CPRE 288: Datasheet Trainer (Camera Controller)

Introduction: The purpose of this document is to help those new to Datasheets work through the feeling of being overwhelmed by the information provided by datasheets. Tips are first given for how to go about using a datasheet. The remainder of the document then provides a representative section/chapter of a datasheet for a device called a Camera Controller.

Know the Organization: When first using a Datasheet, it is important to understand how information is organized. For microcontroller Datasheets, there is typically a section/chapter for each device. All sections/chapters of a given datasheet tend to have near identical organization. It is common for a section/chapter to have the following parts: **a) Overview, b) Details for each capability, and c) Register Description.** ***Overview:*** This part often gives the big picture purpose of the device, a list of capabilities, and a high-level diagram of how data moves through the device. ***Details for each capability:*** For a given capability, details are given on how the capability works, and how to configure the device to use that capability. ***Register Description:*** For each register associated with the device, this part gives details of the purpose of each bit in the register. This part is useful for quickly looking up device usage details once one has an overall understanding of how the device works. **By far** this is most often utilized part of a datasheet chapter/section.

How to Read: **1)** Take *15 minutes* to scan though the appropriate chapter/section to understand what type of information is provided and where. As examples: Is there a diagram that shows how data moves through the device? Are there areas that explain how to initialize and configure the device? Are there code examples? Where are the Register Descriptions? **2)** Take *20 minutes* to read the Overview and understanding the provided device diagram. **3)** Take *10 minutes* to scan through the details provided for the various capabilities. **4)** Take *15 minutes* to read the Register Descriptions. This is helpful for gaining insight into what bits you will need to care about. **5)** After taking ~60 minutes for steps 1-4, start to use the datasheet to accomplish the task you wish to perform.

Overview: The Camera Controller provides software accessible registers to simplify interacting with small cameras. Logic internal to the controller handles low-level details associated with the communication protocol used across the physical cable connecting a camera to the microcontroller. By abstracting away these complexities, the embedded systems developer can focus on higher level tasks. This internal logic also frees the microcontroller's CPU from implementing the communication protocol, allowing it to be used for computing higher level aspects of a system.

The software accessible registers allow a camera to be configured for different Frame Rates, Resolutions, Color modes, and Frame capture modes. They also allow developers to interact with a camera in a "Polling" or "Interrupt" based manner.

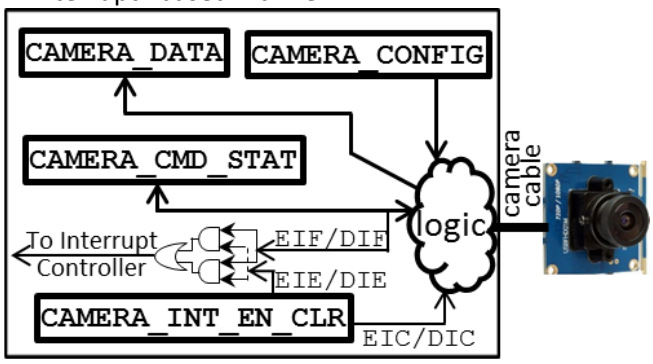

Figure 1: Camera Controller Block Diagram

Block Diagram: Figure 1 provides a high-level view of the internal workings of the Camera Controller. Its software accessible registers are shown in bold boxes.

Configuration Options: Most camera configuration options are specified using the **CAMERA_CONFIG** register. When a 1 is written to the *Enable* bit of this register, the contents of the register are sent to the camera to update its configuration and the Camera Controller is enabled. As a configuration update is being sent to the camera, the user must wait for the configuration to complete before attempting to use the camera. The *Config Pending* bit of the **CAMERA_CMD_STAT** register indicates when a configuration update has completed.

Frame Capture: Two frame capturing modes are supported: One-shot, and Continuous. In Continuous Mode, once a "Snap" command is given by writing a 1 to the *Snap* bit of the **CAMERA_CMD_STAT** register, the camera will take pictures at the specified speed until the Camera Controller is disabled. In One-shot mode, once a "Snap" command is given, the camera will only take a single picture, and then wait for another "Snap" command to be given.

Frame Retrieval: Software uses the **CAMERA_DATA** register to retrieve Frame data 8-bits at a time. The *Data Ready* bit of the **CAMERA_CMD_STAT** register indicates when a new 8-bit pixel of data has been received. Reading **CAMERA_DATA** resets the *Data Ready* bit to 0. Note: **CAMERA_DATA** is overwritten when a new pixel is received, thus if **CAMERA_DATA** is not read in a timely fashion pixel data will be lost.

Color Modes: Two color modes are supported: 8-bit color, and 8-bit gray scale. The detailed register description of the **CAMERA_DATA** register provides the 8-bit format used for each of these modes.

Interrupts: The camera controller supports the option of generating interrupts for two types of events: Data Received, and Error occurrence. When Data Received Interrupts are enabled in the **CAMERA_INT_EN_CLR** register, an interrupt signal is sent to the Interrupt Controller each time new data is received by the **CAMERA_DATA** register. Similarly, an interrupt signal is sent when an Error occurs during communication with a camera. Additionally, flags in the **CAMERA_CMD_STAT** register corresponding to the interrupts enabled will be set to 1 on the occurrence of the corresponding events.

The developer must check within their Interrupt Service Routine (ISR) which specific type of event has occurred, since the controller has only one interrupt signal for the entire device. Also within their ISR they must clear the appropriate interrupt flag, or future interrupts will not occur.

Register Descriptions: Name (Memory Map Location)

CAMERA_CONFIG (0xFFFF_A000): This register allows software to configure several aspects of a camera.

7	6	5	4	3	2	1	0
RSV	CLM	RS1	RS0	SP2	SP1	SP0	EN
RO	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	Reserved (RSV)	None
6	Color Mode (CLM)	0: Gray Scale 1: Color
5:4	Resolution (RS) (Width x Height)	0: 100x100 pixels 1: 320x240 pixels 2: 640x480 pixels 3: 1280x1024 pixels
3:1	Speed (SP) (Frames per second: FPS)	0: 1 FPS 1: 30 FPS 2: 60 FPS 3: 120 FPS 4: 240 FPS >4: Undefined
0	Enable (EN)	0: Disabled 1: Enabled

CAMERA_DATA (0xFFFF_A001): This register allows software to read data from a camera one pixel at a time.

7	6	5	4	3	2	1	0
R2	R1	R0	G2	G1	G0	B1	B0
RO	RO	RO	RO	RO	RO	RO	RO

CLM = 0

Bit	Name	Function
7:0	Pixel Data	0x00: Black 0x01-0xFE: Gray Shade 0xFF: White

CLM = 1

Bit	Name	Function
7:5	Red(R)	0-7: Red Intensity level
4:2	Green(G)	0-7: Green Intensity level
1:0	Blue(B)	0-3: Blue Intensity level

Key: RO (Read Only), WO (Write Only), RW (Read and Write)

CAMERA_CMD_STAT (0xFFFF_A002): This register allows software to send commands to, and receive status from a camera.

7	6	5	4	3	2	1	0
EIF	DIF	RSV	ER	DRY	SM	S	CP
RO	RO	RW	RO	RO	RW	WO	RO

Bit	Name	Function
7	Error Int Flag(EIF)	If EIE=1, set to 1 when an error occurs
6	Data Int Flag(DIF)	If DIE=1, set to 1 when new data received
5	Reserved (RSV)	None
4	Error (ER)	1: Error has occurred 0: Reading CAMEAR_DATA sets to 0
3	Data Ready (DRY)	1: New data ready 0: Reading CAMERA_DATA sets to 0
2	Snap Mode (SM)	0:One shot Mode 1:Continuous shot Mode
1	Snap (S)	1: Take picture(s)
0	Config Pending (CP)	0: Config completed 1: Config in process

CAMERA_INT_EN_CLR (0xFFFF_A003): This allows software to enable interrupts, and to clear an interrupt once it has occurred.

7	6	5	4	3	2	1	0
RSV	RSV	EIC	DIC	RSV	RSV	EIE	DIE
RO	RO	WO	WO	RO	RO	RW	RW

Bit	Name	Function
7:6	Reserved (RSV)	None
5	Error Interrupt Clear (EIC)	0: No effect 1: Clear EIF
4	Data Interrupt Clear (DIC)	0: No effect 1: Clear DIF
3:2	Reserved (RSV)	None
1	Error Interrupt Enable (EIE)	0:Disable Error Ints 1:Enable Error Ints
0	Data Interrupt Enable (DIE)	0: Disable Data received interrupts 1: Enable Data received interrupts