



OPTIMIZING THE ST6 A/D CONVERTER ACCURACY

INTRODUCTION

When using the internal Analog to Digital Converter of the ST62 family and maximum A/D converter accuracy is required, it is desirable to filter out any noise present on the analog input, but also noise present on the ground and V_{CC} supply lines of the MCU as V_{CC} is also the voltage reference of the A/D converter. Good decoupling must be made with capacitors on the analog input and between V_{CC} and ground. It is also recommended to put the MCU in wait state while the conversion is in progress, so as to minimize noise injected into V_{CC} by the operation of the micro-controller itself.

Finally, when enough time is available, it is highly recommended to make several successive A/D conversions and take an average of the results. This is the most effective way to get the most accuracy out of the ST6 family A/D converter.

The following code fragment demonstrates a burst of 256 successive measurements, after which the average is put into the accumulator. The whole routine takes approximately 30 milliseconds with an 8 MHz clock. When less time is available, it is of course possible to reduce the number of conversions: 8, 16 or 32 conversions also give good results, although the most conversions give the best results.

Optimizing The St6 A/d Converter Accuracy

```
;***** SUBROUTINE AVERAGE *****
;description: measures ADC input 256 times and stores average      *
;      of the 256 measures into accumulator                        *
;*****
average
    lid ior,10h    ; global enable interrupts
    clr aver_lo    ;aver_lo, aver_hi and count are RAM registers
    clr aver_hi
    ldi count,255  ; set for 256 measurements
aver1
    ldi adcc,10110000b ; start conversion with interrupt
    wait
    ld a,adc
;===== two byte addition of adc to 16-bit word:
    add a, aver_lo
    jrnC aver2
    inc aver_hi
aver2 ld aver_lo,a
;===== end of two byte addition
    ld a,count
    jrZ aver4
    dec count
    jp aver1    ;do it 256 times
aver4 ld a,aver_lo
    cpi a,127    ;round to next value if decimal part >0.5
    jrc aver3
    inc aver_hi
aver3 ld a, aver_hi ;store high byte of result into accumulator,
    ret          ;the low byte is not significant
;***** interrupt service routine *****
adcint
    ldi adcc,10h
    reti

;***** interrupt vector *****
.org 0ff0h
    jp adcint
```

“THE PRESENT NOTE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS WITH INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM THE CONTENT OF SUCH A NOTE AND/OR THE USE MADE BY CUSTOMERS OF THE INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.”

Information furnished is believed to be accurate and reliable. However, STMicroelectronics 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. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics.

All other names are the property of their respective owners

© 2004 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia – Belgium - Brazil - Canada - China – Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com