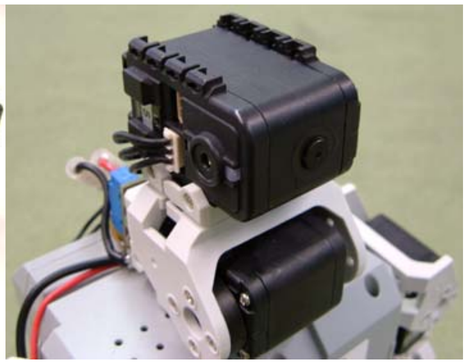
Embedded Vision module for Bioloid
Quick Start

Introduction

This document describes the functionality of a vision module for small robots designed and developed at the Computer Science department of Freie Universität Berlin. The aim of this project is to develop a cheap and lightweight integrated camera and image processor which acquires images and process them using object recognition methods. The results are accessed via a serial interface and therefore are simple and compact enough to be further processed by a low power CPU like a microcontroller. The vision module was used by the RoboCup humanoid robots of FU-Berlin, the "FUmanoids", (http://fumanoid.mi.fu-berlin.de), which won the 3rd Place of the RoboCup 2007 Humanoid League in Atlanta. The hardware platform used for the robots in this project is a Bioloid humanoid from Robotis.
Users who are not familiar with the Dynamixel TTL communication protocol can obtain additional information from the documents published by Robotis Co.

Module Specifications

- Low weight: approx. 33g including Dynamixel housing.
- Color region growing: Up to 15 color regions can be detected in each frame.
- Detects separated regions with the same color as well as well as regions with different colors (no coordinate averaging, real region growing).
- Calculates color, center, amount of pixels and bounding box of each region.
- Image processing runs at 8FPS, 160x120xYCrCb
- Up to 255 different color areas can be defined in the built-in color lookup table.
- The color lookup table can be accessed through the serial bus.
- Built-in, adjustable noise filter allows the detection of both small objects and large ones according to the color, even in very noisy images.
- Adjustable region thresholding removes automatically unwanted small color regions, freeing place for others.
- Supports communication baud rate of 1Mbps.
- Bioloid TTL communication protocol facilitates direct attachment of the device to a Bioloid communication bus.
- Direct access to all camera adjustments makes it possible to use the device either with auto exposure/white balance or manual settings which is ideal for situations whit constant lighting such as RoboCup.
- Raw image output for debugging or lookup table adjustment.

Device operation

The viosion module operates in 2 different modes named “Calibration” and “Implementation”. The calibration phase is for adjusting camera settings as well as color definitions in the look-up table. All settings will be saved into the FLASH/EEPROM memory of the module. This means that under the same lighting and color conditions, there is no need to calibrate the device after every power on. After calibration, the module is ready to be used in implementation phase. In this phase the module is only connected to the CM5 and can receive commands to sample and process the image and return image processing results.

Calibration Phase

In calibration phase the device should be attached as shown in Fig.1 through the CM5 to a PC. The firmware of CM5 should be replaced with the calibration firmware (Calib.hex) using the boot loader. This program facilitates the direct connection of the PC to the vision module.

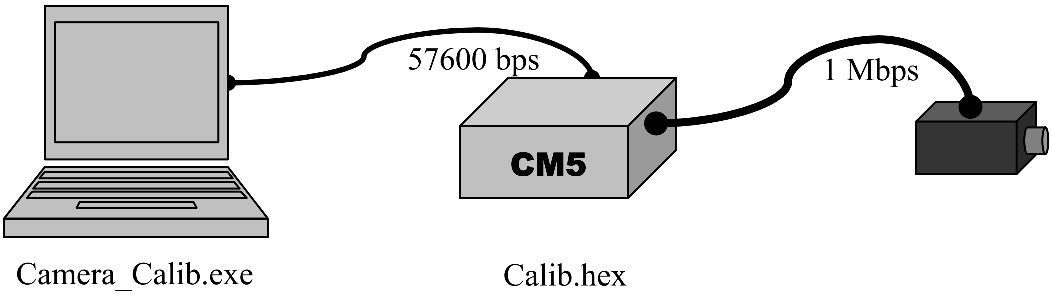


Fig. 1: Connection of the module to a PC in Calibration Phase

The CM5 should be connected to a power supply i.e. AC adaptor or battery. After calibration you can reload your own program to the CM5. An application named “Camera_Calib.exe” is then used to access the camera via a PC. Fig. 2 shows a snapshot of the program.

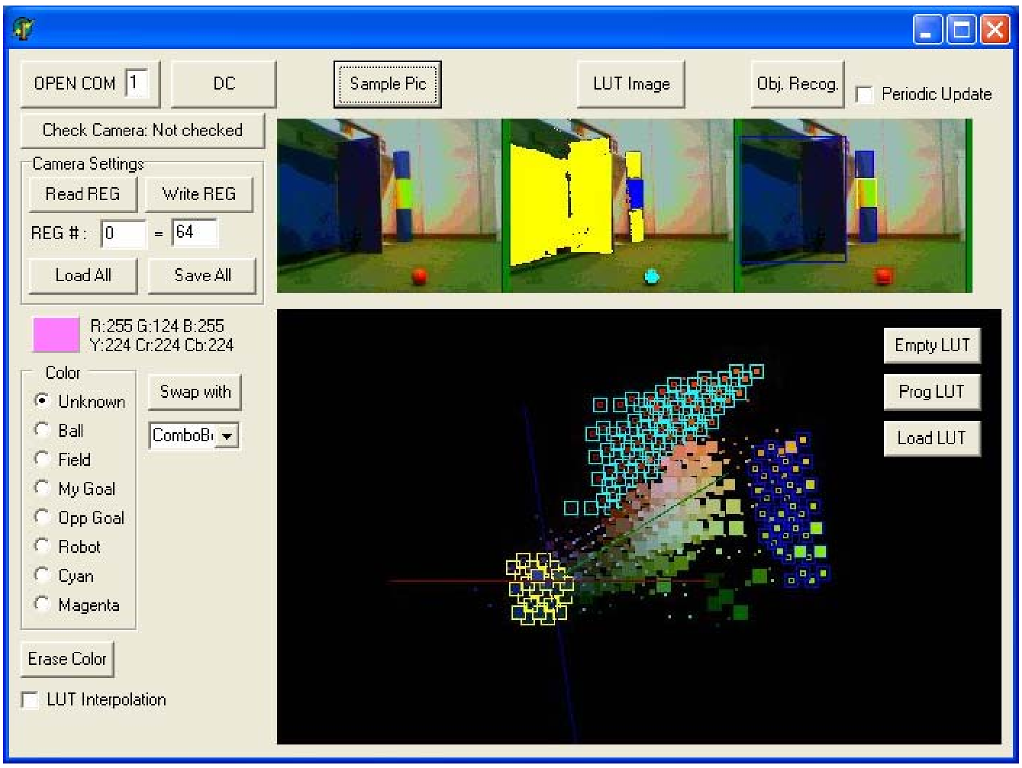


Fig. 2: Snapshot of the calibration user interface

Description of the available functions

-OPEN COM / DC
These buttons start and stop the serial connection of the device to the CM5. The user should change the index of the COM port to the proper one.

- Check Camera
Used to check whether the module can be accessed via the user interface. Any of the following problems will produce the message “Not Found”, otherwise the message “Found” will be shown.

- One of the cables (Serial or TTL) is not connected or damaged.
- The CM5 is turned off
- The CM5 is not loaded with Calib.hex
- The camera is stuck in a wrong mode and should be reset
- The CM5 is stuck somewhere and needs to be reset
- Something else is broken! (Camera/CM5/Cables)

In order to reset the camera the cable should be unplugged. It is recommended to wait a few seconds to let the capacitors discharge, and plug the device again. The CM5 can be reset normally using the reset key.

- Camera settings
The camera inside the module has a DSP which can be configured by accessing its registers each register can be read and written using the calibration interface. To read a register its hexadecimal index must be given in the “*REG#*” box. By pushing the “*ReadREG*” Button the hexadecimal content will be displayed. A similar procedure should be followed to write into a register. A summary of camera registers is shown in the following table.

Register Description

Register	Symbol	Address (Hex)	Default (Hex)	Description
Device ID	DEVID	00	40	Product ID, Revision No.
Sensor Control A	SCTRA	01	09	Operation mode, X/Y flip, Image size
Sensor Control B	SCTRB	02	00	Power down, Clock division
Sensor Control C	SCTRC	03	01	Sensor Internal control Register
Row Start Address High	RSAH	08	00	Row Start Address[8]
Row Start Address Low	RSAL	09	02	Row Start Address[7:0]
Column Start Address High	CSAH	0a	00	Column Start Address[9:8]
Column Start Address Low	CSAL	0b	02	Column Start Address[7:0]
Window Height High	WIHH	0c	01	Window Height Address[8]
Window Height Low	WIHL	0d	e0	Window Height Address[7:0]
Window Width High	WIWH	0e	02	Window Width Address[9:8]
Window Width Low	WIWL	0f	80	Window Width Address[7:0]
HBLANK Time High	HBLANKH	10	00	HBLANK Time [15:8]
HBLANK Time Low	HBLANKL	11	d0	HBLANK Time [7:0]
VBLANK Time High	VBLANKH	12	00	VBLANK Time [15:8]
VBLANK Time Low	VBLANKL	13	08	VBLANK Time [7:0]
Red Color Gain	RCG	14	10	Gain for Red Pixel Output
Green Color Gain	GCG	15	10	Gain for Green Pixel Output
Blue Color Gain	BCG	16	10	Gain for Blue Pixel Output
Preamp Gain	PREAMP	17	10	Preamp Gain for Pixel Output
Preamp Gain Min	PREMIN	18	00	Preamp Gain Min Value for AE
Preamp Gain Max	PREMAX	19	3f	Preamp Gain Max Value for AE
Preamp Gain Nominal	PRENOM	1a	10	Preamp Gain Normal Value for AE
ASP Bias	ASPBIAS	1b	13	Amp Bias, Pixel Bias

Reset Clamp	RSTCLMP	1c	07	Reset Level Clamping Value
ADC Bias	ADCBIAS	20	0f	ADC Bias
Red Pixel Black Offset	OREDI	21	7f	ADC Offset Value for Light-shielded Red Pixel
Green Pixel Black Offset	OGRNI	22	7f	ADC Offset Value for Light-shielded Green Pixel
Blue Pixel Black Offset	OBLUI	23	7f	ADC Offset Value for Light-shielded Blue Pixel
Red Pixel Active Offset	OREDU	24	RO	ADC Offset Value for Active Red Pixel
Green Pixel Active Offset	OGRNU	25	RO	ADC Offset Value for Active Green Pixel
Blue Pixel Active Offset	OBLUU	26	RO	ADC Offset Value for Active Blue Pixel
Black Level Threshold	BLCTH	27	ff	Black Level Threshold Value
ISP Function Enable	ISPFEN	30	0f	Image processing functions enable
ISP Output Format	OUTFMT	31	39	Image data output format control
ISP Output Polarity	OUTINV	32	00	Output signal polarity control
Green Edge Threshold	EDGETH	33	00	Green pixel edge thr. for 3x3 color interpolation
Color Matrix Coefficient 11	CMA11	34	2e	Color matrix coefficient 11
Color Matrix Coefficient 12	CMA12	35	c5	Color matrix coefficient 12
Color Matrix Coefficient 13	CMA13	36	0c	Color matrix coefficient 13
Color Matrix Coefficient 21	CMA21	37	0d	Color matrix coefficient 21
Color Matrix Coefficient 22	CMA22	38	3c	Color matrix coefficient 22
Color Matrix Coefficient 23	CMA23	39	f7	Color matrix coefficient 23
Color Matrix Coefficient 31	CMA31	3a	f8	Color matrix coefficient 31
Color Matrix Coefficient 32	CMA32	3b	cf	Color matrix coefficient 32
Color Matrix Coefficient 33	CMA33	3c	39	Color matrix coefficient 33
AE Mode 1	AEM1	60	39	Auto exposure mode selection 1
AE Mode 2	AEM2	61	ba	Auto exposure mode selection 2
Integration Time High	INTH	63	07	Integration Time [23:16]
Integration Time Middle	INTM	64	a1	Integration Time [15:8]
Integration Time Low	INTL	65	20	Integration Time [7:0]
AE Target	AETGT	66	70	Frame Luminance Target Value
AE Lock & Fine Tune Boundary	AELBND	67	a2	Y frame mean value displacement boundary from AE target where AE goes into Lock state. Fine tuning boundary is also specified.
AE Unlock Boundary	AEUNLCK	68	2a	Y frame mean value displacement from AE target where AE update speed transits from 2x integration unit speed to 1x integration unit speed.
AE Integration Step High	AEINCH	6a	1	Integration Increment Step Unit [17:16]
AE Integration Step Middle	AEINCM	6b	e8	Integration Increment Step Unit [15:8]
AE Integration Step Low	AEINCL	6c	48	Integration Increment Step Unit [7:0]
AE Integration Limit High	AELMH	6d	17	Integration Time Limit [23:16]
AE Integration Limit Middle	AELMM	6e	d7	Integration Time Limit [15:8]
AE Integration Limit Low	AELML	6f	84	Integration Time Limit [7:0]
AWB Mode 1	AWBM1	70	41	AWB mode selection 1
AWB Mode 2	AWBM2	71	2	AWB mode selection 2
Cb Target	CBTGT	73	80	Cb Plane Target Frame Mean Value. Normal white point is 80h.
Cr Target	CRTGT	74	80	Cr Plane Target Frame Mean Value. Normal white point is 80h.
AWB Lock Boundary	AWBLB	75	2	Cb/Cr Frame Mean Displacement from Cb Target and Cr Target where AWB goes into LOCK state.
AWB Unlock Boundary	AWBULB	76	06	Displacement from ideal white pixel where AWB release from LOCK state
AWB White Pixel Boundary	AWBWPB	77	30	Displacement from ideal white pixel where AWB recognizes a pixel as a white pixel affected by light source.
Y Digital Gain	YGAIN	78	40	Y digital gain for Auto Exposure Control
Cb Digital Gain	CBGAIN	79	40	Cb digital gain for Auto White Balance control
Cr Digital Gain	CRGAIN	7a	40	Cr digital gain for Auto White Balance control
AE Status	AEST	7b	RO	AE operation status
AWB Status	AWBST	7c	RO	AWB operation status
Y Frame Mean	YFMEAN	7d	RO	Y Frame Mean Value
Cb Frame Mean	**CBFMEAN**	7e	RO	Cb Frame Mean Value
Cr Frame Mean	**CRFMEAN**	7f	RO	Cr Frame Mean Value

Minimum Anti-Banding Gain	BNDGMIN	80	08	Minimum gain value with Anti-Banding enabled
Maximum Anti-Banding Gain	**BNDGMAX**	81	18	Maximum gain value with Anti-Banding enabled
Integration-Scan Plane Offset High	ISOFSH	82	RO	Integration-Scan Plane Offset[23:16]
Integration-Scan Plane Offset Middle	ISOFSM	83	RO	Integration-Scan Plane Offset[16:8]
Integration-Scan Plane Offset Low	ISOFSL	84	RO	Integration-Scan Plane Offset[7:0]
AWB Luminance High Boundary	AWBLUHI	8a	C8	During CbCr frame mean value calculation, AWB discards pixel of which luminance is larger than this register value.
AWB Luminance Low Boundary	**AWBLULO**	8b	0a	During CbCr frame mean value calculation, AWB discards pixel of which luminance is smaller than this register value.
AWB Valid Number	AWBNO	8c	02	AWB update when the number of valid color pixel is larger than (this minimum value x 64)

RO: These registers are read only, and cannot be written.

Important note
Based on the hardware and firmware design of the module, some of the registers cannot be changed or should only be changed in a limited scale; otherwise it leads to corrupted communication between the module and the camera, or timing failures.

To get a more detailed description of the camera settings, it is recommended to refer to the camera data sheet. The camera has the part number HV7131GP and is a product of Magnachip. See www.magnachip.com for more details.

- *Load All/Save All*
These two buttons are used to load the all the registers with the data stored in a text file named "camerareg.txt" or save them back. To restore default settings, the file "camerareg.txt" should be replaced with the default one, under the same name. Any change to the camera registers is simultaneously stored in the EEPROM of the module, and will be reloaded to the camera at power up.

Sampling Pictures and adjusting lookup table

- *Sample Pic*
This button is used to download a raw image from the camera. Because of the limited bandwidth of the serial communications, several sequential frames are read partially instead of a whole single frame. Therefore the camera and the subject should not move until sampling is finished. Sampling takes normally 4-5 seconds. This image will be then used to adjust the look-up table. An error message will be shown if there is a problem with data communication.

- Look-up table presentation.
Since for each pixel there are 12 bits used for color coding (Y:4,Cr:4,Cb:4), 4096 possible colors can be recognized. These colors can be presented as points inside a cube called color space. The image at the bottom of the form contains a 3D representation of the color space, showing each existing color inside the sampled image as a square with dimensions proportional to the density of the color in the image. This helps selecting the correct color set according to different objects. User can rotate this space by holding the mouse key down and moving over the surface of the Box.

- Marking and editing colors in the Look up table
The Lookup Table is an array of codes assigned to each possible color. There are 2 ways to mark a color as an object inside the look-up table. In the first step the user should select the

color category. This is done by clicking on the proper object in the color box. It is then possible either to click on a position in the sampled image which contains the proposed color, or to click on it in the color space. The color together with its related RGB and YCrCB values are shown as the user moves the pointer over the image or color space. The application shows 3 copies of the downloaded image. The left image shows the original version of the downloaded image. The middle one is the same as the left one except the colors which are defined in the lookup table which are shown with highlighted color as in Fig.1. This helps to check whether the marked colors correctly cover the selected object. The right image is used to present the image processing results as bounding box overlays.
To remove a color from the lookup table, it is enough to mark it as unknown i.e. by selecting the unknown color and clicking over it on the image or in the color space. It is also possible to remove all marked points of a color by selecting the color and clicking the "*Erase Color*" button. To swap two color groups the user should select destination color from the list and click on "*Swap with*" button. Because it can be hard to click on every single occurrence of a color in the space, and because the colors of a group are usually neighbors in color space, the user can add more points to the group using a single click if "*LUT Interpolation*" is checked. For example to mark a triangle in the space one should only click the vertexes. This is easier but not as exact as selecting the points individually.

- Prog. LUT, Load LUT, Empty LUT

There are two copies of the lookup table, one in the user interface, and another one inside the module. It is possible to read the contents of the lookup table from the module, change them and write them back to it. This is done by "*Prog. LUT*" and "*Load LUT*" buttons. To erase the look-up table of the user interface the user can use "*Empty LUT*". Note that the contents of the look-up table can only be accessed by the module, after programming them into the module. Any unsaved information will be lost as the application is closed.

- Final presentation of the results

After calibrating the module it is possible to check the results of the processed image using the "*Obj Recog.*" button. The results are shown as rectangles around the found objects over the current sampled image. To view the results periodically, user should check "*Periodic Update*".

Implementation Phase

The communication between the module and CM5 in the implementation phase has the same form as Dynamixel servos such as AX-12. Therefore, users who are not familiar with this protocol are recommended to refer to AX-12 datasheet for more information.
The next table shows the instructions available in the vision module.

Instruction	Function	Value	No. of Param.
PING	No action. Used for obtaining a Status Packet	0x01	0
READ DATA	Reading Results of Region Detection	0x02	2
READ REG	Reading Camera Registers	0x0C	2
WRITE REG	Writing Camera Registers (1)	0x0D	2
CAPTURE	Capture and Process the Next Frame (1)	0x0E	0
RAWSAMPLE	Sample the Raw Image (used by GUI) (2)	0x0F	0
LUT MANAGE	Entering LUT Manage Mode (used by GUI) (2)	0x10	0
RD FILTHR	Reading Noise Filter Thresholds	0x11	2
WR FILTHR	Writing Noise Filter Thresholds (1)	0x12	2
RD REGTHR	Reading Region Filter Thresholds	0x13	2
WR REGTHR	Writing Region Filter Thresholds (1)	0x14	2

(1) No return packets are generated for these instructions.
(2) These instructions use different protocol rather than Dynamixel standard packets.

-PING
This instruction is used to check whether the device exists and is ready to receive the next instruction. The instruction is the same as in AX-12.

-READ DATA
This instruction is used to read image processing results. It is the same as the READ instruction in Dynamixel, except that the accessed area is not the register area. This command accepts multi byte read. The data structure is described further in this section.

-READ REG
This instruction is to read the content of camera registers. It is the same as READ instruction in Dynamixel. This command accepts multi byte read.

-WRITE REG
This instruction is to write the content of camera registers. It is the same as the WRITE instruction in Dynamixel except that it accepts only single byte write.

-CAPTURE
This instruction starts capturing and processing of the next available frame. It takes approximately 100-125 ms to process a full frame, but this command always waits for the beginning of a frame. This may take twice the time in the worst case. To receive the maximum frame rate, the CM5 should be synchronized with the module to send the command with correct timing. During Image processing the module can not communicate with CM5. CM5 should pole the functionality of the device using the PING command before sending the next instruction.

-RAW SAMPLE
With this instruction the camera module transmits a full frame of raw image data. This instruction is used when the GUI receives a request to sample a raw image. This instruction does not use the Dynamixel packet protocol, and therefore should not be used in implementation mode.

-LUT MANAGE
After receiving this instruction, the module enters the programming mode, in this mode the device accepts no more packets, but other instructions assigned to manage the look-up table, such as erasing, reading and writing into it. This instruction is used by User interface during calibration phase and should not be used in implementation phase.

-RD FILTHR, WR FILTHR
These instructions make access to the threshold values of the noise filter. Noise filter thresholds are actually the minimum number of neighbor pixels in a scan line which should be counted as a part of a region. For each color a separate 8-bit threshold should be defined. Default values are 8 for unknown color and 2 for others. The address field contains the index of the color category (0 = unknown,…). The structure of these instructions are the same as in Dynamixel READ and WRITE, however multi-byte write is not supported.

-RD REGTHR, WR REGTHR
These instructions provide access to the threshold values of the region filter. The region filter thresholds define the minimum number of pixels inside a region, regions with fewer pixels are filtered away. For each color a separate 16-bit threshold should be defined. The address field contains 2*index of the color category (0 = unknown,…). Default values are

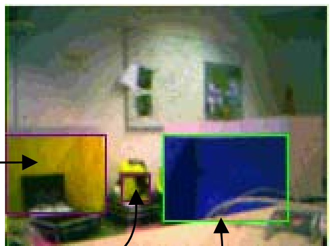
(0,5,50,50,50,100,50,50). No region is built for color 0 (unknown). The structure of the commands are the same as in Dynamixel, however multi-byte write is not supported.

- Output data format

Following example shows the produced output of the module according to a given image. Up to 15 regions can be read from address 0x10 to 0xFF using the ***READ DATA(0x02)*** instruction. Each region occupies 16 bytes which are as follows.

- **Region Index**: Contains the value 0 if the region is invalid and nonzero otherwise.
- **Region Color**: Color category of the detected region. (0 = Unknown, 1 = Ball , …)
- **Number of pixels**: Number of detected pixels inside the region.
- **Sum of X Coordinates**: Result of addition of the X coordinates of all detected pixels. Can be divided by *Number of Pixels* to calculate average X.
- **Sum of Y Coordinates**: Result of addition of the Y coordinates of all detected pixels. Can be divided by *Number of Pixels* to calculate average Y.
- **Max X**: Bounding box right margin.
- **Min X**: Bounding box left margin.
- **Max Y**: Bounding box bottom margin.
- **Min Y**: Bounding box top margin.

Region Index	Region Color	Number of Pixels		Sum of X Coordinates				Sum of Y Coordinates				Max X	Min X	Max Y	Min Y
01	04	3B	05	63	89	00	00	FC	A3	01	00	31	00	67	41
02	03	24	08	83	65	03	00	53	A6	02	00	8C	4F	6B	41
03	04	5F	00	5B	17	00	00	07	22	00	00	47	3A	61	55
00	00	03	00	36	00	00	00	47	01	00	00	13	10	6D	6D
00	00	03	00	3F	00	00	00	4A	01	00	00	16	13	6E	6E
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00



- Capture / Read Sequence

After sending a *CAPTURE* instruction, the device is not accessible for 100-125ms. Although it is possible to poll the availability of results with the *PING* instruction, it is recommended that a sequence of READ/CAPTURE instructions be sent using a timer interrupt which should be synchronized using PING instruction to avoid missed frames. Note that it is better to read the results of the last frame and send a capture command immediately, rather than sending a capture command and waiting for the results to arrive.