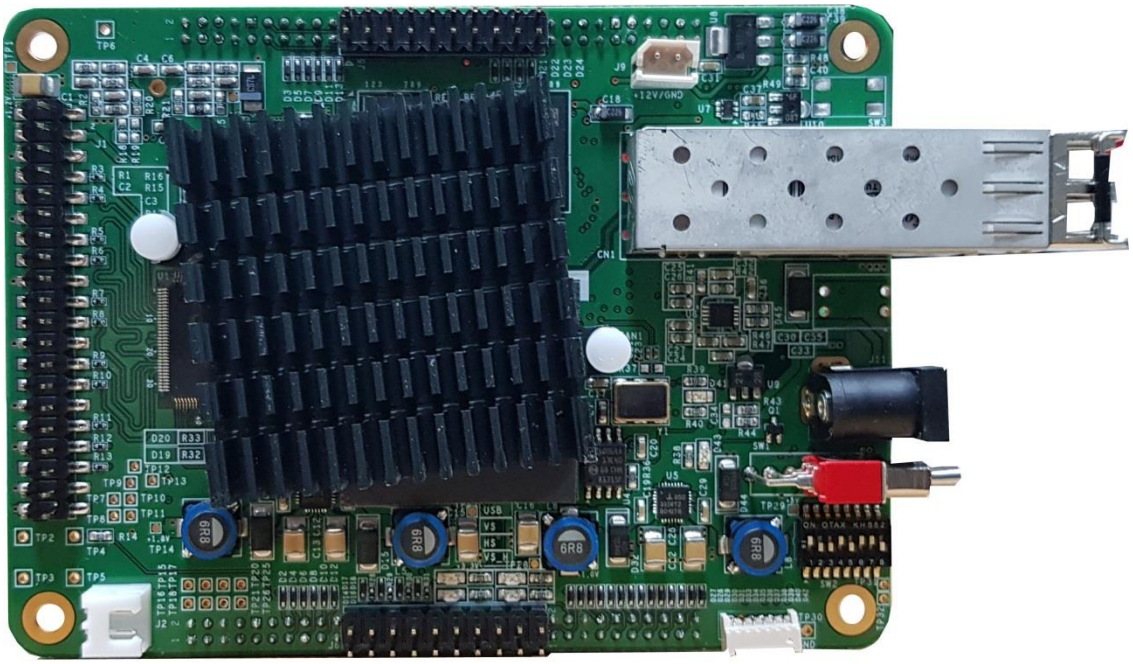
# MIPI-OPT06

## User Manual

**Ver. 1.1**





DAQ SYSTEM
Measurement & Automation

# Contents


## 1. Summary

1-1. Product Specification ---------- 2
1-2. Application Filed ---------- 3

## 2. MIPI-OPT06 Board Description

2-1. Board Layout ---------- 4
2-2. Board Description ---------- 5
2-3. I/O Terminal Pin Map ---------- 6
(1) CN1 Connector ---------- 6
(2) J1 Connector ---------- 6
(3) J3 Connector ---------- 9
(4) J4 Connector ---------- 10
(5) J5 Connector ---------- 11
(6) J6 Connector ---------- 12
(7) J7 Connector ---------- 13
(8) J8 Connector ---------- 14
(9) J11 Power Connector ---------- 15
(10) SW1 Switch ---------- 15
(11) SW2 Switch ---------- 16

## 3. Sample Program ---------- 17

3-1. Board Function ---------- 18
3-2. Image Function ---------- 19
3-3. Clock Function ---------- 22
3-4. Power/Digital/IO Function ---------- 23
3-5. I2C Function ---------- 24
3-6. Auto Test Function ---------- 25
3-7. SPI Function ---------- 25
3-8. OS SPI Function ---------- 26
3-9. MISC Function ---------- 26
3-10. STATUS Function ---------- 27

## Appendix

A-1. Board Size ---------- 28
A-2. Repair Regulations ---------- 29

# 1. Summary

## 1-1. Product Specification

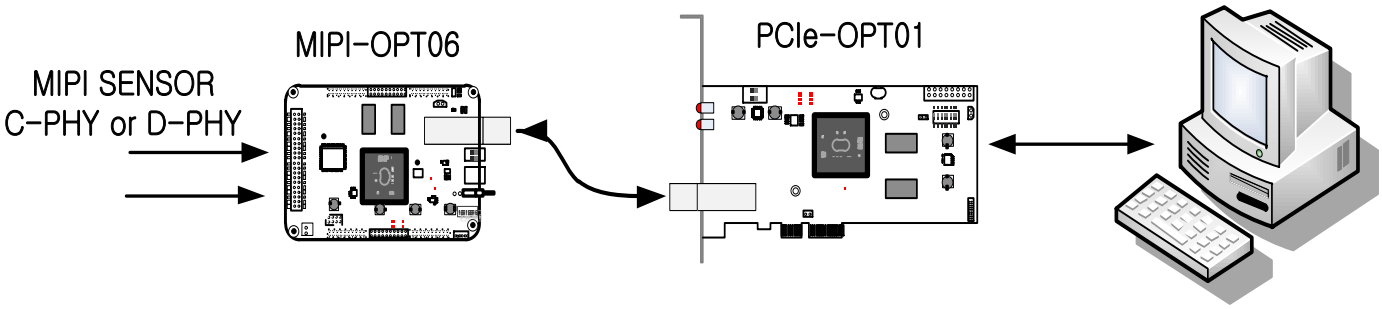
| Item | Description | Remark |
|---|---|---|
| **Hardware** | | |
| PC Interface | Fiber-Transmission Transceiver | SFP(small Form Factor) |
| Operation Power | +12VDC/400mA | External 12V DC Power<br>(A6-Type : 5.5x2.1mm) |
| Video Interface | MIPI CSI C-PHY 3 Lane<br>MIPI CSI D-PHY 4 Lane | 2.2 ~ 2.3 Gsym / 1 Lane<br>2.5Gbps / 1 Lane |
| Input/Output No. | Sensor GPIO 4bit<br>General GPIO 12bit<br>User GPIO 12bit | <br><br>3.3V GPIO 4bit, User GPIO 8bit |
| On-board Memory | 2Gb (DDR3) x2 | |
| Communication | I2C/SPI | |
| Number of boards used simultaneously | Max. 4 | |
| **Software** | | |
| OS | Windows 2000/XP/7/8/10 (32/64bit) | Use with PCIe-OPT01 |
| API | | |
| Development | | |
| Support | | |
| **Environmental conditions** | | |
| Operating temperature range | 0 ~ 50℃ (0 ~ 60℃ when using fan) | |
| Storage temperature range | -20 ~ 80℃ | |
| Humidity range | Non-condensing | |
| Board size | 110mm x 80mm<br>140mm x 80mm | PCB Board Size<br>Size including Case/SFP |

## 1-2. Application Field

- Frame Grabber
- Test for Variable MIPI Sensor

The MIPI-OPT06 board transmits the C-PHY or D-PHY MIPI (Mobile Industry Processor Interface) signal of the sensor board to the PC in an optical method. Two signals of C-PHY or D-PHY can be selected and used. The received signal is processed by the software (application) provided by the DAC system in the PC and displayed as an image.

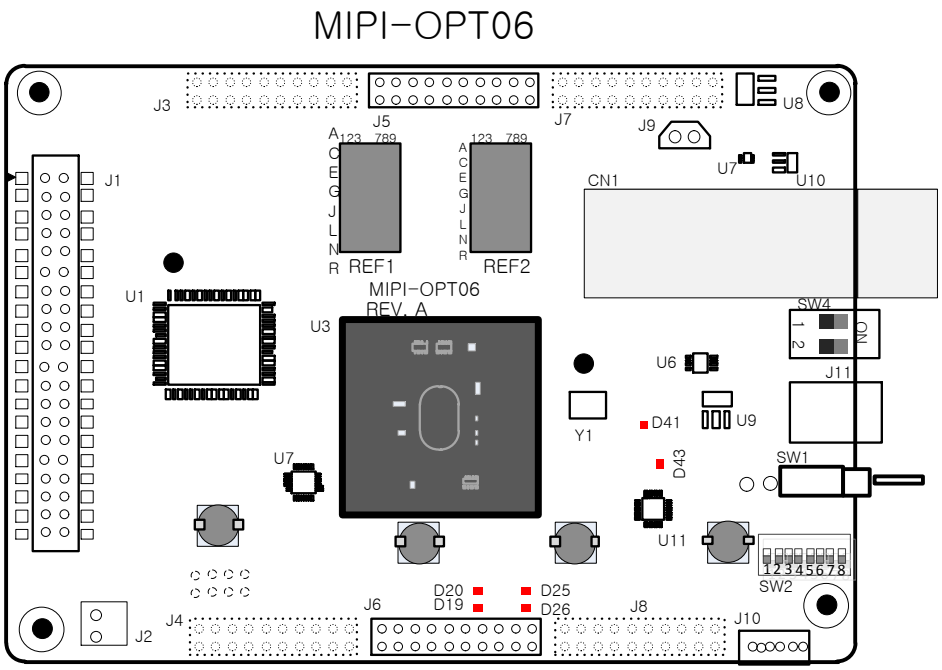
[Figure 1-1] shows an example of using the input board.




[Figure 1-1. MIPI-OPT06 Example of Use]

# 2. MIPI-OPT06 Board Description

## 2-1. Board Layout




[Figure 2-1. MIPI-OPT06 Layout]

| No. | Name | Description/Remark |
|---|---|---|
| **1** | CN1 | SFP (8Gbps) |
| **2** | U8, U9 | 1.2V, 2.5V, 3.3V |
| **3** | U1 | MIPI C-PHY/D-PHY Combo Chip |
| **4** | U3 | FPGA |
| **5** | REF1/REF2 | Image Buffer (DDR3 256MB) |
| **6** | J1 | MIPI Board Connection Connector (For MIPI Signals) |
| **7** | J3~ J8 | General Purpose IO or PWR/OS/AD Communication |
| **8** | J11 | Power Adaptor (12V) |
| **9** | SW1 | Power Switch |

There are 6 important LEDs on the board, and the description of each is as follows.

**D20**: Lights up after completion of initialization.

**D19**: Vertical Synchronization signal line (Vsync) is connected.
This is a mark for visually checking Vsync.

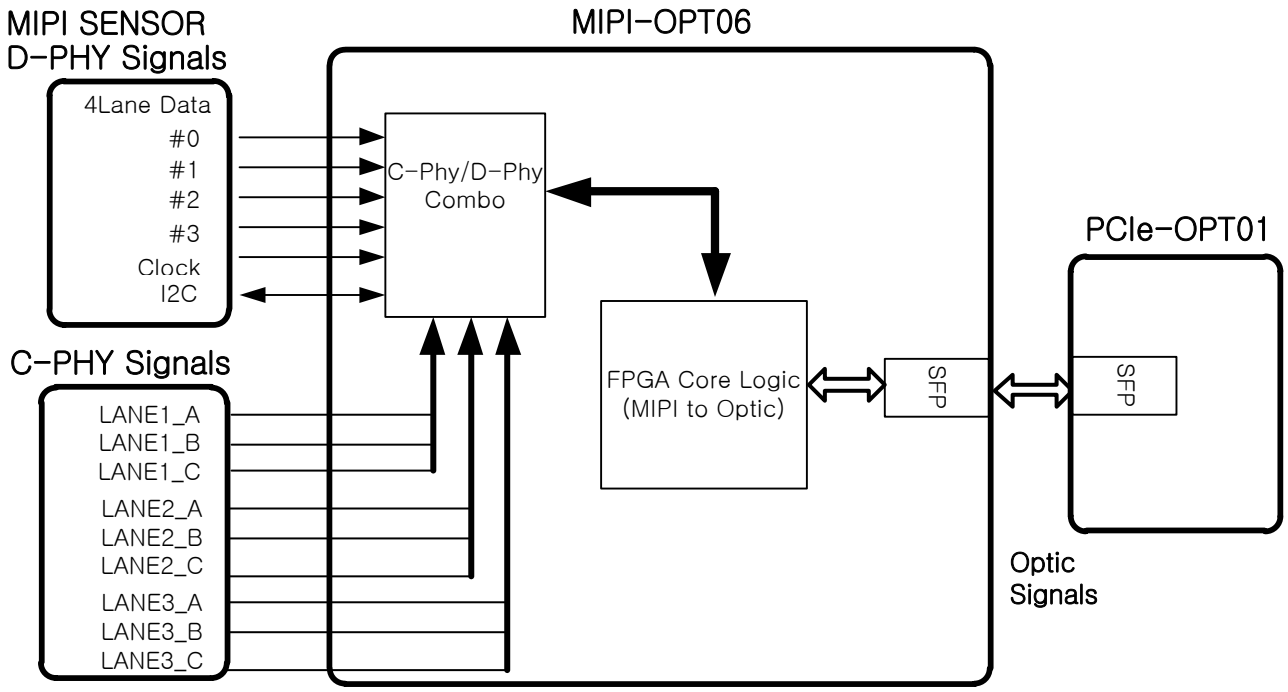
**D25**: Horizontal Synchronization signal line (Hsync) is connected.
This is a mark to visually check Hsync.

**D26**: Displays the vertical synchronization (Vsync) divided by 1/16.
This is a mark for visually checking Vsync.
**D41**: Lights when the board is configured and ready for operation.
**D43**: Lights when 3.3V power is applied.

## 2-2. Board Description

The MIPI-OPT06 board converts MIPI D-PHY or C-PHY signals received from MIPI sensors and transmits light. All functions are controlled by the FPGA.

The block diagram of MIPI-OPT06 is as shown in [Figure 2-2].




[Figure 2-2. MIPI-OPT06 Block Diagram]

The program of the FPGA core logic uses JTAG, and functions to save the logic program in the FPGA Program Logic and download it when power is applied..

## 2-3. I/O Terminal Pin map

### (1) CN1 Connector

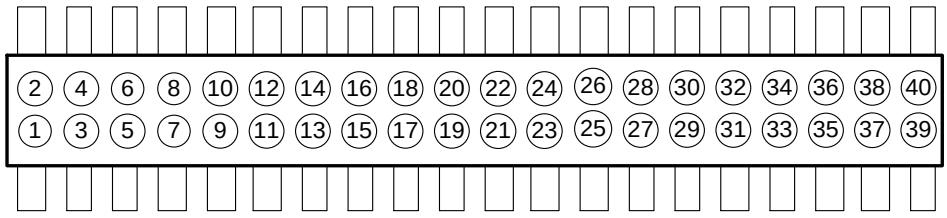
In the case of MIPI-OPT06, a Small Form Factor Pluggable (SFP) connector is used as a Fiber-Transmission Transceiver. The SFP transceiver is designed to support various optical transmissions such as SONET, Gigabit Ethernet, and Fiber Channel. It supports hot-pluggable transceiver and can be connected to network device motherboard with fiber or copper networking cable.

The SFP is connected by a module that connects to the cage and connector, and has Tx (Transceiver) and Rx (Receiver) together.

[Figure 2-3. SFP & SFP Cage]

### (2) J1 Connector (2x20 Pin Straight Male SMD Type)

It is connected to the MIPI SENSOR board and the signals are as follows.




[Figure 2-4. J1 Connector (Top View)]

[Table 1. J1 Connector Description – **When used as C-Phy**]

| No. | Name | Description | Remark |
|---|---|---|---|
| 1 | **SP0** | SENSOR Power | |

| | | | |
|---|---|---|---|
| 2 | **SP1** | SENSOR Power | |
| 3 | **SP2** | SENSOR Power | |
| 4 | **SP3** | SENSOR Power | |
| 5 | **GND** | Ground | |
| 6 | **GND** | Ground | |
| 7 | **SCL** | Serial Clock | |
| 8 | **C_L0_RXA** | C_lane0_RX_A | C : C-Phy, D : D-Phy |
| 9 | **SDA** | Serial Data | |
| 10 | **C_L0_RXB** | C_lane0_RX_B | C : C-Phy, D : D-Phy |
| 11 | **GND** | Ground | |
| 12 | **GND** | Ground | |
| 13 | **ENB** | Enable | |
| 14 | **C_L0_RXC** | C_lane0_RX_C | C : C-Phy, D : D-Phy |
| 15 | **S_RESET** | Reset | |
| 16 | **-** | - | C : C-Phy, D : D-Phy |
| 17 | **GND** | Ground | |
| 18 | **GND** | Ground | |
| 19 | **S_GPIO0** | Sensor GPIO0 | Using the PCI_SDIO_xxx function |
| 20 | **C_L2_RXA** | C_lane2_RX_A | C : C-Phy, D : D-Phy |
| 21 | **S_GPIO1** | Sensor GPIO1 | Using the PCI_SDIO_xxx function |
| 22 | **C_L2_RXB** | C_lane2_RX_B | C : C-Phy, D : D-Phy |
| 23 | **GND** | Ground | |
| 24 | **GND** | Ground | |
| 25 | **S_GPIO2** | Sensor GPIO2 | Using the PCI_SDIO_xxx function |
| 26 | **C_L2_RXC** | C_lane2_RX_C | C : C-Phy, D : D-Phy |
| 27 | **S_GPIO3** | Sensor GPIO3 | Using the PCI_SDIO_xxx function |
| 28 | **-** | - | C : C-Phy, D : D-Phy |
| 29 | **GND** | Ground | |
| 30 | **GND** | Ground | |
| 31 | **GND** | Ground | |
| 32 | **C_L1_RXA** | C_lane1_RX_A | C : C-Phy, D : D-Phy |
| 33 | **GND** | Ground | |
| 34 | **C_L1_RXB** | C_lane1_RX_B | C : C-Phy, D : D-Phy |
| 35 | **MCLK** | Master Clock | |
| 36 | **C_L1_RXC** | C_lane1_RX_C | C : C-Phy, |
| 37 | **GND** | Ground | |
| 38 | **GND** | Ground | |

| | | | |
|---|---|---|---|
| 39 | **SP4** | SENSOR Power | |
| 40 | **SP5** | SENSOR Power | |

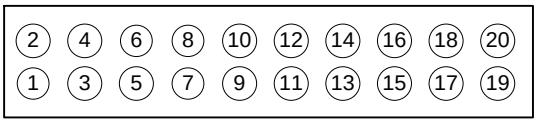
[Table 2. J1 Connector Description – **When used as D-Phy**]

| No. | Name | Description | Remark |
|---|---|---|---|
| 1 | **SP0** | SENSOR Power | |
| 2 | **SP1** | SENSOR Power | |
| 3 | **SP2** | SENSOR Power | |
| 4 | **SP3** | SENSOR Power | |
| 5 | **GND** | Ground | |
| 6 | **GND** | Ground | |
| 7 | **SCL** | Serial Clock | |
| 8 | **D_L0_RXP** | D_lane0_RX_P | C : C-Phy, D : D-Phy |
| 9 | **SDA** | Serial Data | |
| 10 | **D_L0_RXN** | D_lane0_RX_N | C : C-Phy, D : D-Phy |
| 11 | **GND** | Ground | |
| 12 | **GND** | Ground | |
| 13 | **ENB** | Enable | |
| 14 | **D_L1_RXP** | D_lane1_RX_P | C : C-Phy, D : D-Phy |
| 15 | **S_RESET** | Reset | |
| 16 | **D_L1_RXN** | D_lane1_RX_N | C : C-Phy, D : D-Phy |
| 17 | **GND** | Ground | |
| 18 | **GND** | Ground | |
| 19 | **S_GPIO0** | Sensor GPIO0 | Using the PCI_SDIO_xxx function |
| 20 | **D_L2_RXP** | D_lane2_RX_P | C : C-Phy, D : D-Phy |
| 21 | **S_GPIO1** | Sensor GPIO1 | Using the PCI_SDIO_xxx function |
| 22 | **D_L2_RXN** | D_lane2_RX_N | C : C-Phy, D : D-Phy |
| 23 | **GND** | Ground | |
| 24 | **GND** | Ground | |
| 25 | **S_GPIO2** | Sensor GPIO2 | Using the PCI_SDIO_xxx function |
| 26 | **D_L3_RXP** | D_lane3_RX_P | C : C-Phy, D : D-Phy |
| 27 | **S_GPIO3** | Sensor GPIO3 | Using the PCI_SDIO_xxx function |
| 28 | **D_L3_RXN** | D_lane3_RX_N | C : C-Phy, D : D-Phy |
| 29 | **GND** | Ground | |
| 30 | **GND** | Ground | |
| 31 | **GND** | Ground | |

| | | | |
|---|---|---|---|
| 32 | **D_CLK_RXP** | D_CLK_RX_P | C : C-Phy, D : D-Phy |
| 33 | **GND** | Ground | |
| 34 | **D_CLK_RXN** | D_CLK_RX_N | C : C-Phy, D : D-Phy |
| 35 | **MCLK** | Master Clock | |
| 36 | **-** | - | C : C-Phy, |
| 37 | **GND** | Ground | |
| 38 | **GND** | Ground | |
| 39 | **SP4** | SENSOR Power | |
| 40 | **SP5** | SENSOR Power | |

**(3) J3 Connector**

It is connected to the MIPI power board and is a connector related to sensor power and some communication.




[Figure 2-5. J3 Connector (Top View)]

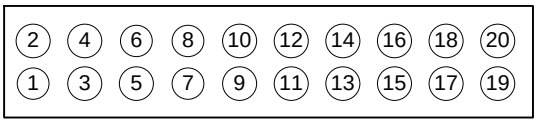
[Table 3. J3 Connector Description]

| No. | Name | Description | Remark |
|---|---|---|---|
| 1 | **GND** | Ground | |
| 2 | **GND** | Ground | |
| 3 | **SP0** | SENSOR Power | 0.9 ~ 4.1V |
| 4 | **SP1** | SENSOR Power | 0.9 ~ 4.1V |
| 5 | **SP2** | SENSOR Power | 0.9 ~ 4.1V |
| 6 | **SP3** | SENSOR Power | 0.9 ~ 4.1V |
| 7 | **GND** | Ground | |
| 8 | **GND** | Ground | |
| 9 | **SCL** | Serial Clock | 3.3V |
| 10 | **SDA** | Serial Data | 3.3V |
| 11 | **PWR_SCL** | Power Serial Clock | 3.3V |
| 12 | **PWR_SDA** | Power Serial Data | 3.3V |
| 13 | **AD_SCL** | AD Converter Serial Clock | 3.3V |
| 14 | **AD_SDA** | AD Converter Serial Data | 3.3V |
| 15 | **GND** | Ground | |
| 16 | **GND** | Ground | |
| 17 | **OS_SCL** | Open/Short Serial Clock | 3.3V |

| | | | |
|---|---|---|---|
| 18 | **OS_SDA** | Open/Short Serial Data | 3.3V |
| 19 | **GND** | Ground | |
| 20 | **GND** | Ground | |

### (4) J4 Connector

It is connected to the MIPI power board, and is a connector related to sensor power, SPI communication, and some GPIO (General Purpose In/Out).




[Figure 2-6. J4 Connector (Top View)]

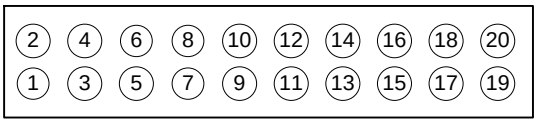
[Table 4. J4 Connector Description]

| No. | Name | Description | Remark |
|---|---|---|---|
| 1 | **GND** | Ground | |
| 2 | **GND** | Ground | |
| 3 | **SP4** | User Defined Power | 0.9 ~ 4.1V |
| 4 | **SP5** | User Defined Power | 0.9 ~ 4.1V |
| 5 | **GND** | Ground | |
| 6 | **GND** | Ground | |
| 7 | **SPI_SCK** | Serial Clock | 3.3V |
| 8 | **SPI_MOSI** | Master Output, Slave Input | 3.3V |
| 9 | **SPI_EN** | Slave Select | 3.3V |
| 10 | **SPI_MISO** | Master Input, Slave Output | 3.3V |
| 11 | **5V_IN** | 5V Power | |
| 12 | **V_IO2** | IO2 Power | |
| 13 | **GPIO0** | General Purpose IO0 | Using the PCI_PWR_DIO_xxx function |
| 14 | **GPIO1** | General Purpose IO1 | Using the PCI_PWR_DIO_xxx function |
| 15 | **GPIO2** | General Purpose IO2 | Using the PCI_PWR_DIO_xxx function |
| 16 | **GPIO3** | General Purpose IO3 | Using the PCI_PWR_DIO_xxx function |
| 17 | **GPIO4** | General Purpose IO4 | Using the PCI_PWR_DIO_xxx function |
| 18 | **GPIO5** | General Purpose IO5 | Using the PCI_PWR_DIO_xxx |

| | | | function |
|---|---|---|---|
| 19 | **GND** | Ground | |
| 20 | **GND** | Ground | |

**(5) J5 Connector**

It is a connector used for Power/AD Convertor/OpenShort communication line and some GPIOs.



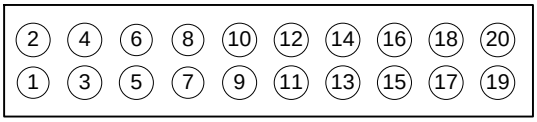

[Figure 2-7. J5 Connector (Top View)]

[Table 5. J5 Connector Description]

| No. | Name | Description | Remark |
|---|---|---|---|
| 1 | **GND** | Ground | |
| 2 | **GND** | Ground | |
| 3 | **SCL** | Serial Clock | 3.3V |
| 4 | **SDA** | Serial Data | 3.3V |
| 5 | **PWR_SCL_FU** | Power Serial Clock | 3.3V |
| 6 | **PWR_SDA_FU** | Power Serial Data | 3.3V |
| 7 | **AD_SCL_FU** | AD Converter Serial Clock | 3.3V |
| 8 | **AD_SDA_FU** | AD Converter Serial Data | 3.3V |
| 9 | **OS_SCL_FU** | Open/Short Serial Clock | 3.3V |
| 10 | **OS_SDA_FU** | Open/Short Serial Data | 3.3V |
| 11 | **GND** | Ground | |
| 12 | **GND** | Ground | |
| 13 | **VCC12_WALL** | 12V Adaptor Power | 12V |
| 14 | **VCC12_WALL** | 12V Adaptor Power | 12V |
| 15 | **GPIO8** | General Purpose IO8 | 1.5 ~ 3.3V |
| 16 | **GPIO9** | General Purpose IO9 | 1.5 ~ 3.3V |
| 17 | **GPIO10** | General Purpose IO10 | 1.5 ~ 3.3V |
| 18 | **GPIO11** | General Purpose IO11 | 1.5 ~ 3.3V |
| 19 | **GND** | Ground | |
| 20 | **GND** | Ground | |

### (6) J6 Connector

This connector is used for SPI (Serial Peripheral Interface) and some GPIOs.



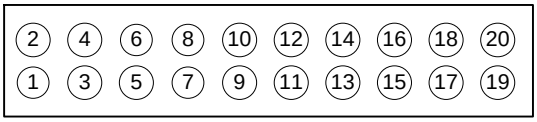

[Figure 2-8. J6 Connector (Top View)]

[Table 6. J6 Connector Description]

| No. | Name | Description | Remark |
|---|---|---|---|
| 1 | **GND** | Ground | |
| 2 | **GND** | Ground | |
| 3 | **SPI_CLK_CON** | Serial Clock | 3.3V |
| 4 | **SPI_MOSI_CON** | Master Output, Slave Input | 3.3V |
| 5 | **SPI_EN_CON** | Slave Select | 3.3V |
| 6 | **SPI_MISO_CON** | Master Input, Slave Output | 3.3V |
| 7 | **GND** | Ground | |
| 8 | **5V_IN** | 5V Power | 5V |
| 9 | **V_IO2** | IO2 Power | V_IO2 Power |
| 10 | **GND** | Ground | |
| 11 | **GPIO0** | General Purpose IO0 | 1.5 ~ 3.3V |
| 12 | **GPIO1** | General Purpose IO1 | 1.5 ~ 3.3V |
| 13 | **GPIO2** | General Purpose IO2 | 1.5 ~ 3.3V |
| 14 | **GPIO3** | General Purpose IO3 | 1.5 ~ 3.3V |
| 15 | **GPIO4** | General Purpose IO4 | 1.5 ~ 3.3V |
| 16 | **GPIO5** | General Purpose IO5 | 1.5 ~ 3.3V |
| 17 | **GPIO6** | General Purpose IO6 | 1.5 ~ 3.3V |
| 18 | **GPIO7** | General Purpose IO7 | 1.5 ~ 3.3V |
| 19 | **GND** | Ground | |
| 20 | **GND** | Ground | |

### (7) J7 Connector

This connector is connected to the MIPI power board.



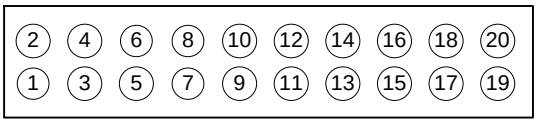

[Figute 2-9. J7 Connector (Top View)]

[Table 7. J7 Connector Description]

| No. | Name | Description | Remark |
|---|---|---|---|
| 1 | **GND** | Ground | |
| 2 | **GND** | Ground | |
| 3 | **GPIO8** | General Purpose IO8 | Using the PCI_PWR_DIO_xxx function |
| 4 | **GPIO9** | General Purpose IO9 | Using the PCI_PWR_DIO_xxx function |
| 5 | **GPIO10** | General Purpose IO10 | Using the PCI_PWR_DIO_xxx function |
| 6 | **GPIO11** | General Purpose IO11 | Using the PCI_PWR_DIO_xxx function |
| 7 | **GND** | Ground | |
| 8 | **GND** | Ground | |
| 9 | **5V_IN** | 5V Power | 5V |
| 10 | **5V_IN** | 5V Power | 5V |
| 11 | **5V_IN** | 5V Power | 5V |
| 12 | **5V_IN** | 5V Power | 5V |
| 13 | **GND** | Ground | |
| 14 | **GND** | Ground | |
| 15 | **VCC12_WALL** | 12V Adaptor Power | 12V |
| 16 | **VCC12_WALL** | 12V Adaptor Power | 12V |
| 17 | **VCC12_WALL** | 12V Adaptor Power | 12V |
| 18 | **VCC12_WALL** | 12V Adaptor Power | 12V |
| 19 | **GND** | Ground | |
| 20 | **GND** | Ground | |

### (8) J8 Connector

It is connected to the MIPI power board and is a connector related to GPIO (General Purpose In/Out).




[Figure 2-10. J8 Connector (Top View)]

[Table 8. J8 Connector Description]

| No. | Name | Description | Remark |
|---|---|---|---|
| 1 | **GND** | Ground | |
| 2 | **GND** | Ground | |
| 3 | **GPIO6** | General Purpose IO6 | Using the PCI_PWR_DIO_xxx function |
| 4 | **GPIO7** | General Purpose IO7 | Using the PCI_PWR_DIO_xxx function |
| 5 | **GND** | Ground | |
| 6 | **GND** | Ground | |
| 7 | **U3_GPIO0** | User General Purpose IO0 | Using the PCI_DIO33_xxx function |
| 8 | **U3_GPIO1** | User General Purpose IO1 | Using the PCI_DIO33_xxx function |
| 9 | **U3_GPIO2** | User General Purpose IO2 | Using the PCI_DIO33_xxx function |
| 10 | **U3_GPIO3** | User General Purpose IO3 | Using the PCI_DIO33_xxx function |
| 11 | **U_GPIO0** | User General Purpose IO0 | Using the PCI_DIO_xxx function |
| 12 | **U_GPIO1** | User General Purpose IO1 | Using the PCI_DIO_xxx function |
| 13 | **U_GPIO2** | User General Purpose IO2 | Using the PCI_DIO_xxx function |
| 14 | **U_GPIO3** | User General Purpose IO3 | Using the PCI_DIO_xxx function |
| 15 | **U_GPIO4** | User General Purpose IO4 | Using the PCI_DIO_xxx function |
| 16 | **U_GPIO5** | User General Purpose IO5 | Using the PCI_DIO_xxx function |
| 17 | **U_GPIO6** | User General Purpose IO6 | Using the PCI_DIO_xxx function |
| 18 | **U_GPIO7** | User General Purpose IO7 | Using the PCI_DIO_xxx function |
| 19 | **GND** | Ground | |
| 20 | **GND** | Ground | |

**(9) J11 Power Connector**

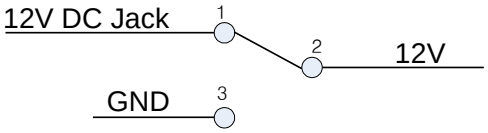
It is an external 12V DC Jack (A6 Type: 5.5x2.1mm) power connector of DC-005(2.0) standard.

**(Recommended for basic use)**


[Figure 2-11. Rated power]

**(10) SW1 Switch**

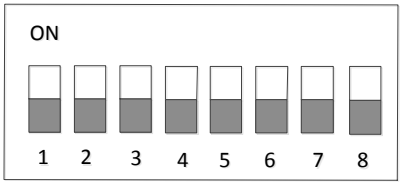
When the terminal is raised with the board power On/Off switch, the 12V power is turned on.




[Figure 2-12. SW1 Switch]

### (11) SW2 Switch

It is set at the factory as an optional function. (Customer use is prohibited)




[Figure 2-13. SW2 Switch (Top View)]

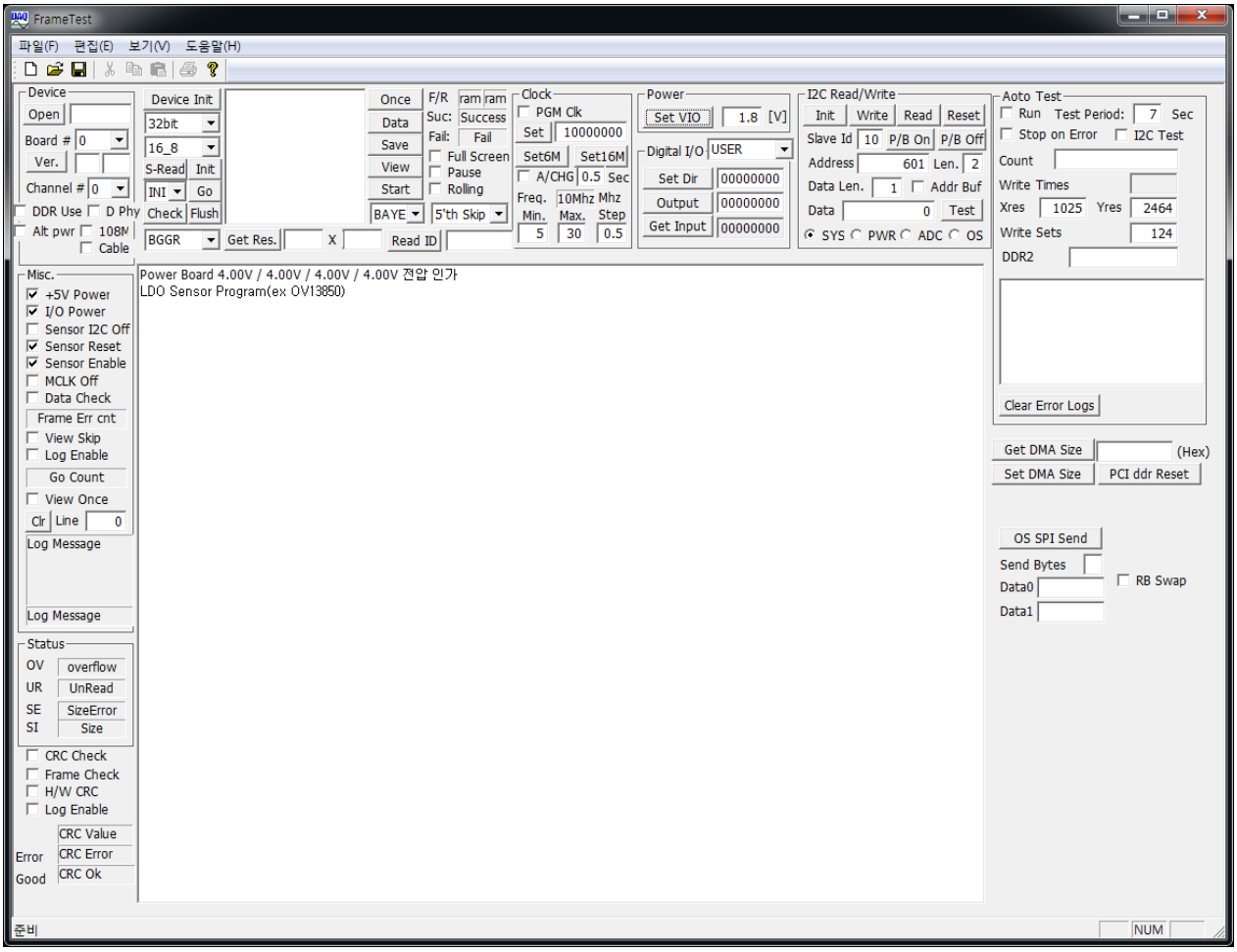
[Table 9. SW2 PIN-OUT Description]

| No. | Description |
|---|---|
| 1 | BIAS0 |
| 2 | BIAS1 |
| 3 | BIAS2 |
| 4 | BIAS3 |
| 5 | BIAS4 |
| 6 | BIAS5 |
| 7 | BIAS6 |
| 8 | BIAS7 |

# 3. Sample Program

In the Exe folder of the CDROM provided with the board, a sample program "FrameTest.exe" is provided for easy use of the board. This program runs on a PC equipped with a PCIe-OPT01 board. By displaying Frame Data as a hexadecimal value, it is stored in memory or hard disk so that developers can utilize the frame data needed. In order to test the sample program, the driver of the board must be installed first.

The sample program is provided in the form of a source so that the API provided to use the board can be tested briefly, so the user can modify it and use it.



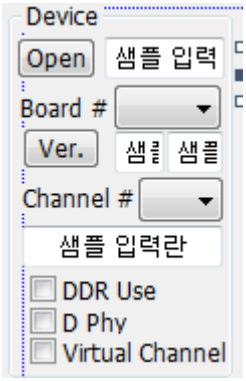

[Figure 3-1. Sample Program "FrmTest.exe"]

API (Application Programming Interface) is required to use the above sample program. API is provided in the form of "DLL", and import library and header file are required to compile. All files specified above are included on the supplied CDROM. In order to run the sample program normally, it must be in the API DLL (mipi_iot.dll) or in the Windows system folder or the folder specified by the Path environment variable.

The description of each menu bar is as follows. The menu bar not described here is an unused function.

**(Note) The sample program execution sequence is to select Board # and Channel # first, then click "Open" ➔ Select the format suitable for the camera in "Data mode" ➔ Select the camera data width Select from among "8, 16, 24, 32bit" ➔ "Device Init" ➔ Select the sensor *.ini file in S-Read (*.ini file address and data selection, in case of D-Phy, select D Phy) ➔ Go (Check "Get Res." resolution) ➔ Image is real-time show**

## 3.1 Board Function




**(1) "Device Open" button**

Starts the device of the selected board. If it is "0", it means that there is no device or no connection.

**(2) "Board # :"**

If the board is multi, the board number is assigned.

Currently, 4 can be selected from 0 to 3. Each board number is identified by a switch (J10) in the board..

**(3) " Ver." button**

Shows the current FPGA version and firmware version.

**(4) "Channel # :"**

Specifies the fiber channel number. If it is "0", it is connected to the lower optical channel (CN2), and if it is "1", it is connected to the upper optical channel (CN1).

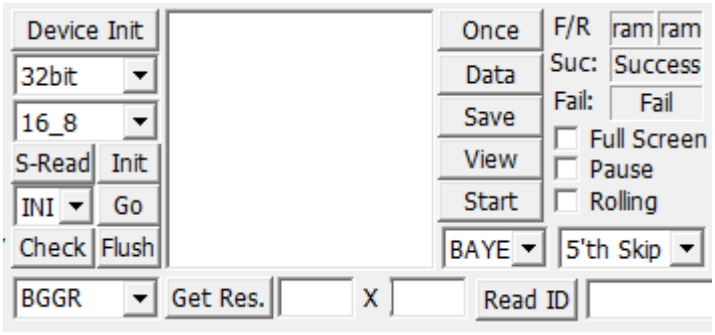
**(5) "Sample Input"** : Among MIPI-OPT06 or MIPI-OPT08, the name of the board connected to PCIe-OPT01 appears.

**DDR Use** : It uses DDR memory.

**D Phy** : When selected, set D-Phy as the LVDS input mode. The default setting is C-PHY.

**Virtual Channel** : MIPI Virtual Channel is used when selected.

## 3.2 Image Frame Function




(1) **"Device Init" button**

Initialize the image frame function. It is performed only once when the first power is applied. Select Video Data Mode from among 8bit, 16bit, 24bit, and 32bit.

(2) **"S-Read" button**

Read the sensor initialization file. Depending on the address_data size (16_8, 16_16) above, it is possible to send commands to the INI file at once, or use I2C read/write commands line by line. The structure and description of the ini file are as follows. The following example is a 1 byte structure of address_data size of 16_8 address 2 bytes data.

Ex) SONY13M_full.ini file structure

```
[REGISTER]
Slave    0x10   //change slave ID as Sensor

SLEEP    100
0x3087   0x53
0x309D   0x94
0x30A1   0x08
0x30AA   0x04
0x30B1   0x00
0x30C7   0x00
0x3115   0x0E
0x3118   0x42
0x311D   0x34
0x3121   0x0D
```

```
0x3212  0xF2
0x3213  0x0F

.....

0x3306  0x12
0x3307  0x03
0x3308  0x0D
0x3309  0x05
0x330A  0x09
0x330B  0x04
0x330C  0x08
0x330D  0x05
0x330E  0x03
0x3318  0x65
0x3348  0xE0
0x0100 0x01   //Streaming
```

(3) **"Init" button**

Initialize the sensor by selecting "SNI, T1, T2, SPI".

(4) **"Go" button**

Open the device, initialize it, open the corresponding ini file and get the resolution all at once

(5) **"Check" button**

Check the USB connection. In case of an error, "LVDS Check Error" is displayed.

(6) **"Flush" button**

Initialize the LVDS buffer.

(7) **"RGGB;BGGR;GRBG;BGGR" :** Choose from Bayer Mode

(8) **"Get Res." button**

Shows the image resolution.

(9) **"Once" button**

Press the Toggle button to display the screen once.

(10) **"Data" button**

It reads the image frame saved on the board to the PC (Hexa value). If the image frame is not saved on the board, you have to wait until the saving is completed.

(11) **"Save" button**

It is used to save the frame image data read into the PC as a binary file file.

(12) **"View" button**

Start sending images.

(13) **"Start" button**

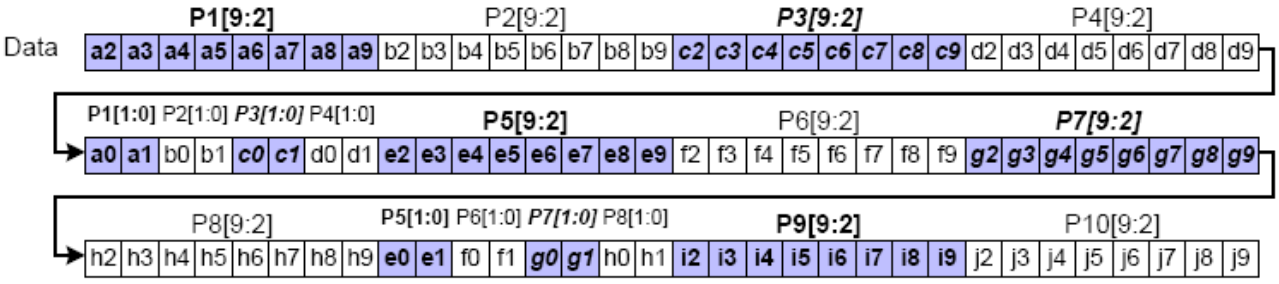
Start image transfer with "Start" and "stop" Toggle buttons.

(14) **"BAYER;RGB;YUV;USER" :** Select custom or image input format type

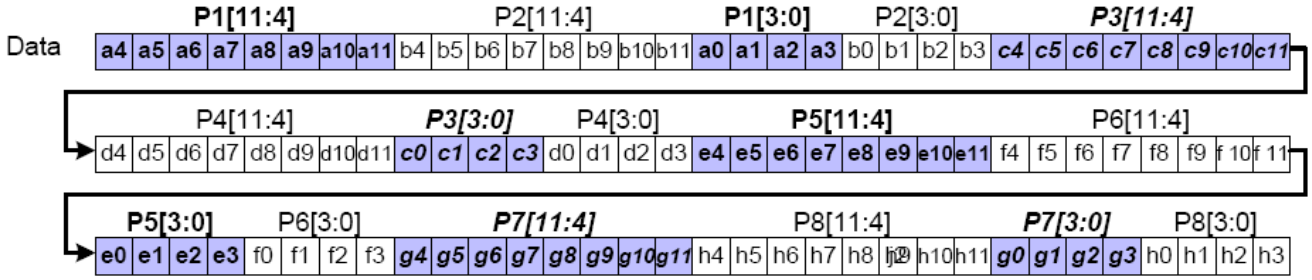
**"No Skip;** There is no byte skip.

**"5'th Skip" :** When input data is 10bit RAW data, skip the 5th byte on selection.

For example, if the input data is a 10-bit Bayer, 8 bits are stored in the 5th byte, each of which is 1 bit excluding 3 bytes of RGB and 1 byte. When the Bayer is processed and displayed on the screen, the 5th byte is not needed, so it is used to remove it.




**"3'th Skip" :** When input data is 12bit RAW data, **s**kip the 3th byte on selection.




(15) **"F/R : " : (Right)** Shows the number of frames displayed on the screen.

**(Left)** Shows the actual number of frames sent from the sensor.

**"Suc: " :** Shows the number of successful image transfers.

**"Fail: " :** Shows the number of failed image transfers.

**"Full Scr." :** Displays the screen in real resolution.

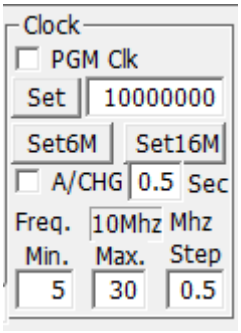
**"Pause" :** Pause the screen.

**"Rolling" :** Update image data without using GetFrame function.

(16) **"Read ID" button**

Shows MIPI ID.

## 3.3 Clock Function




(1) **"PGM Clk" toggle**

Select the corresponding Sensor Clock.

(2) **"Set" button**

It is set according to the frequency set next to the Sensor Clock. In the above case, it is set to 10MHz.

(3) **"Set 6M" button**

Set the Sensor Clock to 6MHz.

(4) **"Set 16M" button**

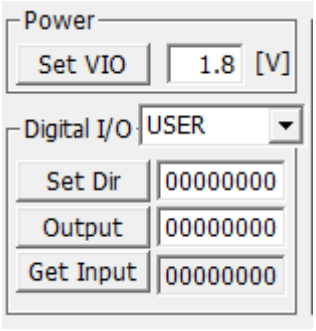
Set the Sensor Clock to 16MHz..

(5) **"A/CHG" toggle**

If you check, the Min. Max. It can be tested by periodically setting the interval of the frequency determined by Step.

Ex) In the above case, the frequency increases in units of 0.5MHz between 5 and 30MZ, and the period increases in units of 0.5 x 1000ms = 500ms as the number next to A/CHG.

## 3.4 Power/Digital/IO Function




(1) **"Set VIO" button**

User GPIO7..0 IO voltage value (default 1.8V) can be specified and set. (0 ~ 3.3V)

(2) **Digital I/O (USER;Sensor;User3.3V;PWR)** : Selection

(Based on Power Board connection)

**PCI_DIO_XXXX** : User GPIO7..0 General Purpose I/O)

**PCI_SDIO_XXXX** : Sensor GPIO3..0 (General Purpose I/O)

**PCI_DIO33_XXXX** : User 3.3V GPIO3.3 (General Purpose I/O)

**PCI_PWR_DIO_XXXX**: Power GPIO GPIO11..0 (General Purpose I/O)

(3) **"Set Dir" button**

Set whether to use each GPIO port as input or output.

If the last bit is "0", it is an input, and if it is "1", it is an output.

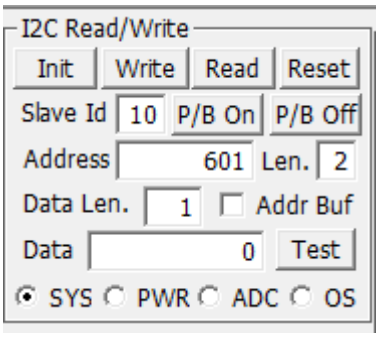
(4) **"Output" button**

According to the digital I/O selection, the value ("0" or "1") in the next box is output to the GPIO port.

(5) **"Get Input" button**

According to the digital I/O selection, the value of the GPIO input/output port is read.

## 3.5 I2C Function




(1) **"Init" button**

Initialize SYS/PWR/ADC/OS I2C communication speed. The initial speed is set to 100 KHz.

(2) **"Write" Button**

It writes the data of the corresponding address in the selected mode among the SYS, PWR, ADC, and OS modes below as much as the data size at the given address.

(3) **"Read" Button**

It reads the data of the corresponding address in the selected mode among the SYS, PWR, ADC, and OS modes below as much as the data size at the given address.

(4) **"Reset" Button**

Initializes the resources of the I2C system of the system (SYS) module.

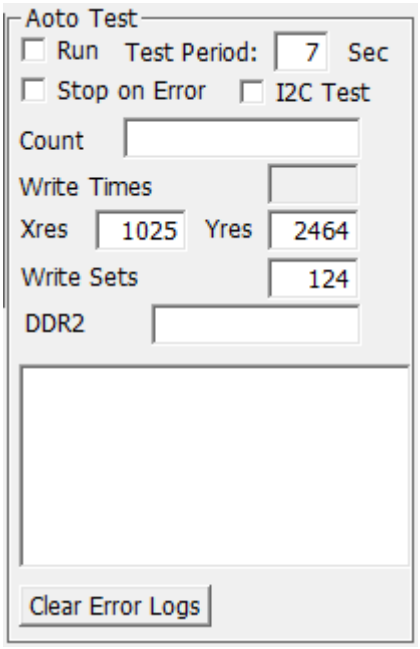
(5) **"Slave ID" :** Slave ID
**"Address " :** Slave Register Address
**"Len. :" :** Address Value
**"Data Len. :" :** Data Value
**"Addr Buf" toggle :** Used in SYS mode when selected, address instead of address use a buffer.
**"Data :" :** data you want to send
The above values are variables used for I2C Read or I2C Write.

(6) **"P/B On" or "P/B Off" button**

On/Off test the power 3.3V on the Power Board.

(7) **"SYS" : System Board**
**"PWR" : Power Board**
**"ADC" : AD Converter Board**
**"OS" : Open Short Board**

## 3.6 Auto Test Function




(1) **"RUN" toggle :** Device Open, Init, Clk Set, Sensor power test, etc. are repeatedly performed during the setting period of the Test Period according to the Write Times with the size of the Xres/Yres resolution given below.

**"Test Period" :** The test cycle can be set in seconds.

**"Stop on Err" :** Stop error output.

**"I2C Test" :** Test I2C of SYS, PWR, ADC board.

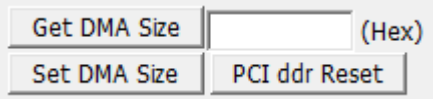
A related message appears on the log screen below, and in case of failure, a "Fail" message appears.

**Count** : AutoTest Count, Write error values and Reset Error values are output.

(2) **"Clear Error Log" Button**

Clear the error log.

## 3.7 SPI Function

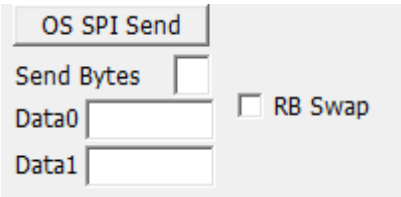



**"Get DMA Size" Button :** Get the DMA buffer size.

**"Set DMA Size" Button : S**et the DMA buffer size.

**"PCI ddr Reset" Button :** Initialize DDR memory.

## 3.8 OS SPI Function



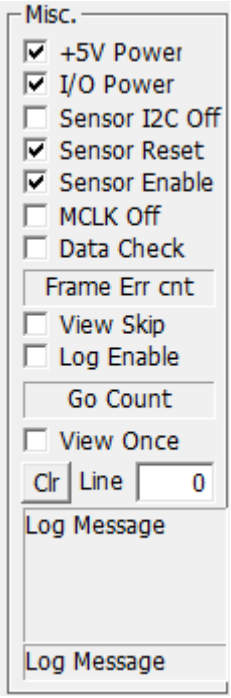

**"OS SPI Send" button :** Send data to the sensor.

**"Send Bytes" :** The size of bytes to send to the sensor

**"Data0" :** Data0 to send to sensor

**"Data1" :** Data1 to send to sensor

## 3.9 MISC Function




Several types of states can be selected and used.

**"+5V Power" :** Outputs VIO power.

**"I/O Power" :** Turns on/off the power of external signals used for MIPI.

**"Sensor I2C Off" :** Turns on/off the Sensor I2C operation.

**"Sensor Reset" :** Set the Reset output of the Sensor to High or Low.

**"Sensor Enable" :** Set Enable output of Sensor to High or Low.

**"MCLK Off" :** Turns the master clock output On/Off.

**"Data Check" :** If the frame data is an error, it is counted in the box below.

**"View Skip" : S**top the video

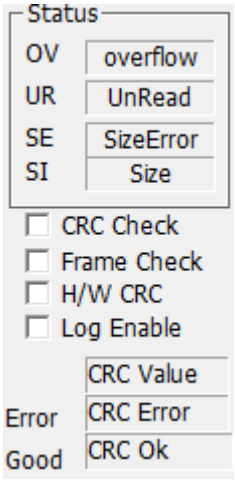
**"Log Enable" :** Enable the log file in the upper right corner.

**"View Once" :** Show the video once.

(1) **"Clr" button**

Initialize the log message screen below.

## 3.10 STATUS Function




It shows the number of the following 4 error states that occur during image transmission.

**"OV" : Overflow**

**"UR" : UnRead**

**"SE" : SizeError**

**"SI" : Size**

**"CRC Check" :** Activate the LVDS Check Sum function.

**"Frame Check" :** Enable Frame CRC.

**"H/W CRC" :** Enable hardware CRC.

**"Log Enable" :** Activate the Log screen.

**CRC Value** : When CRC Check is performed, the CRC value is displayed.

**CRC ERROR** : Shows the number of CRC errors when checking CRC

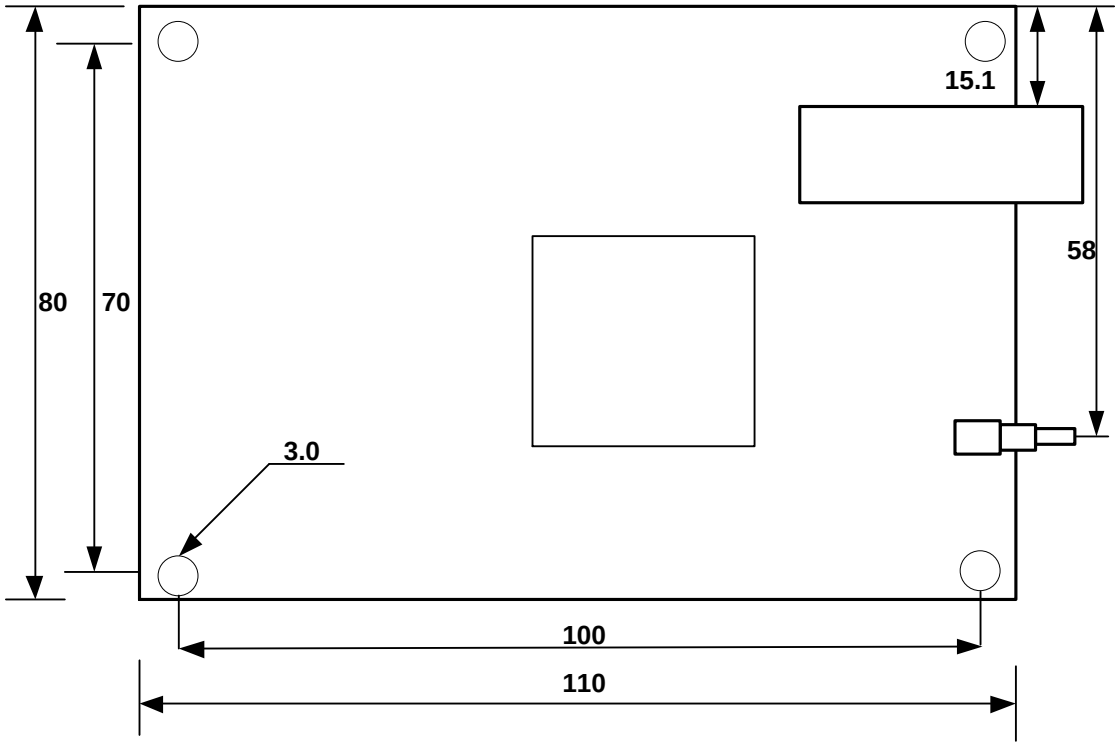
**CRC OK** : When CRC Check, the number of CRC OKs is displayed.

# Appendix

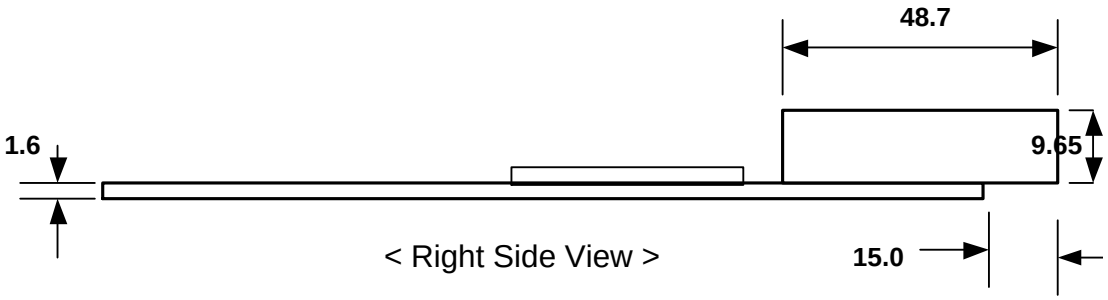
### A-1. Board Size

The external dimensions of the board are as follows.

**(For detailed dimensions, please ask the person in charge.)**




< Top View >




< Right Side View >

**A-2. Repair Regulations**

Thank you for purchasing a DAQSYSTEM product. Please refer to the following regarding customer service regulated by DAQSYSTEM.

(1) Read the user manual before using the DAQSYSTEM product and follow the instructions.

(2) When returning the product to be repaired, please include the symptoms of the failure and send it to the head office.

(3) The warranty period for all DAQSYSTEM products is one year.
-. The warranty period is counted from the date the product is shipped from DAQSYSTEM.
-. Peripherals and third-party products not manufactured by DAQSYSTEM are covered by the manufacturer's warranty.
-. If you need repair, please contact the Contact Point below.

(4) Even during the warranty period, repairs will be charged in the following cases.
① Failure or damage caused by use without following the user's manual
② Breakdown or damage caused by customer's negligence during product transportation after purchase
③ Failure or damage due to natural phenomena such as fire, earthquake, flood, lightning, pollution, or power supply exceeding the recommended range
④ Failure or damage caused by inappropriate storage environment (eg, high temperature, high humidity, volatile chemicals, etc.)
⑤ Breakdown or damage due to unfair repair or modification
⑥ Products whose serial number has been changed or deliberately removed
⑦ If DAQSYSTEM determines that it is the customer's fault due to other reasons

(5) Customer is responsible for shipping costs for returning the repaired product to DAQSYSTEM.

(6) The manufacturer is not responsible for any problems caused by incorrect use, regardless of our warranty.

# MEMO

## Contact Point

Web site : https://www.daqsystem.com

Email : postmaster@daqsystem.com

DAQ SYSTEM
Measurement & Automation