products and services detailed therein. This includes, but is not limited to, claims or damages based on the infringement of patents, copyrights, mask work and/or other intellectual property rights.

Certain intellectual property owned by InvenSense and described in this document is patent protected. No license is granted by implication or otherwise under any patent or patent rights of InvenSense. This publication supersedes and replaces all information previously supplied. Trademarks that are registered trademarks are the property of their respective companies. TDK InvenSense sensors should not be used or sold in the development, storage, production or utilization of any conventional or mass-destructive weapons or for any other weapons or life threatening applications, as well as in any other life critical applications such as medical equipment, transportation, aerospace and nuclear instruments, undersea equipment, power plant equipment, disaster prevention and crime prevention equipment.

TDK InvenSense