

CAMERA



[THE BLAZINGCORE SERIES]

WITH SUPERIOR NUMBER CRUNCHING ABILITIES AND PERIPHERAL HANDLING ON OUR CUSTOM EMBEDDED OS,
RAPID PROTOTYPING IS NOW EASY... AND BLAZING FAST.

CAM

Provides access to the CMOS c3038 camera module.

Member	Description
Init	Does the necessary one-time initialising required for the Camera Module to be controlled thereafter
GrabFrame	Captures a single digital frame from the camera module and stores it into the GRAM of the OLED
Mode	Set frame capture size
Speed	Set capture rate
Brightness	Set brightness level of the camera module
Contrast	Set contrast level of the camera module
Position	Position to display the captured frame on the OLED
AGC	Turn On/Off Auto Gain Control
ReadReg	Provides read access to the registers of the camera module
WriteReg	Provides write access to the registers of the camera module
MONO	Provide access to methods in Mono (Greyscale) Mode
RGB	Provide access to methods in RGB (Colour) Mode

INITIALIZING THE CAMERA MODULE

The Camera Module must be initialised before any frame grabbing can be done.

INIT

Does the necessary one-time initialising required for the Camera Module to be controlled thereafter

CAM.Init()

Code:

```
PUBLIC SUB MAIN()  
  
  'Init Camera Module  
  CAM.INIT()  
  
END SUB
```

GRABFRAME

Grabs a single frame from the camera module and writes it directly to the OLED's GRAM for display.

CAM.GrabFrame()

Code:

```
PUBLIC SUB MAIN()  
  
  'Init Camera Module  
  CAM.Init()  
  DELAY(200)  
  'Init OLED  
  OLED.Init(0)  
  
  DO  
    CAM.GrabFrame()  
  LOOP  
  
  'Frame rate for this program is ~17-18fps, resolution of  
  160x120px
```

VISION SYSTEM CAMERA COMMANDS

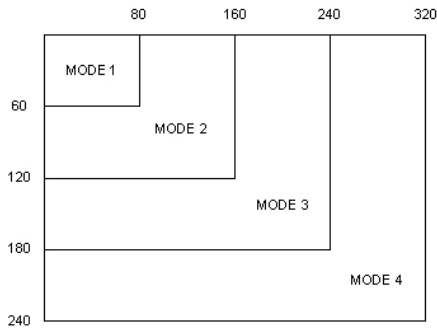
MODE

Set frame capture size

CAM.Mode (ByVal Mode As Integer)

Modes Available:

- Mode '1': 80(W) x 60(H) pixels
- Mode '2': 160(W) x 120(H) pixels (Default)
- Mode '3': 240(W) x 180(H) pixels (Digital Zoom)
- Mode '4': 320(W) x 240(H) pixels (Full-Screen) (Digital Zoom)



NOTE:
ORIENTATION = 0 MODE 1,2,3,4
ORIENTATION = 1 MODE 1,2,3 ONLY

Code:

```
PUBLIC SUB MAIN()  
  
    'Init Camera Module  
    CAM.Init()  
    DELAY(200)  
    OLED.Init(0)  
    CAM.Mode(2) 'Set camera capture size to 160 x 120px  
  
    DO  
        CAM.GRABFRAME() 'Capture to OLED screen  
    LOOP  
  
END SUB
```

SPEED

Set capture rate

CAM.Speed (ByVal Speed As Integer)

Parameter/s:

- **Speed**; Integer Speed Range of 2 – 64, with 2 being the fastest and 64 being the slowest.

BRIGHTNESS

Brightness control of the imaging sensor.

CAM.Brightness (ByVal BrightnessLevel As Integer)

Parameter/s:

- **Brightness Level**; Brightness can be adjusted using the range of 0 – 255, where 255 is the highest and 0 is the lowest.

CONTRAST

Contrast control of the imaging sensor; corresponds to bit settings in the OV6630 Register for Contrast Control.

CAM.Contrast (ByVal ContrastLevel As Integer)

Parameter/s:

- **Contrast Level**; Range of 0 – 255, where 255 is the highest and 0 is the lowest.

With Reference to the OV6630 Datasheet;

OV6630 Contrast Control Register Description

CTR[7] – selects gain at the dark area. “0” – gain=0.5 and “1” – gain=1.

CTR[6:4] – reserved.

CTR[3:0] – Contrast adjustment. “FFh” is highest and “00h” is lowest.

AGC

Set Auto Gain Control On/Off

CAM.AGC (ByVal AGC_Mode As Integer)

Parameter/s:

- **AGC_Mode**;
- ‘0’: Turn Off Auto Gain Control
- ‘1’: Turn On Auto Gain Control

POSITION

Sets the position to display the frame from the camera module on the OLED.

CAM.Position (ByVal X As Integer, ByVal Y As Integer)

Parameter/s:

- *X*; Integer x-coordinate
- *Y*; Integer y-coordinate

READREG

Return contents of the OV6630 register at a specified address.

Variable = **CAM.ReadReg** (Register Address)

WRITEREG

Write data to the OV6630 register at a specified address.

CAM.WriteReg (Register Address, Value)

Code:

```
DIM I AS INTEGER
CAM.WriteReg(17, 5) ' SPEED
I = CAM.ReadReg(17)
Debug.Print CSTR(I)
```

VISION SYSTEM CAMERA COMMANDS

MONO

Provide access to methods in Greyscale Mode

Member	Description
GrabFrame2Mem	Capture frame to screen and memory at the same time
Min	Returns the minimum colour value found within captured image memory
Max	Returns the maximum colour value found within captured image memory

.GRABFRAME2MEM

Capture greyscale frame to screen and memory at the same time

`Cam.Mono.GrabFrame2Mem (ByRef MemoryArray() As Byte)`

.MIN

Returns the minimum colour value found within captured image memory

`Variable = Cam.Mono.Min () As Integer`

.MAX

Returns the maximum colour value found within captured image memory

`Variable = Cam.Mono.Max () As Integer`

VISION SYSTEM CAMERA COMMANDS

RGB

Provide access to methods in RGB Colour Mode

Member	Description
Frame	Capture frame from camera module and display a specific region on the OLED screen
GrabFrame2Mem	Capture frame to screen and memory at the same time

.FRAME

Capture frame from camera module and display a specific region on the OLED screen, at a position specified using the command CAM.Position

Cam.RGB.Frame (ByVal X1 As Integer, ByVal X2 As Integer, ByVal Y1 As Integer, ByVal Y2 As Integer)

Position Setting

```
CAM.Position(0,0)  
(X,Y) = (0,0)
```

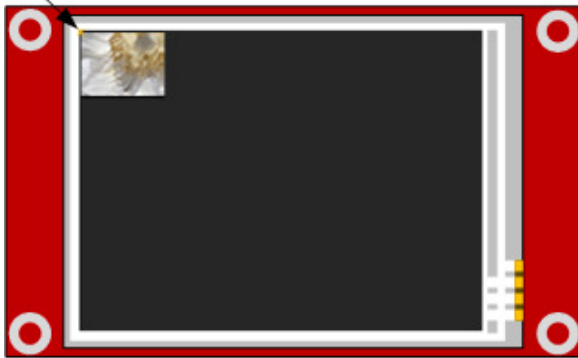


Image Captured From Camera



Default: 160 x 120 px



Specified Frame Region

CAM.RGB.Frame (X1 , X2 , Y1 , Y2)

.GRABFRAME2MEM

Capture RGB Colour frame to screen(5/6/5) and memory(8/8/8) at the same time

Cam.RGB.GrabFrame2Mem (ByRef MemoryArray() As Byte)

LATEST DOCUMENTATION

All of our documentations are constantly updated to provide accurate and/or new information that we feel would help you with developing with our products.

The latest documentation may be obtained from our website: <http://www.aiscube.com/main/downloads.html>

HOW YOU CAN HELP

You can help us to improve our documentations by emailing to us or posting a thread in our forum, reporting any mistakes/typos or errata that you might spot while reading our documentation.

Email: TechSupport@aiscube.com

Forum: <http://forum.aiscube.com>

DISCLAIMER

All information in this documentation is provided 'as-is' without any warranty of any kind.

The products produced by AIS Cube are meant for rapid prototyping and experimental usage; they are not intended nor designed for implementation in environments that constitute high risk activities.

AIS Cube shall assume no responsibility or liability for any indirect, specific, incidental or consequential damages arising out of the use of this documentation or product.

COPYRIGHT© 2009 - 2011 AIS CUBE. ALL RIGHTS RESERVED.

ALL PRODUCT AND CORPORATE NAMES APPEARING IN THIS DOCUMENTATION MAY OR MAY NOT BE REGISTERED TRADEMARKS OR COPYRIGHTS OF THEIR RESPECTIVE COMPANIES. AND ARE ONLY USED FOR IDENTIFICATION OR EXPLANATION FOR THE OWNER'S BENEFIT. WITH NO INTENT TO INFRINGE.

SONATA IDE AND BLAZINGCORE(BCORE) ARE TRADEMARKS OF AIS CUBE IN SINGAPORE AND/OR OTHER COUNTRIES. ALL IMAGES DEPICTING THE BLAZINGCORE OR ANY PART OF IT IS COPYRIGHTED.

ALL OTHER TRADEMARKS OR REGISTERED TRADEMARKS ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS.