M25PXX-EPRK
USER MANUAL

M25Pxx Evaluation Programmer/Reader Kit

This is an installation guide and tutorial for the Evaluation Programmer/Reader Kit for the M25P10 device. The key features of this device are:

– The M25P10 is based on a Flash memory cell.
– The M25P10 is a *serial* access memory.
– The data transfer rate is high (up to 20 MHz), thereby compensating for the long access time of standard speed serial access memories.
– The M25P10 can be *page programmed* (with up to 128 Bytes in a page). This feature considerably shortens the programming time.
– The M25P10 is a 1Mbit device, divided into 4 sectors, each containing 256 pages of 128 bytes.
– The Erase instruction applies to a sector, not to a page.

INSTALLATION PROCEDURE

The M25Pxx Programmer/Reader Evaluation Kit consists of:

(A) Evaluation Programmer/Reader PCB (printed circuit board)

(B) AC/DC power supply block

(C) Parallel cable (25-pin connectors)

(D) 3½-inch diskette

(E) M25P10 DIP8 device

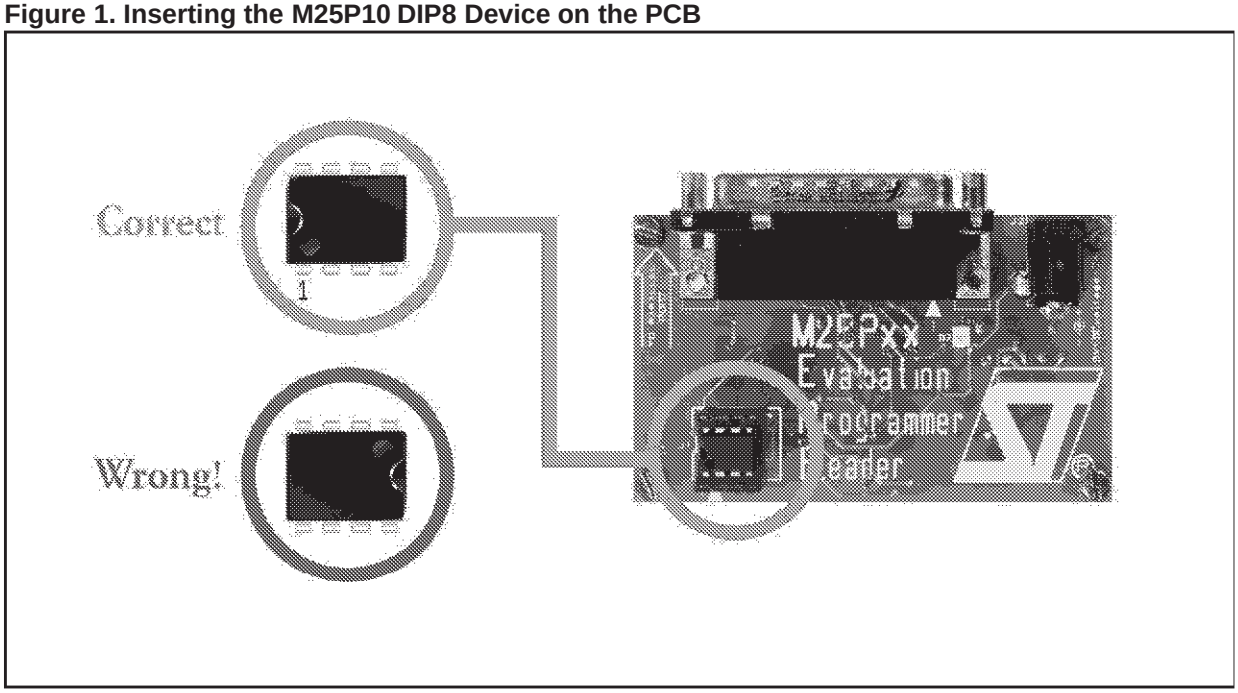
Step A1: Insert the M25P10 DIP8 device

– With the Evaluation Programmer/Reader PCB (A) in front of you, so that the words “Evaluation Programmer Reader” are upright.
– Insert the M25P10 DIP8 device (E) as shown in Figure 1, so that the notch of the DIP8 device is on your left (pin 1 is on the bottom left) of the M25Pxx Evaluation Programmer/Reader PCB

Step A2: Power up the Evaluation Programmer/Reader

– Ensure the power supply is set to 12V DC, and the central pin set to be positive and the outer ring negative. Alternatively, any external DC voltage in the range 9V to 24V may be used (50mA minimum required)
– Insert the DC supply jack (B) in the Evaluation Programmer/Reader (A).
– Insert the AC power block (B) plug in the mains outlet.
■ the LED should be on (if not, double check that AC is available and that the DC jack is inserted and configured with the central pin positive and the outer ring negative)

Figure 1. Inserting the M25P10 DIP8 Device on the PCB



Step A3: Connect the Evaluation Programmer/Reader to the host PC

– Insert one end of the 25-pin cable (C) into the Evaluation Programmer/Reader
– Insert the other end of the 25-pin cable (C) into the host PC (parallel port)

Step A4: Run the software from your PC[1]

Note: For accurate timings, it is preferable to shut down all applications and have only the EVA25P10.exe running.

– Insert the 3 1/2-inch floppy disk (D) into the diskette drive
– If the Host PC is running under Windows 3.1x, skip to Step A5
– If the Host PC is running under Windows 95 or 98, skip to Step A6

Step A5: PC running under Windows 3.1

– Click on the File Manager icon
– Select the 3 1/2-inch floppy disk reader
– Copy (drag) all files from diskette to a new directory on the host PC
– Move to this directory and double click the “EVA25P10.exe” file

Step A6: PC running under Windows 95 or 98

– Click the START icon
– Click Windows Explorer icon
– Select the 3 1/2-inch floppy disk reader
– Copy (drag) all files from diskette to a new directory on the host PC
– Move to this directory and double click the “EVA25P10.exe” file

Now you are ready to access the M25P10 serial paged Flash memory!

1.EVA25P10.exe is supported by Windows 3.1 and Windows 95/98 only (not fully supported by Windows NT)

TUTORIAL

Step B1: Call the M25P10 Evaluation Kit software driver

Please refer to the Installation procedure (see inside the Evaluation Kit box) and call *EVA25P10.exe*.

The screen is divided in several specific areas:

```
+-------------------------------------------------------------------------------+
| M25P10 |  Hardware Write Control W=1    131072x8 bits  LAST ADDRESS: 1FFFF  |
+-------------------------------------------------------------------------------+
 Read Bytes      |
 Wren+Page Prog  |
 Page PrOg       |   000000 18181818181818181818181818181818
                 |   000010 18181818181818181818181818181818
 RDSR            |   000020 18181818181818181818181818181818
 WRSR            |   000030 18181818181818181818181818181818
 WREN            |   000040 DADADADADADADADADADADADADADADADA
 WRDI            |   000050 DADADADADADADADADADADADADADADADA
                 |   000060 DADADADADADADADADADADADADADADADA
 SeCtor Erase    |   000070 DADADADADADADADADADADADADADADADA
 BulK Erase      |   000080 3D3D3D3D3D3D3D3D3D3D3D3D3D3D3D3D
                 |   000090 3D3D3D3D3D3D3D3D3D3D3D3D3D3D3D3D
 Deep PoWer Down |   0000A0 3D3D3D3D3D3D3D3D3D3D3D3D3D3D3D3D
 Read SiGnature  |   0000B0 3D3D3D3D3D3D3D3D3D3D3D3D3D3D3D3D
                 |   0000C0 01010101010101010101010101010101
 Toggle W pin    |   0000D0 01010101010101010101010101010101
                 |   0000E0 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
 TraNsfer data   |   0000F0 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
 Quit            |       F1(help) UP(previous) DOWN(next) <-+(execute)
+-------------------------------------------------------------------------------+
|     READ DATA BYTES: read bytes from the specified address (up to 256)      |
+-STMicroelectronics-----------------------------------------------------------+
```

Center area:

– Memory content is displayed (the first 256 bytes in this example)

– or some specific messages concerning the selected instruction or command.

Left column:

– M25Pxx instructions and Evaluation Kit commands

Bottom row (*This is important. Do not miss this!*):

– Explanations about the selected instruction or command.

Step B2: Reading the first 256 bytes

– Select the ***Read Bytes*** command (in the left column) with the help of the Up and Down arrows or by typing “R”,.

– Press ***Enter*** (twice)

■ **The result is**: you see the 256 first bytes displayed on the screen.

Step B3: Programming one byte

– Select the ***Wren + Page Prog*** command (in the left column) with the help of the Up and Down arrows or by typing “W”,

– Press ***Enter***

– Select the following, to program the first byte with ABh:

```
from NVM address: hex 000000
Number of data : dec 001
Data value : hex AB
```

– Press ***Enter***

■ **The result is**: starting at address 00000h, one byte is programmed with the value ABh and the time taken to program this byte is displayed on the screen

To check that the byte is properly programmed

– Select the ***Read*** instruction (as explained in Step B1)

■ **The result is**: the byte ABh appears in memory location 00000h

---

What is the WREN instruction before the Page Prog instruction?

The WREN instruction (WRite ENable) must precede each Program (or Erase) instruction; *otherwise* the Page Program instruction is *not* executed. This data protection mechanism is useful when the M25P10 is running in a noisy environment where spurious Page Prog instructions might improperly appear.

The following can be tried:

a) Select the Page Prog command (in the left column) by typing “O” (or with the help of the Up and Down arrows, followed by “Enter”).

– Press ***Enter***

■ The result is: *NO PROGRAMMING CYCLE* appears on the screen because a WREN instruction was not sent before the Page Prog instruction.

Step B4: Programming one page of 128 bytes

– Select the ***Wren + Page Prog*** command (in the left column) with the help of the Up and Down arrows or by typing “W”,

– Press ***Enter***

– Select the following, to program 128 bytes with 0Fh, from address 00080h:

```
from NVM address: hex 000080
Number of data : dec 128
Data value : hex 0F
```

– Press ***Enter***

■ **The result is**: starting at address 00080h, 128 bytes are programmed with a **single** instruction. The time to program these 128 bytes is displayed on the screen.

To check that the byte is properly programmed

– Select the ***Read*** instruction (as explained in Step B2)

■ **The result is**: the bottom 128 bytes now contain 0Fh

Step B5: Overwriting one byte

The byte at address 00000h was previously programmed with ABh (that is 1010.1011b).

Let's repeat Step B3 and program it with the complementary value (that is 0101.0100b = 54h):

– Select the ***Wren + Page Prog*** (on the left column) with the help of the Up and Down arrows or by typing "W",

– Press ***Enter***

– Select the following:

```
from NVM address: hex 00000
Number of data : dec 001
Data value : hex 54
```

– Press ***Enter***

■ **The result is**: the byte at location 00000h is overwritten and time to program this byte is displayed on the screen.

To check that the byte is properly over-written

Select the ***Read*** instruction (as explained in Step B2)

■ **The result is**: the first byte now contains 00h, as a Prog instruction can ONLY Program 0's (and cannot Program 1's). Byte programming is a bit-wise AND operation:

– the value stored in memory before the Prog instruction

– ANDed with the new value to be programmed).

The M25P10 is a serial paged Flash. The update of a any byte of Flash memory is limited to over-writing 0s on 1s. It is not possible to update a byte with any new value without first erasing the whole sector (or bulk) before starting the programming.

Step B6: Erasing one Sector

– Select the ***WREN*** command (in the left column) with the help of the Up and Down arrows or by typing "E",

– Press ***Enter***

– Select the ***Sector Erase*** command (on the left column) with the help of the Up and Down arrows or by typing "C",

– Press ***Enter***

■ **The result is*:*** the time to Erase one Sector (Sector0) is displayed on the screen. Please note that the Erase instruction can erase one sector but cannot erase anything smaller.

To check that Sector0 is properly erased

– Select the ***Read*** instruction (as explained in Step B2)

■ **The result is**: all bytes are erased (FFh)

What is the WREN instruction before the sector Erase instruction?

The WREN instruction (WRite ENable) must precede each Erase (or Program) instruction; *otherwise* the Erase instruction is *not* executed. This data protection mechanism is useful when the M25P10 is running in a noisy environment where spurious Sector Erase instructions might improperly appear.

The following can be tried:

a) Select the Erase command (in the left column) by typing “C” (or with the help of the Up and Down arrows, followed by “Enter”).

– Press ***Enter***

■ The result is: *NO PROGRAMMING CYCLE* appears on the screen because a WREN instruction was not sent before the Sector Erase instruction.

Step B7: Data protection

Data stored in the M25P10 can be protected against spurious programming when three conditions are met:

– the *Write Protect* input pin is driven low (logical 0)

– the Most Significant Bit (b7) of the status Register is set to 1

– BP1 and BP0 (b3 and b3) of the Status Register are set to either 01, 10 or 11.

Example: protection of the upper quarter (Sector3)

– Select the ***WREN*** command (in the left column) with the help of the Up and Down arrows or by typing “E”,

– Press ***Enter***

– Select the ***WRSR*** (Write status register) command (on the left column) with the help of the Up and Down arrows or by typing “S”,

– Enter “*1000.0100*”, that is b7 is set and (b3,b2)=(01)=upper quarter. The time required to program the Status Register is displayed on the screen.

– Select the ***Toggle W pin*** command (in the left column) with the help of the Up and Down arrows or by typing “T”,

– Looking at the upper row of the screen, you read:

“`Hardware Write Control=1`” the *Write Protect* input pin is not active.

– Press ***Enter***: this sets the W input low.

– Look at the upper row of the screen, you read:

“`Hardware Write Control=0`” the *Write Protect* input pin is active.

■ **The result is**: the contents of Sector3 are write Protected with (BP0,BP1) = (0,1)

To check that Sector3 is Write Protected

– Try to Program the value 55h into one page (of 128 bytes) in Sector3 (from address 18000h)

– Select the ***Wren + Page Prog*** command (in the left column) with the help of the Up and Down arrows or by typing “W”,

– Select the following (one byte to be programmed with 55h):

`from NVM address: hex 0`**`18`**`000`

`Number of data : dec` **`128`**

`Data value : hex` **`55`**

– Press ***Enter***

■ **The result is**: the message “NO PROGRAMMING CYCLE” shows that the Program instruction could *not* be executed, all bytes remain unmodified at FFh (as Sector3 had been previously erased to FFh).

APPENDIX1: IMPORTING BINARY FILES

The three most popular formats for binary files are:

– *".bin"* files containing only the data bytes, with no control character.
– *".hex"* files containing the data bytes with some control characters.
– *".s19"* files containing the data bytes with some control characters.

Binary files formats

The ".hex" files are generated according to the following rules:

– Byte 1: Header character (always a colon ":")
– Bytes 2,3: record length
– Bytes 4,5,6,7: Starting address
– Bytes 8,9: always 00h,00h
– Bytes 10+N: Data bytes
– Last bytes: 2 bytes of checksum.

The ".s19" files are generated according to the following rules:

– Byte 1: Header character (always "S")
– Bytes 2: type of record (1 = data record)
– Bytes 3,4: number of bytes of data in the record, including checksum and address data
– Bytes 5,6,7,8: address of the memory where this record must be stored
– Bytes 9+N: Data bytes
– Last bytes: 2 bytes of checksum.

Step C1: Importing 256Kbit from your binary file to the Host PC

The floppy diskette offers the following three drivers:

– ConvBIN.exe for use when your source binary file is a *".bin"* file
– ConvHEX.exe for use when your source binary file is a *".hex"* file
– ConvS19.exe for use when your source binary file is a *".s19"* file

1) Make sure that ALL the necessary files are stored in the same directory:
   – EVA25P10.exe
   – eetest.hlp
   – ConvBin.exe
   – ConvHex.exe
   – ConvS19.exe
   – Your binary source file to be read

2) Call the correct driver

The ConvXXX.exe is called under the DOS prompt (in the directory into which you have copied all the other files), with the following format:

ConvXXX.exe `sourcefile.XXX hostfile.ram [return]`

Where:

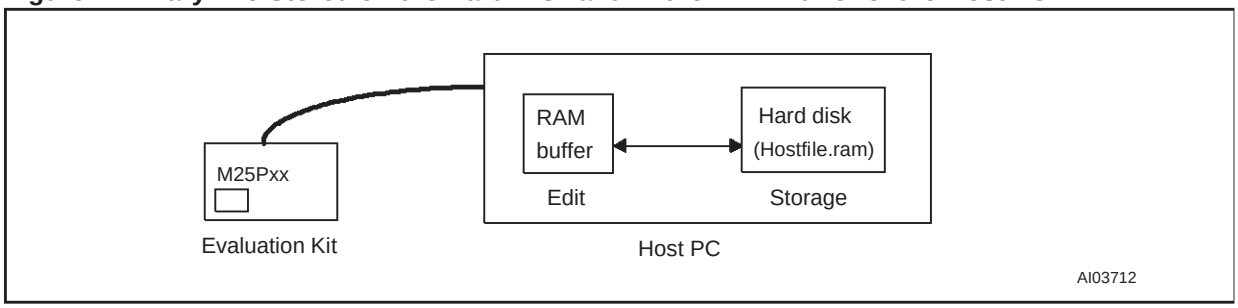
– `Sourcefile.XXX` is the name of your binary file
– `Hostfile` is the name that you choose for the file to be stored in the Host PC

To check that the file is properly copied

– Select the ***Transfer data*** command (in the left column) with the help of the Up and Down arrows or by typing “N”,
– Select the ***Load .ram file*** command (in the left column) with the help of the Up and Down arrows or by typing “L”,
– Type the name of the .ram file (for example, *“hostfile.ram”* in the above paragraph), then press ***Enter***,
– Select the ***Edit RAM*** command (on the left column) with the help of the Up and Down arrows or by typing “E”,

■ **The result is**: your binary file is now converted to “.ram” format and is stored both on the hard disk and in the RAM buffer of the Host PC, and the content of these files is now displayed on the screen.

Figure 2. Binary File Stored on the Hard Disk and in the RAM Buffer of the Host PC



Step C2: Copying *hostfile* into the M25P10

Your binary file (limited to 256Kbit) is now available in the Host PC, and it is possible:

– To Program the M25P10 with the content of the binary file (download *hostfile* into M25P10)
– To Edit (and modify) some byte values of *hostfile*.
– To Save your modifications to disk.

1) To Program the M25P10 with the content of the binary file

The content of *hostfile.ram* can be copied to the M25P10 in blocks of 256Kbit (one sector).

– Select the ***Copy RAM>Sector*** command (on the left column) with the help of the Up and Down arrows or by typing “C”,
– Press ***Enter***
– Select the Sector number (0,1,2 or 3),
– Press ***Enter***

■ **The result is*:*** the sector is programmed with the content of the binary file.

To check that the data are properly programmed

– Select the ***Sector verify*** command (in the left column) with the help of the Up and Down arrows or by typing “S”,
– Press ***Enter***
– Select the Sector number (0,1,2 or 3),
– Press ***Enter***
■ **The result is**: all the bytes in the sector are compared with the content of the binary file (256Kbit)

2) To Edit (and Modify) some byte values

The copy of the binary file in Host PC (in RAM) can be modified.

– Select the ***Edit RAM*** command (in the left column) with the help of the Up and Down arrows or by typing “E”,
– Press ***Enter***
– Select the ***Edit RAM*** command (in the left column) with the help of the Up and Down arrows or by typing “E”,
– Press ***Enter***
– Choose the address where data are to be modified

3) To Save the binary file in a format directly compatible with the Evaluation Kit

The new hostfile.ram (modified by the ***Edit RAM*** command) can be stored on the Host PC hard disk with the help of the ***Save .ram file*** command.

– Select the ***Save .ram file*** command. (in the left column) with the help of the Up and Down arrows or by typing “S”,
– Choose the name of the file to be stored on the hard disk and press ***Enter***

This file is now stored in the same directory as all the other Evaluation Kit files and can be recalled using the ***Load .ram file*** command.