CONSIDERATIONS REGARDING THE PARASITE SWITCHING PHENOMENON AND METHODS OF PROTECTING THE APPLICATIONS WITH A MICROCONTROLLER

O. STAN[1], D. MIHOC[2]

Articolul tratează modul de conectare a butoanelor şi a matricilor de butoane la porturile de intrare/ieşire ale unui microcontroler flash din familia PIC produs de Microchip, unde sunt analizate fenomenele comutaţiei parazite la tranziţia on/off a acestor tipuri de dispozitive.

Sunt prezentate unele metode de protecţie software împotriva influenţei negative pe care le au aceste fenomene asupra activităţii microcontrolerului, deci implicit asupra rulării corecte şi fără întrerupere a programului implementat în memoria microcontrolerului.

În opinia autorilor reprezintă contribuţie originală, introducerea metodei hardware de diminuare a efectului, prin montarea unui condensator între masă şi intrarea microcontrolerului, combinat cu utilizarea unor trasee de cablaj pe distanţe cât mai scurte între buton şi intrările microcontrolerului.

The article dwells upon the connecting mode of the buttons and the key pad at the input/output side of a flash microcontroller from the PIC family made by the Microchip, where the phenomena of parasite switching at the on/off transition of this type of devices are analised.

Some methods of software protection against the bad influence of these phenomena on the microcontroller's activity and implicitly, on the accurate and uninterrupted running of the program implemented in the microcontroller's memory are presented.

In the author's opinion represent an original contribution, the hardware method which limits the effect, through the assembling of a capacitor between the board and the input pin of the microcontroller, using traces at short distances by connecting a button to the inputs pin of the microcontroller.

Key words: Button, Pull-up, Pull-down, Key Pad, switch contact debounce.

[1] PhD Student, Dept. of Automation and Computers, University "Politehnica" of Bucharest, Romania, E-mail: ovidiu.stan@merck.ro
[2] Prof., Dept. of Automation and Computers, University "Politehnica" of Bucharest, Romania

1. Introduction

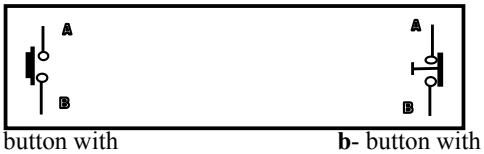
So far, the studies performed on different types of switches and keys at the on/off transition show that the reponse time depends on the type and dimension of the switches, on the switching speed of the devices, on the imperfect mechanical contacts and, especially, on the values of the currents and switching voltage.

As a consequence of the previously mentioned causes, a **„switch contact debounce"** phenomenon can appear – parasite switching and in case of using „DIP" (Dual Inline Pin) or Push-Pull buttons at the input or output pins of a microcontroller a protection against the disturbing influence of the false switching phenomenon is necessary by using software or hardware methods. [1]

This phenomenon can be abstractly considered as a short period of time in which oscillations on the contacts of the buttons in the transition between Hi - Low state and reverse appear. A comparative study of the microcontroller applications (regarded as microsystems) can lead to a simple analogy between the negative influence over the microcontroller's behaviour or the processing program and the effect of the perturbations of the industrial equipments on the automatic control systems (as macrosystems). In both situations, the priority to keep on researching in order to find solutions to satisfy at least, for the moment, the need of immunity at the any-type and any-level influences still remains.

2. Types of keys used in the applications with a microcontroller

The keys or the key pad are mechanical devices of different sizes and forms used to disconnect or connect two points of a circuit their selection depending on the purpose of the application and on the place where these operations of the developing application must be done. Usually, the keys are DIP (Dual Inline Pin), the Push-key or key pad types. The contacts of the keys can be NO (normal opened) or NC (normal closed), depending on their function.



a- button with normal opened contact

b- button with normal closed contact

Fig.1 a, b - Types of contacts [2]

In Fig.1.a when pressing the NO (normal open) contact key points A and B of the circuit will connect.

In Fig. 1.b when pressing the NC (normal closed) contact key points A and B are disconnected.

3. The connection mode of the keys to the microcontroller

The keys can be connected to a microcontroller in two ways:

a. With a resistor connected at Vdd (supply voltage) and the input pin of the microcontroller: Pull – up mode.

b. With a resistor connected to the ground (GND) and the input pin of the microcontroller: Pull-down mode.

a. With a pull-up resistor

In this situation, the key is connected to the GND and the input pin of the microcontroller, and the resistor is connected to the power supply voltage to realize the polarization current of the input – (see Fig. 2)

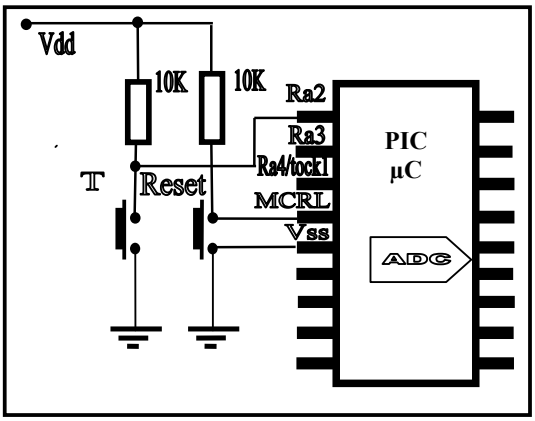


Fig. 2 - Button with a pull-up resistor [2]

This is necessary for the protection at the digital signal glitch, which represents a distortion or the appearance of a very short time impulse on the digital signal High state mode, which can be considered to be an unidentified and uncertain state having a reverse logical direction compared with the useful signal and which can create logical errors in the routine of the program – (see Fig. 3)

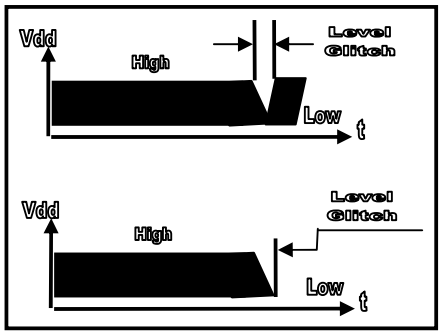


Fig. 3 - Level glitch [1]

b. With a pull-down resistor

In this situation, the key is connected to the power supply voltage and to the input pin of the microcontroller while the resistor is connected to the ground (see Fig. 4).

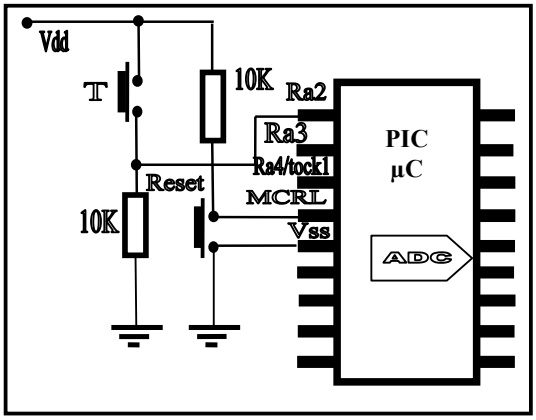


Fig. 4 - Button with a pull-down resistor [2]

4. Transition time at on/off switching

The transition time Δt depends the on key type and represents the necessary time for the signal to become steady after switching it from a High to a Low state.

It can be measured in the laboratory using a digiscope or osciloscope with memory. After repetated measurements, it can be quantified at 2-8 ms.

In Fig. 5 it is shown the transition time from a High state (corresponding to the reading time of the key's state by the microcontroller) to a Low state

(corresponding to the acceptance time for reading the key's state) and the reverse, from a Low to a High state.

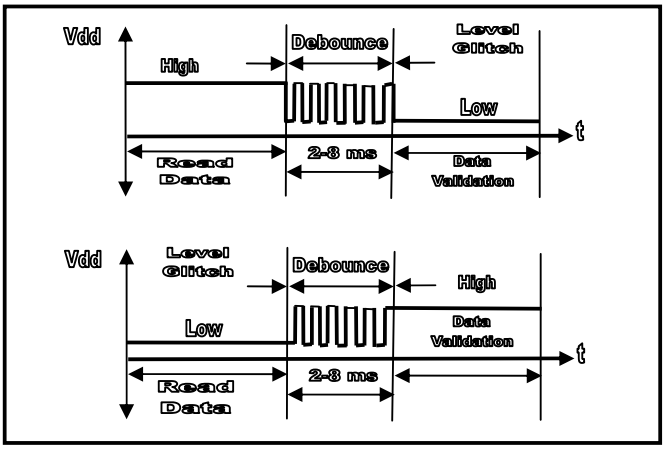


Fig. 5 - Transition time at the on/off switching and back [1]

5. Protection software modalities for a false detection switching

The method consists in a least two times reading procedure of the key's state through a subroutine program (macro) enclosed in the main program which uses a software **delay time** (dT) the fundamental condition being to ensure enough time (plenty of time) for the implementation of the operation:

$$\mathbf{dT > \Delta t \quad or} \tag{1}$$

$$\mathbf{dT > 10\ ms} \tag{2}$$

When detecting a closed contact the microcontroller implements a subroutine program from inside of the main program appendage while the on/off transition of the key is ended and that one key push represents a single impulse.

One example of subroutine (**keypress debounce**) able to supervise the push and the release of a used key in order to solve the problems of parasite switching can have the following form: [1]

```
Button macro HiLo, Port, Bit, Delay, Address
        Local   Exit            ; Local labels
        Local Loop
        If HiLo = = 0           ; Is the key pressed?
        Btscf Port, Bit         ; Is input line LOW?
        Else
        Btf ss Port, Bit        ; Is input line HIGH?
        End if
        Go to Exit              ; If key has not been pressed, exit the macro
        Wait Delay              ; Delay for key debounce
Loop
        If HiLo = = 0
        Btf ss Port, Bit        ; Is the key released?
        Else
        Bt fsc Port, Bit
        End if
        Go to Loop
Wait Delay              ; Delay for key debounce
Call Address            ; Call the service subprogram
Exit                    ; Exit the macro
        Endm            ; End the macro
```

6. Explanations:

HiLo – can be „0" or „1" logic and represents the increasing or decreasing front where the subroutines can be processed when a key is pressed.

Port – is a port of the microcontroller where the key must be connected (in the case of the PIC 16F84 microcontroller it can be PORT A or PORT B)
Bit – is a bit of the port and the taste is connected to it.

Delay – is a number from 0 to 255 used to give the time required to detect key debounce (contact oscillation) – to stop. It is calculated as follows:

TIME = Delay x 2 to 10 ms.

Address – is the address of the subroutine which does the necessary instrunctions for the pressing of a key after detecting an event (an accident) coming from the keyboard.

7. Protection hardware modalities for a false detection switching

The method consists in connecting a capacitor of the 100 pF value with the input pin of the microcontroller to the ground (for memories of this state for a few microseconds) and using the length short trace by connecting a button closely to the inputs pin of the microcontroller.

Likewise, the method offers a good EMI (Electromotor forces Influences) protection because it represents a very good help for the internal ADC converter resolution calculated for the value reading the Hi - Lo transition and reverse. (see Fig. 6)

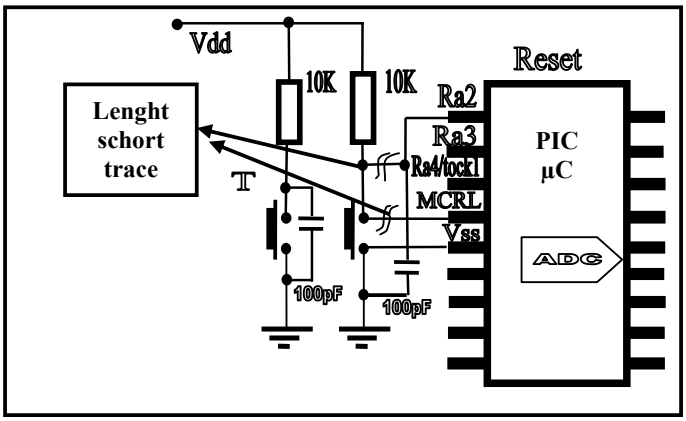


Fig. 6 - The capacitor is connected in parallel mode from button or between input pin of the microcontroller to the ground

8. Conclusions

1. In comparison with the software method in the hardware method these problems can be partly solved by connecting a converter to a key but, a welldone subroutine enclosed in the user program offers better results. The program can be adjusted until the false detection is completely eliminated.

2. In some situations a simple delay can be enough but if you want to solve more simultanous tasks a simple delay will force the microcontroller to be stationed for a long time and so, other operations could be failed.(ex. data inputs or the output port can be disconnected to a display)

3. A dynamic reading problem of the key can appear: if there is not enough reading time between the two reading operations and a single press procedure of the key takes more than 10 ms it will look like a succesive pushing procedure (false detection)and, in fact, it will be considered two or more repeated pressing operations.

If these issues are not anticipated at the conception of the program an error can appear or the microcontroller can produce more than one impulse at the output by a single pressing operation of the key.

R E F E R E N C E S

[1]*Wouter Van Ooijen, V. Surducan,* - Microcontrollers for all - Risoprint Cluj Napoca /2003
[2] Mikroelectronika, - Basic for PIC Microcontrollers/2003 - www. Mikroelectronika. co.yo
[3] *Cristian .Secrieru,* - PIC's Microcontrollers /2002 - www.Cristian Secrieru @.co.yo