

## Overview

This document describes the demo project for PmodSF. This project shows basic functionalities of a PmodSF using the PmodSF library.

## Reference

- PmodSF Programmer's Reference Manual
- Digilent PmodSF Reference Manual and Schematic
- Cerebot 32MX4 reference manual and Schematic
- ST Micro M25P16/ M25P128 Data Sheet

## Required Hardware/Software

- PmodSF (16Mbits or 128Mbits)
- Cerebot 32MX4
- MPLAB

## Set-Up

1. Connect J1 on PmodSF to the upper row of connector JB. Make sure the side with the signal name (SEL, SDI, SDO...) is facing upward. Refer to the schematic to ensure that the right pins are connected.
2. Connect Cerebot 32MX4 to the computer and run the program through MPLAB.

For more information on how to run the program through MPLAB, please refer to "...tutorial"

## Description

This project goes through the following step:

1. Initialize SPI2.

PmodSF is accessed through SPI communication protocol. For the convenience of this project SPI2 is selected. SPI2 is located on the upper row of connector JB on Cerebot32MX4. SS is set to the port G pin 9, which is the upper rightmost pin of the connector. Clock rate is set to 40MHz. For the maximum clock rate for the chip, please refer to the PmodSF Programmer's Reference Manual. For more information of the SPI, please refer to "...Tutorial".

2. Get the model of the chip and the highest address by calling `GetDeviceInfo()`.

3. Erase the entire memory with `BulkErase()` and verify with `NormalRead()`.

Note: This function may take a while to execute.

4. Write a page to the memory address 0x110000 with `WritePage()` and verify with `NormalRead()`.

5. Erase a sector that contains the address 0x110000 with `SectorErase()` and verify with `FastRead()`.

6. Protect the Sector 31 with `WriteProtect()`.

7. Write a byte to an unprotected address 0x000000 with `WritePage()` and verify that data has been send back with `NormalRead()`.

8. Write a byte to a protected address 0x1F0000 with `WritePage()` and verify data has been sent back with `NormalRead()`.

9. Execute `Sleep()` and `Wake()`. These functions are only available on the chip M25P16.

When `Sleep()` is called, the device ignores all commands other than `Wake()`.

10. Verify `Sleep()` and `Wake()` with `WritePage()` and `NormalRead()`.

For details of each individual function, please refer to the PmodSF Programmer's Reference Manual.