AN404
APPLICATION NOTE

WRITE PROTECTION IN THE I²C and XI²C EEPROM FAMILIES

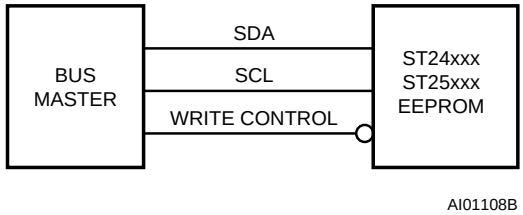
EEPROM is one of the most flexible of the non-volatile memory technologies, capable of being read, erased and written a byte at a time, or a block at a time. Although this flexibility is of obvious advantage, it is desirable that it be tempered with some degree of protection against inadvertent write or erase (caused by noise, software crash, hardware failure, or some other malfunction). After all, the main reason for choosing to use non-volatile memory is to be able to store persistent data: data that is important enough to be remembered through periods of power failure, and from one power-on session to the next.

The M24Cxx and ST24/25xxx families of serial EEPROM devices are fully compatible with the I²C protocol. In addition, though, each device offers extra data protection facilities for use in guarding against the possibility of inadvertent write operations.

GLOBAL PROTECTION AGAINST INADVERTENT ERASE/WRITE

The Write Control ($\overline{WC}$) input is present on the M24Cxx and ST24/25Wxx devices (not on ST24/25Cxx devices). It offers a mechanism for the global protection of the memory contents, for example as shown in Figure 1. When $\overline{WC}$ is driven high by the external circuitry (normally by the bus master), write and erase operations are ignored; when driven low, they are accepted, and executed.

Figure 1. EEPROM Interface Using the Write Control Line



A typical application normally holds the $\overline{WC}$ line high, to put the EEPROM in its protected (read-only) mode. Then, when a write or erase operation is to be performed, it performs the following sequence:

1. The $\overline{WC}$ line is taken low
2. The write or erase operation is performed.
3. The $\overline{WC}$ line is taken high again.

Any write or erase operations received whilst the EEPROM is in its protected mode are ignored. The particular behaviour of the Page Write operation, though, warrants some further attention:

When the EEPROM is in its unprotected mode ($\overline{WC}$ held low), and a Page Write operation is issued, the device's internal address counter is incremented after each received byte. When the last byte is received, this leaves the internal address counter at the right value for the first byte of the following Page Write operation. When the EEPROM is in its protected mode ($\overline{WC}$ held high), and a Page Write operation is issued, no data are written, but the internal address counter is still incremented after each byte, except for the last received byte – the EEPROM internal address counter remains at the last received byte address.

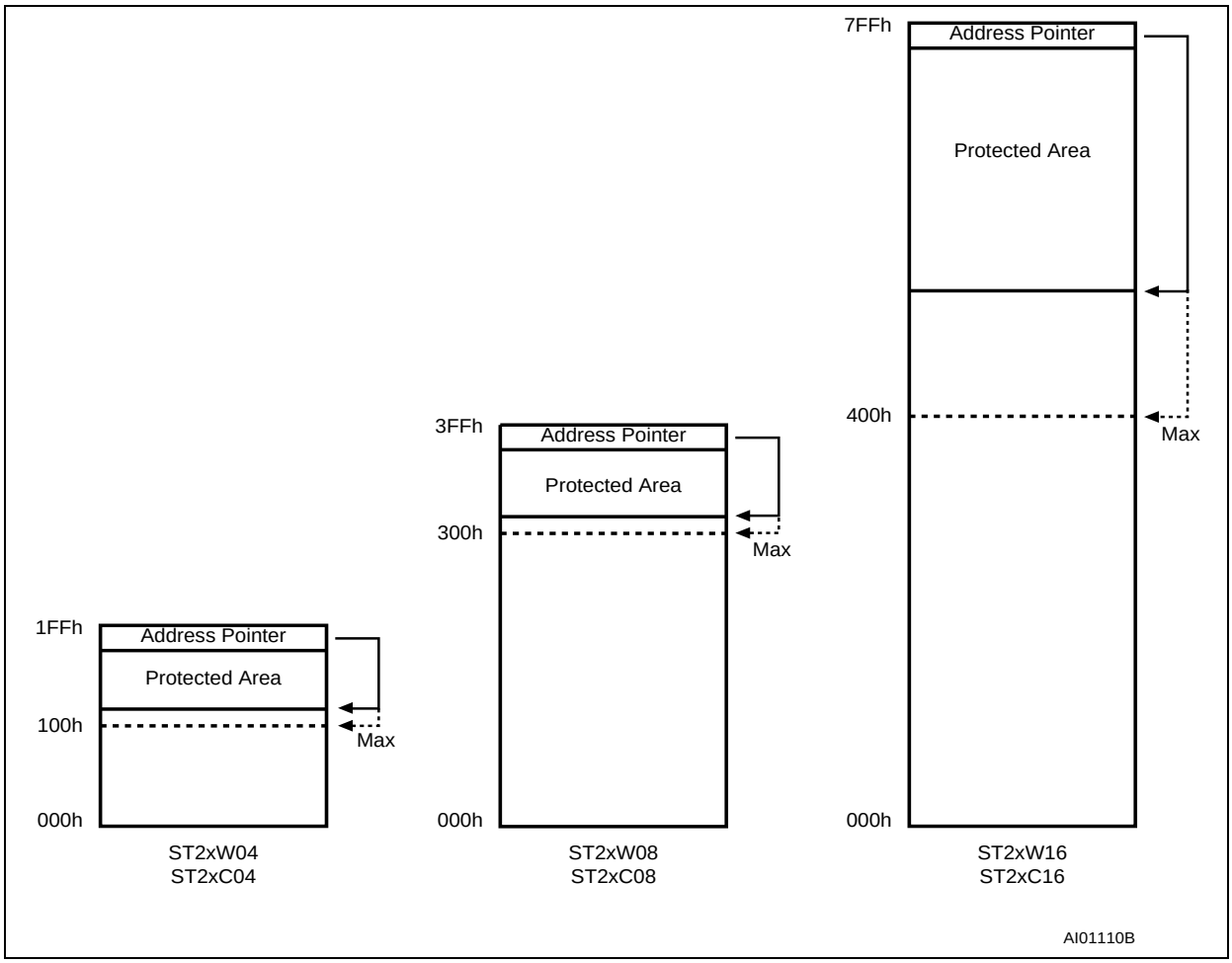
SELECTIVE PROTECTION AGAINST INADVERTENT WRITE

The members of the ST24/25x04, ST24/25x08 and ST24/25x16 families each offer a selective protection mechanism, in addition to the global protection offered by the $\overline{WC}$ pin. If the PRE pin is held high, and the bit b2 in the last byte of the EEPROM contains a '0', part of EEPROM becomes write protected, and behaves like ROM. The remaining area continues to function as unprotected EEPROM (though this can still be protected by taking the $\overline{WC}$ input high). The protected region extends from the last byte of the EEPROM down to the address pointed to by the most significant bits in the last byte (as shown in Figure 2).

The last byte in the memory has an address of 1FFh in the ST24/25x04, of 3FFh in the ST24/25x08, and of 7FFh in the ST24/25x16. The maximum size of the protected memory, as depicted in Figure 2, is:

- 256 bytes (the top half of the memory) for the ST24/25C04 and ST24/25W04
- 256 bytes (the top quarter of the memory) for the ST24/25C08 and ST24/25W08
- 1024 bytes (the top half of the memory) for the ST24/25C16 and ST24/25W16.

Figure 2. Size and Location of the Protected Areas Under the Control of the PRE Signal



Using the PRE Pin to Control the Writing of the Protected Area

The sequence for protecting an area of the memory is as follows:

1. The PRE pin is held low (by the bus master).
2. The new contents of the memory are written.
3. The address of the start of the protected region is written to the last byte. The contents of bit b2 are set to '0'. (See the next page for the precise format of this byte).
4. PRE pin is taken high (by the bus master).

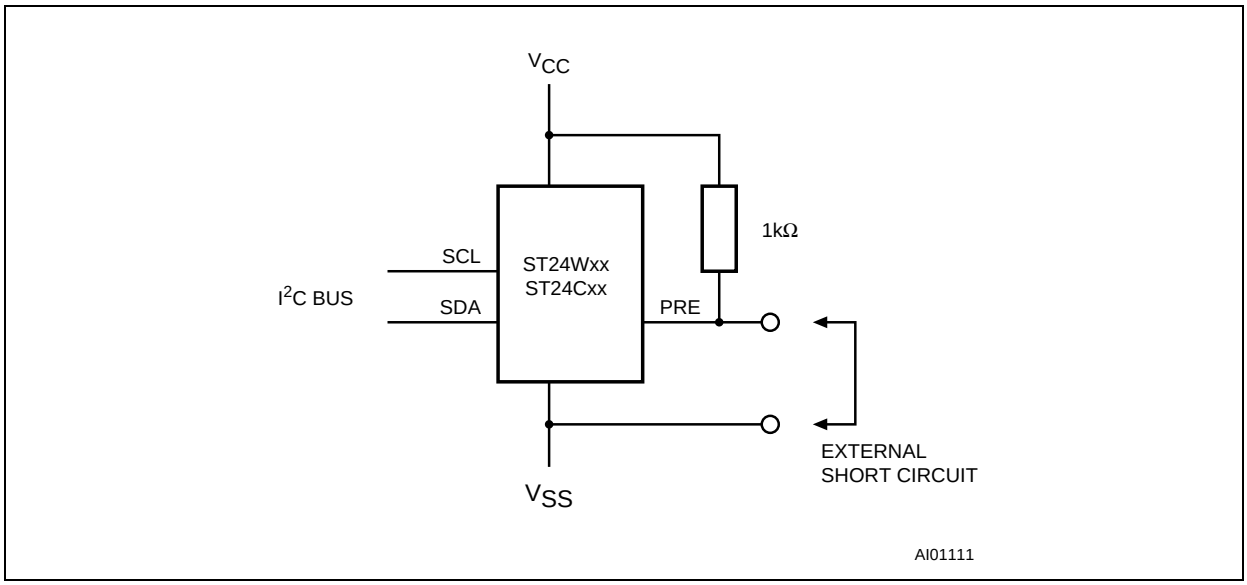
Data above the byte pointed by the address pointer are now write protected, and cannot be modified. This part of the EEPROM is now functionally equivalent to a block of ROM.

The dynamic use of the PRE signal, for controlled writing of the protected region by the bus master, is similar to that for the $\overline{WC}$ signal. That is, the sequence is as follows:

1. The PRE line is taken low
2. The write operation is performed.
3. The PRE line is taken high again.

The PRE pin may be driven dynamically by the bus master, as indicated in the above sequence. Alternatively, it can be wired permanently to V_{CC} (to disable further writes once the manufacturer has programmed the protected area) or wired permanently to V_{SS} (if the PRE control is not to be used at all).

Figure 3. Static Drive of the PRE Signal



Alternatively, the PRE pin can be pulled high, as shown in Figure 3. This arrangement allows the application manufacturer to program the protected area of the EEPROM as follows:

1. The components, including the EEPROM and its pull-up resistor, are assembled on the board.
2. The PRE pin is forced low (typically using a short circuit between the PRE pin and the V_{SS} pin).
3. The application data (such as fabrication date, serial number, customer or dealer area) are written in the protected area, along with the address pointer to the start of the application data, and the value '0' in bit b2 (see the next page for the format of this address).
4. The external short circuit of the PRE line is removed.

When the application board is delivered to the end user, the data in the application area cannot be modified. It is as if they were written in ROM. The remainder of the memory can be used, by the end user, as normal EEPROM.

Format of the Last Byte

The last byte of the EEPROM contains two fields of data:

- the write control bits (in the least significant portion)
- the address of the start of the protected area (in the most significant portion)

Bit b2 of the write control bits is used, in combination with the input on the PRE pin, to control the write access to the protected area. That is, the write protection function can be controlled by hardware, via the PRE input, or by software, via the setting of the b2 bit. The control function is as follows:

- The area of memory is write protected if PRE is held high AND bit b2 of the last byte is set to '0'.
- The area of memory is write enabled if PRE is held low OR bit b2 of the last byte is set to '1'.

The address of the start of the protected area only represents the most significant portion of the physical address. That is, the address of the start of the protected area can only be specified in steps of 8 for the ST24/25W04 and ST24/25W08, and in steps of 16 for the ST24/25W16.

Figure 4 shows the format for the last byte for members of the ST24/25x04 and ST24/25x08 families.

- The 5 most significant bits (b7 to b3) make up the address field
- The 3 least significant bits (b2 to b0) make up the write control field

The address pointer is treated as an 8-bit integer with the 5 most significant bits derived from the address field, and the 3 least significant bits of the address assumed to be '0' (independent of the setting of the b2, b1 and b0 control bits). The address pointer can therefore define a protected area with a maximum size of 256 bytes, in steps of 8 bytes.

Figure 4. Format of the Last Byte and the Address Pointer in 4K and 8K Devices

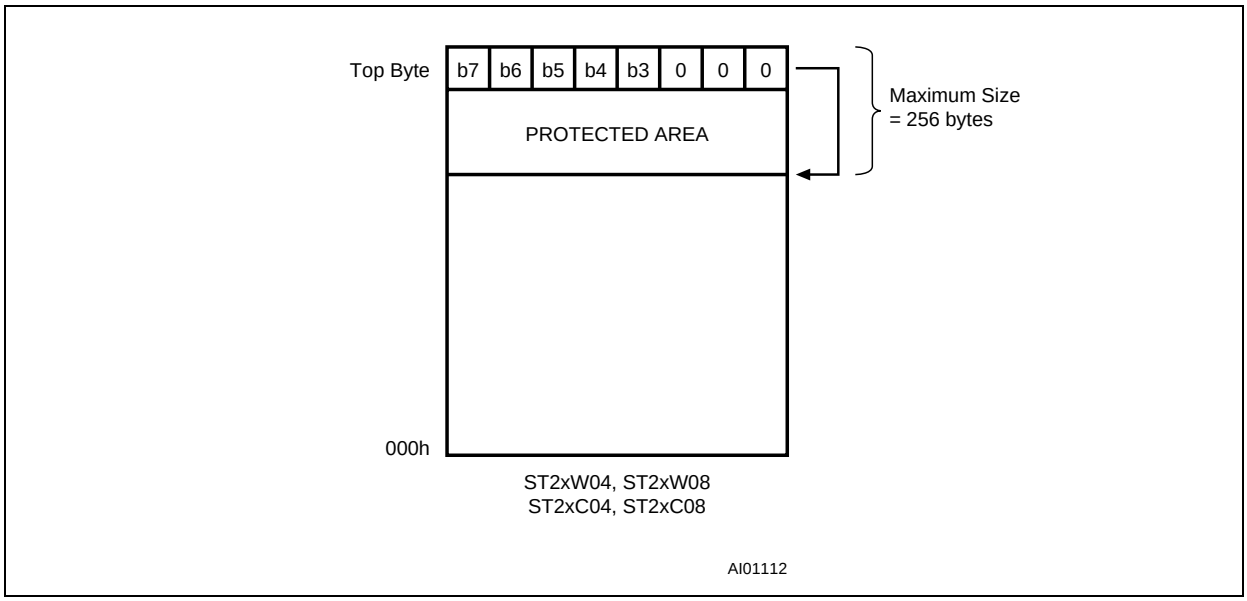
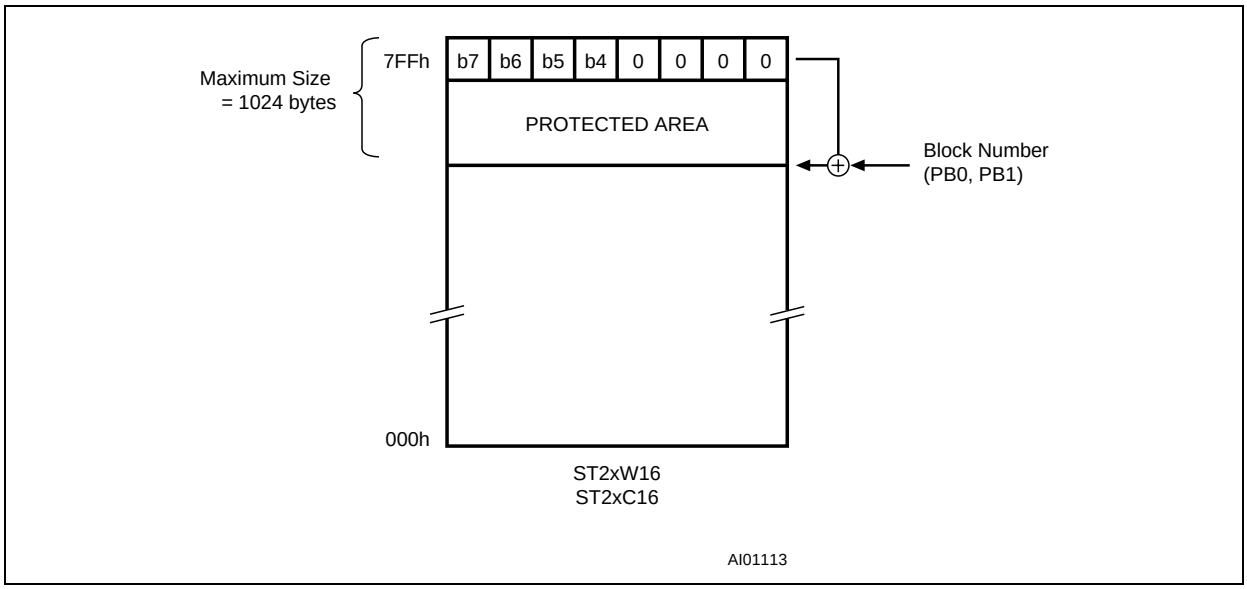


Figure 5 shows the format for the last byte for members of the ST24/25x16 family.

- The 4 least significant bits (b3 to b0) make up the write control field
- The 4 most significant bits (b7 to b4) make up the address field

The address pointer is treated as a 10-bit integer with the 2 most significant bits derived from the state of the PB1 and PB0 pins of the chip, the next 4 bits derived from the address field, and the 4 least significant bits of the address assumed to be '0'. The address pointer can therefore define a protected area with a maximum size of 1024 bytes, in blocks of 256 bytes (as specified externally by the state of the PB1 and PB0 pins), and in steps of 16 bytes (because the 4 least significant bits are taken to be '0').

Figure 5. Format of the Last Byte and the Address Pointer in 16 K Device



USE OF THE DEVICE SELECT BYTE AND CHIP ENABLE PINS

Many of ST's I^2C devices (notably the members of the ST24/25x01, ST24/25x02 and M24Cxx families) have data dependent chip enable pins (E0, E1 and E2). These inputs are designed principally to allow more than one device to be placed on the I^2C bus, as described in Application Note *AN1005*. A unique three-bit address, or identification code, is hard-wired to each chip's pins, and each chip only responds to commands on the I^2C bus that bear the corresponding device select code in the device select byte.

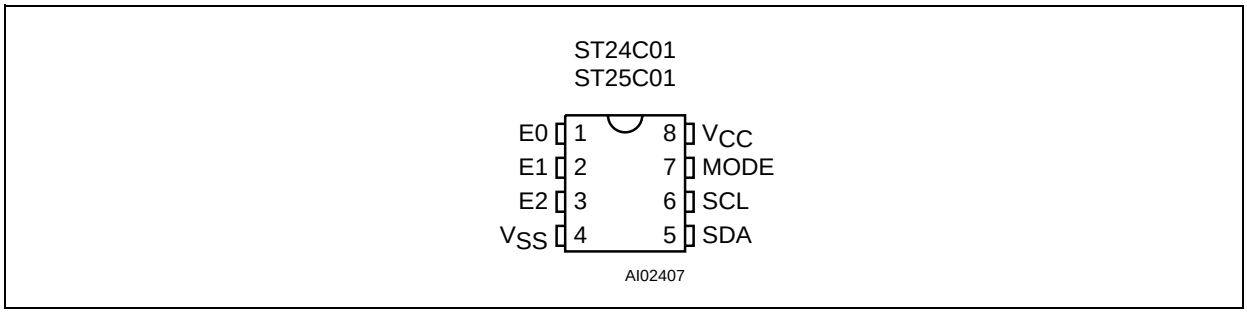
However, if the I^2C bus is not fully populated, it is possible to use this facility as an extra protection mechanism. If the application hardware detects an abnormal condition, such as excessive noise on the bus or power supply levels going out of specification, it can set the E0, E1 and E2 inputs to the identifier of a non-existent device. Provided that the microcontroller never addresses this non-existent device, the EEPROM will be hidden from any further accesses (read accesses will be prevented, too, as well as erase and write accesses).

PIN-OUT AND DEVICE SELECT BYTE: VARIATIONS WITHIN THE FAMILY

Three mechanisms for write protection have been described in this document. Only the members of the ST24/25x04 and ST24/25x08 families offer all three mechanisms. Usually, the designer is restricted to the one or two mechanisms offered by the device that is being used in the application. It is, of course, probably memory capacity, not write protection mechanism, that determines the choice of EEPROM for the application. So, the designer needs to consult the data sheet of the chosen device when deciding on an appropriate write protection strategy.

Table 1 summarises the differences in pin-out for each of the members of the ST24/25xxx family. For illustration, Figure 6 shows the pin-out of the ST24/25C01, serial EEPROM, 1 Kbit device.

Figure 6. Pin-out for the ST24/25C01 Serial EEPROM (1 Kbit)



The pin-out varies, for pins 1, 2, 3 and 7, between the individual members of the ST24/25Cxx and ST24/25Wxx families, as summarised in Table 1.

Table 1. Pin Compatibility in the ST24/25xxx Series

Pin Number	1	2	3	4	5	6	7	8
ST24C01/ST25C01	E0	E1	E2	V_{SS}	SDA	SCL	MODE	V_{CC}
ST24W01/ST25W01	E0	E1	E2	V_{SS}	SDA	SCL	$\overline{WC}$	V_{CC}
ST24C02/ST25C02	E0	E1	E2	V_{SS}	SDA	SCL	MODE	V_{CC}
ST24W02/ST25W02	E0	E1	E2	V_{SS}	SDA	SCL	$\overline{WC}$	V_{CC}
ST24C04/ST25C04	PRE	E1	E2	V_{SS}	SDA	SCL	MODE	V_{CC}
ST24W04/ST25W04	PRE	E1	E2	V_{SS}	SDA	SCL	$\overline{WC}$	V_{CC}
ST24C08/ST25C08	PRE	nc	E	V_{SS}	SDA	SCL	MODE	V_{CC}
ST24W08/ST25W08	PRE	nc	E	V_{SS}	SDA	SCL	$\overline{WC}$	V_{CC}
ST24C16/ST25C16	PRE	PB0	PB1	V_{SS}	SDA	SCL	MODE	V_{CC}
ST24W16/ST25W16	PRE	PB0	PB1	V_{SS}	SDA	SCL	$\overline{WC}$	V_{CC}
ST24E16/ST25E16	E0	E1	E2	V_{SS}	SDA	SCL	$\overline{WC}$	V_{CC}

The variation in the use of pins 1, 2 and 3 leads to variations in the composition of the first byte of each I^2C data transfer (the device select byte), as summarised in Table 2.

Table 2. Composition of the Device Select Byte

Bits in the Device Select Byte	b7	b6	b5	b4	b3	b2	b1	b0
ST24C01/ST25C01	1	0	1	0	E2	E1	E0	R/$\overline{W}$
ST24W01/ST25W01								
ST24C02/ST25C02	1	0	1	0	E2	E1	E0	R/$\overline{W}$
ST24W02/ST25W02								
ST24C04/ST25C04	1	0	1	0	E2	E1	A8	R/$\overline{W}$
ST24W04/ST25W04								
ST24C08/ST25C08	1	0	1	0	E	A9	A8	R/$\overline{W}$
ST24W08/ST25W08								
ST24C16/ST25C16	1	0	1	0	A10	A9	A8	R/$\overline{W}$
ST24W16/ST25W16								
ST24E16/ST25E16	1	0	1	0	E2	E1	E0	R/$\overline{W}$

If you have any questions or suggestions concerning the matters raised in this document, please send them to the following electronic mail addresses:

apps.eeprom@st.com (for application support)

ask.memory@st.com (for general enquiries)

Please remember to include your name, company, location, telephone number and fax number.