

# AP16036

## C16x

### PEC - Lowering the own Interrupt Priority

Microcontrollers



Never stop thinking.

<b>Revision History:</b>		2004-02	<b>V 1.0</b>
Previous Version:		-	
Page	Subjects (major changes since last revision)		
All	Updated Layout to Infineon Corporate Design, updated release to 1.0, Content unchanged!		

Controller Area Network (CAN): License of Robert Bosch GmbH

**We Listen to Your Comments**

Any information within this document that you feel is wrong, unclear or missing at all?  
Your feedback will help us to continuously improve the quality of this document.

Please send your proposal (including a reference to this document) to:

**[mcdocu.comments@infineon.com](mailto:mcdocu.comments@infineon.com)**



**Edition 2004-02**

**Published by  
Infineon Technologies AG  
81726 München, Germany**

**© Infineon Technologies AG 2006.  
All Rights Reserved.**

#### **LEGAL DISCLAIMER**

THE INFORMATION GIVEN IN THIS APPLICATION NOTE IS GIVEN AS A HINT FOR THE IMPLEMENTATION OF THE INFINEON TECHNOLOGIES COMPONENT ONLY AND SHALL NOT BE REGARDED AS ANY DESCRIPTION OR WARRANTY OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE INFINEON TECHNOLOGIES COMPONENT. THE RECIPIENT OF THIS APPLICATION NOTE MUST VERIFY ANY FUNCTION DESCRIBED HEREIN IN THE REAL APPLICATION. INFINEON TECHNOLOGIES HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND (INCLUDING WITHOUT LIMITATION WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY) WITH RESPECT TO ANY AND ALL INFORMATION GIVEN IN THIS APPLICATION NOTE.

#### **Information**

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office ([www.infineon.com](http://www.infineon.com)).

#### **Warnings**

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

<b>Table of Contents</b>		<b>Page</b>
1	Abstract .....	4
2	Lower the "own" priority by changing the ILVL-field in the PSW-Register .....	4
2.1	The interrupted-level was "low" .....	4
2.2	The interrupted-process has a higher priority then the targeted priority for the "End-of-PEC-Interrupt" service routine .....	4
3	A different approach: .....	4

## 1 Abstract

Very often it is needed to lower the interrupt priority while running a interrupt-service-routine (ISR),like after the last PEC-transfer.

**Example:** For storing a time-critical signal, a PEC-transfer will be used. Because the processing of the stored data is not time-critical it will be done in the ISR triggered by the last PEC transfer.

The PEC will be initialized with a counter value of "1". So, after a single PEC-transfer an additional Interrupt is generated. This interrupt is on the same high interrupt level as the PEC (14 or 15). If this PEC-triggered ISR stays at this level nearly every other interrupt is disabled and has to wait, even while the program is now running the non-critical part.

Now it is on the programmers responsibility to free the system for higher priority-levels. There are different possibilities to do that.

Lower the “own” priority by changing the ILVL-field in the PSW-Register

## 2 Lower the “own” priority by changing the ILVL-field in the PSW-Register

On the first view this looks like the best and easiest solution, although it will have severe disadvantages.

### 2.1 The interrupted-level was “low”

The CPU level is 0, no other interrupt is served. Now, an Interrupt with a single PEC-transfer with a followed “End-of-PEC-Interrupt” is running. Within the ISR the priority is lowered for other high-level-requests. The system will work correctly.

### 2.2 The interrupted-process has a higher priority then the targeted priority for the “End-of-PEC-Interrupt” service routine

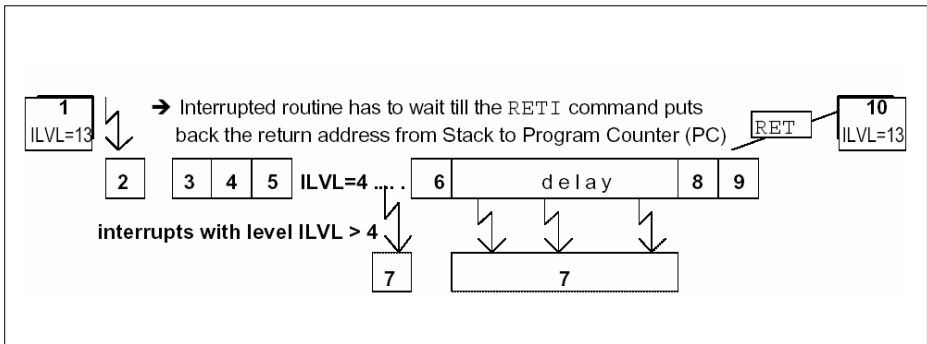
The targeted priority for the “End-of-PEC-Interrupt” service routine is defined as the priority which is changed in the ISR.

**Example:** An Event with the high priority “ILVL=13” can accept a latency within 10 $\mu$ s. Only PEC-transfers are usually able to interrupt level 13. Because a PEC-transfer will only “steal” one cycle, it will not be a problem due to the latency requirement. Every PEC-transfer (PEC-counter <255) will end up with the “End-of-PEC-Interrupt” and will take more or less time in processing. Assuming the serial interface received 20 character, which need further processing in the “End-of-PEC-Interrupt”-service-routine. And assuming the time-window for this sequence is 50ms.

**Lower the “own” priority by changing the ILVL-field in the PSW-Register**

**The following sequence displays this case:**

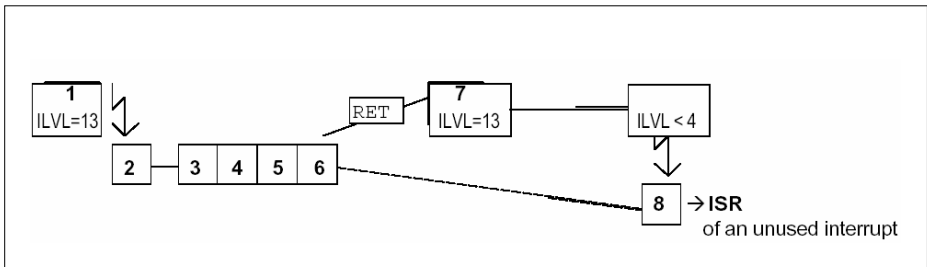
1. CPU running with high-priority-event at level ILVL=13
2. The last character of a block(20) is received by the serial interface
3. PEC-transfer is followed by the “End-of-PEC-Interrupt”
4. Reinitialize the PEC for the serial interface
5. Lower the “own” priority-level to ILVL=4
6. Start processing of the 20 characters, will take max. 50ms.
7. different interrupts with level ILVL>4 will occur
8. Finish the processing of the 20 characters.
9. ISR will be terminated with `RETI`
10. Reestablishing of interrupted routine at level ILVL=13



**Figure 1 With this setup a latency-time smaller than 10µs is nearly impossible to guarantee!**

### 3 A different approach:

1. CPU running with high-priority-event at level ILVL=13
2. The last character of a block(20) is received by the serial interface
3. PEC-transfer is followed by the “End-of-PEC-Interrupt”
4. Reinitialize the PEC for the serial interface
5. Setting of an unused interrupt-request-flags(at level ILVL=4) which is pointing to the routine to processing the 20 characters
6. Terminate “End-of-PEC-Interrupt” with `RETI`
- 7. Reestablishing of interrupted routine at level ILVL=13**
8. Routine to process the 20 characters will wait till its priority (ILVL=4) is high enough to interrupt the current CPU priority.



**Figure 2**

The disadvantage of this solution is that 2 interrupt-vectors for one event are used, and note that the code-size will increase slightly.

Alternatively could the interrupt priority during the ISR in the ILVL-Register be lowered (changed), then setting the “own” Interrupt-Request-Flag and processing the `RETI` instruction. A check of the current interrupted CPU level at the beginning of the ISR has to be implemented, too.

This should only be used if there are NO other interrupt vectors available and NO event which need a PEC response would occur during this procedure.

Because of the above mentioned reasons, “lowering the *own* priority” should be avoided.



<http://www.infineon.com>

Published by Infineon Technologies AG