



**Design Notes:**

The sample-code is used to describe how to design the s/w for MXIC Serial flash. The following sample codes are suitable for MX25L3205D, MX25L1605A, MX25L1605D, MX25L1635D, MX25L3235D, MX25L8005, MX25U8035, MX25L4005A, MX25L2005, MX25L1005, MX25L1005A, MX25L512.

We would like to suggest customers to check the ID codes firstly before porting your algorithms. It can be easy to double confirm that flash device could be accepted command sets correctly.

Send data (unsigned char indata) // Send 1 byte data to Serial flash

```

{
  unsigned char k;
  for ( k = 0; k++; k< 8)
  {
    if( (indata & 0x80) == 0x80)
      SI = 1;
    else
      SI = 0;
    SCLK = 1;
    SCLK = 0;
    indata = (indata << 1);
  }
}

```

Unsigned char Get data () //Receive a byte from Serial flash

```

{
  unsigned char k, temdata =0, i;
  for ( k = 0; k++; k< 8)
  {
    if ( SO == 1 )
    {
      i = (0x80 >> k);
      temdata = (temdata || i);
    }
    SCLK = 1;
    SCLK = 0;
  }
  return (temdata);
}

```













**The following sample codes (Sector\_Erase\_Op) are suitable for Serial flash, MX25L3205D, MX25L1605D, MX25L3235D, MX25L1635D, MX25L1605A, MX25L8005, MX25L4005A, MX25L2005, MX25L1005, MX25L1005A, MX25L512.**

**Sector\_Erase\_Op() //Erase a sector (4K byte)**

```

{
    EraseOkFlag = TRUE;           /* clear the erase ok flag to be TRUE */

    CS# low ();
    Send data (0x06);           // Setting Write Enable Latch bit
    CS# high ();

    CS# low ();
    Send data (0x20);           //Write Sector Erase command = 0x20;
    Send data (AD1);           //Write Block address = AD1 = A23 – A16;
    Send data (AD2);           //Write Block address = AD2 = A15 – A8;
    Send data (AD3);           //Write Block address = AD3 = A7 – A0
    CS# high ();

    While(1)                    // set a loop to check the flash's Status Register till ready or time out
    {
        CS # low ();
        Send data (0x05);       //Write Read Status command = 0x05;
        StatusReg = Get data (); // read status from flash
        CS# high ();

        if( (StatusReg & 0x01) == 0x00) // If the Q0(write in progress bit) == '0' is ready or '1' busy
            break;
        else if( time-out )
        {
            EraseOKFlag = FALSE;
            break;
        }
    }

    return (EraseOkFlag);      /* report the final result to main program */
}                               /* end of Sector_erase_Op */

```

////////////////////////////////////



```

Page_Program_Op() //Program 256 byte per page
{
  int i;
  ProgramOkFlag = TRUE;           // clear the program ok flag to be TRUE

  CS# low ();
  Send data (0x06);               // Setting Write Enable Latch bit
  CS# high ();

  CS# low ();
  Send data (0x02);               //Write Page Program command
  Send data (AD1);               //Write Block address = AD1 = A23 – A16;
  Send data (AD2);               //Write Block address = AD2 = A15 – A8;
  Send data (AD3);               //Write Block address = AD3 = A7 – A0

  For (i=0; i<256 ; i++)          // set a loop to down load whole page data into flash's buffer
  {
    Send data (Wdi);             // WDi stands for write data 0 ~ 255 bytes
  }
  CS# high ();

  While(1)                        // set a loop to check the flash's Status Register till ready or time out
  {
    CS # low ();
    Send data (0x05);           //Write Read Status command = 0x05;
    StatusReg = Get data ();    // read status from flash
    CS# high ();

    if( (StatusReg & 0x01) == 0x00) // If the Q0(write in progress bit) == '0' is ready or '1' busy
      break;
    else if( time-out )
    {
      ProgramOKFlag = FALSE;
      break;
    }
  }
  return (ProgramOkFlag);       /* report the final result to main program */
}                                /* end of Program_Op */

```

////////////////////////////////////









