

Miramar Camera GenICam Interface

Revision History

Date	Rev No.	Description	By
Sep 2023	1	Initial draft	Jeremy Hong
6/10/2024	2	Add register definitions	Bing Wen
6/12/2024	2.01	Add more register definitions	Bing Wen
6/14/2024	2.02	Add user data flash address	Bing Wen
7/16/2024	3.0	Modified Register definitions for firmware 3.0	Bing Wen
8/23/2024	3.1	Register value change for firmware >3.1, FPGA >=1.6.7	Bing Wen
9/30/2024	3.1.1	Remove invalid registers	Bing Wen
2/24/2025	3.3	Register value change for firmware >3.2, FPGA >=1.7.6	Bing Wen
3/21/2025	3.3.1	Change length of 0x27000008 From 2 to 4	Bing Wen
10/20/2025	4.0	Register value change for firmware > 4.0, FPGA >=1.7.38	Bing Wen

Contents

REVISION HISTORY	1
INTRODUCTION	2
GENICAM PROTOCOL	2
HARDWARE IMPLEMENTATION.....	2
REGISTER DEFINITIONS	2
EXAMPLE TASKS	3
READ MCU FIRMWARE VERSION AND FPGA VERSION	3
NUC USING EXTERNAL SHUTTER	4
OUTPUT RAW DATA	4

OUTPUT YUYV DATA FOR OPERATING SYSTEM CAMERA APPS	4
SAVE DATA TO FLASH	4
SAVE CURRENT SETTINGS AS DEFAULT	4

Introduction

Miramar cameras use Genicam protocol. This document will discuss implementation of writing and reading registers from the camera, and registers that control the camera behavior.

GenICam Protocol

For details on how to format writes and reads for GenICam protocol, please refer to the GenICam document “GenICam GenCP Generic Control Protocol Version 1.3”.

Hardware implementation

Miramar Cameras with USB_C connector implemented the GenICam interface on serial ports using USB. The USB port has VID=0x1FC9 and PID=00A3. If the correct serial port is chosen, reading 64 bytes from address 0x08 will return “OBSIDIAN SENSORS INC.”.

Cameras with MIPI/GMSL connector has GenICam protocol implemented through I2C bus. It is documented separately.

The existing communication buffer size is **4096** bytes. Do not send or request more than 4096 bytes of data in each transaction.

Register definitions

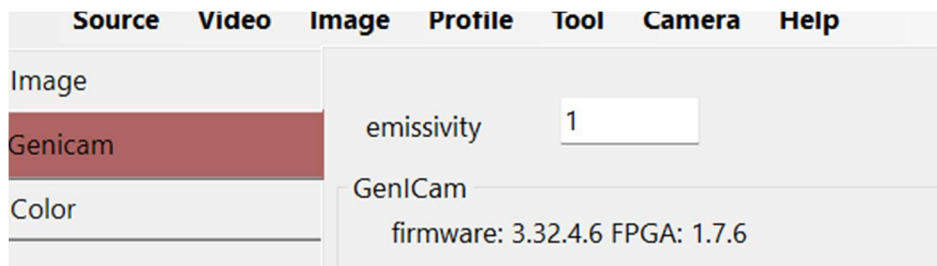
Adress	Length (bytes)	Read/Write	functions
0x020C	64	R	Firmware Version String (ASCII)
0x1000	1	R	Camera board temperature, in Centigrade
0x100E	1	R	shutter temperature = (value[0x100E]+256*value[0x100F])/100, in Centigrade
0x100F	1	R	
0x1010	1	R	FPGA Major version
0x1011	1	R	FPGA Minor version
0x1012	1	R	FPGA Sub Minor version
0x2001	1	R/W	0/1: Video stream disable/enable
0x24000008	4	R/W	Bit 4:0: number of frames to collect during background accumulation
0x2005	1	R/W	0: shutter disabled

			1: shutter enabled, odd polarity 2: shutter enabled, even polarity
0x200A	1	R/W	Colormap index
0x200D	1	W	Write 1 to save current MCU settings as default setting, which is loaded after every reboot
0x2030	1	W	Write 1 to save current FPGA settings as default setting, which is loaded after every reboot
0x200F	1	R/W	Automatic shutter calibration 0: disable 1:enable
0x2010	1	W	X10 = Maximal number of seconds between shutter calibration
0x2011	1	R/W	Delay between shutter movement and background accumulation, in number of frames
0x2035	1	W	Write 1 to reboot immediately
0x2036	1	W	Perform background accumulation
0x2B000000	4	R/W	Bit 0: colormap enable.
0x2C000000	4	R/W	Bit 0: AGC enable Bit 1: CLAHE enable Bit 2: black hot/ white hot Bit 3: signed/unsigned. Don't change other bits.
0x609D3000	2,650,111	R/W	0x609D3000 - 0x60c59FFF stores user defined data in flash. Users can use these registers to store data that needs to be preserved after power cycles. - Data is stored in sectors. Each sector is 4096 bytes in size. - The starting address must be dividable by 4096. The address should have format 0Xxxxx x000. User data should be saved when video stream is turned off

Example tasks

Read MCU firmware version and FPGA version

The MCU firmware version and FPGA version will be displayed by ThermalCam App once a camera is connected successfully.



By code, for MCU firmware version, read 64 bytes from 0x020c. For FPGA version read 4 bytes from 0x2000_0014.

Example code in c#:

```
var value = readRegister(0x020c, 64, CloseAfter: false);
```

```
_FirmwareVersionString = System.Text.Encoding.Default.GetString(value);

value = readRegister(0x20000014, 4, CloseAfter: false); ;
_FPGAVersionString = value[2].ToString() + "." + value[1].ToString() + "." +
value[0].ToString();
```

NUC using external shutter

Disable shutter, write 0 to 0x2005

Put blackbody in front of the lens, then write 1 to 0x2036

Output raw data

Disable AGC/CLAHE by reading/writing to 0x2C000000

Disable colormap by writing to 0x2B000000

Output YUYV data for operating system camera apps

Disable AGC/CLAHE by reading/writing to 0x2C000000

Disable colormap by writing to 0x2B000000

Save Data to flash

Disable video stream, write 0 to 0x2001

Write data to address between 0x609D3000 and 0x60c59FFF in unit of sectors.

Reboot by writing 1 to 0x2035

Save current settings as default

write 1 to 0x200D

write 1 to 0x2030