

### DRIVING AN ANALOG KEYBOARD WITH THE ST7 ADC

By Microcontroller Division Applications

## 1 INTRODUCTION

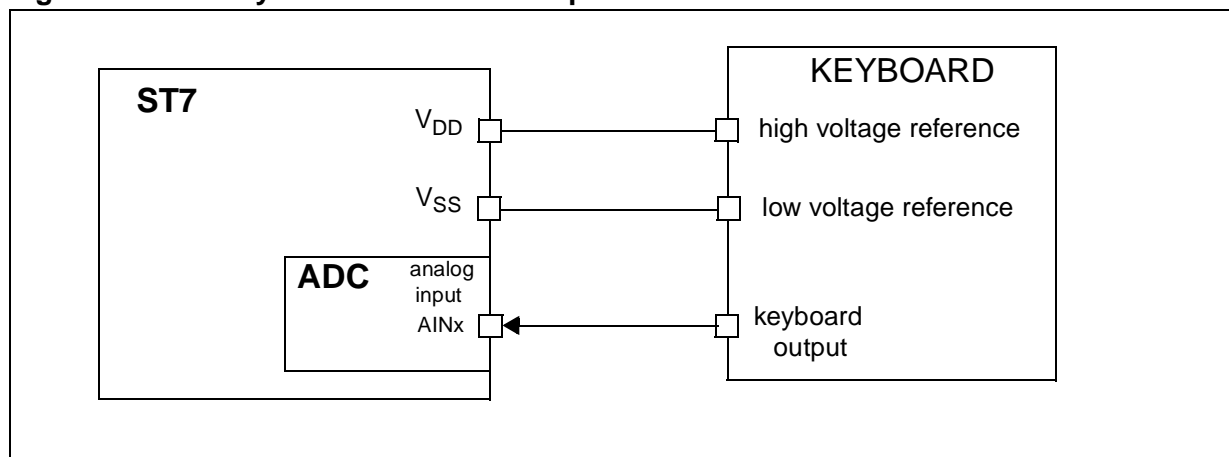
The goal of this application note is to present a standard example of the use of the Analog to Digital Converter (ADC) of the ST7.

In this note, the ADC is used to convert a 16-key analog keyboard. The technique for the keyboard is to connect the keys by resistive dividers to one of the converter inputs. The Application note is based on ST72F324 which has an embedded 10-bit ADC.

## 2 ST7 / KEYBOARD INTERFACE

Connect the analog keyboard to one of the analog inputs of the ST7 and connect the high and low voltage references as shown in Figure 1.

**Figure 1. ST7 / keyboard interface set-up**



## 3 ST72324 CONFIGURATION

The application has been tested with a ST72F324 configured as described below. Refer to the datasheet for more details on configuring the ST72F324.

### 3.1 I/O CONTROL

Depending on the number of pins available on the device package, the ST7 theoretically allows up to 16 multiplexed analog inputs to the ADC, but in practice, on the ST72F324 device, there can be up to 12 analog inputs. They are alternate functions I/O Port D (PD0-PD5), Port

F (PF0, PF4), and Port C (PC0, PC1, PC5, PC7). The I/O port pins used by the ADC must be configured as floating inputs to avoid conflicts in alternate function mode.

Refer to the Data Sheet for information on configuring the I/O ports.

### 3.2 ANALOG TO DIGITAL CONVERTER

The ST7 ADC is a 10-bit successive approximation converter, with internal sample and hold-circuitry.

#### 3.2.1 ADC control

You control the ADC using the ADC Control Status Register (ADCCSR).

EOC	SPEED	ADON	0	CH3	CH2	CH1	CH0
-----	-------	------	---	-----	-----	-----	-----

The EOC bit is the end of conversion bit:

- When this bit is set, the conversion is done and result can be read from the ADC Data Registers (ADCDRL & ADCDRH).
- When the bit is reset, the conversion is not complete.

The SPEED bit:

- When this bit is set,  $f_{ADC} = f_{CPU} / 2$ .
- When this bit is reset,  $f_{ADC} = f_{CPU} / 4$ .

The ADON bit:

- Enable ADC and start conversion when this bit is set.
- Disable ADC and stop conversion when this bit is reset.

CH3-CH0 bits:

- They are used to select which analog input to convert. In the ST72324 there are 12 analog pins.

#### 3.2.2 Characteristics

The conversion time is 15 ADC cycles including a sampling time of 4 ADC cycles.

The ADC is linear and the digital result of the conversion is given by the formula:

$$\text{Digital Result} = \frac{1023 * \text{Input Voltage}}{\text{Reference Voltage}}$$

#### 3.2.3 Process

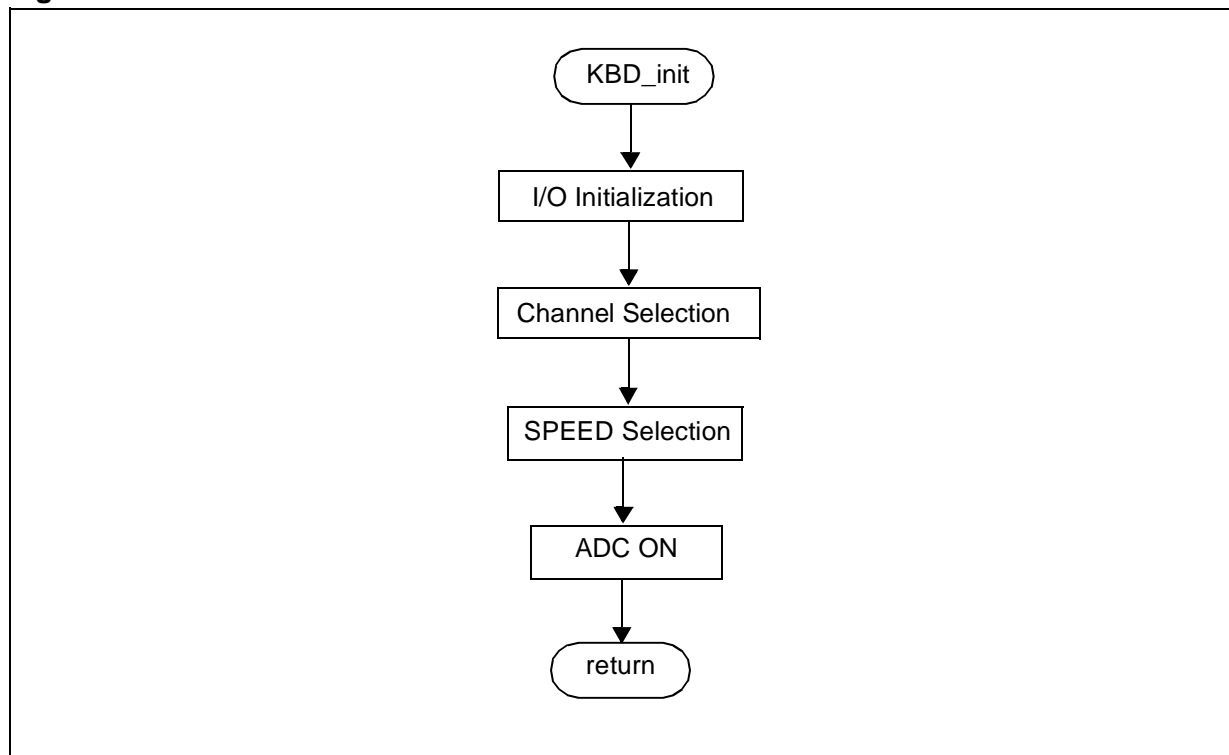
First the analog input pins must be configured as floating inputs (see Section 3.1).

Then the analog channel to convert must be selected using CH3-CH0 bits of ADCCSR register.

Then select the SPEED of conversion.

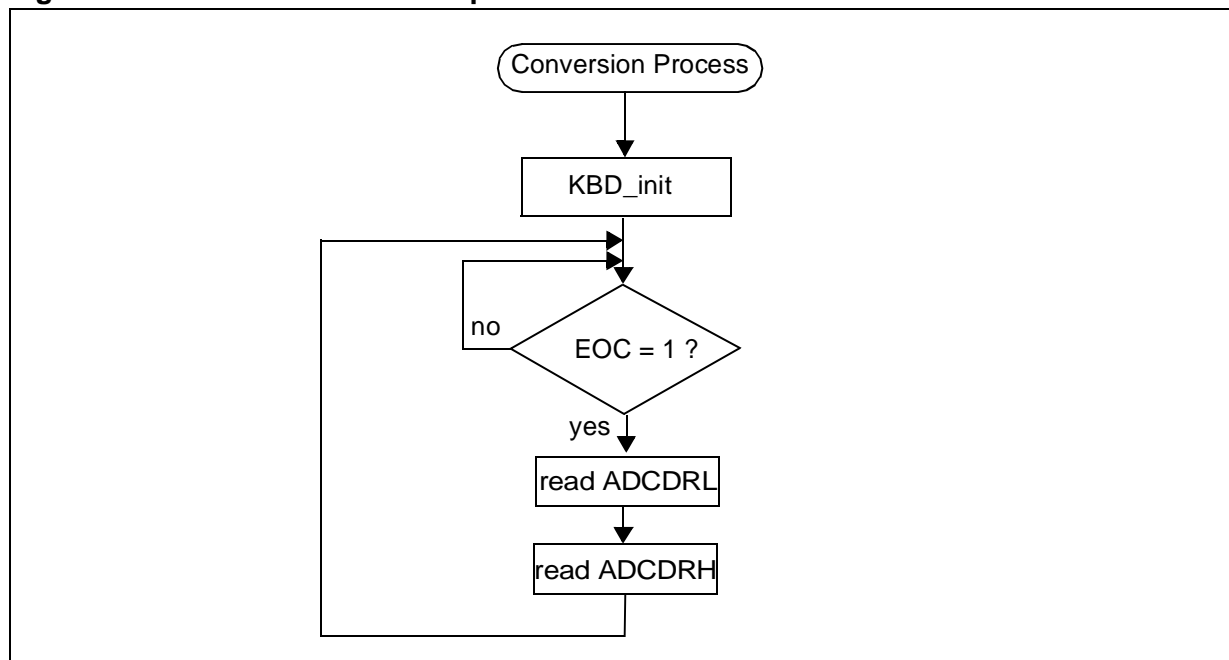
Setting the ADON bit will switch the converter on.

**Figure 2. Flowchart: initialization of the ADC**



Once a conversion is done, the EOC bit is set by hardware. It will be reset when the ADCDRH register is read.

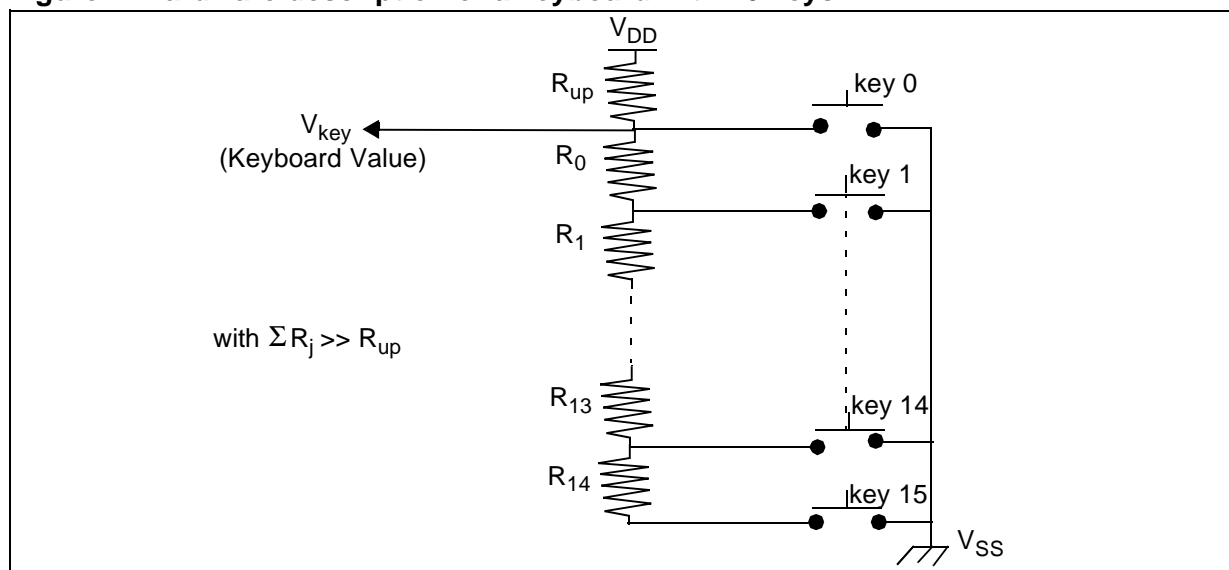
Once enabled, conversions will run continuously until the peripheral is disabled.

**Figure 3. Flowchart: conversion process**

## 4 ANALOG KEYBOARD

### 4.1 PRINCIPLE

The purpose is to recognize a key when pressed. In an analog keyboard each key is associated with a voltage. The description of an analog keyboard is given by Figure 4.

**Figure 4. Hardware description of a keyboard with 16 keys**

$R_{up}$  is a pull-up resistor. So, when no key is pressed,  $V_{key}$  is equal to  $V_{DD}$ .

When 'key i' is pressed (where i is in the range 1 to 15), the resistor  $R_{i-1}$  is connected to  $V_{SS}$ . Then we have a resistive divider

and  $V_{key}$  is given by the formula:

$$V_{key\ i} = \frac{(V_{DD} - V_{SS}) \sum_{j=0}^{i-1} R_j}{R_{up} + \sum_{j=0}^{i-1} R_j}$$

So the corresponding voltage of each key is given by the values of the resistors. An equal distribution of voltage between  $V_{DD}$  and  $V_{SS}$  is usually recommended.

To recognize a key, the user will measure  $V_{key}$  and will be able to decide which key was pressed.

## 4.2 PRACTICAL LIMITATIONS

Theoretically, with an 10-bit ADC, 1023 keys can be decoded. But potential errors must be taken into account. They can come from the power supply, the key resistivity, the resistor tolerance, the ADC conversion errors.

The resistor tolerance is the main limitation as usually 5% tolerance resistors are used. It is advised to use a 1% tolerance resistor for the pull-up. Changing this resistor greatly improves the keyboard as the pull-up has an influence on every key.

The ADC total unadjusted error (TUE) for the device is specified as 4 LSB. So, it has to be taken into account to avoid any key decision error.

These parameters will reduce the number of keys that can be efficiently decoded.

## 5 KEY DETECTION

The ST7 is a digital microcontroller. It uses its ADC to measure  $V_{key}$ . It is then coded in 10 bits. As the ADC is linear, the best decision is taken when the voltage levels of the keys follow an equal distribution between  $V_{DD}$  and  $V_{SS}$ .

In our application, a 16-key keyboard is used. So, the best associated  $V_{key}$  value of 'key i' is given by:

$$V_{key\ i} = \frac{(V_{DD} - V_{SS}) * i}{16}$$

The problem is that you cannot choose the perfect values for the resistors. In our application, the following resistor values were used (see Table 1):

**Table 1. Example of resistor values for 16 keys**

Resistor	Value ( $\Omega$ )	Resistor	Value ( $\Omega$ )
$R_{up}$	1K	R7	220
R0	68	R8	270
R1	75	R9	390
R2	82	R10	560
R3	100	R11	820
R4	120	R12	1K2
R5	150	R13	2K7
R6	180	R14	75

The digital values of the keys after conversion are given in Table 2.

When a key is pressed and after conversion, a decision must be taken on its value. Upper and lower limits of the detection for each key must be defined. These values are the middle of two subsequent typical values, which gives the best noise margin between keys.

The software uses the lower limits to make its decision. They are given in Table 2.

Table 2. Key Values

KEY	typical digital value	lower digital limit of KEY	KEY	typical digital value	lower digital limit of KEY
no key	0x3FF	0x3DF			
'F'	0x3C0	0x38E	'7'	0x1C0	0x1A0
'E'	0x37E	0x35E	'6'	0x17E	0x15E
'D'	0x33D	0x31D	'5'	0x13C	0x11C
'C'	0x303	0x2E3	'4'	0x0FB	0x0DB
'B'	0x2C2	0x2A2	'3'	0x0BC	0x09C
'A'	0x27E	0x25E	'2'	0x080	0x060
'9'	0x23D	0x21D	'1'	0x041	0x021
'8'	0x200	0x1E0	'0'	0x000	0x000

## 6 DRIVING AN ANALOG KEYBOARD WITH THE ST7 ADC

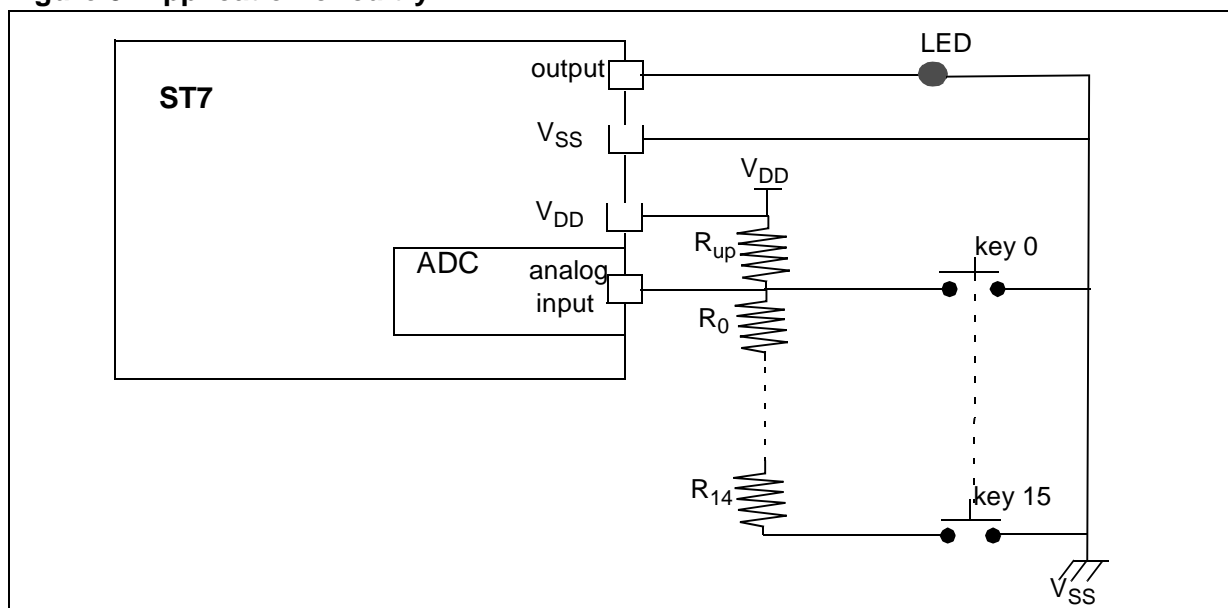
The complete software is provided in an attached zip file (in assembly & C using ST7 software library version 1.1) to this application note. It is only an example. It is up to the user to adapt it to the application.

### 6.1 GENERAL DESCRIPTION

A 16-key keyboard is connected to the ST72F324 through the AIN0 pin on Port D.

A LED is connected to an output pin (PC0) to indicate to the user that a key is detected.

**Figure 5. Application circuitry**



The software uses a polling strategy. It makes conversions continuously (no wake-up process).

First the software initializes the I/Os and the ADC.

When a conversion is done, the software decides if a key was pressed or not.

If not, it waits before analyzing the result of a new conversion.

If a key is pressed, it compares it with the former key to know if the key is stabilized. Then, a LED is switched on.



## 6.2 FLOWCHARTS

The main flowcharts of the application are given below in Figure 6, Figure 7 & Figure 8. The initialization routine is presented in Figure 2.

**Figure 6. Flowchart: main program**

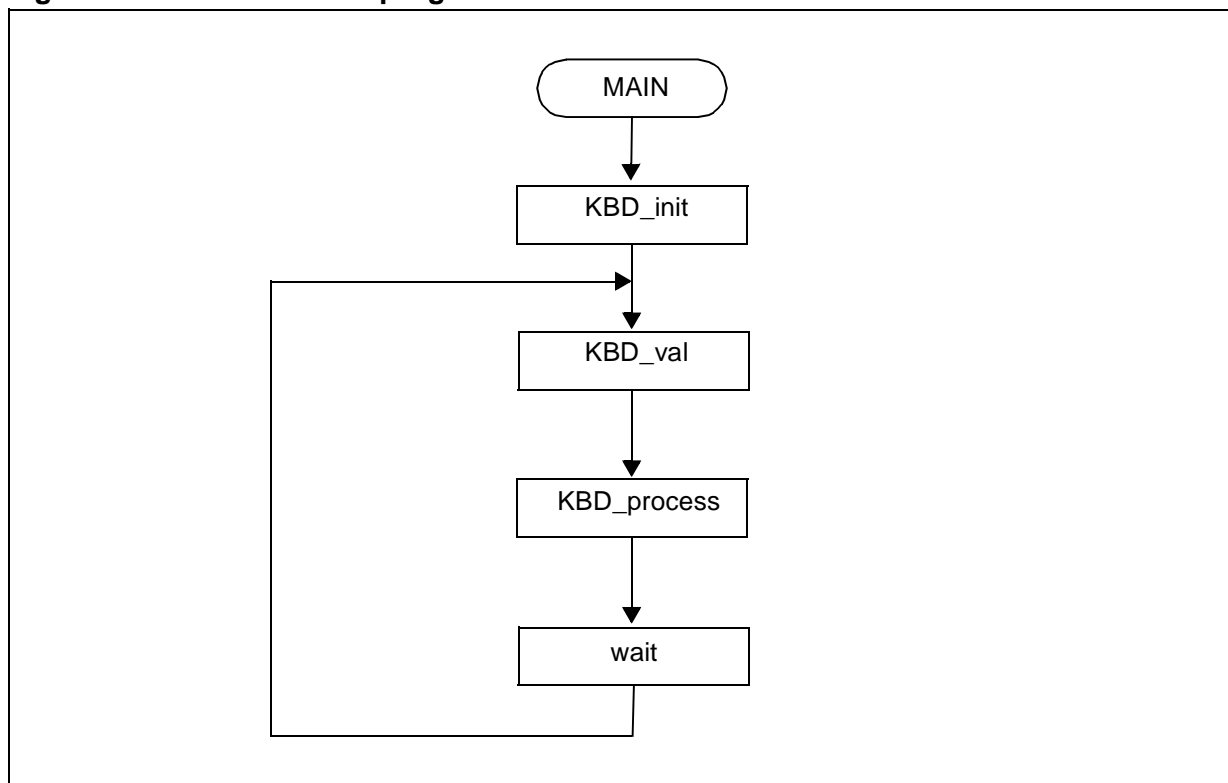


Figure 7. Flowchart: key detection

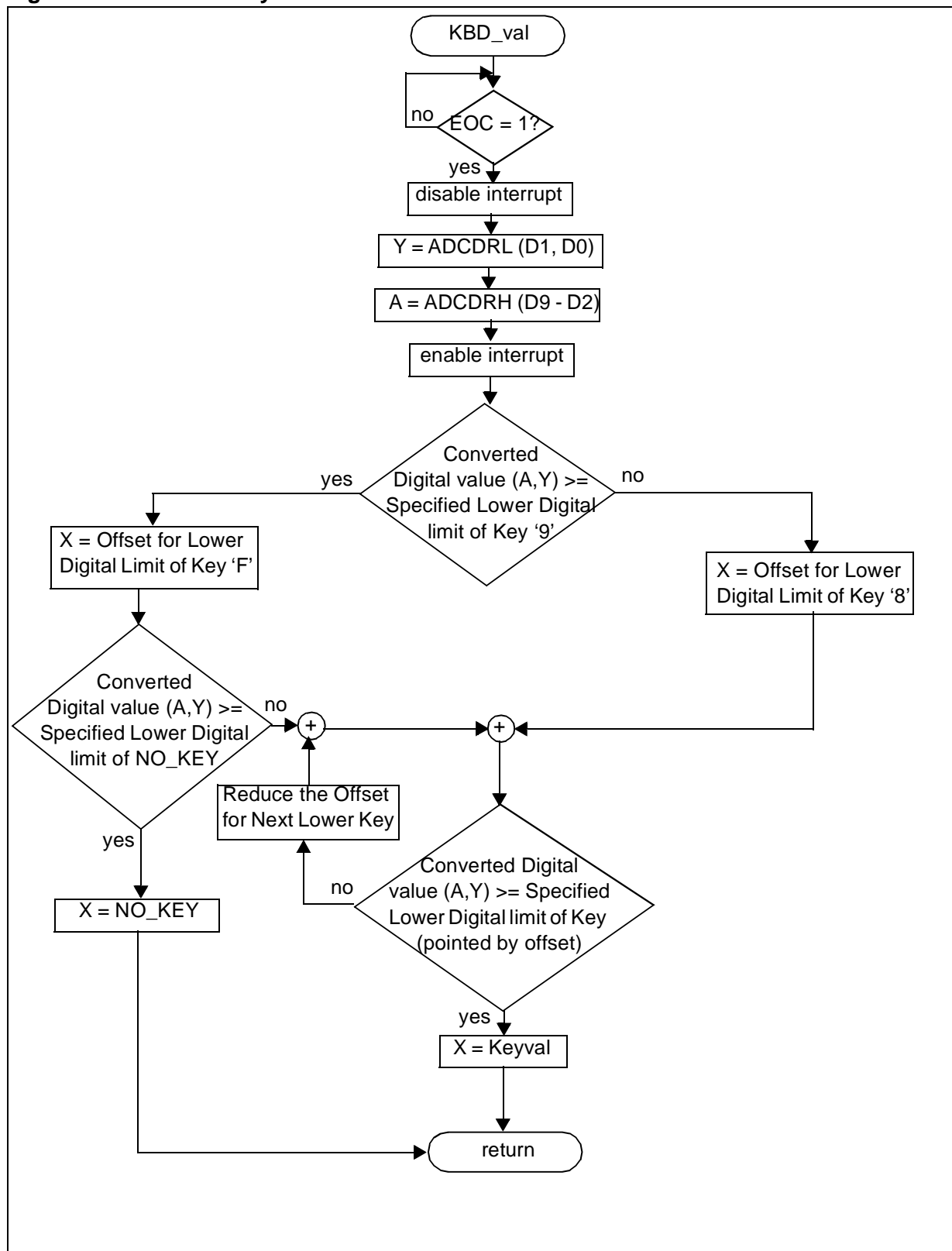
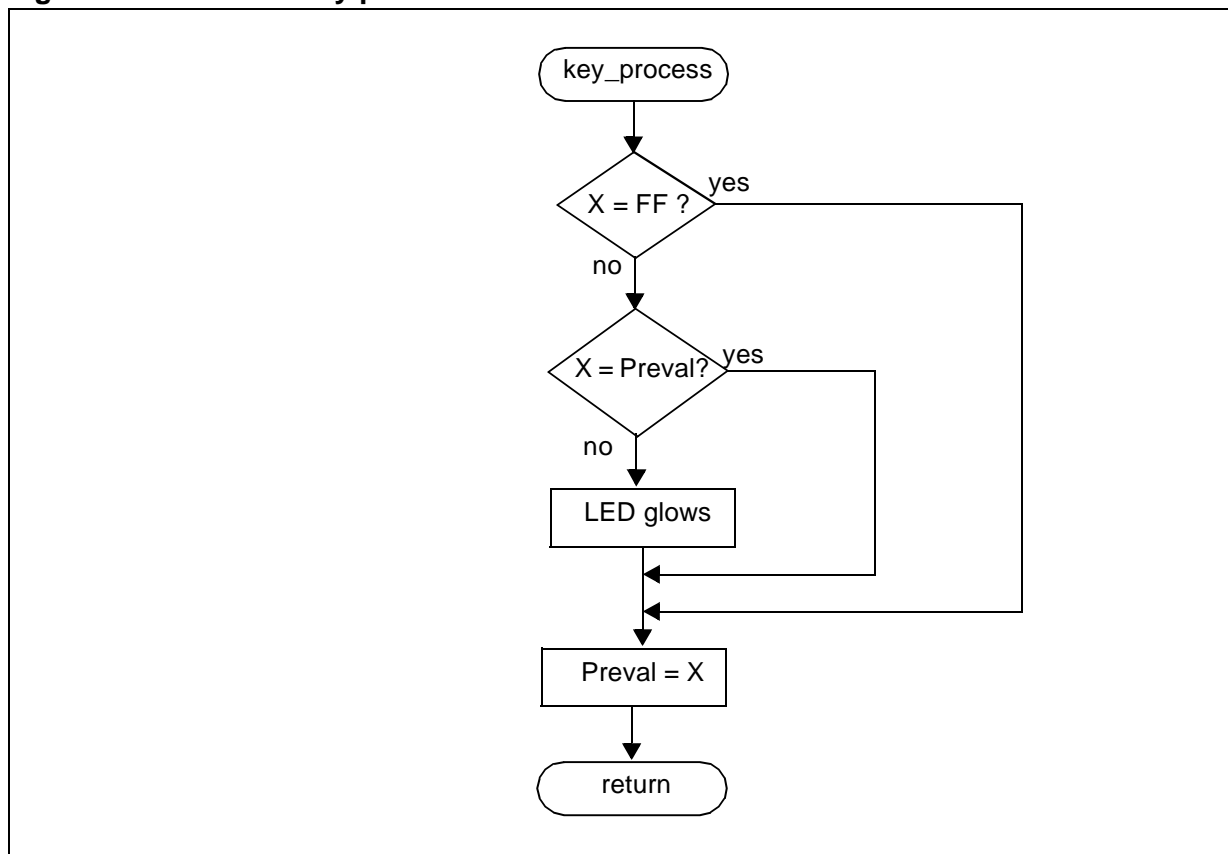


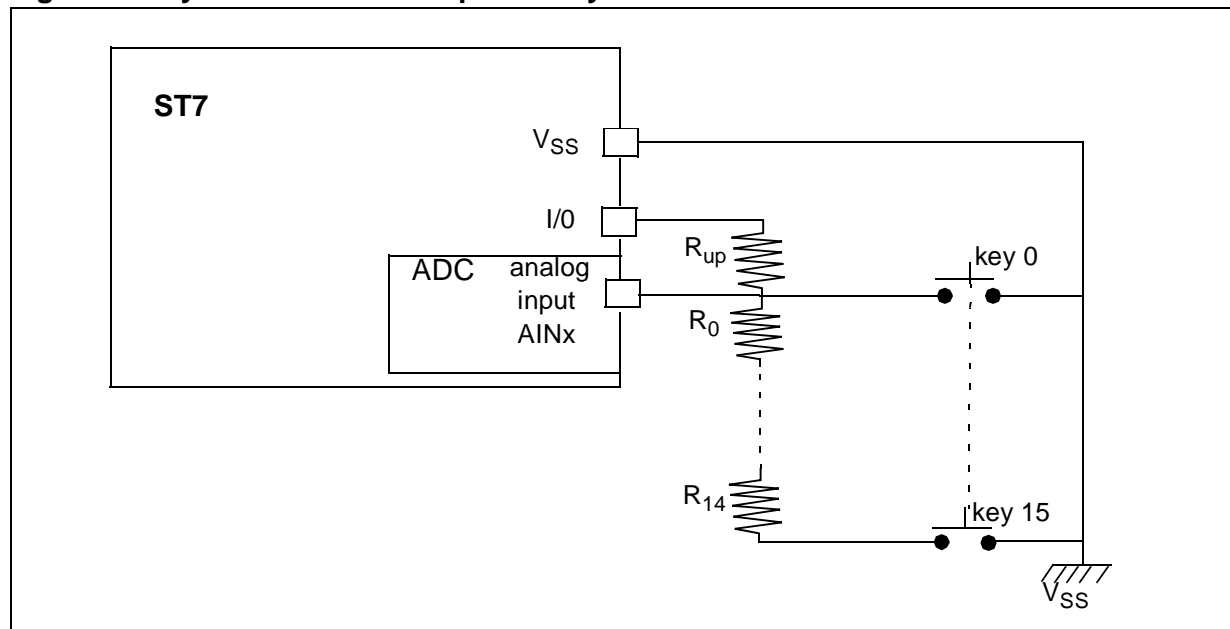
Figure 8. Flowchart: key process



## 6.3 EXTENSION FOR WAKE UP

With the ST7, it is possible to generate a wake-up function. This can be achieved by a modification of the circuit. The pull-up resistor must be connected to an additional port pin. During key polling, this pin is in output mode and active high, thus switching  $V_{DD}$  to the pull-up resistor.

**Figure 9. Keyboard with wake-up circuitry.**



During the wait for a key to be pressed, the I/O pin used for the pull-up is switched into a high impedance state (e.g. open drain output mode). The pin used as the ADC input while polling is switched to interrupt input with pull-up mode.

So if any key is pressed an interrupt will be generated if the voltage at this pin is below the Schmitt trigger low level threshold. The serial resistors in the keyboard chain must not be too high in this case, therefore the maximum number of keys is reduced in comparison to normal mode.

### 6.4 SOFTWARE

All the source files in assembly code as well as in C code are given in the zip file with this application note. The C code uses software library (ST7 software library version 1.1).

The source files are for guidance only. STMicroelectronics shall not be held liable for any direct, indirect or consequential damages with respect to any claims arising from use of this software.

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 the express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics

©2003 STMicroelectronics - All Rights Reserved.

Purchase of I<sup>2</sup>C Components by STMicroelectronics conveys a license under the Philips I<sup>2</sup>C Patent. Rights to use these components in an I<sup>2</sup>C system is granted provided that the system conforms to the I<sup>2</sup>C Standard Specification as defined by Philips.

STMicroelectronics Group of Companies

Australia - Brazil - Canada - China - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan  
Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

<http://www.st.com>