# M24M02E-U

Datasheet

## 2-Mbit serial I²C bus EEPROM with unique identifier, configurable device address, and software write protection registers

SO8N

(150 mil width)

| Product status |
| --- |
| M24M02E-U |

| Product label |
| --- |
| LONGEVITY 10 YEARS COMMITMENT |

## Features

### I²C interface

- Compatible with the following I²C bus modes:
  - 1 MHz (Fast-mode Plus)
  - 400 kHz (Fast-mode)
  - 100 kHz (Standard-mode)

### Memory

- 2-Mbit (256-Kbyte) of EEPROM
- Page size: 256-byte

### Identification page

- 256-byte locked in read-only mode at factory delivery

### UID

- 128-bit (16-byte) unique factory-programmed serial number

### Supply voltage

- Wide voltage range: 1.6 V to 5.5 V

### Temperature

- Operating temperature range: -40 °C to +85 °C

### Fast write cycle time

- Byte and page write within 4 ms (typically 3.3 ms)

### Performance

- Enhanced ESD/latch-up protection
- More than 4 million write cycles
- More than 200-year data retention
- Fast wake-up time (less than 5 µs)

### Ultralow-power current consumption

- 350 nA (typical) in standby mode
- 100 µA (typical) for read current
- 500 µA (typical) for write current

**Advanced features**

- Configurable device address register
- Device type identifier register (read-only)
- Software write protection register
- Hardware write protection of the whole memory array
- Random and sequential read modes

**Packages**

- SO8N (ECOPACK2-compliant)

## Application

Applications of EEPROM UID include:

- Improved traceability for accessory recognition
- Enhanced repairability
- Promoting sustainability in the consumer and industrial segments, such as:
  - Data centers
  - Logistics
  - Healthcare
  - Personal electronics

# 1 Description

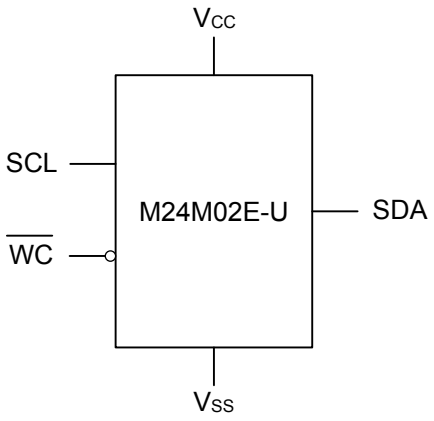
The M24M02E-U is a 2-Mbit I²C-compatible EEPROM (electrically erasable programmable memory) organized as 256 K x 8 bits. It can operate with a supply voltage from 1.6 V to 5.5 V with a clock frequency up to 1 MHz, over an ambient temperature range from -40 °C to +85 °C.

The device offers three 8-bit registers, namely: the device type identifier (DTI) register, the configurable device address (CDA), and the software write protection (SWP) register.

The device also offers a 256-byte identification page. This additional page stores a 128-bit (16-byte) unique factory-programmed serial number, locked in read-only mode at factory delivery.

The uniqueness of the serial number is ensured across the whole portfolio of serial EEPROMs manufactured by STMicroelectronics, offering a unique identifier.

**Figure 1. Logic diagram**

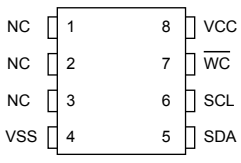



DT76056V1

**Table 1. Signal names**

| Signal name | Function | Direction |
|---|---|---|
| SDA | Serial data | I/O |
| SCL | Serial clock | Input |
| $\overline{WC}$ | Write control | Input |
| $V_{CC}$ | Supply voltage | - |
| $V_{SS}$ | Ground | - |

**Figure 2. 8-pin package connections, top view**




DT54532V1

1. NC: Not connected
2. See Section 10: Package information for package dimensions, and how to identify pin 1.

# 2 Signal description

## 2.1 Serial clock (SCL)

SCL is an input. The signal applied on it is used to strobe the data available on SDA(in) and to output the data on SDA(out).

## 2.2 Serial data (SDA)

SDA is an input/output used to transfer data in or out of the device. SDA(out) is an open-drain output that can be wired-AND with other open-drain or open-collector signals on the bus. A pull-up resistor must be connected from serial data (SDA) to $V_{CC}$ (Figure 20 and Figure 21 indicate how to calculate the value of the pull-up resistor).

## 2.3 Write control ($\overline{WC}$)

This input signal is useful for protecting the whole content of the memory from inadvertent write operations. Write operations are:

- Disabled to the whole memory array when write control is driven high.
- Enabled when write control is either driven low or left floating.

When the write control signal is driven high, the device select and address bytes are acknowledged, but data bytes are not acknowledged.

## 2.4 $V_{SS}$ (ground)

$V_{SS}$ is the reference for the $V_{CC}$ supply voltage.

## 2.5 Supply voltage ($V_{CC}$)

### 2.5.1 Operating supply voltage ($V_{CC}$)

Before selecting the memory and issuing instructions to it, a valid and stable $V_{CC}$ voltage within the specified [$V_{CC}$(min), $V_{CC}$(max)] range must be applied (see operating conditions in Section 9: DC and AC parameters).

To secure a stable DC supply voltage, it is recommended to decouple the $V_{CC}$ line with a suitable capacitor (usually from 10 nF to 100 nF) close to the $V_{CC}$/$V_{SS}$ package pins.

This voltage must remain stable and valid until the end of the transmission of the instruction and, for a write instruction, until the completion of the internal write cycle ($t_W$).

### 2.5.2 Power-up conditions

The $V_{CC}$ voltage increases from 0 V up to the minimum $V_{CC}$ operating voltage (see operating conditions in Section 9: DC and AC parameters). Once $V_{CC}$ is greater than or equal to the minimum $V_{CC}$ level, the controller must wait for at least $t_{WU}$ before sending the first command to the device. See Table 19. AC characteristics in Fast-mode and Table 20. AC characteristics in Fast-mode Plus for the value of the wake-up time parameter.

### 2.5.3 Device reset

To prevent inadvertent write operations during power-up, a power-on reset (POR) circuit is included.

At power-up, the device does not respond to any instruction until $V_{CC}$ has reached the internal reset threshold voltage. This threshold is lower than the minimum $V_{CC}$ operating voltage (see operating conditions in Section 9: DC and AC parameters). When $V_{CC}$ passes over the POR threshold, the device is reset and enters the standby power mode. The device must not be accessed until $V_{CC}$ reaches a valid and stable DC voltage within the specified [$V_{CC}$(min), $V_{CC}$(max)] range (see operating conditions in Section 9: DC and AC parameters).

Similarly, during power-down, when the $V_{CC}$ decreases, the device must not be accessed once $V_{CC}$ drops below $V_{CC}$(min). When $V_{CC}$ drops below the power-on-reset threshold voltage, the device stops responding to any instruction sent to it.

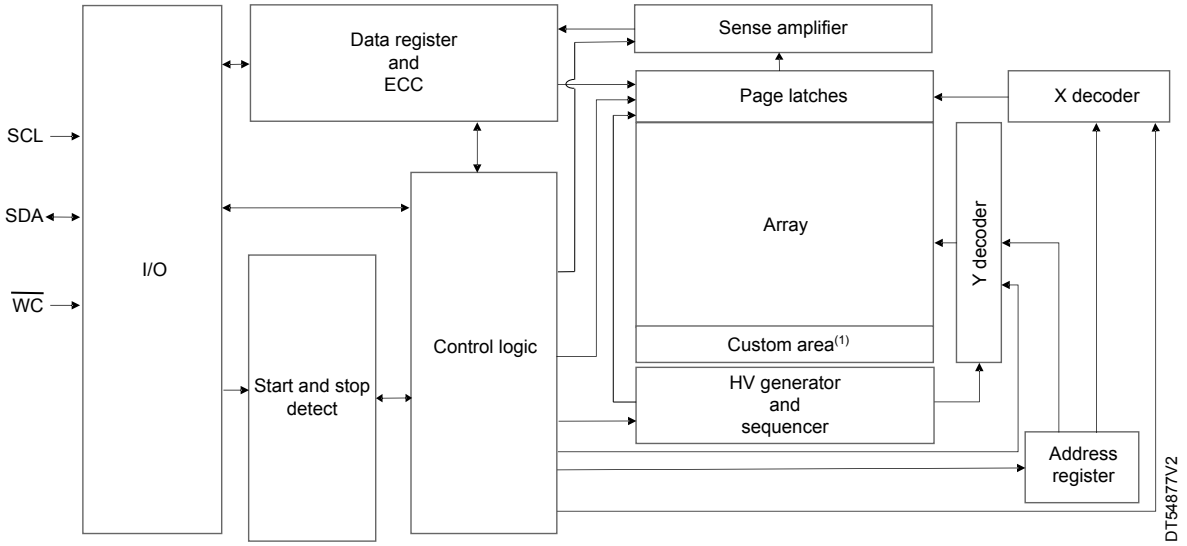
### 2.5.4 Power-down conditions

During power-down, when the $V_{CC}$ decreases to 0 V, the device must be in the standby power mode. This mode is reached after decoding a stop condition, with no internal write cycle in progress.

# 3 Memory organization

The memory is organized as shown below.

**Figure 3. Block diagram**




1. *DTI, CDA, SWP registers, unique identifier, and identification page.*

# 4 Device features

## 4.1 Device type identifier register (DTI)

The DTI is an 8-bit register permanently locked in read-only mode. This register is factory programmed with device type identifier bits (DTI3, DTI2, DTI1, and DTI0) set to 1011, and with device type identifier lock bit (DTIL) set to 1 to freeze definitively the register. DTI3, DTI2, DTI1, and DTI0 define the device type identifier address in the device select code. At power-up, the device loads the last configuration of DTI3, DTI2, DTI1, DTI0, and DTIL values.

This register is read by issuing the read device type identifier instruction. This instruction uses the same protocol and format as the random address read (from the memory array) except for the following differences (refer to Table 9, Table 10, and Table 11).

- Device type identifier = 1011
- MSB address bits A15, A14, and A13 must be equal to 111
- MSB address bits from A12 to A8 are don't care
- LSB address bits from A7 to A0 are don't care

The description of the device type identifier register is given in Table 2.

**Table 2. Device type identifier register**

| bit 7 | bit 6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|---|---|---|---|---|---|---|---|
| DTI3 = 1 | DTI2 = 0 | DTI1 = 1 | DTI0 = 1 | X(1) | X | X | DTIL = 1 |

1. *X = Don't care bit. Read as 0.*

*Note: The factory default value is 10110001.*

**Table 3. Device type identifier register description**

| Bits | Function |
|---|---|
| b7 to b4 | **DTI3, DTI2, DTI1, DTI0:** Device type identifier bits.<br>b7, b6, b5, b4 are used to configure the device type identifier of the device select code.<br>• (b7, b6, b5, b4) = (1, 0, 1, 1) the device type identifier is 1011 (factory default value)<br>Note: Bits b7 to b4 are locked upon factory delivery. |
| b3 to b1 | Don't care bits. Read as 0. (b3, b2, b1) = (0, 0, 0) |
| b0 | **DTIL**: Device type identifier lock bit.<br>b0 indicates that the DTI register status is in read-only mode.<br>• b0 = 1 the device type identifier lock bit is equal to 1 (factory default value)<br>Note: Bit b0 is locked upon factory delivery. |

## 4.2 Configurable device address register (CDA)

The CDA is an 8-bit register allowing the user to define a configurable device address (C2), and includes a specific bit, named device address lock (DAL), to permanently freeze the configurable device address register. This register can be read and written by issuing the read or write configurable device address instruction. These instructions use the same protocol and format as the random address read or page write (from/into memory array) except for the following differences (refer to Table 9, Table 10, and Table 11):

- Device type identifier = 1011
- MSB address bits A15, A14, and A13 must be equal to 110
- MSB address bits from A12 to A8 are don't care
- LSB address bits from A7 to A0 are don't care

C2 and DAL are defining the chip enable address in the device select code and the device address lock. These bits can be written and reconfigured with a write command.

At power-up or after reprogramming, the device loads the last configuration of C2 and DAL values. To prevent unwanted change of configurable device address bits, the M24M02E-U protects the CDA register permanently freezing it in read-only mode. The update of the CDA register is disabled (read-only) when the DAL bit is set to 1 (DAL = 1).

In the same way, the update of the CDA register is enabled when the DAL bit is set to 0 (DAL = 0). Sending more than one byte during a write configurable device address command aborts the write cycle (CDA register content does not change).

*Note:*

- *Updating the DAL bit from 0 to 1 is an irreversible action: the C2 and DAL bits cannot be updated anymore.*
- *If the write control input ($\overline{WC}$) is driven high, or if the DAL bit is set to 1 the write configurable device address command is not executed, the accompanying data byte is not acknowledged, as shown in Figure 6, and the write cycle does not start.*

The description of the configurable device address register is given in Table 4.

**Table 4. Configurable device address register**

| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|
| X(1) | X | X | X | C2 | X | X | DAL |

1. *X = Don't care bits. Read as 0.*

*Note: The factory default value is 00000000.*

**Table 5. Configurable device address register description**

| Bits | Function |
|---|---|
| b7 to b4, b2 and b1 | Don't care bits. Read as 0.<br>(b7, b6, b5, b4, b2, b1) = (0, 0, 0, 0, 0, 0) |
| b3 | **C2**: Configurable device address bit.<br>b3 is used to configure up to two possibilities of chip enable address:<br>• b3 = 0: the chip enable address is 0<br>• b3 = 1: the chip enable address is 1 |
| b0 | **DAL**: Device address lock bit.<br>b0 locks the CDA register in read-only mode:<br>• b0 = 0: bits b3 and b0 can be modified<br>• b0 = 1: bits b3 and b0 cannot be modified and therefore the CDA register is locked<br>Note: Bits b3 and b0 can be updated (if b0 = 0) in the same write instruction. Setting b0 from 0 to 1 is an irreversible action. |

## 4.3 Software write protection register (SWP)

The SWP is a nonvolatile 8-bit register that allows the user to protect a specific area of the memory against the write instruction. The SWP offers four nonvolatile bits to configure by the user:

- Two bits for setting the size of the write-protected memory, identified as block protection bits (BP0, BP1)
- One bit to enable or disable the write protection of the desired area, identified as write protect activation bit (WPA)
- One bit to definitively freeze the SWP in read-only mode, identified as write protection lock bit (WPL)

This register can be read and written by issuing the read or write software write protection register instructions. These instructions use the same protocol and format as the random address read or page write (from/into memory array) except for the following differences (refer to Table 9, Table 10, and Table 11):

- Device type identifier = 1011
- MSB address bits A15, A14, and A13 must be equal to 101
- MSB address bits from A12 to A8 are don't care
- LSB address bits from A7 to A0 are don't care

BP1 and BP0 are the block protection bits. WPL is the write protect lock bit and WPA is the write protect activation bit. These bits can be written and reconfigured with a write command. At power-up, the device loads the last configuration of the SWP register value.

The user can update the SWP register as often as the WPL bit stays at 0. Writing more than one byte discard the write cycle (software write protection register content is not changed).

To prevent unwanted change of software write-protection register bits, the M24M02E-U protects the SWP register, freezing it permanently in read-only mode. The update of the SWP register is disabled (read-only) when the WPL bit is set to 1 (WPL = 1). In the same way, the update of the SWP register is enabled when the WPL bit is set to 0 (WPL = 0).

When WPL is set to 1, and in the case of writing to the software write protection register, the device select and address bytes are acknowledged, the data byte is not acknowledged and the write cycle does not start.

*Note:*

- *Updating the WPL bit from 0 to 1 is an irreversible action: the WPA, BP1, BP0, and WPL bits cannot be updated any more*
- *If the write control input ($\overline{WC}$) is driven high or if the WPL bit is set to 1, the write command on the software write protection register is not executed and the accompanying data byte is not acknowledged, as shown in Figure 10*

The description of the software write protection register is given in the following table:

**Table 6. Software write protection register values**

| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|
| X(1) | X | X | X | WPA | BP1 | BP0 | WPL |

1. *X = Don't care bits. Read as 0.*

*Note: The factory default value is 00000000.*

**Table 7. Software write protection register description**

| Bits | Function |
|---|---|
| b7 to b4 | • Don't care bits. Read as 0. (b7, b6, b5, b4) = (0, 0, 0, 0) |
| b3 | **WPA**: Write protect activation bit.<br>b3 enables or disables the write protection:<br>• b3 = 0: no write protection. The whole memory can be written.<br>• b3 = 1: write protection active. The memory block is protected according to BP bits setting. |
| b2 to b1 | **BP1, BP0**: block protection bits<br>b2 and b1 define the size of the memory block to be protected against write instruction:<br>• (b2, b1) = (0, 0): the upper quarter of memory is write-protected<br>• (b2, b1) = (0, 1): the upper half of memory is write protected<br>• (b2, b1) = (1, 0): the upper ¾ of memory is write protected<br>• (b2, b1) = (1, 1): the whole memory is write protected |
| b0 | **WPL**: write protection lock bit<br>b0 locks the write protection register value.<br>• b0 = 0: bits [b3: b0] can be modified<br>• b0 = 1: bits [b3: b0] cannot be modified and the write protection register is locked.<br>Note: bits b3 to b0 can be updated (if b0 = 0) in the same write instruction. Setting b0 from 0 to 1 is an irreversible action. |

## 4.4 Identification page

This is an additional 256-byte page, permanently locked in read-only mode. The user can read it by issuing the read identification page instruction. This instruction uses the same protocol and format as the random address read (from the memory array), except for the following differences (refer to Table 9, Table 10, and Table 11):

- Device type identifier = 1011
- MSB address bits A15, A14, and A13 must be equal to 000
- MSB address bits from A12 to A8 are don't care
- LSB address bits from A7 to A0 define the byte address inside the identification page

This page also stores the unique identifier (see Section 4.5). As it is locked, the data bytes transferred during the write identification page instruction are not acknowledged (NO ACK). The identification page is set with the first 16 bytes containing the value of the UID. All the other bytes are written to FFh.

## 4.5 Unique identifier (UID)

The M24M02E-U provides an additional feature: a serial number programmed at factory level, and locked in read-only mode within the identification page. This preprogrammed, 16-byte unique ID is a 128-bit serial number.

The 128-bit serial number is unique across the all STMicroelectronics UID-family EEPROM devices.

This UID can be read by issuing the read identification page instruction.

- Device type identifier = 1011
- MSB address bits A15, A14, and A13 must be equal to 000
- MSB address bits from A12 to A8 must be equal to 0
- LSB address bits from A7 to A4 must be equal to 0
- LSB address bits from A3 to A0 define the UID byte address inside the identification page

The description of the UID is given in the following table.

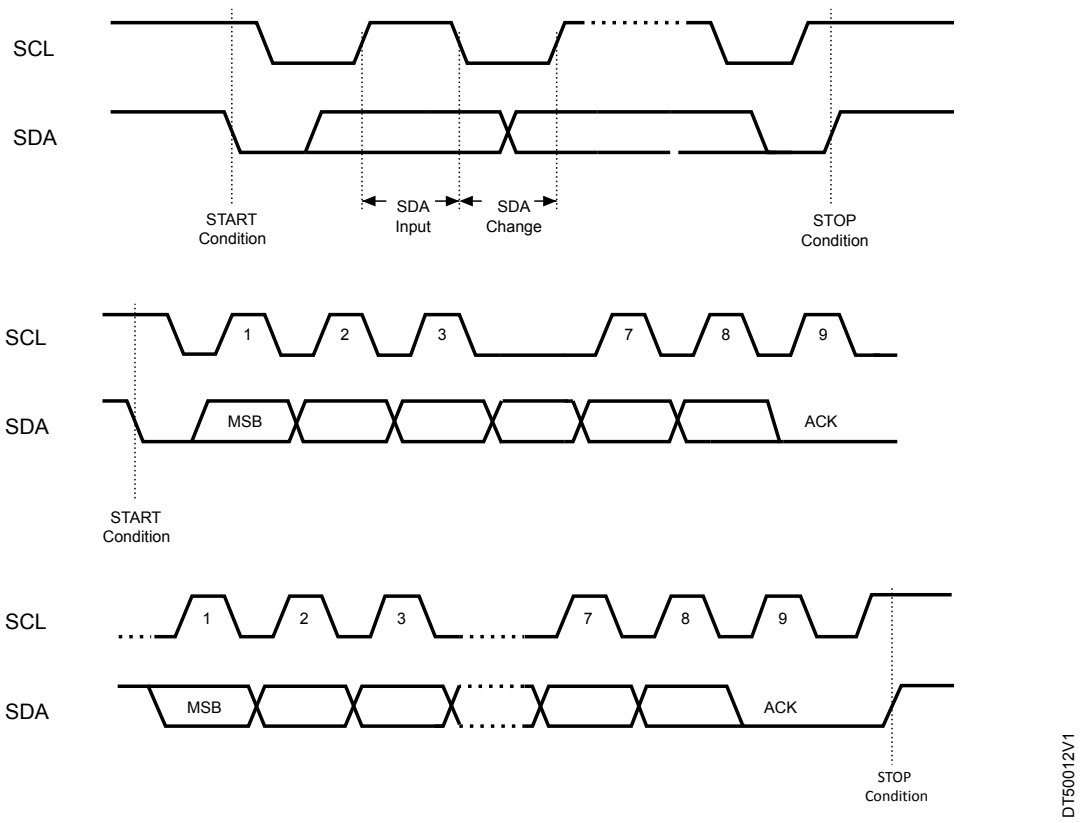
**Table 8. UID description**

| UID address in the identification page (Hex) | Content | Value (Hex) |
|---|---|---|
| 00 | Header - STMicroelectronics code | 20 |
| 01 | Header - Bus protocol | E0 |
| 02 | Header - Density | 12 |
| 03 | Header - Unused | FF |
| 04 | UID | |
| 05 | | |
| 06 | | |
| 07 | | |
| 08 | | |
| 09 | | |
| 0A | | |
| 0B | | |
| 0C | | |
| 0D | | |
| 0E | | |
| 0F | | |

# 5 Device operation

The device supports the I²C protocol summarized in Figure 4. Any device that sends data onto the bus is defined as a transmitter, and any device that reads the data is defined as a receiver. The device that controls the data transfer is known as the bus controller, and the other as the target. A data transfer can only be initiated by the bus controller, which also provides the serial clock for synchronization. The device is always a target in all communications.

**Figure 4. I²C bus protocol**



## 5.1 Start condition

The start condition is identified by a falling edge of serial data (SDA) while the serial clock (SCL) is stable in the high state. This condition must precede any data transfer instruction. The device continuously monitors the SDA and SCL for a start signal, except during a write cycle.

## 5.2 Stop condition

The stop condition is identified by a rising edge of serial data (SDA) while the serial clock (SCL) is stable in the high state. This condition terminates the communication between the device and the bus controller. A read instruction followed by NO ACK can be followed by a stop condition to force the device into the standby mode. A stop condition at the end of a write instruction triggers the internal write cycle.

## 5.3 Data input

During data input, the device samples the serial data (SDA) on the rising edge of the serial clock (SCL). For proper device operation, the SDA must be stable during the rising edge of the SCL, and the SDA signal must change only when the SCL is driven low.

## 5.4 Acknowledge bit (ACK)

The acknowledge bit is used to indicate a successful byte transfer. The bus transmitter, whether a bus controller or target device, releases serial data (SDA) after sending eight bits of data. During the ninth clock pulse period, the receiver pulls SDA low to acknowledge the receipt of the eight data bits.

## 5.5 Device addressing

To start communication between the bus controller and the target device, the bus controller must initiate a start condition. Following this, the bus controller sends the device select code and byte address as specified in Table 9, Table 10, and Table 11.

When the device select code is received, the device responds only if the bit b3 value match the values of the C2 bit programmed in the configurable device address register.

If a match occurs, the corresponding device gives an acknowledgment on serial data (SDA) during the ninth bit time. If the device does not acknowledge the device select code, the device deselects itself from the bus, and goes into standby mode (therefore it does not acknowledge the device select code).

The eighth bit is the read/write bit ($R\overline{W}$). This bit is set to 1 for read and 0 for write operations.

The 256-Kbyte (2-Mbit) are addressed with 18 address bits, the 16 lower address bits being defined by the two address bytes and the most significant address bit (A17 and A16) being included in the device select code (see Table 9).

**Table 9. Device select code**

| Features | Device type identifier bits | | | | Chip enable address bit(1) | Address bits | | $R\overline{W}$ |
|---|---|---|---|---|---|---|---|---|
| | Bit 7 (MSB)(2) | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 (LSB) |
| Memory | 1 | 0 | 1 | 0 | C2 | A17 | A16 | $R\overline{W}$ |
| Device type identifier | 1 | 0 | 1 | 1 | C2 | X(3) | X | $R\overline{W}$ |
| Configurable device address | 1 | 0 | 1 | 1 | C2 | X | X | $R\overline{W}$ |
| Software write protection | 1 | 0 | 1 | 1 | C2 | X | X | $R\overline{W}$ |
| Identification page | 1 | 0 | 1 | 1 | C2 | X | X | $R\overline{W}$ |
| UID | 1 | 0 | 1 | 1 | C2 | X | X | $R\overline{W}$ |

1. *C2 is compared with the value read on bit b3 of the CDA register.*
2. *The most significant bit, b7, is sent first.*
3. *X = Don't care bit.*

**Table 10. First byte address**

| Features | Bit 7 (MSB)(1) | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 (LSB) |
|---|---|---|---|---|---|---|---|---|
| Memory | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |
| Device type identifier | 1 | 1 | 1 | X(2) | X | X | X | X |
| Configurable device address | 1 | 1 | 0 | X | X | X | X | X |
| Software write protection | 1 | 0 | 1 | X | X | X | X | X |
| Identification page | 0 | 0 | 0 | X | X | X | X | X |
| UID | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

1. *The most significant bit, b7, is sent first.*
2. *X = Don't care bit.*

**Table 11. Second byte address**

| Features | Bit 7 (MSB)[1] | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 (LSB) |
|---|---|---|---|---|---|---|---|---|
| Memory | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| Device type identifier | X[2] | X | X | X | X | X | X | X |
| Configurable device address | X | X | X | X | X | X | X | X |
| Software write protection | X | X | X | X | X | X | X | X |
| Identification page | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| UID | 0 | 0 | 0 | 0 | A3 | A2 | A1 | A0 |

1. *The most significant bit, b7, is sent first.*
2. *X = Don't care bit*

# 6 Instructions

## 6.1 Write operations on memory array

Following a start condition the bus controller sends a device select code with the R/W bit ($R\overline{W}$) reset to 0. The device acknowledges this, as shown in Figure 5, and waits for two address bytes. The device responds to each address byte with an acknowledge bit, and then waits for the data byte. See Table 9, Table 10, and Table 11 how to address the memory array. The 256-Kbyte (2-Mbit) are addressed with 18 address bits, the 16 lower address bits being defined by the two address bytes and the most significant address bit (A17 and A16) being included in the device select code (see Table 9).

When the bus controller generates a stop condition immediately after a data byte ACK bit (in the tenth bit time slot), either at the end of a byte write or a page write, the internal write cycle $t_W$ is triggered. A stop condition at any other time slot does not trigger the internal write cycle.

After the stop condition and the successful completion of an internal write cycle ($t_W$), the device internal address counter is automatically incremented to point to the next byte after the last modified byte.

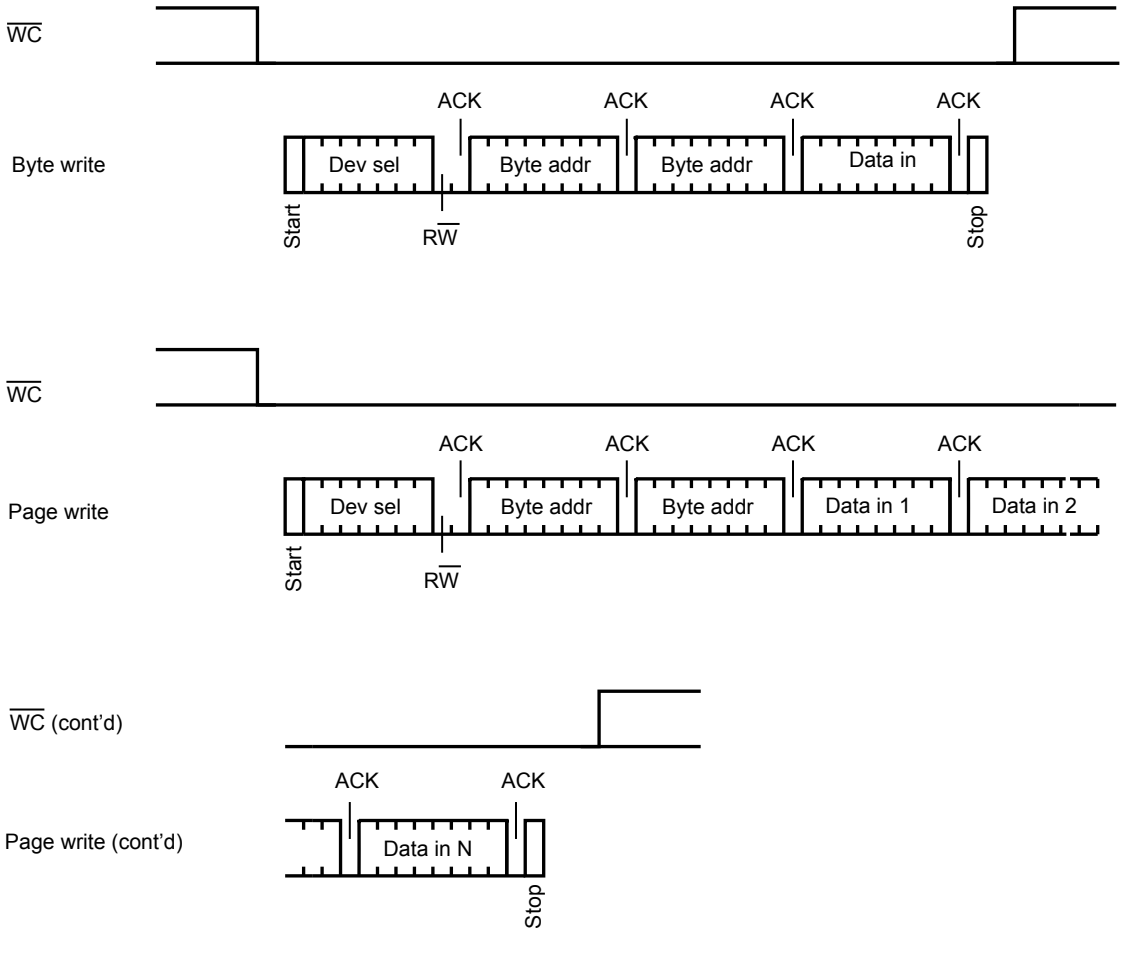
During the internal write cycle, serial data (SDA) is disabled internally, and the device does not respond to any requests.

If the addressed area is write protected by software through the SWP setting or hard protected through $\overline{WC}$ pin driven high, the write instruction is not executed and the accompanying data bytes are not acknowledged, as shown in Figure 6.

### 6.1.1 Byte write

After the device select code and the address bytes, the bus controller sends one data byte. If the addressed location is write-protected, through the SWP setting or through the $\overline{WC}$ pin being driven high, the device replies with NO ACK, and the location is not modified. If the addressed location is not write-protected, the device replies with ACK. The bus controller terminates the transfer by generating a stop condition, as shown in Figure 5.

**Figure 5. Write mode sequences without write protection (data write enabled)**



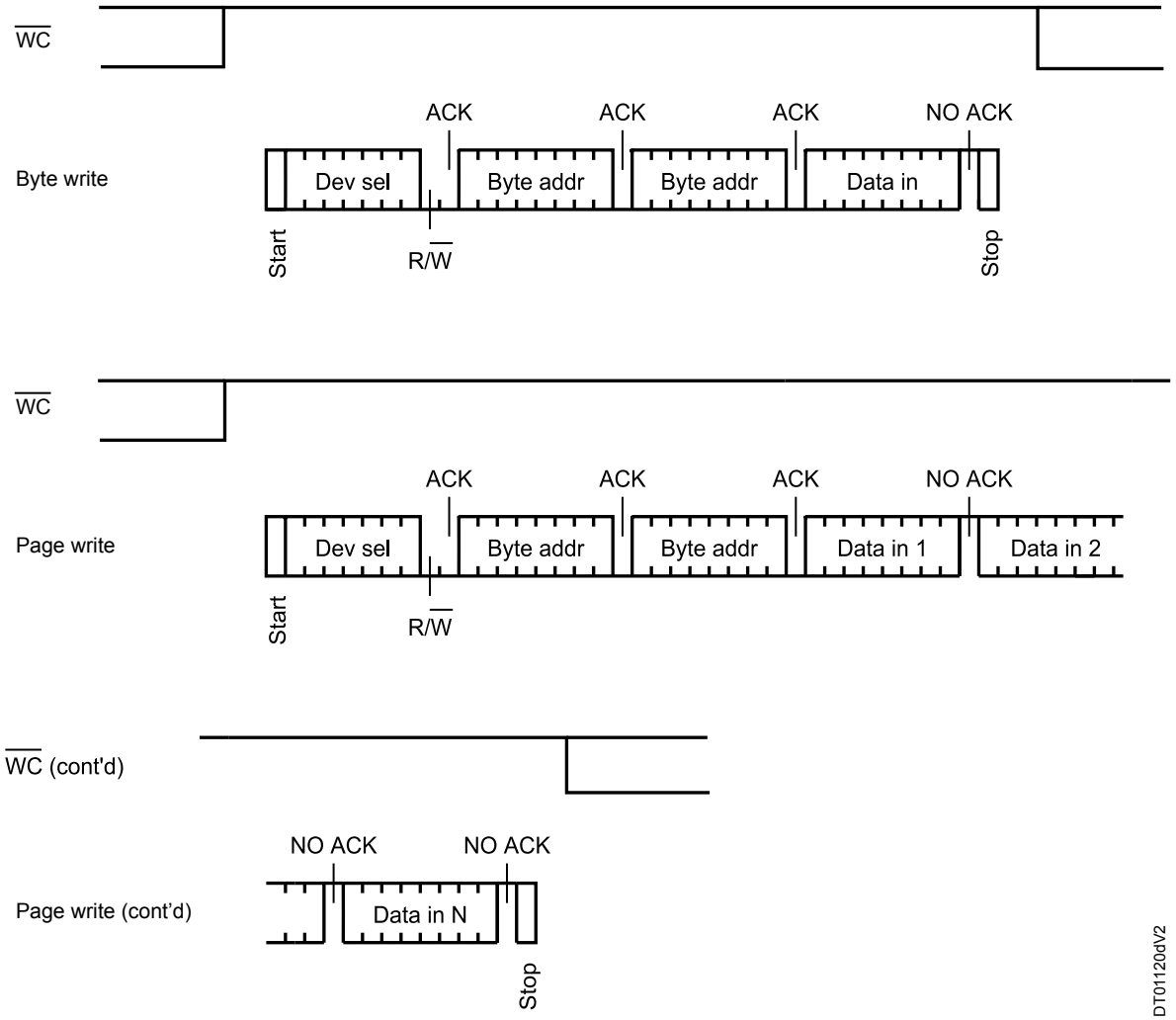
### 6.1.2 Page write

The page write mode allows up to 256-byte to be written in a single write cycle, provided they are all located on the same page. This means that the most significant memory address bits, from A17 to A8, are the same. If more bytes are sent than fit up to the end of the page, a roll-over occurs: the bytes exceeding the page end are written on the same page, from location 0.

The bus controller sends from 1 to 256 bytes of data, each of which is acknowledged by the device if the addressed bytes are not write-protected through the SWP settings or the $\overline{WC}$ pin (driven low). In the opposite case, when the addressed bytes are write-protected by SWP settings or the $\overline{WC}$ pin (driven high), the contents of the addressed memory location are not modified, and each data byte is followed by a NO ACK, as shown in Figure 6. After each transferred byte, the internal page address counter is incremented.

The transfer is terminated by the bus controller generating a stop condition.

**Figure 6. Write mode sequences with write protection (data write inhibited)**



## 6.2 Write operations on registers and minimizing delays

### 6.2.1 Write operation on DTI register and identification page

Write operations on the device type identifier (DTI) register and the identification page are not allowed.

These two features are delivered locked in read-only mode.

### 6.2.2 Write operation on CDA register

Write operations on the configurable device address register are performed according to the state of the device address lock bit (DAL) or the status of the $\overline{WC}$ line.

If the configurable device address register is write protected by software with DAL = 1 or hard protected with $\overline{WC}$ line driven high, the write operation on this register is not executed and the accompanying data byte is not acknowledged as shown in Figure 8.

Following a start condition the bus controller sends a device select code with the R/W bit ($R\overline{W}$) set to 0. The device acknowledges this, as shown in Figure 7, and waits for the address bytes where the register is located. The device responds to each address byte with an acknowledge bit, and then waits for the data byte. See in Table 9, Table 10, and Table 11 how to address the configurable device address register.

When the bus controller generates a stop condition immediately after the data byte ACK bit (in the tenth bit time slot), the internal write cycle $t_W$ is triggered. A stop condition at any other time slot does not trigger the internal write cycle.

During the internal write cycle, serial data (SDA) is disabled internally, and the device does not respond to any requests (NO ACK).

If the bit C2 has been reconfigured with a correct write command, the device acknowledges if the chip enable address of the device select code is equal to the new value of C2; otherwise NO ACK is sent. Sending more than one byte aborts the write cycle, and the configurable device address content does not change. Bits C2 and DAL can be updated (DAL = 0 to 1) in the same program instruction.

**Figure 7. Write CDA register (data write enabled)**

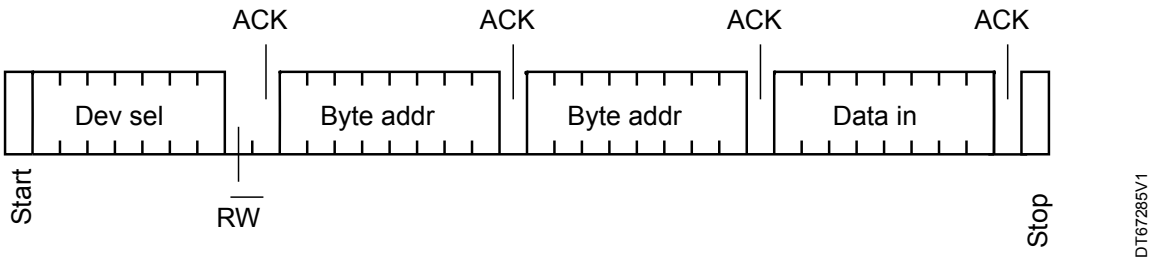
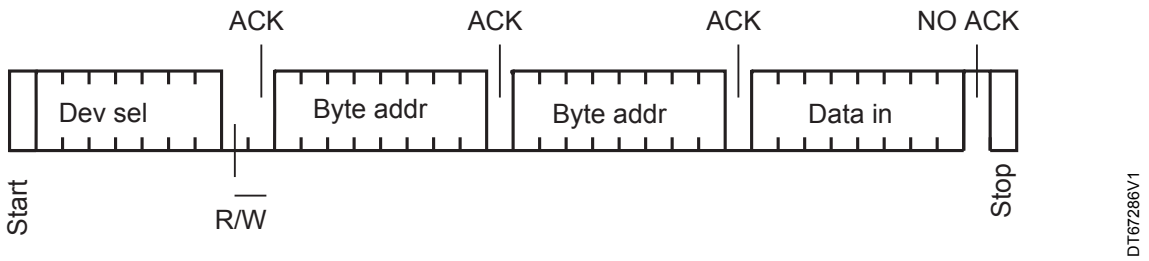



**Figure 8. Write CDA register (data write inhibited)**



### 6.2.3 Write operation on SWP register

Write operations on SWP register are performed according to the state of the write protect lock bit (WPL) or to the status of the $\overline{WC}$ line.

Following a start condition the bus controller sends a device select code with the R/W bit ($R\overline{W}$) set to 0. The device acknowledges this, as shown in Figure 9, and waits for the address bytes where the SWP register is located. The device responds to each address byte with an acknowledge bit, and then waits for the data byte. See in Table 9, Table 10, and Table 11 how to address the software write protection register.

When the bus controller generates a stop condition immediately after the data byte ACK bit (in the tenth bit time slot), the internal write cycle $t_W$ is triggered. A stop condition at any other time slot does not trigger the internal write cycle.

During the internal write cycle, serial data (SDA) is disabled internally, and the device does not respond to any requests (NO ACK).

Sending more than one byte aborts the write cycle (software write protection register content is not changed).

If the SWP register is already locked or hard write protected with $\overline{WC}$ line driven high, the write operation is not executed and the accompanying data byte is not acknowledged as shown in Figure 10.

**Figure 9. Write SWP register (data write enabled)**

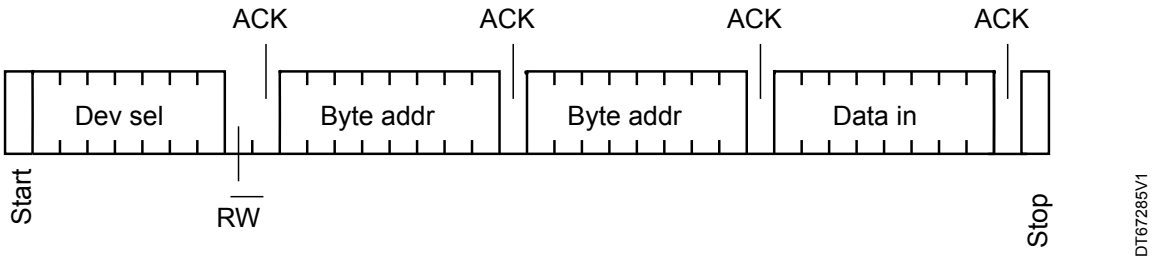
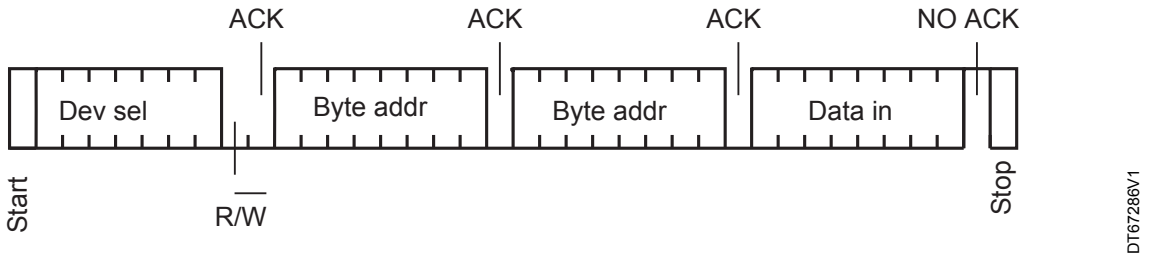



**Figure 10. Write SWP register (data write inhibited)**



### 6.2.4 Minimizing write delays by polling on ACK

During the internal write cycle, the device disconnects itself from the bus, and writes a copy of the data from its internal latches to the memory cells. The maximum write time ($t_W$) is shown in Table 19. AC characteristics in Fast-mode and Table 20. AC characteristics in Fast-mode Plus, but the typical time is shorter. The bus controller can implement a polling sequence to utilize this feature.

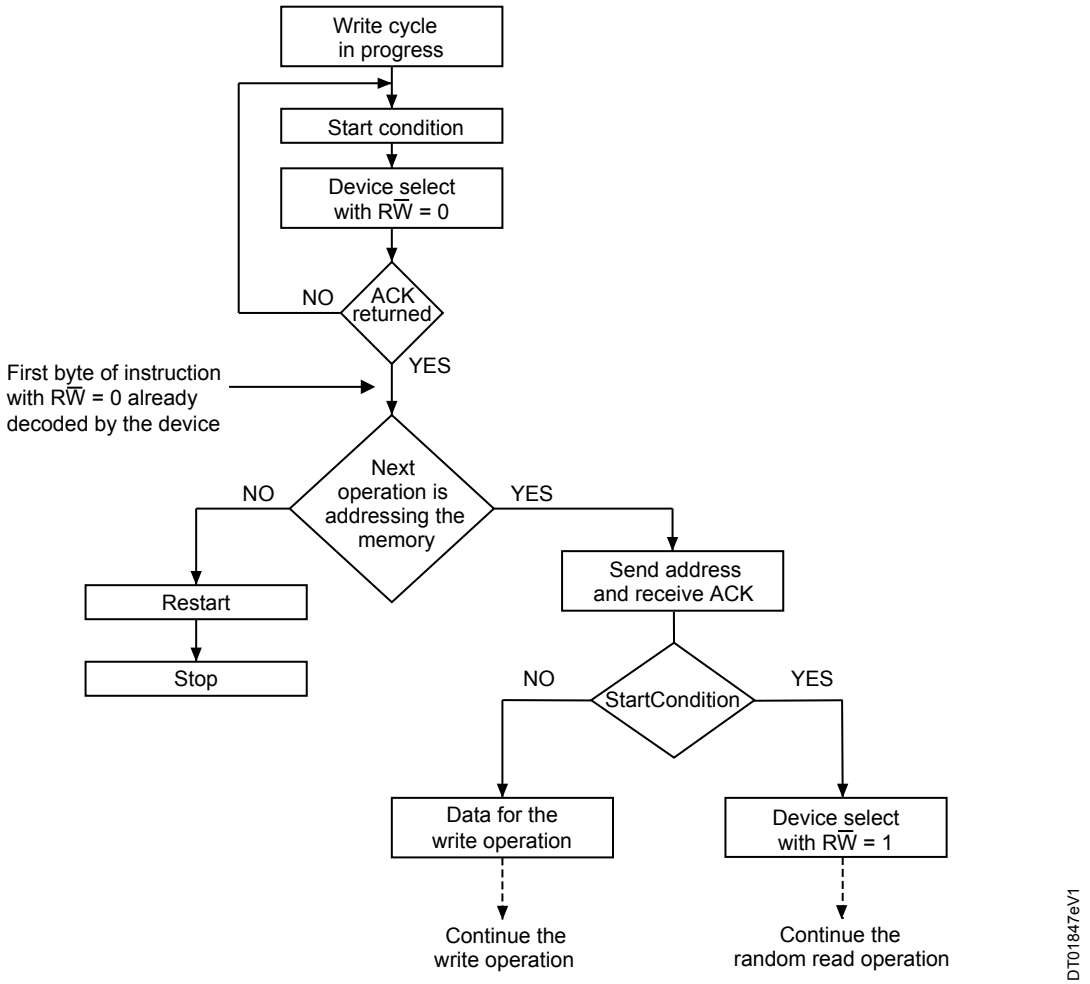
The sequence, as shown in Figure 11, is:

- Initial condition: A write cycle is in progress.
- Step 1: The bus controller issues a Start condition followed by a device select code (the first byte of the new instruction).
- Step 2: If the device is busy with the internal write cycle, NO ACK is returned and the bus controller goes back to step 1. If the device has terminated the internal write cycle, it responds with an ACK, indicating that the device is ready to receive the second part of the instruction (the first byte of this instruction having been sent during step 1).

*Note: Updating the configurable device address register with C2 reconfigured, the device returns ACK only if:*

- *The chip enable address of the device select code is equal to the new C2 value.*
- *An internal write cycle is completed (a new C2 values has been programmed in the chip enable register).*

**Figure 11. Write cycle polling flowchart using ACK**




*Note: The seven most significant bits of the device select code in a random read (bottom right box in the figure above) must match those of the device select code in the write operation (polling instruction).*

## 6.3 ECC (error correction code) and write cycling

ECC is an internal logic function transparent for the I²C communication protocol.

The ECC logic is implemented on each group of four bytes (located at addresses [4*N, 4*N+1, 4*N+2, 4*N+3], where N is an integer). Within a group, if a single bit happens to be erroneous during a read operation, the ECC detects and replaces it with the correct value. The read reliability is therefore much improved.

Even if the ECC function is performed on groups of four bytes, a single byte can be written/cycled independently. In this case, the ECC function also writes/cycles the three other bytes located in the same group. As a consequence, the maximum cycling budget is defined at group level and the cycling can be distributed over the four bytes of the group: the sum of the cycles seen by byte0, byte1, byte2, and byte3 of the same group must remain below the maximum value defined in Table 16. Cycling performance by groups of four bytes.

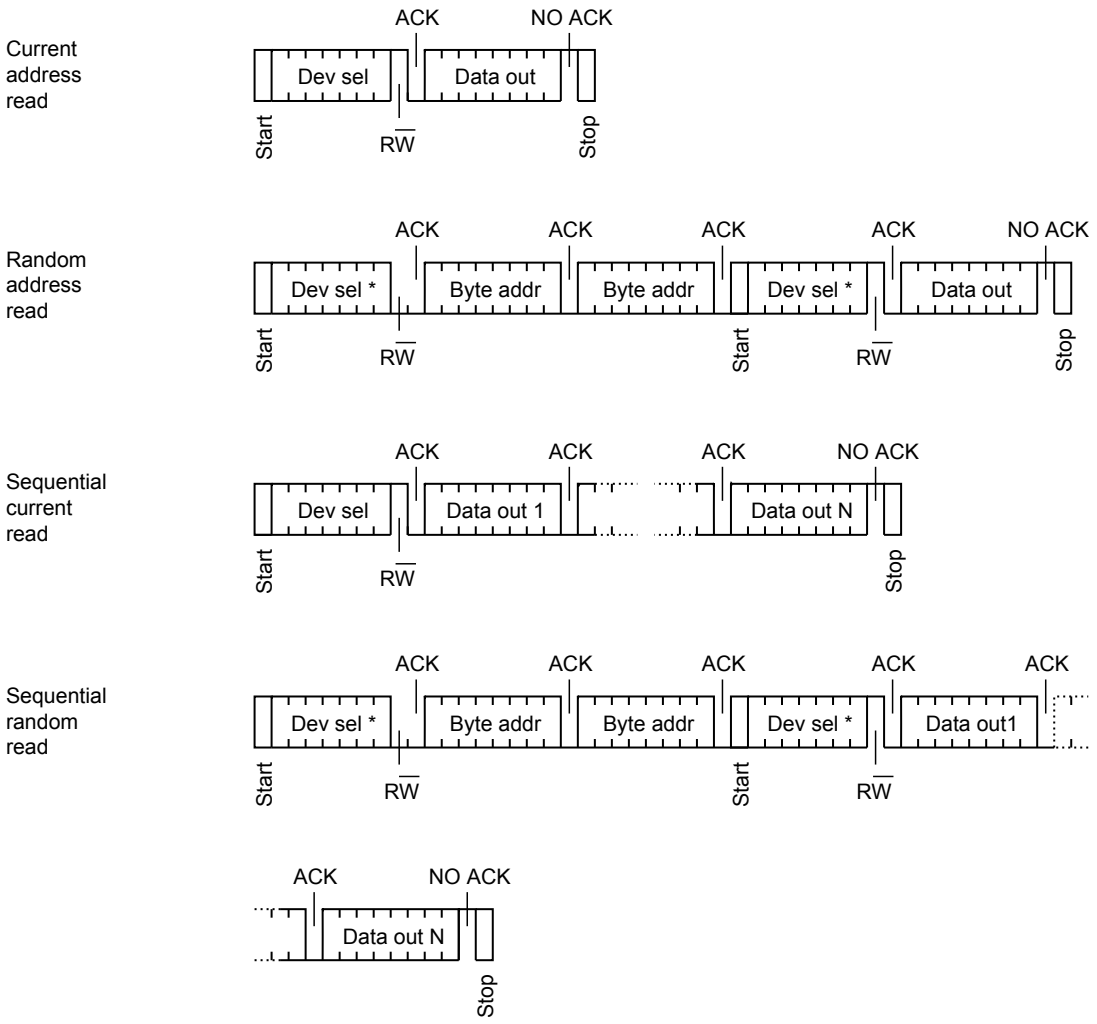
## 6.4 Read operations on memory array

Following a start condition the bus controller sends a device select code with the R/W bit ($\overline{RW}$) set to 0. The device acknowledges this and waits for the two-byte address. The device responds to each address byte with an acknowledge bit. Then, the bus controller sends another start condition, and repeats the device select code, with the $\overline{RW}$ bit set to 1. The device acknowledges this, and outputs the contents of the data. See in Table 9, Table 10, and Table 11, how to address the memory array.

After each byte read (data out), the device waits for an acknowledgment (data in) during the ninth bit time. If the bus controller does not acknowledge during this interval, the device terminates the data transfer and switches to its standby mode after a stop condition.

After the successful completion of a read operation, the internal address counter is incremented by one, to point to the next byte address.

**Figure 12. Read mode sequences**




*Note: *: The seven most significant bits of the first device select code in a random read must match those of the device select code of the write operation.*

### 6.4.1 Random address read

A dummy write is first performed to load the address into this address counter (as shown in Figure 12) but without sending a stop condition. Then, the bus controller sends another start condition, and repeats the device select code, with the R$\overline{W}$ bit set to 1. The device acknowledges this, and outputs the contents of the addressed byte. The bus controller must not acknowledge the byte, and terminates the transfer with a stop condition.

### 6.4.2 Current address read

For the current address read operation, following a start condition, the bus controller sends only a device select code with the R$\overline{W}$ bit set to 1. The device acknowledges this, and outputs the byte addressed by the internal address counter. The counter is then incremented. The bus controller terminates the transfer with a stop condition, as shown in Figure 12, without acknowledging the byte.

*Note: The address counter value is defined by instructions accessing either the memory, the registers, or the identification page. When accessing the registers or the identification page, the address counter value is loaded with the byte location, therefore the next current address read in the memory uses this new address counter value. When accessing the memory, it is safer to always use the random address read instruction (this instruction loads the address counter with the byte location to read in the memory, see Section 6.4.1) instead of the current address read instruction.*

### 6.4.3 Sequential read

This operation can be used after a current address read or a random address read. The bus controller does acknowledge the data byte output, and sends additional clock pulses so that the device continues to output the next byte in sequence. To terminate the stream of bytes, the bus controller must not acknowledge the last byte, and must generate a stop condition, as shown in Figure 12.

The output data comes from consecutive addresses, with the internal address counter automatically incremented after each byte output. After the last memory address, the address counter rolls-over, and the device continues to output data from the memory address 00h.

## 6.5 Read operations on registers and identification page

### 6.5.1 Read operation on DTI register

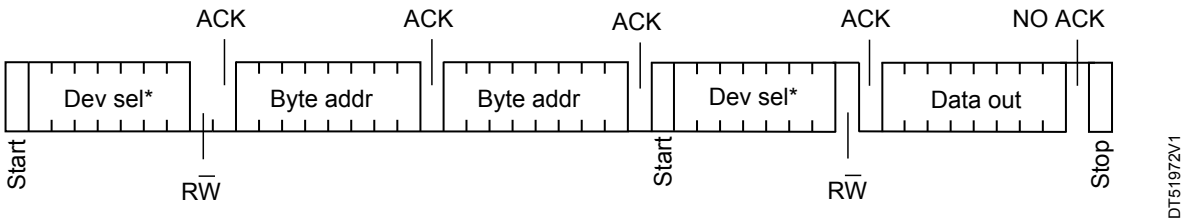
Following a start condition the bus controller sends a device select code with the R/W bit ($R\overline{W}$) set to 0. The device acknowledges this and waits for the address bytes where the DTI register is located. The device responds to each address byte with an acknowledge bit. Then, the bus controller sends another start condition, and repeats the device select code, with the $R\overline{W}$ bit set to 1. The device acknowledges this, and outputs the contents of the DTI register. See in Table 9, Table 10, and Table 11 how to address the device type identifier register.

After the successful completion of a read device type identifier, the device internal address counter is not incremented by one, to point to the next byte address. Reading more than one byte with the sequential random read command loops on reading the device type identifier register value.

To terminate the stream of data byte, the bus controller must not acknowledge the byte, and must generate a stop condition, as shown in Figure 13.

The device type identifier register cannot be read while a write cycle ($t_W$) is ongoing.

**Figure 13. Random read DTI register**




*Note: *: The seven most significant bits of the first device select code in a random read must match those of the device select code in the write operation.*

### 6.5.2 Read operation on CDA register

Following a start condition the bus controller sends a device select code with the R/W bit ($R\overline{W}$) set to 0. The device acknowledges this and waits for the address bytes where the CDA register is located. The device responds to each address byte with an acknowledge bit. Then, the bus controller sends another start condition, and repeats the device select code, with the $R\overline{W}$ bit set to 1. The device acknowledges this, and outputs the contents of the CDA register. See in Table 9, Table 10, and Table 11 how to address the configurable device address register.

After the successful completion of a read configurable device address, the device internal address counter is not incremented by one, to point to the next byte address. Reading more than one byte loops on reading the configurable device address register value.

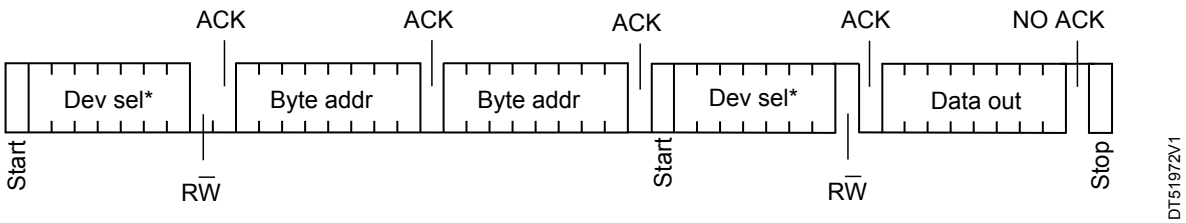
To terminate the stream of data byte, the bus controller must not acknowledge the byte, and must generate a stop condition, as shown in Figure 14.

The configurable device address register cannot be read while a write cycle ($t_W$) is ongoing.

The configurable device address bit (C2) value can be checked by sending the device select code.

- If the chip enable address b3 sent in the device select code matches the C2 value, the device sends an ACK.
- Otherwise, the device answers NO ACK.

**Figure 14. Random read CDA register**




*Note: *: The seven most significant bits of the first device select code in a random read must match those of the device select code in the write operation.*

### 6.5.3 Read operation on SWP register

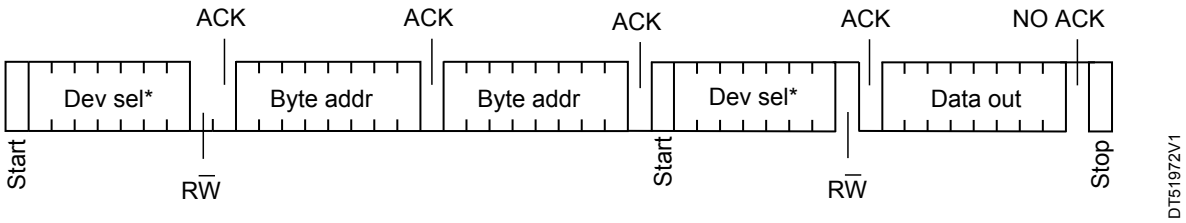
Following a start condition the bus controller sends a device select code with the R/W bit ($R\overline{W}$) set to 0. The device acknowledges this and waits for the address bytes where the SWP register is located. The device responds to each address byte with an acknowledge bit. Then, the bus controller sends another start condition, and repeats the device select code, with the $R\overline{W}$ bit set to 1. The device acknowledges this, and outputs the contents of the SWP register. See Table 9, Table 10, and Table 11 how to address the software write protection register.

After the successful completion of a read operation on SWP, the device internal address counter is not incremented by one, to point to the next byte address. Reading more than one-byte with the sequential random read command loops on reading the SWP register value.

To terminate the stream of data byte, the bus controller must not acknowledge the byte, and must generate a stop condition, as shown in Figure 15.

The SWP register cannot be read while a write cycle ($t_W$) is ongoing.

**Figure 15. Random read SWP register**




*Note: *: The seven most significant bits of the first device select code in a random read must match those of the device select code in the write operation.*

### 6.5.4 Read operation on identification page

Following a start condition the bus controller sends a device select code with the R/W bit ($R\overline{W}$) set to 0. The device acknowledges this and waits for the address bytes where the identification page is located. The device responds to each address byte with an acknowledge bit. Then, the bus controller sends another start condition, and repeats the device select code, with the $R\overline{W}$ bit set to 1. The device acknowledges this, and outputs the contents of the identification page. See in Table 9, Table 10, and Table 11 how to address the identification page.

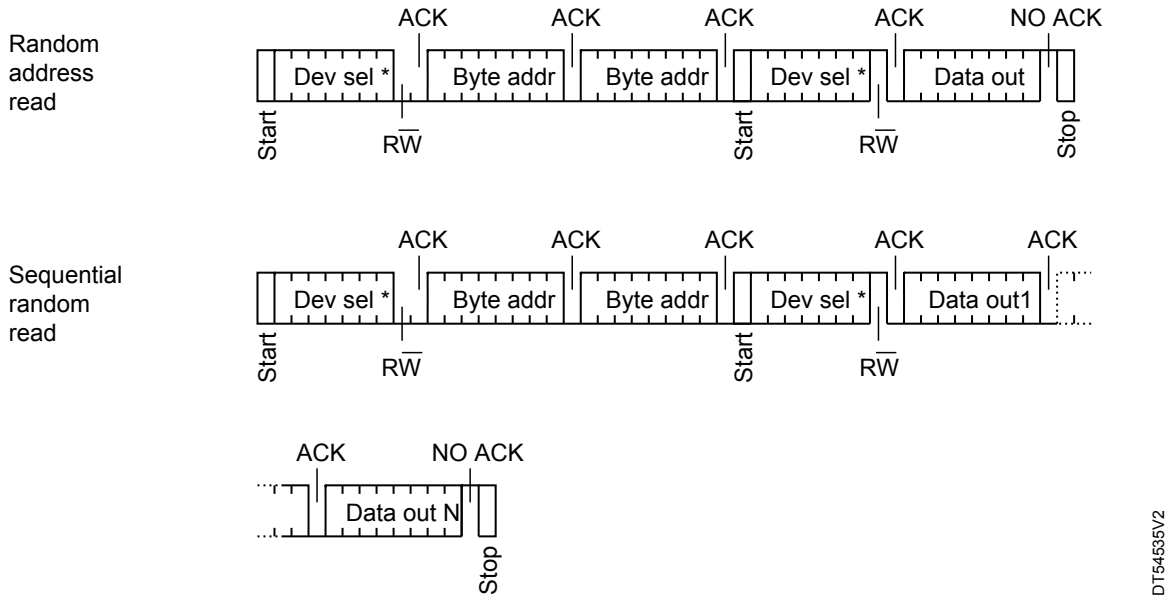
After each byte read (data out), the device waits for an acknowledgment (data in) during the ninth bit time.

The output data of the identification page comes from consecutive addresses, with the internal address counter automatically incremented after each byte output. After the last identification page address (FFh), the address counter rolls-over, and the device continues to output data from identification page address 00h.

To terminate the stream of data byte, the bus controller must not acknowledge the byte, and must generate a stop condition, as shown in Figure 16.

If the bus controller does not acknowledge during this ninth time, the device terminates the data transfer as shown in Figure 16 and switches to its standby mode.

**Figure 16. Random read identification page**




*Note: *: The seven most significant bits of the first device select code in a random read must match those of the device select code in the write operation.*

### 6.5.5 Read lock status on identification page

The lock or unlock status of the identification page can be checked by transmitting a specific truncated command.

Following a start condition the bus controller sends a device select code with the R/W bit ($\overline{RW}$) set to 0. The device acknowledges this and waits for the address bytes where the identification page is located. The device responds to each address byte with an acknowledge bit, and then waits for the data byte. See in Table 9, Table 10, and Table 11 how to address the identification page.

The device returns an acknowledge bit after the data byte if the identification page is unlocked (unlock status) as shown in Figure 17, otherwise a NO ACK bit as shown in Figure 18, if the identification page is locked (lock status).

Right after this, it is recommended to transmit to the device a start condition followed by a stop condition, so that:

- Start: the truncated command is not executed because the start condition resets the device internal logic
- Stop: the device is then set back into standby mode by the stop condition

**Figure 17. Read lock status (identification page unlocked)**

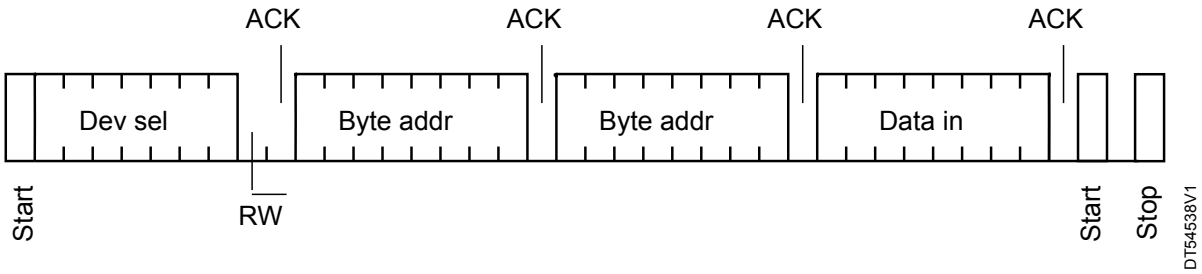
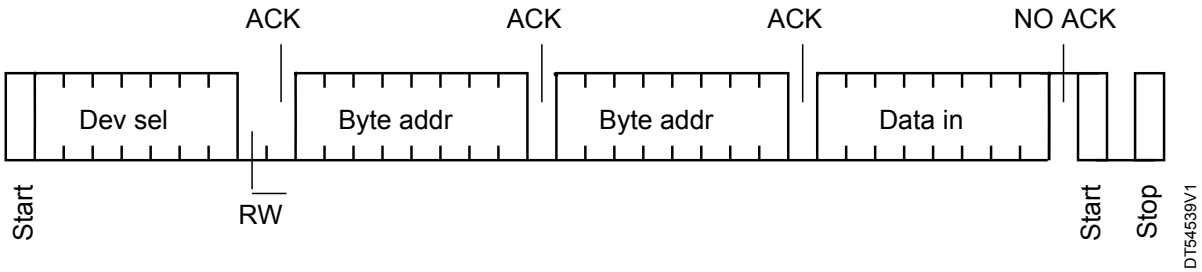



**Figure 18. Read lock status (identification page locked)**




*Note: As the identification page is delivered in read-only mode, the EEPROM consistently behaves as described in Figure 18.*

# 7 Initial delivery state

At factory delivery, the device is delivered with:

- All the memory array bits set to 1 (each byte contains FFh)
- The DTI register locked and set to 10110001 (B1h)
- The CDA register set to 00000000 (00h)
- The SWP register set to 0000000 (00h)
- The identification page is locked and set with the first 16 bytes containing the value of the UID. The content of the following bytes is FFh.

# 8 Maximum ratings

Stressing the device outside the ratings listed in Table 12 may permanently damage it. These are stress ratings only, and operation of the device at these, or any other conditions outside those indicated in the operating sections of this specification, is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Table 12. Absolute maximum ratings**

| Symbol | Parameter | Min. | Max. | Unit |
|---|---|---|---|---|
| $T_A$ | Ambient operating temperature | -40 | 130 | °C |
| $T_{STG}$ | Storage temperature | -65 | 150 | °C |
| $T_{LEAD}$ | Lead temperature during soldering | See note(1) | | °C |
| $I_{OL}$ | DC output current (SDA = 0) | - | 5 | mA |
| $V_{IO}$ | Input or output range | -0.50 | 6.5 | V |
| $V_{CC}$ | Supply voltage | -0.50 | 6.5 | V |
| $V_{ESD}$ | Electrostatic pulse (human body model)(2) | - | 4000 | V |

1. *Compliant with JEDEC standard J-STD-020 (for small body, Sn-Pb or Pb-free assembly), the ST ECOPACK 7191395 specification, and the European directive on restrictions of hazardous substances (RoHS directive 2011/65/EU of July 2011).*
2. *Positive and negative pulses applied on different combinations of pin connections, according to ANSI/ESDA/JEDEC JS-001 (C1 = 100 pF, R1 = 1500 Ω, R2 = 500 Ω).*

# 9 DC and AC parameters

This section summarizes the operating and measurement conditions, and the DC and AC characteristics.

**Table 13. Operating conditions**

| Symbol | Parameter | Min. | Max. | Unit |
|---|---|---|---|---|
| $V_{CC}$ | Supply voltage | 1.6 | 5.5 | V |
| $T_A$ | Ambient operating temperature | -40 | 85 | °C |
| $f_C$ | Operating clock frequency | - | 1 | MHz |

**Table 14. AC measurement conditions**

| Symbol | Parameter | Min. | Max. | Unit |
|---|---|---|---|---|
| $C_{bus}$ | Load capacitance | - | 100 | pF |
| - | SCL input rise/fall time, SDA input fall time | - | 50 | ns |
| - | Input levels | 0.2 $V_{CC}$ to 0.8 $V_{CC}$ | | V |
| - | Input and output timing reference levels | 0.3 $V_{CC}$ to 0.7 $V_{CC}$ | | V |

**Figure 19. AC measurement I/O waveform**

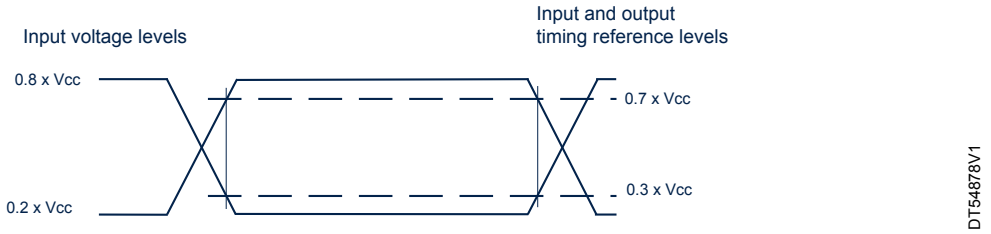



**Table 15. Input parameters**

| Symbol | Parameter | Test condition | Min. | Max. | Unit |
|---|---|---|---|---|---|
| $C_{IN}$(1) | Input capacitance (SDA) | - | - | 8 | pF |
| $C_{IN}$(1) | Input capacitance (other pins) | - | - | 6 | pF |
| $Z_L$(2) | Input impedance ($\overline{WC}$)(3) | $V_{IN}$ < 0.3 $V_{CC}$ | 30 | - | kΩ |
| $Z_H$(2) | | $V_{IN}$ > 0.7 $V_{CC}$ | 500 | - | kΩ |

1. *Specified by design - Not tested in production.*
2. *Evaluated by characterization - Not tested in production.*
3. *The memory is selected (after a start condition).*

**Table 16. Cycling performance by groups of four bytes**

| Symbol | Parameter | Test condition | Max. | Unit |
|---|---|---|---|---|
| Ncycle | Write cycle endurance(1) | $T_A \le 25$ °C, $V_{CC}(min) < V_{CC} < V_{CC}(max)$ | 4.000.000 | Write cycle(2) |
| | | $T_A = 85$ °C, $V_{CC}(min) < V_{CC} < V_{CC}(max)$ | 1.200.000 | |

1. *The write cycle endurance is defined by characterization and qualification. For devices embedding the ECC functionality, the write cycle endurance is defined for a group of four bytes located at addresses [4*N, 4*N+1, 4*N+2, 4*N+3] where N is an integer.*
2. *A Write cycle is executed when either a page write, a byte write, or a write registers instruction is decoded. When using the byte write, or the page write, refer also to Section 6.3: ECC (error correction code) and write cycling.*

**Table 17. Memory cell data retention**

| Parameter | Test condition | Min. | Unit |
|---|---|---|---|
| Data retention(1) | $T_A$ = 55 °C | 200 | Year |

1. *The data retention behaviour is checked in production, while the data retention limit is extracted from the characterization and qualification results.*

**Table 18. DC characteristics**

| Symbol | Parameter | Test conditions | Min. | Max. | Unit |
|---|---|---|---|---|---|
| $I_{LI}$ | Input leakage current (SCL, SDA) | $V_{IN} = V_{SS}$ or $V_{CC}$; device in standby mode | - | ± 2 | µA |
| $I_{LO}$ | Output leakage current | SDA in high-Z, external voltage applied on SDA: $V_{SS}$ or $V_{CC}$ | - | ± 2 | µA |
| $I_{CC}$ | Supply current (read) | $f_C$ = 400 kHz | - | 0.5 | mA |
| | | $f_C$ = 1 MHz | - | 1 | mA |
| $I_{CC0}$ | Supply current (write) | Averaged over $t_W$, $V_{CC} \le 3.3$ V | - | 1.2(1) | mA |
| | | Averaged over $t_W$, $V_{CC} > 3.3$ V | - | 2(1) | |
| $I_{CC1}$ | Standby supply current | Device not selected(2) $V_{IN} = V_{SS}$ or $V_{CC}$; $V_{CC} < 2.5$ V | - | 1 | µA |
| | | Device not selected(2) $V_{IN} = V_{SS}$ or $V_{CC}$; $V_{CC} \ge 2.5$ V | - | 2 | µA |
| $V_{IL}$ | Input low voltage (SCL, SDA, $\overline{WC}$) | 1.6 V ≤ $V_{CC}$ < 2.5 V | -0.45 | 0.25 $V_{CC}$ | V |
| | | 2.5 V ≤ $V_{CC}$ ≤ 5.5 V | -0.45 | 0.30 $V_{CC}$ | V |
| $V_{IH}$ | Input high voltage (SCL, SDA) | 1.6 V ≤ $V_{CC}$ < 2.5 V | 0.75 $V_{CC}$ | 6.5 | V |
| | | 2.5 V ≤ $V_{CC}$ ≤ 5.5 V | 0.70 $V_{CC}$ | 6.5 | V |
| | Input high voltage ($\overline{WC}$) | 1.6 V ≤ $V_{CC}$ < 2.5 V | 0.75 $V_{CC}$ | $V_{CC}$ + 0.6 | V |
| | | 2.5 V ≤ $V_{CC}$ ≤ 5.5 V | 0.70 $V_{CC}$ | $V_{CC}$ + 0.6 | V |
| $V_{OL}$ | Output low | $I_{OL}$ = 1 mA, $V_{CC}$ = 1.6 V | - | 0.2 | V |
| | | $I_{OL}$ = 2.1 mA, $V_{CC}$ = 2.5 V or $I_{OL}$ = 3 mA, $V_{CC}$ = 5.5 V | - | 0.4 | V |

1. *Evaluated by characterization - Not tested in production.*
2. *The device is not selected after power-up, after a read instruction (after the stop condition), or after the completion of the internal write cycle $t_W$ ($t_W$ is triggered by the correct decoding of a write instruction).*

**Table 19. AC characteristics in Fast-mode**

| Symbol | Alt. | Parameter | Min. | Max. | Unit |
|---|---|---|---|---|---|
| $f_C$ | $f_{SCL}$ | Clock frequency | - | 400 | kHz |
| $t_{CHCL}$ | $t_{HIGH}$ | Clock pulse width high | 600 | - | ns |
| $t_{CLCH}$ | $t_{LOW}$ | Clock pulse width low | 1300 | - | ns |
| $t_{QL1QL2}$(1) | $t_F$ | SDA (out) fall time | 20(2) | 300 | ns |
| $t_{XH1XH2}$(1) | $t_R$ | Input signal rise time | (3) | (3) | ns |
| $t_{XL1XL2}$(1) | $t_F$ | Input signal fall time | (3) | (3) | ns |
| $t_{DXCH}$ | $t_{SU:DAT}$ | Data in setup time | 100 | - | ns |
| $t_{CLDX}$ | $t_{HD:DAT}$ | Data in hold time | 0 | - | ns |
| $t_{CLQX}$(4) | $t_{DH}$ | Data out hold time | 100 | - | ns |
| $t_{CLQV}$(5) | $t_{AA}$ | Clock low to next data valid (access time) | - | 900 | ns |
| $t_{CHDL}$ | $t_{SU:STA}$ | Start condition setup time | 600 | - | ns |
| $t_{DLCL}$ | $t_{HD:STA}$ | Start condition hold time | 600 | - | ns |
| $t_{CHDH}$ | $t_{SU:STO}$ | Stop condition set up time | 600 | - | ns |
| $t_{DHDL}$ | $t_{BUF}$ | Time between Stop condition and next Start condition | 1300 | - | ns |
| $t_{WLDL}$(1)(6) | $t_{SU:WC}$ | $\overline{WC}$ set up time (before the start condition) | 0 | - | µs |
| $t_{DHWH}$(1)(7) | $t_{HD:WC}$ | $\overline{WC}$ hold time (after the stop condition) | 1 | - | µs |
| $t_W$ | $t_{WR}$ | Write cycle time | - | 4(8) | ms |
| $t_{NS}$(1) | - | Pulse width ignored (input filter on SCL and SDA) - single glitch | - | 50 | ns |
| $t_{WU}$(9)(10) | - | Wake-up time | - | 5 | µs |

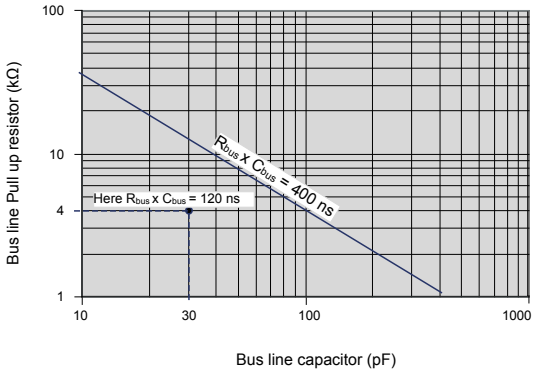
1. *Evaluated by characterization - Not tested in production.*
2. *With $C_L$ = 10 pF.*
3. *There is no min. or max. values for the input signal rise and fall times. It is however recommended by the I²C specification that the input signal rise and fall times be more than 20 ns and less than 300 ns when $f_C$ < 400 kHz.*
4. *To avoid spurious start and stop conditions, a minimum delay is placed between SCL = 1 and the falling or rising edge of SDA.*
5. *$t_{CLQV}$ is the time (from the falling edge of SCL) required by the SDA bus line to reach either 0.3$V_{CC}$ or 0.7$V_{CC}$, assuming that the $R_{bus}$ × $C_{bus}$ time constant is within the values specified in Figure 20.*
6. *$\overline{WC}$ = 0 setup time condition to enable the execution of a write command.*
7. *$\overline{WC}$ = 0 hold time condition to enable the execution of a write command.*
8. *3.3 ms typical.*
9. *Specified by design - Not tested in production.*
10. *Wake-up time: Delay between the $V_{CC}$(min) stable and the first accepted command.*

**Table 20. AC characteristics in Fast-mode Plus**

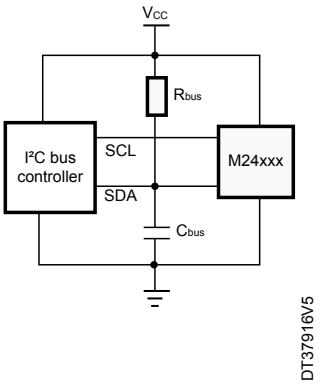
| Symbol | Alt. | Parameter | Min. | Max. | Unit |
|---|---|---|---|---|---|
| $f_C$ | $f_{SCL}$ | Clock frequency | - | 1 | MHz |
| $t_{CHCL}$ | $t_{HIGH}$ | Clock pulse width high | 260 | - | ns |
| $t_{CLCH}$ | $t_{LOW}$ | Clock pulse width low | 500 | - | ns |
| $t_{XH1XH2}$ | $t_R$ | Input signal rise time | (1) | (1) | ns |
| $t_{XL1XL2}$ | $t_F$ | Input signal fall time | (1) | (1) | ns |
| $t_{QL1QL2}$(2) | $t_F$ | SDA (out) fall time | 20(3) | 120 | ns |
| $t_{DXCH}$ | $t_{SU:DAT}$ | Data in setup time | 50 | - | ns |
| $t_{CLDX}$ | $t_{HD:DAT}$ | Data in hold time | 0 | - | ns |
| $t_{CLQX}$(4) | $t_{DH}$ | Data out hold time | 100 | - | ns |
| $t_{CLQV}$(5) | $t_{AA}$ | Clock low to next data valid (access time) | - | 450 | ns |
| $t_{CHDL}$ | $t_{SU:STA}$ | Start condition setup time | 250 | - | ns |
| $t_{DLCL}$ | $t_{HD:STA}$ | Start condition hold time | 250 | - | ns |
| $t_{CHDH}$ | $t_{SU:STO}$ | Stop condition setup time | 250 | - | ns |
| $t_{DHDL}$ | $t_{BUF}$ | Time between Stop condition and next Start condition | 500 | - | ns |
| $t_{WLDL}$(2)(6) | $t_{SU:WC}$ | $\overline{WC}$ set up time (before the start condition) | 0 | - | µs |
| $t_{DHWH}$(2)(7) | $t_{HD:WC}$ | $\overline{WC}$ hold time (after the stop condition) | 1 | - | µs |
| $t_W$ | $t_{WR}$ | Write cycle time | - | 4(8) | ms |
| $t_{NS}$(2) | - | Pulse width ignored (input filter on SCL and SDA) | - | 50 | ns |
| $t_{WU}$(9)(10) | - | Wake-up time | - | 5 | µs |

1. *There is no min. or max. values for the input signal rise and fall times. It is however recommended by the I²C specification that the input signal rise and fall times be less than 120 ns when $f_C$ < 1 MHz.*
2. *Evaluated by characterization - Not tested in production.*
3. *With $C_L$ = 10 pF.*
4. *To avoid spurious Start and Stop conditions, a minimum delay is placed between SCL = 1 and the falling or rising edge of SDA.*
5. *$t_{CLQV}$ is the time (from the falling edge of SCL) required by the SDA bus line to reach either 0.3 $V_{CC}$ or 0.7 $V_{CC}$, assuming that the Rbus × Cbus time constant is within the values specified in Figure 21.*
6. *$\overline{WC}$ = 0 setup time condition to enable the execution of a write command.*
7. *$\overline{WC}$ = 0 hold time condition to enable the execution of a write command.*
8. *3.3 ms typical.*
9. *Specified by design - Not tested in production.*
10. *Wake-up time: Delay between the $V_{CC}$(min) stable and the first accepted commands.*

**Figure 20. $R_{bus}$ value versus bus parasitic capacitance ($C_{bus}$) for an $I^2C_{bus}$ ($f_C$ = 400 kHz)**

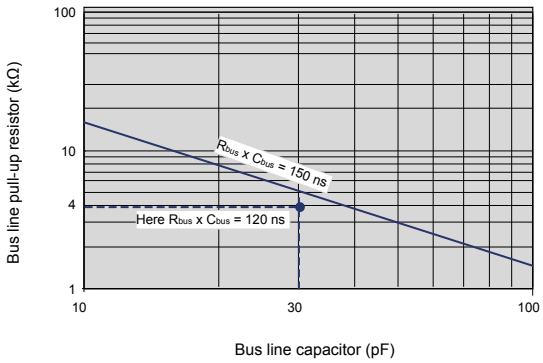



The $R_{bus}$ x $C_{bus}$ time constant must be below the 400 ns time constant line displayed on the left

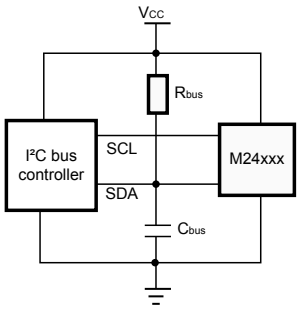



DT37916V5

**Figure 21. $R_{bus}$ value versus bus parasitic capacitance ($C_{bus}$) for an $I^2C$ bus ($f_C$ = 1 MHz)**

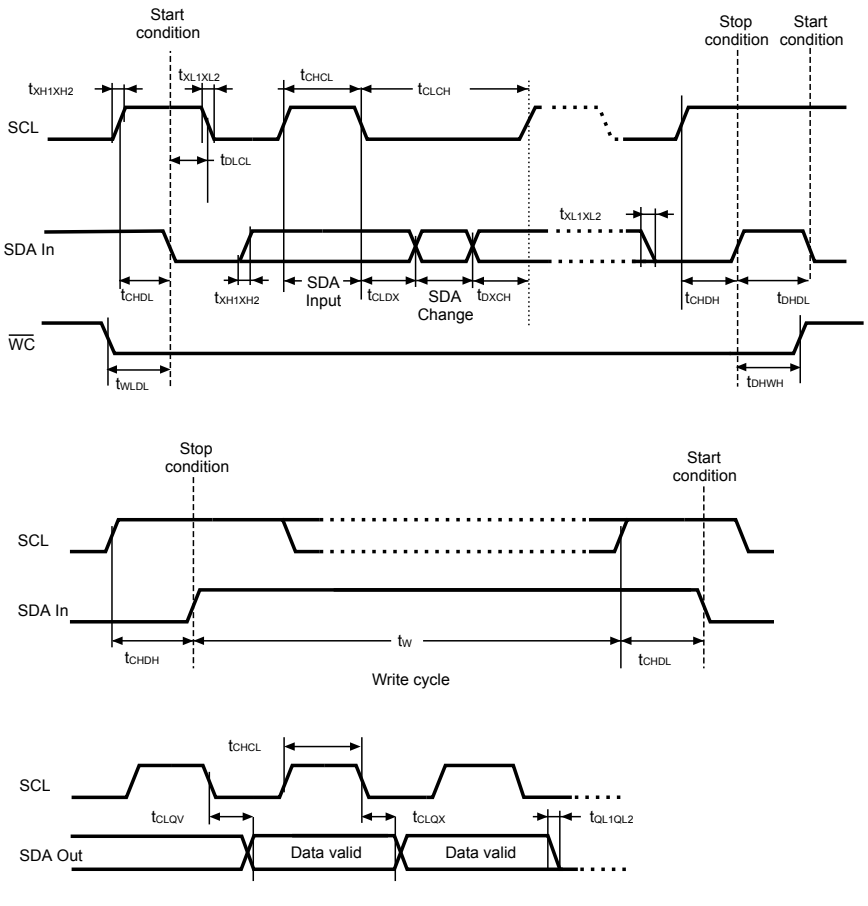



The $R_{bus}$ x $C_{bus}$ time constant must be below the 150 ns time constant line displayed on the left




DT19745V8

Figure 22. **AC waveforms**




DT00795IV1

# 10 Package information

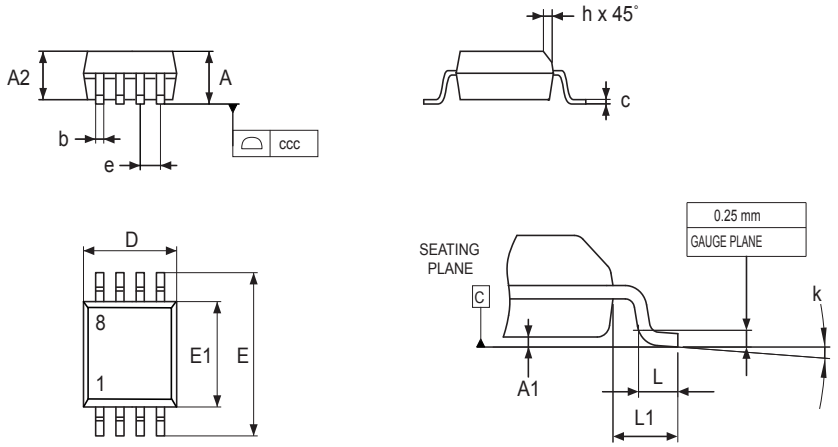
To meet environmental requirements, ST offers these devices in different grades of ECOPACK packages, depending on their level of environmental compliance. ECOPACK specifications, grade definitions, and product status are available at: www.st.com.

ECOPACK is an ST trademark.

## 10.1 SO8N package information

This SO8N is an 8-lead, 4.9 x 6 mm, plastic small outline, 150 mil body width package.

**Figure 23. SO8N - Outline**




1. *Drawing is not to scale.*

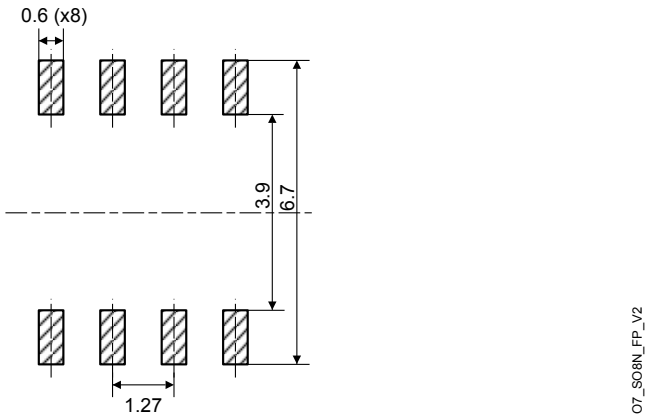
**Table 21. SO8N - Mechanical data**

| Symbol | millimeters | | | inches [1] | | |
|---|---|---|---|---|---|---|
| | Min. | Typ. | Max. | Min. | Typ. | Max. |
| A | - | - | 1.750 | - | - | 0.0689 |
| A1 | 0.100 | - | 0.250 | 0.0039 | - | 0.0098 |
| A2 | 1.250 | - | - | 0.0492 | - | - |
| b | 0.280 | - | 0.480 | 0.0110 | - | 0.0189 |
| c | 0.170 | - | 0.230 | 0.0067 | - | 0.0091 |
| D[2] | 4.800 | 4.900 | 5.000 | 0.1890 | 0.1929 | 0.1969 |
| E | 5.800 | 6.000 | 6.200 | 0.2283 | 0.2362 | 0.2441 |
| E1[3] | 3.800 | 3.900 | 4.000 | 0.1496 | 0.1535 | 0.1575 |
| e | - | 1.270 | - | - | 0.0500 | - |
| h | 0.250 | - | 0.500 | 0.0098 | - | 0.0197 |
| k | 0° | - | 8° | 0° | - | 8° |
| L | 0.400 | - | 1.270 | 0.0157 | - | 0.0500 |
| L1 | - | 1.040 | - | - | 0.0409 | - |
| ccc | - | - | 0.100 | - | - | 0.0039 |

1. *Values in inches are converted from mm and rounded to four decimal digits.*
2. *Dimension D does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed 0.15 mm per side*
3. *Dimension E1 does not include interlead flash or protrusions. Interlead flash or protrusions shall not exceed 0.25 mm per side.*

*Note: The package top may be smaller than the package bottom. Dimensions D and E1 are determinated at the outermost extremes of the plastic body exclusive of mold flash, tie bar burrs, gate burrs, and interleads flash, but including any mismatch between the top and bottom of the plastic body. The measurement side for mold flash, protusions, or gate burrs is the bottom side.*

**Figure 24. SO8N - Footprint example**




1. *Dimensions are expressed in millimeters.*

# 11 Ordering information

**Table 22. Ordering information scheme**

Example: M24 M02E - U F MN 6 T P

**Device type**

M24 = $I^2C$ serial access EEPROM

**Device function**

M02E = 2 Mbit (256 K x 8 bit)

**Device family**

U = With UID

**Operating voltage**

F = $V_{CC}$ = 1.6 V to 5.5 V

**Package**(1)

MN = SO8 (150 mil width)

**Device grade**

6 = Industrial device tested with standard test flow over -40 to 85 °C

**Option**

T = Tape and reel packing

Blank = Tube packing

**Plating technology**

P or G = ECOPACK2

1. *ECOPACK2 (RoHS compliant and free of brominated, chlorinated, and antimony oxide flame retardants).*

*Note: For a list of available options (memory, package, and so on) or for further information on any aspect of this device, contact your nearest ST sales office.*

*Note: Parts marked as “ES” or “E” are not yet qualified and therefore not approved for use in production. ST is not responsible for any consequences resulting from such use. In no event will ST be liable for the customer using any of these engineering samples in production. ST’s Quality department must be contacted prior to any decision to use these engineering samples to run a qualification activity.*

# Revision history

Table 23. **Document revision history**

| Date | Revision | Changes |
|---|---|---|
| 07-May-2025 | 1 | Initial release. |

# Contents

# List of tables

# List of figures

**IMPORTANT NOTICE – READ CAREFULLY**

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.