```c
#include "flash_spi.h"


void flash_initialize(void)
{
  RCC->APB2ENR |= RCC_APB2ENR_AFIOEN | RCC_APB2ENR_IOPBEN | RCC_APB2ENR_IOPCEN ;
  RCC->APB1ENR |= RCC_APB1ENR_SPI2EN;

  GPIOB->CRH |= GPIO_CRH_CNF15_1 | GPIO_CRH_MODE15;
  GPIOB->CRH &= ~GPIO_CRH_CNF15_0;

  GPIOB->CRH |= GPIO_CRH_CNF13_1 | GPIO_CRH_MODE13;
  GPIOB->CRH &= ~GPIO_CRH_CNF13_0;

  GPIOB->CRH |= GPIO_CRH_CNF14_1 | GPIO_CRH_MODE14;
  GPIOB->CRH &= ~GPIO_CRH_CNF14_0;

  GPIOB->CRH &= ~GPIO_CRH_CNF10;
  GPIOB->CRH |= GPIO_CRH_MODE10;


  SPI2->CR1 = 0x00;
  SPI2->CR1 |= SPI_CR1_SSM | SPI_CR1_SSI;
  SPI2->CR2 &= ~SPI_CR2_SSOE;
  SPI2->CR1 |= SPI_CR1_BR_2 | SPI_CR1_BR_1;
  SPI2->CR1 |= SPI_CR1_SPE | SPI_CR1_MSTR;

  flash_select(false);
}

void flash_select(bool select)
{
  if (select)
  {
    GPIOB->BRR |= GPIO_BRR_BR10;
  }
  else
  {
    GPIOB->BSRR |= GPIO_BSRR_BS10;
  }
}
void flash_use(void)
{
  flash_initialize();
  delay_ms(1);
  flash_select(true);
  delay_ms(1);

  send_byte(0x03);
  send_byte(0x00);
  send_byte(0x00);
  send_byte(0x00);

  uint8_t z[16];
  for (int i =0; i < 16; i++)
  {
    z[i] = receive_byte();
  }
  CMD_2_LCD(LCD_LN1);
  for (int i =0; i < 8; i++)
  {
    send_2_LCD1(z[i]);
  }
  CMD_2_LCD(LCD_LN2);
  for (int i =8; i < 16; i++)
  {

    send_2_LCD1(z[i]);
  }
  flash_select(false);
  delay_ms(1000);
  CMD_2_LCD(LCD_CLR);
}
```

```
  74  void page_program(uint32_t *array, int length)
  75  {
  76    flash_initialize();
  77    delay_ms(1);
  78    flash_select(true);
  79    delay_ms(1);
  80
  81    send_byte(0x06);
  82    flash_select(false);
  83    delay_ms(1);
  84    flash_select(true);
  85    delay_ms(1);
  86    send_byte(0x02);
  87    send_byte(0x00);
  88    send_byte(0x00);
  89    send_byte(0x00);
  90    int incr = 24;
  91    uint32_t temp;
  92    uint8_t temp_byte;
  93    for(int i = 0; i < length; i++)
  94    {
  95      for (int j = 0; j < 2; j++)
  96      {
  97
  98        for (int k = 0; k < 4; k++)
  99       {
 100        temp = *(&(array[0]) + (2*i + j));
 101        temp_byte = (temp >> (incr));
 102        incr -= 8;
 103        send_byte(temp_byte);
 104       }
 105       incr = 24;
 106      }
 107    }
 108
 109    flash_select(false);
 110    delay_ms(1);
 111  }
 112  void chip_erase(void)
 113  {
 114    flash_initialize();
 115    delay_ms(1);
 116    flash_select(true);
 117    delay_ms(1);
 118
 119    send_byte(0x06);
 120    flash_select(false);
 121    delay_ms(1);
 122    flash_select(true);
 123    delay_ms(1);
 124    send_byte(0x20);
 125    send_byte(0x00);
 126    send_byte(0x00);
 127    send_byte(0x00);
 128    delay_ms(100);
 129    flash_select(false);
 130    delay_ms(1);
 131
 132  }
 133  void page_read(uint32_t *array)
 134  {
 135    flash_initialize();
 136    delay_ms(1);
 137    flash_select(true);
 138    delay_ms(1);
 139
 140    send_byte(0x03);
 141    send_byte(0x00);
 142    send_byte(0x00);
 143    send_byte(0x00);
 144    uint32_t temp1;
 145    for (int i =0; i < 64; i++)
 146    {
```

```
147       temp1 += receive_byte();
148       temp1 = temp1 << 8;
149       temp1 += receive_byte();
150       temp1 = temp1 << 8;
151       temp1 += receive_byte();
152       temp1 = temp1 << 8;
153       temp1 += receive_byte();
154       array[i] = temp1;
155       temp1 = 0;
156     }
157
158     flash_select(false);
159     delay_ms(10);
160
161   }
```