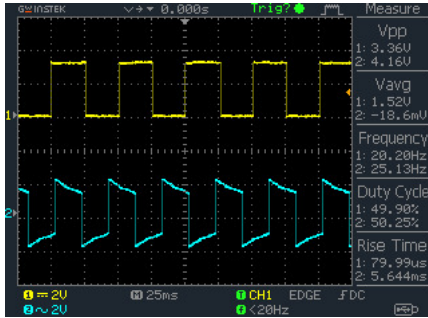I am trying to run an application which captures incoming Y only image data on DCMI interface.

Pixel clk is 13.5MHz, field rate is 50Hz (PAL video), 8 bit luminance only data and discrete sync signals.

MCU is STM32F767IG and camera device is a TVO5150AM1 decoder.

The problem I am facing here is that DCMI capture rate is not as same as VSYNC signal.

In continues mode, When VSYNC frequency is 50Hz, DCMI captured fieldrate is 40Hz.

I had placed a toggle pin at VsyncEventCallback and FrameEventCallback to detect their rates. Yellow one represents Frame(field)rate while the blue one represents VSYNC frequency.

There are similar issues in snapshot mode which causes a frame to be captured from somewhere other that start of a line.

I have to add that it does not overrun, as no errorcallback is triggered.

DCMI configuration code in continues mode:

```
oid MX_DCMI_Init(void)
{

  hdcmi.Instance = DCMI;
  hdcmi.Init.SynchroMode = DCMI_SYNCHRO_HARDWARE;
  hdcmi.Init.PCKPolarity = DCMI_PCKPOLARITY_RISING;
  hdcmi.Init.VSPolarity = DCMI_VSPOLARITY_HIGH;
  hdcmi.Init.HSPolarity = DCMI_HSPOLARITY_LOW;
  hdcmi.Init.CaptureRate = DCMI_CR_ALL_FRAME;
  hdcmi.Init.ExtendedDataMode = DCMI_EXTEND_DATA_8B;
  hdcmi.Init.JPEGMode = DCMI_JPEG_DISABLE;
  hdcmi.Init.ByteSelectMode = DCMI_BSM_ALL;
  hdcmi.Init.ByteSelectStart = DCMI_OEBS_ODD;
  hdcmi.Init.LineSelectMode = DCMI_LSM_ALL;
  hdcmi.Init.LineSelectStart = DCMI_OELS_ODD;
  if (HAL_DCMI_Init(&hdcmi) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

}

void HAL_DCMI_MspInit(DCMI_HandleTypeDef* dcmiHandle)
{

  GPIO_InitTypeDef GPIO_InitStruct;
  if(dcmiHandle->Instance==DCMI)
  {
  /* USER CODE BEGIN DCMI_MspInit 0 */

  /* USER CODE END DCMI_MspInit 0 */
    /* DCMI clock enable */
    __HAL_RCC_DCMI_CLK_ENABLE();

    /**DCMI GPIO Configuration
    PA4     ------> DCMI_HSYNC
    PA6     ------> DCMI_PIXCLK
    PC6     ------> DCMI_D0
    PC7     ------> DCMI_D1
    PC8     ------> DCMI_D2
    PC9     ------> DCMI_D3
    PC11    ------> DCMI_D4
    PD3     ------> DCMI_D5
    PG9     ------> DCMI_VSYNC
    PI6     ------> DCMI_D6
    PI7     ------> DCMI_D7
    */
    GPIO_InitStruct.Pin = GPIO_PIN_4|GPIO_PIN_6;
    GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
    GPIO_InitStruct.Alternate = GPIO_AF13_DCMI;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

    GPIO_InitStruct.Pin = GPIO_PIN_6|GPIO_PIN_7|GPIO_PIN_8|GPIO_PIN_9
                          |GPIO_PIN_11;
    GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
    GPIO_InitStruct.Alternate = GPIO_AF13_DCMI;
    HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

    GPIO_InitStruct.Pin = GPIO_PIN_3;
    GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
```

```
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
    GPIO_InitStruct.Alternate = GPIO_AF13_DCMI;
    HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);

    GPIO_InitStruct.Pin = GPIO_PIN_9;
    GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
    GPIO_InitStruct.Alternate = GPIO_AF13_DCMI;
    HAL_GPIO_Init(GPIOG, &GPIO_InitStruct);

    GPIO_InitStruct.Pin = GPIO_PIN_6|GPIO_PIN_7;
    GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
    GPIO_InitStruct.Alternate = GPIO_AF13_DCMI;
    HAL_GPIO_Init(GPIOI, &GPIO_InitStruct);

    /* DCMI DMA Init */
    /* DCMI Init */
    hdma_dcmi.Instance = DMA2_Stream7;
    hdma_dcmi.Init.Channel = DMA_CHANNEL_1;
    hdma_dcmi.Init.Direction = DMA_PERIPH_TO_MEMORY;
    hdma_dcmi.Init.PeriphInc = DMA_PINC_DISABLE;
    hdma_dcmi.Init.MemInc = DMA_MINC_ENABLE;
    hdma_dcmi.Init.PeriphDataAlignment = DMA_PDATAALIGN_WORD;
    hdma_dcmi.Init.MemDataAlignment = DMA_MDATAALIGN_WORD;
    hdma_dcmi.Init.Mode = DMA_CIRCULAR;
    hdma_dcmi.Init.Priority = DMA_PRIORITY_VERY_HIGH;
    hdma_dcmi.Init.FIFOMode = DMA_FIFOMODE_ENABLE;
    hdma_dcmi.Init.FIFOThreshold = DMA_FIFO_THRESHOLD_FULL;
    hdma_dcmi.Init.MemBurst = DMA_MBURST_INC4;
    hdma_dcmi.Init.PeriphBurst = DMA_PBURST_SINGLE;
    if (HAL_DMA_Init(&hdma_dcmi) != HAL_OK)
    {
      _Error_Handler(__FILE__, __LINE__);
    }

    __HAL_LINKDMA(dcmiHandle,DMA_Handle,hdma_dcmi);

    /* DCMI interrupt Init */
    HAL_NVIC_SetPriority(DCMI_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DCMI_IRQn);
  /* USER CODE BEGIN DCMI_MspInit 1 */

  /* USER CODE END DCMI_MspInit 1 */
  }
}??????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
```