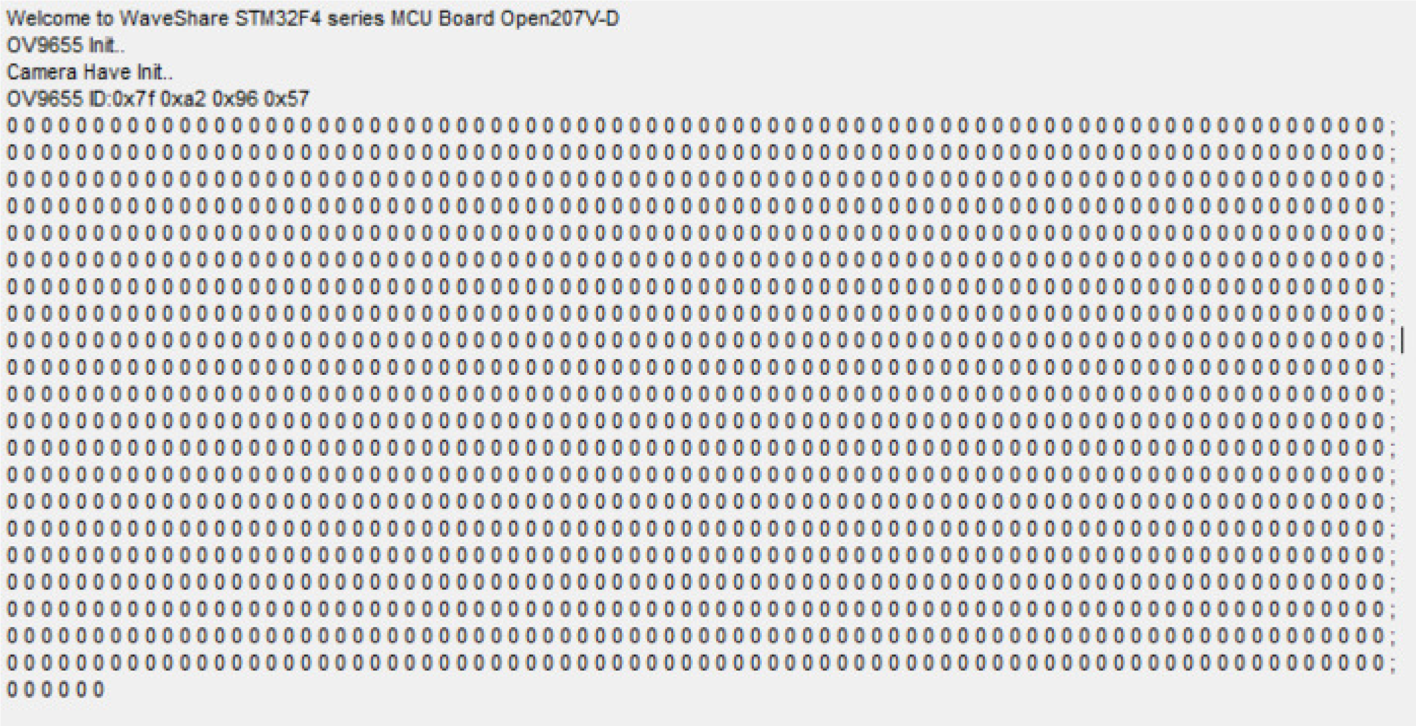

```
Welcome to WaveShare STM32F4 series MCU Board Open207V-D
OV9655 Init..
Camera Have Init..
OV9655 ID:0x7f 0xa2 0x96 0x57
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
0 0 0 0 0 0
```

Hello all, I am a new of this part and It is my first discussion in here. Please give a lot of comments as you can. It would be very helpful.
Currently, I am trying to communicate DMA DCMI and UART2 with the pc.
I refered a lot of documents and discussion in here. I used and refered and modified given example(which is supposed to communicate with the LCD). However, the result of the communication is worng.
I received some datas but it might be worng. I attached the code and the result. Please give me some advices to get good result. Thanks

```
************************main part************************************
#include "stm32f4xx.h"
#include "usart.h"
//#include "LCD/LCD.h"
#include "camera/dcmi_OV9655.h"
#define WIDTH 72
#define HEIGHT 80
#define BYTES_PER_PIXEL 2

unsigned int frame_flag=0;
unsigned int  K=0;

int main(void)
{
  int a ;

 OV9655_IDTypeDef OV9655ID;

 USART_Configuration();
 //USART_NVIC_Config();


 printf("\r\n\r\nWelcome to WaveShare STM32F4 series MCU Board Open207V-D\r\n");
 printf("OV9655 Init..\r\n");


 if(DCMI_OV9655Config()==0){
```

```
   printf("Camera Have Init..\r\n");

  }
  else {

   printf("OV9655 Init fails!!\r\n");

   }

  DCMI_OV9655_ReadID(&OV9655ID);
  printf("OV9655 ID:0x%x 0x%x 0x%x 0x%x\r\n",OV9655ID.Manufacturer_ID1,
OV9655ID.Manufacturer_ID2, OV9655ID.PID, OV9655ID.Version);
  /* Enable DMA transfer */
    DMA_Cmd(DMA2_Stream1, ENABLE);

   /* Enable DCMI interface */
    DCMI_Cmd(ENABLE);

   /* Start Image capture */
    DCMI_CaptureCmd(ENABLE);
    SendPicture();

 while (1);


   }



 void DMA2_Stream1_IRQHandler(void)
 {
  static  int K;
   //Test on DMA2 Channel1 Transfer Complete interrupt
   if(DMA_GetITStatus(DMA2_Stream1,DMA_IT_TCIF1) ==  SET)
   {
   frame_flag = 1;//when frame_flag =1,all the data will be send through serial port in main function while loop
     DMA_ClearITPendingBit(DMA2_Stream1,DMA_IT_TCIF1);
   }
   if(DMA_GetITStatus(DMA2_Stream1,DMA_IT_TEIF1) ==  SET)
   {
     sprintf(&tempBuff[0]," Dma error \r\n");SendChar2(tempBuff[0]);
   printf("%c",tempBuff[0]);
     DMA_ClearITPendingBit(DMA2_Stream1,DMA_IT_TEIF1);
   }
 }

 void DCMI_IRQHandler(void)
 {
  static volatile int line,col,i,j = 0;
     if(DCMI_GetFlagStatus(DCMI_FLAG_FRAMERI) == SET)
     {

    printf("Frame got\r\n");
    DCMI_ClearFlag(DCMI_FLAG_FRAMERI);
     }
    if(DCMI_GetFlagStatus(DCMI_FLAG_OVFRI) == SET)
     {
   printf("overflow\r\n");
```

```
    DCMI_ClearFlag(DCMI_FLAG_OVFRI);
     }

 }

void Delay(uint32_t nTime)
{
 unsigned int  del;
del = nTime;
 while(del != 0);
}

uint32_t tempBuffer(uint32_t index)
{
return tempBuff[index];
}

int SendChar2(unsigned char ch)
{
 /* Wait till holding buffer empty */

 while(USART_GetFlagStatus(USART2, USART_FLAG_TXE) == RESET);
USART_SendData(USART2, (unsigned int) ch);
 return (ch);
}

void SendPicture(void)
{
 //while(USART_GetFlagStatus(USART2, USART_FLAG_TXE) == RESET);
//SendChar2('O');
//while(USART_GetFlagStatus(USART2, USART_FLAG_TXE) == RESET);
//SendChar2('K');

 for(K=0;K< WIDTH*HEIGHT*BYTES_PER_PIXEL/4;K++)
 {

 printf("%x \n",Buffervalue(4*K+1));
 printf("%x \n",Buffervalue(4*K+3));
 if((K+1)%40==0)
 printf("; \r\n");
 }
//while(1);
}

**************************************dcmiov9655 part
************************************
/**
 ******************************************************************************
 * @file    DCMI/OV9655_Camera/dcmi_OV9655.c
 * @author  MCD Application Team
 * @version V1.0.0
 * @date    18-April-2011
 * @brief   This file includes the driver for OV9655 Camera module mounted on
 *          STM322xG-EVAL board RevA and RevB.
 ******************************************************************************
 * @attention
 *
 * THE PRESENT FIRMWARE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING
CUSTOMERS
```

```
  * WITH CODING INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE
  * TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY
  * DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS
ARISING
  * FROM THE CONTENT OF SUCH FIRMWARE AND/OR THE USE MADE BY CUSTOMERS OF THE
  * CODING INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.
  *
  * <h2><center>&copy; COPYRIGHT 2011 STMicroelectronics</center></h2>
  ******************************************************************************
  */

/* Includes ------------------------------------------------------------------*/
#include "camera/dcmi_OV9655.h"
#define ROW 72
#define COLUMNS 80  //I got only 80 columns instead of 88 columns
#define BYTESPERPIX 2

/** @addtogroup DCMI_OV9655_Camera
  * @{
  */

/* Private typedef -----------------------------------------------------------*/
/* Private define ------------------------------------------------------------*/
#define  TIMEOUT  2
/* Bits definitions ----------------------------------------------------------*/
/* Private macro -------------------------------------------------------------*/
/* Private variables ---------------------------------------------------------*/
/* Private function prototypes -----------------------------------------------*/
/* Private functions ---------------------------------------------------------*/
static void Delay(uint32_t nTime);
static void Delay_ms(uint32_t nTime);

uint32_t outputBuffer[144*160];

/**
  * @brief  Configures the DCMI to interface with the OV9655 camera module.
  * @param  None
  * @retval None
  */

uint8_t DCMI_OV9655Config(void)
{

  MCO1_Init();
  /* I2C1 will be used for OV9655 camera configuration */
  SCCB_GPIO_Config();



 DCMI_OV9655_QVGASizeSetup();
  Delay_ms(0xfff);
  /* Reset and check the presence of the OV9655 camera module */
  if(DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x12, 0x80))
  {
    return (0xFF);
  }
  Delay_ms(0xfff);

  /* OV9655 Camera size setup */
#if defined (QQVGA_SIZE)
```

```
  DCMI_OV9655_QQVGASizeSetup();
#elif defined (QVGA_SIZE)
  DCMI_OV9655_QVGASizeSetup();
#endif

  /* Set the RGB565 mode */
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS, OV9655_COM7, 0x63);
//  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS, OV9655_COM15, 0x10);

  /* Invert the HRef signal*/
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS, OV9655_COM10, 0x08);


  /* Configure the DCMI to interface with the OV9655 camera module */
  DCMI_Config();

  return (0x00);
}

void DCMI_Config(void)
{
  DCMI_InitTypeDef DCMI_InitStructure;
  GPIO_InitTypeDef GPIO_InitStructure;
  DMA_InitTypeDef  DMA_InitStructure;
  NVIC_InitTypeDef NVIC_InitStructure;

  /* Enable DCMI GPIOs clocks */
  RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA | RCC_AHB1Periph_GPIOB |
RCC_AHB1Periph_GPIOC | RCC_AHB1Periph_GPIOE, ENABLE);

  /* Enable DCMI clock */
  RCC_AHB2PeriphClockCmd(RCC_AHB2Periph_DCMI, ENABLE);

  /* Connect DCMI pins to AF13 *****************************************************/
  GPIO_PinAFConfig(GPIOA, GPIO_PinSource4, GPIO_AF_DCMI);//HSYNC
  GPIO_PinAFConfig(GPIOA, GPIO_PinSource6, GPIO_AF_DCMI);//PCLK
  GPIO_PinAFConfig(GPIOB, GPIO_PinSource7, GPIO_AF_DCMI);//VSYNC

//RED
  GPIO_PinAFConfig(GPIOC, GPIO_PinSource6, GPIO_AF_DCMI);
  GPIO_PinAFConfig(GPIOC, GPIO_PinSource7, GPIO_AF_DCMI);

 GPIO_PinAFConfig(GPIOC, GPIO_PinSource8, GPIO_AF_DCMI);

//Blue
 GPIO_PinAFConfig(GPIOC, GPIO_PinSource9, GPIO_AF_DCMI);    //??
 GPIO_PinAFConfig(GPIOE, GPIO_PinSource4, GPIO_AF_DCMI);

  GPIO_PinAFConfig(GPIOB, GPIO_PinSource6, GPIO_AF_DCMI);       //??

 GPIO_PinAFConfig(GPIOE, GPIO_PinSource5, GPIO_AF_DCMI);
 GPIO_PinAFConfig(GPIOE, GPIO_PinSource6, GPIO_AF_DCMI);


  /* DCMI GPIO configuration *****************************************************/


/*
  D0 -- PC6
  D1 -- PA10
```

```
  D2 -- PC8
  D3 -- PC9
  D4 -- PE4
  D5 -- PB6
  D6 -- PE5
  D7 -- PE6

  PCK - PA6
  HS -- PA4
  VS -- PB7
*/

  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4 | GPIO_Pin_10 ;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
  GPIO_InitStructure.GPIO_OType = GPIO_OType_OD;
  GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP ;
  GPIO_Init(GPIOA, &GPIO_InitStructure);

  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6 | GPIO_Pin_7;
  GPIO_Init(GPIOB, &GPIO_InitStructure);

  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6 | GPIO_Pin_8 | GPIO_Pin_9;
  GPIO_Init(GPIOC, &GPIO_InitStructure);

  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5 | GPIO_Pin_6 | GPIO_Pin_4;
  GPIO_Init(GPIOE, &GPIO_InitStructure);

  /* PCLK(PA6) */
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6;
  GPIO_Init(GPIOA, &GPIO_InitStructure);


  /* DCMI configuration *******************************************************/

     DCMI_InitStructure.DCMI_CaptureMode = DCMI_CaptureMode_SnapShot;
     DCMI_InitStructure.DCMI_ExtendedDataMode = DCMI_ExtendedDataMode_8b;
     DCMI_InitStructure.DCMI_CaptureRate = DCMI_CaptureRate_All_Frame;
     DCMI_InitStructure.DCMI_VSPolarity = DCMI_VSPolarity_High;
     DCMI_InitStructure.DCMI_HSPolarity = DCMI_HSPolarity_Low;
     DCMI_InitStructure.DCMI_PCKPolarity = DCMI_PCKPolarity_Rising;
     DCMI_InitStructure.DCMI_SynchroMode = DCMI_SynchroMode_Hardware;



     DCMI_Init(&DCMI_InitStructure);

  /* DCMI Interrupts config ***************************************************/
 // DCMI_ITConfig(DCMI_IT_VSYNC, ENABLE);
  //DCMI_ITConfig(DCMI_IT_LINE, ENABLE);
  //DCMI_ITConfig(DCMI_IT_FRAME, ENABLE);
  //DCMI_ITConfig(DCMI_IT_ERR, ENABLE);



  /* Configures the DMA2 to transfer Data from DCMI to the LCD ****************/
  /* Enable DMA2 clock */
  RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_DMA2, ENABLE);
```

```
  /* DMA2 Stream1 Configuration */
  DMA_DeInit(DMA2_Stream1);

 //DMA_StructInit(&DMA_InitStructure);//added

  DMA_InitStructure.DMA_Channel = DMA_Channel_1;
  DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t)(&DCMI->DR);
  DMA_InitStructure.DMA_Memory0BaseAddr = (uint32_t)outputBuffer;//
(sizeof(outputBuffer)/sizeof(uint32_t));
  DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralToMemory;
  DMA_InitStructure.DMA_BufferSize = ROW*COLUMNS*BYTESPERPIX/4;/* size of image in bytes/4 */
  DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
  DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
  DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_Word;
  DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_Word;
  DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;//DMA_Mode_Circular
  DMA_InitStructure.DMA_Priority = DMA_Priority_VeryHigh;
  DMA_InitStructure.DMA_FIFOMode = DMA_FIFOMode_Enable;
  DMA_InitStructure.DMA_FIFOThreshold = DMA_FIFOThreshold_Full;
  DMA_InitStructure.DMA_MemoryBurst = DMA_MemoryBurst_Single;
  DMA_InitStructure.DMA_PeripheralBurst = DMA_PeripheralBurst_Single;

  DMA_Init(DMA2_Stream1, &DMA_InitStructure);
 DMA_ITConfig(DMA2_Stream1, DMA_IT_TC, ENABLE);
  DMA_ITConfig(DMA2_Stream1, DMA_IT_TE, ENABLE);

 NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1);
 NVIC_InitStructure.NVIC_IRQChannel = DCMI_IRQn;
  NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 1;
  NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
  NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
  NVIC_Init(&NVIC_InitStructure);
  DCMI_ITConfig(DCMI_IT_VSYNC, ENABLE);
  DCMI_ITConfig(DCMI_IT_LINE, ENABLE);
  DCMI_ITConfig(DCMI_IT_FRAME, ENABLE);
  DCMI_ITConfig(DCMI_IT_OVF, ENABLE);
  DCMI_ITConfig(DCMI_IT_ERR, ENABLE);

  DMA_Cmd(DMA2_Stream1, ENABLE);
  DCMI_Cmd(ENABLE);

 }

 /**
  * @brief  Set PA8 Output SYSCLK/2.
  * @param  None
  * @retval None
  */
 void MCO1_Init(void)
 {
  GPIO_InitTypeDef GPIO_InitStructure;

  /* Enable GPIOs clocks */
  RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);

  /* Configure MCO (PA8) */
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
```

```
 GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
 GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
 GPIO_Init(GPIOA, &GPIO_InitStructure);

 RCC_MCO1Config(RCC_MCO1Source_HSE, RCC_MCO1Div_1);
}

/**
  * @brief  Reset the OV9655 SCCB registers.
  * @param  None
  * @retval None
  */
void DCMI_OV9655_Reset(void)
{
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS, OV9655_COM7,
SCCB_REG_RESET);
}

/**
  * @brief  Set the QVGA size(240*320).
  * @param  None
  * @retval None
  */

// uint8_t OV9655_QVGA_table[]=
// {

// };
void DCMI_OV9655_QVGASizeSetup(void)
{

  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x00, 0x00);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x01, 0x80);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x02, 0x80);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x03, 0x02);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x04, 0x00);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x09, 0x03);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x0b, 0x57);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x0e, 0x1);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x0f, 0xc0);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x10, 0x50);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x11, 0x80);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x12, 0x63);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x13, 0xef);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x14, 0x3a);
```

```
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x15, 0x18);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x16, 0x24);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x17, 0x18);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x18, 0x04);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x19, 0x01);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x1a, 0x81);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x1e, 0x00); /*0x20*/
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x24, 0x3c);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x25, 0x36);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x26, 0x72);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x27, 0x08);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x28, 0x08);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x29, 0x15);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x2a, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x2b, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x2c, 0x08);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x32, 0x12);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x33, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x34, 0x3f);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x35, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x36, 0x3a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x38, 0x72);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x39, 0x57);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x3a, 0xca);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x3b, 0x04);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x3d, 0x99);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x3e, 0x02);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x3f, 0xc1);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x40, 0xd0);
Delay(TIMEOUT);
```

```
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x41, 0x41);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x42, 0xc0);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x43, 0x0a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x44, 0xf0);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x45, 0x46);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x46, 0x62);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x47, 0x2a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x48, 0x3c);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x4a, 0xfc);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x4b, 0xfc);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x4c, 0x7f);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x4d, 0x7f);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x4e, 0x7f);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x4f, 0x98);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x50, 0x98);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x51, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x52, 0x28);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x53, 0x70);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x54, 0x98);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x58, 0x1a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x59, 0x85);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x5a, 0xa9);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x5b, 0x64);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x5c, 0x84);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x5d, 0x53);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x5e, 0x0e);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x5f, 0xf0);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x60, 0xf0);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x61, 0xf0);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x62, 0x00);
```

```
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x63, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x64, 0x02);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x65, 0x20);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x66, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x69, 0x0a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x6b, 0x0a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x6c, 0x04);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x6d, 0x55);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x6e, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x6f, 0x9d);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x70, 0x21);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x71, 0x78);
Delay(TIMEOUT);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x72, 0x11);
Delay(TIMEOUT);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x73, 0x01);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x74, 0x10);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x75, 0x10);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x76, 0x01);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x77, 0x02);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x7A, 0x12);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x7B, 0x08);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x7C, 0x16);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x7D, 0x30);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x7E, 0x5e);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x7F, 0x72);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x80, 0x82);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x81, 0x8e);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x82, 0x9a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x83, 0xa4);
Delay(TIMEOUT);
```

```
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x84, 0xac);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x85, 0xb8);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x86, 0xc3);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x87, 0xd6);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x88, 0xe6);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x89, 0xf2);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x8a, 0x24);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x8c, 0x80);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x90, 0x7d);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x91, 0x7b);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x9d, 0x02);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x9e, 0x02);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x9f, 0x7a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xa0, 0x79);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xa1, 0x1f);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xa4, 0x50);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xa5, 0x68);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xa6, 0x4a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xa8, 0xc1);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xa9, 0xef);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xaa, 0x92);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xab, 0x04);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xac, 0x80);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xad, 0x80);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xae, 0x80);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xaf, 0x80);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xb2, 0xf2);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xb3, 0x20);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xb4, 0x20);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xb5, 0x00);
```

```
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xb6, 0xaf);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xb6, 0xaf);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xbb, 0xae);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xbc, 0x7f);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xbd, 0x7f);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xbe, 0x7f);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xbf, 0x7f);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xbf, 0x7f);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xc0, 0xaa);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xc1, 0xc0);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xc2, 0x01);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xc3, 0x4e);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xc6, 0x05);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xc7, 0x81);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xc9, 0xe0);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xca, 0xe8);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xcb, 0xf0);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xcc, 0xd8);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xcd, 0x93);
  Delay(TIMEOUT);
}

/**
  * @brief  Set the QQVGA size(120*160).
  * @param  None
  * @retval None
  */
void DCMI_OV9655_QQVGASizeSetup(void)
{


  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x00, 0x00);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x01, 0x80);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x02, 0x80);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x03, 0x02);
  Delay(TIMEOUT);
```

```
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x04, 0x03);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x09, 0x01);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x0b, 0x57);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x0e, 0x61);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x0f, 0x40);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x11, 0x01);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x12, 0x62);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x13, 0xc7);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x14, 0x3a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x15, 0x08);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x16, 0x24);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x17, 0x18);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x18, 0x04);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x19, 0x01);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x1a, 0x81);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x1e, 0x20);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x24, 0x3c);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x25, 0x36);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x26, 0x72);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x27, 0x08);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x28, 0x08);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x29, 0x15);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x2a, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x2b, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x2c, 0x08);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x32, 0xa4);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x33, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x34, 0x3f);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x35, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x36, 0x3a);
```

```
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x38, 0x72);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x39, 0x57);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x3a, 0xcc);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x3b, 0x04);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x3d, 0x99);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x3e, 0x0e);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x3f, 0xc1);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x40, 0xc0);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x41, 0x41);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x42, 0xc0);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x43, 0x0a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x44, 0xf0);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x45, 0x46);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x46, 0x62);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x47, 0x2a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x48, 0x3c);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x4a, 0xfc);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x4b, 0xfc);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x4c, 0x7f);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x4d, 0x7f);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x4e, 0x7f);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x4f, 0x98);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x50, 0x98);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x51, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x52, 0x28);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x53, 0x70);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x54, 0x98);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x58, 0x1a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x59, 0x85);
Delay(TIMEOUT);
```

```
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x5a, 0xa9);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x5b, 0x64);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x5c, 0x84);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x5d, 0x53);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x5e, 0x0e);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x5f, 0xf0);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x60, 0xf0);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x61, 0xf0);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x62, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x63, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x64, 0x02);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x65, 0x20);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x66, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x69, 0x0a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x6b, 0x1a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x6c, 0x04);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x6d, 0x55);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x6e, 0x00);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x6f, 0x9d);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x70, 0x21);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x71, 0x78);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x72, 0x22);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x73, 0x02);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x74, 0x10);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x75, 0x10);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x76, 0x01);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x77, 0x02);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x7A, 0x12);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x7B, 0x08);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x7C, 0x16);
```

```
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x7D, 0x30);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x7E, 0x5e);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x7F, 0x72);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x80, 0x82);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x81, 0x8e);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x82, 0x9a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x83, 0xa4);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x84, 0xac);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x85, 0xb8);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x86, 0xc3);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x87, 0xd6);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x88, 0xe6);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x89, 0xf2);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x8a, 0x24);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x8c, 0x80);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x90, 0x7d);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x91, 0x7b);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x9d, 0x02);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x9e, 0x02);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0x9f, 0x7a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xa0, 0x79);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xa1, 0x40);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xa4, 0x50);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xa5, 0x68);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xa6, 0x4a);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xa8, 0xc1);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xa9, 0xef);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xaa, 0x92);
Delay(TIMEOUT);
DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xab, 0x04);
Delay(TIMEOUT);
```

```
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xac, 0x80);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xad, 0x80);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xae, 0x80);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xaf, 0x80);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xb2, 0xf2);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xb3, 0x20);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xb4, 0x20);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xb5, 0x00);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xb6, 0xaf);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xb6, 0xaf);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xbb, 0xae);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xbc, 0x7f);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xbd, 0x7f);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xbe, 0x7f);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xbf, 0x7f);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xbf, 0x7f);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xc0, 0xaa);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xc1, 0xc0);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xc2, 0x01);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xc3, 0x4e);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xc6, 0x05);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xc7, 0x82);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xc9, 0xe0);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xca, 0xe8);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xcb, 0xf0);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xcc, 0xd8);
  Delay(TIMEOUT);
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS,0xcd, 0x93);
  Delay(TIMEOUT);
}

/**
  * @brief  Read the OV9655 Manufacturer identifier.
  * @param  OV9655ID: pointer to the OV9655 Manufacturer identifier.
```

```
  * @retval None
  */
void DCMI_OV9655_ReadID(OV9655_IDTypeDef* OV9655ID)
{
 uint8_t temp;
 DCMI_SingleRandomRead(OV9655_DEVICE_WRITE_ADDRESS, OV9655_MIDH , &temp);
 OV9655ID->Manufacturer_ID1 = temp;
 DCMI_SingleRandomRead(OV9655_DEVICE_WRITE_ADDRESS, OV9655_MIDL , &temp);
 OV9655ID->Manufacturer_ID2 = temp;
 DCMI_SingleRandomRead(OV9655_DEVICE_WRITE_ADDRESS, OV9655_VER , &temp);
 OV9655ID->Version = temp;
 DCMI_SingleRandomRead(OV9655_DEVICE_WRITE_ADDRESS, OV9655_PID , &temp);
 OV9655ID->PID = temp;
}

/**
  * @brief  Set the Internal Clock Prescaler.
  * @param  OV9655_Prescaler: the new value of the prescaler.
  *         This parameter can be a value between 0x0 and 0x1F
  * @retval None
  */
void DCMI_OV9655_SetPrescaler(uint8_t OV9655_Prescaler)
{
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS, OV9655_CLKRC,
OV9655_Prescaler);
}

/**
  * @brief  Select the Output Format.
  * @param  OV9655_OuputFormat: the Format of the ouput Data.
  *         This parameter can be one of the following values:
  *           @arg OUTPUT_FORMAT_RAWRGB_DATA
  *           @arg OUTPUT_FORMAT_RAWRGB_INTERP
  *           @arg OUTPUT_FORMAT_YUV
  *           @arg OUTPUT_FORMAT_RGB
  * @retval None
  */
void DCMI_OV9655_SelectOutputFormat(uint8_t OV9655_OuputFormat)
{
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS, OV9655_COM7,
OV9655_OuputFormat);
}

/**
  * @brief  Select the Output Format Resolution.
  * @param  OV9655_FormatResolution: the Resolution of the ouput Data.
  *         This parameter can be one of the following values:
  *           @arg FORMAT_CTRL_15fpsVGA
  *           @arg FORMAT_CTRL_30fpsVGA_NoVArioPixel
  *           @arg FORMAT_CTRL_30fpsVGA_VArioPixel
  * @retval None
  */
void DCMI_OV9655_SelectFormatResolution(uint8_t OV9655_FormatResolution)
{
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS, OV9655_COM7,
OV9655_FormatResolution);
}
```

```
/**
  * @brief  Set the new value of OV9655 registers
  * @param  OV9655_Register: the OV9655 Register to be configured.
  * @param  Register_Val: The new value to be set
  * @retval None
  */
void DCMI_OV9655_SetRegister(uint8_t OV9655_Register, uint8_t Register_Val)
{
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS, OV9655_Register, Register_Val);
}

/**
  * @brief  Select the HREF Control signal option
  * @param  OV9665_HREFControl: the HREF Control signal option.
  *         This parameter can be one of the following value:
  *           @arg OV9665_HREFControl_Opt1: HREF edge offset to data output.
  *           @arg OV9665_HREFControl_Opt2: HREF end 3 LSB
  *           @arg OV9665_HREFControl_Opt3: HREF start 3 LSB
  * @retval None
  */
void DCMI_OV9655_HREFControl(uint8_t OV9665_HREFControl)
{
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS, OV9655_HREF,
OV9665_HREFControl);
}

/**
  * @brief  Select the RGB format option
  * @param  OV9665_RGBOption: the RGB Format option.
  *         This parameter can be one of the following value:
  *           @arg RGB_NORMAL
  *           @arg RGB_565
  *           @arg RGB_555
  * @retval None
  */
void DCMI_OV9655_SelectRGBOption(uint8_t OV9665_RGBOption)
{
  DCMI_SingleRandomWrite(OV9655_DEVICE_WRITE_ADDRESS, OV9655_COM15,
OV9665_RGBOption);
}


/**
  * @}
  */
static void Delay(uint32_t nTime)
{
  while(nTime--);
}

static void Delay_ms(uint32_t nTime)
{
  while(nTime--)
   {Delay(1000);}

}

uint32_t Buffervalue(uint32_t index)
{
```

```
 return outputBuffer[index];
}
```

**********The result part **********