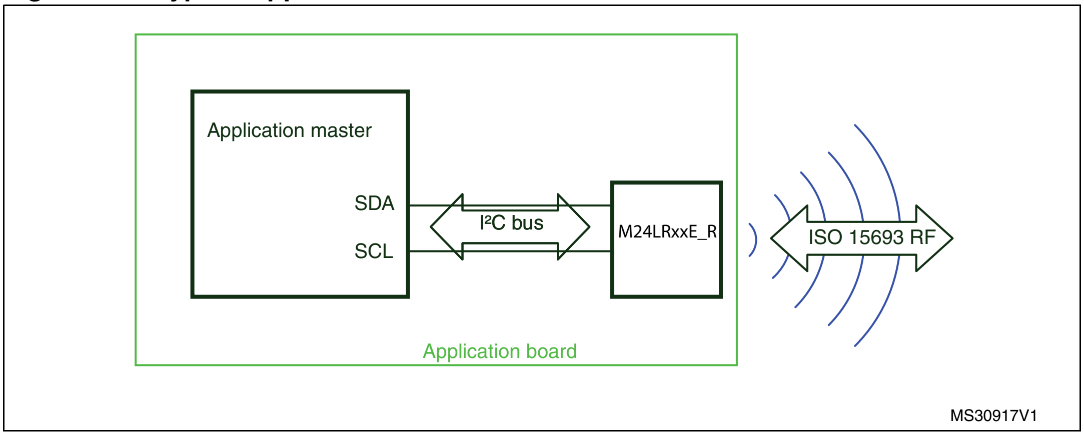
AN4125
Application note

How to manage simultaneous I²C and RF data transfers with the M24LRxxE-R

Introduction

The M24LRxxE_R is an EEPROM device designed to be accessed via two different interfaces: a wired I²C interface and a standard contactless ISO 15693 RFID interface.

Figure 1. Typical application of an M24LRxxE_R dual interface EEPROM



ST has published various supporting application notes explaining how the RF interface works and the basic principles of passive RFID technology. These documents are available from: *www.st.com/dualeeprom*.

The possibility of using two different interfaces to control the dual-interface EEPROM implies two host controllers: a microcontroller with an I²C bus and an ISO 15693 RFID reader. Due to their nature, these two host controllers are not synchronized, which means that both controllers might try to access the M24LRxxE_R concurrently.

To manage this kind of situation, the M24LRxxE_R has a built-in circuitry able to handle possible concurrent communications and powering activities from the RF and I²C sides.

This application note describes how the M24LRxxE_R arbitration circuitry operates. It applies to the products listed in *Table 1*.

Table 1. Applicable products

Type	Part numbers
Memory products	M24LR04E-R, M24LR16E-R, M24LR64E-R

Contents

List of tables

List of figures

1 RF - I²C arbitration mechanism description

The M24LRxxE_R arbitration circuitry is twofold. It contains:

- a power management unit that handles the power coming potentially from the RF or the I²C side
- a communication arbitration unit that tackles potential concurrent communications from the RF and the I²C sides

1.1 Communications and power supply conditions

The power supply management unit has been designed to allow for flexibility, especially when both the RF power and the wired power line are active at the same time.

The basic principle is:

- When supplied only from the RF side:
  - the M24LRxxE_R can be accessed only by the RF reader
- When supplied from both the V_{CC} pin and the RF field:
  - the M24LRxxE_R will serve the first decoded command (either RF or I²C) and will not decode any command from the other interface (either I²C or RF) until the first decoded command is complete.

Table 2. Four possible combinations of power supply sources

Possible cases	V_{CC}	RF field	Actions
Case 1	0 V or not connected	Off	The M24LRxxE_R is reset.
Case 2	0 V or not connected	**On**	RF data transfers: yes I²C data transfers: no
Case 3	**On**(1)	**On**	RF data transfers: yes I²C data transfers: yes (see *Section 1.2: Communication arbitration when the RF and I²C channels are both active* for details).
Case 4	**On**(1)	Off	RF data transfers: no I²C data transfers: yes

1. V_{CC} is “On” when the value is between V_{CC}min and V_{CC}max. Please refer to the M24LRxxE_R datasheet for full details.

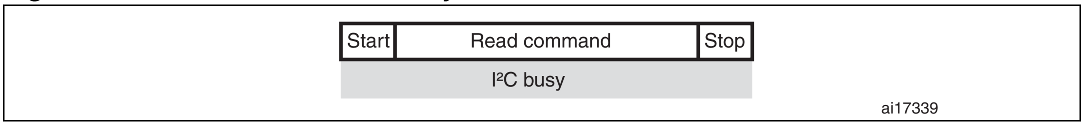
1.2 Communication arbitration when the RF and I^{2}C channels are both active

Arbitration depends on whether the I^{2}C and RF channels are in the busy state. *Section 1.2.1* and *Section 1.2.2* give the definitions of the I^{2}C and RF busy states, respectively.

1.2.1 I^{2}C busy states

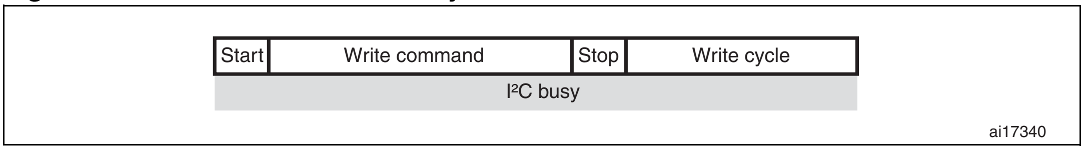
When decoding an I^{2}C read command, the M24LRxxE_R is in the I^{2}C busy state from the Start condition until the Stop condition.

Figure 2. I^{2}C read command busy state



When decoding an I^{2}C write command, the M24LRxxE_R is in the I^{2}C busy state from the Start condition until the completion of the write cycle (triggered by the Stop condition).

Figure 3. I^{2}C write command busy state



1.2.2 RF busy states

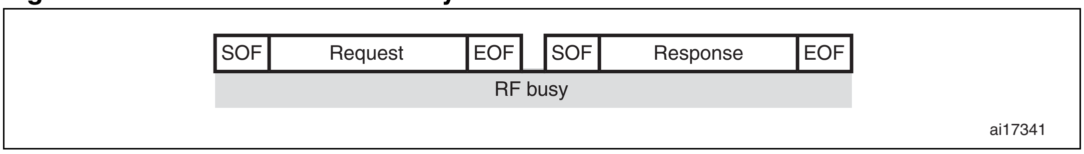
In most cases, an RF command is defined as a received request initiated by the SOF (start of frame) and terminated by the decoding of the EOF (end of frame) of the response frame.

RF commands can be gathered into several groups:

Read command group

When decoding an RF read command, the M24LRxxE_R is in the RF busy state from the SOF (start of frame) of the request frame until the EOF (end of frame) of the response frame. The figure below shows the RF busy state of commands in the read group.

Figure 4. RF read command busy state



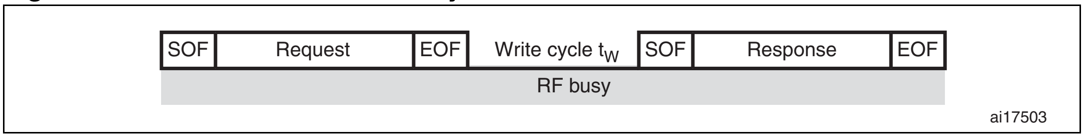
Commands in the RF read command group are:

- Read Block, Fast Read Single Block, Read Multiple Blocks, Fast Read Multiple Blocks
- Get System Info
- Select
- Reset to Ready
- Get Multiple Block Security Status
- Initiate, Fast Initiate
- Inventory Initiated

Write command group

When decoding an RF write command, the M24LRxxE_R is in the RF busy state from the SOF (start of frame) of the request frame until the EOF (end of frame) of the response frame. Write commands include a write cycle t_W. The figure below shows the RF busy state of commands in the write group.

Figure 5. RF write command busy state



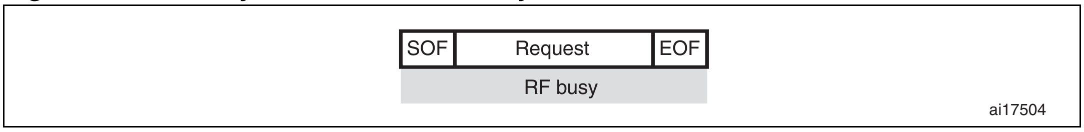
Commands in the RF write command group are:

- Write Block
- Write AFI, Lock AFI
- Write DSFID, Lock DSFID
- Write-sector Password, Lock-sector Password, Present-sector Password

Stay Quiet command

The Stay Quiet command is the only command defined as a single request frame (not followed by a response frame). The M24LRxxE_R is in the RF busy state during the whole [SOF …. EOF] sequence as shown in the figure below.

Figure 6. RF Stay Quiet command busy state



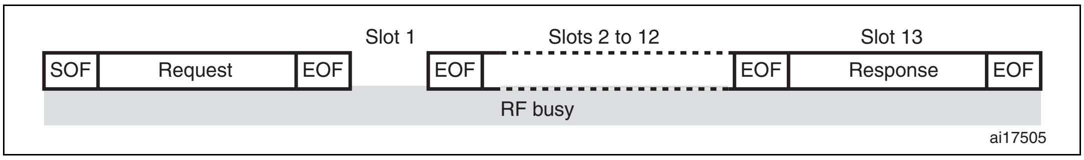
Inventory command

An Inventory command is used when several M24LRxxE_R devices are inside the range of the same RF electromagnetic field.

When the Inventory command scans 16 slots, the M24LRxxE_R is in the RF busy state from the SOF (start of frame) of the request frame until the EOF (end of frame) of the response frame.

Note: The addressed M24LRxxE_R device might stay a long time in the RF busy state if it is decoded during the last (16^{th}) time slot.

Figure 7. Example of an Inventory command where the M24LRxxE_R is decoded in Slot 13



1.2.3 Arbitration

When both interfaces are active (as defined in Case 3 in *Table 2: Four possible combinations of power supply sources*1), the M24LRxxE_R decodes and executes the first received command, as detailed in *Table 3: Possible cases of communication arbitration*.

Table 3. Possible cases of communication arbitration

Initial state	Event	M24LRxxE_R action
M24LRxxE_R is in the I²C busy state: V_{CC} active and an I²C command is being decoded or executed	RF command transmitted during an I²C command	RF command is not decoded
M24LRxxE_R is in the RF busy state: an RF command is being decoded or executed	V_{CC} active and I²C command transmitted during an RF command	I²C command is not decoded

2 Recommendations when developing the M24LRxxE_R application software

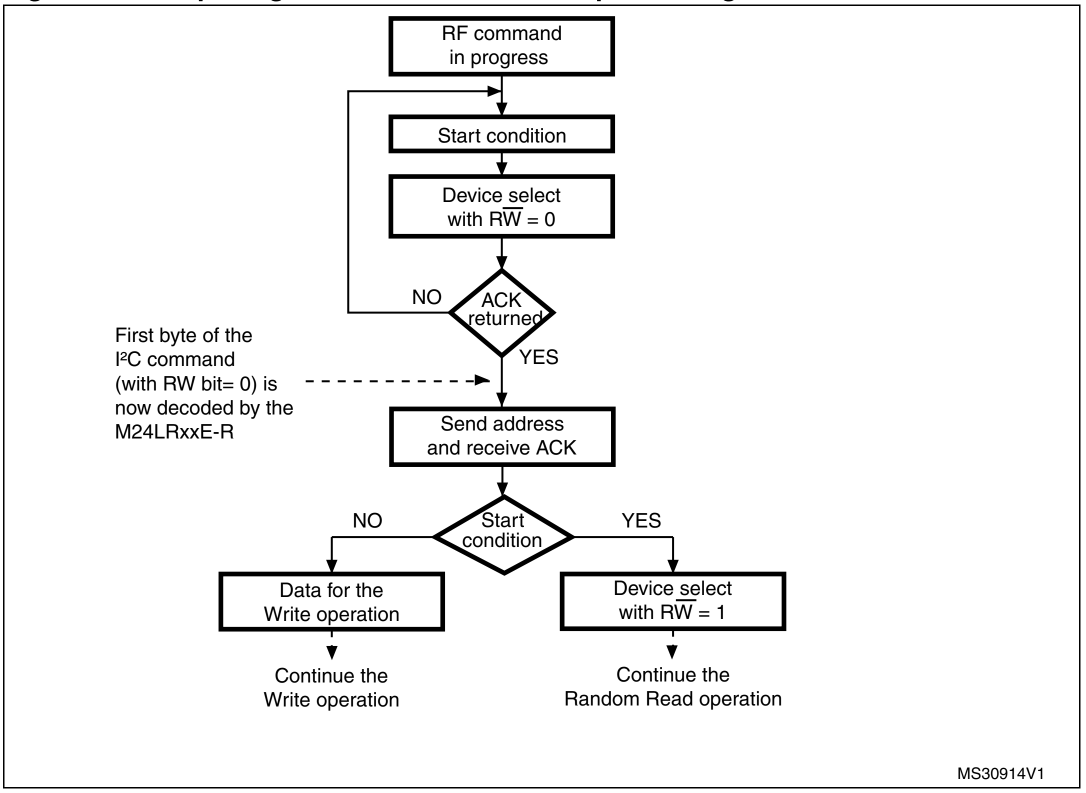
The application software has to take into account that a command might not be executed if the other channel (I²C or RF) is already processing a command. The application software should therefore check the M24LRxxE_R busy status before sending a command.

2.1 Issuing a command through the I²C channel

2.1.1 I²C request while the RF channel is busy

If the M24LRxxE_R is processing a command from the RF channel, no command issued on the I²C bus will be executed, therefore none of the bytes transmitted on the I²C bus will be acknowledged (NoAck). This information can be considered as the RF busy state[(a)] and the application's I²C software should include a polling loop on the RF busy state (with a timeout limit) when issuing a command on the I²C bus. In this way, the I²C command can be completed once the RF command under process has completed.

Figure 8. I²C polling when the RF channel is processing a command



a. In the same way as during an internal write cycle, the M24LRxxE_R is "busy" during t_W (please refer to the M24LRxxE_R datasheet for more details about the polling loop during t_W).

Important

It is paramount to exactly carry out the I²C polling sequence described in *Figure 8* in order to keep the M24LRxxE_R in a constant I²C busy state.

- **Right method:** once the device select is acknowledged, the I²C command starts executing until full completion, that is, until the transmission of the Stop condition which ends the command (or at the end of the write cycle t_W, for a write command).
- **Wrong method:** looping on the device select until it is acknowledged, sending a Stop condition and then initiating a new I²C command: this is inadequate as an RF request might have been served between [Ack] and the new I²C command (time slot during which the M24LRxxE_R is not in the I²C busy state.

Note: If the application is disturbed by too great a number of decoded RF commands, it might be convenient that the I²C bus master prompts the application to stop RF requests so that the I²C bus can access the M24LRxxE_R.

2.1.2 I²C requests and RF time slots

Application software management

In most cases, the application fully controls the I²C bus. On the other hand, it cannot always predict RF commands. To have a robust application, the M24LRxxE_R should be fullly controlled through the I²C bus, that is, the application master has to:

1. determine when the I²C commands have to be transmitted
2. determine the time slots during which RF transfers may be processed

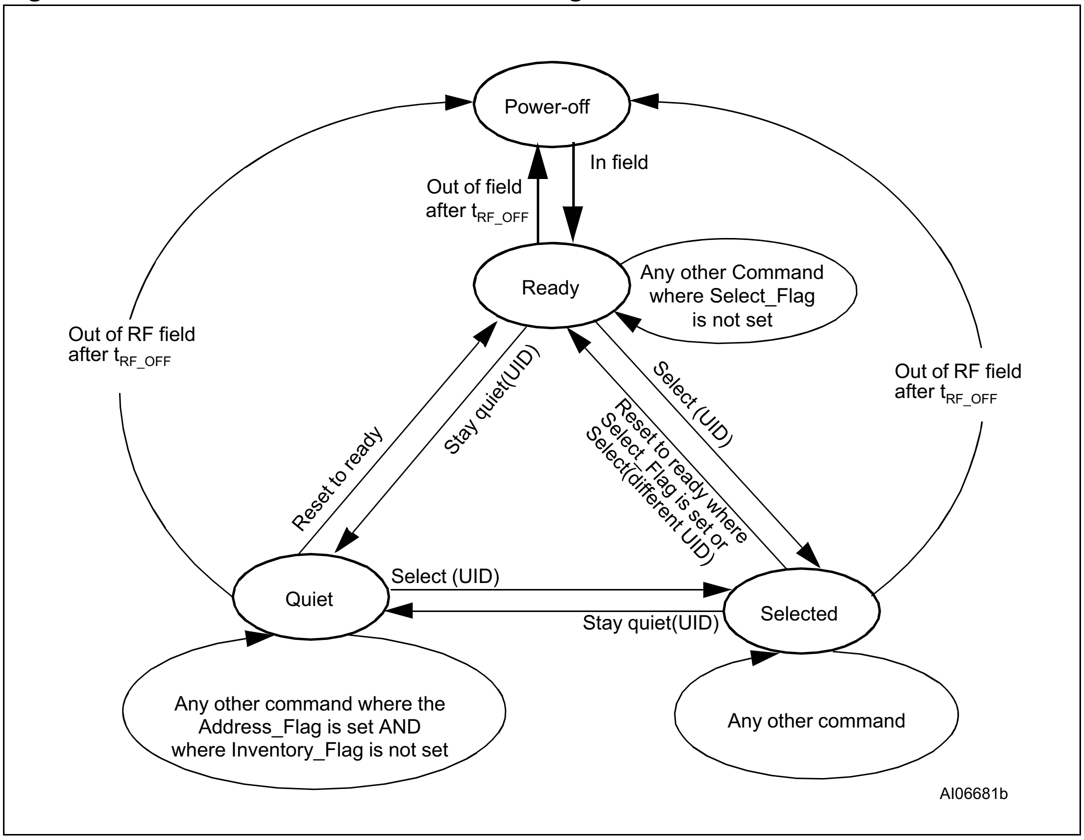
The reason for this is that RF commands might not be properly transmitted (for example, if the M24LRxxE_R leaves the RF field). The I²C bus Master has to prevent this from happening by applying the following rules:

- The Master determines when the I²C commands have to be transmitted

  The Master delivers the supply voltage (through one of its I/Os) to the M24LRxxE_R's V_{CC} pin only when an I²C data transfer is under way
- The Master determines the RF time slots

  The Master stops supplying (I/O in HiZ) the M24LRxxE_R through its V_{CC} pin upon completion of the I²C data transfer. RF transfers are processed more safely when the V_{CC} pin is not supplied, because:
  - If the decoded RF command is correct, it is executed (no need to supply power through the V_{CC} pin)
  - If the RF command is truncated (M24LRxxE_R is outside the RF field), the M24LRxxE_R is reset (Power-off state, see *Figure 9*).

Figure 9. M24LRxxE_R state transition diagram

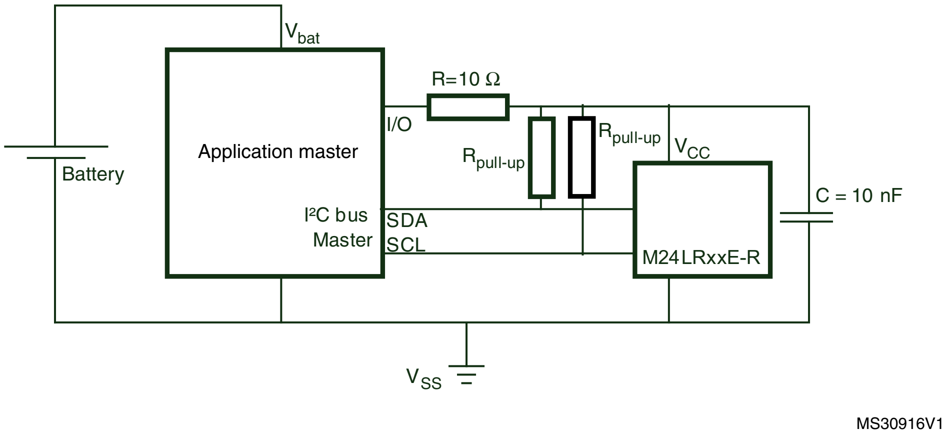


1. The M24LRxxE_R returns to the Power Off state if the tag is out of the RF field for at least t_{RF_OFF}.

Application hardware architecture

The application Master should control the I²C bus lines and the power supply line so as to keep full control of the M24LRxxE_R. *Figure 10* shows a typical hardware schematic.

Figure 10. Optimal hardware schematic of an M24LRxxE_R application



This type of hardware architecture is optimal in applications where power saving is a key feature (like portable applications supplied from a battery). The supply voltage can be directly delivered to an ultralow power microcontroller (for instance the STM8L, information available from *http://www.st.com/mcu/*). This implementation makes it possible to keep the application supply current in the 1 µA range (value of the STM8L supply current when in the Active-Alt mode) and:

- the application saves power when in the Standby mode, as the battery does not supply the standby current to the M24LRxxE_R (40 µA) nor the current through the SDA pull-up resistor
- the application controls the M24LRxxE_R in a safe mode (the V_{CC} pin is supplied by the Master only when an I²C request is being processed)

2.1.3 An I²C request was interrupted

A Start condition defines the I²C channel as busy until the completion of the I²C command (Stop condition) or until I²C timeout. If for some uncontrolled reason, inadvertent unterminated instructions are sent to the I²C bus, the M24LRxxE-R features a timeout mechanism that automatically resets the I²C logic block.

The I²C busy state is reset either:

- by decoding a device select byte different from 1010 XXXXb,
- by decoding a Stop condition,
- by the completion of the internal write cycle (t_W, triggered by a decoded write instruction), or
- after timeout.

The best way for the I²C bus Master to clear a spurious busy state is to periodically issue a [Start+Stop] sequence.

Note: *In noisy applications, ST recommends to implement the "9 Start + 1 Stop" sequence described in AN1471 (available from the ST website: www.st.com).*

2.2 Issuing a command through the RF channel

Case 1: the M24LRxxE_R is processing an I²C command

If the M24LRxxE_R is processing a command from the I²C channel, no command issued on the RF channel will be executed (the RF command will not provide any response) when the M24LRxxE_R is I²C busy.

The application's RF software should include an "I²C busy polling loop" (including a timeout as there might not be a response) when issuing an RF command. In this way, the RF command is always correctly executed once the I²C commands under execution are completed.

Case 2: the M24LRxxE_R application is powered on

The first condition for a safe design of an M24LRxxE_R application is that all the sensitive data stored in the M24LRxxE_R memory are protected with RF passwords, so that a spurious RF command could not modify these data.

The second condition for a safe application design is that the application Master fully controls the M24LRxxE_R. We know that, as explained in *Section 2.1.2*, if the V_{CC} pin is not supplied, and the RF field drops to zero while the M24LRxxE_R is decoding an RF command, then the M24LRxxE_R is reset. This means that the Master has to supply the M24LRxxE_R's V_{CC} pin only during an I²C transfer, and leave the V_{CC} pin floating the rest of the time (see *Figure 10*).

Depending on the application's RF data transfer flow, it might also be wise to add a third level of safety:

- once an RF session (several commands) is completed, blindly send a Write-sector Password command with a wrong password value: this will set the internal flag defining the status of the presented password (flag set as "wrong password presented").

Case 3: the M24LRxxE_R application is not powered

This is the typical case where the application is packed in a box (at the end of the production line) and the data update is performed through RF.

The only condition for a safe application design is that the sensitive data stored in the M24LRxxE_R memory are protected with RF passwords, so that a spurious RF command could not modify these data.

Table 4. M24LRxxE_R status according to command and V_{CC} supply

Command processed by the M24LRxxE_R	V_{CC} pin status	Device status
I²C command	V_{CC} supplied	I²C busy state
RF command	V_{CC} supplied or V_{CC} = high impedance	The M24LRxxE_R is fully dedicated to RF commands

3 Revision history

Table 5. Document revision history

Date	Revision	Changes
15-Jun-2012	1	Initial release.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com