

DMX

- We want to use an STM32G071K8 as a DMX decoder in our product.
- An UART interface is used to get DMX data.
 - I am used to use the Frame Error detection flag as a “Break” detection (new frame start)
 - Then I launch the DMA to get the full frame.
 - The next frame error or the idle (frame error is more accurate for DMX) is used as end of frame and the got data can be used by the firmware.

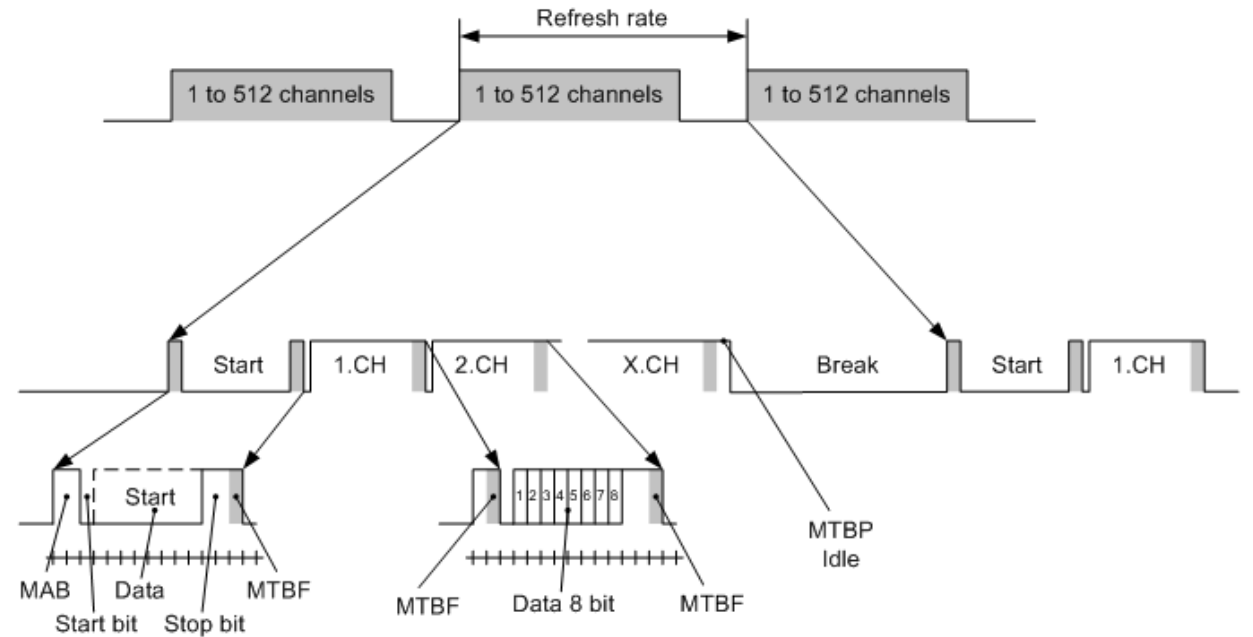


Fig 0: DMX frames

Code

- UART, Clock (default 16Mhz) and one test output IO are configured with STMCube.
- In this test, TEST_IO should toggle every time we have a frame error, which should be for every “Break” in the DMX frame.
- I have 2 DMX consoles to make this test :
 - a small one which only send 30 bytes
 - A classic one which send 513 bytes.
- Since DMX consoles can have different output signal, I use a CRMX receiver to get the data.

```
void init_uart3(void){

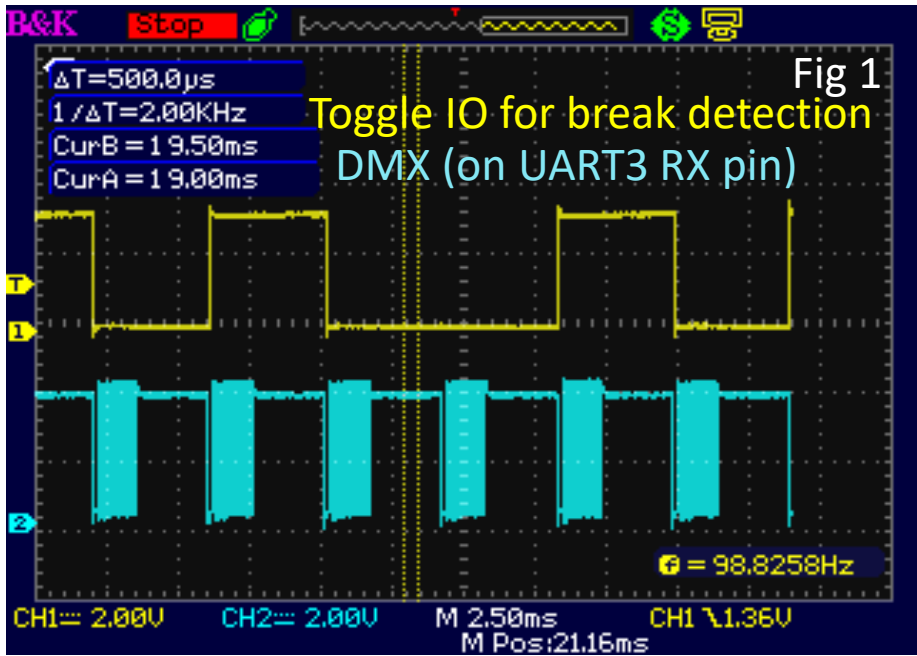
/* DMA controller clock enable */
__HAL_RCC_DMA1_CLK_ENABLE();

/* DMA interrupt init */
/* DMA1_Channel1_IRQn interrupt configuration */
HAL_NVIC_SetPriority(DMA1_Channel1_IRQn, 0, 0);
HAL_NVIC_EnableIRQ(DMA1_Channel1_IRQn);

huart3.Instance = USART3;
huart3.Init.BaudRate = 250000;
huart3.Init.WordLength = UART_WORDLENGTH_8B;
huart3.Init.StopBits = UART_STOPBITS_2;
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_ONE_BIT_SAMPLE_DISABLE;
huart3.Init.ClockPrescaler = UART_PRESCALER_DIV1;
huart3.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
if (HAL_UART_Init(&huart3) != HAL_OK){
    Error_Handler();
}
__HAL_UART_ENABLE_IT(&huart3,UART_IT_ERR);

}

void USART3_4_LPUART1_IRQHandler(void)
{
    volatile static uint8_t uartData;
    if((USART3->ISR & USART_ISR_FE) == USART_ISR_FE){
        HAL_GPIO_TogglePin(OUT_TEST_GPIO_Port,OUT_TEST_Pin);
    }
    USART3->ICR |= (3<<1);
    USART3->ICR |= (4<<1);
    USART3->ICR |= (1<<1);
    uartData = USART3->RDR;    // clear data flag
}
```



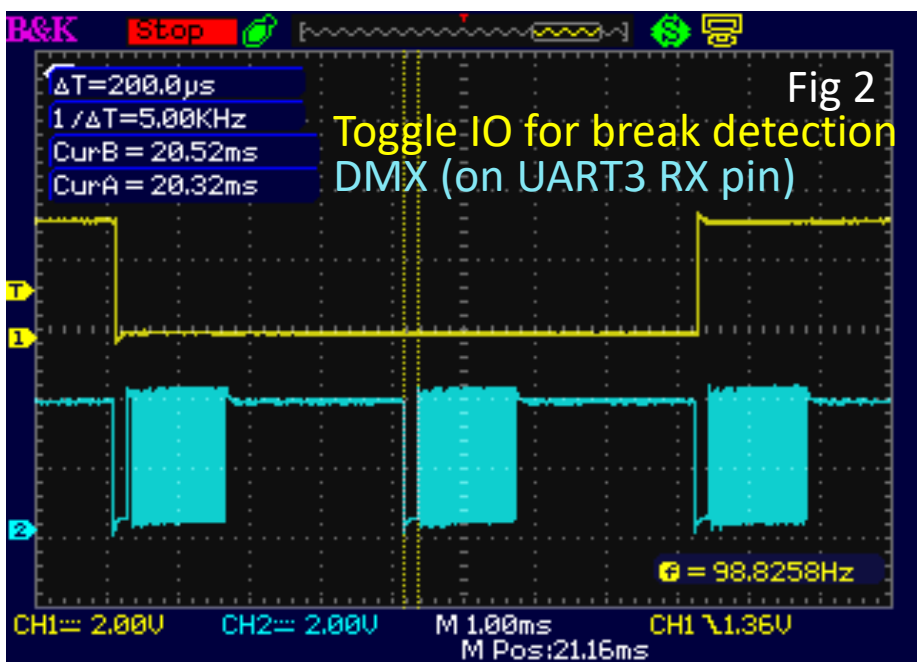
SWIT controller used with CRMX adapter

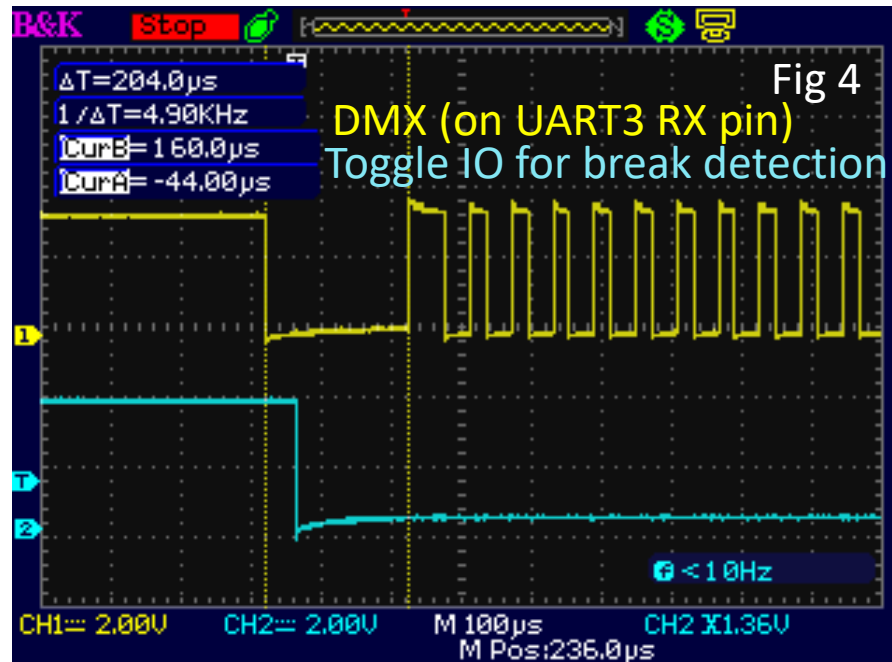
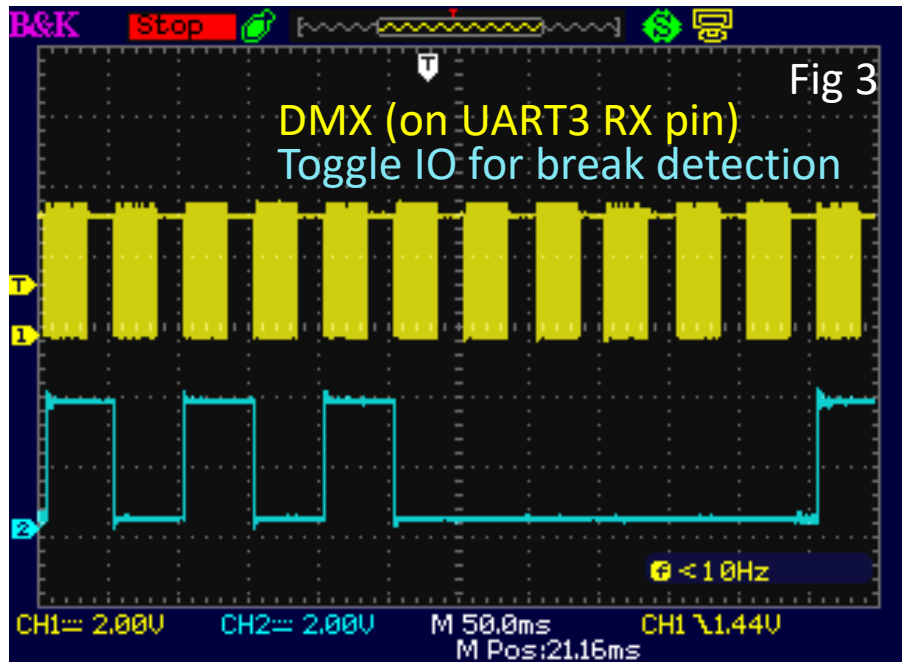
Fig 1:
DMX input on UART 3 RX pin: 1 break then 30 bytes in that case

The yellow part shows the TEST IO which toggle every time we enter the UART interrupt with FE flag (to detect the break).

But strangely, it does not do it at every break why ?

Fig2 show a zoom on a failing break detection (yellow does not toggle between the two cursors)





DMX console used with CRMX adapter

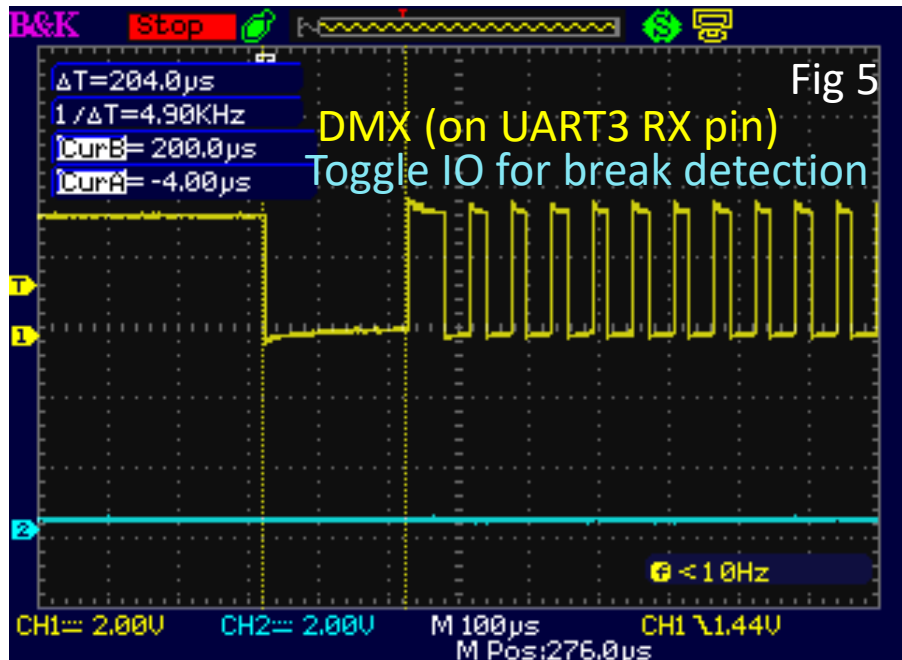


Fig 3:

DMX input on UART 3 RX pin: 1 break then 513 bytes in that case

The blue part shows an IO which toggle every time we enter the UART interrupt with FE flag (to detect the break).

But strangely, it does not do it at every break why ?

Fig4 shows a zoom on a normal break detection (blue does not toggle between the two cursors)

Fig5 shows a zoom on a failing break detection (blue does not toggle between the two cursors)

Frame Error detection

- With both DMX console I have different results, I have tested with 2 more and I have the same issue :
 - with small frame length, the Frame Error flag is detected most of the time.
 - 1 miss time to time but not a huge deal since the DMX is a very redundant protocole.
 - With full DMX frame length, the frame error flag is only detected time to time
 - We cannot have a fluid light effect on our product since a lot of DMX frame are not captured.
- The frame error detection seems not accurate with STM32G071K8
 - I did not find any information yet on any forum.
 - **Is it a known issue from ST ?**