WAT-910BD

H/W MANUAL

Ver.1.0

Nov 06, 2013

Watec Co.,Ltd.

History

Ver. No.	Date	Changes	Remark
1.00	2013.11.06		First edition

Contents

1. INTRODUCTION

This manual is to explain the hardware features of the WAT-910BD.
Description requiring electrical knowledge needed to properly use the camera is included in the contents of this document.

Please read this document carefully, and use and/or install the camera after full understanding.
Under improper installation, it may cause serious damage to the connected equipment and the camera.
Please contact the distributor or dealer from which the WAT-910BD was purchased, if you do not understand the installation, operation or safety instructions laid out in our manuals.

How to use the OSD(On-Screen Display) menu and setting each function, for more information, please refer to "WAT-910BD OSD Manual" together. In addition, how to operate and install the camera, please refer "WAT-910BD Operation Manual" together with this document.

2. LEGAL DISCLAIMER

Design and specifications are subject to change for improvement without notice.

Watec is not responsible for any inconvenience or damages to the video and monitoring recording equipment caused by misuse, mis-operation or improper wiring of our equipment.

Technical information contained herein is for reference only and does not convey any license by any implication or otherwise under any intellectual property right or other right of Watec or third parties.
Watec does not assume responsibility for any right infringements arising out of the use of this information.

3. OVERVIEW

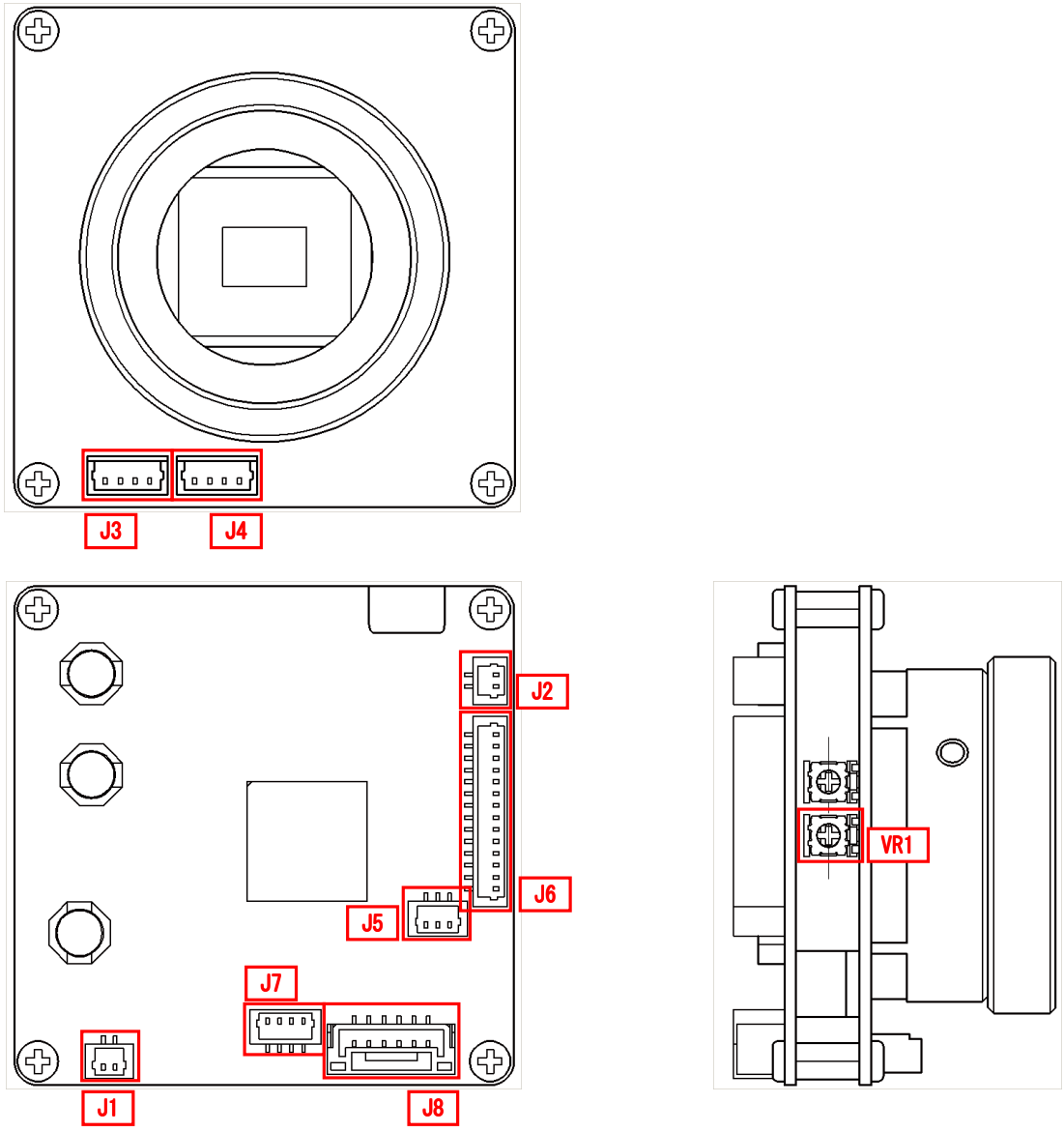
WAT-910BD is an ultra-high sensitivity B/W board camera equipped with 1/2 " format CCD.
By adopting the latest high-sensitivity CCD and DSP with a variety of functions to the signal processing unit, WAT-910BD realized various functions and higher performances.

* In addition to the automatic exposure control function, high speed shutter and the electronic iris, WAT-910BD is equipped with low-speed shutter (multi-field integration) function for dark field imaging. High-quality imaging is now possible by the combination of these functions without the illumination of the object.
* By adopting the latest DSP to the video processing block, a variety of digital signal processing functions such as noise reduction, dynamic range correction, white blemish correction and motion detection are available.
* For setting each functions, you can use user-friendly OSD menu or SPI communication to directly set the functions.
* Digital output (parallel output) terminal is equipped for the video output, in addition to the conventional analog output
* Since WAT-910BD is small in size (42mm square) and lightweight (approx.40g)design, making it ideal for building into a variety of applications.

4. DESCRIPTION OF PARTS

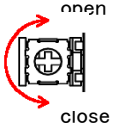
The name and function of each parts are as follows.

When wiring, please be careful of the following pin arrangement and connect correctly.

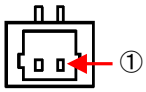
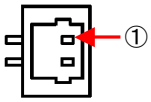
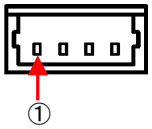
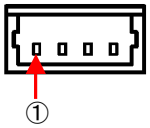
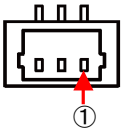
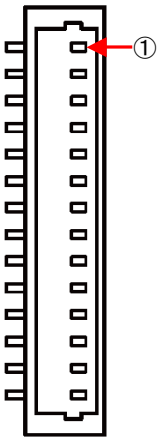
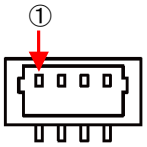


(Fig. 4-1) Location of functional connectors

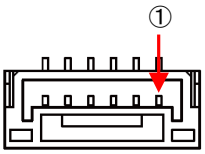
(Fig. 4-2) DC IRIS LENS adjustment VR

	Parts shape	Name	Description
VR1		Adjustment VR for DC IRIS Lens	clockwise : DC iris Lens toward to open counterclockwise : DC iris Lens toward to close *This adjustment is working only J3 output

(Fig. 4-3) Connector Pin Function

	Parts Shape	Name	Parts No.	manufacturer	Pin No.	Description	I/O
J1		POWER	BM02B-SRSS	JST	①	power in (DC+12V)	I
					②	power GND(COMMON)	GND
J2		VIDEO OUT	BM02B-SRSS	JST	①	VIDEO OUTPUT	O
					②	VIDEO GND(COMMON)	GND
J3		DC IRIS CTL.	53047-0410	molex	①	Sense−	O
					②	Sense+	-
					③	Drive+	O
					④	Drive−	GND
J4		VIDEO IRIS CTL.	53047-0410	molex	①	Vcc(DC+12V) OUT	O
					②	NC(Non Conection)	-
					③	IRIS OUT	O
					④	GND(COMMON)	GND
J5		GPIO	BM03B-SRSS	JST	①	MOTION DETECT output	O
					②	GND (common)	GND
					③	(reserved)	-
J6		Digital Video Output	BM14B-SRSS	JST	①	Vertical Blanking	O
					②	Horizontal Blanking	O
					③	GND (common)	GND
					④	CLOCK	O
					⑤	GND (common)	GND
					⑥	DATA 7(MSB)	O
					⑦	DATA 6	O
					⑧	DATA 5	O
					⑨	DATA 4	O
					⑩	DATA 3	O
					⑪	DATA 2	O
					⑫	DATA 1	O
					⑬	DATA 0(LSB)	O
					⑭	GND (common)	GND
J7		SPI Communication	BM04B-SRSS	JST	①	SPI SLD (active L)	I
					②	SPI SCL	I
					③	SPI SDA	I/O
					④	GND (common)	GND

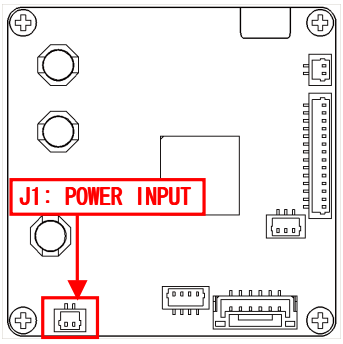
(Fig.4-3) Connector Pin Function (continued)

	Parts Shape	Name	Parts No.	manufacturer	Pin No.	Description	I/O
J8		OSD CTL. (5key interface)	BM06B-GHS-TBT	JST	①	GND (COMMON)	GND
					②	SET	I
					③	LEFT	I
					④	RIGHT	I
					⑤	DOWN	I
					⑥	UP	I

4.1 POWER INPUT

(1) Input Voltage

Recommended supply voltage range and the absolute maximum ratings are as follows.



(Fig.4-4) Power input specifications	
item	value (range)
Supply Voltage Range (Reccommended)	12[V]+/-10[%]
Supply Voltage Range (Capable)	DC +8.0 - 14.0[V]
Absolute Maximum Ratings	less than DC +16.0[V]

Normally, We recommend to use WAT-910BD in "Supply Voltage Range(recommended) ".
Using in range of "Supply Voltage Range(Capable)", the camera may be NOT damaged.
In this case(especially lower voltage condition), please check followings.

*When shooting a significantly brighter object, blooming or smear increases might occur.

*When using VIDEO IRIS LENS, the lens iris does not open, or the iris opening and closing is slow.

In the case of using "supply voltage range(capable)", please adopt it after fully evaluation of proper operation of the camera/system.

In case of using power supply voltage exceed the maximum rating, it will cause a serious damage to camera and/or to other equipment connecting to the camera.

(2) Power Consumption

Power consumption by the camera is about 1.32W. (Supply voltage: DC +12V, current consumption: 110mA)
This value is Typical. (It has been measured under the following conditions)

*Measuring condition:

1) The camera function settings are all default.(equal values after running the OSD menu "RESET")
2) Using a manual iris lens, the lens iris is fully closed.

The power consumption of the camera and efficiency depend on the supply voltage.
Under the above conditions, the measurement values of the power consumption and current consumption related to the supply voltage are shown in following table.

(Fig.4-5) Power Consumption(Typical)									
Supply Voltage(V)	8.0	8.5	9.0	9.5	10.0	10.5	11.0	11.5	12.0
Current Consumption(mA)	148.8	141.6	134.9	128.6	123.8	119.0	114.8	111.0	107.0
Power Consumption(W)	1.19	1.20	1.21	1.22	1.24	1.25	1.26	1.28	1.28
Supply Voltage(V)	12.5	13.0	13.5	14.0					
Current Consumption(mA)	104.2	101.1	98.4	95.7	*Note:				
Power Consumption(W)	1.30	1.31	1.33	1.34	Typical value, Not guaranteed.				

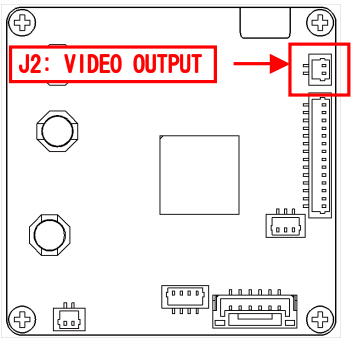
Please note the following points in the selection and design of the power supply used.

* When a auto iris lens (Video iris/ DC iris) is used, the power consumption will increase.
  The increment of current consumption is related to lens specification.
* By the combination of the camera functions to be used, the current value will change.

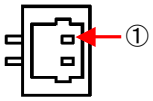
Please select or design power supply of enough performance, in consideration of camera usage.

4.2 VIDEO OUTPUT

An analog video output terminal of 75 ohm impedance.
75 ohm termination is needed in the video signal receiver equipment.

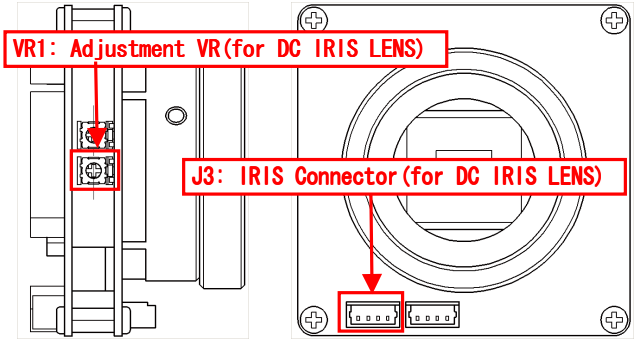


(Fig. 4-6) VIDEO output

	Parts Shape (Pin No.)		Pin Name	Description
J2		①	VIDEO OUTPUT	75 ohm impedance
		②	VIDEO GND (COMMON)	

4.3 DC IRIS LENS CONTROL

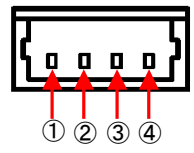
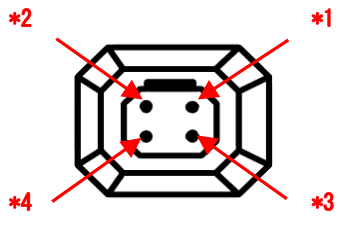
This connector is used to connect a DC IRIS LENS.
When connecting the lens, be careful to wire after confirming the pin assignments.



(1) Wiring to the lens

Pin assignments of JEITA standard iris connector and J3 connector terminal have one-to-one correspondence as following.

(Fig. 4-5) Pin assignments of DC IRIS connector

	Parts Shape (Pin No.)		Pin Name	female IRIS connector (JEITA spec.)	
J3		①	Sense−	*1	
		②	Sense+	*2	
		③	Drive+	*3	
		④	Drive−	*4	

(2) Adjustment method of the lens iris

The lens iris is adjusted with next steps.

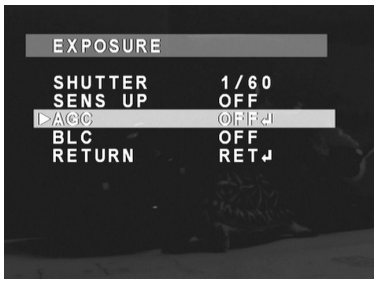
STEP1: Fix the shutter speed by using OSD menu. (SETUP->EXPOSURE->SHUTTER)
Recommended setting of SHUTTER is either OFF(EIA:1/60, CCIR:1/50) or FL(EIA:1/100, CCIR:1/120).

STEP2: Turn off "SENSE UP" function.

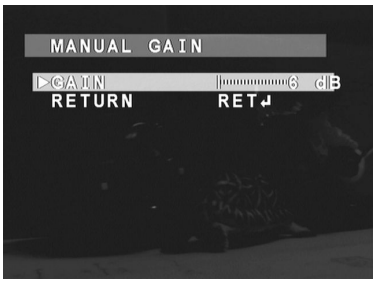
STEP3: Turn off "AGC" function.
Set "GAIN" to the minimum value (6dB) by using "MANUAL GAIN" menu.

STEP4: Return to the EXPOSURE menu, by adjusting the VR1, set the proper lens iris.

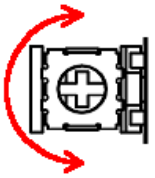
STEP5: If the adjustment of lens iris is correct, when AGC is switched to LOW from OFF, the video signal will be a little brighter.



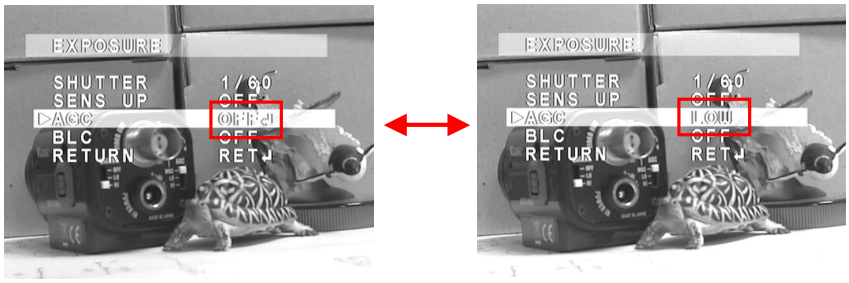
STEP1,2: OSD SETTINGS



STEP3: MANUAL GAIN SETTING

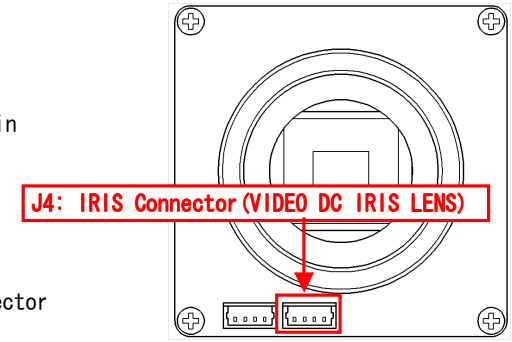


STEP4: IRIS ADJUSTMENT



STEP5: When switching to AGC LOW from OFF, and the video screen is to be slightly brighter, it is the proper result of lens iris adjustment.

4.4 VIDEO IRIS LENS CONTROL

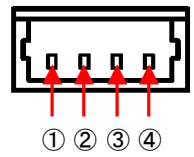
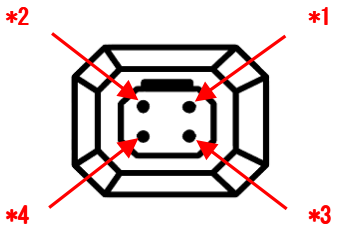


This connector is used to connect a VIDEO IRIS LENS.
When connecting the lens, be careful to wire after confirming the pin assignments.

(1) Wiring to the lens

Pin assignments of JEITA recommended iris connector and J4 connector terminal have one-to-one correspondence as following.

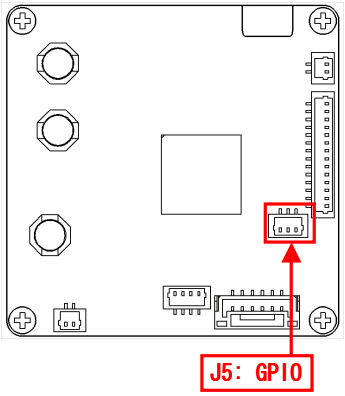
(Fig. 4-6) Pin configuration of VIDEO IRIS connector

	Parts Shape(Pin No.)		Pin Name	female IRIS connector(JEITA spec.)	
J4	 ① ② ③ ④	①	Vcc(DC+12V) OUT	*1	
		②	NC(Non Conection)	*2	
		③	IRIS OUT	*3	
		④	GND(COMMON)	*4	

(2) Adjustment method of the lens iris

How to adjust the lens iris is the same as the DC-iris lens, but location and function of the adjustment VR is different by each lens used.
About the adjustment VR(location and function), please refer to the instruction manual of the lens.

4.5 GPIO(MOTION DETECT)

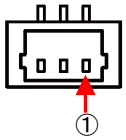


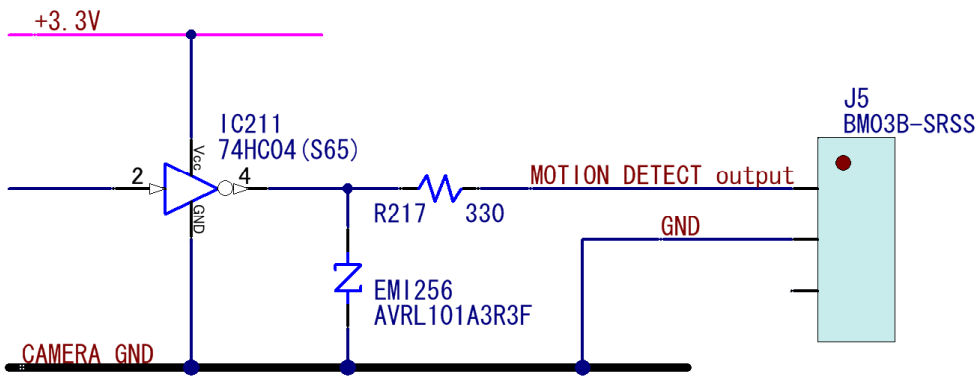
When using MOTION DETECTION function of WAT-910BD,
it is possible to output the detection results from GPIO connector.
For information about how to set up and use the motion detection feature, please refer to the OSD manual.

OSD SETTING

Motion detecting message

(Fig. 4-7) Pin configuration of GPIO(MOTION DETECT) connector

	Parts Shape(Pin No.)		Pin Name	Description
J5	 ①	①	MOTION DETECT output	When detecting a motion, output voltage is changed to H level.
		②	GND (common)	CAMERA GND(common)
		③	(reserved)	

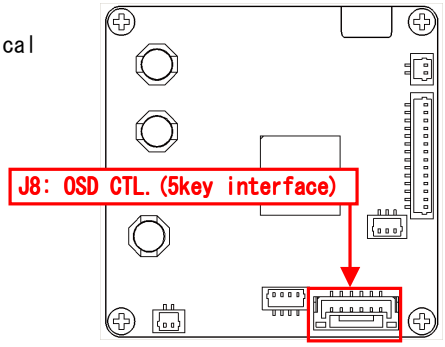


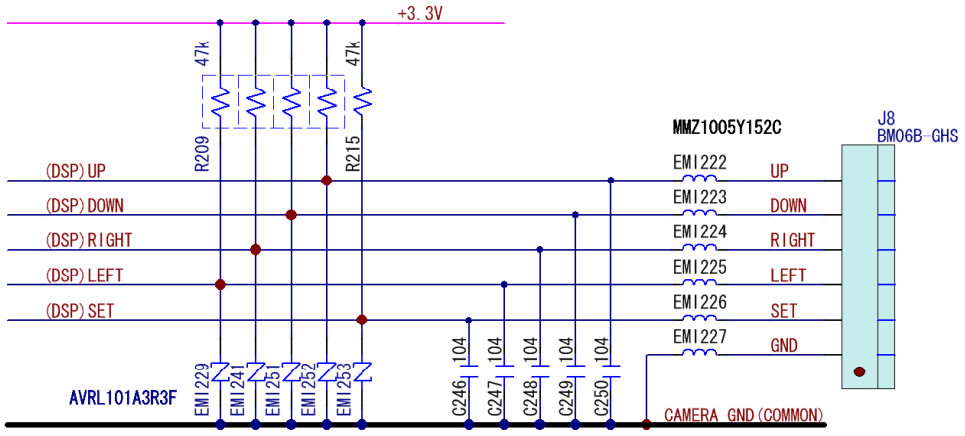
(Fig.4-8) Electrical Characteristic Table (MOTION DETECT output)

Parameter	Symbol	Min	Typ	Max	Unit	Remarks
Output High Voltage	Voh	2.4			V	Ioh = 1mA
Output Low Voltage	Vol			0.4	V	Iol = -1mA
Output Time	T		2		S	

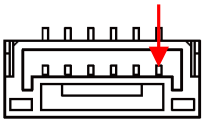
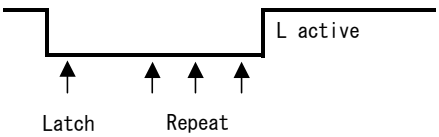
4.6 OSD CONTROL

In order to operate the OSD, either a remote controller (RC-01) or electrical contacts(circuit) is required.
The terminals of the J8 are pulled up with resistors inside of the camera.
To move cursor and/or select a item on OSD, input the L pulse in each terminals by touching with the GND terminal.





(Fig.4-9) Connector Pin Function

	Parts Shape(Pin No.)		Pin Name	Description
J8		①	GND(COMMON)	
		②	SET	
		③	LEFT	
		④	RIGHT	
		⑤	DOWN	
		⑥	UP	

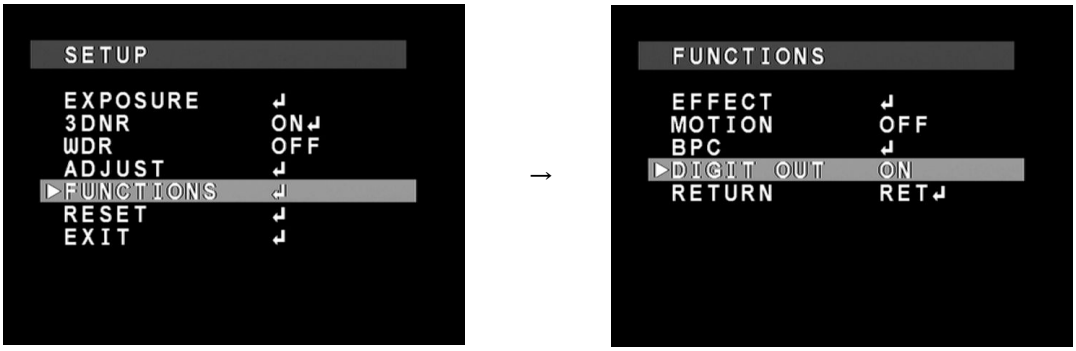
Repeat input:
While each pin is held in the L state, DSP will repeat the latch operation.

5. DIGITAL VIDEO OUT

Digital Video data can be output via the Digital Video Output Connector (J6) on the WAT-910BD.
Use the OSD Menu or the SPI communication to switch the Digital Video Data Output ON/OFF.

(1) Operation by OSD

From the [SETUP] menu, select [FUNCTIONS], from the [FUNCTIONS] menu, switch the [DIGITAL OUT] feature between ON/OFF.



For details regarding the operation of the OSD menu, please refer to the OSD Manual.

(2) Operation through SPI Communication

By using the SPI communication, it is possible to switch the Digital Video Output ON/OFF without going through the OSD menu.

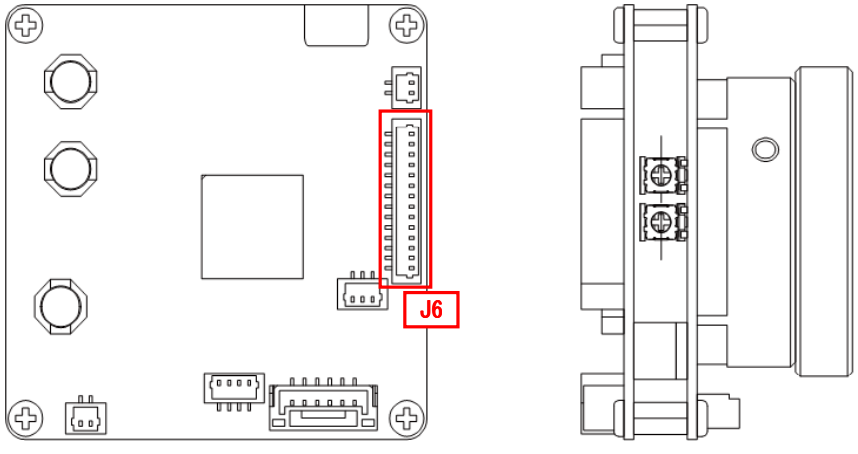
(Fig.5-1) SPI communication address (Digital out)

PARAMETER NAME	ADDRESS	DESCRIPTION
DIGIT OUT	0x5E2, bit 0	(Digital Output On/Off) 0:OFF, 1:ON

Please refer to section 6. SPI Communication on how to communicate via SPI and details on the parameter addresses.

(3)Digital Video Output Connector

The figure below shows the location of the Digital Video Output Connector (J6).

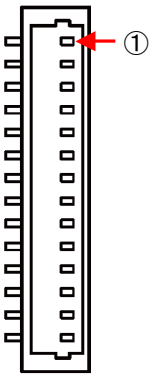


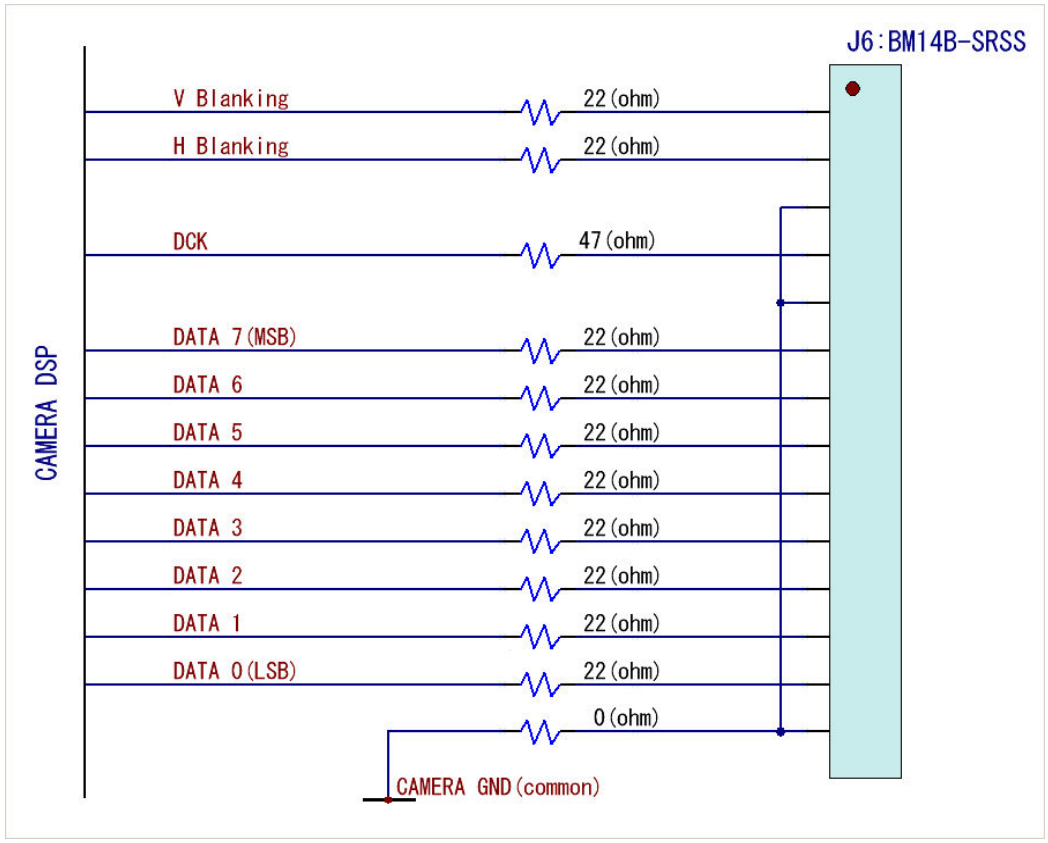
(Fig.5-2) Digital Output Connector(J6)

5.1 ELECTRICAL CHARACTERISTICS

Each pin of the J6 Connector is connected directly to the Camera's DSP through a resistor.
(3.3[V] CMOS level output. Does not go through buffer/driver.)
The pinouts for each pin and a simplified circuit diagram are shown below.

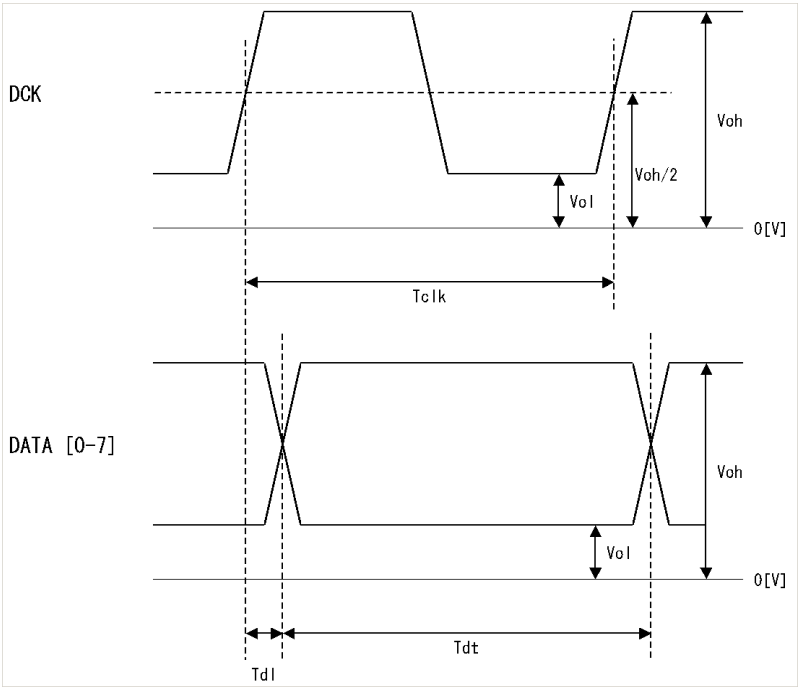
(Fig.5-3) Pin Function Descreptions

	Parts shape	Parts No.	manufacturer	Function	Pin No.	description	I/O
J6		BM14B-SRSS	JST	Digital Video Output	①	Vertical Blanking	O
					②	Horizontal Blanking	O
					③	GND (common)	—
					④	Digital Clock(DCK)	O
					⑤	GND (common)	—
					⑥	DATA 7(MSB)	O
					⑦	DATA 6	O
					⑧	DATA 5	O
					⑨	DATA 4	O
					⑩	DATA 3	O
					⑪	DATA 2	O
					⑫	DATA 1	O
					⑬	DATA 0(LSB)	O
					⑭	GND (common)	—



(Fig.5-4) partial schematic of Digital Video Output

The electrical characteristics for each pin are as described below.



(Fig.5-5) Waveform and Sybols (DCK, DATA)

(Fig.5-6) Electrical Characteristic Table (DCK, DATA)

Parameter	Symbol	Min	Typ	Max	Unit	Remarks
Output High Voltage	Voh	2.4			V	Ioh = 6mA
Output Low Voltage	Vol			0.4	V	Iol = -6mA
DCK Cycle Time(EIA)	Tclk		34.9		nS	1/(28.636363MHz)
DCK Cycle Time(CCIR)	Tclk		35.2		nS	1/(28.375MHz)
Data Output Time(EIA)	Tdt		34.9		nS	
Data Output Time(CCIR)	Tdt		35.2		nS	
Data Output Delay Time	Tdl			1.5	nS	

5.1 DIGITAL OUTPUT FORMAT

(1)Video Data Format

The WAT-910BD is a camera that conforms to the EIA/CCIR video format.
The Analog Video Output (Video Output Connector:J2) is a composite video output which conforms to the EIA/CCIR standard, however, the digital video output does not fully comply with these standards.
The video format of the digital data output is in accordance with table 5.1-7.

(Fig5-7) Digital Video Format

		EIA		CCIR	
		ODD field	EVEN field	ODD field	EVEN field
Frame rate [Frame/sec]		30		25	
Total line [Line/Frame]		525		625	
Blanking [Line/Field]		22	21	27	26
Active Video	[Line/Field]	241	241	286	286
	[Line/Frame]	482		572	
Effective Pixel [Pixel/line]		756		739	

(2)Data Format

The output digital data conforms to the ITU-R BT.656, however, the output data clock rate (DCK) frequency is as shown in the figure below.

(Fig.5-8) Digital Output Clock

	EIA	CCIR
DCK [MHz]	28.636363	28.375000

※Caution: Does not comply with the ITU-R BT.656 Standard DCK=27[MHz] output.

The Chroma Signal, Luminance Signal, Blanking Data, and TRC Data are output in sequence from the Digital Output Connector (J6).
TRC Code, Chroma Signal, and Luminance Signal data format comply with the ITU-R BT.656 standards, however, due to the difference in DCK frequency, the Blanking Data Number, Effective Data Number, and Total Data Number do not comply with ITU-R BT.656(DCK=27MHz). The Figure below shows the output data number for each.

(Fig.5-9) Digital Data Format (1 Line Period)

	EIA	CCIR	for reference (BT.656 EIA/CCIR)
EAV [Byte]	4		4/4
BLK(H) [Byte]	300	330	268/280
SAV [Byte]	4		4/4
Effective DATA [Byte]	1512	1478	1440/1440
Total Data [Byte]	1820	1808	1716/1728
Effective Data Bit [bit]	8		
Video Data level [Decimal]	1 - 254		

(3)Output Timing for Digital Data

The effective image data is outputs the Luminance Data and Color Difference Data in Y/Cb/Cr=4:2:2 format.
(The output sequence is repeated in the Cb,Y,Cr,Y,… sequence)
Since the WAT-910BD is a monochrome camera, the Cb,Cr data are all 80[h].

The Horizontal Blanking Period (without TRC Code) is Cb,Cr=80[h], Y=10[h], repeated output.

The TRC Code, which is used to set the Efective Pixel Period, is output with every horizontal line.
The TRC Code complies with the ITU-R BT.656 standard, and output the 4 Byte Codes below.

ＳＡＶ： Start of Active Video [hex]	FF	00	00	XX
ＥＡＶ： End of Active Video [hex]	FF	00	00	XX

The 4th Byte of the TRC code (XX) is output according to the Figure below.

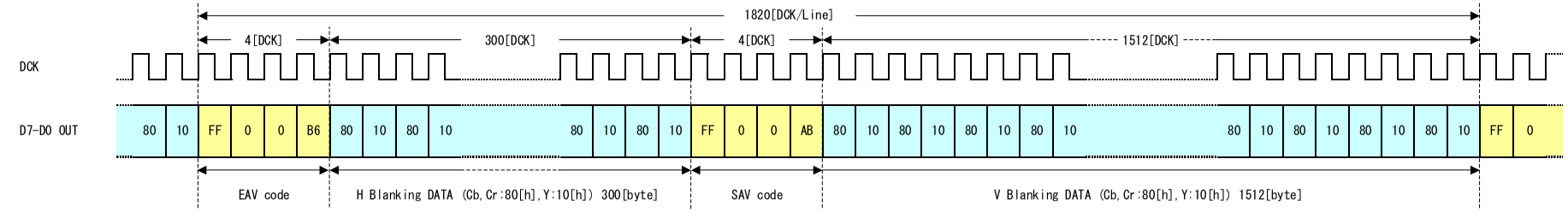
(Fig.5-10) TRC(Timing Reference Code)

	(Fix)	F	V	H	P3	P2	P1	P0	CODE
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	[hex]
ODD Field H Blanking(SAV)	1	0	0	0	0	0	0	0	80
ODD Field H Blanking(EAV)	1	0	0	1	1	1	0	1	9D
ODD Field V Blanking(SAV)	1	0	1	0	1	0	1	1	AB
ODD Field V Blanking(EAV)	1	0	1	1	0	1	1	0	B6
EVEN Field H Blanking(SAV)	1	1	0	0	0	1	1	1	C7
EVEN Field H Blanking(EAV)	1	1	0	1	1	0	1	0	DA
EVEN Field V Blanking(SAV)	1	1	1	0	1	1	0	0	EC
EVEN Field V Blanking(EAV)	1	1	1	1	0	0	0	1	F1

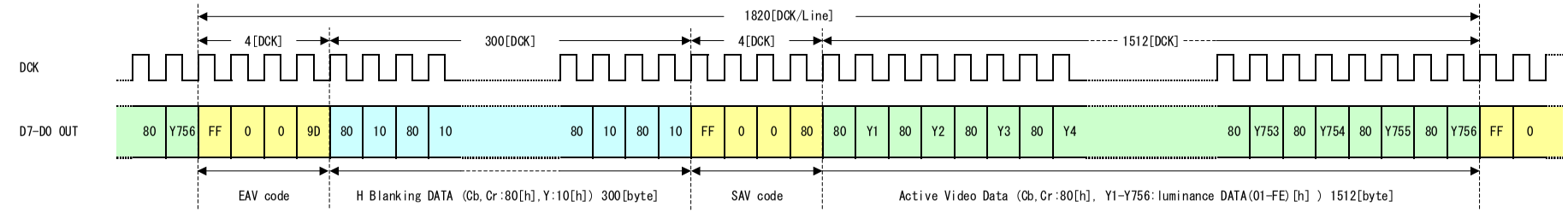
*NOTE: F=0 : during field 1 F=1 : during field 2
V=0 : elsewhere V=1 : during field blanking
H=0 : in SAV H=1 : in EAV
P0-P3 : protection bits

Fig.5-11(EIA)/Fig.5-12(CCIR) shows the output timing for 1 line worth of digital data.
Fig.5-13(EIA)/Fig.5-14(CCIR) shows the output timing for 1 frame worth of digital data.

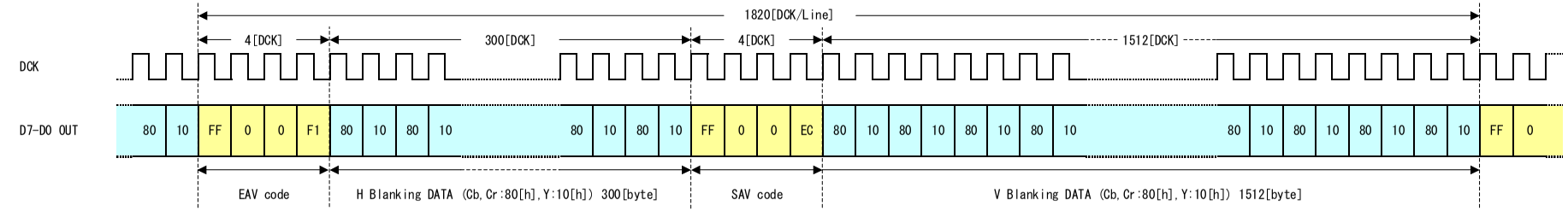
(1) EIA ODD Field Blanking line



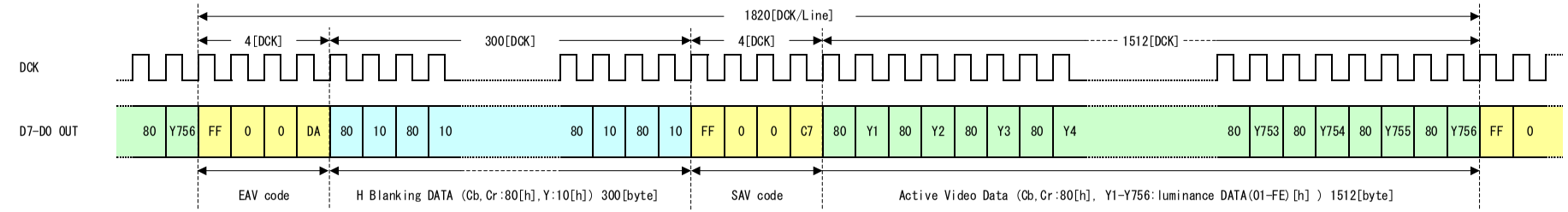
(2) EIA ODD Field Active Video line



(3) EIA EVEN Field Blanking line

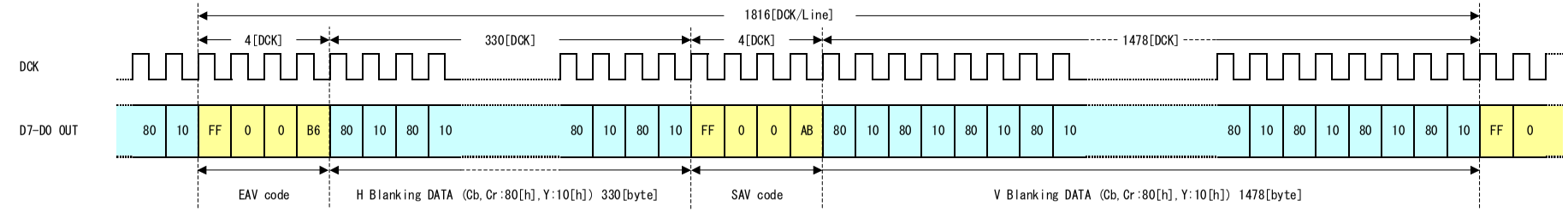


(4) EIA EVEN Field Active Video line

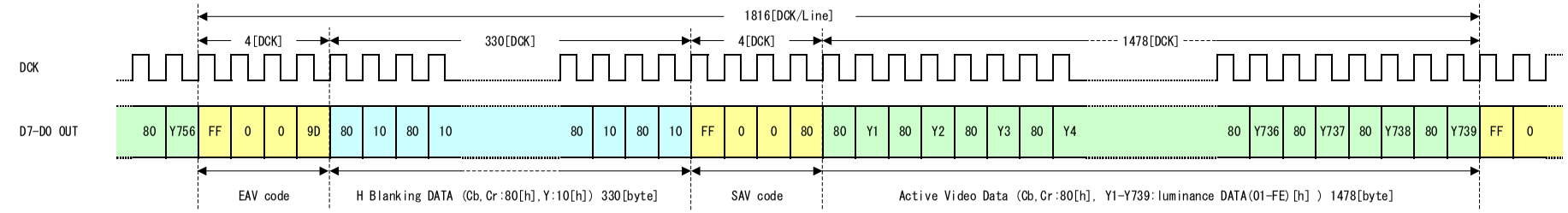


(Fig 5-11) Line Data Format (EIA)

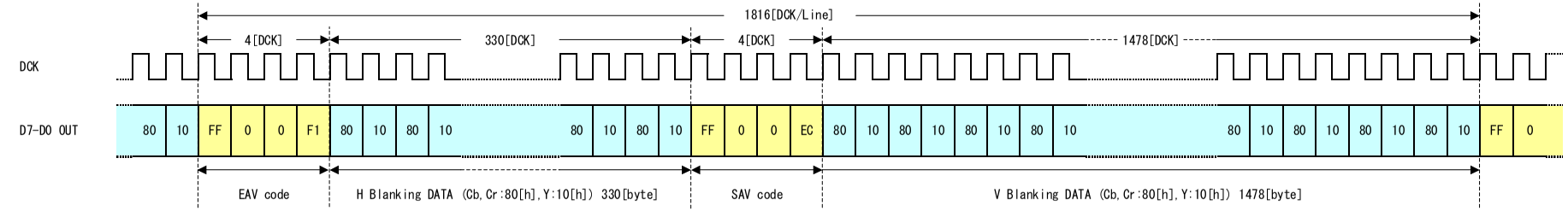
(1) CCIR ODD Field Blanking line



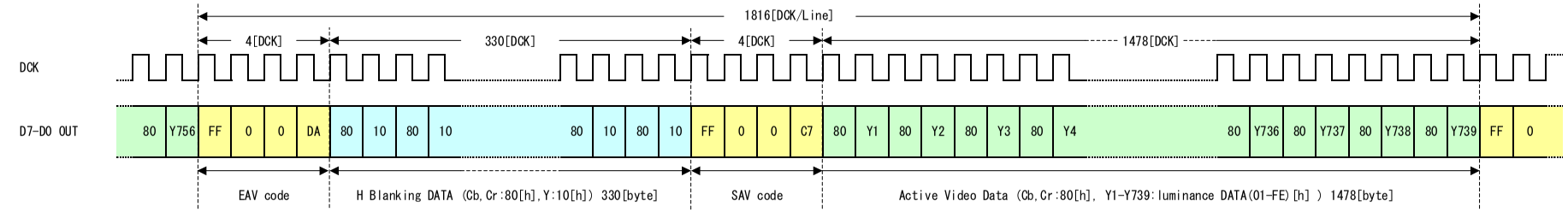
(2) CCIR ODD Field Active Video line



(3) CCIR EVEN Field Blanking line



(4) CCIR EVEN Field Active Video line



(Fig 5-12) Line Data Format (CCIR)

(Fig.5-13) Digital Data Format(EIA)

		Line#	EAV(4byte)				H Blanking Data(300byte)									SAV(4byte)				V Blanking / Active Video Data(1512byte)																
FLD 1	V BLK	4	FF	0	0	B6	80	10	80	10	→	80	10	80	10	FF	0	0	AB	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		5	FF	0	0	B6	80	10	80	10	→	80	10	80	10	FF	0	0	AB	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		↓	↓				↓ (H Blanking Data)									↓				↓ (V Blanking Data)																
		21	FF	0	0	B6	80	10	80	10	→	80	10	80	10	FF	0	0	AB	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		22	FF	0	0	B6	80	10	80	10	→	80	10	80	10	FF	0	0	AB	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
	VIDEO	23	FF	0	0	9D	80	10	80	10	→	80	10	80	10	FF	0	0	80	80	Y1	80	Y2	80	Y3	80	Y4		80	Y753	80	Y754	80	Y755	80	Y756
		24	FF	0	0	9D	80	10	80	10	→	80	10	80	10	FF	0	0	80	80	Y1	80	Y2	80	Y3	80	Y4		80	Y753	80	Y754	80	Y755	80	Y756
		↓	↓				↓ (H Blanking Data)									↓				↓ (Active Video Data)																
		262	FF	0	0	9D	80	10	80	10		80	10	80	10	FF	0	0	80	80	Y1	80	Y2	80	Y3	80	Y4		80	Y753	80	Y754	80	Y755	80	Y756
		263	FF	0	0	9D	80	10	80	10		80	10	80	10	FF	0	0	80	80	Y1	80	Y2	80	Y3	80	Y4		80	Y753	80	Y754	80	Y755	80	Y756
		264	FF	0	0	B6	80	10	80	10	→	80	10	80	10	FF	0	0	AB	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		265	FF	0	0	B6	80	10	80	10	→	80	10	80	10	FF	0	0	AB	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		266	FF	0	0	B6	80	10	80	10	→	80	10	80	10	FF	0	0	AB	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
FLD2	V BLK	267	FF	0	0	F1	80	10	80	10	→	80	10	80	10	FF	0	0	EC	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		268	FF	0	0	F1	80	10	80	10	→	80	10	80	10	FF	0	0	EC	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		↓	↓				↓ (H Blanking Data)									↓				↓ (V Blanking Data)																
		283	FF	0	0	F1	80	10	80	10	→	80	10	80	10	FF	0	0	EC	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		284	FF	0	0	F1	80	10	80	10	→	80	10	80	10	FF	0	0	EC	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
	VIDEO	285	FF	0	0	DA	80	10	80	10	→	80	10	80	10	FF	0	0	C7	80	Y1	80	Y2	80	Y3	80	Y4		80	Y753	80	Y754	80	Y755	80	Y756
		286	FF	0	0	DA	80	10	80	10	→	80	10	80	10	FF	0	0	C7	80	Y1	80	Y2	80	Y3	80	Y4		80	Y753	80	Y754	80	Y755	80	Y756
		↓	↓				↓ (H Blanking Data)									↓				↓ (Active Video Data)																
		524	FF	0	0	DA	80	10	80	10	→	80	10	80	10	FF	0	0	C7	80	Y1	80	Y2	80	Y3	80	Y4		80	Y753	80	Y754	80	Y755	80	Y756
		525	FF	0	0	DA	80	10	80	10	→	80	10	80	10	FF	0	0	C7	80	Y1	80	Y2	80	Y3	80	Y4		80	Y753	80	Y754	80	Y755	80	Y756
	V BLK	1	FF	0	0	F1	80	10	80	10	→	80	10	80	10	FF	0	0	EC	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		2	FF	0	0	F1	80	10	80	10	→	80	10	80	10	FF	0	0	EC	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		3	FF	0	0	F1	80	10	80	10	→	80	10	80	10	FF	0	0	EC	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		DCK#	1513	1514	1515	1516	1517	1518	1519	1520	→	1813	1814	1815	1816	1817	1818	1819	1820	1	2	3	4	5	6	7	8	→	1505	1506	1507	1508	1509	1510	1511	1512
		Pixel#	757		758		759		760		→	907		908		909		910		Y1		Y2		Y3		Y4		→	Y753		Y754		Y755		Y756	

(Fig.5.1-13) Digital Data Format(EIA)

(Fig.5-14) Digital Data Format(CCIR)

		Line#	EAV(4byte)				H Blanking Data(330byte)									SAV(4byte)				V Blanking / Active Video Data(1478byte)																
FLD 1	V BLK	4	FF	0	0	B6	80	10	80	10	→	80	10	80	10	FF	0	0	AB	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		5	FF	0	0	B6	80	10	80	10	→	80	10	80	10	FF	0	0	AB	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		↓	↓				↓ (H Blanking Data)									↓				↓ (V Blanking Data)																
		26	FF	0	0	B6	80	10	80	10	→	80	10	80	10	FF	0	0	AB	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		27	FF	0	0	B6	80	10	80	10	→	80	10	80	10	FF	0	0	AB	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
	VIDEO	28	FF	0	0	9D	80	10	80	10	→	80	10	80	10	FF	0	0	80	80	Y1	80	Y2	80	Y3	80	Y4		80	Y736	80	Y737	80	Y738	80	Y739
		24	FF	0	0	9D	80	10	80	10	→	80	10	80	10	FF	0	0	80	80	Y1	80	Y2	80	Y3	80	Y4		80	Y736	80	Y737	80	Y738	80	Y739
		↓	↓				↓ (H Blanking Data)									↓				↓ (Active Video Data)																
		312	FF	0	0	9D	80	10	80	10		80	10	80	10	FF	0	0	80	80	Y1	80	Y2	80	Y3	80	Y4		80	Y736	80	Y737	80	Y738	80	Y739
		313	FF	0	0	9D	80	10	80	10		80	10	80	10	FF	0	0	80	80	Y1	80	Y2	80	Y3	80	Y4		80	Y736	80	Y737	80	Y738	80	Y739
	V BLK	314	FF	0	0	B6	80	10	80	10	→	80	10	80	10	FF	0	0	AB	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		315	FF	0	0	B6	80	10	80	10	→	80	10	80	10	FF	0	0	AB	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		316	FF	0	0	B6	80	10	80	10	→	80	10	80	10	FF	0	0	AB	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
FLD2		317	FF	0	0	F1	80	10	80	10	→	80	10	80	10	FF	0	0	EC	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		318	FF	0	0	F1	80	10	80	10	→	80	10	80	10	FF	0	0	EC	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		↓	↓				↓ (H Blanking Data)									↓				↓ (V Blanking Data)																
		338	FF	0	0	F1	80	10	80	10	→	80	10	80	10	FF	0	0	EC	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		339	FF	0	0	F1	80	10	80	10	→	80	10	80	10	FF	0	0	EC	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
	VIDEO	340	FF	0	0	DA	80	10	80	10	→	80	10	80	10	FF	0	0	C7	80	Y1	80	Y2	80	Y3	80	Y4		80	Y736	80	Y737	80	Y738	80	Y739
		341	FF	0	0	DA	80	10	80	10	→	80	10	80	10	FF	0	0	C7	80	Y1	80	Y2	80	Y3	80	Y4		80	Y736	80	Y737	80	Y738	80	Y739
		↓	↓				↓ (H Blanking Data)									↓				↓ (Active Video Data)																
		624	FF	0	0	DA	80	10	80	10	→	80	10	80	10	FF	0	0	C7	80	Y1	80	Y2	80	Y3	80	Y4		80	Y736	80	Y737	80	Y738	80	Y739
		625	FF	0	0	DA	80	10	80	10	→	80	10	80	10	FF	0	0	C7	80	Y1	80	Y2	80	Y3	80	Y4		80	Y736	80	Y737	80	Y738	80	Y739
	V BLK	1	FF	0	0	F1	80	10	80	10	→	80	10	80	10	FF	0	0	EC	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		2	FF	0	0	F1	80	10	80	10	→	80	10	80	10	FF	0	0	EC	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		3	FF	0	0	F1	80	10	80	10	→	80	10	80	10	FF	0	0	EC	80	10	80	10	80	10	80	10		80	10	80	10	80	10	80	10
		DCK#	1479	1480	1481	1482	1483	1484	1485	1486	→	1809	1810	1811	1812	1813	1814	1815	1816	1	2	3	4	5	6	7	8	→	1471	1472	1473	1474	1475	1476	1477	1478
		Pixel#	740		741		742		743		→	905		906		907		908		Y1		Y2		Y3		Y4		→	Y736		Y737		Y738		Y739	

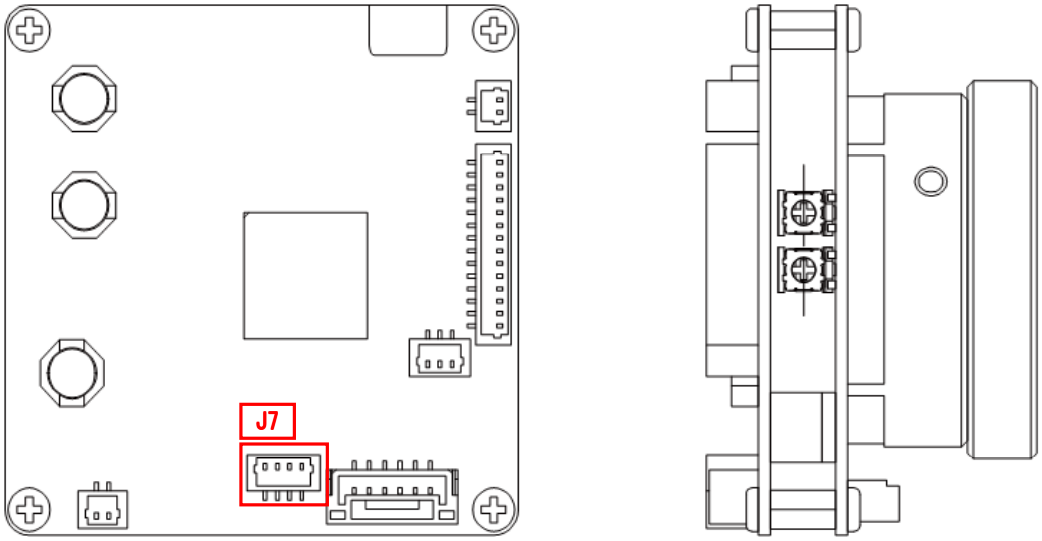
(Fig.5.1-14) Digital Data Format(CCIR)

6. SPI COMMUNICATION

By using the SPI communication connector (J7), various functions of WAT-910BD can be controlled directly, without using OSD.

(1) SPI communication connector

Refer to the following figure for the position of SPI communication connector (J7).

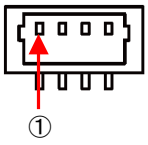


(Fig. 6-1) SPI Communication Connector(J7)

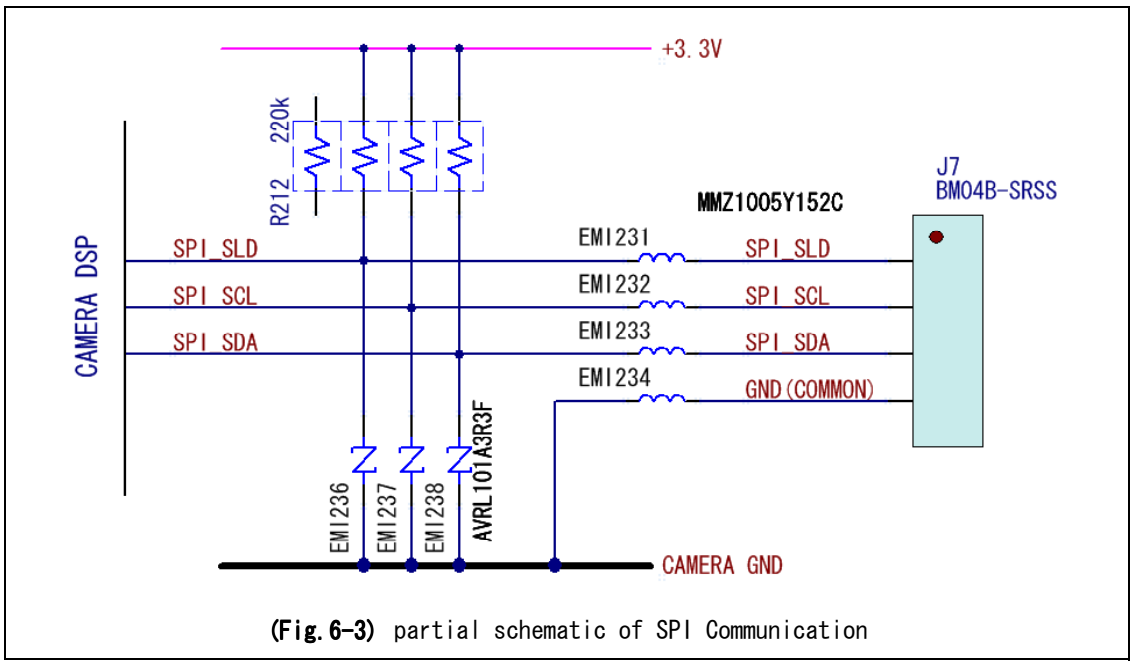
6.1 ELECTRICAL CHARACTERISTICS

The pin arrangement and the outline circuit diagram of each pin are shown below.

(Fig.6-2) Pin Function and Descriptions

	Parts Shape	Name	Parts No.	manufacturer	Pin No.	Description	I/O
J7		SPI Communication	BM04B-SRSS	JST	①	SPI SLD (active L)	I
					②	SPI SCL	I
					③	SPI SDA	I/O
					④	GND (common)	GND

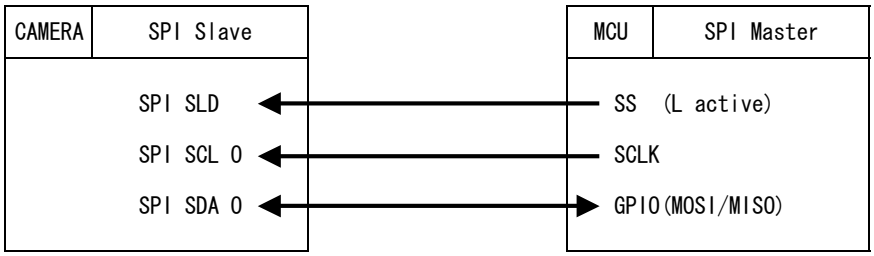
All pins of J7 connector are directly wired to Camera DSP through the bead.
(EMI231-234, inductor for high frequency.3.3[V], CMOS logic input/output. There are no buffers/drivers.)
And the varistor elements are connected between each terminal and camera GND as countermeasure for the electrostatic discharge and/or surge damage.
Each pin is connected to 3.3V(internal camera circuit) through resistors (220k ohm).



(Fig.6-3) partial schematic of SPI Communication

(1) Preparation

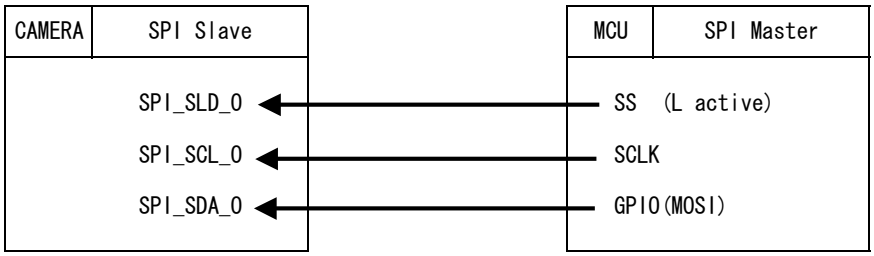
Though the communication specification of WAT-910BD applies to SPI bus specification fundamentally, it is implemented as 3 wire type.
Connection of the communication port of external MCU (master) and a camera (slave) is made as follows.



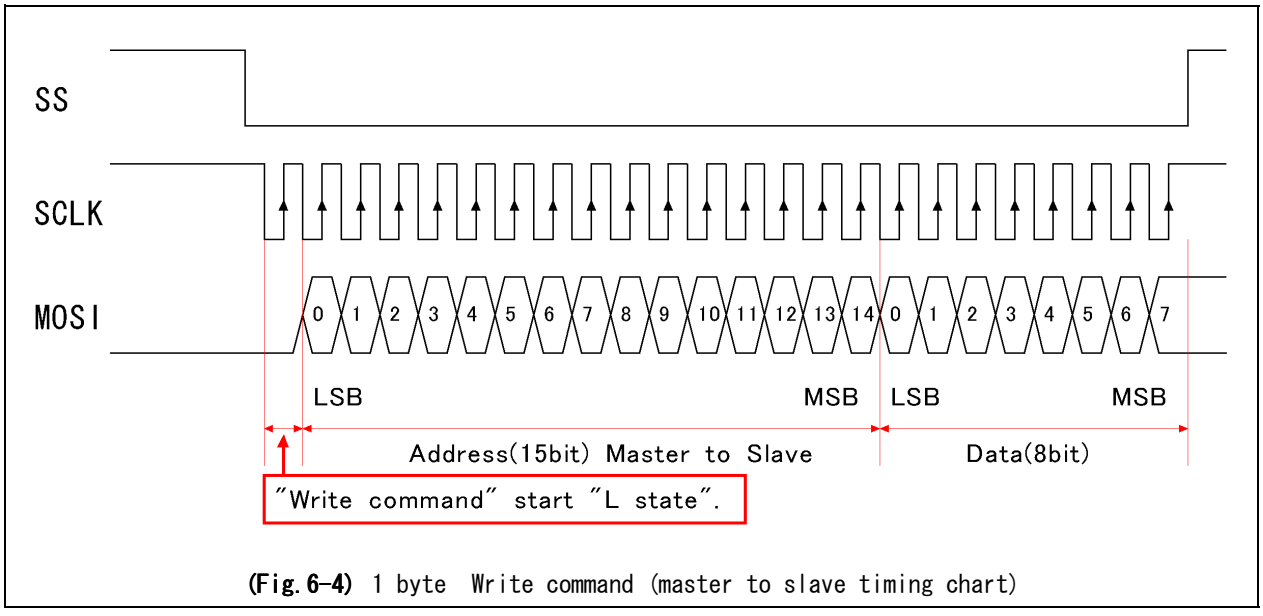
Both directions (input and output) are required for GPIO of external MCU (master), and in order to do bidirectional communication (MISO/MOSI), it needs to be switched direction (input or output).

(2) Command and Timing Chart(SPI Write Command)

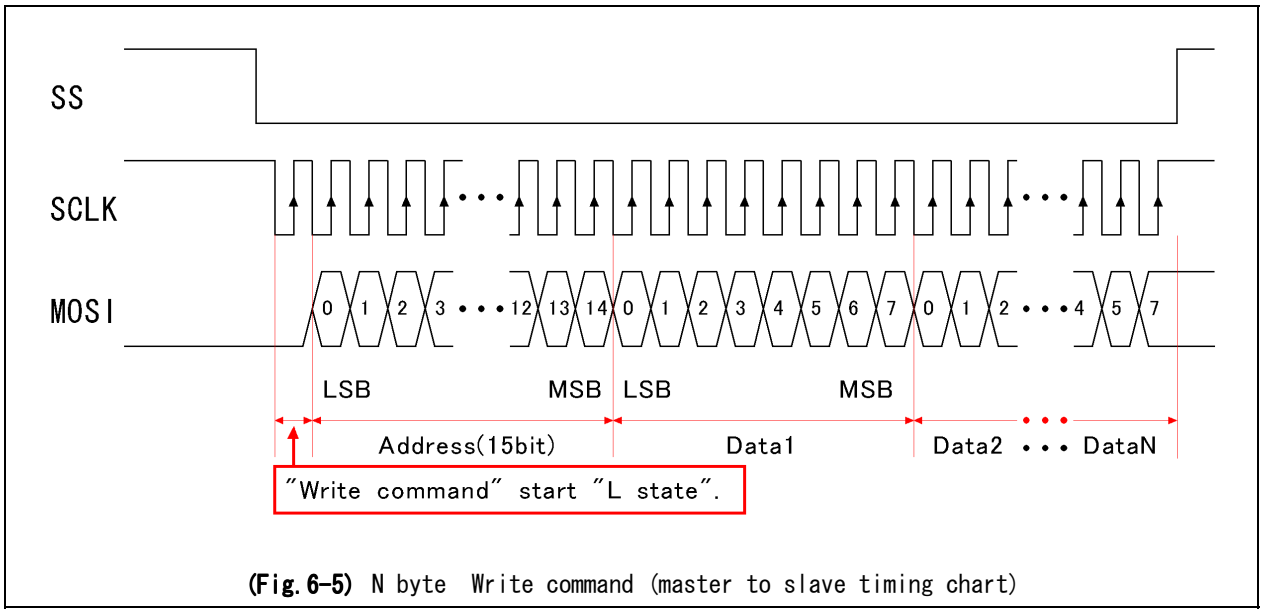
All the Write commands are direction from the external MCU to the camera DSP (master to slave).
So, GPIO of external MCU is used as MOSI and a command is outputted to camera SPI SDA as it is.
(GPIO of external MCU is always used as output port.)



H/L level of MISO/MOSI changes are synchronize with SCLK falling edge, and Data latch is done synchronizing with the rising up edge of SCLK.
This operation is the same about both directions(master to slave and slave to master communication).
The timing charts of Write command (master to slave only) are shown with following.



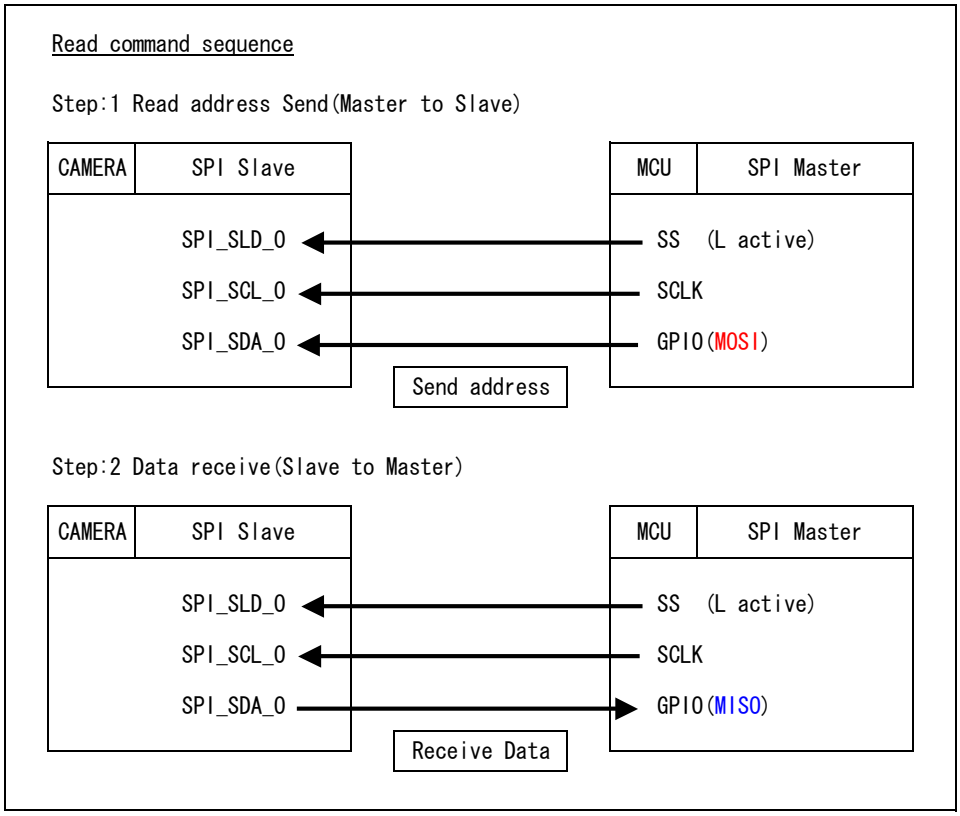
(Fig. 6-4) 1 byte Write command (master to slave timing chart)



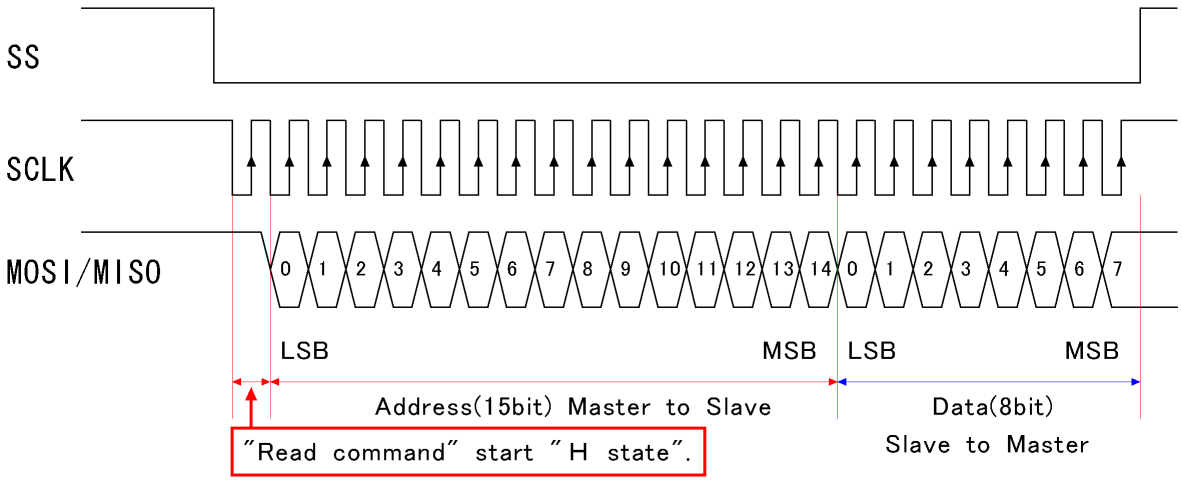
(Fig.6-5) N byte Write command (master to slave timing chart)

(3) Command and Timing Chart(SPI Read Command)

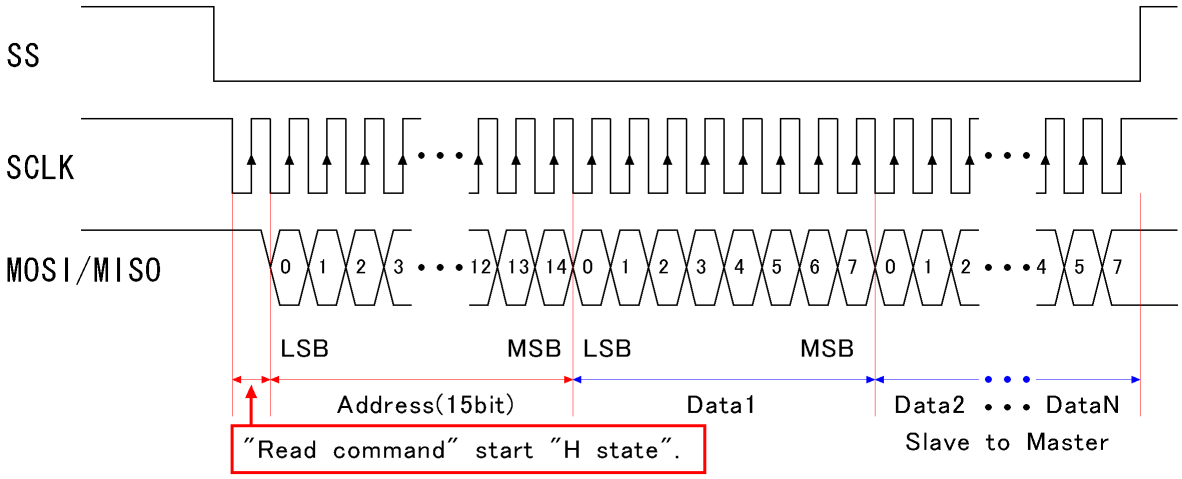
In the case of Read command, after sending the address of the needed data from external MCU to Camera DSP (master -> slave), it is outputted to external MCU from Camera DSP (slave -> master).
So, GPIO of external MCU is used as MOSI at the time of address sending, and it is used as MISO at the time of data receive.
(At first, when sending address period, GPIO of external MCU is used as output port, then it is switched to input port.)



The timing charts of Read command (after send command, it will be returned response to master from slave) is shown with following.

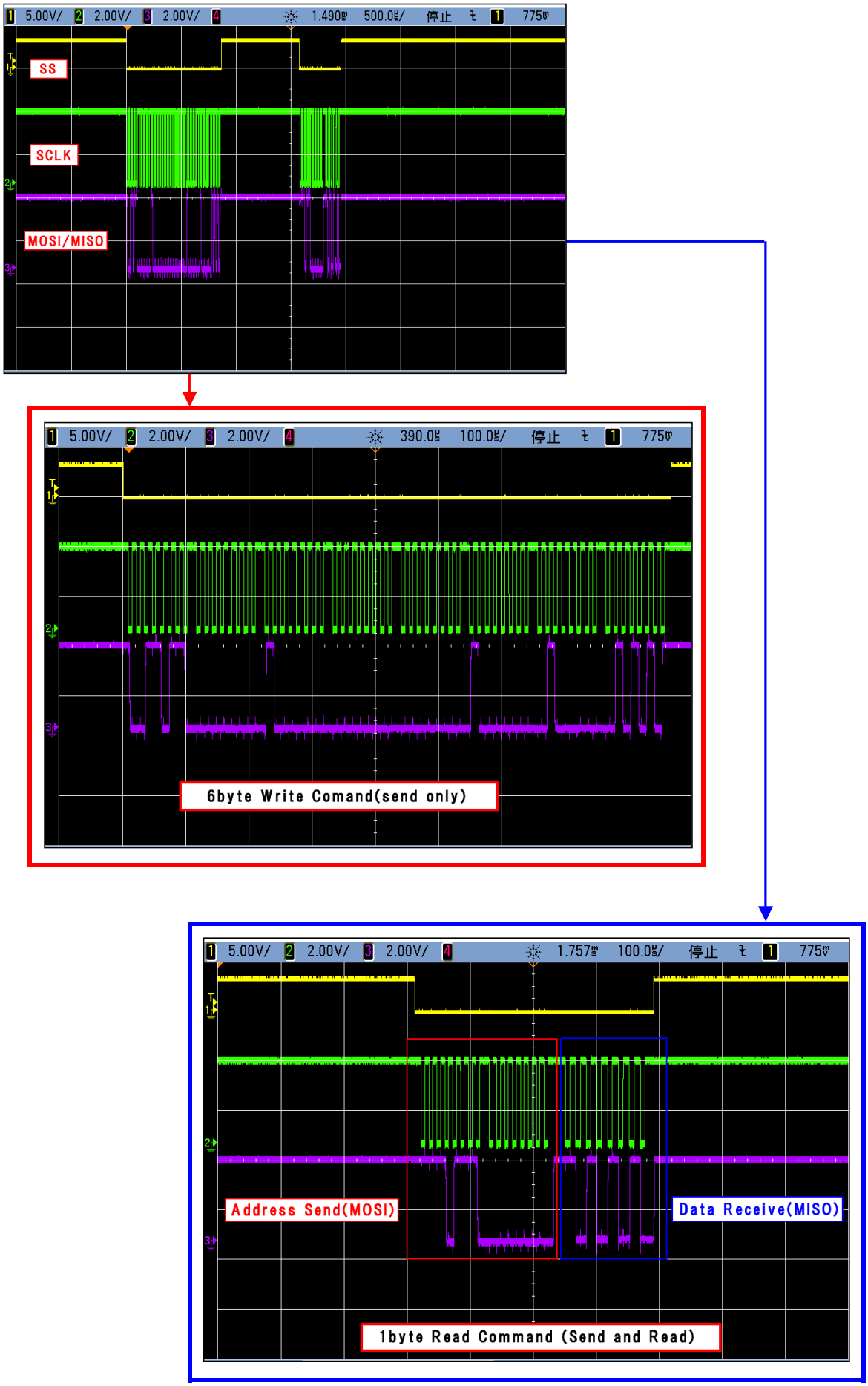


(Fig. 6-6) 1 byte Read command (master to slave and slave to master timing chart)



(Fig. 6-7) N byte Read command (master to slave and slave to master timing chart)

(4) Actual wave form of "SPI Read Command"



6.2 CTL-COMMAND PROTOCOL

Camera CTL-COMMAND is executed by inside MCU on camera DSP.
"CTL-COMMAND processing" is doing by "indirect parameter accessing".
Camera MCU check the indirect registers to fetch CTL-COMMAND, and execute it immediately.
Thus, to control camera and/or to read/write camera parameter, external MCU should send command, address, parameter value, check-sum and status CODE to indirect registers by using SPI write method.
Indirect registers to use SPI communications are shown in following table.

(Fig.6-8) Indirect Registers for SPI Control

No.	Symbol	Address	Description
1	C1	0x0036	Command1
2	C2	0x0037	Command2
3	ADR	0x0038	Parameter Address
4	DAT	0x0039	Parameter Data
5	CS	0x003A	Check sum (C1+C2+ADR+DAT)
6	ST	0x003B	status CODE

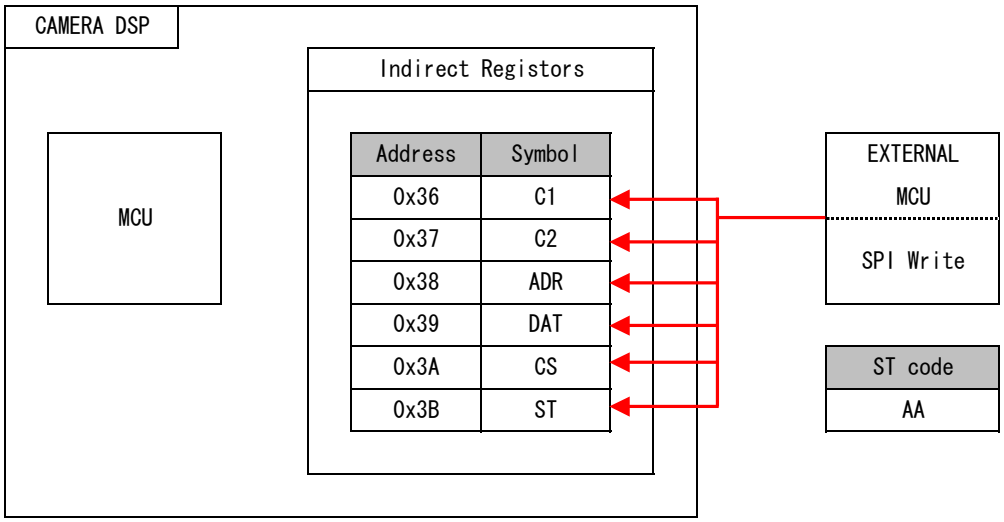
The value of ST(0x3B) register shows the status of the end of normal execution / error of CTL-COMMAND, and turns into one of the following values.

(Fig.5.2-9) ST register CODE

No.	CODE	Description
1	AA	DO COMMAND (This status code is written by EXT.MCU only.)
2	55	EXEC.NORMALLY (This status code is written by CAM MCU only.)
3	A5	EXEC.ERROR (This status code is written by CAM MCU only.)

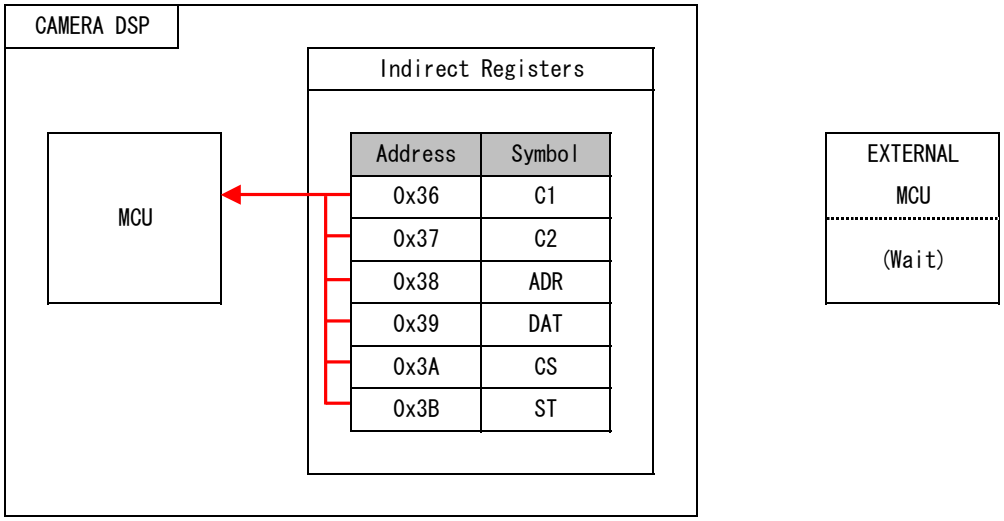
Execution of camera CTL-COMMAND is specifically expressed at the following steps.

Step.1: CTL-COMMAND SEND(EXT.MCU -> CAM DSP)



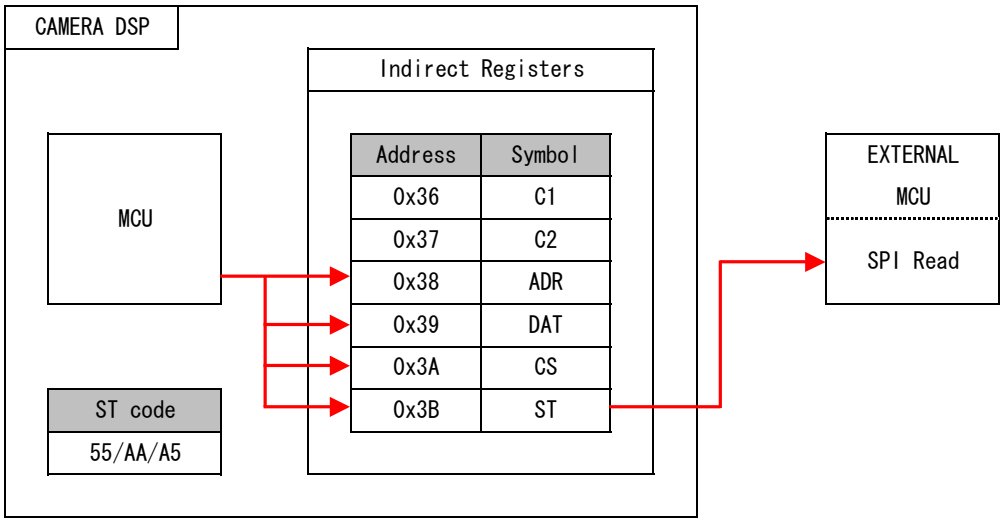
Set CTL-COMMAND to indirect registers from external MCU by SPI write command.

Step.2: COMMAND LATCH and EXECUTION(CAM MCU)



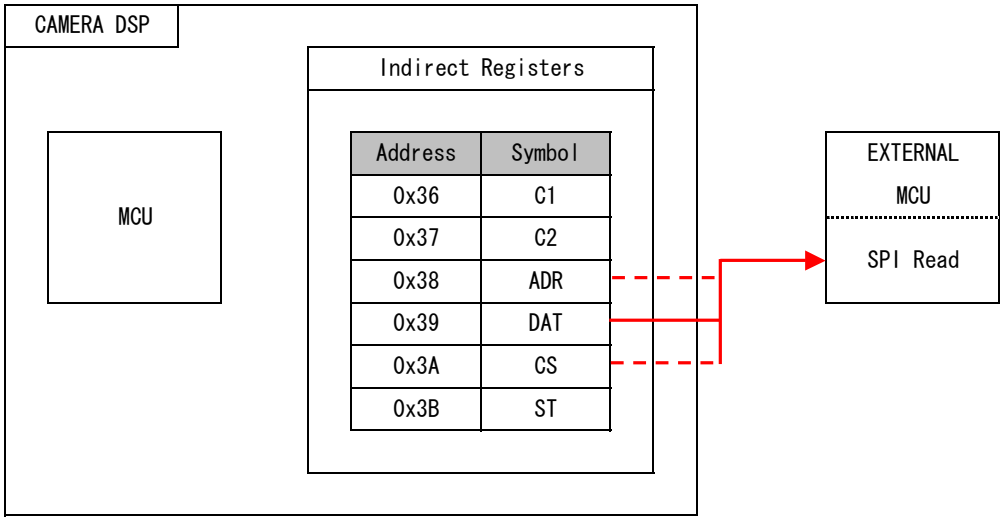
CAM MCU capture CTL-COMMAND from indirect registers and execute them.

Step.3: CHECK ST CODE



After command execution, camera MCU set execution results in indirect registers.
For checking the end of CTL-COMMAN execution , external MCU should do SPI read and check data in ST(0x3B) register.
If ST code is 55, it means "Command Execution Normally END" of CTL-COMMAND.
In this case, CAM MCU response is set ADR/DAT/CS(0x38/0x39/0x3A) registers.
If ST code is A5, it means "Command Execution ERROR" of CTL-COMMAND.
In this case, we should check command format(correctness of ADR/DATA/CS data), and send correct command to indirect registers.
If ST code is stay AA, CTL-COMMAND execution is in execution.

Step.4: RESPONSE READ



After checking ST (CODE:55) by using SPI Read command, get DAT(0x39) by using SPI read command.
This is a basic step for CTL-COMMAND(camera parameter read).
If you need fully secure, it need to check indirect register value from ADR to CS (from 0x38 to 0x3A).
In this case, you get CS value which equal ADR+DAT value.

6.3 CAMERA CTL-COMMAND

"Camera CTL-COMMAND" are shown in below.

(1) CAMERA PARAMETER READ1(parameter address: 0x400 - 0x4FF)

SPI Write "data set"

Symbol	C1	C2	ADR	DAT	CS	ST
Value	00	00	*1	00	*2	AA

*1: Parameter address lower byte. Assignable value range is 00 to FF.
*2: Check sum (lower byte of "C1 + C2 + ADR + DAT)
*In READ command, DAT value is always set 00.

SPI Read "data set"(camera MCU response)

Symbol	ADR	DAT	CS	ST
Value	*1	*2	*3	*4

*1: Parameter address lower byte. Same as SPI Write data.
*2: Paramwter value
*3: Check sum (lower byte of "ADR + DAT)
*4: ST CODE(AA:in execution 55:EXEC.NORMALLY A5:EXEC.ERROR)

(2) CAMERA PARAMETER READ2(parameter address: 0x500 - 0x5FF)

SPI Write "data set"

Symbol	C1	C2	ADR	DAT	CS	ST
Value	00	01	*1	00	*2	AA

*1: Parameter address lower byte. Assignable value range is 00 to FF.
*2: Check sum (lower byte of "C1 + C2 + ADR + DAT")
*In READ command, DAT value is always set 00.

SPI Read "data set"(camera MCU response)

Symbol	ADR	DAT	CS	ST
Value	*1	*2	*3	*4

*1: Parameter address lower byte. Same as SPI Write data.
*2: Paramwter value
*3: Check sum (lower byte of "ADR + DAT")
*4: ST CODE(AA:in execution 55:EXEC.NORMALLY A5:EXEC.ERROR)

(3) CAMERA PARAMETER WRITE1(address: 0x400 – 0x4FF)

SPI Write "data set"

Symbol	C1	C2	ADR	DAT	CS	ST
Value	00	80	*1	*2	*3	AA

*1: Parameter address lower byte. Assignable value range is 00 to FF.
*3: Parameter Value.
*2: Check sum (lower byte of "C1 + C2 + ADR + DAT")

SPI Read "data set"(camera MCU response)

Symbol	ADR	DAT	CS	ST
Value	*1	*2	*3	*4

*1: Parameter address lower byte.(Same as SPI Write data.)
*2: Parameter value.(Same as SPI Write data.)
*3: Check sum (lower byte of "ADR + DAT")
*4: ST CODE(AA:in execution 55:EXEC.NORMALLY A5:EXEC.ERROR)

(4) CAMERA PARAMETER WRITE2(address: 0x500 – 0x5FF)

SPI Write "data set"

Symbol	C1	C2	ADR	DAT	CS	ST
Value	00	81	*1	*2	*3	AA

*1: Parameter address lower byte. Assignable value range is 00 to FF.
*2: Check sum (lower byte of "C1 + C2 + ADR + DAT)
*In READ command, DAT value is always set 00.

SPI Read "data set"(camera MCU response)

Symbol	ADR	DAT	CS	ST
Value	*1	*2	*3	*4

*1: Parameter address lower byte.(Same as SPI Write data.)
*2: Parameter value.(Same as SPI Write data.)
*3: Check sum (lower byte of "ADR + DAT")
*4: ST CODE(AA:in execution 55:EXEC.NORMALLY A5:EXEC.ERROR)

(5) AREA DISPLAY ON/OFF

SPI Write "data set"

Symbol	C1	C2	ADR	DAT	CS	ST
Value	01	*1	00	*2	*3	AA

*1: WINDOW SELECT(BLC = 0x00, HSBLC = 0x01)

*2: AREA DISPLAY ON: 0x01 / AREA DISPLAY OFF: 0x00

*3: Check sum (lower byte of "C1 + C2 + ADR + DAT")

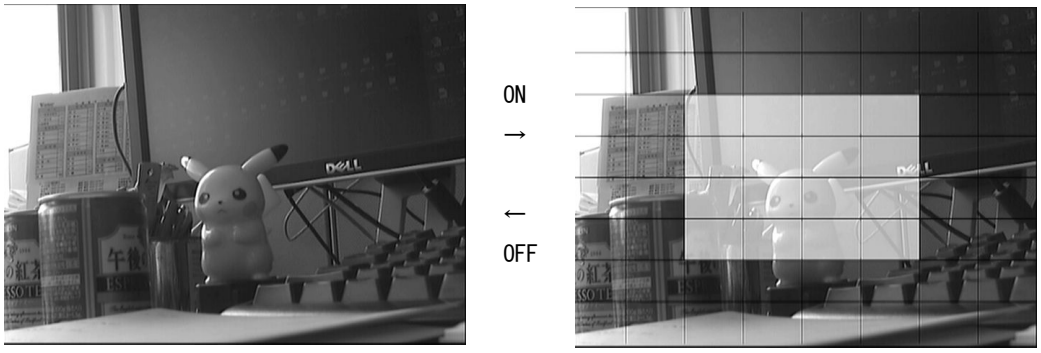
SPI Read "data set"(camera MCU response)

Symbol	ADR	DAT	CS	ST
Value	00	*4	*5	*6

*4: Parameter value.(Same as SPI Write data.)

*5: Check sum (lower byte of "ADR + DAT")

*6: ST CODE(AA:in execution 55:EXEC.NORMALLY A5:EXEC.ERROR)



(6) CAMERA OSD CONTROL(5 KEY OPERATION)

SPI Write "data set"

Symbol	C1	C2	ADR	DAT	CS	ST
Value	02	00	00	*1	*2	AA

*1: 5 key direction(See following table)

*2: Check sum (lower byte of "C1 + C2 + ADR + DAT)

5 key direction	DAT	CS
LEFT	01	03
RIGHT	02	04
UP	03	05
DOWN	04	06
SET	05	07

SPI Read "data set"(camera MCU response)

Symbol	ADR	DAT	CS	ST
Value	00	*3	*4	*5

*3: Parameter value.(Same as SPI Write data.)

*4: Check sum (lower byte of "ADR + DAT")

*5: ST CODE(AA:in execution 55:EXEC.NORMALLY A5:EXEC.ERROR)

(6) CAMERA PARAMETER SAVE

SPI Write "data set"

Symbol	C1	C2	ADR	DAT	CS	ST
Value	01	FF	00	00	00	AA

SPI Read "data set"(camera MCU response)

Symbol	ADR	DAT	CS	ST
Value	00	00	00	*1

*1: ST CODE(AA:in execution 55:EXEC.NORMALLY A5:EXEC.ERROR)

6.4 CAMERA PARAMETER

Please do not change the bit or the byte data with the notation of "-" among following tables.

(Camera operation will not be guaranteed.)

Address	Bit							
	7	6	5	4	3	2	1	0
0x400	-	-	-	-	-	-	-	-
0x401	-	-	-	-	AGC MODE 0x00:OFF 0x01:LOW 0x02:MID 0x03:HI		EI 0:OFF 1:ON	-
0x402	-	-	-	FIX SHUTTER SPEED 0x00:x256, 0x01:x128, 0x02:x64, 0x03:x32, 0x04:x16, 0x05:x8, 0x06:x4, 0x07:x2, 0x08:EI, 0x09:OFF(E:1/60,C:1/50), 0x0A:FL, 0x0B:1/250, 0x0C:1/500, 0x0D:1/1000, 0x0E:1/2000, 0x0F:1/5000, 0x10:1/10000, 0x11:1/100000				
0x403	-	-	-	-	-	-	-	-
0x404	-	-	-	-	-	-	-	-
0x405	-	-	-	-	-	-	-	-
0x406	-	-	-	-	SENS UP FUNCTION 0:OFF 1:ON	SENS UP MAX 0x00:x2, 0x01:x4, 0x02:x8, 0x03:x16, 0x04:x32, 0x05:x64, 0x06:x128, 0x07:x256(FLD)		
0x407	-	-	-	-	-	-	-	-
0x408	-	-	-	-	-	-	-	-
0x409	-	-	-	-	-	-	-	-
0x40A	-	-	-	-	-	-	-	-
0x40B	-	-	-	-	-	-	-	-
0x40C	-	-	-	-	-	-	-	-
0x40D	-	-	-	-	-	-	-	-
0x40E	-	-	-	-	-	-	-	-
0x40F	-	-	-	-	-	-	-	-
0x410	-	-	-	-	-	-	-	-
0x411	AGC LOW MAX actual AGC LOW MAX(dB) = ([0x05 … 0xFF] * 4 * 0.035) + 5.3 (notice 1) Be sure to set 5 or more large values. (notice 2) Be sure to set large value than AGC MIN.							
0x412	AGC MID MAX actual AGC MID MAX(dB) = ([0x05 … 0xFF] * 4 * 0.035) + 5.3 (notice 1) Be sure to set 5 or more large values. (notice 2) Be sure to set large value than AGC MIN.							
0x413	AGC HI MAX actual AGC HI MAX(dB) = ([0x05 … 0xFF] * 4 * 0.035) + 5.3 (notice 1) Be sure to set 5 or more large values. (notice 2) Be sure to set large value than AGC MIN.							
0x414	AGC MIN actual AGC MIN(dB) = ([0x05 … 0xFF] * 4 * 0.035) + 5.3 (notice 1) Be sure to set 5 or more large values. (notice 2) Be sure to set small value less than all AGC LO/MID/HI MAX.							

Address	Bit							
	7	6	5	4	3	2	1	0
0x415	—	—	—	—	—	—	—	—
0x416	—	—	—	—	—	—	—	—
0x417	—	—	—	—	MANUAL GAIN(AGC OFF GAIN)			
					0x01:6(dB) … 0x24:41(dB)			
0x418	—	—	—	—	HSBLC LEVEL			
					0x01:1(min) … 0x08:8(max)			
0x419	HSBLC WINDOW V START POSITION				HSBLC WINDOW H START POSITION			
	0x00:0(TOP) … 0x07:7(BOTTOM)				0x00:0(LEFT) … 0x07:7(RIGHT)			
0x420	HSBLC WINDOW V SIZE				HSBLC WINDOW H SIZE			
	0x01:1(min) … 0x08:8(max)				0x01:1(min) … 0x08:8(max)			
0x41B	—	—	—	—	BLC GAIN		BLC MODE	
					0x00:LOW, 0x02:MID, 0x03:HI		0x00:OFF, 0x01:BLC, 0x10:HSBLC	
0x41C	BLC WINDOW V START POSITION				BLC WINDOW H START POSITION			
	0x00:0(TOP) … 0x07:7(BOTTOM)				0x00:0(LEFT) … 0x07:7(RIGHT)			
0x41D	BLC WINDOW V SIZE				BLC WINDOW H SIZE			
	0x01:1(min) … 0x08:8(max)				0x01:1(min) … 0x08:8(max)			
0x41E	—	—	—	—	—	—	—	—
0x41F	—	—	—	—	—	—	—	—
0x420	—	—	—	—	—	—	—	—
0x421	—	—	—	—	—	—	—	—
0x422	—	—	—	—	—	—	—	—
0x423	—	—	—	—	—	—	—	—
0x424	—	—	—	—	—	—	—	—
0x425	—	—	—	—	—	—	—	—
0x426	—	—	—	—	—	—	—	—
0x427	—	—	—	—	—	—	—	—
0x428	—	—	—	—	—	—	—	—
0x429	—	—	—	—	—	—	—	—
0x42A	—	—	—	—	—	—	—	—
0x42B	—	—	—	—	—	—	—	—
0x42C	—	—	—	—	—	—	—	—
0x42D	—	—	—	—	—	—	—	—
0x42E	—	—	—	—	—	—	—	—
0x42F	—	—	—	—	—	—	—	—
0x430	—	—	—	—	—	—	—	—
0x431	—	—	—	—	—	—	—	—
0x432	—	—	—	—	—	—	—	—
0x433	—	—	—	—	—	—	—	—

Address	Bit							
	7	6	5	4	3	2	1	0
0x434	—	—	—	—	—	—	—	—
0x435	—	—	—	—	—	—	—	—
0x436	—	—	—	—	—	—	—	—
0x437	—	—	—	—	—	—	—	—
0x438	—	—	—	—	—	—	—	—
0x439	—	—	—	—	—	—	—	—
0x43A	—	—	—	—	—	—	—	—
0x43B	—	—	—	—	—	—	—	—
0x43C	—	—	—	—	—	—	—	—
0x43D	—	—	—	—	—	—	—	—
0x43E	—	—	—	—	—	—	—	—
0x43F	—	—	—	—	—	—	—	—
0x440	—	—	—	—	—	—	—	—
0x441	—	—	—	—	—	—	—	—
0x442	—	—	—	—	—	—	—	—
0x443	—	—	—	—	—	—	—	—
0x444	—	—	—	—	—	—	—	—
0x445	—	—	—	—	—	—	—	—
0x446	—	—	—	—	—	—	—	—
0x447	—	—	—	—	—	—	—	—
0x448	—	—	—	—	—	—	WDR MODE 0x00:OFF 0x01:USER1 0x02:USER2	
0x449	WDR USER1 H-LEVEL 0x00:0(LEVEL) … 0x0F:15(LEVEL)				WDR USER1 L-LEVEL 0x00:0(LEVEL) … 0x0F:15(LEVEL)			
0x44A	WDR USER2 H-LEVEL 0x00:0(LEVEL) … 0x0F:15(LEVEL)				WDR USER2 L-LEVEL 0x00:0(LEVEL) … 0x0F:15(LEVEL)			
0x44B	—	—	—	—	—	—	—	—
0x44C	—	—	—	—	—	—	—	—
0x44D	—	—	—	—	—	—	—	—
0x44E	—	—	—	—	—	—	—	—
0x44F	—	—	—	—	—	—	—	—
0x450	—	—	—	—	—	—	—	3DNR 0:OFF 1:ON
0x451	—	3DNR LEVEL 0x00:0(LEVEL) … 0x64:100(LEVEL)						
0x452	—	—	—	—	—	—	—	—
0x453	—	—	—	—	—	—	—	—
0x454	—	—	—	—	—	—	—	—
0x455	—	—	—	—	—	—	—	—

Address	Bit							
	7	6	5	4	3	2	1	0
0x456	—	—	—	—	—	—	—	—
0x457	—	—	—	—	—	—	—	—
0x458	—	—	—	SHARPNESS 0x00:0(LEVEL) … 0x1F:31(LEVEL)				
0x459	—	—	—	—	—	—	—	—
0x45A	—	—	—	—	—	—	—	—
0x45B	—	—	—	—	—	—	—	—
0x45C	—	—	—	—	—	—	—	—
0x45D	—	—	—	—	—	—	—	—
0x45E	—	—	—	—	—	—	—	—
0x45F	—	—	—	—	—	—	—	—
0x460	—	—	—	—	—	—	—	MOTION 0:OFF 1:ON
0x461	—	MOTION VIEW 0:OFF 1:ON	—	—	—	—	—	—
0x462	MOTION AREA1 SENSITIVITY 0x32:0, 0x31:1 … 0x28:10 … 0x1E:20 … 0x14:30 … 0x0B:39, 0x0A:40							
0x463	MOTION AREA2 SENSITIVITY 0x32:0, 0x31:1 … 0x28:10 … 0x1E:20 … 0x14:30 … 0x0B:39, 0x0A:40							
0x464	MOTION AREA3 SENSITIVITY 0x32:0, 0x31:1 … 0x28:10 … 0x1E:20 … 0x14:30 … 0x0B:39, 0x0A:40							
0x465	MOTION AREA4 SENSITIVITY 0x32:0, 0x31:1 … 0x28:10 … 0x1E:20 … 0x14:30 … 0x0B:39, 0x0A:40							
0x466	MOTION AREA1 START H POSITION 0x04:(LEFT) … 0xBB:(RIGHT) (note) Sum of "H POSITION" and "H size" is needed to be under 0xBF.							
0x467	MOTION AREA1 START V POSITION 0x01:(TOP) … 0x8F:(BOTTOM) (note) Sum of "V POSITION" and "V size" is needed to be under 0x90.							
0x468	MOTION AREA1 H SIZE 0x04:(min) … 0xBB:(max) (note) Sum of "H POSITION" and "H size" is needed to be under 0xBF.							
0x469	MOTION AREA1 V SIZE 0x01:(min) … 0x90:(max) (note) Sum of "V POSITION" and "V size" is needed to be under 0x90.							

Address	Bit							
	7	6	5	4	3	2	1	0
0x46A	MOTION AREA2 START H POSITION 0x04:(LEFT) … 0xBB:(RIGHT) (note) Sum of "H POSITION" and "H size" is needed to be under 0xBF.							
0x46B	MOTION AREA2 START V POSITION 0x01:(TOP) … 0x8F:(BOTTOM) (note) Sum of "V POSITION" and "V size" is needed to be under 0x90.							
0x46C	MOTION AREA2 H SIZE 0x04:(min) … 0xBB:(max) (note) Sum of "H POSITION" and "H size" is needed to be under 0xBF.							
0x46D	MOTION AREA2 V SIZE 0x01:(min) … 0x90:(max) (note) Sum of "V POSITION" and "V size" is needed to be under 0x90.							
0x46E	MOTION AREA3 START H POSITION 0x04:(LEFT) … 0xBB:(RIGHT) (note) Sum of "H POSITION" and "H size" is needed to be under 0xBF.							
0x46F	MOTION AREA3 START V POSITION 0x01:(TOP) … 0x8F:(BOTTOM) (note) Sum of "V POSITION" and "V size" is needed to be under 0x90.							
0x470	MOTION AREA3 H SIZE 0x04:(min) … 0xBB:(max) (note) Sum of "H POSITION" and "H size" is needed to be under 0xBF.							
0x471	MOTION AREA3 V SIZE 0x01:(min) … 0x90:(max) (note) Sum of "V POSITION" and "V size" is needed to be under 0x90.							
0x472	MOTION AREA4 START H POSITION 0x04:(LEFT) … 0xBB:(RIGHT) (note) Sum of "H POSITION" and "H size" is needed to be under 0xBF.							
0x473	MOTION AREA4 START V POSITION 0x01:(TOP) … 0x8F:(BOTTOM) (note) Sum of "V POSITION" and "V size" is needed to be under 0x90.							
0x474	MOTION AREA4 H SIZE 0x04:(min) … 0xBB:(max) (note) Sum of "H POSITION" and "H size" is needed to be under 0xBF.							
0x475	MOTION AREA4 V SIZE 0x01:(min) … 0x90:(max) (note) Sum of "V POSITION" and "V size" is needed to be under 0x90.							
0x476	—	—	—	—	MOTION AREA4 DISPLAY 0:OFF 1:ON	MOTION AREA3 DISPLAY 0:OFF 1:ON	MOTION AREA2 DISPLAY 0:OFF 1:ON	MOTION AREA1 DISPLAY 0:OFF 1:ON
0x477	—	—	—	—	—	—	—	—
0x478	—	—	—	—	—	—	—	—
0x479	—	—	—	—	—	—	—	—
0x47A	—	—	—	—	—	—	—	—

Address	Bit							
	7	6	5	4	3	2	1	0
0x47B	—	—	—	—	—	—	—	—
0x47C	—	—	—	—	—	—	—	—
0x47D	—	—	—	—	—	—	—	—
0x47E	—	—	—	—	—	—	—	—
0x47F	—	—	—	—	—	—	—	—
0x480	—	—	—	—	—	—	—	—
0x481	—	—	—	—	—	—	—	—
0x482	—	—	—	—	—	—	—	—
0x483	—	—	—	—	—	—	—	—
0x484	—	—	—	—	—	—	—	—
0x485	—	—	—	—	—	—	—	—
0x486	—	—	—	—	—	—	—	—
0x487	—	—	—	—	—	—	—	—
0x488	—	—	—	—	—	—	—	—
0x489	—	—	—	—	—	—	—	—
0x48A	—	—	—	—	—	—	—	—
0x48B	—	—	—	—	—	—	—	—
0x48C	—	—	—	—	—	—	—	—
0x48D	—	—	—	—	—	—	—	—
0x48E	—	—	—	—	—	—	—	—
0x48F	—	—	—	—	—	—	—	—
0x490	—	—	—	—	—	—	—	—
0x491	—	—	—	—	—	—	—	—
0x492	—	—	—	—	—	—	—	—
0x493	—	—	—	—	—	—	—	—
0x494	—	—	—	—	—	—	—	—
0x495	—	—	—	—	—	—	—	—
0x496	—	—	—	—	—	—	—	—
0x497	—	—	—	—	—	—	—	—
0x498	—	—	—	—	—	—	—	—
0x499	—	—	—	—	—	—	—	—
0x49A	—	—	—	—	—	—	—	—
0x49B	—	—	—	—	—	—	—	—
0x49C	—	—	—	—	—	—	—	—
0x49D	—	—	—	—	—	—	—	—
0x49E	—	—	—	—	—	—	—	—
0x49F	—	—	—	—	—	—	—	—
0x4A0	—	—	—	—	—	—	—	—
0x4A1	—	—	—	—	—	—	—	—
0x4A2	—	—	—	—	—	—	—	—
0x4A3	—	—	—	—	—	—	—	—
0x4A4	—	—	—	—	—	—	—	—
0x4A5	—	—	—	—	—	—	—	—
0x4A6	—	—	—	—	—	—	—	—
0x4A7	—	—	—	—	—	—	—	—

<table>
<tr><th rowspan="2">Address</th><th colspan="8">Bit</th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0x4A8</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>ZOOM
0:OFF
1:ON</td></tr>
<tr><td>0x4A9</td><td colspan="8">ZOOM(magnification)
0x00:x1.0, 0x01:x1.1 … 0x09:x1.9, 0x0A:x2.0, 0x0B:x2.1 … 0x13:x2.9, 0x14:x3.0,
0x15:x3.1 … 0x1D:x3.9, 0x1E:x4.0, 0x1F:x4.5, 0x20:x5.0, 0x21:x5.5, 0x22:x6.0,
0x23:x6.5, 0x24:x7.0, 0x25:x7.5, 0x26:x8.0, 0x27:x9, 0x28:x10, 0x29:x11, 0x2A:x12,
0x2B:x13, 0x2C:x14, 0x2D:x15, 0x2E:x16, 0x2F:x18, 0x30:x20, 0x31:x22, 0x32:x24,
0x33:x26, 0x34:x28, 0x35:x30, 0x36:x32</td></tr>
<tr><td>0x4AA</td><td colspan="8">ZOOM(PAN)
0x00:-100(LEFT), 0x01:-99 … 0x63:-1, 0x64:0(CENTER), 0x65:+1 … 0xC7:+99, 0xC8:+100(RIGHT)</td></tr>
<tr><td>0x4AB</td><td colspan="8">ZOOM(TILT)
0x00:-100(UP), 0x01:-99 … 0x63:-1, 0x64:0(CENTER), 0x65:+1 … 0xC7:+99, 0xC8:+100(DOWN)</td></tr>
<tr><td>0x4AC</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4AD</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4AE</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4AF</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4B0</td><td>—</td><td>—</td><td>—</td><td>—</td><td>NEG.IMAGE
0:OFF
1:ON</td><td colspan="2">FLIP
0x00:OFF 0x01:H
0x02:V 0x03:HV</td><td>FREEZE
0:OFF
1:ON</td></tr>
<tr><td>0x4B1</td><td>—</td><td>—</td><td>—</td><td colspan="5">GAMMA
0x00:USER, 0x01:0.05 … 0x09:0.45 … 0x14:1.0</td></tr>
<tr><td>0x4B2</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4B3</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4B4</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4B5</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4B6</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4B7</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4B8</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4B9</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4BA</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4BB</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4BC</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4BD</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4BE</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4BF</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4C0</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4C1</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4C2</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4C3</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4C4</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0x4C5</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
</table>

Address	Bit							
	7	6	5	4	3	2	1	0
0x4C6	—	—	—	—	—	—	—	—
0x4C7	—	—	—	—	—	—	—	—
0x4C8	—	—	—	—	—	—	—	—
0x4C9	—	—	—	—	—	—	—	—
0x4CA	—	—	—	—	—	—	—	—
0x4CB	—	—	—	—	—	—	—	—
0x4CC	—	—	—	—	—	—	—	—
0x4CD	—	—	—	—	—	—	—	—
0x4CE	—	—	—	—	—	—	—	—
0x4CF	—	—	—	—	—	—	—	—
0x4D0	—	—	—	—	—	—	—	—
0x4D1	—	—	—	—	—	—	—	—
0x4D2	—	—	—	—	—	—	—	—
0x4D3	—	—	—	—	—	—	—	—
0x4D4	—	—	—	—	—	—	—	—
0x4D5	—	—	—	—	—	—	—	—
0x4D6	—	—	—	—	—	—	—	—
0x4D7	—	—	—	—	—	—	—	—
0x4D8	—	—	—	—	—	—	—	—
0x4D9	—	—	—	—	—	—	—	—
0x4DA	—	—	—	—	—	—	—	—
0x4DB	—	—	—	—	—	—	—	—
0x4DC	—	—	—	—	—	—	—	—
0x4DD	—	—	—	—	—	—	—	—
0x4DE	—	—	—	—	—	—	—	—
0x4DF	—	—	—	—	—	—	—	—
0x4E0	—	—	—	—	—	—	—	—
0x4E1	—	—	—	—	—	—	—	—
0x4E2	—	—	—	—	—	—	—	—
0x4E3	—	—	—	—	—	—	—	—
0x4E4	—	—	—	—	—	—	—	—
0x4E5	—	—	—	—	—	—	—	—
0x4E6	—	—	—	—	—	—	—	—
0x4E7	—	—	—	—	—	—	—	—
0x4E8	—	—	—	—	—	—	—	—
0x4E9	—	—	—	—	—	—	—	—
0x4EA	—	—	—	—	—	—	—	—
0x4EB	—	—	—	—	—	—	—	—
0x4EC	—	—	—	—	—	—	—	—
0x4ED	—	—	—	—	—	—	—	—
0x4EE	—	—	—	—	—	—	—	—
0x4EF	—	—	—	—	—	—	—	—
0x4F0	—	—	—	—	—	—	—	—
0x4F1	—	—	—	—	—	—	—	—
0x4F2	—	—	—	—	—	—	—	—

Address	Bit							
	7	6	5	4	3	2	1	0
0x4F3	—	SETUP ON/OFF 0:0(IRE) 1:7.5(IRE)	SETUP LEVEL 0x00 … 0x0x3F 0x1C:0(IRE)/0x31:7.5(IRE)					
0x4F4	—	—	—	—	—	—	—	—
0x4F5	—	—	—	—	—	—	—	—
0x4F6	—	—	—	—	—	—	—	—
0x4F7	—	—	—	—	—	—	—	—
0x4F8	—	—	—	—	—	—	—	—
0x4F9	—	—	—	—	—	—	—	—
0x4FA	—	—	—	—	—	—	—	—
0x4FB	—	—	—	—	—	—	—	—
0x4FC	—	—	—	—	—	—	—	—
0x4FD	—	—	—	—	—	—	—	—
0x4FE	—	—	BPC LEVEL 0x01:1 … 0x04:4			BPC FLD 0x00:x4, 0x01:x8, 0x02:x16, 0x03:x32, 0x04:x64		
0x4FF	—	—	—	—	—	—	—	—

Address	Bit							
	7	6	5	4	3	2	1	0
0x500	—	—	—	—	—	—	—	—
0x501	—	—	—	—	—	—	—	—
0x502	—	—	—	—	—	—	—	—
0x503	—	—	—	—	—	—	—	—
0x504	—	—	—	—	—	—	—	—
0x505	—	—	—	—	—	—	—	—
0x506	—	—	—	—	—	—	—	—
0x507	—	—	—	—	—	—	—	—
0x508	—	—	—	—	—	—	—	—
0x509	—	—	—	—	—	—	—	—
0x50A	—	—	—	—	—	—	—	—
0x50B	—	—	—	—	—	—	—	—
0x50C	—	—	—	—	—	—	—	—
0x50D	—	—	—	—	—	—	—	—
0x50E	—	—	—	—	—	—	—	—
0x50F	—	—	—	—	—	—	—	—
0x510	—	—	—	—	—	—	—	—
0x511	—	—	—	—	—	—	—	—
0x512	—	—	—	—	—	—	—	—
0x513	—	—	—	—	—	—	—	—
0x514	—	—	—	—	—	—	—	—
0x515	—	—	—	—	—	—	—	—
0x516	—	—	—	—	—	—	—	—
0x517	—	—	—	—	—	—	—	—
0x518	—	—	—	—	—	—	—	—
0x519	—	—	—	—	—	—	—	—
0x51A	—	—	—	—	—	—	—	—
0x51B	—	—	—	—	—	—	—	—
0x51C	—	—	—	—	—	—	—	—
0x51D	—	—	—	—	—	—	—	—
0x51E	—	—	—	—	—	—	—	—
0x51F	—	—	—	—	—	—	—	—
0x520	—	—	—	—	—	—	—	—
0x521	—	—	—	—	—	—	—	—
0x522	—	—	—	—	—	—	—	—
0x523	—	—	—	—	—	—	—	—
0x524	—	—	—	—	—	—	—	—
0x525	—	—	—	—	—	—	—	—
0x526	—	—	—	—	—	—	—	—
0x527	—	—	—	—	—	—	—	—
0x528	—	—	—	—	—	—	—	—
0x529	—	—	—	—	—	—	—	—
0x52A	—	—	—	—	—	—	—	—
0x52B	—	—	—	—	—	—	—	—
0x52C	—	—	—	—	—	—	—	—

Address	Bit							
	7	6	5	4	3	2	1	0
0x52D	—	—	—	—	—	—	—	—
0x52E	—	—	—	—	—	—	—	—
0x52F	—	—	—	—	—	—	—	—
0x530	—	—	—	—	—	—	—	—
0x531	—	—	—	—	—	—	—	—
0x532	—	—	—	—	—	—	—	—
0x533	—	—	—	—	—	—	—	—
0x534	—	—	—	—	—	—	—	—
0x535	—	—	—	—	—	—	—	—
0x536	—	—	—	—	—	—	—	—
0x537	—	—	—	—	—	—	—	—
0x538	—	—	—	—	—	—	—	—
0x539	—	—	—	—	—	—	—	—
0x53A	—	—	—	—	—	—	—	—
0x53B	—	—	—	—	—	—	—	—
0x53C	—	—	—	—	—	—	—	—
0x53D	—	—	—	—	—	—	—	—
0x53E	—	—	—	—	—	—	—	—
0x53F	—	—	—	—	—	—	—	—
0x540	—	—	—	—	—	—	—	—
0x541	—	—	—	—	—	—	—	—
0x542	—	—	—	—	—	—	—	—
0x543	—	—	—	—	—	—	—	—
0x544	—	—	—	—	—	—	—	—
0x545	—	—	—	—	—	—	—	—
0x546	—	—	—	—	—	—	—	—
0x547	—	—	—	—	—	—	—	—
0x548	—	—	—	—	—	—	—	—
0x549	—	—	—	—	—	—	—	—
0x54A	—	—	—	—	—	—	—	—
0x54B	—	—	—	—	—	—	—	—
0x54C	—	—	—	—	—	—	—	—
0x54D	—	—	—	—	—	—	—	—
0x54E	—	—	—	—	—	—	—	—
0x54F	—	—	—	—	—	—	—	—
0x550	—	—	—	—	—	—	—	—
0x551	—	—	—	—	—	—	—	—
0x552	—	—	—	—	—	—	—	—
0x553	—	—	—	—	—	—	—	—
0x554	—	—	—	—	—	—	—	—
0x555	—	—	—	—	—	—	—	—
0x556	—	—	—	—	—	—	—	—
0x557	—	—	—	—	—	—	—	—
0x558	—	—	—	—	—	—	—	—
0x559	—	—	—	—	—	—	—	—

Address	Bit							
	7	6	5	4	3	2	1	0
0x55A	—	—	—	—	—	—	—	—
0x55B	—	—	—	—	—	—	—	—
0x55C	—	—	—	—	—	—	—	—
0x55D	—	—	—	—	—	—	—	—
0x55E	—	—	—	—	—	—	—	—
0x55F	—	—	—	—	—	—	—	—
0x560	—	—	—	—	—	—	—	—
0x561	—	—	—	—	—	—	—	—
0x562	—	—	—	—	—	—	—	—
0x563	—	—	—	—	—	—	—	—
0x564	—	—	—	—	—	—	—	—
0x565	—	—	—	—	—	—	—	—
0x566	—	—	—	—	—	—	—	—
0x567	—	—	—	—	—	—	—	—
0x568	—	—	—	—	—	—	—	—
0x569	—	—	—	—	—	—	—	—
0x56A	—	—	—	—	—	—	—	—
0x56B	—	—	—	—	—	—	—	—
0x56C	—	—	—	—	—	—	—	—
0x56D	—	—	—	—	—	—	—	—
0x56E	—	—	—	—	—	—	—	—
0x56F	—	—	—	—	—	—	—	—
0x570	—	—	—	—	—	—	—	—
0x571	—	—	—	—	—	—	—	—
0x572	—	—	—	—	—	—	—	—
0x573	—	—	—	—	—	—	—	—
0x574	—	—	—	—	—	—	—	—
0x575	—	—	—	—	—	—	—	—
0x576	—	—	—	—	—	—	—	—
0x577	—	—	—	—	—	—	—	—
0x578	—	—	—	—	—	—	—	—
0x579	—	—	—	—	—	—	—	—
0x57A	—	—	—	—	—	—	—	—
0x57B	—	—	—	—	—	—	—	—
0x57C	—	—	—	—	—	—	—	—
0x57D	—	—	—	—	—	—	—	—
0x57E	—	—	—	—	—	—	—	—
0x57F	—	—	—	—	—	—	—	—
0x580	—	—	—	—	—	—	—	—
0x581	—	—	—	—	—	—	—	—
0x582	—	—	—	—	—	—	—	—
0x583	—	—	—	—	—	—	—	—
0x584	—	—	—	—	—	—	—	—
0x585	—	—	—	—	—	—	—	—
0x586	—	—	—	—	—	—	—	—

Address	Bit							
	7	6	5	4	3	2	1	0
0x587	—	—	—	—	—	—	—	—
0x588	—	—	—	—	—	—	—	—
0x589	—	—	—	—	—	—	—	—
0x58A	—	—	—	—	—	—	—	—
0x58B	—	—	—	—	—	—	—	—
0x58C	—	—	—	—	—	—	—	—
0x58D	—	—	—	—	—	—	—	—
0x58E	—	—	—	—	—	—	—	—
0x58F	—	—	—	—	—	—	—	—
0x590	—	—	—	—	—	—	—	—
0x591	—	—	—	—	—	—	—	—
0x592	—	—	—	—	—	—	—	—
0x593	—	—	—	—	—	—	—	—
0x594	—	—	—	—	—	—	—	—
0x595	—	—	—	—	—	—	—	—
0x596	—	—	—	—	—	—	—	—
0x597	—	—	—	—	—	—	—	—
0x598	—	—	—	—	—	—	—	—
0x599	—	—	—	—	—	—	—	—
0x59A	—	—	—	—	—	—	—	—
0x59B	—	—	—	—	—	—	—	—
0x59C	—	—	—	—	—	—	—	—
0x59D	—	—	—	—	—	—	—	—
0x59E	—	—	—	—	—	—	—	—
0x59F	—	—	—	—	—	—	—	—
0x5A0	—	—	—	—	—	—	—	—
0x5A1	—	—	—	—	—	—	—	—
0x5A2	—	—	—	—	—	—	—	—
0x5A3	—	—	—	—	—	—	—	—
0x5A4	—	—	—	—	—	—	—	—
0x5A5	—	—	—	—	—	—	—	—
0x5A6	—	—	—	—	—	—	—	—
0x5A7	—	—	—	—	—	—	—	—
0x5A8	—	—	—	—	—	—	—	—
0x5A9	—	—	—	—	—	—	—	—
0x5AA	—	—	—	—	—	—	—	—
0x5AB	—	—	—	—	—	—	—	—
0x5AC	—	—	—	—	—	—	—	—
0x5AD	—	—	—	—	—	—	—	—
0x5AE	—	—	—	—	—	—	—	—
0x5AF	—	—	—	—	—	—	—	—
0x5B0	—	—	—	—	—	—	—	—
0x5B1	—	—	—	—	—	—	—	—
0x5B2	—	—	—	—	—	—	—	—
0x5B3	—	—	—	—	—	—	—	—

Address	Bit							
	7	6	5	4	3	2	1	0
0x5B4	—	—	—	—	—	—	—	—
0x5B5	—	—	—	—	—	—	—	—
0x5B6	—	—	—	—	—	—	—	—
0x5B7	—	—	—	—	—	—	—	—
0x5B8	—	—	—	—	—	—	—	—
0x5B9	—	—	—	—	—	—	—	—
0x5BA	—	—	—	—	—	—	—	—
0x5BB	—	—	—	—	—	—	—	—
0x5BC	—	—	—	—	—	—	—	—
0x5BD	—	—	—	—	—	—	—	—
0x5BE	—	—	—	—	—	—	—	—
0x5BF	—	—	—	—	—	—	—	—
0x5C0	—	—	—	—	—	—	—	—
0x5C1	—	—	—	—	—	—	—	—
0x5C2	—	—	—	—	—	—	—	—
0x5C3	—	—	—	—	—	—	—	—
0x5C4	—	—	—	—	—	—	—	—
0x5C5	—	—	—	—	—	—	—	—
0x5C6	—	—	—	—	—	—	—	—
0x5C7	—	—	—	—	—	—	—	—
0x5C8	—	—	—	—	—	—	—	—
0x5C9	—	—	—	—	—	—	—	—
0x5CA	—	—	—	—	—	—	—	—
0x5CB	—	—	—	—	—	—	—	—
0x5CC	—	—	—	—	—	—	—	—
0x5CD	—	—	—	—	—	—	—	—
0x5CE	—	—	—	—	—	—	—	—
0x5CF	—	—	—	—	—	—	—	—
0x5D0	—	—	—	—	—	—	—	—
0x5D1	—	—	—	—	—	—	—	—
0x5D2	—	—	—	—	—	—	—	—
0x5D3	—	—	—	—	—	—	—	—
0x5D4	—	—	—	—	—	—	—	—
0x5D5	—	—	—	—	—	—	—	—
0x5D6	—	—	—	—	—	—	—	—
0x5D7	—	—	—	—	—	—	—	—
0x5D8	—	—	—	—	—	—	—	—
0x5D9	—	—	—	—	—	—	—	—
0x5DA	—	—	—	—	—	—	—	—
0x5DB	—	—	—	—	—	—	—	—
0x5DC	—	—	—	—	—	—	—	—
0x5DD	—	—	—	—	—	—	—	—
0x5DE	—	—	—	—	—	—	—	—
0x5DF	—	—	—	—	—	—	—	—

Address	Bit							
	7	6	5	4	3	2	1	0
0x5E0	—	—	—	—	—	—	—	BPC 0:(OFF) 1:START
0x5E1	—	—	—	—	—	—	—	FACTORY RESET 0:OFF 1:ON
0x5E2	—	—	—	—	—	—	—	DIGIT OUT 0:OFF 1:ON
0x5E3	—	—	—	—	—	—	—	—
0x5E4	—	—	—	—	—	—	—	—
0x5E5	—	—	—	—	—	—	—	—
0x5E6	—	—	—	—	—	—	—	—
0x5E7	—	—	—	—	—	—	—	—
0x5E8	—	—	—	—	—	—	—	—
0x5E9	—	—	—	—	—	—	—	—
0x5EA	—	—	—	—	—	—	—	—
0x5EB	—	—	—	—	—	—	—	—
0x5EC	—	—	—	—	—	—	—	—
0x5ED	—	—	—	—	—	—	—	—
0x5EE	—	—	—	—	—	—	—	—
0x5EF	—	—	—	—	—	—	—	—
0x5F0	—	—	—	—	—	—	—	—
0x5F1	—	—	—	—	—	—	—	—
0x5F2	—	—	—	—	—	—	—	—
0x5F3	—	—	—	—	—	—	—	—
0x5F4	—	—	—	—	—	—	—	—
0x5F5	—	—	—	—	—	—	—	—
0x5F6	—	—	—	—	—	—	—	—
0x5F7	—	—	—	—	—	—	—	—
0x5F8	—	—	—	—	—	—	—	—
0x5F9	—	—	—	—	—	—	—	—
0x5FA	—	—	—	—	—	—	—	—
0x5FB	—	—	—	—	—	—	—	—
0x5FC	—	—	—	—	—	—	—	—
0x5FD	—	—	—	—	—	—	—	—
0x5FE	—	—	—	—	—	—	—	—
0x5FF	—	—	—	—	—	—	—	—

6.5 SAMPLE CODE

The following pages are example of the camera parameter change by SPI communication.
In the examples, AGC mode is changed by using the method of 3WIRE SPI communication.

(1)SAMPLE CODE

This sample source code is installed in and tested on "R8C/1B" MCU made by Renesas.
Therefore, it is necessary to change source code into the specification of the microcomputer to be used.

```
//-----------------------------------------------------------------------
//
//  FILE        : SPI_TEST_3W.c
//                SPI communication test program
//  DATE        : 2012.02.27
//  DESCRIPTION : Test program to change camera parameters
//                by using MCU GPIO.
//                (SPI 3wire communication test)
//  Watec  S.Igarashi
//  CPU GROUP   :1B
//
//-----------------------------------------------------------------------

#include    sfr_r81b.h

#define     TRUE       1
#define     FALSE      0
#define     CLKMS    500             // for wait counter

//  global variables-------------------------------------------------------
unsigned char SEND_BUFF[0x10] ;      // Send Buffer(16 byte)
unsigned char READ_BUFF[0x10] ;      // Receive Buffer(16 byte)
unsigned char SW_STATUS ;            // SW input value
unsigned char SW_S_C[4] ;            // temporary variables for avoid SW chattering
unsigned int  Adr_C  = 0x400 ;       // Address counter

// prototyping ----------------------------------------------------------
// checking to which SW is ON
unsigned char    check_key_status(void) ;

// SPI Write (3wire)
unsigned char    SPI_WRITE_3W(unsigned char, unsigned char) ;

// SPI Read (3wire)
unsigned char    SPI_READ_3W(unsigned char, unsigned char) ;

// CAMERA CONTROL SUB ROUTINES
unsigned char    AGC_MODE(void) ;
unsigned char    BLC_AREA_ON(void) ;
unsigned char    BLC_AREA_OFF(void) ;
```

```
unsigned char    INDIRECT_READ(unsigned int) ;
// etc.
void             wait_ms(unsigned int) ;
void             wait_nop(void) ;


// ---- main loop -----------------------------------------------------
void main(void)
{
    int     i ;                     // loop counter

    // initialize MCU
    // inhibit interrupt(INT0)
    int0en  = 0 ;
    prc0    = 1 ;
    hra00   = 1 ;
    cm06    = 0 ;
    wait_nop() ;
    hra01   = 1 ;
    ocd2 = 1 ;

    // initialize GPIO port ------------------------------------------
    // port direction (input = 0 / output = 1)
    // port in group1(P1) are all SW input
    pd1_0 = 0 ;             // UP
    pd1_1 = 0 ;             // DOWN
    pd1_2 = 0 ;             // LEFT
    pd1_3 = 0 ;             // RIGHT
    pd1_4 = 0 ;             // SET
    pd1_5 = 0 ;             // DIP1
    pd1_6 = 0 ;             // DIP2
    pd1_7 = 0 ;             // DIP3

    // port I/O setting
    // P3_3,4,5,7 are using SPI communication
    pd3_3 = 0 ;             // MISO/SIMO(3wire)
    pd3_4 = 1 ;             // SCLK
    pd3_5 = 1 ;             // SS
    pd3_7 = 1 ;
    pd4_5 = 1 ;             // LED for status display

    // initialize output port
    p3_4 = 1 ;              // SCLK = H
    p3_5 = 1 ;              // SS = H
    p3_7 = 1 ;
    p4_5 = 1 ;              // LED OFF

    // initialize SW input port ALL OFF (ON = L)
    p1 = 0xFF ;

    //----- MAIN LOOP -------------------------------------------------------
    while(1){
        i = 0 ;                         // reset counter
```

```
        while(i<3){                     // read SW status in 3 times
            SW_S_C[i] = p1 ;
            wait_ms(10) ;               // 5ms waite
            SW_S_C[i+1] = p1 ;          // read one more
            if (SW_S_C[i] == SW_S_C[i+1]){
                i++ ;
            }else{
                i = 0 ;
            }
        }


        // if SW status equal 3 times, check previous SW status.
        // if it is not equal, key input was changed and stable.
        if(SW_STATUS != SW_S_C[3]){
            SW_STATUS = SW_S_C[3] ;
            // judge which SW is ON and processing SPI communication.
            if(check_key_status() != TRUE){
                p4_5 = 0 ;              // LED lit ON
            }else{
                p4_5 = 1 ;
            }
        }
    }
}

// end of main-------------------------------------------------------------------
//-------------------------------------------------------------------------------
// FUNCTION: check_key_status
//
//  Decode SW status and process SPI communication.
//  if illegal code was read, return FALSE.
//-------------------------------------------------------------------------------
unsicheck_key_status(void)
{
    unsigned    char    i ;
    unsigned    char    RetV ;          // TRUE/FALSE

    // decode 5 key
    switch( SW_STATUS & 0x1F ){
        case 0x1E:                      // 5key UP
            // INC address and Read Parameter
            if (Adr_C == 0x5FF){
                Adr_C = 0x400 ;
            }else{
                Adr_C++ ;
            }
            RetV = INDIRECT_READ(Adr_C) ;
            break ;
        case 0x1D:                      // 5key DOWN
            // DEC adress and Read Parameter
            if (Adr_C == 0x400){
                Adr_C = 0x5FF ;
```

```
            }else{
                Adr_C-- ;
            }
            RetV = INDIRECT_READ(Adr_C) ;
            break ;
        case 0x1B:                          // 5key LEFT
            RetV = BLC_AREA_OFF() ;
            break ;
        case 0x17:                          // 5key RIGHT
            RetV = BLC_AREA_ON() ;
            break ;
        case 0x0F:                          // 5key SET
            // AGC MODE change
            RetV = AGC_MODE() ;
            break ;
        default:


            RetV = TRUE ;                   // nothing to do
            break ;
    }

    return RetV ;
}

unsigned char AGC_MODE(void)
{
    unsigned char       AGC_V ;         // AGC MAX = 0:OFF/1:LOW/2:MID/3:HI
    unsigned char       RetV ;          // temp

    // AGC MAX
    // ADDRESS 0x401 READ
    SEND_BUFF[2] = 0x00 ;
    SEND_BUFF[3] = 0x00 ;
    SEND_BUFF[4] = 0x01 ;
    SEND_BUFF[5] = 0x00 ;
    SEND_BUFF[6] = 0x01 ;
    SEND_BUFF[7] = 0xAA ;
    if (SPI_WRITE_3W(0x36, 8) == FALSE ){
        return      FALSE ;
    }else{
        RetV = SPI_READ_3W(0x38, 4) ;
    }
                                            // AGC MODE value (bit3-2)
    AGC_V = (READ_BUFF[1] & 0x0C) >> 2 ;    // mask 00001100 and shift
    if (AGC_V == 3){                        // 0->1->2->3->0
        AGC_V =0;
    }else{
        AGC_V++ ;
    }
    wait_ms(1) ;

    // ADDRESS 0x401 WRITE
```

```
    SEND_BUFF[2] // BANK-0
    SEND_BUFF[3] = 0x80 ;
    SEND_BUFF[4] = 0x01 ;
    SEND_BUFF[5] = (READ_BUFF[1] & 0xF3) + (AGC_V << 2) ;
    SEND_BUFF[6] = SEND_BUFF[2] + SEND_BUFF[3] + SEND_BUFF[4] + SEND_BUFF[5] ;
    SEND_BUFF[7] = 0xAA ;
    return SPI_WRITE_3W(0x36, 8) ;
}

unsigned char BLC_AREA_ON(void)
{

    SEND_BUFF[2]    = 0x01 ;
    SEND_BUFF[3]    = 0x00 ;
    SEND_BUFF[4]    = 0x00 ;
    SEND_BUFF[5]    = 0x01 ;                // BLC AREA ON
    SEND_BUFF[6]    = 0x02 ;
    SEND_BUFF[7]    = 0xAA ;

    return SPI_WRITE_3W(0x36, 8) ;
}



unsigned char BLC_AREA_OFF(void)
{

    SEND_BUFF[2]    = 0x01 ;
    SEND_BUFF[3]    = 0x00 ;
    SEND_BUFF[4]    = 0x00 ;
    SEND_BUFF[5]    = 0x00 ;                //BLC AREA OFF
    SEND_BUFF[6]    = 0x01 ;
    SEND_BUFF[7]    = 0xAA ;

    return SPI_WRITE_3W(0x36, 8) ;
}

unsigned char INDIRECT_READ(unsigned int Pm_Adr)
{
    unsigned char Bank ;
    unsigned char Adrs ;

    Bank = (unsigned char)(Pm_Adr >> 8) - 0x04 ;
    Adrs = (unsigned char)(Pm_Adr) ;

    SEND_BUFF[2]    = 0x00 ;
    SEND_BUFF[3]    = Bank ;
    SEND_BUFF[4]    = Adrs ;
    SEND_BUFF[5]    = 0x00 ;
    SEND_BUFF[6]    = (unsigned char)(Bank + Adrs) ;
    SEND_BUFF[7]    = 0xAA ;

    if (SPI_WRITE_3W(0x36, 8) == TRUE ){
```

```
        return SPI_READ_3W(0x38, 4) ;
    }else{
        return FALSE ;
    }
}

//---------------------------------------------------------------------
// FUNCTION: SPI_WRITE_3W
//
//  SPI WRITE SEND_BUFF[0...(W_Byte-1)]
//  In case of Indirect access, it must W_Adr = 0x36 and W_Byte = 8.
//  In case of 3Wire, it need change GPIO(MISO/MOSI) direction
//  input/output.
//---------------------------------------------------------------------
unsigned char   SPI_WRITE_3W(unsigned char W_Adr, unsigned char W_Byte)
{
    int i ;
    int j ;
    int WriteCMD ;
    unsigned char   SendBit ;

    // make SPI WRITE COMMAND
    WriteCMD = (int)W_Adr << 1 ;
    SEND_BUFF[0] = (unsigned char)WriteCMD ;            // lower 8bit
    SEND_BUFF[1] = (unsigned char)(WriteCMD >> 8) ;     // higher 8bit

    i = 0 ;             // Byte Counter



    // initialize SPI bus
    p3_5 = 1 ;          // SS   = H
    p3_4 = 1 ;          // SCLK = H

    wait_nop() ;

    // SPI WRITE COMMAND send start
    pd3_3 = 1;                                      // set port direction MOSI(output)
    p3_3 = 1 ;                                      // MOSI = H
    p3_5 = 0 ;                                      // SS   = L
    while( i < W_Byte ){                            // W_Byte must 8 in INDIRECT ACCESS
        j = 1 ;                                     // Bit Mask
        while(j < 0xFF){                            // 8bit send
            SendBit = SEND_BUFF[i] & j ? 1 : 0 ;    // bit test
            j = j << 1 ;                            // j *= 2 ;
            p3_4 = 0 ;                              // SCLK = L
            p3_3 = SendBit ;                        // output LSB first bit data on MOSI
            wait_nop() ;
            p3_4 = 1 ;                              // SCLK = H(latch)
        }
        i ++ ;                                      // Next byte
        wait_nop() ;
    }
```

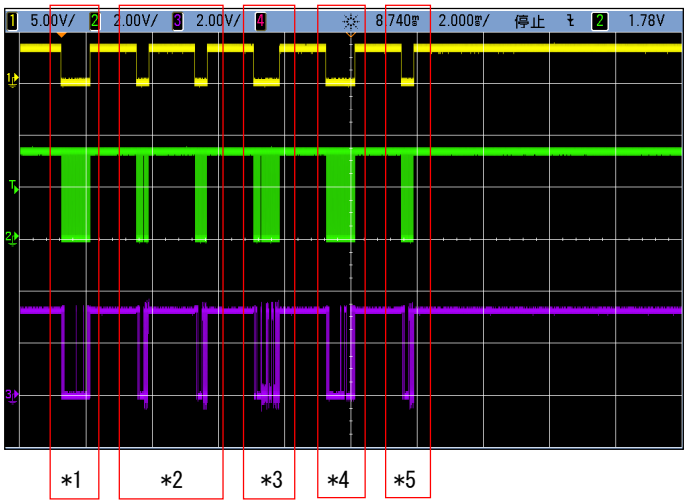
```
    p3_5 = 1 ;                                      // SS = H(END)
    pd3_3 = 0;                                      // set port direction MISO(input)

    wait_ms(1) ;

    // CTL-COMMAND execution is done ?
    READ_BUFF[0] = 0xAA ;                           // initialize
    while( READ_BUFF[0] == 0xAA ){                  // 0xAA means busy
        SPI_READ_3W(0x3B, 1) ;                      // ST CODE read
        wait_ms(1) ;                                // 1mS wait
    }
    if( READ_BUFF[0] == 0x55){                      // 0x55 means EXEC.NORMALLY
        return TRUE ;
    }else{
        return FALSE ;
    }

}

//---------------------------------------------------------------------
// FUNCTION: SPI_READ_3W
//
//  SPI WRITE READ_BUFF[0...(R_Byte-1)]
//  In case of 3Wire, it need change GPIO(MISO/MOSI) direction
//  input/output.
//---------------------------------------------------------------------
unsigned char SPI_READ_3W(unsigned char R_Adr, unsigned char R_Byte)
{
    int     ReadCMD ;                               // SPI READ COMMAND
    int i ;                                         // counter
    int j ;
    unsigned char    SendBit ;


    // make SPI READ COMMAND
    ReadCMD = ((int)R_Adr << 1) + 1 ;
    SEND_BUFF[0] = (unsigned char)ReadCMD ;         // lower 8bit
    SEND_BUFF[1] = (unsigned char)(ReadCMD >> 8) ;  // higher 8bit

    i = 0 ;                                         // Byte Counter

    // initialize SPI bus
    p3_5 = 1 ;                                      // SS   = H
    p3_4 = 1 ;                                      // SCLK = H

    wait_nop() ;

    // SPI READ COMMAND send start
    pd3_3 = 1;                                      // set port direction MOSI(output)
    p3_3 = 1 ;                                      // MOSI = H
    p3_5 = 0 ;                                      // SS   = L
    wait_nop() ;
```

```
    while( i < 2 ){                                 // 2 byte loop
        j = 1 ;                                     // bit mask
        while(j < 0xFF){                            // 8bit send
            SendBit = SEND_BUFF[i] & j ? 1 : 0 ;    // Bit Test
            j = j << 1 ;                            // j *= 2 ;
            p3_4 = 0 ;                              // SCLK = L
            p3_3 = SendBit ;                        // output LSB first bit data on MOSI
            wait_nop() ;
            p3_4 = 1 ;                              // SCLK = H(latch)
        }
        i ++ ;                                      // next byte
        wait_nop() ;
    }

    // SPI READ start
    pd3_3 = 0 ;                                     // set port direction MISO(input)

    wait_nop() ;
    wait_nop() ;
    wait_nop() ;
    wait_nop() ;
    i = 0 ;
    while( i < R_Byte ){                            // Loop R_Byte times
        j = 0 ;                                     // reset bit counter
        READ_BUFF[i] = 0x00 ;                       // clear buffer
        while(j < 8){                               // 8bit receive
            p3_4 = 0 ;                              // SCLK = L(Data Change)
            wait_nop() ;
            wait_nop() ;
            p3_4 = 1 ;                              // SCLK = H(latch)
            // LSB first receive from MISO
            READ_BUFF[i] += (unsigned char)p3_3 << j ;
            j ++ ;                                  // Next Bit
        }
        i ++ ;                                      // Next Byte
    }
    p3_5 = 1 ;                                      // SS = H(END)
    return TRUE ;
}


//---------------------------------------
// msec wait
//---------------------------------------
void    wait_ms(unsigned int wms)
{
    unsigned int w_cnt ;

    while ( wms--){
        w_cnt = CLKMS ;
        while(w_cnt--){}
    }
}
```

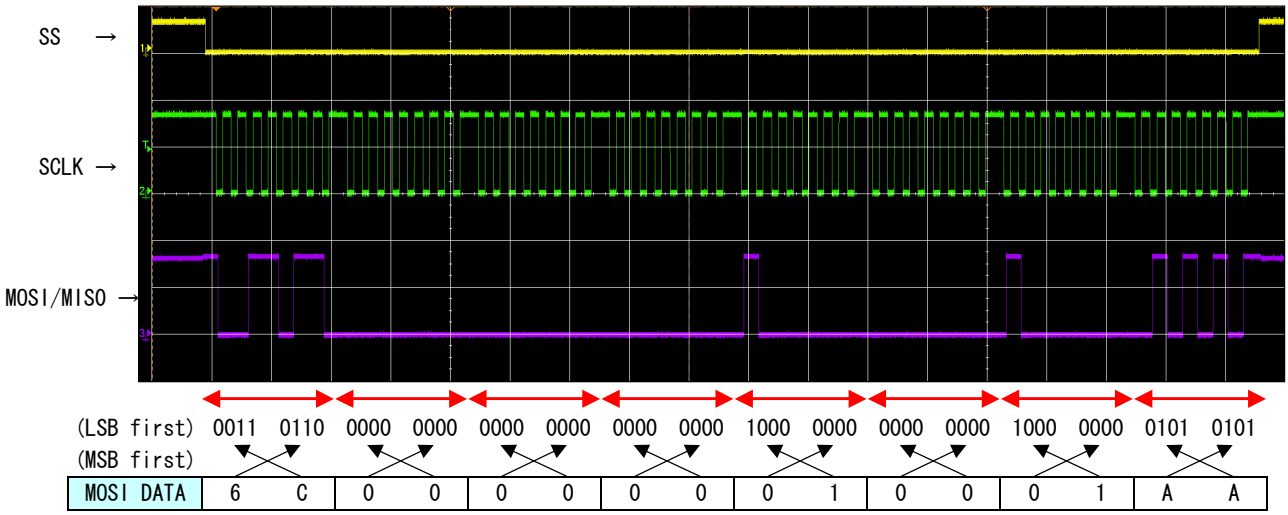
```
void    wait_nop(void)
{
    asm( "nop" ) ;
    asm( "nop" ) ;
    asm( "nop" ) ;
    asm( "nop" ) ;
    asm( "nop" ) ;
    asm( "nop" ) ;
}
```

(2) SAMPLE WAVEFORM

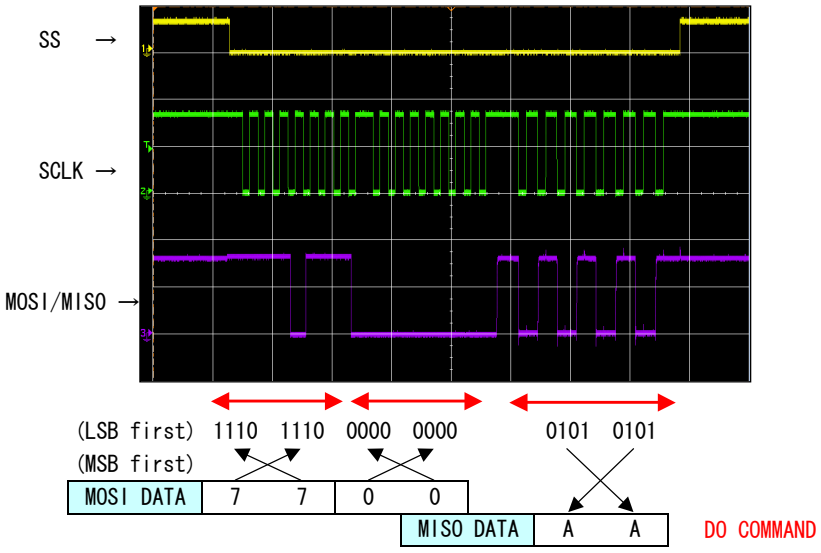
e.g. Change AGC MODE (MID -> HI) by using SPI(3wire) communucation.



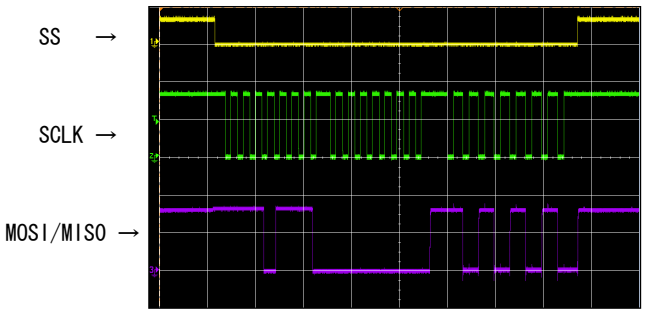
Step *1: Send "CAMERA PARAMETER READ1 command" to indirect registers(CAMERA: Address 0x36 - 0x3B) to get "AGC MODE: Address=0x401" parameter from CAMERA.
Following wave form is above operation by using "SPI write command".



Step *2: Send "SPI read command" to get and check ST CODE(address:0x3B).

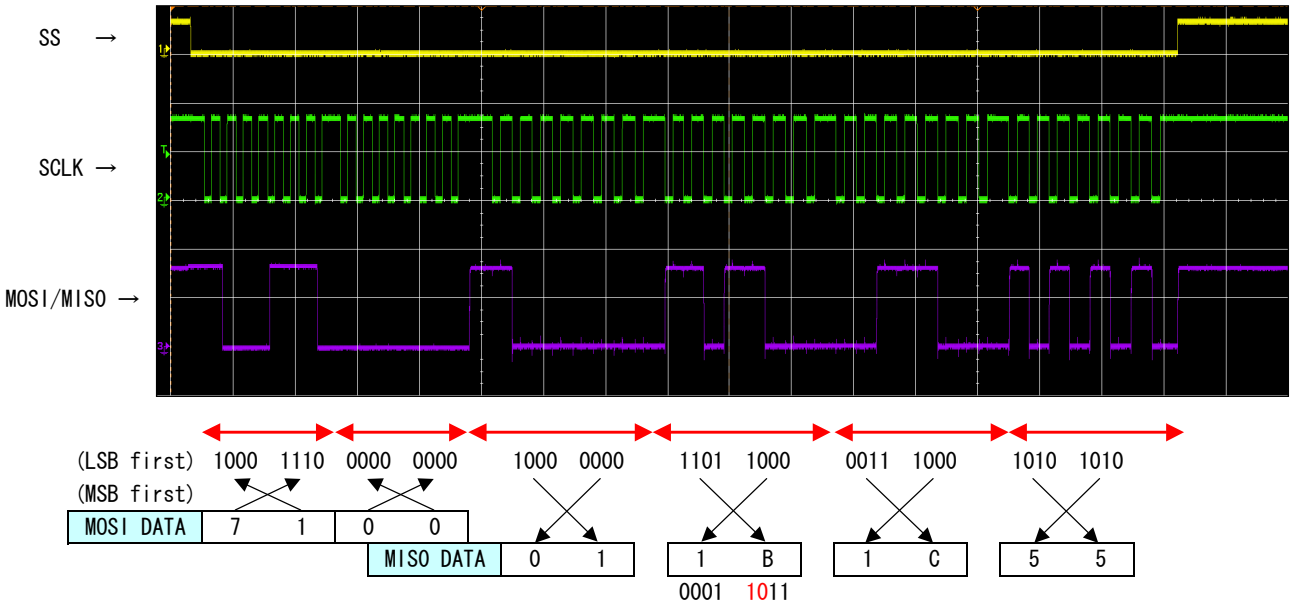


In this example, "ST CODE" is still "0xAA". It means that CAMERA MCU is busy.
Therefore, after about 1(mS) waiting, It needs retrying.



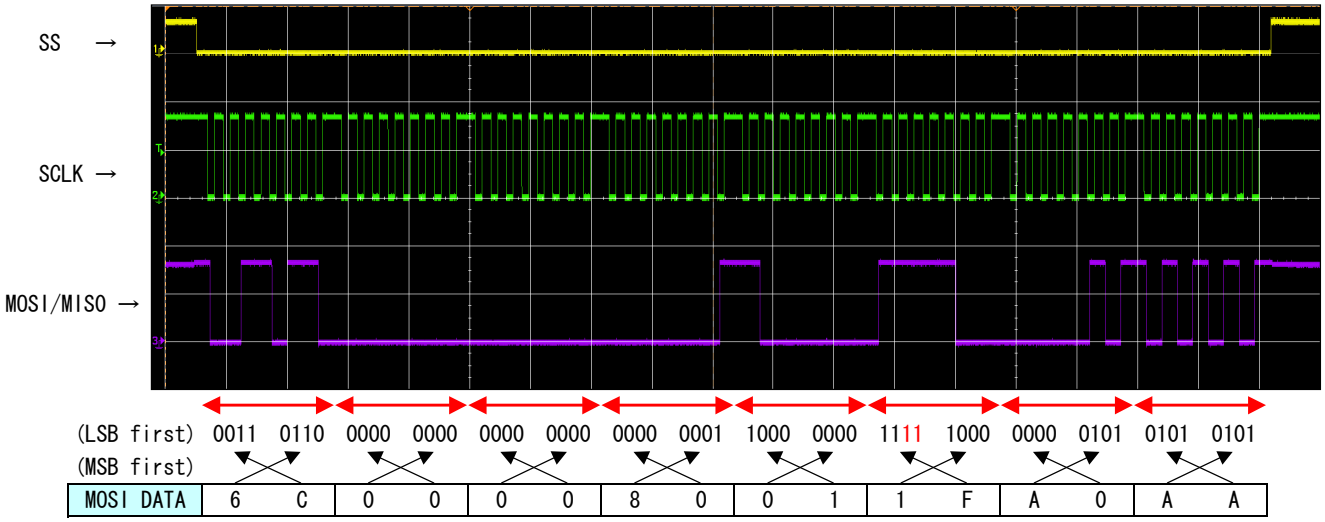
← Retry and get ST CODE = EXEC.NORMALLY.

Step *3: By using "SPI read command", get "AGC MODE: Address=0x401" parameter value from CAMERA.

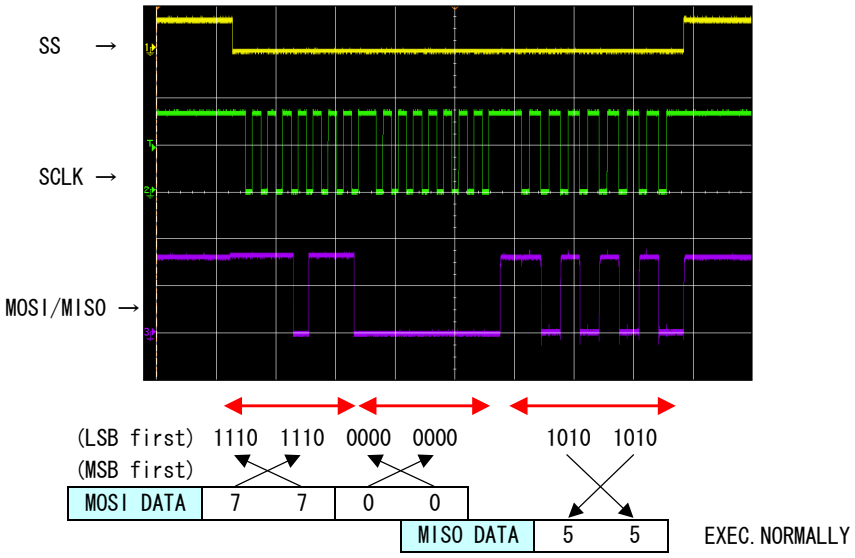


"AGC MODE: Address=0x401, Bit 4-3" is 2. It means "AGC MODE = MID".
To change AGC MODE = HI, set 1 to Bit4 and 1 to Bit 3.
Therefore, prepare data (0x1F=00011111) and send to CAMERA.

Step *4: Send "CAMERA PARAMETER WRITE1 command" to indirect registers(CAMERA: Address 0x36 - 0x3B) to set "AGC MODE: Address=0x401" parameter to CAMERA.
Following wave form is above operation by using "SPI write command".



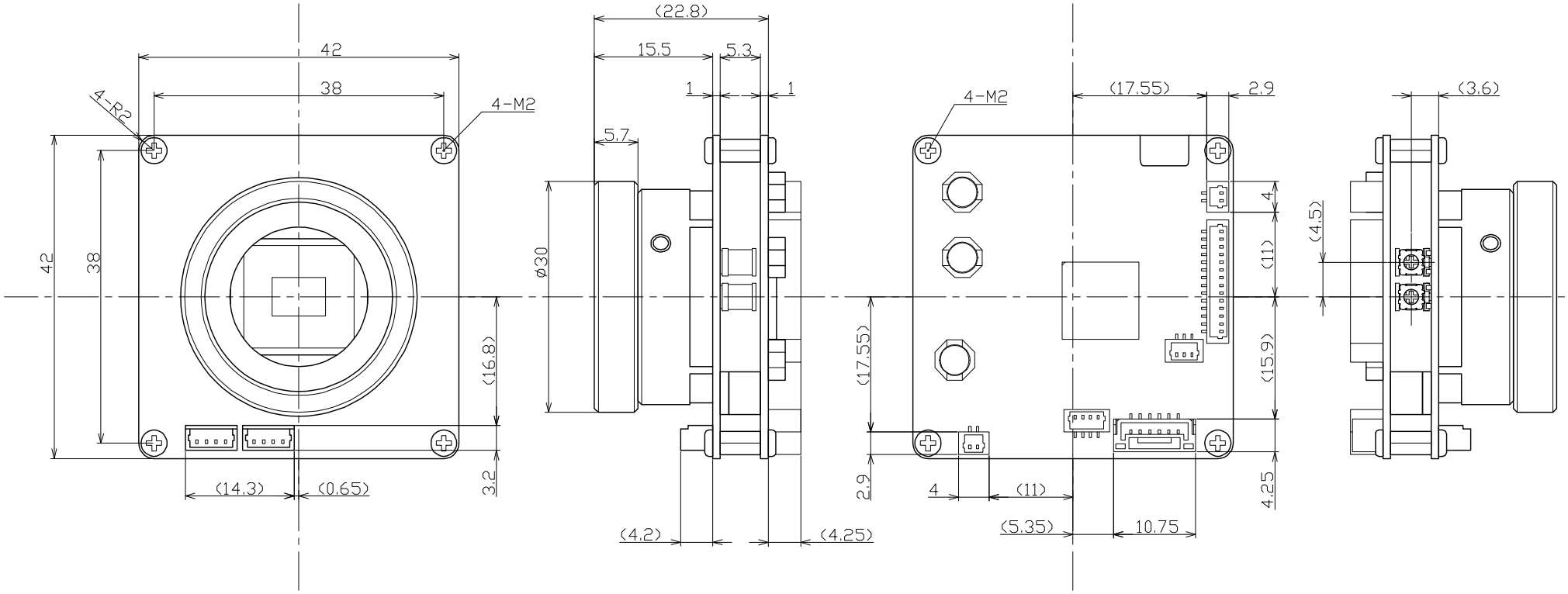
Step *5: Send "SPI read command" to get and check ST CODE(address:0x3B).



7. SPECIFICATIONS

Model		WAT-910BD (EIA)	WAT-910BD (CCIR)
Pick-up element		1/2 inch interline transfer CCD image sensor	
Number of total pixels		811 (H) × 508 (V)	795 (H) × 596 (V)
Number of effective pixels		768 (H) × 494 (V)	752 (H) × 582 (V)
Unit cell size		8.4um (H) × 9.8um (V)	8.6um (H) × 8.3um (V)
Synchronizing system		Internal	
Scanning system		2:1 interlace	
Video output		Composite Video, 1.0 V(p-p), 75Ω, unbalanced	
Resolution (Horizontal)		More than 570TVL (Center)	
Minimum illumination		0.0000025lx. F1.4 (Shutter: x256, AGC: HIGH, γ : 0.35, 3DNR ON)	
S/N		More than 52dB (AGC=6dB, γ=1.0)	
Function settings		OSD (On Screen Display) operated by 5key interface	
Shutter	fix speed	x256, x128, x64, x32, x16, x8, x4, x2(field) OFF:1/60(EIA),1/50(CCIR), FL:1/100(EIA),1/120(CCIR), 1/250, 1/500, 1/1000, 1/2000, 1/5000, 1/10000, 1/100000	
	Electronic IRIS	1/60(EIA),1/50(CCIR) - 1/100000(sec), x256(field:default x8) - 1/100000(sec)	
AGC	ON	LO:6-30dB / MID:6-34.5dB / HI:6-41dB	
	OFF	6-41dB (1dB step) selectable	
Gamma characteristics		0.05 - 1.0 (0.05 step) / USR (S-Curve) (default 0.45)	
Noise reduction		3DNR ON / OFF (default ON:50)	
Wide dynamic range		D-WDR (Digital Wide Dynamic Range) OFF / USER1 / USER2	
Motion detection		ON / OFF Max. 4 areas selectable (Out-put terminal available)	
Remote CTL		3Wired SPI interface	
Digital output		ON / OFF BT.656 compatible 8bit digital output with H/V sync. (3.3V CMOS) Pixel clock: 28.6363MHz(EIA),28.375MHz(CCIR)	
Back light compensation		OFF / BLC / HSBLC	
White blemish correction		Up to 64 dots correctable by OSD operation	
Sharpness		32levels selectable(default 4)	
Setup-Level		0IRE / 7.5IRE	0IRE
Freeze		ON / OFF	
Flip		OFF / H-FLIP / V-FLIP / HV-FLIP	
Zoom		ON / OFF ZOOM magnification selectable(x1.0～x32.0)	
Neg.Image		ON / OFF	
Lens mount		CS	
Lens iris		Video / DC	
Power supply		DC+12V±10%	
Power consumption		1.32W (110mA)	
Operating temperature		- 10℃ ～ + 50℃	
Storage temperature		- 30℃ ～ + 70℃	
Operating/Storage humidity		Less than 95% RH (Without condensation)	
Weight		Approx. 40g	

8. DIMENSIONS



Unit:(mm)