# Errata Sheet

# RV-8803-C7

## Silicon Limitation

# TABLE OF CONTENTS

# 1. DEVICE LIMITATION I$^2$C COMMUNICATION

## 1.1. SILICON IDENTIFICATION

Production date code: M xxx bb
xxx = 412 to 706
bb = various
Part Designation: 8803

Hint: For the functional description of the devices without this silicon limitation (from date code 707), refer to Application Manual revision 1.3 and higher.

## 1.2. DESCRIPTION

The RV-8803-C7 has a built-in Watchdog function to autonomously RESET the I$^2$C Interface after a Bus-Timeout of 950 ms.
There is a short time window where the RESET is triggered erroneously. With standard I$^2$C-bus firmware, checking the Acknowledge, there are no false results.
Due to the automatic initialization of the I$^2$C timeout function at I$^2$C START it is possible that the RV-8803-C7 triggers an erroneous Bus-Timeout Reset and consequently responds with “no acknowledge” for the duration of maximum 61 µs. This unwanted Bus-Timeout Reset of the I$^2$C Interface occurs when a START condition is sent 950 ms + n * 1000 ms (n = 0, 1, 2, …) after a previous START condition.

## 1.3. PROBABILITY

The RV-8803-C7 will only send the “no acknowledge” for 10 ms if the consecutive communication is taking place 950 ms after the previous one.
The probability to hit the critical time window is slim: 10 ms / 1.0 s, equal to the ratio of 1:100.

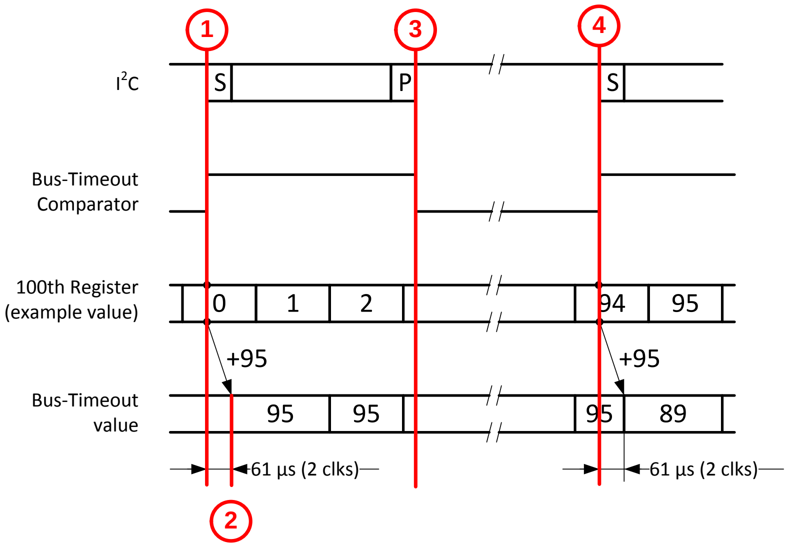
## 1.4. WORKAROUND

Software has to verify every acknowledgement from RV-8803-C7 and when a “no acknowledge” is detected has to begin communication procedure again with the I$^2$C Start condition.

## 1.5. DETAILED DESCRIPTION

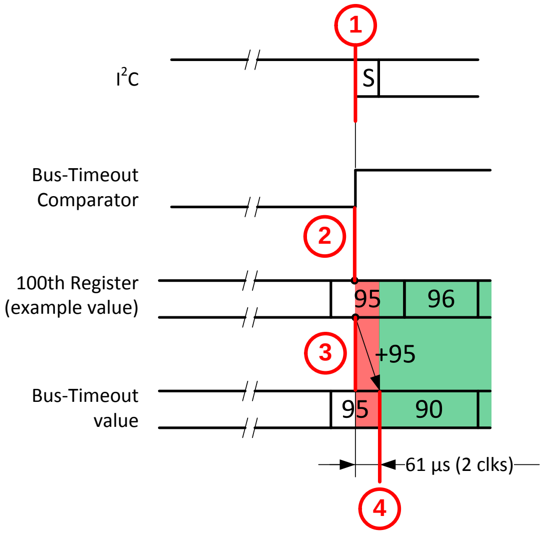
**Initialization of the I²C Interface Bus-Timeout by RV-8803-C7:**

1. At START condition, the Bus-Timeout Comparator is enabled.
2. It takes 2 clocks 32’768 Hz (61 µs) to read the value of the 1/100th seconds, adding 95/100th and write this as a Bus-Timeout target into the comparator.
3. At STOP condition, the comparator is disabled.
4. At the consecutive START (after STOP), the comparator is enabled with previous Bus-Timeout target value before the next value is calculated and written into the comparator.



**Erroneous Bus-Timeout Reset Scenario 1: New START exactly 950 ms after previous START.**

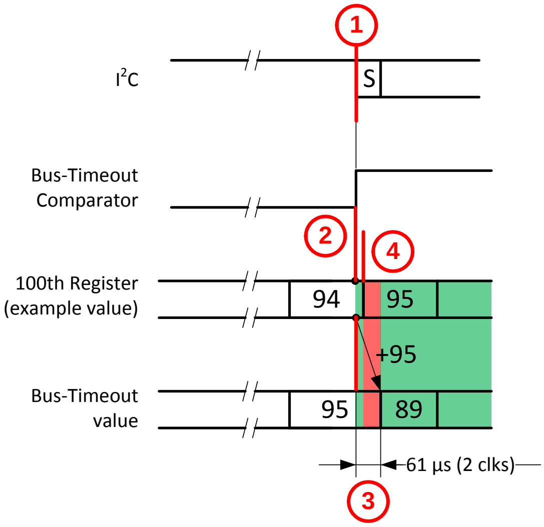
1. At START condition, the comparator is enabled.
2. At the consecutive START the comparator is enabled with previous Bus-Timeout target value.
3. If the actual 1/100th of Second matches with the previous Bus-Timeout target value, a RESET of the I²C Interface will be triggered immediately.
4. It takes 2 clocks 32’768 Hz (61 µs) to update the new Bus-Timeout target value.



Summarized: At START condition of the I²C Interface the Bus-Timeout comparator is enabled. If there is a match between the current value in the 1/100th Seconds register and the previous Bus-Timeout value, the comparator triggers an I²C Bus-Timeout RESET and the RV-8803-C7 will reply immediately with “no acknowledge”.

**Erroneous Bus-Timeout Reset Scenario 2: New START shortly before 950 ms after previous START**

1. At START condition, the comparator is enabled.
2. At the consecutive START the comparator is enabled with previous Bus-Timeout target value. As long as no match with the current 1/100th second is detected no RESET will be triggered.
3. It takes 2 clocks 32’768 Hz (61 µs) to update the new Bus-Timeout target value.
4. If during the 61 µs required for up-dating the Bus-Timeout value the 1/100th of Second is incremented and match the previous Bus-Timeout target value, the I²C Interface will be RESET.



Summarized: At START condition of the I²C Interface the Bus-Timeout comparator is enabled. The 1/100th Seconds register might have the value “previous Bus-Timeout value less one” and the RV-8803-C7 will replay with acknowledge. As soon as the there is a match between the current value in the 1/100th Seconds register and the previous set Bus-Timeout value, the comparator triggers an I²C Bus-Timeout RESET and the RV-8803-C7 will reply with “no acknowledge”.

## 1.6. WORKAROUND DETAILED

It's recommended to follow I$^2$C standard protocol and check ACK "acknowledge". When "no acknowledge" is detected, repeat communication procedure beginning with I$^2$C Start.

Based on the worst-case combination, a maximum of three consecutive i2c_write(); accesses may fail due to a Bus-Timeout RESET and respond with "no acknowledge":

- most unlucky timing interval of the START condition
- maximum I$^2$C Interface clock-speed (400 kHz)
- maximum 61 µs requirement up-dating the Bus-Timeout.

Workaround: Repeat I$^2$C communication procedure maximum 4 x times.

# 2. DEVICE LIMITATION I$^2$C STOP

## 2.1. SILICON IDENTIFICATION

Production date code: M xxx bb
xxx = 707 to 853
bb = various
Part Designation: 8803

Hint: For the functional description of the devices without this silicon limitation (from date code 901), refer to the new Application Manual, revision 1.5.
The WORKAROUND described in this errata sheet is no more mandatory, but existing customer software with this workaround doesn't need to be changed.

## 2.2. DESCRIPTION

The RV-8803-C7 does not detect correctly the I$^2$C STOP condition generated by the master to abort a data transfer. On the other hand, with an I$^2$C START condition a data transfer is correctly executed and the bus logic of the RV-8803-C7 is correctly reset.

After an I$^2$C Write Operation followed by an I$^2$C STOP the bus logic of the RV-8803-C7 remains active and in special case where clocking is done on clock input line SCL without making a new START condition (e.g. when free clocking I$^2$C bus) the previous write sequence will be continued and data are erroneously written to the registers and RV-8803-C7 will respond with Acknowledge (ACK). Since data line SDA is high during this period of clocking, FFh is written to the current register the address pointer is pointing to. Because the address pointer is incremented automatically the following registers are also overwritten with FFh.

## 2.3. PROBLEM

The not detected I$^2$C STOP condition after an I$^2$C Write Operation by the RV-8803-C7 has an influence on two specific work situations:

1. When the last transmission was an I$^2$C Write Operation followed by an I$^2$C STOP and when RV-8803-C7 is integrated in a circuit with backup operation (refer to section 8.1. in the Application Manual) and the main power voltage $V_{DD}$ of the MCU with the I$^2$C bus is powered on and off alternately, the RV-8803-C7 understands this voltage transitions as clocking signal and the previous write sequence will be continued and FFh is erroneously written to the registers and RV-8803-C7 will respond with Acknowledge (ACK).

2. When the last transmission was an I$^2$C Write Operation followed by an I$^2$C STOP and when the application is using a free-clocking function for the I$^2$C bus, the RV-8803-C7 misunderstands this free-clocking pulses on clock input line SCL as normal clocking signal and the previous write sequence will be continued and FFh is erroneously written to the registers and RV-8803-C7 will respond with Acknowledge (ACK).

## 2.4. WORKAROUND

Complete the I$^2$C-bus access always with a Read Operation followed by the STOP condition.

```
/////////////////////////////////////////////////////////////////////////////////
// When Writing to a register (or registers), complete the access with a Reading.
// This example with the Seconds Register is described in pseudo code.
//
// Writing
I2C START
Slave Address, 64h;              // Slave Address RV-8803-C7 for a write operation
Jump to Seconds Register (00h)   // 00h = Seconds Register Address
Write 30 to the Register         // For example 30 seconds
I2C STOP
//
// Complete the access with a Reading
I2C START
Slave Address, 64h               // Slave Address RV-8803-C7 for a write operation
Jump to Seconds Register (00h)   // 00h = Seconds Register Address
I2C STOP
I2C START
Slave Address, 65h               // Slave Address RV-8803-C7 for a read operation
Read value from the Register     // Reads value from the Seconds Register
Return value                     // Returns actual Seconds value
I2C STOP
/////////////////////////////////////////////////////////////////////////////////
```

## 3. DOCUMENT REVISION HISTORY

| Date | Revision # | Revision Details |
|---|---|---|
| April 2016 | 1.0 | First release |
| May 2016 | 1.1 | Completed description, 1.2.<br>Added probability, 1.3. |
| October 2017 | 1.2. | Added “solved from date code 707”, front page<br>Added “end date 706”, 1.1.<br>Added Device Limitation I$^2$C Stop, 2. |
| February 2019 | 1.3 | Added Silicon Identification, 2.1. |

**Information furnished is believed to be accurate and reliable. However, Micro Crystal assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. In accordance with our policy of continuous development and improvement, Micro Crystal reserves the right to modify specifications mentioned in this publication without prior notice. This product is not authorized for use as critical component in life support devices or systems.**

MICRO CRYSTAL SWITZERLAND
A COMPANY OF THE SWATCH GROUP

Micro Crystal AG
Muehlestrasse 14
CH-2540 Grenchen
Switzerland

Phone +41 32 655 82 82
Fax +41 32 655 82 83
sales@microcrystal.com
www.microcrystal.com