Hello,

I'm tring to read and write data from a SPI Flash Memory M25P40 with an STM32F103VDT

As first try, I started to read the flash memory serial number.

I used Cube32Mx to generate the initialization code.

This is the SPI3 initialisation function:

```
static 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_HARD_OUTPUT;
  hspi3.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_8;
  hspi3.Init.FirstBit = SPI_FIRSTBIT_MSB;
  hspi3.Init.TIMode = SPI_TIMODE_DISABLE;
  hspi3.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
  hspi3.Init.CRCPolynomial = 10;
  if (HAL_SPI_Init(&hspi3) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

}??????????????????????
```

I configure the SPI to use the Interrupt and this is my SPI code:

```
uint8_t SPI_Read( SPI_Channel_t channel, uint8_t* buffer, uint16_t bufferSize, uint32_t timeout )
{
  uint8_t opResult;

  opResult  = ( HAL_SPI_Receive_IT( SPI_Handlers[channel], buffer, bufferSize ) == HAL_OK ) ? TRUE : FALSE;
  opResult &= (osSemaphoreWait( SPI_RX_SemaphoreId[channel], timeout ) == osOK ? TRUE : FALSE );

  return opResult;
}


uint8_t SPI_Write( SPI_Channel_t channel, uint8_t* buffer, uint16_t bufferSize, uint32_t timeout )
{
  uint8_t opResult;

  opResult  = ( HAL_SPI_Transmit_IT(SPI_Handlers[channel], buffer, bufferSize) == HAL_OK ) ? TRUE : FALSE;
  opResult &= ( osSemaphoreWait( SPI_TX_SemaphoreId[channel], timeout ) == osOK ? TRUE : FALSE );

  return opResult;
}

uint8_t SPI_WriteRead( SPI_Channel_t channel, uint8_t* bufferIn, OUT uint8_t* bufferOut, uint16_t bufferSize, uint32_t timeout )
{
  uint8_t opResult;

  opResult  = ( HAL_SPI_TransmitReceive_IT(SPI_Handlers[channel], bufferIn, bufferOut, bufferSize) == HAL_OK ) ? TRUE : FALSE;
  opResult &= ( osSemaphoreWait( SPI_TXRX_SemaphoreId[channel], timeout ) == HAL_OK ? TRUE : FALSE );

  return opResult;
}

/*========== Interrupt callback ==========*/

void HAL_SPI_TxCpltCallback(SPI_HandleTypeDef *hspi)
{
  SPI_Channel_t channel;

        if( hspi == SPI_Handlers[SPI_BLUETOOTH] ) { channel = SPI_BLUETOOTH; }
  else  if( hspi == SPI_Handlers[SPI_EFLASH]    ) { channel = SPI_EFLASH;    }
  else
  {
    /* TODO: error */
    LOG_Print( LOG_SOURCE_SPI, LOG_LEVEL_ERROR, "[SPI] Tx cb unknown channel");
    return;
  }

  osSemaphoreRelease( SPI_TX_SemaphoreId[channel] );
}


void HAL_SPI_RxCpltCallback(SPI_HandleTypeDef *hspi)
{
  SPI_Channel_t channel;
```

```
        if( hspi == SPI_Handlers[SPI_BLUETOOTH] ) { channel = SPI_BLUETOOTH; }
  else  if( hspi == SPI_Handlers[SPI_EFLASH]    ) { channel = SPI_EFLASH;    }
  else
  {
    /* TODO: error */
    LOG_Print( LOG_SOURCE_SPI, LOG_LEVEL_ERROR, "[SPI] Rx cb unknown channel");
    return;
  }

  osSemaphoreRelease( SPI_RX_SemaphoreId[channel] );
}


void HAL_SPI_TxRxCpltCallback(SPI_HandleTypeDef *hspi)
{
  SPI_Channel_t channel;

        if( hspi == SPI_Handlers[SPI_BLUETOOTH] ) { channel = SPI_BLUETOOTH; }
  else  if( hspi == SPI_Handlers[SPI_EFLASH]    ) { channel = SPI_EFLASH;    }
  else
  {
    /* TODO: error */
    LOG_Print( LOG_SOURCE_SPI, LOG_LEVEL_ERROR, "[SPI] TxRx cb unknown channel");
    return;
  }

  osSemaphoreRelease( SPI_TXRX_SemaphoreId[channel]);
}????????????????????????????????????????????????????????????????????????????????????????????
```

As you can see in the above code, the semaphore is released by the Tx/Rx/TxRx interrupt.

When I try to perform a simple serial number read, the Rx Interrupt is not raised.

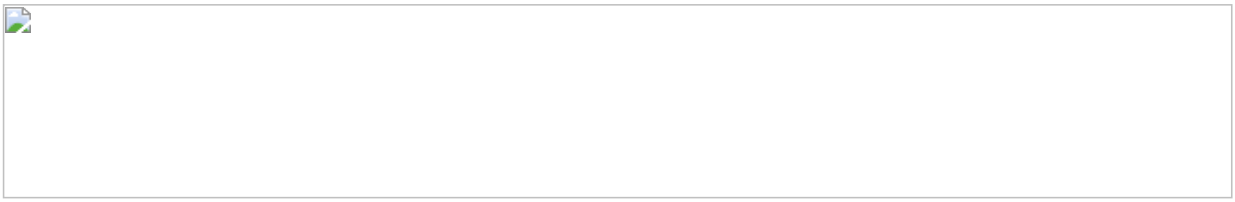
This is the code to read the flash memory serial number:

```
opResult = SPI_Write( SPI_EFLASH, 0x9F, 1, EFLASH_WRITE_COMMAND_TIMEOUT );

/* The serial number has 20 bytes */
opResult =  SPI_Read( SPI_EFLASH, buffer, 20, EFLASH_READ_COMMAND_TIMEOUT );????
```

Unfortunately, the read operation fails dut to semaphore timeout. This happens because the Rx interrupt is not raised.

I also the SPI communication with a logic analyzer and the serial number bytes are exchanged on the SPI ad you can see in the below image:



Any suggestion?

I checked everything... But maybe I forget something.

Thanks!