# AN4421
# Application note

## Migrating from M24LR to M24SR series

## Introduction

Both M24LRxx and M24SRxx dynamic tags can be accessed through the I²C-bus. This application note aims at helping to design an application able to interface through the I²C-bus either the M24LRxx[a] or the M24SRxx[b]. Some references are also made when accessing the standard M24xxx (EEPROM accessed through the I²C-bus).

a. Also accessible through RF (ISO15693 standard)

b. Also accessible through RF (ISO14443 standard)

# Contents

# List of tables

# List of figures

# 1 Hardware and AC/DC considerations

## 1.1 Hardware considerations

Both M24LRxx and M24SRxx devices are offered in an 8-pin package. The hardware differences between the two devices are detailed in*Table 1*.

**Table 1. Pinout differences between M24LRxx and M24SRxx devices**

| | M24LRxx | M24SRxx | Comments |
|---|---|---|---|
| Pin1 | E0 | RF disable | E0 input functionality is different from the RF disable input functionality |
| Pin2 | AC0 | AC0 | Antenna coil |
| Pin3 | AC1 | AC1 | Antenna coil |
| Pin4 | Vss | Vss | Ground |
| Pin5 | SDA | SDA | Serial Data |
| Pin6 | SCL | SCL | Serial Clock |
| Pin7 | E1 | GPO | E1 input functionality is different from the General Purpose Output functionality. |
| Pin8 | Vcc | Vcc | Supply voltage |

Although pin1 and pin7 differ between the M24SRxx and the M24LRxx, the I²C specific (and RF specific) inputs are identical for the M24SRxx and M24LRxx. Both M24LRxx and M24SRxx can therefore be accessed through the I²C-bus, on the same pins, SDA and SCL (pin5 and pin6). The only pin to connect differently is pin7 (input for the M24LRxx and output for the M24SRxx).

## 1.2 AC/DC characteristics

### 1.2.1 AC/DC differences

*Table 2* summarizes the AC/DC differences to consider when designing an application moving from M24LRxx to M24SRxx (or reverse move).

**Table 2. AC/DC differences between M24LRxx and M24SRxx devices**

| | M24LRxx | M24SRxx | Comments |
|---|---|---|---|
| Vcc supply voltage | 1.8 V / 5.5 V | 2.7 V / 5.5 V | |
| Operating temperature range | -40°C / 85°C | | |
| Maximum clock frequency | 400 kHz | 1 MHz | See *Section 1.2.2* |
| Supply current during a Write | 700 µA | 550 µA | Worst case value, at max. SCL frequency, no RF signal on AC0/AC1, Vcc = 5.5 V |
| Supply current during a Read | 500 µA | | |
| Write cycle (byte or page) | 5 ms | | |

### 1.2.2 Pull resistor on SDA line

The I²C-bus specification defines the SDA output as an open drain so that, if two devices are in conflict while outputting a "1" and a "0" at the same time, this conflict cannot induce critical current peaks.

The "0" level is driven by the open drain output, while the "1" level is sourced with an external pull up resistor $R_{bus}$.

This $R_{bus}$ resistor value has to be:

- High enough so that the current flowing through $R_{bus}$ and SDA remains lower than the specified $I_{ol}$ value (a few mA)
- Reasonably low so that the cut-off frequency[(a)] defined with the SDA line parasitic capacitor $C_{bus}$ is higher than the maximum SCL clock frequency.

Depending on the application parasitic capacitor value $R_{bus}$, $C_{bus}$ can be defined as explained in *Figure 1* curves.

**Figure 1. Maximum $R_{bus}$ value versus bus parasitic capacitance ($C_{bus}$) for an I²C bus at maximum frequency $f_C$ = 400 kHz**

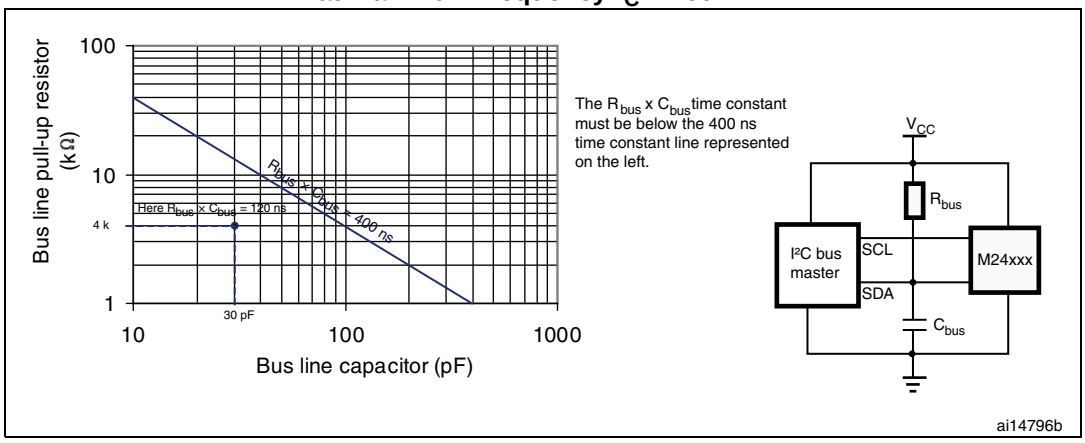
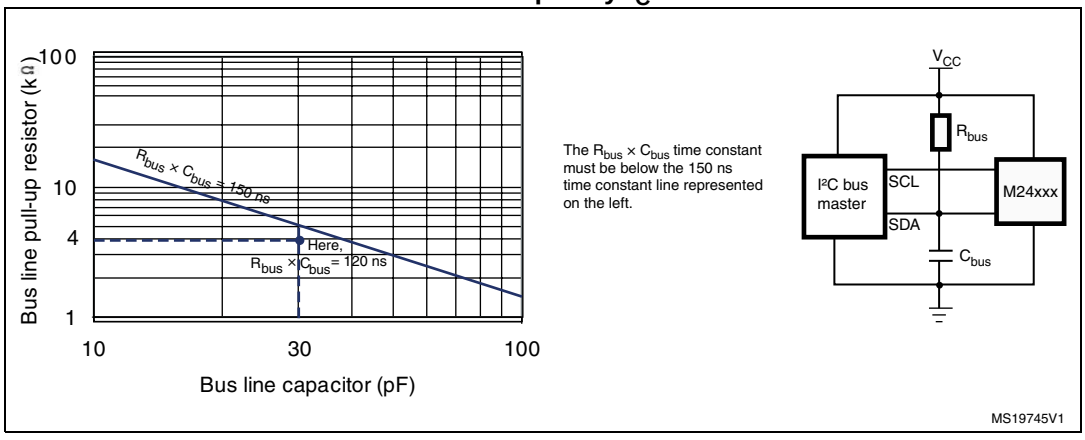



**Figure 2. Maximum $R_{bus}$ value versus bus parasitic capacitance ($C_{bus}$) for an I²C bus at maximum frequency $f_C$ = 1 MHz**




a. The time constant = $R_{bus}$ * $C_{bus}$ defines the cut-off frequency: $f_{cut\text{-}off}$ = ½ * PI * $R_{bus}$ * $C_{bus}$

# 2 Software considerations

## 2.1 I²C-bus and standard EEPROM memories (M24xxx)

The I²C-bus specification defines a data transfer as a string of bytes inside which the first byte handles one $\overline{RW}$ bit defining the transfer direction: from Master to Slave or from Slave to Master.

**Table 3. Device select byte (first byte)**

| Device type identifier(1) | | | | Chip enable address | | | $\overline{RW}$ |
|---|---|---|---|---|---|---|---|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| 1 | 0 | 1 | 0 | E2 | E1 | E0 | $\overline{RW}$ |

1. The most significant bit, b7, is sent first.

M24xxx is always a slave device. To start a communication between the bus master and the slave device, the bus master must first initiate a Start condition. Following this, the bus master sends the Device select byte. The Device select byte consists of a 4-bit device type identifier and up to 3 Chip enable address bits. A Device type identifier handling any value other than 1010b (to select the memory) is not acknowledged by the memory device.

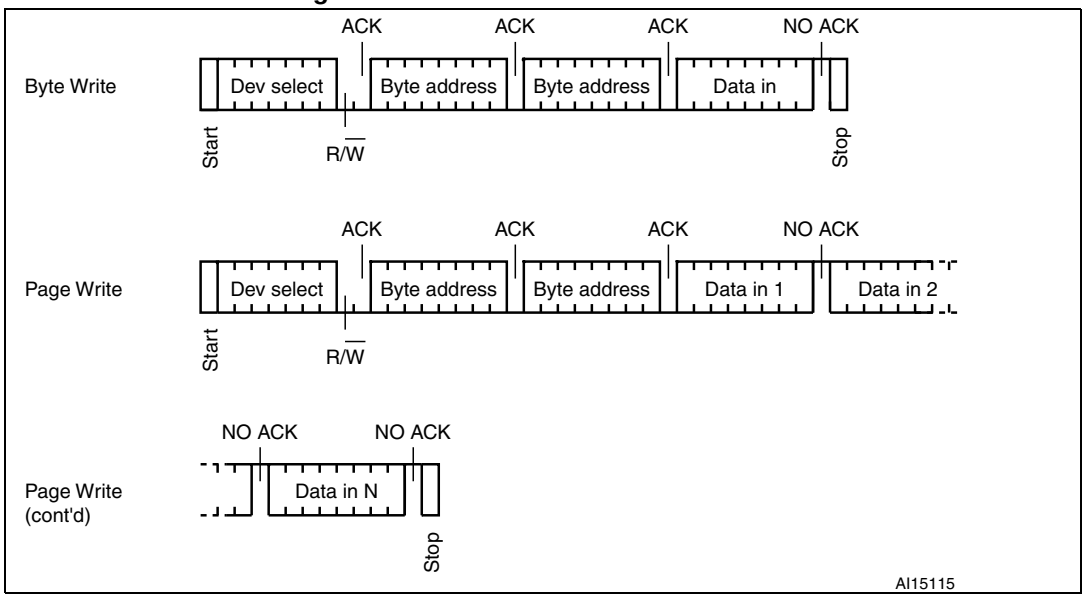
When the device select code is received, the memory device only responds if the Chip enable address is the same as the value decoded on the Ei inputs.

The 8th bit is the Read/Write bit ($\overline{RW}$). This bit is set to 1 for Read (from Slave) and 0 for Write (to Slave) operations.

## 2.2 Differences when accessing a standard EEPROM (M24xxx) or an M24LRxx dynamic tag (ISO15693) through the I²C-bus

The standard EEPROMs are accessed, after a Start condition, with a sequence composed of the Device Select byte (where the $\overline{RW}$ bit defines the data transfer direction) followed by two address bytes and the data bytes. This sequence is ended by the bus Master sending a Stop condition, as in the Write sequence example shown in *Figure 3*.

**Figure 3. Access to a standard EEPROM**




The M24LRxx dynamic tags are accessed with the same Read and Write sequences as the sequences used for the standard EEPROM M24xxx, that is a sequence initiated with a Start condition and composed of the Device Select byte followed by the address bytes and the data bytes. This sequence is ended by the bus Master sending a Stop condition, as shown in *Figure 3*.

The difference between a standard EEPROM M24xxx and the M24LRxx comes from the Device Select byte content:

- The standard EEPROM Device Select byte is defined in *Table 4*.

**Table 4. Standard EEPROM Device Select byte**

| Device type identifier(1) | | | | Chip enable address(2) | | | $\overline{RW}$ |
|---|---|---|---|---|---|---|---|
| **b7** | **b6** | **b5** | **b4** | **b3** | **b2** | **b1** | **b0** |
| 1 | 0 | 1 | 0 | E2 | E1 | E0 | $\overline{RW}$ |

1. The most significant bit, b7, is sent first.
2. E0 bit and E1 bit are compared against the respective values read from external pins E0, E1 of the memory device.

- The M24LRxx Device Select byte is defined in *Table 5*.

**Table 5. M24LRxx Device Select byte**

| Device type identifier(1) | | | | Chip enable address(2) | | | $\overline{RW}$ |
|---|---|---|---|---|---|---|---|
| **b7** | **b6** | **b5** | **b4** | **b3** | **b2** | **b1** | **b0** |
| 1 | 0 | 1 | 0 | E2(3) | E1 | E0 | $\overline{RW}$ |

1. The most significant bit, b7, is sent first.
2. E0 bit and E1 bit are compared against the respective values read from external pins E0, E1 of the memory device.
3. E2 is not connected to any external pin, see M24LRxx for details.

The reader can see that bit E2 makes the difference between the two products; this has to be taken into account when developing the software routine accessing either a standard EEPROM or an M24LRxx dynamic tag.

## 2.3 Differences when accessing an ISO15693 dynamic tag (M24LRxx) or an ISO/IEC 1443 dynamic tag (M24SRxx) through the I²C-bus

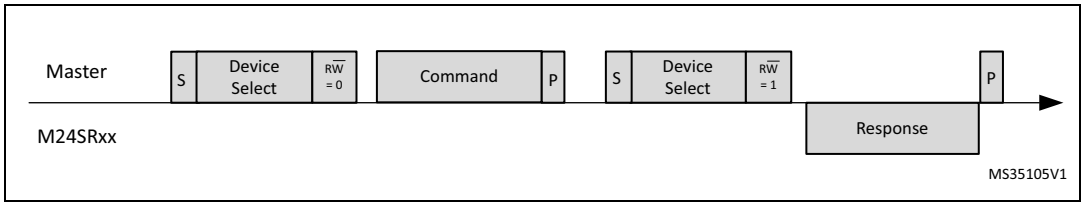
The M24SRxx dynamic tags are accessed through the I²C-bus protocol where each byte is acknowledged by the device receiving the transmitted byte, but the sequence content is different from the M24LRxx sequence.

For the M24SRxx, the I²C communication is built on a system of command and reply exchange:

1. The I²C-bus Master starts the communication by sending a request, that is a Device Select byte with the $\overline{RW}$ bit set to 0 followed by the command field.
2. Once a valid request is received by the M24SRxx, it prepares its answer.
3. Then the I²C-bus Master sends a response request, that is a Device Select byte with the $\overline{RW}$ bit set to 1, followed by the data received from the M24SRxx.

*Figure 4* shows a simplified overview of the (command+response) sequence between the I²C-bus Master and the M24SRxx.

**Figure 4. Command+response sequence between the I²C-bus Master and the M24SRxx**




1. S is the I²C Start condition.
2. $\overline{RW}$ is the 8th bit of Device Select.
3. P is the I²C Stop condition.

*Note: The M24SRxx command response details are offered in the AN4433 (Storing data into the NDEF memory of M24SR).*

# 3 Revision history

Table 6. Document revision history

| Date | Revision | Changes |
|---|---|---|
| 17-Mar-2014 | 1 | Initial release. |