AN3355
Application note

Designing an application board compatible with either standard M24LRxx-R or energy-harvesting M24LRxxE-R devices

Introduction

The goal of this application note is to describe how to design application boards accepting either standard M24LRxx-R or energy-harvesting M24LRxxE-R devices.

Contents

List of tables

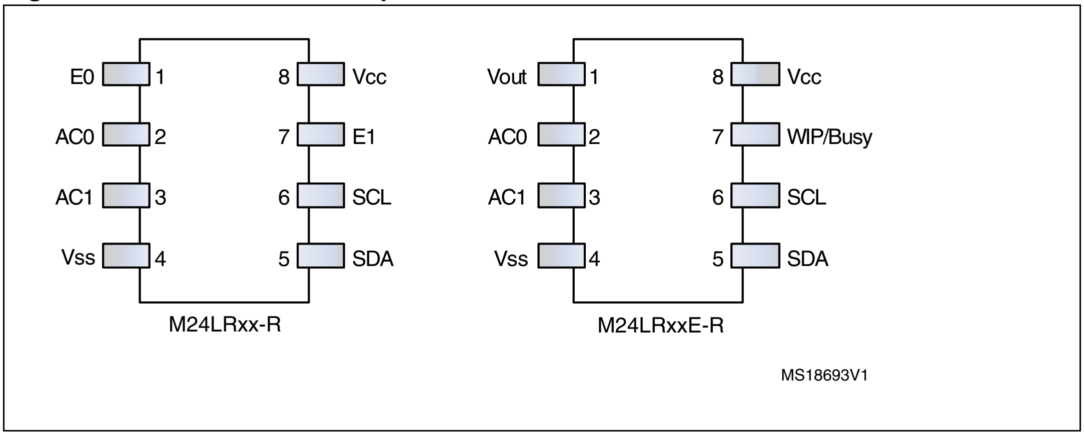
List of figures

1 M24LRxx-R and M24LRxxE-R pin-out comparison

The standard dual-interface M24LRxx-R and the energy-harvesting M24LRxxE-R devices have small differences in pin-out, which apply to only two pins:

- M24LRxx-R
  - Pin1: E0 input (used to define the hardware address of the I²C-bus slave)
  - Pin7: E1 input (used to define the hardware address of the I²C-bus slave)
- M24LRxxE-R
  - Pin1: Vout (output used to deliver supply voltage to the application)
  - Pin7: WIP/Busy output signal (during a Write cycle)

Figure 1. Two families, two pin-outs



When designing a new application, it can be useful to design an application board which can embed either the standard M24LRxx-R or the energy-harvesting M24LRxxE-R devices.

2 Designing a board compatible with both devices

The purpose of this application note is to design an application board that accepts either the standard M24LRxx-R (dual-interface) or the M24LRxxE-R (energy-harvesting) devices.

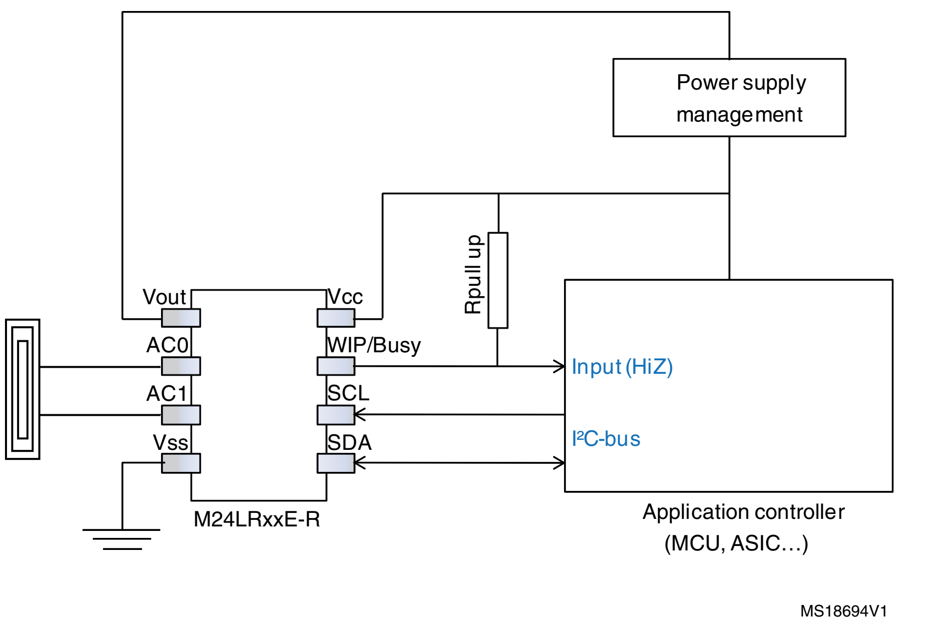
Let's see how pin1 and pin7 are used in a typical application based on M24LRxxE-R.

- Pin1: M24LRxxE-R Vout pin, providing a power supply to the application.

  This pin is actually connected to the input of the application power supply management block.
- Pin7: M24LRxxE-R WIP/Busy output.

  This pin is connected to one input of the application controller.

Figure 2. Typical schematic of an application powered from M24LRxxE-R



If we substitute the M24LRxxE-R (offering the Energy harvesting function) with an M24LRxx-R (without energy-harvesting output), we have to check that the two pins 1 and 7 do not induce malfunction or stress in the application.

Conditions applying to pin1:

The M24LRxx-R pin1 is the E0 input. Its logical value is internally read as:

- either E0=1 when directly connected to V_{CC},
- or E0=0 when directly connected to V_{SS},
- or E0=0 when left unconnected (HiZ).

Comparing this with the M24LRxxE-R requirements, we can see that:

- If pin1 is directly connected to V_{CC}:
  - OK for the M24LRxx-R,
  - not compatible with the M24LRxxE-R: this would short-circuit Vout to V_{CC}.
- If pin1 is directly connected to V_{SS}:
  - OK for the M24LRxx-R,
  - not compatible with the M24LRxxE-R: this would short-circuit Vout to V_{SS}.
- If pin1 is left unconnected (HiZ):
  - OK for the M24LRxx-R.
  - OK for the M24LRxxE-R.

Table 1. Compatibility overview (Pin1)

Pin1	M24LRxxx-R	M24LRxxE-R	Result
Connected to V_{cc}	E0=1	Conflict on Vout	Not possible
Connected to V_{ss}	E0=0	Conflict on Vout	Not possible
Left unconnected (or connected to a high impedance)	E0=0	OK	OK

We can see that the correct way to design an application accepting either M24LRxx-R or M24LRxxE-R is to define E0=0 for the M24LRxx-R.

Conditions applying to pin7:

The M24LRxx-R pin7 is the E1 input. Its logical value is internally read as:

- either E1=1 when directly connected to V_{CC},
- or E1=0 when directly connected to V_{SS},
- or E1=0 when left unconnected (HiZ).

Comparing this with the M24LRxxE-R requirements, we can see that, consequently:

- If pin7 is directly connected to V_{CC}:
  - OK for the M24LRxx-R,
  - not compatible with the M24LRxxE-R: this would create a short-circuit when WIP/Busy outputs a 0.
- If pin7 is directly connected to V_{SS}:
  - OK for the M24LRxx-R,
  - not compatible with the M24LRxxE-R: this would force WIP/Busy signal to V_{SS} and therefore inhibit the WIP/Busy information.
- If pin7 is left unconnected (HiZ):
  - OK for the M24LRxx-R,
  - OK for the M24LRxxE-R.

Table 2. Compatibility overview (Pin7)

Pin7	M24LRxxx-R	M24LRxxE-R	Result
Connected to V_{CC}	E1=1	Conflict on WIP/Busy	Not possible
Connected to V_{SS}	E1=0	WIP/Busy = 0 (always)	WIP/Busy info inhibited
Left unconnected (or connected to a high impedance)	E1=0	OK	OK

One can see that the correct way to design an application accepting either M24LRxx-R or M24xxxE-R is to define E1=0.

3 Summary

When designing an application board that will accept either M24LRxx-R or M24LRxxE-R, the following rules should be kept in mind:

M24LRxx-R

The Device Select byte (first byte transmitted by the I²C-bus master selecting the device to be accessed) is defined as shown in *Table 3*.

Table 3. M24LRxx-R Device Select byte

	Device type identifier(1)				Chip Enable address(2)			$R\overline{W}$
	b7	b6	b5	b4	b3	b2	b1	b0
Device select code	1	0	1	0	E2(3)	E1	E0	$R\overline{W}$

1. The most significant bit, b7, is sent first.
2. E0 and E1 are compared against the respective external pins on the memory device.
3. E2 selects either the memory of the system sector.

The inputs E0,E1 have to be defined as 0,0 (which is the case when the E0,E1 pins are left unconnected). The Device Select byte is therefore:

- either 1.0.1.0. 0.0.0.RW (where RW is the Read/Write bit) when addressing data in the memory array,
- or 1.0.1.0. 1.0.0.RW (where RW is the Read/Write bit) when addressing the system parameter sector.

M24LRxxE-R

In this case of PCB implementation, the E0 and E1 inputs are connected to the power supply management block and the WIP/Busy output is connected to a high impedance input, as shown in *Figure 2*.

The Device Select byte (first byte transmitted by the I²C-bus master) defines the device to be selected. The Device Select byte is defined as shown in the following table.

Table 4. M24LRxxE-R Device Select byte

	Device type identifier(1)				Chip Enable address			$R\overline{W}$
	b7	b6	b5	b4	b3	b2	b1	b0
Device select code	1	0	1	0	E2(2)	1	1	$R\overline{W}$

1. The most significant bit, b7, is sent first.
2. E2 selects either the memory of the system sector.

The Device Select byte is therefore:

- either 1.0.1.0. 0.1.1.RW (where RW is the Read/Write bit) when addressing data in the memory array,
- or 1.0.1.0. 1.1.1.RW (where RW is the Read/Write bit) when addressing the system parameter sector.

4 Revision history

Table 5. Document revision history

Date	Revision	Changes
29-Mar-2011	1	Initial release.
24-May-2011	2	Updated *Section 3: Summary*.