DIGITAL DESIGN OPTIMIZATION FOR BIDIRECTIONAL PORTS OF THE MICROCONTROLLERS

Alin Dan POTORAC
Faculty of Electrical Engineering, "Stefan cel Mare" University of Suceava, Romania
alinp@eed.usv.ro, http://alin.usv.ro

Abstract: Using in an efficient way the port lines of a microcontroller means to expand the usual interfacing possibilities, with no costs increasing, by means of some special techniques. As part of a larger study deployed by the author, the paper is describing some methods for efficient use of microcontrollers' bidirectional ports lines when interfacing with keypads. Some new interface solutions were proposed in order to obtain a larger number of managed keys with minimum requirements like extra devices or microcontrollers' resources. Based on these hardware solutions specific algorithms were developed and here described. Also, mathematical functions were generated to evaluate and compare the results.

Key words: *bidirectional port lines, microcontrollers, interfacing, keypad, optimization.*

I. INTRODUCTION

Developing human interfaces for digital systems is a modern problem related with increasing demands for digital equipments on the market. Most of the systems are using keypads interfaces and the needed number of buttons is larger every day. In our days there is a challenge to connect a large number of keys using a reasonable number of microcontrollers' ports lines and as few as possible adds-on. PCs keypads are an example. The final scope is not only to reduce the lines number but to find cheaper solutions and best compromises between adding new or powerful devices or implementing alternative solutions which are efficiently using, as much as possible, the hardware futures like bidirectional port lines, A/D inputs, comparator inputs, PWM outputs.

The paper is focusing on some solutions and case studies tested or implemented by the author. An extensive study was deployed but here, due to limited space, only few situations will be considered, namely the bidirectional port lines case of microcontrollers and ways of interaction with keypads. The original interfacing methods are evaluated based on the possible number of keys and port lines number.

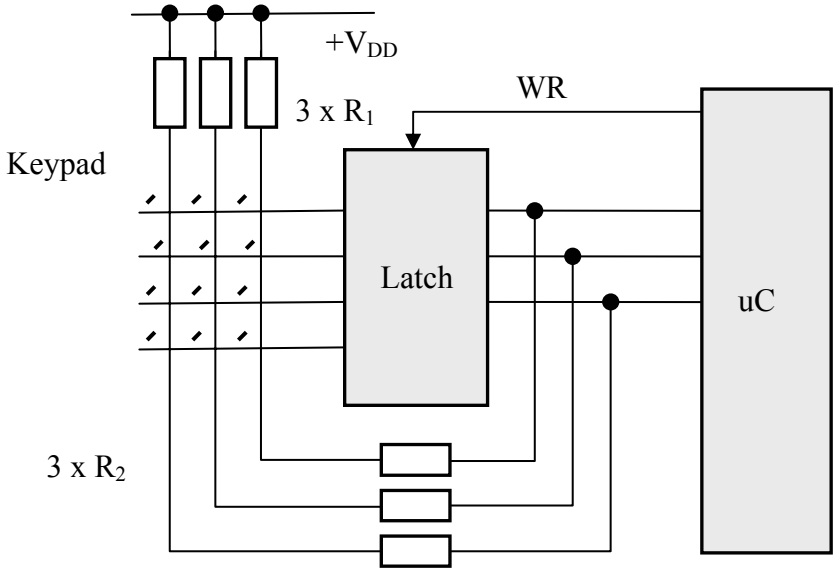

Figure 1. Keypad interface optimized for bidirectional lines port

The maximum number of the controlled keys is

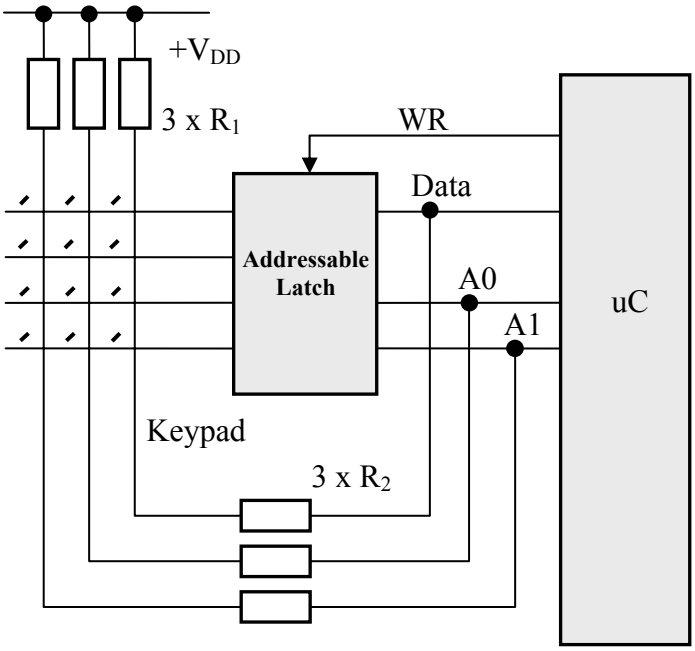

Figure 2. Addressable latch keypad interface for bi-directional lines port

2. USING LATCHES TO MINIMIZE PORT LINES NUMBER FOR BIDIRECTIONAL PORTS

The new microcontrollers' generations are including bidirectional port lines, [4], [5]. This facility offers the possibility of exploiting the two way communication of each line, for example by writing out data into a latch register and than reading the latch outputs via the keypad matrix (figure 1).

Once the latch is up-loaded with the binary word having only one active line ("0" for figure 1, but is depending on the algorithm and the way of connecting the resistors network), the port lines directions are reversed. The data is read from the column lines. If no key is pressed, "1111" is returned, if a key is pressed, the corresponding line goes to zero.

The R_2 resistors are necessary to avoid the latch outputs feedback to the inputs while the load process is occurring. They introduce as much impedance value as necessary to assure the logical priorities for the logical outputs of the microcontroller port. R_1 resistors are pull-up type and can be omitted if internal microcontroller pull-up resistors are available.

Pressing two keys at the time is only possible on the same line if write-to-latch and read processes are speed enough. Otherwise two opposite outputs are short-connected and protection resistors are necessary. The problem can be solved connecting series diodes on each latch output. Diodes must be on only for active line (if active "low", the diodes cathodes have to be connected to the latch output).

$$B_L = (n-1)^2 \qquad (1)$$

We can observe that for 5 bidirectional lines port 16 keys are available.

The efficiency of this principle can be significant improved if an addressable latch is used instead of the direct one. Figure 2 is shown this solution. This time the maximum number of controlled keys is significantly increased as the value is

$$B_{AL} = 2^{n-2} \cdot (n-1) \qquad (2)$$

That is because 2 output lines are reserved for WR signal and DATA signal, so only *n-2* are available as output while, when lines are working like inputs, only WR line has to be left out to avoid overwriting (by WR accidentally activation from the keypad).

Let's observe that for the same 5 lines like before the maximum number of controlled keys is 32.

Another solution based on one bidirectional port line and one output line is presented in figure 3. The principle consists in using a shift-register, ring counter type, [3], [1], which here is initialized by the rising edge of enable signal, E. This is done by using a passive derivation circuit built with C_d and R_d. D is used to send to the ground the negative pulse resulting from derivation process (that one corresponding to the falling edge of the validation signal E). When E signal is remaining in "0" or "1" state, with no

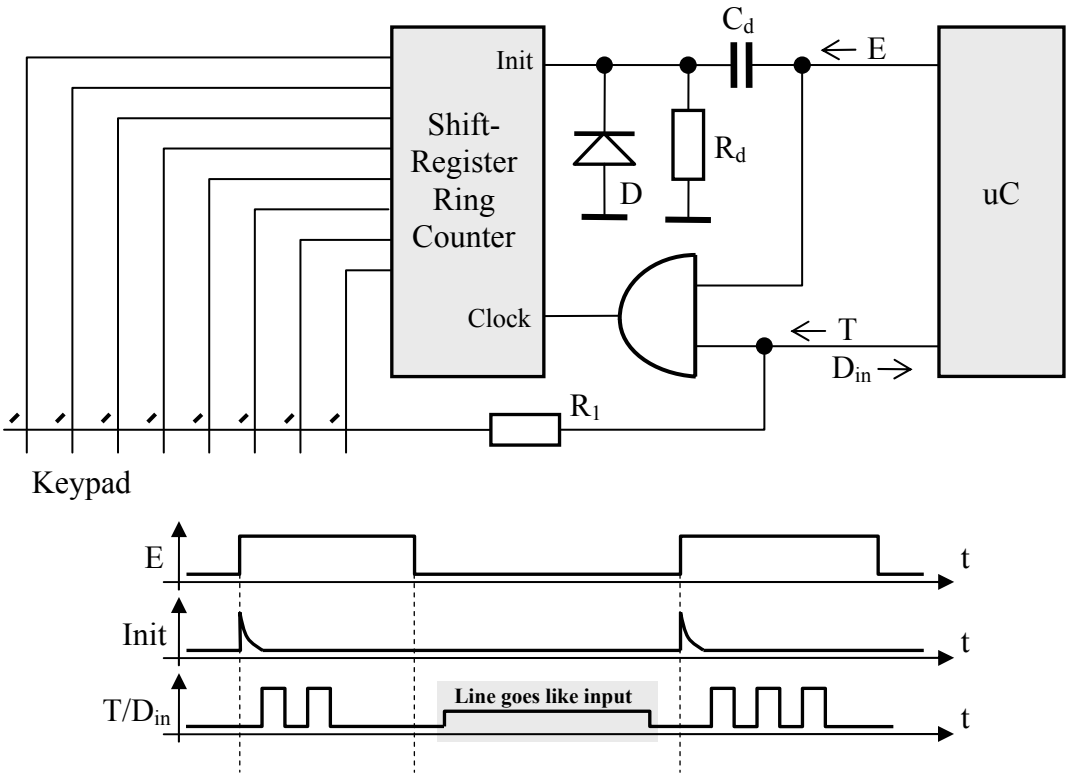

Figure 3. Optimized keypad interface with shift register and bidirectional port lines

transitions, no pulses appear on *Init* input of the ring shift register. In the same time E signal is driving an AND gate able to validate clock pulses T passing to the register. So, with E and T signals the shift register can be driven to any available state, particularly in a state having only one output active (supposing that state to be "1" and all other in "0"). In the next step the former clock line T has to be reversed, as input for the data coming from the keypad. Validation line E has to remain in "0" during reading procedure to maintain the shift register clock signal to zero. The algorithm continues moving the active shift register output, one by one, to another position and reading again and so on.

The R_1 resistor is very important since it assure that a key pressed during the register state changing (E=1) is not acting like a clock pulse generator and replacing the state with an unknown one. If a key is pressed during this phase the

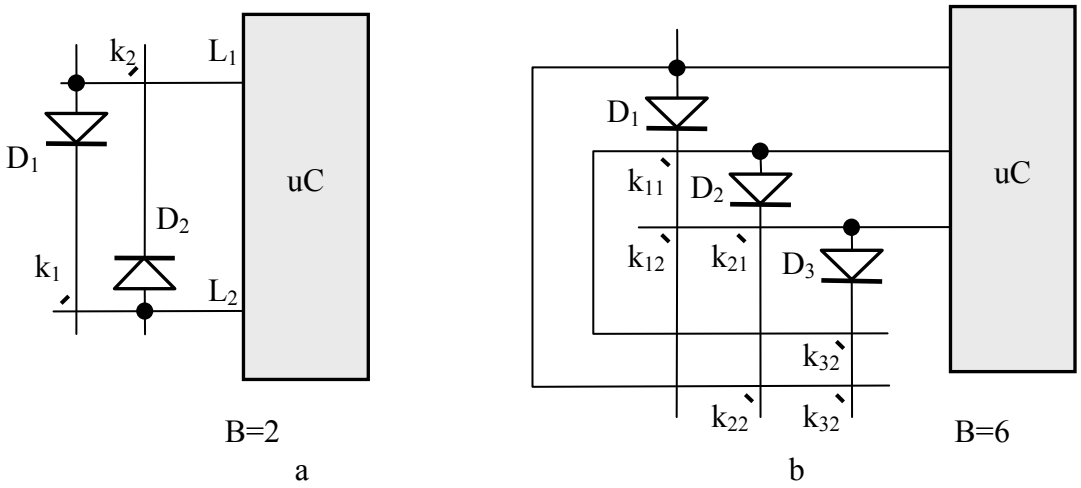

Figure 4. Elementary diodes matrix for bidirectional port lines (one diode per line)

logical output T is imposing its state whatever the keypad line state is.

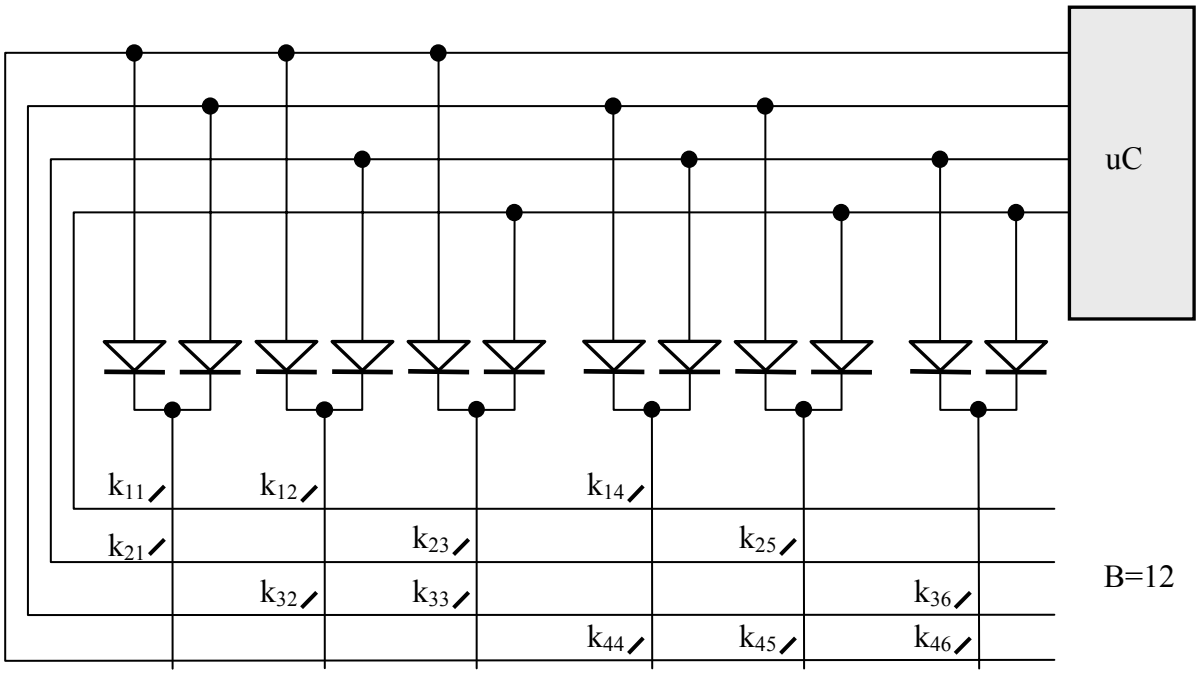

Figure 5. Diodes matrix for bidirectional port lines (two diodes per line)

Two keys can't be pressed in the same since this way two outputs are short-circuit connected even the reading process is a sequential one. Using series diodes connected on each register output the problem can be easy override.

3. OPTIMIZED SOLUTION FOR KEYPAD INTERFACE WITH SWITCHED DIODE MATRIX AND BIDIRECTIONAL LINES

Based on the fact that bidirectional lines are available, there is a possibility of using this propriety to alternatively switch some diodes connected from one port line to another one, [2], (figure 4.a).

The idea is to declare one line like output (L_2, active low) and other one like input (L_1), in the first step. In the second step, the lines are reversed. We have to note that internal pull-up resistors of input lines, available on most microcontrollers, have to be used in order to assure the polarization current for corresponding diodes.

The classical matrix solution for 2 port lines corresponds to 1 button while now 2 buttons are available.

The advantages are more evident when more than 2 lines are allocated (3 port lines, figure 4.b). Every line is having one diode able to touch any from the other two lines via 2 keys, so we have (3-1) x 3 = 6 keys.

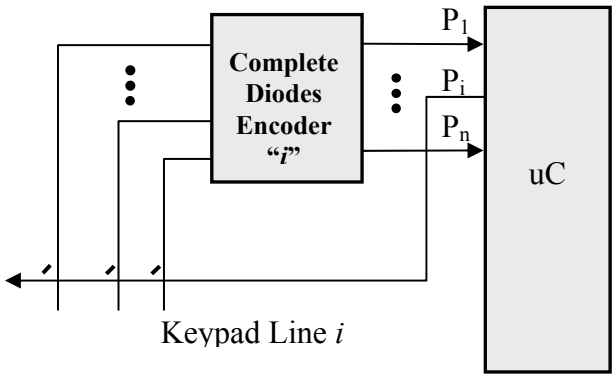

Figure 6. Keypad interface principle for diodes matrix, for one output line, i

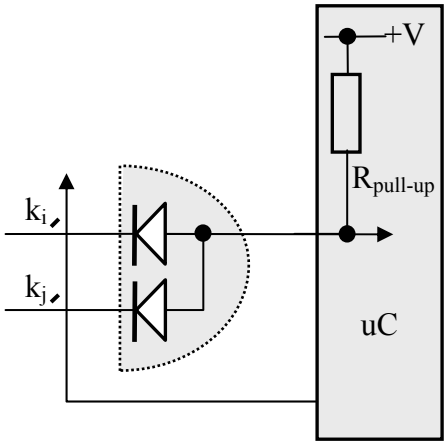

Figure 7. Diodes AND gates used in keypad encoder

Generalizing, for one diode per line, on can manage B_{1D} keys:

$$B_{1D} = (n-1)\cdot n = (n-1)\cdot C_n^1 \quad (3)$$

If two diodes per line are used is easy to observe that:

$$B_{2D} = (n-2)\cdot C_n^2 \quad (4)$$

or generally, for k diodes per line are used, than:

$$B_{kD} = (n-k)\cdot C_n^k \quad (5)$$

Figure 5 is shown the alternative diagram with 2 diodes per line and 4 port lines.

Of course, for n port lines any combination of *j = 1, 2 ... k* diodes per line can be used so that finally the maximum amount of the managed buttons is

$$\begin{aligned} B_D &= B_{1D} + B_{2D} + \ldots + B_{kD} = \\ &= C_n^1\cdot(n-1) + C_n^2\cdot(n-2) + \ldots + C_n^k\cdot(n-k) \\ B_D &= \sum_{j=1}^{k} C_n^j\cdot(n-j);\quad 1\le k\le n-1 \end{aligned} \quad (6)$$

For a given number of available bidirectional port lines assumed to be *n*, the maximum number of diodes is *n-1* because anytime we need to preserve at least one line to connect the other key terminal with. So, the maximum is to be obtaining as:

$$\begin{aligned} &\max(B_D) \xrightarrow{k=n-1} B_{D\max} \\ &B_{D\max} = \sum_{j=1}^{n-1} C_n^j\cdot(n-j) = \\ &= \sum_{j=1}^{n-1} \frac{n!}{(n-j)!\cdot j!}(n-j) = \\ &= \sum_{j=1}^{n-1} \frac{n!}{(n-j-1)!\cdot j!} \end{aligned} \quad (7)$$

The table 1 is presenting some comparative results for different number of port lines and different numbers of diodes. As we can see, like example, for 7 port lines and maximum 3 diodes per line the number of buttons is 287, a quite large amount.

The results are surprising because the coding efficiency is greater than expected. For example, with 3 port lines and maximum 2 diodes per line, according with the table and above formula, the number of controlled keys is 9 while usually 3 lines are allowing encoding $2^3 = 8$ keys (direct encoding). The result is more surprising for a larger number of port lines and diodes per line.

The explanation comes from reading procedure as shown bellow, related again with figure 5.

Table 1. Comparative results

Nr.	Interface Type	Number of port lines	Number of keys	Average number of keys per line
1.	Latch	3 5 7	4 16 36	1.33 3.20 5.14
2.	Addressable Latch	3 5 7	4 32 192	1.33 6.40 27.42
3.	Diodes, one per line	3 5 7	6 20 42	2.00 4.00 6.00
4.	Diodes, two per line	3 5 7	3 30 105	1.00 6.00 15.00
5.	Diodes, three per line	3 5 7	0 20 140	- 4.00 20.00
6.	Diodes – one, two and three per line	3 5 7	9 70* 287**	3.00 14.00 41.00

* the value is calculated for maximum 3 diodes per line; maximum is obtained with 4 diodes and it is 75.

** the value is calculated for maximum 3 diodes per line; maximum is obtained with 6 diodes and it is 441.

Considering *n* port lines, every time one has to be kept as output and the others *n-1* are programmed as inputs. On the inputs are available 2^{n-1} possibilities, each possibility being encoded with the right combination of diodes. So, the maximum number of combination is obtained when the encoder is processing every possibility and this should be calculated as C_{Dmax}:

$$C_{D\max} = n\cdot 2^{n-1} \quad (8)$$

For each output from *n* possible there is one input corresponding with no key pressed (having all input bits as "1"), so the maxim number of possible keys is

$$B_{D\max} = n\cdot 2^{n-1} - n = n\cdot(2^{n-1}-1) \quad (9)$$

Is easy to demonstrate that

$$n\ge 3 \quad \Rightarrow \quad n\cdot(2^{n-1}-1) \rangle\, 2^n, \quad (10)$$

This means that for any $n \ge 3$ the diodes matrix is more efficient for keypad interfacing even than direct encoding. This means that we have more keys than classic encoding possibilities for a given number of bits, available when reading directly the ports.

To manage this situation at the level of the coding binary word we have to take into account that the binary possible combinations (considering a number of port lines n = 3) is given by possible combination of lines status as is shown in

the table bellow.

Table 2.Lines states

0↑ 0 0	**0 0↑ 0**	**0 0 0↑**
0↑ 0 1	**0 0↑ 1**	**0 1 0↑**
0↑ 1 0	**1 0↑ 0**	**1 0 0↑**
0↑ 1 1	1 0↑ 1	1 1 0↑

where the symbol 0↑ means a "low" output while others are usual inputs (as "seen" by the microcontroller).

This way each output has now three states: "0", "1" and "0↑". These three states could encode 3^3 states (figure 4.b), generally n^3 states, but some values are not permitted by the algorithm because at least one bit has to be in "0↑" state every time; also, some combination are associated with "no pressed button" (last line in the table above - all bits in "1" except one who has to be "0↑"). More, first line from the table corresponds with two keys pressed in the same time, any 2 keys combination (for 3 lines) being identified as one of the following: "0↑ 0 0", "0 0↑ 0", "0 0 0↑". The situations correspond to the last line from the table.

As shown, if two keys are pressed in the same time the system will return a different code than normal for each individual key. In figure 4.b for example, if k_{11} is pressed the code will be "0 0↑ 1". For k_{12} the code is "0 1 0↑". For k_{11} and k_{12} pressed in the same time the sequentially generated codes are "0 0↑ 0" when reading k_{11} or "0 0 0↑" when reading k_{12}, combinations which are quite different that we are expecting. If only one diode per line is used the situation can be managed considering that only one "0" is allowed in every words, any other situation being reported as error. If more than one diode per line is used, pressing two buttons simultaneously could generate the code for a third button that means an unmanageable situation.

Basically, the switched diodes matrix configured as described above is a complete encoder, see figure 6, built with AND gates, each gate being implemented at his turn with diodes (figure 7), [1].

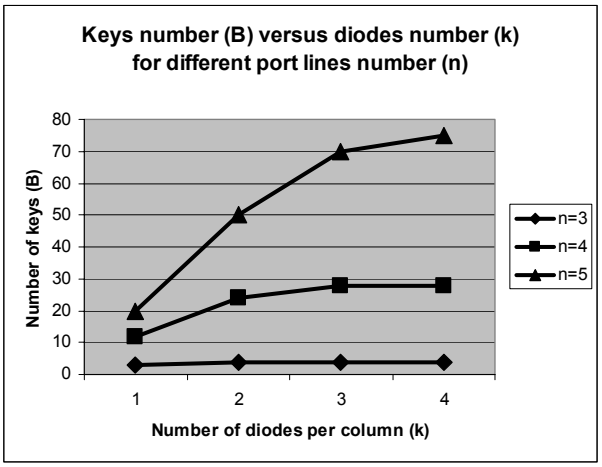

Figure 8. Efficiency of diodes switched matrix method

Graphical representation of dependences between the number of controlled keys (B) and the number of diodes per line (k) for different number of port lines (n = 2, 3, 4) is shown in figure 8. The relevant increase is for $2 \le k \le n$-2.

This method, based on the switched diodes matrix, is far away the most efficient and cost effective one. Hundreds of keys can be managed by the only means of adding some tens of cheap devices like diodes are.

If adding a communication port to the system a remote large keypad can be controlled.

IV. CONCLUSIONS

Starting with classical matrix keypad interface used with unidirectional ports, some improvements were proposed in order to obtain a better efficiency at the level of the compromise costs – available number of interface keys. Even based on usual unidirectional port lines interface the possible keys number could be increase by adding cheap additional devices instead of using powerful and more expensive microcontrollers. If bidirectional port lines are available, as shown above, a much better efficiency can be obtained by exploiting the programmable two ways of data directions. A category of methods consist in using latches, addressable latches or shift registers to periodically "freeze" the lines logical status and than, reversing the data directions, to read the new state induced by the matrix keys. Another category of methods is based on a particularly switched diodes matrix as results of implementing a diode based logical encoder. Both categories lead to better solutions compared with classical ones. Higher efficiency could be obtained only if multilevel, analogical type, port lines are available.

Almost all above solution were tested by the author like stand-alone systems or included in complex digital systems. The implementations were very successful in spite of the elaborated software algorithm which has to be developed in some cases. At the level of the associated software the programmer has to pay attention to the digital filtering of the data and to the technical way of programming and reprogramming the I/O ports. Polling technique is the most appropriate method but interrupts techniques can also be used with no extensive compromises.

Starting from these principles keypads with large number of keys can be easily designed. Depending on the digital system structure, many alternative solutions can also be available.

REFERENCES

[1] Potorac A.D., *The Basis of Digital Design*, Ed.Matrix, Bucuresti, 2002.

[2] Velchev V., *A keypad Controller for Bidirectional Key Matrix*, Microchip "Design for Dollars", 1997.

[3] Taub H., Schilling D., *Digital Integrated Electronics*, Ed.McGrawHill, 1977.

[4] Microchip Technical Library CD-ROM, 2002.

[5] www.microchip.com