AN1471
APPLICATION NOTE

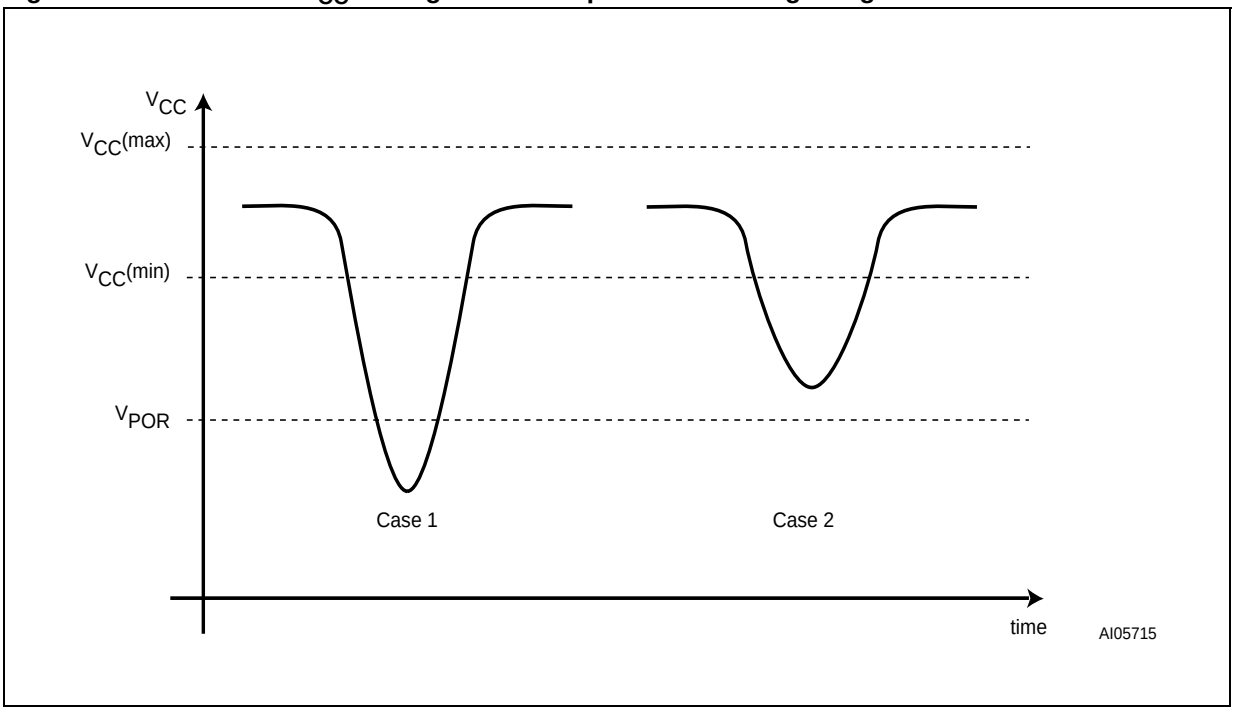
What Happens to the M24xxx I²C EEPROM If the I²C Bus Communication is Stopped?

This document describes what needs to be done to set an M24xxx device back to a known state if it has been suddenly stopped before completion of the current I²C instruction, for example due to a power failure at the Bus Master.

The answer can be structured under the following main headings, as summarised by the two cases shown in Figure 1:

- Case 1: V_{CC} dropped below the Power on Reset threshold, V_{POR}
- Case 2: V_{CC} dropped to a value between the minimum operating voltage and the Power on Reset threshold

Figure 1. Two Cases of V_{CC} Falling out of its Specified Working Range



The minimum operating voltage, V_{CC}(min), is 4.5 V for a M24xxx, 2.5 V for a M24xxx-W, and 1.8 V for a M24xxx-S or M24xxx-R. The V_{POR} threshold is about 1.5 V.

CASE 1: V_{CC} DROPS BELOW THE POWER ON RESET THRESHOLD, V_{POR}

Once V_{CC} drops below V_{POR}, the M24xxx internal logic is frozen. When V_{CC} rises again, and goes above the V_{POR} threshold, the M24xxx internal logic is reset to a known state (with the address counter initialized to 0), and the device is ready, while V_{CC} is above V_{CC}(min), to decode any newly incoming instructions.

CASE 2: V_{CC} DROPS TO A VALUE BETWEEN THE MINIMUM OPERATING VOLTAGE AND THE POWER ON RESET THRESHOLD

When V_{CC} eventually recovers, to be within the specified voltage range ($V_{CC}(max)>V_{CC}>V_{CC}(min)$), the internal state of the M24xxx is in an unknown state. The Bus Master (the microcontroller or processor) and other components might well have failed, with clock and data lines now being improperly driven.

The M24xxx internal logic must be re-initialized. The analysis of this situation can be structured under the following sub-headings:

- The interrupted transmission was an *Incoming Data Byte*
- The interrupted transmission was an *Outgoing Data Byte (during a READ cycle)*

The interrupted transmission was an *Incoming Data Byte*

The issue of a STOP condition is sufficient to abort the transmission. However, if the last transmitted instruction was a WRITE, the STOP condition is also able to start the internal Write cycle (if the STOP condition occurs after the 9th clock cycle). It is therefore risky to send a single STOP condition.

It is recommended, instead, to issue a START condition first, followed by a STOP condition. The START condition aborts the transmission, and leaves the M24xxx waiting for a Device Select Code; the STOP condition then sets the M24xxx in stand-by mode.

The internal address counter will *not* have been reset to 0 by this (unless V_{CC} drops below the V_{POR} threshold). So the next instruction should not be a Current Read or Sequential Current Read, since neither of these defines an address. Instead, the next instruction must be a Byte Random Read, Sequential Random Read or Write.

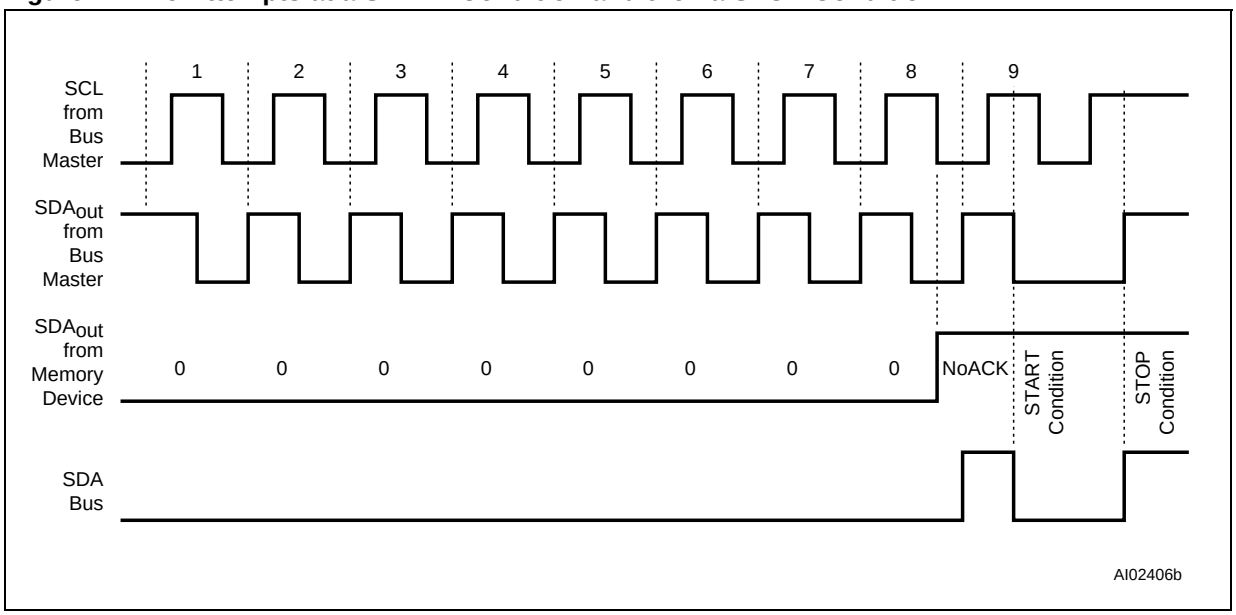
The interrupted transmission was an *Outgoing Data Byte (during a READ cycle)*

Figure 2 shows the waveform of two overlapping trains of pulses, on SCL and SDA, that are recommended for the Bus Master to send to the M24xxx. This is not a standard sequence from the I^2C protocol, but a specially designed sequence for clearing the device to a known state.

The main part of the sequence consists of nine rising edges of SCL, interleaved with nine attempts by the Bus Master to force a falling edge on SDA while SCL is High. In this way, the Bus Master makes nine attempts at causing a START condition. It is certain to succeed on the nineth attempt, if not before, and can be followed by a STOP condition, thus putting the device in its known standby state.

The reason for the possible failure of the previous eight attempts at causing a START condition is depicted in Figure 2. The M24xxx might have been left in a state in which it was part way through clocking out a byte on the SDA bus. If the bit to be clocked out is a zero, the memory device will hold the SDA line Low, thus preventing the Bus Master from putting a falling edge on this line. In the worst case, the memory device might be at the start of clocking out eight 0s, if the data byte has been 00h, as shown in Figure 2.

Figure 2. Nine Attempts at a START Condition and then a STOP Condition



Again, the internal address counter will *not* have been reset to 0 by this (unless V_{CC} drops below the V_{POR} threshold). So the next instruction should not be a Current Read or Sequential Current Read, since neither of these defines an address. Instead, the next instruction must be a Byte Random Read, Sequential Random Read or Write.

Universal Reset Sequence

Conservatively, the method shown in Figure 2 will work regardless of whether the device had been stopped during an incoming byte transfer, or during an outgoing byte transfer. It can be used as a universal method of resetting the memory device whenever an undefined state has been detected to have occurred on the I^2C bus.

For current information on ST products, please consult our pages on the world wide web:
www.st.com

If you have any questions or suggestions concerning the matters raised in this document, please send them to the following electronic mail addresses:

apps.serial-flash@st.com (for application support)
ask.memory@st.com (for general enquiries)

Please remember to include your name, company, location, telephone number and fax number.