Optimization techniques in interface design

Potorac, Alin Dan

December 2003

University of Suceava, Computer Department
9, University Street, RO-720225, SUCEAVA
alin@usv.ro

Abstract

The paper is presenting different optimization techniques for keyboard interfacing to microcontroller ports based on multifunction programmable proprieties associated with new generation of microcontrollers. Using bidirectional port lines and/or A-to-D inputs a quite large number of keys can be interfaced. Some original solutions are proposed and evaluated.
Keywords: *interfacing, keyboards, bidirectional ports, A-to-D inputs.*

1 Introduction

Having logical outputs and inputs lines available to the microcontroller ports, the classical way of efficiently interfacing with keypads is the matrix structure. Since almost all today microcontrollers have programmable logic inputs/outputs and many of them include analog input lines, then optimized interfaces could be designed based on commutation diodes matrix and circular alternative programming of the port lines.

2 Using port lines facilities to optimize keypad interfaces

2.1 Bidirectional port lines based solutions

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, [1], see figure 1.
The idea is to declare one line like output (L_2, active low) and other one like input (L_1), in the first step (figure 2.a). In the second step, the lines are reversed. We

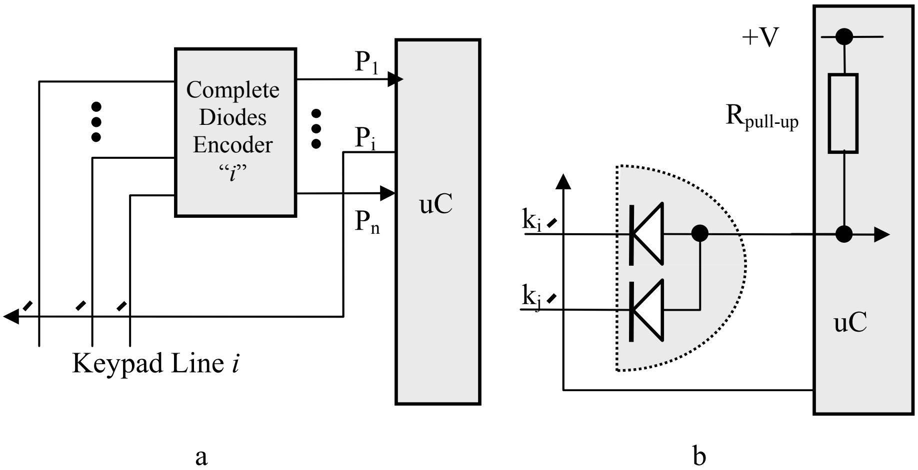


Figure 1: Keypad interface principle (a) for diodes matrix and diodes AND gate (b) used in keypad encoder

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 (for 3 port lines, see figure 2.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.
Generalizing, for one diode per line, on can manage B_{1D} keys:

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

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

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

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

$$B_{kD} = (n-k)\cdot C_n^k \tag{3}$$

Figure 3 is shown the alternative diagram with 2 diodes per line and 4 port lines [4]. For n port lines any combination of $j = 1, 2 \ldots k$ diodes per line can be used so that finally the maximum amount of the managed buttons is

$$B_D = B_{1D} + B_{2D} + \ldots + B_{kD} = C_n^1 \cdot (n-1) + C_n^2 \cdot (n-2) + \ldots + C_n^k \cdot (n-k)$$
$$B_D = \sum_{j=1}^{k} C_n^j \cdot (n-j); \quad 1 \le k \le n-1 \tag{4}$$

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 obtained as:

$$\max(B_D) \xrightarrow{k=n-1} B_{D\max}$$
$$B_{D\max} = \sum_{j=1}^{n-1} C_n^j \cdot (n-j) = \sum_{j=1}^{n-1} \frac{n!}{(n-j)! \cdot j!}(n-j) = \sum_{j=1}^{n-1} \frac{n!}{(n-j-1)! \cdot j!} \tag{5}$$

As an 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 above formulas, the number of controlled keys is 9 while usually 3 lines are allowing encoding $2^3 = 8$ keys (if direct complete encoding would be implemented somehow). 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 3.

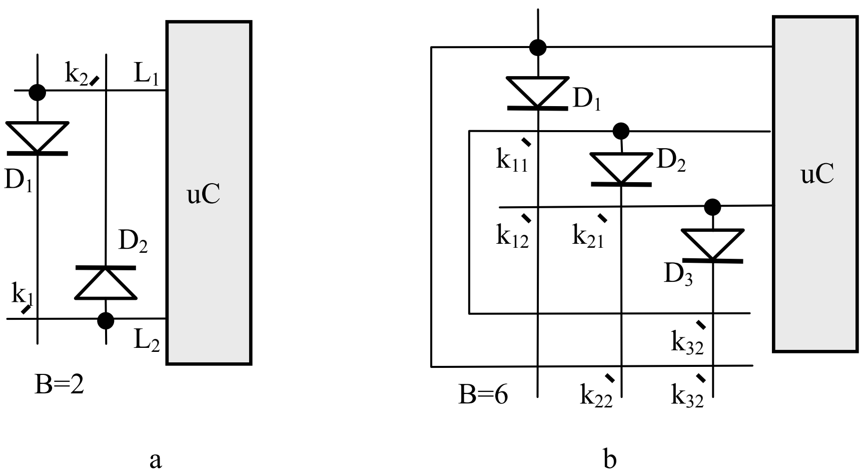


Figure 2: Elementary diodes matrix for bidirectional port lines (one diode per line)

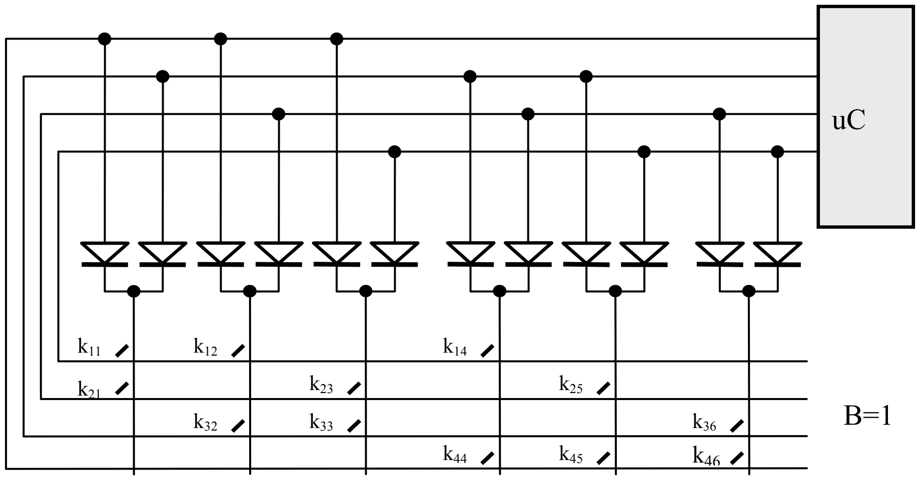


Figure 3: Diodes matrix for bidirectional port lines (two diodes per line)

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}:

$$C_{D\max} = n \cdot 2^{n-1} \tag{6}$$

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

$$B_{D\max} = n \cdot 2^{n-1} - n = n \cdot (2^{n-1} - 1) \tag{7}$$

Is easy to demonstrate that

$$n \geq 3 \quad \Rightarrow \quad n \cdot (2^{n-1} - 1) \rangle\, 2^n \tag{8}$$

This means that for any n ≥ 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.

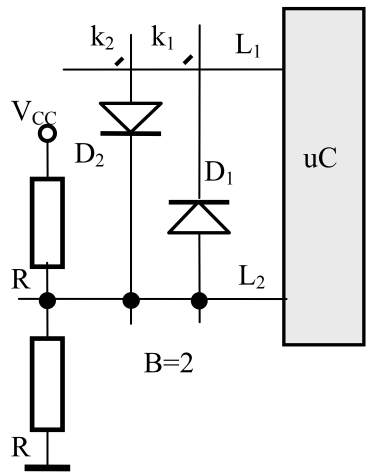
0↑ 0 0	0 0↑ 0	0 0 0↑
0↑ 0 1	0 0↑ 1	0 1 0↑
0↑ 1 0	1 0↑ 0	1 0 0↑
0↑ 1 1	1 0↑ 1	1 1 0↑

Table 2: Three lines port state

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, 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 usual situations correspond to the lines 2 and 3 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 2.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.



L1 (logical output)	L2 (analogical input)	Decision
0	Low	K1 on
1	High	K2 on
x	half	none

Figure 4: Elementary diodes matrix for logical output and analogical input port lines (one diodes pair per line)

Basically, the switched diodes matrix configured as described above is a complete encoder, see again figure 1, built with AND gates, each gate being implemented in turn with diodes, [2][4].
This method, based on the switched diodes matrix, is quite efficient and cost effective one if bidirectional programmed port lines are available. Hundreds of keys can be managed by the only means of adding some tens of low-cost devices such as diodes are.

2.2 Optimized solution based on multifunctional port lines

New generation of microcontrollers are including analogical inputs/outputs port lines [3] at very low cost. For this reason is quite possible to exploit these kinds of facilities since one port line could be programmed as logical type (output or input) or as analogical type (A-to-D or comparator input, PWM output).
The original idea here described is based on the use of port lines like logical output and A-to-D inputs. Developing some former research of the author [4][5] as described before, a modified diodes matrix can be used. The principle consists in using pairs of diodes instead of one piece like in above solution. Having analog inputs available, resistive dividers are necessary to define a voltage values, which are different from the values associated with logical "0" (GND) and logical "1" (VCC). This voltage value is proposed to be half of the voltage scale, VCC/2, using symmetrical resistors (figure 4). If VCC/2 is read by the analogical input

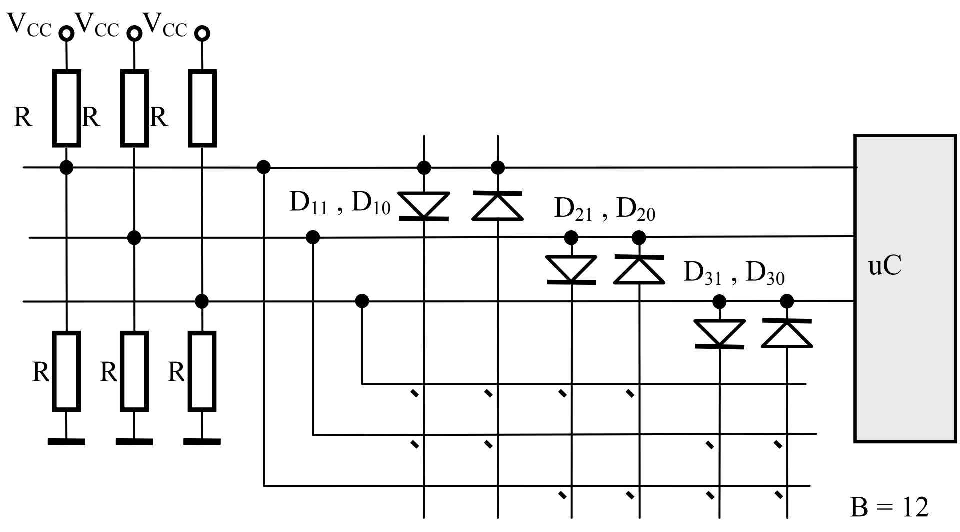


Figure 5: Diodes matrix for logical output/analogical input port lines (one pair of diodes per line)

supposed to be L2, than "no pressed key" decision is invoked. If a low voltage is detected on L2, this voltage is for sure coming from logical "0" on L1 and k1 key pressed (switched on). If a high voltage is detected on L2, than this voltage can come from a logical "1" on L1 and k2 key pressed. This judgment is based on the diodes state when a lower or higher voltage ("0" or"1") is supplied on one terminal compared with half of the scale voltage on the other terminal.
Because any circular permutations of the line functions are possible by programming, the solution can be expanded for three lines (figure 5) or more. Is easy to observe that every solution proposed for using diodes matrix with bidirectional port lines could be implemented this time by replacing every diode with pairs of diodes.
The big advantage is the possibility of doubling the number of keypad keys. Since was already shown before that for k diodes the number of keys is (3), than now we have a larger number of keys:

$$B_{kDpair} = 2 \cdot (n-k) \cdot C_n^k = 2 \cdot B_{kD} \tag{9}$$

All key number above evaluation could be now applied if multiplied by 2. So, because any combination of 1, 2 ore more (generally k) pairs of diodes are possible for each port line, than

$$B_{Dp} = B_{1Dp} + B_{2Dp} + \ldots + B_{kDp} = 2 \cdot [C_n^1 \cdot (n-1) + C_n^2 \cdot (n-2) + \ldots + C_n^k \cdot (n-k)]$$

$$B_{Dp} = 2 \cdot \sum_{j=1}^{k} C_n^j \cdot (n-j); \quad 1 \le k \le n-1 \tag{10}$$

Based on these conclusions we could observe that, for 7 port lines and maximum 3 pairs of diodes per line (similar number of lines like in example above, see 2.1) the number of buttons is 574, a very large amount. The number of switched diodes is also larger and can be calculated as

$$N = n + 2 \cdot n + \quad \cdots \quad + i \cdot n = n \cdot \sum i \tag{11}$$

where *n* is the number of used port lines and *i* is the maximum number of diodes (solution 2.1) or pairs of diodes (when *i* becomes *2i*, solution 2.2) per line. For considered example we need 42 diodes (solution 2.1) or 84 diodes (solution 2.2).

3 Conclusions

Some improvements were proposed in order to obtain a better efficiency at the level of the compromise costs – available number of interface keys in microcontroller based systems. Even using usual unidirectional port lines interface the possible keys number could be increased 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 directions of data. A classical 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 [5]. Another category of methods is based on a particularly switched diodes matrix as results of implementing a diode based logical encoder. The efficiency can be increased more (twice as in example above) if the port lines have analog-to-digital conversion facilities (or window comparator programmable inputs). All categories lead to better solutions compared with simple matrix ones. Based on these conclusions, if multilevel, analogical type port lines are available, higher efficiency could be theoretically obtained if other threshold devices are used (Zener diodes or multilevel voltage references, for example).

Since no relevant bibliography is available, the author designed, optimized (as described above) and tested almost all proposed methods 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. This is mainly related to the conversion of the three-state logic to binary (see table 2). 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.

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 further developed.

References

[1] Velchev V. (1997) “A keypad Controller for Bidirectional Key Matrix”, Microchip “Design for Dollars”.

[2] Taub H., Schilling D. (1977) Digital Integrated Electronics, Ed.McGrawHill.

[3] Microchip Technical Library CD-ROM (2002).

[4] Potorac Alin Dan (2003) “Digital Design Optimization for Bidirectional Ports of the Microcontrolers”, Acta Electrotehnica, ISSN 1224-2497, Vol.44, Nr. 1, Ed. Mediamira Cluj-Napoca, Romania, p. 46-52.

[5] Potorac Alin Dan (2003) “Solutions to Minimize the Port Lines Number for Keypad Interfaces to Microcontrolers” - SIELA 2003 – XIII-th International Symposium on Electrical Apparatus and Technologies, ISBN 954-90209-2-4, Vol.1, Plovdiv, Bulgaria, p.182-187.