Drive for better vision

HM0360 Motion Detection Setting for Context B

April 8, 2025

Himax Technologies, Inc.
奇景光電股份有限公司

HM0360 Motion Detection Overview

1.1 MD overview

The HM0360 is an ultra-low power 1/6" VGA CMOS image sensor with an embedded Motion Detection **(MD)** feature. The MD is comprised of 16 x 16 Region of Interest **(ROI)**, where each ROI detects motion independently. The status **(e.g. motion or no motion)** of each MD window is indicated by individual motion flag, which can be read through serial register or embedded data. The motion detection sensitivity can be programmed globally, such that setting parameters are uniformly applied to each window.

An MD event can be programmed to trigger the sensor interrupt. The interrupt status can be read through register and embedded data stream. The interrupt can also drive the INT digital output pin that can be connected to an external processor or directly to a LED as a status indicator. As shown in the block diagram below, the interrupt can be qualified based on a programmable minimum MD block number threshold, and the interrupt can be extended or delayed **(MD latency)**. Additionally, the HM0360 video data interface can be disabled while maintaining the INT output pin function to support use-cases such as Wake from Motion **(WfM)**.

HM0360 Motion Detection Function Block

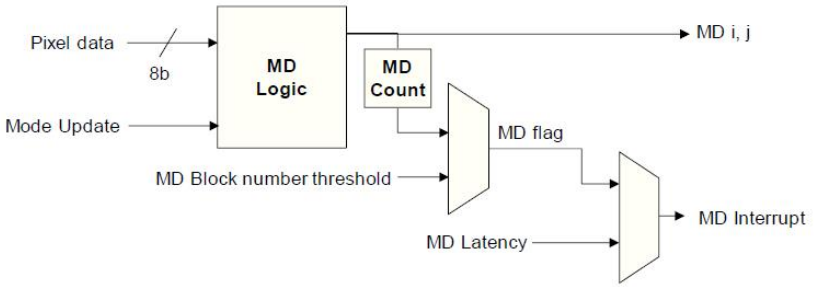

Figure 1.1: MD function simplified block diagram

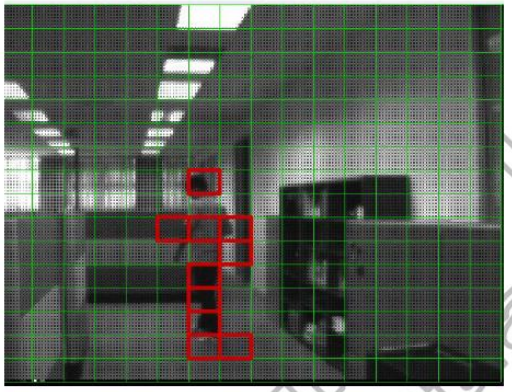

Figure 1.2: MD function demonstration on Himax evaluation kit

In the example above, a 16x16 Motion Detection ROI borders are generated by the Himax HiMOS evaluation software based on the sensor settings. The red boxes are drawn by the HiMOS software based on decoded embedded data of positive MD flags. In this example of a person walking across an office space, the HM0360 MD detected movement across multiple MD ROI, which can trigger the interrupt to wake a host processor.

HM0360 Motion Detection Setting for Context B

- ~\allon_sensor_tflm_freertos\cis_sensor\cis_hm0360\HM0360_OSC_Bayer_640x480_setA_VGA_setB_QVGA_md_8b_ParallelOutput_R2.i
- This sensor setting selects context B as Motion Detection setting
- Please use context B motion detection registers to fine-tune the sensitivity of motion detection

Address main	Address context A	Address context B	Byte	Register name	Description
0x2080	0x354B	0x35A5	[0]	MD_CTRL	[0]: Motion detect enable
0x2081	0x354D	0x35A7	[7:0]	ROI_START_END_V	[7:4]: ROI_END_V [3:0]: ROI_START_V
0x2082	0x354E	0x35A8	[7:0]	ROI_START_END_H	[7:4]: ROI_END_H [3:0]: ROI_START_H
0x2083	-	-	[6:0]	MD_TH_MIN	Threshold min value
0x2084	0x3550	0x35AA	[5:0]	MD_TH_STR_L	Threshold strength
0x2085	0x354F	0x35A9	[5:0]	MD_TH_STR_H	Threshold strength
0x2099	-	-	[6:0]	MD_LIGHT_COEF	md_light_coef
0x209A	-	-	[7:0]	MD_IIR_PARAMETER	IIR Filter
0x209B	0x354C	0x35A6	[7:0]	MD_BLOCK_NUM_TH	MD_block_number threshold
0x209C	-	-	[4:0]	MD_LATENCY	MD_latency_frame
0x209D	-	-	[7:0]	MD_LATENCY_TH	[7:4]: md_latency_s_threshold [3:0]: md_latency_m_threshold
0x209E	-	-	[3:0]	MD_CTRL1	MD_interrupt_control [2]: RESERVED [1]: motion interrupt enable [0]: motion interrupt select 0: original flag 1: latency flag
0x20A1 ~ 0x20C0	-	-	[7:0]	MD_ROI_OUT	Read only motion flag for each MD block

Table 1.1: MD register table

MD Region of Interest Setting

2.3 MD region of interest setting

The active region of the Motion Detection is based on a simple coordinate system from 0 to 15 and is selected by ROI_VERTICAL and ROI_HORIZONTAL register as shown in the figure and table below.

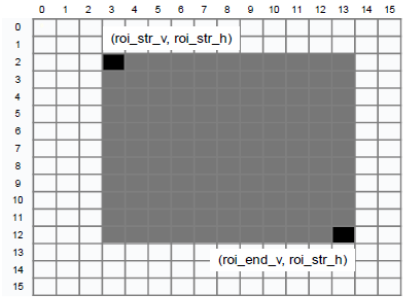

Figure 2.2: MD window and coordinates

Address main	Address context A	Address context B	Byte	Register name
0x2081	0x354D	0x35A7	[7:4]	ROI_END_V
			[3:0]	ROI_START_V
0x2082	0x354E	0x35A8	[7:4]	ROI_END_H
			[3:0]	ROI_START_H

Table 2.3: MD ROI registers

An example of the register programming based on Figure 2.2: MD window and coordinates is as follows:

- ***ROI_START_V (roi_str_v) = 2 & ROI_END_V (roi_end_v) = 12***
  Set register 0x2081 (for main register) = 0xC2
- ***ROI_START_H (roi_str_h) = 3 & ROI_END_H (roi_end_h) = 13***
  Set register 0x2082 (for main register) = 0xD3

MD Sensitivity Setting

2.4 MD sensitivity

Each ROI independently detects motion by comparing the algorithmic values of the current frame and previous frames. Three example sets of register are provided below, based on “Low”, “Medium”, and “High” Sensitivity.

Context B Register address	Register name	Low sensitivity	Medium sensitivity	High sensitivity
0x35AA	MD_TH_STR_L	0 x 30	0 x 20	0 x 10
0x35A9	MD_TH_STR_H	0 x 30	0 x 20	0 x 10
0x35A5	MD_LIGHT_COEF	0x41	0x31	0x21
0x209A	MD_IIR_PARAMETER	0 x 00	0 x 80	0 x F0

Table 2.4: MD registers sample for different motion sensitivity level

MD Block Number Threshold Setting

MD block number threshold

Register 0x35A6 sets the threshold for the number of active MD ROI with positive event to trigger the MD motion flag. If the number of motion blocks **(which are in the activated ROI area)** is greater than the user defined number **(MD_block_number threshold)**, then the motion flag is true. The minimum value is 1 and the maximum value is the number of MD ROI – 1.

		Context B			
0x209B	0x354C	0x35A6	[7:0]	MD_BLOCK_NUM_TH	MD_block_number threshold

MD Interrupt Latency

3.2 MD interrupt latency

The MD latency can extend or delay the MD Interrupt based on programmable number of frames. To enable MD latency, set register 0x2080[5:4] to the selected latency mode. To enable the Motion Interrupt based on the latency control, set 0x209E[1] to 1.

Address	Byte	Register name	Description
0x2080	[7:0]	MD_CTRL	[7]: RESERVED [6]: RESERVED [5:4]: MD latency select [3:2]: RESERVED [1]: RESERVED [0]: Motion detect enable
0x209C	[4:0]	MD_LATENCY	MD_latency_frame
0x209D	[7:0]	MD_LATENCY_TH	[7:4]: md_latency_s_threshold [3:0]: md_latency_m_threshold
0x209E	[3:0]	MD_CTRL1	MD_interrupt_control [2]: RESERVED [1]: Motion interrupt enable [0]: Motion interrupt select 0: Original flag 1: Latency flag

Table 3.2: MD latency registers

MD Interrupt Latency Mode 1

3.2.1 Latency mode 1 – Extend interrupt from motion frame

In this mode, once the motion flag is HIGH, it will remain HIGH for a user defined number of frames as programmed by register 0x209C[4:0]. When the motion interrupt is high based on the latency extension, new motion flag will not be extended.

For example, if register 0x209C[4:0]=0x02, then the motion interrupt will extend for two frames from the first frame with a motion event. In the second set, a motion flag occurred while the motion interrupt is already high, as such, the motion interrupt is not extended further.

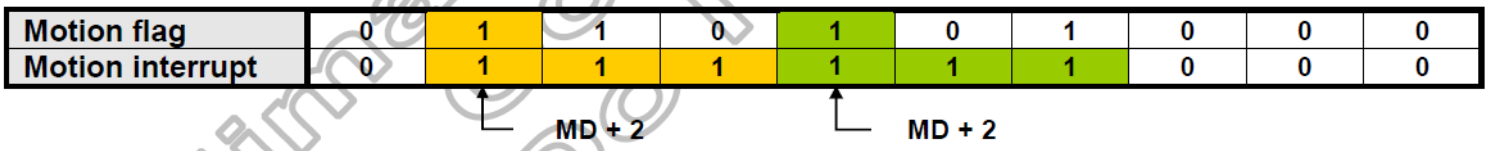

Figure 3.1: MD example – Latency mode 1

MD Interrupt Latency Mode 2

3.2.2 Latency mode 2 – Extend interrupt from last motion frame

In this mode, once the motion flag is HIGH, it will remain HIGH for a user defined number of frames as programmed by register 0x209C[4:0]. When a new motion flag occurs, even when the Interrupt is high from the previous latency extension, the Interrupt will be extended by the programmed count.

For example, if register 0x209C[4:0]=0x02, then the motion interrupt will extend for two frames from the first frame with a motion event. When the motion flag is HIGH for any frame, the latency will continue to extend the Interrupt.

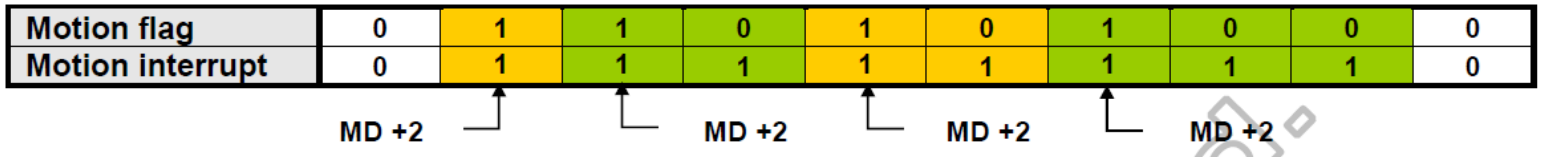

Figure 3.2: MD example – Latency mode 2

MD Interrupt Latency Mode 3

3.2.3 Latency mode 3 – Delay motion interrupt

In this mode, the motion interrupt is delayed by the number of consecutive frames with motion flag, as programmed by register 0x209D[3:0]. The motion interrupt will remain high for the number of consecutive frames without motion flag, as programmed by register 0x209D[7:4].

For example, if register 0x209D[3:0]=0x02 and register 0x209D[7:4]=0x02, then the motion interrupt will be delayed until two consecutive frames of motion. Once the motion interrupt is HIGH, it will remain HIGH until two consecutive frames without motion.

Motion flag	1	1	1	0	0	1	0	0	0	1
Motion interrupt	0	0	1	1	1	1	1	1	0	0

MD +2 No MD + 2

Figure 3.3: MD example – Latency mode 3

Current MD Setting

- ~\allon_sensor_tflm_freertos\cis_sensor\cis_hm0360\HM0360_OSC_Baye
  r_640x480_setA_VGA_setB_QVGA_md_8b_ParallelOutput_R2.i

```
0x35a5, 0x41    // Context B MD_LIGHT_COEF [0x41/0x31/0x21]
0x35a6, 0x04    // Context B MD_BLOCK_NUM_TH
0x35a7, 0xe0    // Context B ROI_V
0x35a8, 0xf0    // Context B ROI_H
0x35a9, 0x30    // Context B MD_TH_STR_H [0x30/0x20/0x10]
0x35aa, 0x30    // Context B MD_TH_STR_L [0x30/0x20/0x10]

0x2080, 0x31    // MD_CTRL [5:4]: MD latency select, [0]: Motion detect enable
0x209a, 0x00    // MD_IIR_PARAMETER [0x00/0x80/0xf0]
0x209d, 0x33    // MD_LATENCY_TH [7:4]: s, [3:0]: m
0x209e, 0x07    // MD enable latency
```

Drive for better vision