

# 펌웨어 개발 및 회로 설계 기초

-4-

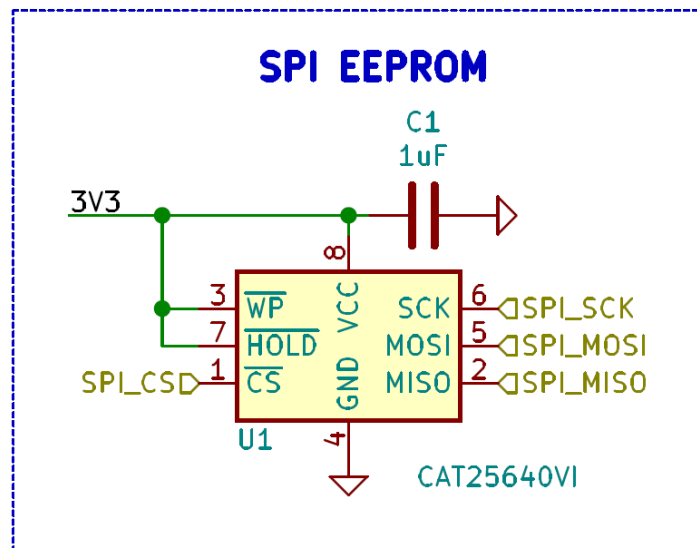
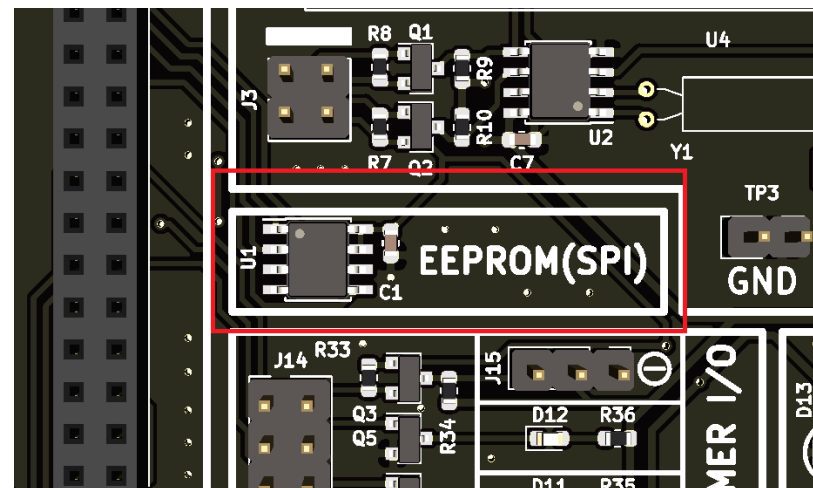
2019-02-21

# 1. SPI

# 1. SPI

## • EEPROM

- EEPROM이란 저장장치의 일종으로 비휘발성이다.
- 보드에는 현재 CAT25640이란 EEPROM이 부착되어있고, 다음과 같은 연결을 가지고 있다.

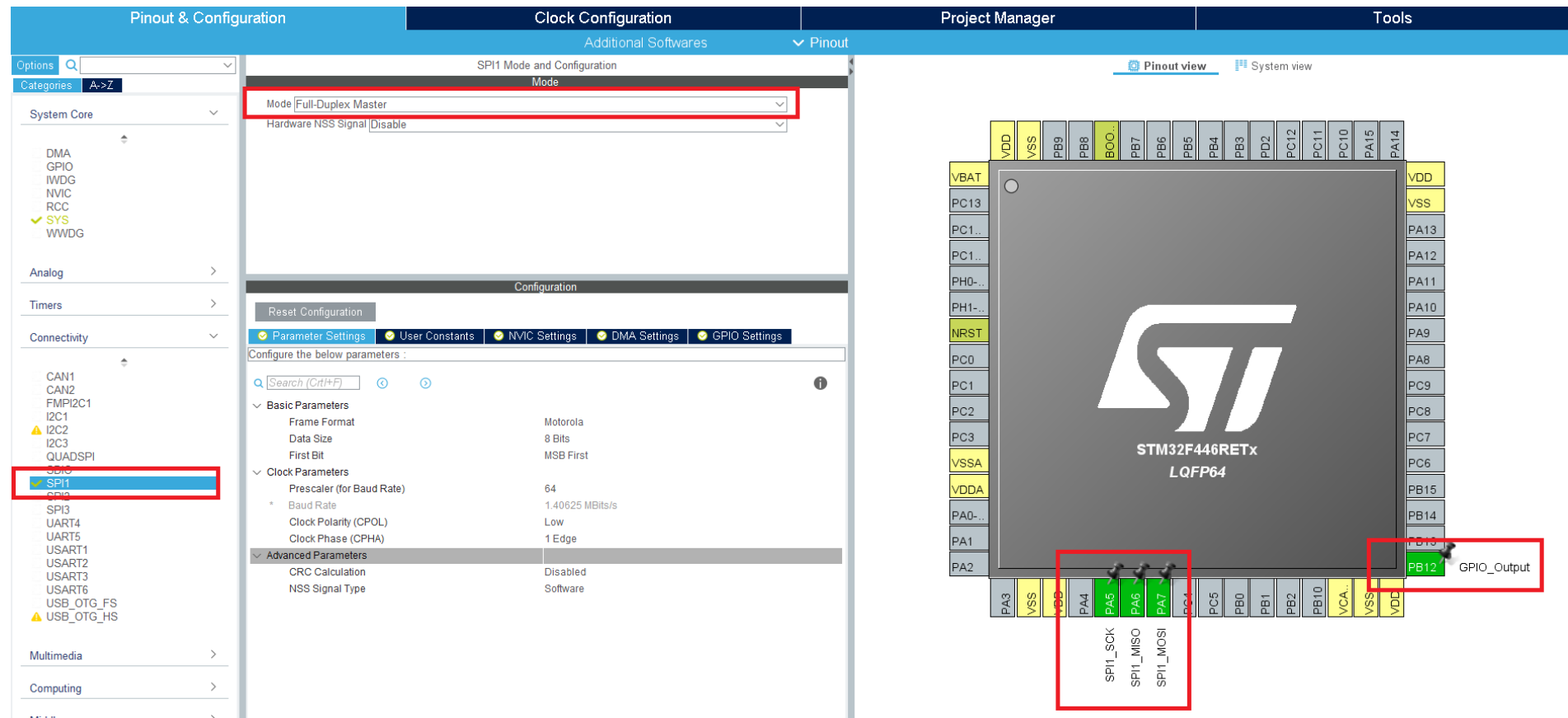


PC9	R1	R2	PC8
PB8	R3	R4	PC6
PB9	R5	R6	PC5
	R7	R8	
	R9	R10	PD8
SPI_SCK	PA5	R11	PA12
SPI_MISO	PA6	R13	PA11
SPI_MOSI	PA7	R15	PB12
PB6	R17	R18	PB11
PC7	R19	R20	
ENC_SW	PA9	R21	PB2
PWM_A	PA8	R23	PB1
I2C_SCL	PB10	R25	PB15
ENC_CH1	PB4	R27	PB14
ENC_CH2	PB5	R29	PB13
I2C_SDA	PB3	R31	
PA10	R33	R34	PC4
USART_TX	PA2	R35	PF5
USART_RX	PA3	R37	PF4

# 1. SPI

- EEPROM

- PA5, PA6, PA7을 SPI1으로 설정하고, PB12를 GPIO OUTPUT으로 설정한다.
- 그 다음, SPI1을 "Full Duplex Master"로 설정한다.



# 1. SPI

- EEPROM

- 해당 EEPROM의 데이터 시트를 보면, 아래와 같은 SPI Frame을 가진다.
- 이때 **CPOL**, **CPHA**값은?

**CPOL = Low**  
**CPHA = 1 Edge**

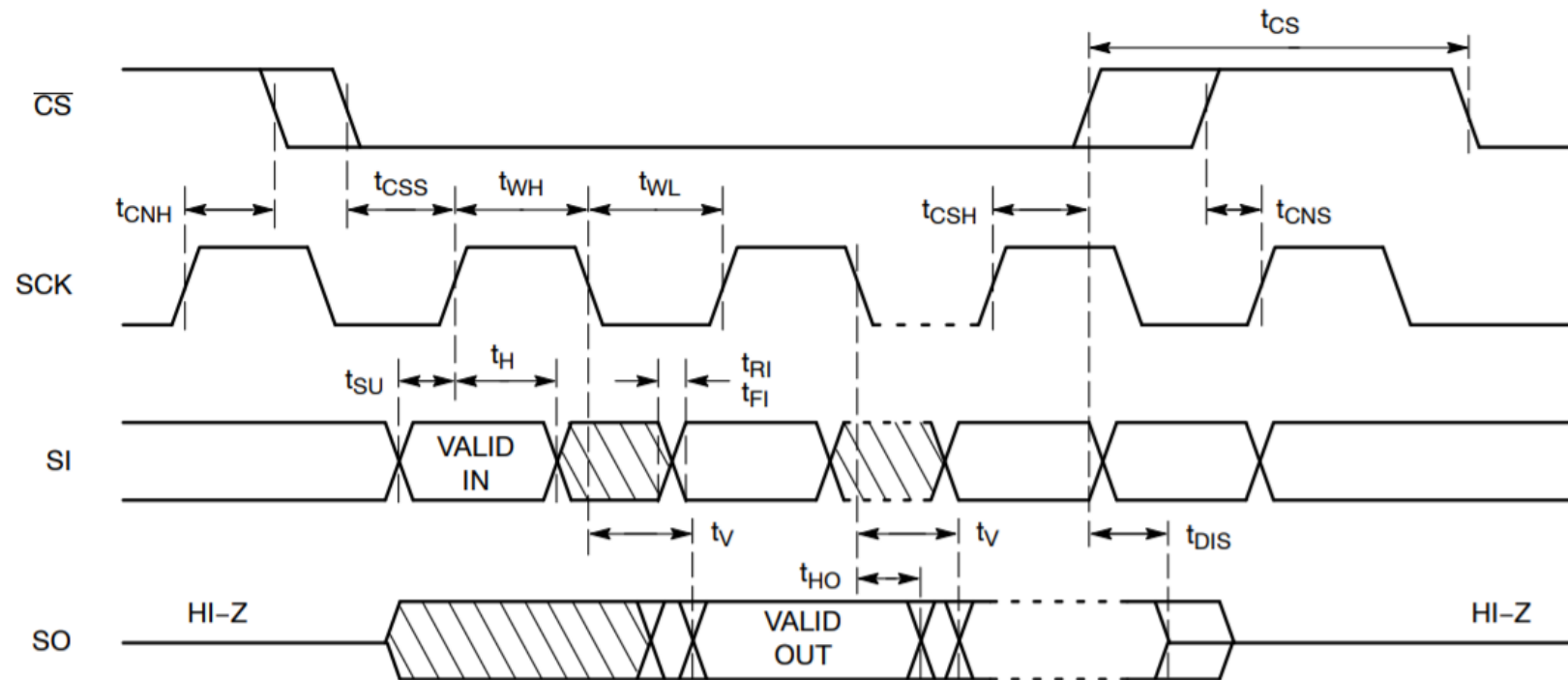
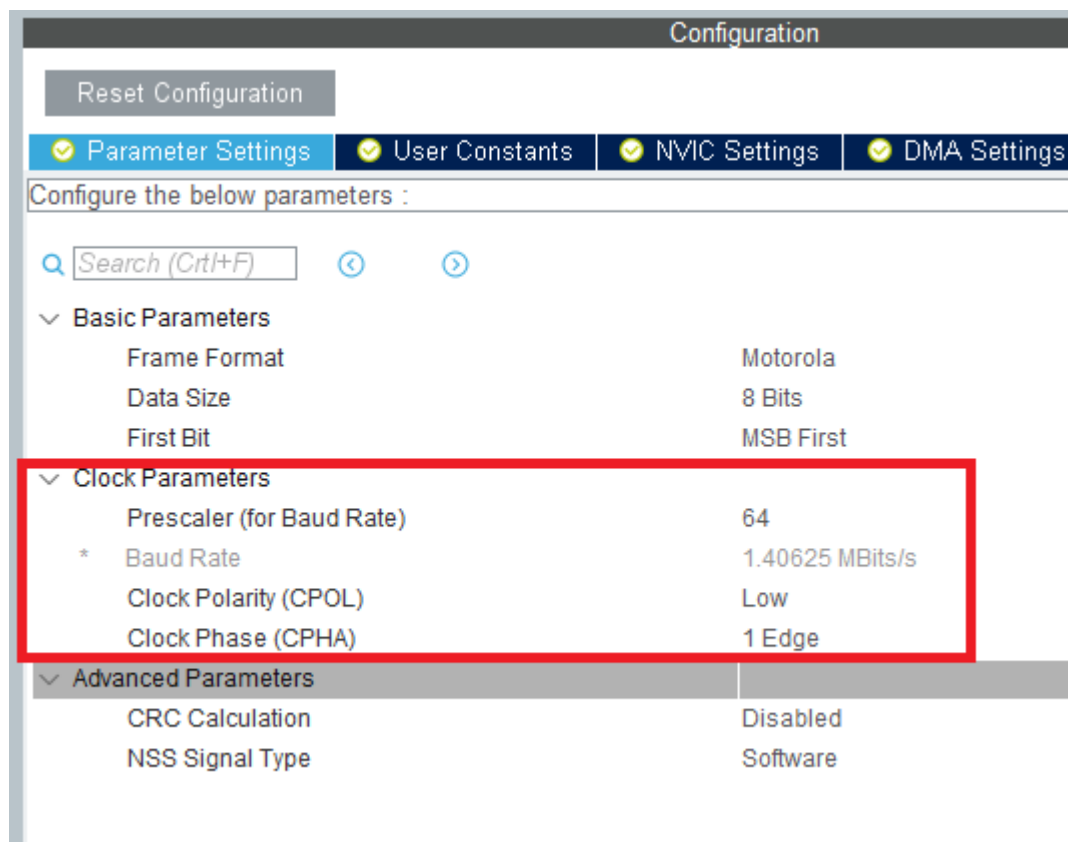


Figure 2. Synchronous Data Timing

# 1. SPI

- EEPROM

- 이에 따라 SPI 설정을 다음과 같이 해 준다.



Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings

Configure the below parameters :

Search (Ctrl+F) ⏪ ⏩

Basic Parameters

Frame Format	Motorola
Data Size	8 Bits
First Bit	MSB First

Clock Parameters

Prescaler (for Baud Rate)	64
* Baud Rate	1.40625 MBits/s
Clock Polarity (CPOL)	Low
Clock Phase (CPHA)	1 Edge

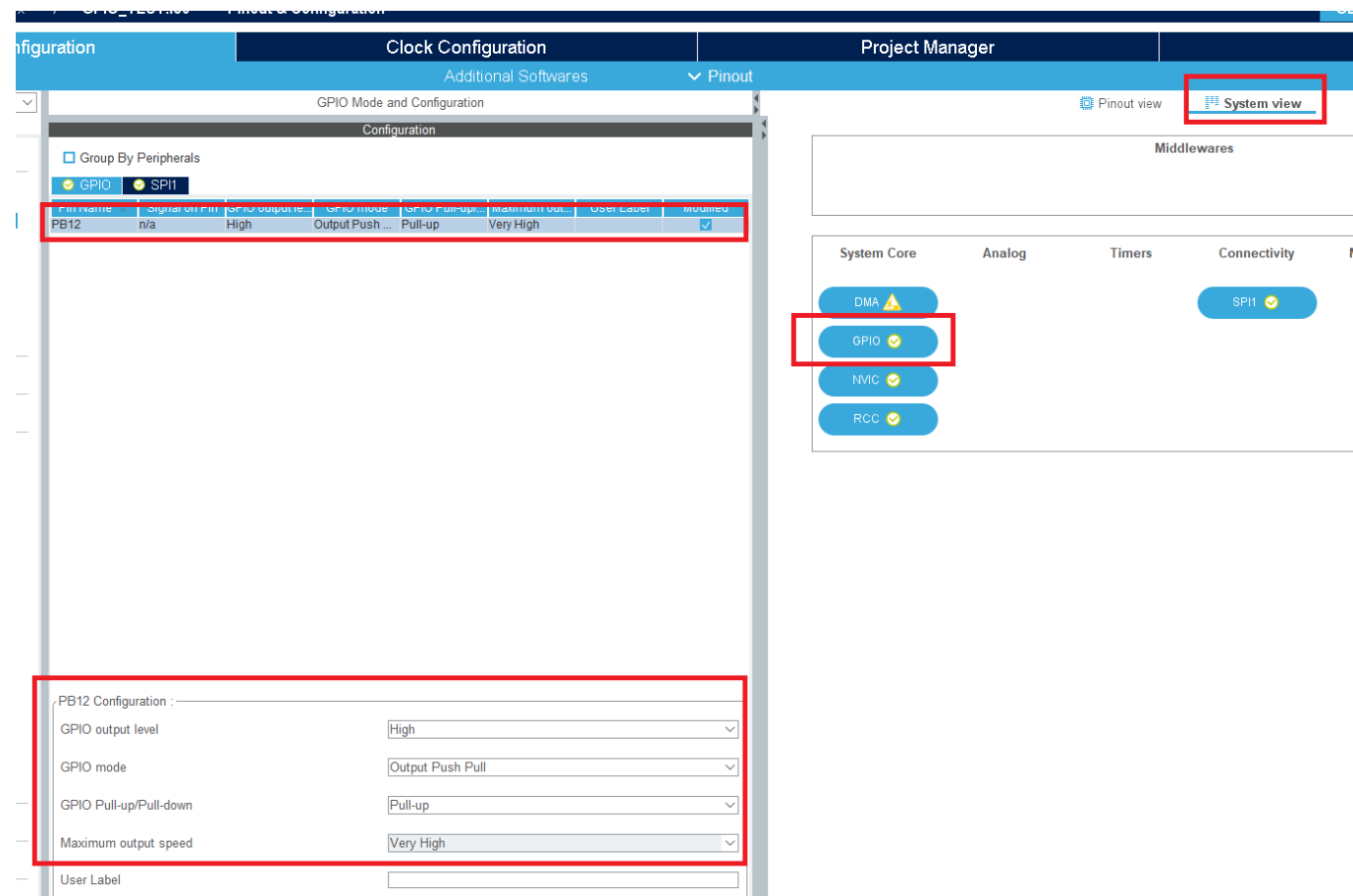
Advanced Parameters

CRC Calculation	Disabled
NSS Signal Type	Software

# 1. SPI

- EEPROM

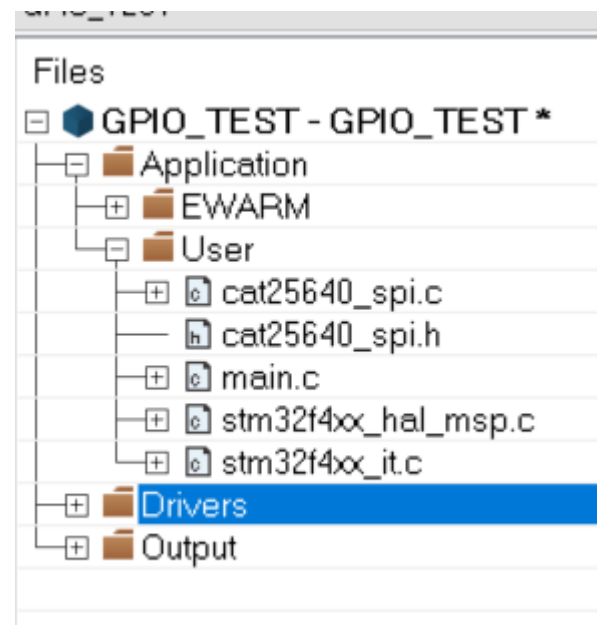
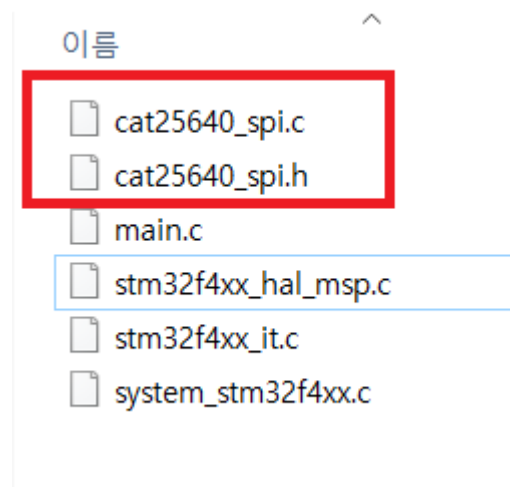
- 그 다음, NSS 핀인 PB12를 다음과 같이 설정해준다.
- PULL-UP을 해 주고, Output Speed를 Very High로 해준다.
- 코드를 생성한다.



# 1. SPI

- EEPROM

- 미리 주어진 "cat25640\_spi.h" 및 "cat25640\_spi.c"를 Src 폴더에 복사하고, IAR에서 파일을 추가한다.





# 1. SPI

- EEPROM

- Main.c 상단에 해당 헤더 파일을 인클루드 한다.

```
/* Private includes -----  
/* USER CODE BEGIN Includes */  
#include "cat25640_spi.h"  
/* USER CODE END Includes */
```

- EEPROM에 보내고 받을 변수를 선언해준다.

```
/* Private user code -----  
/* USER CODE BEGIN 0 */  
uint8_t txData[10], rxData[10];  
/* USER CODE END 0 */
```

# 1. SPI

- EEPROM

- 다음과 같이 txData에 데이터를 써서 EEPROM에 입력한 후, 1초 뒤에 EEPROM으로부터 값을 읽는 코드를 작성하자.
- 코드 업로드 후, Live watch를 통해 값을 관찰하자.

```
/* USER CODE BEGIN 2 */
EEPROM_Init(&hspi1, GPIOB, GPIO_PIN_12);

uint8_t i;
for(i = 0 ; i < 10; i++){
    txData[i] = i;
}
EEPROM_writeBytes(0x00, txData, 10);

HAL_Delay(1000);

EEPROM_readBytes(0x00, rxData, 10);

/* USER CODE END 2 */
```

# 1. SPI

## • EEPROM

- 다음 Timing Diagram을 통해 해당 코드를 이해해 보자.

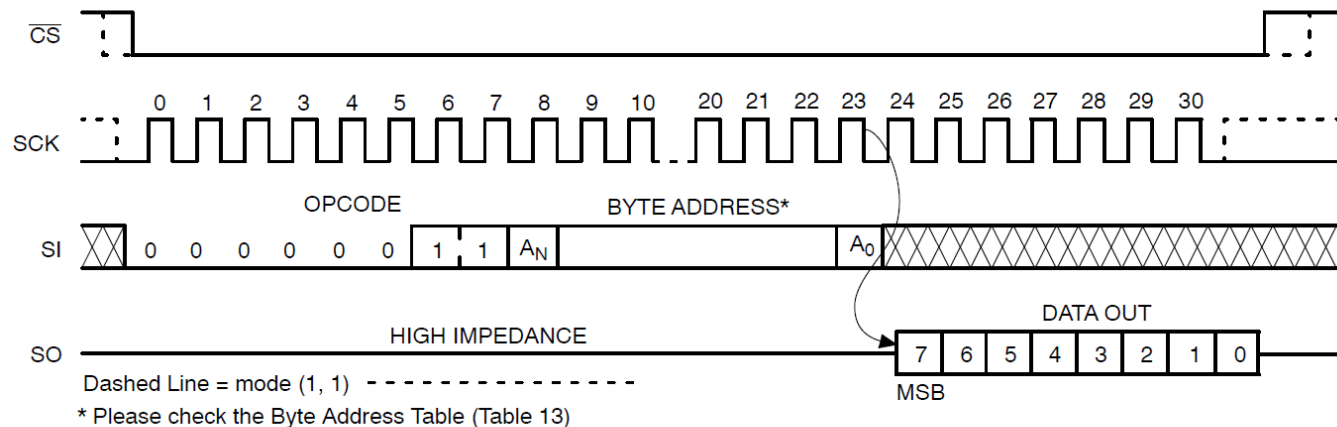


Figure 9. READ Timing

```
// EEPROM Read Operation (page 11)
void EEPROM_readBytes(uint16_t address, uint8_t* data, uint8_t len){

    uint8_t header[3];

    // To read from memory, the host sends a READ instruction followed by a 16bit address
    // Page 11, Figure 9
    header[0] = EEPROM_READ;
    header[1] = (uint8_t)(address >> 8);
    header[2] = (uint8_t)address;

    HAL_GPIO_WritePin(_eeprom.csPort, _eeprom.csPin, GPIO_PIN_RESET); // Start SPI Frame
    HAL_SPI_Transmit(_eeprom.hspi, header, 3, 100);

    // After receiving the last address bit, the CAT25640 will respond by shifting out data on the SO pin
    HAL_SPI_Receive(_eeprom.hspi, data, len, 100);
    HAL_GPIO_WritePin(_eeprom.csPort, _eeprom.csPin, GPIO_PIN_SET); // End SPI Frame
}
```