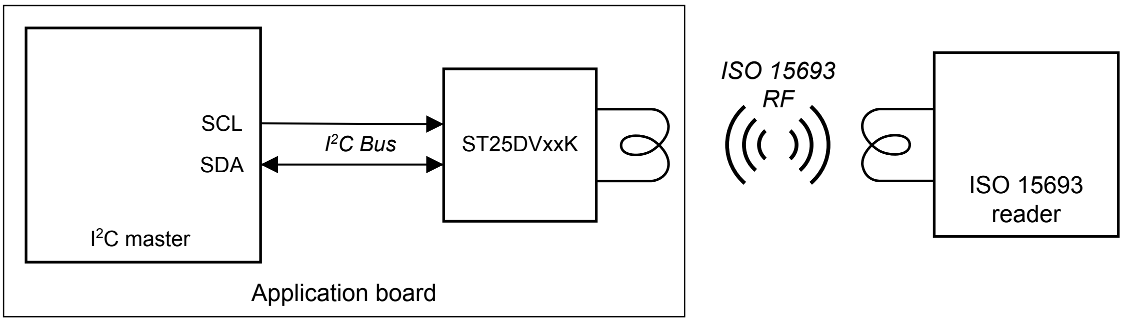
How to manage simultaneous I²C and RF data transfer with an ST25DVxxK device

Introduction

The ST25DV04K, ST25DV16K, or ST25DV64K (hereinafter referred to as ST25DVxxK) is a dynamic tag device designed to be accessed through two different interfaces: a wired I²C interface and a standard contactless ISO 15693 RFID interface.

Figure 1. **Typical application of an ST25DVxxK dual interface EEPROM**



The possibility of using two different interfaces to control the dual-interface dynamic tag 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 ST25DVxxK concurrently. To manage this kind of situation, the ST25DVxxK has built-in arbitration circuitry to handle possible concurrent communications and powering activities from the RF and I²C sides.

This application note describes the operation of the ST25DVxxK arbitration circuitry.

Table 1. **Applicable products**

Product type	Part numbers
Dynamic NFC tags	ST25DV04K ST25DV16K ST25DV64K

1 RF - I²C arbitration mechanism description

The ST25DVxxK arbitration circuitry is twofold. It contains:

- a power management unit that handles the power coming potentially from the RF or the I^2C side
- a communication arbitration unit that tackles potential concurrent communications from the RF and the I^2C 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 ST25DVxxK can be accessed only by the RF reader
- When supplied from both the VCC pin and the RF field:
  - the ST25DVxxK serves the first decoded command (either RF or I²C) and it doesn't 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 ST25DVxxK is reset.
Case 2	0 V or not connected	**On**	RF data transfers: yes I^2C data transfers: no
Case 3	**On** (1)	**On**	RF data transfers: yes I^2C 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 I^2C data transfers: yes

1. *V_{CC} is "On" when the value is between V_{CCmin} and V_{CCmax}x. Please refer to the ST25DVxxK datasheet for full details.*

1.2 Communication arbitration when the RF and I²C channels are both active

Arbitration depends on whether the I²C and RF channels are in the busy state. Section 1.2.1 and Section 1.2.2 give the definitions of the I²C and RF busy states, respectively.

1.2.1 I²C busy states

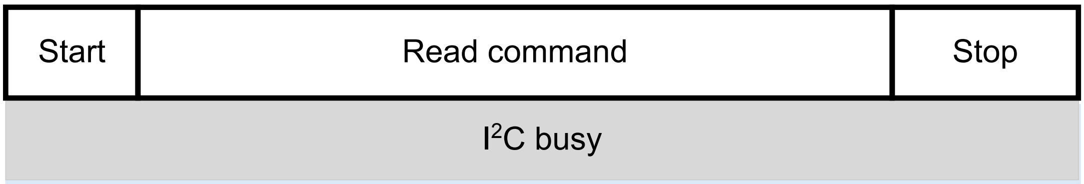
In most cases, an I²C command is initiated by a start condition and terminated by a stop condition. The ST25DVxxK is in I²C busy state when decoding and executing an I²C command.

I²C commands can be gathered into several groups.

Read command group:

When decoding an I²C read command, the ST25DVxxK is in the I²C busy state from the Start condition until the Stop condition.

Figure 2. **I²C read command busy state**



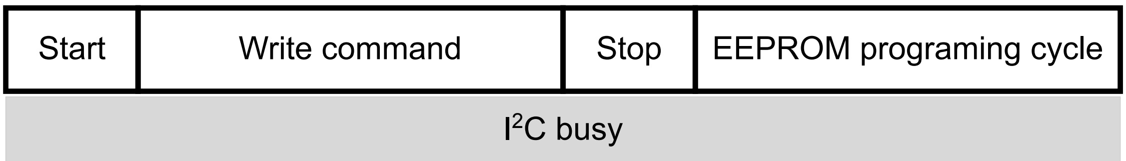
Write command group:

When decoding an I²C write command in EEPROM (user memory or static configuration area), the ST25DV-I2C is in the I²C busy state from the Start condition until the completion of the write cycle (triggered by the Stop condition).

During EEPROM programming cycle, neither I²C or RF communication are possible. I²C commands are not acknowledged and RF commands are either answered with error 0x0F or not answered.

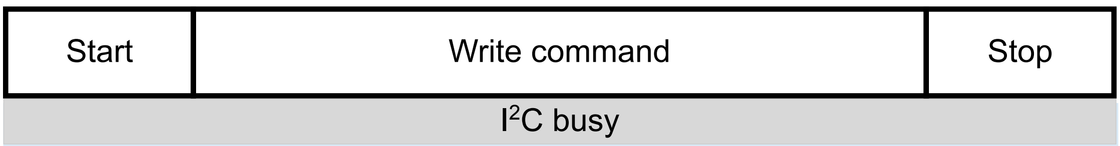
The methods to obtain a precise timing of the EEPROM programming cycle are described in Section 2.1.2 I²C request following an I²C write in EEPROM memory.

Figure 3. **I²C write cycle command busy state when writing in user memory or in system configuration area**



When decoding an I²C write command in the Dynamic configuration registers or in the Fast Transfer Mode mailbox, the ST25DVxxK is in the I²C busy state from the Start condition until the Stop condition (no EEPROM programming cycle).

Figure 4. **I²C write cycle command busy state when writing in dynamic registers or mailbox**

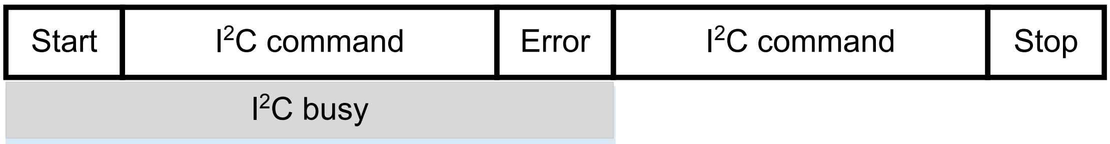


All commands

For any command, the ST25DVxxK exits the I²C busy state before the Stop condition if any I²C error occurs. The possible errors are:

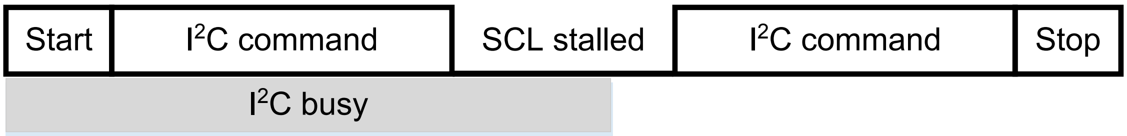
- Wrong device select
- Invalid memory address
- Read or Write not allowed
- I²C master "No Ack" condition
- More than 256 bytes of data in a sequential write command .

Figure 5. I²C busy state for any command when an I²C error occurs



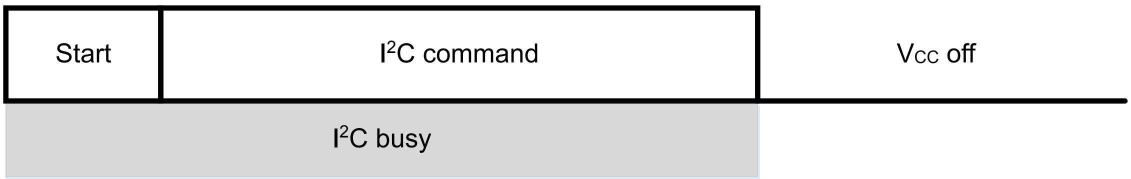
For any command, the ST25DVxxK exits the I²C busy state in case of timeout on SCL clock (see t_{CHCL} and t_{CLCH} in the ST25DVxxK datasheet)

Figure 6. I²C busy state for any command when SCL timeout occurs



For any command, the ST25DVxxK exits the I²C busy state if V_{CC} goes below V_{CCmin}

Figure 7. I²C busy state for any command when V_{CC} goes below V_{CCmin}



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.

The ST25DV-I2C is in RF busy state when decoding and executing an RF command. If an error occurs during the RF command (CRC error, framing error) or if the ST25DV-I2C goes out of the RF field, the ST25DV-I2C exits the RF busy state.

RF commands can be gathered into several groups:

Read command group

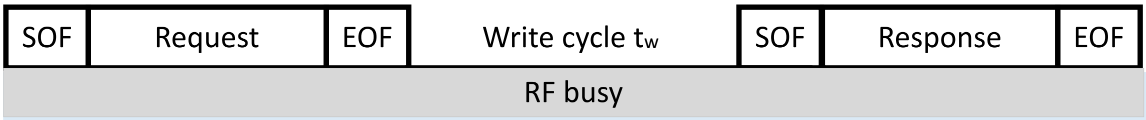
When decoding an RF read command, the ST25DV-I2C is in the RF busy state from the SOF (start of frame) of the request frame until the EOF (end of frame) of the response frame. The figure below shows the RF busy state of commands in the read group.

Figure 8. RF read command busy state

Write command group

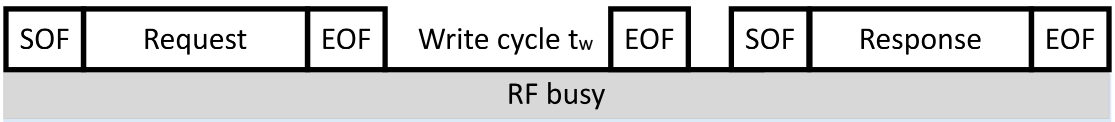
When decoding an RF write command, the ST25DVxxK is in the RF busy state from the SOF (start of frame) of the request frame until the EOF (end of frame) of the response frame. Write commands include a write cycle t_W. The figure below shows the RF busy state of commands in the write group.

Figure 9. RF write command busy state



If the Option_flag is set, the ST25DVxxK answers the request after receiving an additional EOF from the reader. In this case ST25DVxxK is in RF busy state from the SOF of the request until the EOF of the response.

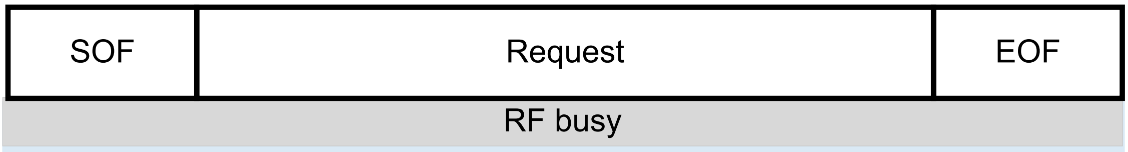
Figure 10. RF write command busy state when Option_flag is set



Stay quiet command

The stay quiet command is the only command defined as a single request frame (not followed by a response frame). The ST25DVxxK is in the RF busy state during the whole [SOF …. EOF] sequence, as shown in the figure below.

Figure 11. RF stay quiet command busy state



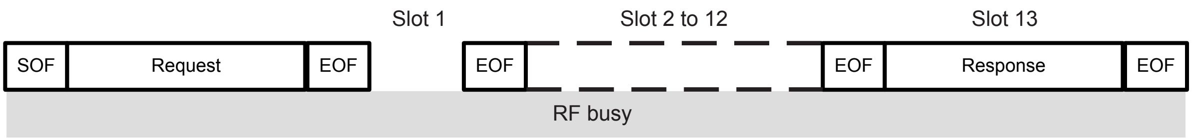
Inventory command

An inventory command is used when several ST25DVxxK devices are inside the range of the same RF electromagnetic field.

When the inventory command scans 16 slots, the ST25DVxxK 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 ST25DVxxK device might stay a long time in the RF busy state if it is decoded during the last (16^{th}) time slot.

Figure 12. Example of an inventory command where the ST25DVxxK is decoded in Slot 13



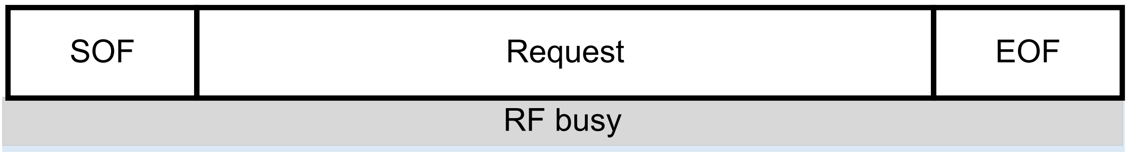
Unanswered commands

If several ISO15693 devices are in range of the same RF reader, ST25DVxxK can receive commands that require no answer. This can happen for command where Address_flag is set and UID is not matching, or when Select_flag is set and ST25DVxxK is not in selected state.

In the same way, if the ST25DVxxK is in Quiet state, and unless the request is a reset to ready command, the ST25DVxxK does not answer to the request.

In such cases the ST25DVxxK is in RF busy state from SOF (start of frame) of the request frame until EOF (end of frame) of the request frame.

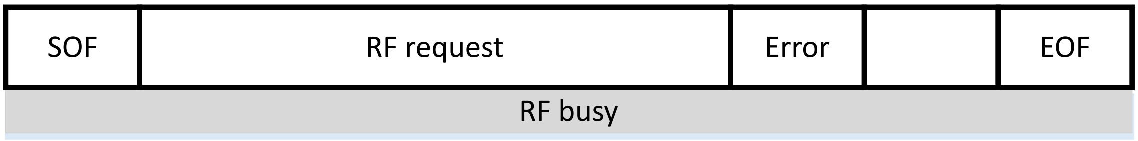
Figure 13. RF unanswered commands busy states



All commands

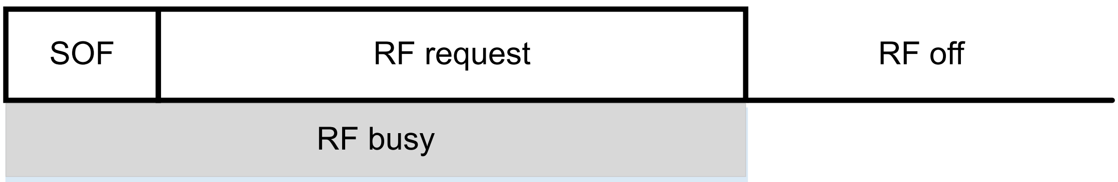
For any command, the ST25DVxxK exits the RF busy state at end of request if any framing or CRC error.

Figure 14. **RF busy state for any command when an RF error occurs**



For any command, the ST25DVxxK exits the RF busy state if the tag exits the RF field.

Figure 15. **RF busy state for any command when ST25DVxxK exists the RF field**



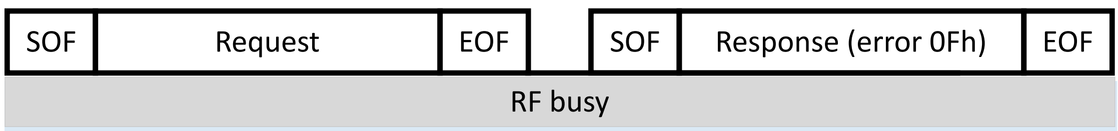
Independently of RF commands, ST25DVxxK has a special RF management feature that allows RF communication to be disabled or put in a sleep mode. ST25DVxxK RF busy state is dependent of this RF management configuration as well.

RF_DISABLE mode

ST25DVxxK can be configured in RF_DISABLE mode thanks to the RF_MNGT register (See section 5.4 RF Management feature in the ST25DVxxK datasheet). In RF_DISBALE mode, ST25DVxxK does not execute RF commands and returns error 0Fh.

When configured in RF_DISABLE mode, the ST25DVxxK is in RF busy state from SOF (start of frame) of the request frame until EOF (end of frame) of the response frame

Figure 16. **RF_DISABLE mode busy state**



RF_SLEEP mode

ST25DVxxK can be configured in RF_SLEEP mode via the RF_MNGT register (see section 5.4 RF Management feature in the ST25DVxxK datasheet). In RF_SLEEP mode, RF communications are disabled and the ST25DVxxK does not receive any RF command.

When configured in RF_SLEEP mode, the ST25DVxxK is in never in RF busy state.

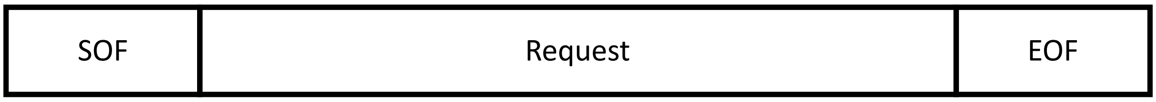


Figure 17. **RF_SLEEP mode busy state**

1.2.3 Arbitration

When both interfaces are active (as defined in Case 3 in Table 2. Four possible combinations of power supply sources), the ST25DVxxK 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	Action
The ST25DVxxK is in the I²C busy state: V_{CC} active and an I²C command is being decoded or executed	RF command received during an I²C command	The RF command is not decoded (1)
The ST25DVxxK is in the RF busy state: an RF command is being decoded or executed	V_{CC} active and I²C command received during an RF command	The I²C command is not decoded (2)

1. *RF commands inventory, stay quiet and addressed RF commands where UID match receive no response. Other RF commands receive error code 0Fh.*
2. *I²C master receives a slave “No Ack” state on 9th bit of first data byte (for instance device select).*

2 Recommendations when developing the application software

The application software has to take into account that a command might not be executed if the other channel (I^2C or RF) is already processing a command. The application software should therefore check the ST25DVxxK busy status before sending a command.

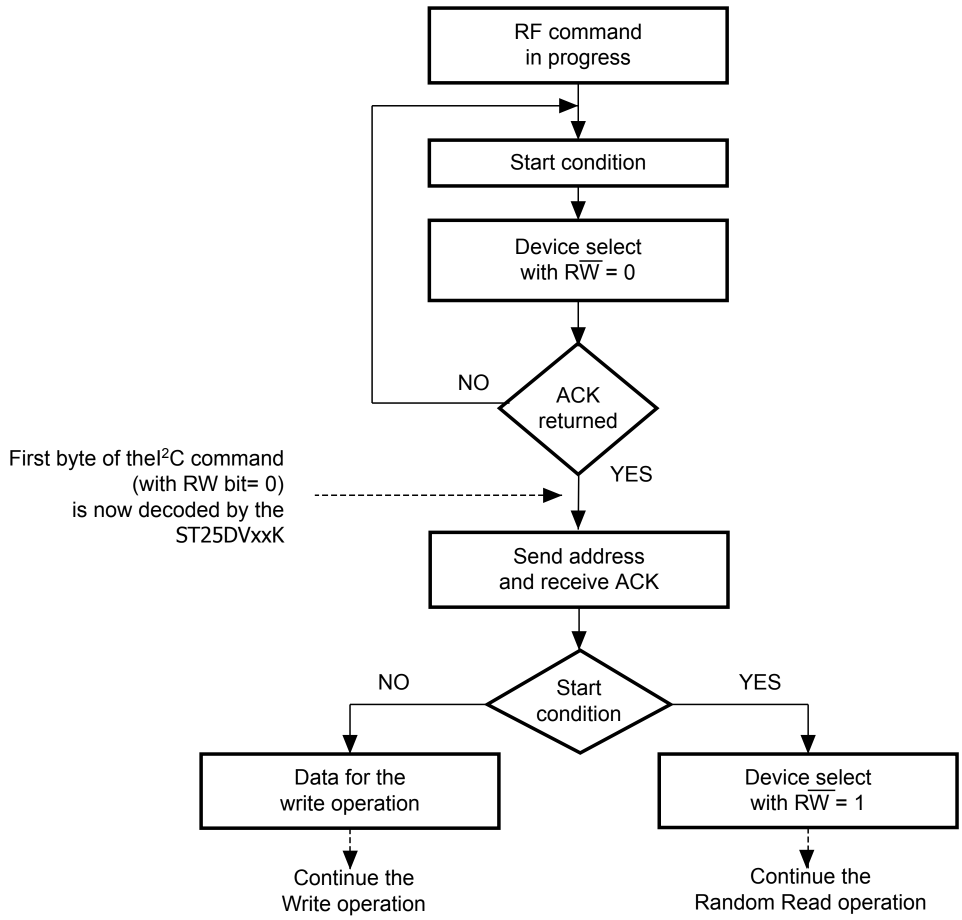
2.1 Issuing a command through the I^2C channel

2.1.1 I²C request while the RF channel is busy

I²C polling

If the ST25DVxxK is processing a command from the RF channel, no command issued on the I^2C bus is executed, therefore none of the bytes transmitted on the I^2C bus is acknowledged (NoAck). This information can be considered as the RF busy state and the application's I^2C software should include a polling loop on the RF busy state (with a timeout limit) when issuing a command on the I^2C bus. In this way, the I^2C command can be completed once the RF command under process has completed.

Figure 18. I²C polling when the RF channel is processing a command

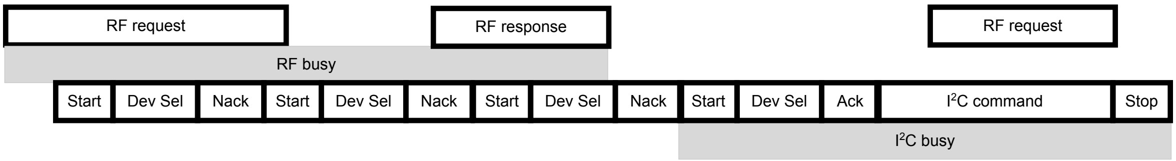


Important

It is paramount to exactly carry out the I²C polling sequence described in Figure 18. I²C polling when the RF channel is processing a command in order to keep the ST25DVxxK 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 EEPROM programming cycle t_W, for a write command).

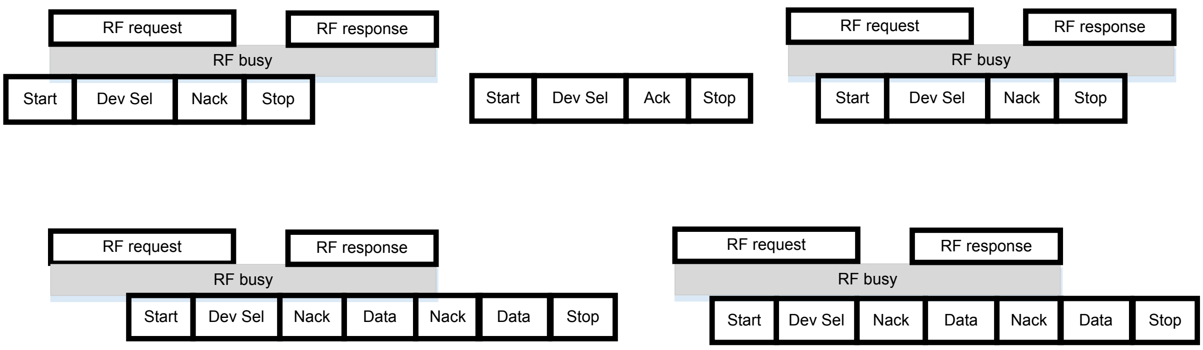
Figure 19. I²C polling right method



- **Wrong method:**
  - looping on the device select until it is acknowledged, sending a Stop condition and then initiating a new I2C command
  - looping on the complete I2C command until it is all acknowledged.

This is inadequate as an RF request might have been served between [Ack] and the new I2C command (time slot during which the ST25DVxxK is not in the I²C busy state).

Figure 20. I²C polling wrong method



2.1.2 I²C request following an I²C write in EEPROM memory

During the EEPROM programming cycle following the stop condition of an I²C write into EEPROM memory, I²C is in busy state (see Figure 3. I²C write cycle command busy state when writing in user memory or in system configuration area). Neither I²C or RF communication are possible. I²C commands are not acknowledged and RF commands are either answered with error 0x0F or not answered.

In order to determine when the next I²C command can be issued, the I²C master must be aware of the duration of the EEPROM programming cycle. This can be done by two different methods: I²C polling or calculated delay.

I²C polling method is identical to the method described in Section 2.1.1 I²C request while the RF channel is busy.

Calculated delay method requires the I²C master to calculate the EEPROM programming cycle duration and to wait the appropriate delay accordingly (after stop condition of the write command) before sending the next I²C command.

An I²C sequential write allows from 1 up to 4 bytes to be programmed in EEPROM in t_W, provided that they all share the same most significant memory address bits b16-b2.

The following C code example calculates the write cycle time:

```
programing_cycle = tw*(((start_address+nb_bytes_to_write-1)>>2)-(start_address>>2)+1);
```

Please refer to the ST25DVxxK datasheet for the t_W value.

2.1.3 Gaining I²C exclusive access

If the I²C host requires exclusive access to the ST25DVxxK, or if the application is disturbed by too great a number of decoded RF commands, it might be convenient that the I²C bus master temporarily disable the RF interface. This is done by configuring the ST25DVxxK in RF_SLEEP, writing value 02h in RF_MNGT_Dyn register (I^2C address 2003h). While in RF_SLEEP mode, the RF busy state never occurs and the I^2C host has exclusive access to the ST25DVxxK.

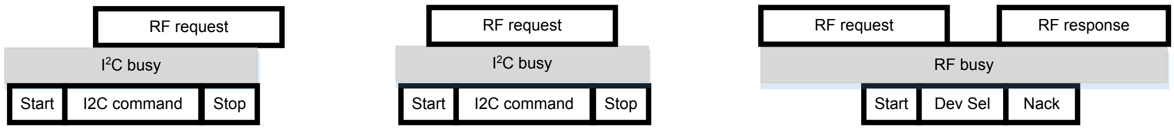
2.2 Issuing a command though the RF channel

2.2.1 RF request while the I²C channel is busy

If the ST25DVxxK is processing a command from the I^2C channel, no command issued on the RF channel is executed when the ST25DVxxK is I^2C busy. inventory, stay quiet and any addressed commands receive no response, other commands receive the error code 0Fh.

The application's RF software should include an "I^2C 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 I^2C commands under execution are completed.

Figure 21. RF polling while I^2C channel is busy



A condition for a safe ST25DVxxK application design is that all the sensitive data stored in the ST25DVxxK memory are protected with RF passwords, so that a spurious RF command cannot modify these data. Once an RF session (several commands) is completed, blindly send a Present Password command with an incorrect password value; this closes any opened security session.

2.3 RF write requests and ST25DVxxK power

As stated in Table 2. Four possible combinations of power supply sources, ST25DVxxK can process RF commands when VCC power is not supplied. Nevertheless, if ST25DVxxK is also powered through VCC, removing VCC during a RF command can abort the current command. If this happens during the programming cycle of an RF write command, this can lead to memory data corruption. This is especially important when programming new RF passwords since corrupted RF password cannot be recovered.

To avoid this issue, ones need to understand the different ways ST25DVxxK can be powered on its wired side through VCC and LPD pins.

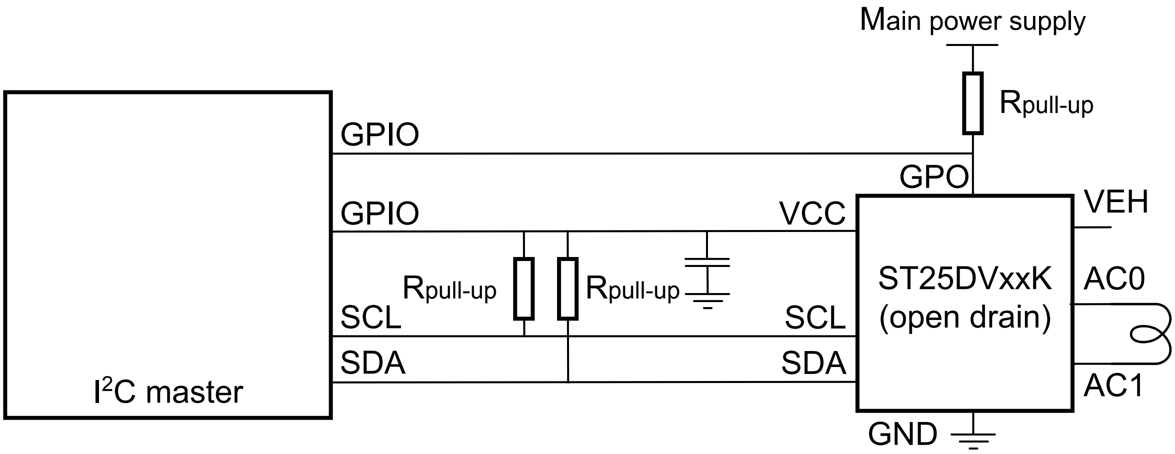
Open drain version

VCC pin can be permanently connect to the power source, or VCC can be delivered by a microcontroller GPIO pin, in which case the microcontroller is fully in control of ST25DVxxK wired power supply.

If ST25DVxxK's VCC is permanently connected to the power source, no special care should be taken regarding RF commands, as VCC never falls during an RF command.

If ST25DVxxK's VCC is controlled by a microcontroller GPIO pin, special cares should be taken. The primary application (for instance the microcontroller) fully controls the I²C bus and power supply, but it cannot always predict RF commands and thus cannot predict the right time to modify ST25DVxxK's VCC state.

Figure 22. **Example hardware application schematic of an ST25DVxxK open drain version where MCU GPIO is used to control ST25DVxxK's VCC**

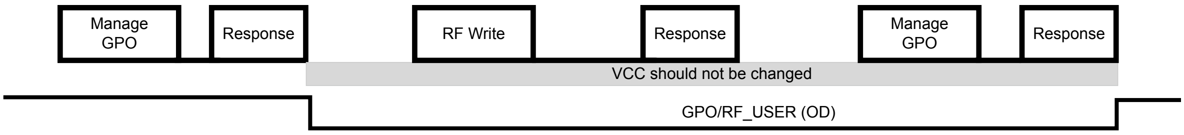


In such configuration, RF application can then use the GPO/RF_USER signal to inform the I²C master of an incoming RF write:

- RF application can use the Manage GPO command to set the GPO/RF_USER signal active before issuing a write command.
- after Write command completion, RF application can use the Manage GPO command again or remove RF field to reset the GPO/RF_USER signal.
- I²C master can read the GPO/RF_USER signal and the associated IT_STS_Dyn register before deciding to modify the VCC state: if GPO/RF_USER is active, the VCC state should not be modified. If GPO/RF_USER is not active, it is safe to shutdown ST25DVxxK's VCC.

Please see GPO Section of the ST25DVxxK datasheet for GPO configuration.

Figure 23. **Protection of RF writes with GPO/RF_USER signal**



RF application can also check if ST25DVxxK is powered through VCC by reading bit 3 of EH_CTRL_Dyn (VCC_ON) before issuing any RF write command.

Alternatively, if use of GPO interrupt is not possible or too complex to implement, a capacitor of minimal value of 1 µF should be connected between ST25DVxxK's VCC and ground. This capacitor ensures that VCC falling time is long enough for RF power to internally recover over VCC power. This allows that RF commands are not affected by falling VCC.

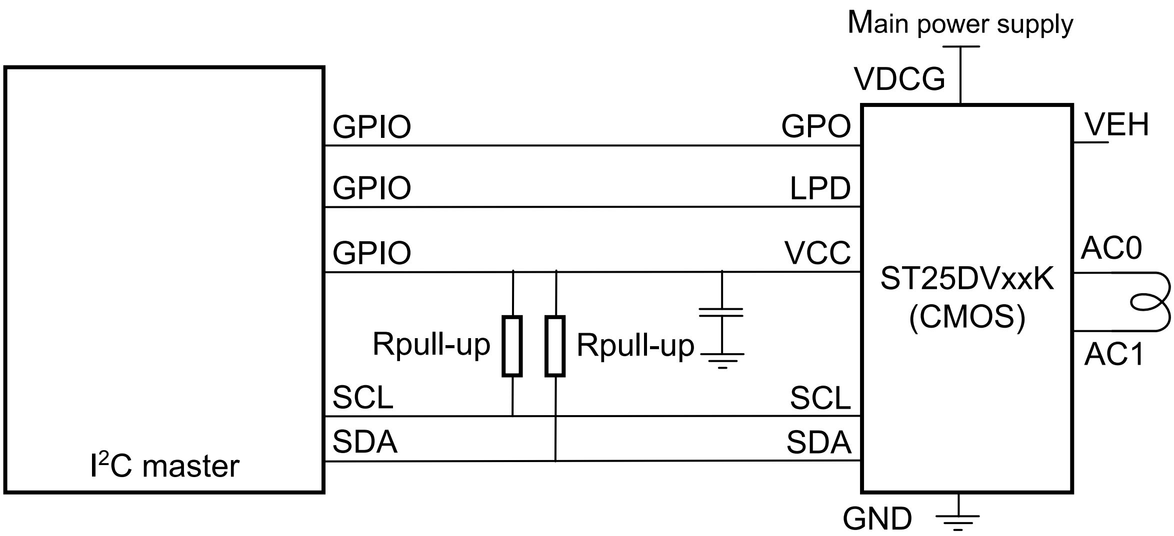
CMOS version

ST25DVxxK's VCC pin is connected to either permanent power supply or microcontroller GPIO pin. LPD pin is connected to and controlled by a microcontroller GPIO pin.

In such configuration, the microcontroller can power off the ST25DVxxK by setting the LPD pin high. Setting LPD pin high shutdowns the ST25DVxxK internal regulator so that it almost consumes no power (only leakage current lower than 1 µA remains. This leakage can safely be suppressed by setting VCC low after having set LPD high).

Setting LPD high, in contrary to setting VCC low, has no impact on on-going RF command. So, the microcontroller can safely do it at any time, white-out caring about RF status.

Figure 24. Hardware application schematic example of an ST25DVxxK CMOS version where MCU GPIO is used to control ST25DVxxK's VCC and LPD



Revision history

Table 4. Document revision history

Date	Revision	Changes
05-Dec-2019	1	Initial release.
04-Nov-2020	2	Changed the document scope from ST Restricted to public
09-Mar-2021	3	Updated: • Section Introduction • Section 1.1 Communications and power supply conditions • Section 1.2.1 I²C busy states • Section 1.2.3 Arbitration • Section 2.1.1 I²C request while the RF channel is busy • Section 2.3 RF write requests and ST25DVxxK power Added: • Section 2.1.2 I²C request following an I²C write in EEPROM memory

Contents

List of tables

List of figures

IMPORTANT NOTICE – PLEASE 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 acknowledgement.

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, please 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.

© 2021 STMicroelectronics – All rights reserved