

Hi,

I adjusts 8-bit SPI clock 16 pals with stm32cubemx but what I see in the oscilloscope this problem could that be?

MCU STM32F302RCxx

```

/**
*****
* File Name      : main.c
* Description    : Main program body
*****
*
* COPYRIGHT(c) 2015 STMicroelectronics
*
* Redistribution and use in source and binary forms, with or without modification,
* are permitted provided that the following conditions are met:
* 1. Redistributions of source code must retain the above copyright notice,
*    this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice,
*    this list of conditions and the following disclaimer in the documentation
*    and/or other materials provided with the distribution.
* 3. Neither the name of STMicroelectronics nor the names of its contributors
*    may be used to endorse or promote products derived from this software
*    without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*****
*/
/* Includes -----*/
#include "stm32f3xx_hal.h"

/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

/* Private variables -----*/
ADC_HandleTypeDef hadc1;

SPI_HandleTypeDef hspi3;

TIM_HandleTypeDef htim6;

UART_HandleTypeDef huart1;
UART_HandleTypeDef huart3;

/* USER CODE BEGIN PV */
/* Private variables -----*/

#define KEY_OUT1_1      GPIOC->BSRRL = GPIO_BSRR_BS_4
#define KEY_OUT1_0      GPIOC->BSRRH = GPIO_BSRR_BS_4

#define KEY_OUT2_1      GPIOC->BSRRL = GPIO_BSRR_BS_5
#define KEY_OUT2_0      GPIOC->BSRRH = GPIO_BSRR_BS_5

#define KEY_OUT3_1      GPIOB->BSRRL = GPIO_BSRR_BS_0

```

```
#define KEY_OUT3_0      GPIOB->BSRRH = GPIO_BSRR_BS_0

#define KEY_OUT4_1      GPIOB->BSRRL = GPIO_BSRR_BS_1
#define KEY_OUT4_0      GPIOB->BSRRH = GPIO_BSRR_BS_1

#define SW_1            ((GPIOC->IDR & 0x0002)>>1)

unsigned char ms_data=0;

/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_ADC1_Init(void);
static void MX_SPI3_Init(void);
static void MX_TIM6_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_USART3_UART_Init(void);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----*/

void delay_ms(unsigned int ms);

/* USER CODE END PFP */

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* Configure the system clock */
    SystemClock_Config();

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_ADC1_Init();
    MX_SPI3_Init();
    MX_TIM6_Init();
    MX_USART1_UART_Init();
    MX_USART3_UART_Init();

    /* USER CODE BEGIN 2 */

    KEY_OUT1_0;
    KEY_OUT2_1;
    KEY_OUT3_1;
    KEY_OUT4_1;

    SPI3->CR1 &= ~SPI_CR1_SPE;          /* Disable SPI1 */
```

```
// SPI3->CR2 |= SPI_CR2_DS_2 | SPI_CR2_DS_1 | SPI_CR2_DS_0; //0111: 8-bit for DS

GPIOC->BSRRL = GPIO_BSRR_BS_13; delay_ms(1000); GPIOC->BSRRH = GPIO_BSRR_BS_13; delay_ms(1000);
GPIOC->BSRRL = GPIO_BSRR_BS_13; delay_ms(1000); GPIOC->BSRRH = GPIO_BSRR_BS_13;
delay_ms(1000);

SPI3->CR1 |= SPI_CR1_SPE;

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    if (SW_1==0)
    {

        GPIOC->BSRRL = GPIO_BSRR_BS_13; delay_ms(1000);

        ms_data=0xAA;

        SPI3->DR = ms_data;
        while((SPI3->SR & SPI_SR_RXNE) != SPI_SR_RXNE){}

        GPIOC->BSRRH = GPIO_BSRR_BS_13;

    }

/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */

}
/* USER CODE END 3 */
}

/** System Clock Configuration
*/
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct;
    RCC_ClkInitTypeDef RCC_ClkInitStruct;
    RCC_PeriphCLKInitTypeDef PeriphClkInit;

    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.HSEPredivValue = RCC_HSE_PREDIV_DIV1;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL9;
    HAL_RCC_OscConfig(&RCC_OscInitStruct);

    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_SYSCLK|RCC_CLOCKTYPE_PCLK1
        |RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;
```

```
HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2);

PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_USART1|RCC_PERIPHCLK_USART3
    |RCC_PERIPHCLK_ADC12;
PeriphClkInit.Usart1ClockSelection = RCC_USART1CLKSOURCE_PCLK2;
PeriphClkInit.Usart3ClockSelection = RCC_USART3CLKSOURCE_PCLK1;
PeriphClkInit.Adc12ClockSelection = RCC_ADC12PLLCLK_DIV1;
HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit);

HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);

HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);
}

/* ADC1 init function */
void MX_ADC1_Init(void)
{
    ADC_ChannelConfTypeDef sConfig;

    /**Common config
    */
    hadc1.Instance = ADC1;
    hadc1.Init.ClockPrescaler = ADC_CLOCK_ASYNC;
    hadc1.Init.Resolution = ADC_RESOLUTION10b;
    hadc1.Init.ScanConvMode = ADC_SCAN_DISABLE;
    hadc1.Init.ContinuousConvMode = ENABLE;
    hadc1.Init.DiscontinuousConvMode = DISABLE;
    hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
    hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
    hadc1.Init.NbrOfConversion = 1;
    hadc1.Init.DMAContinuousRequests = DISABLE;
    hadc1.Init.EOCSelection = EOC_SINGLE_CONV;
    hadc1.Init.LowPowerAutoWait = DISABLE;
    hadc1.Init.Overrun = OVR_DATA_OVERWRITTEN;
    HAL_ADC_Init(&hadc1);

    /**Configure Regular Channel
    */
    sConfig.Channel = ADC_CHANNEL_6;
    sConfig.Rank = 1;
    sConfig.SingleDiff = ADC_SINGLE_ENDED;
    sConfig.SamplingTime = ADC_SAMPLETIME_2CYCLES_5;
    sConfig.OffsetNumber = ADC_OFFSET_NONE;
    sConfig.Offset = 0;
    HAL_ADC_ConfigChannel(&hadc1, &sConfig);
}

/* SPI3 init function */
void MX_SPI3_Init(void)
{
    hspi3.Instance = SPI3;
    hspi3.Init.Mode = SPI_MODE_MASTER;
    hspi3.Init.Direction = SPI_DIRECTION_2LINES;
    hspi3.Init.DataSize = SPI_DATASIZE_8BIT;
    hspi3.Init.CLKPolarity = SPI_POLARITY_LOW;
    hspi3.Init.CLKPhase = SPI_PHASE_1EDGE;
    hspi3.Init.NSS = SPI_NSS_SOFT;
    hspi3.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_64;
    hspi3.Init.FirstBit = SPI_FIRSTBIT_MSB;
    hspi3.Init.TIMode = SPI_TIMODE_DISABLED;
    hspi3.Init.CRCCalculation = SPI_CRCCALCULATION_ENABLED;
    hspi3.Init.CRCPolynomial = 50;
    hspi3.Init.CRCLength = SPI_CRC_LENGTH_8BIT;
    hspi3.Init.NSSPMode = SPI_NSS_PULSE_DISABLED;
}
```

```
    HAL_SPI_Init(&hspi3);
}

/* TIM6 init function */
void MX_TIM6_Init(void)
{
    TIM_MasterConfigTypeDef sMasterConfig;

    htim6.Instance = TIM6;
    htim6.Init.Prescaler = 0;
    htim6.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim6.Init.Period = 359;
    HAL_TIM_Base_Init(&htim6);

    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    HAL_TIMEx_MasterConfigSynchronization(&htim6, &sMasterConfig);
}

/* USART1 init function */
void MX_USART1_UART_Init(void)
{
    huart1.Instance = USART1;
    huart1.Init.BaudRate = 115200;
    huart1.Init.WordLength = UART_WORDLENGTH_8B;
    huart1.Init.StopBits = UART_STOPBITS_1;
    huart1.Init.Parity = UART_PARITY_NONE;
    huart1.Init.Mode = UART_MODE_TX_RX;
    huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart1.Init.OverSampling = UART_OVERSAMPLING_16;
    huart1.Init.OneBitSampling = UART_ONEBIT_SAMPLING_DISABLED ;
    huart1.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
    HAL_UART_Init(&huart1);
}

/* USART3 init function */
void MX_USART3_UART_Init(void)
{
    huart3.Instance = USART3;
    huart3.Init.BaudRate = 38400;
    huart3.Init.WordLength = UART_WORDLENGTH_8B;
    huart3.Init.StopBits = UART_STOPBITS_1;
    huart3.Init.Parity = UART_PARITY_NONE;
    huart3.Init.Mode = UART_MODE_TX_RX;
    huart3.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart3.Init.OverSampling = UART_OVERSAMPLING_16;
    huart3.Init.OneBitSampling = UART_ONEBIT_SAMPLING_DISABLED ;
    huart3.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
    HAL_UART_Init(&huart3);
}

/** Configure pins as
    * Analog
    * Input
    * Output
    * EVENT_OUT
    * EXTI
*/
void MX_GPIO_Init(void)
{
```

```
GPIO_InitTypeDef GPIO_InitStructure;

/* GPIO Ports Clock Enable */
__GPIOC_CLK_ENABLE();
__GPIOF_CLK_ENABLE();
__GPIOA_CLK_ENABLE();
__GPIOB_CLK_ENABLE();
__GPIOD_CLK_ENABLE();

/*Configure GPIO pins : PC13 PC14 PC4 PC5
                        PC6 PC8 PC9 PC10
                        PC11 PC12 */
GPIO_InitStructure.Pin = GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_4|GPIO_PIN_5
                        |GPIO_PIN_6|GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10
                        |GPIO_PIN_11|GPIO_PIN_12;
GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStructure.Pull = GPIO_NOPULL;
GPIO_InitStructure.Speed = GPIO_SPEED_LOW;
HAL_GPIO_Init(GPIOC, &GPIO_InitStructure);

/*Configure GPIO pins : PC15 PC1 PC2 PC3
                        PC7 */
GPIO_InitStructure.Pin = GPIO_PIN_15|GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3
                        |GPIO_PIN_7;
GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
GPIO_InitStructure.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOC, &GPIO_InitStructure);

/*Configure GPIO pins : PA0 PA1 PA2 PA3
                        PA4 PA5 PA6 PA7
                        PA8 PA11 PA15 */
GPIO_InitStructure.Pin = GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3
                        |GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7
                        |GPIO_PIN_8|GPIO_PIN_11|GPIO_PIN_15;
GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStructure.Pull = GPIO_NOPULL;
GPIO_InitStructure.Speed = GPIO_SPEED_LOW;
HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);

/*Configure GPIO pins : PB0 PB1 PB15 PB8
                        PB9 */
GPIO_InitStructure.Pin = GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_15|GPIO_PIN_8
                        |GPIO_PIN_9;
GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStructure.Pull = GPIO_NOPULL;
GPIO_InitStructure.Speed = GPIO_SPEED_LOW;
HAL_GPIO_Init(GPIOB, &GPIO_InitStructure);

/*Configure GPIO pins : PB2 PB12 PB13 PB14
                        PB6 PB7 */
GPIO_InitStructure.Pin = GPIO_PIN_2|GPIO_PIN_12|GPIO_PIN_13|GPIO_PIN_14
                        |GPIO_PIN_6|GPIO_PIN_7;
GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
GPIO_InitStructure.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOB, &GPIO_InitStructure);

/*Configure GPIO pin : PA12 */
GPIO_InitStructure.Pin = GPIO_PIN_12;
GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
GPIO_InitStructure.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);

/*Configure GPIO pin : PD2 */
GPIO_InitStructure.Pin = GPIO_PIN_2;
GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStructure.Pull = GPIO_NOPULL;
GPIO_InitStructure.Speed = GPIO_SPEED_LOW;
HAL_GPIO_Init(GPIOD, &GPIO_InitStructure);
```

```
}

/* USER CODE BEGIN 4 */

void delay_ms(unsigned int ms)
{
    unsigned int i, j;
    for(i=0; i<ms; i++)
    {
        for(j=0; j<3598; j++)
        {
            __nop();
        }
    }
}

/* USER CODE END 4 */

#ifdef USE_FULL_ASSERT

/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t* file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line number,
     ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
    /* USER CODE END 6 */
}

#endif

/**
 * @}
 */

/**
 * @}
 */

/***** (C) COPYRIGHT STMicroelectronics *****/
```

