

# Serial Peripheral Interface Learning Activity

Spencer Hernandez - ECE 3430 - October 26, 2022

## I. INTRO

The purpose of this lab was to become familiar with interfacing to serial flash memory using an SPI serial bus.

## II. EXPLANATION

In order to interface with the flash memory, several methods had to be implemented to correctly configure the MSP for said interfacing. These methods included the following:

- SPISendByte
- SPIReceiveByte
- ReadFlashMemoryID
- ReadFlashMemoryStatusRegister
- WriteFlashMemoryStatusRegister
- ReadFlashMemory
- ByteProgramFlashMemory
- AAIPProgramFlashMemory
- ChipEraseFlashMemory
- SectorBlockEraseFlashMemory
- SetBlockProtection
- FlashMemoryBusy

The main challenge that came from implementing these methods was due to my unfamiliarity with reading datasheets on the level that I needed to for this activity. Once I was able to interpret the timing diagrams, I became far more efficient.

Once all of these methods were implemented, a rigorous test program was ran to verify their functionality. Figure 1 below shows the outputs of the registers that show each test was passed for each method.

Expression	Type	Value	Address
TestHasNotFailed	unsigned char[6]	[1 '\x01', 1 '\x01', 1 '\x01', 1 '\x01', 1 '\x01', ...]	0x2000FFEB
00 [0]	unsigned char	1 '\x01'	0x2000FFEB
00 [1]	unsigned char	1 '\x01'	0x2000FFEB
00 [2]	unsigned char	1 '\x01'	0x2000FFEA
00 [3]	unsigned char	1 '\x01'	0x2000FFEB
00 [4]	unsigned char	1 '\x01'	0x2000FFEC
00 [5]	unsigned char	1 '\x01'	0x2000FFED
00 TestNumber	unsigned char	6 '\x06'	0x2000FFF1
00 NumberOfTestsPassed	unsigned char	6 '\x06'	0x2000FFF2

Fig. 1. SPI Methods Verification