Semiconductor Physics, Quantum Electronics & Optoelectronics, 2020. V. 23, No 3. P. 329-335.

Optoelectronics and optoelectronic devices

Software controlling the LED bar graph displays

A.V. Bushma[1], A.V. Turukalo[2]

[1]Borys Grinchenko Kyiv University, 18/2, Bulvarno-Kudriavska str., 04053 Kyiv, Ukraine
E-mail: o.bushma@kubg.edu.ua
[2]National University of Life and Environmental Sciences of Ukraine,
15, Heroiv Oborony, 03041 Kyiv, Ukraine
E-mail: tyrykalo@gmail.com

Abstract. This work has been devoted to implementation of the software aimed at bi-cycle information models for LED bar graph displays. The principles of constructing software for built-in systems that reduce consumption of microcontroller resources for data output have been shown. A generalized approach to implementation of software support for LED scale in built-in systems based on microcontrollers has been formulated. The detailed practical algorithm concerning bi-cycle excitation of the multi-element LED bar graph display has been offered.

Keywords: LED, bar graph display, bi-cycle excitation, information model, algorithm, microcontroller, built-in system.

https://doi.org/10.15407/spqeo23.03.329
PACS 85.60.Bt, 42.79.Kr, 07.05.Rm

Manuscript received 23.06.20; revised version received 14.07.20; accepted for publication 10.09.20; published online 22.09.20.

1. Introduction

The main attention of developers of human–machine system has focused on ensuring an appropriate level of parameters inherent to the communication channel between the operator and the technical means. In such a channel, the data transfer process is based on an information model (IM), which defines a system of encoding rules for transmitted messages [1]. Among IM means of displaying information for various purposes, the most reliable are bar graph (scale) forms. It is due to the high-level correspondence of the visual form of the symbols using their value and significant information redundancy of the scale counting. The use of these forms allows one to reduce the number of errors and gross misses when reading and interpreting data by the operator, which ensures high reliability of the communication channel with the operator. Usually, in reliable industrial systems and devices information display uses bar graph IM, where the countdown is defined as the total length of the luminous line and the position of its reference end relatively to the external scale marks [2].

Transforming data into a view that is convenient for use across devices is an important task in control systems. So, the structure of the display device is defined by IM that corresponds to the accepted method of encoding messages and describes the algorithm for the synthesis of visual images at the information area (IA).

With the development of microelectronics, micro-controller units (MCU) have become more widely used in control equipment, which has greatly simplified the cycle of product development and manufacturing, since device design has been increasingly reduced concerning software development. With the advent of universal single-chip MCU, this trend in creation of electronic equipment was brought to a peculiar end. It has become possible to build digital devices for different purposes, being based on the same hardware configuration. This approach is often used for information displays in built-in systems (BS). However, up to now, little attention is paid to software support for data representation in a bar graph form in devices with MCU.

This work is devoted to formation of an optimized approach to the practical software implementation of reliable IM for data representation on multi-element LED bar graphs and thus to minimize MCU resources.

2. Bar graph information model for scale display

When designing the modern equipment for BS, it is important to take into account data visualization in the form the most useful for the optimal reception by the operator. One of this reliable solutions is using the bar graph indicator (BGI) based on LEDs.

The practical use of LED bar graph devices has shown that they have a unique set of technical characteristics that make them indispensable in most

industrial products, as well as in special-purpose systems [3]. From ergonomic and practical points of view, for equipment aimed at the individual use, the optimal bar graph display solution must have 30…150 elements in IA [4]. To increase reliability, the message visualization unit is implemented as a matrix of electric connection of *n* groups by *m* elements. Each group includes elements, which weight functions in serial pairs differs by unity. The value of the weight function of a group of elements is defined by its position in IA relatively to the spatial multi-channel measure.

Representation of the transmitted data is synthesized in IA in the form of visual symbols creating the corresponding alphabet. Additive discrete analog form of IM can be described by the following set [5]:

$$\mathbf{\Omega}_{\mathrm{BG}} = \left\{ S_{1\mathrm{BG}},\ S_{2\mathrm{BG}},\ \ldots,\ S_{\nu\mathrm{BG}},\ \ldots,\ S_{(l-1)\mathrm{BG}},\ S_{l\mathrm{BG}} \right\}.$$

Here, $\mathbf{\Omega}_{\mathrm{BG}}$ is the message alphabet, $S_{\nu\mathrm{BG}}$ – ν-th IM character, where $\nu = \overline{1, l}$, l – IM alphabet length.

Matrix electrical connection does not allow simultaneous excitation of whole set of symbol elements, therefore, dynamic $S_{\nu\mathrm{BG}}$ formation is realized by a number of consecutive time intervals (cycles). Their quantity r is determined by IM and corresponds to the algorithm of scanning the IA elements. The minimum possible number of cycles for any bar graph IM for a two-dimensional matrix of IA elements is equal two [5]. To do this, each of the set of excited elements $\tilde{\mathbf{A}}^{\mathrm{M}}_{\nu\mathrm{BG}}$ is divided into a pair of disjoint subsets $\tilde{\mathbf{A}}^{\mathrm{Mq}}_{\nu\mathrm{BG}}$, which are excited in different cycles of formation of the corresponding symbol $S_{\nu\mathrm{BG}}$. Because of the inertia of human vision, a reliable clock rate of cycling about 50–100 Hz should be maintained for continuous information perception [1].

To represent this IM, we can describe the set of IA elements in the form of a two-dimensional matrix $m \times n$. This description (where the excited elements for symbol $S_{\nu\mathrm{BG}}$ are marked with a tilde) will have the form

$$\mathbf{A}_{\mathrm{M}} = \left\| \begin{array}{cccccccccc}
\tilde{a}_{11} & \tilde{a}_{12} & \ldots & \tilde{a}_{1(y_\nu-1)} & \tilde{a}_{1y_\nu} & \tilde{a}_{1(y_\nu+1)} & \ldots & \tilde{a}_{1(m-1)} & \tilde{a}_{1m} \\
\tilde{a}_{21} & \tilde{a}_{22} & \ldots & \tilde{a}_{2(y_\nu-1)} & \tilde{a}_{2y_\nu} & \tilde{a}_{2(y_\nu+1)} & \ldots & \tilde{a}_{2(m-1)} & \tilde{a}_{2m} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\tilde{a}_{(x_\nu-1)1} & \tilde{a}_{(x_\nu-1)2} & \ldots & \tilde{a}_{(x_\nu-1)(y_\nu-1)} & \tilde{a}_{(x_\nu-1)y_\nu} & \tilde{a}_{(x_\nu-1)(y_\nu+1)} & \ldots & \tilde{a}_{(x_\nu-1)(m-1)} & \tilde{a}_{(x_\nu-1)m} \\
\tilde{a}_{x_\nu 1} & \tilde{a}_{x_\nu 2} & \ldots & \tilde{a}_{x_\nu(y_\nu-1)} & \tilde{a}_{x_\nu y_\nu} & a_{x_\nu(y_\nu+1)} & \ldots & a_{x_\nu(m-1)} & a_{x_\nu m} \\
a_{(x_\nu+1)1} & a_{(x_\nu+1)2} & \ldots & a_{(x_\nu+1)(y_\nu-1)} & a_{(x_\nu+1)y_\nu} & a_{(x_\nu+1)(y_\nu+1)} & \ldots & a_{(x_\nu+1)(m-1)} & a_{(x_\nu+1)m} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
a_{(n-1)1} & a_{(n-1)2} & \ldots & a_{(n-1)(y_\nu-1)} & a_{(n-1)y_\nu} & a_{(n-1)(y_\nu+1)} & \ldots & a_{(n-1)(m-1)} & a_{(n-1)m} \\
a_{n1} & a_{n2} & \ldots & a_{n(y_\nu-1)} & a_{n y_\nu} & a_{n(y_\nu+1)} & \ldots & a_{n(m-1)} & a_{n m}
\end{array} \right\|. \qquad (1)$$

The most widespread approach is as follows: two-coordinate matrix of electrically connected elements of the scale is used for image formation by scanning the matrix in one of the coordinates [1]. These methods of scanning allow to generate an arbitrary set of excited elements, since in any of the cycles any number of elements can be excited either of one included row (low order), or one included column (high order) of the LED matrix. Using these two cycles to excite IA elements considerably increases reliability of data output and lower the level of high-frequency electromagnetic noises caused by the LED control unit.

3. Operation principles of the information model

The obligatory condition to form a persistent sight image of any visual symbol is a relative height of the frequency corresponding to image regeneration $f_S = 1/t_S$ over the critical frequency of flicker fusion [1]. In this case, each group of a_{xy} elements that belongs to the respective subset of $S_{\nu\mathrm{BG}}$ is excited only one time during every period for its regeneration within the time interval $\tau_g = t_S/r$, where r is the number of cycles to synthesize a visual image of $S_{\nu\mathrm{BG}}$ on the display. Dynamic joining the elements into groups in accordance with the realized version of bi-cycle IM assumes realization of the data proceeding logic in the following form:

$$\begin{aligned} A^D_\nu = & \left\{ \bigcup_{x=1}^{q} \bigcup_{y=1}^{m} a_{xy} \,\middle|\, \begin{array}{l} t = t_S + \tau_g - 0 \\ t = t_S + 0 \end{array} \right\} \times \\ & \times \bigcup \left\{ \bigcup_{x=1}^{m} \bigcup_{y=1}^{\nu - mq} a_{xy} \,\middle|\, \begin{array}{l} t = t_S + 2\tau_g - 0 \\ t = t_S + \tau_g + 0 \end{array} \right\}, \end{aligned} \qquad (2)$$

where $q = E(\nu/m)$, $E(b)$ is the entier of b, m – quantity of minor elements of matrix, ν – total number of excited elements IA, a_{xy} – element that has a number y in the group with the number x, t – current time, t_S – the start time of the symbol regeneration period, τ_g – time to change cycle. "0" in the description of time indicates that the adjacent intervals are irregular, *i.e.*, represent open intervals.

This IM describes formation of the symbol A_v^D in the dynamic bi-cycle mode. So, there are defined and used two sets of IA elements: A_1 and A_2 groups that belong to two-fold intervals – from $t = t_S + \tau_g - 0$ to $t = t_S + \tau_g + 0$, respectively. During the first one, the subset which begins with the first element and ends with $b_1 = E\left(\frac{v}{m}\right)$ one is excited. In the second time interval, groups of $b_2 = v - mE\left(\frac{v}{m}\right)$ are excited at IA. The change of the current interval to the next occurs at times that are multiples of *k*, where *k* is an arbitrary integer.

It looks as follows to technical realization bi-cycle image synthesis of the symbol $S_{v\mathrm{BG}}$ that corresponds to (1) and (2) one must form related dynamic set $\tilde{\mathbf{A}}^{\mathrm{D}}_{v\mathrm{BG}}$ or the same one in the matrix form $\tilde{\mathbf{A}}^{\mathrm{M}}_{v\mathrm{BG}}$. These two equivalent sets must be divided into two disjoint subsets $\tilde{\mathbf{A}}^{\mathrm{DM}11}_{v\mathrm{BG}}$, $\tilde{\mathbf{A}}^{\mathrm{DM}21}_{v\mathrm{BG}}$ that are excited in different periods of time

$$\tilde{\mathbf{A}}^{\mathrm{M}}_{v\mathrm{BG}} = \tilde{\mathbf{A}}^{\mathrm{D}}_{v\mathrm{BG}} = \left\{\tilde{\mathbf{A}}^{\mathrm{DM}11}_{v\mathrm{BG}}, \tilde{\mathbf{A}}^{\mathrm{DM}21}_{v\mathrm{BG}}\right\}.$$

This procedure is represented as

$$\tilde{\mathbf{A}}^{\mathrm{DM}21}_{v\mathrm{BG}} = \left\| \begin{array}{cccccccc} 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \tilde{a}_{x_v 1} & \tilde{a}_{x_v 2} & \dots & \tilde{a}_{x_v (y_v - 1)} & \tilde{a}_{x_v y_v} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{array} \right\|. \quad (5)$$

These matrixes (4), (5) describe the subsets of the elements that are involved in the display for dynamic formation of a full image corresponding to the symbol $S_{v\mathrm{BG}}$.

4. Technical realization of the information model

To implement the software for IM (1), we used MCU of the Intel 8051 family, which has proven itself well in various applications, and for a long time remains the undisputed leader in the number of companies that produce its modifications [6]. Accordingly, it was an incentive to reduce costs and accumulate technical expertise

$$\tilde{\mathbf{A}}^{\mathrm{M}}_{v\mathrm{BG}} = \tilde{\mathbf{A}}^{\mathrm{DM}}_{v\mathrm{BG}} = \left\| \begin{array}{ccccccccc} \tilde{a}_{11} & \tilde{a}_{12} & \dots & \tilde{a}_{1(y_v-1)} & \tilde{a}_{1y_v} & \tilde{a}_{1(y_v+1)} & \dots & \tilde{a}_{1(m-1)} & \tilde{a}_{1m} \\ \tilde{a}_{21} & \tilde{a}_{22} & \dots & \tilde{a}_{2(y_v-1)} & \tilde{a}_{2y_v} & \tilde{a}_{2(y_v+1)} & \dots & \tilde{a}_{2(m-1)} & \tilde{a}_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{a}_{(x_v-1)1} & \tilde{a}_{(x_v-1)2} & \dots & \tilde{a}_{(x_v-1)(y_v-1)} & \tilde{a}_{(x_v-1)y_v} & \tilde{a}_{(x_v-1)(y_v+1)} & \dots & \tilde{a}_{(x_v-1)(m-1)} & \tilde{a}_{(x_v-1)m} \\ \tilde{a}_{x_v 1} & \tilde{a}_{x_v 2} & \dots & \tilde{a}_{x_v (y_v-1)} & \tilde{a}_{x_v y_v} & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \end{array} \right\|, \quad (3)$$

moreover,

$$\tilde{\mathbf{A}}^{\mathrm{DM}11}_{v\mathrm{BG}} = \left\| \begin{array}{ccccccccc} \tilde{a}_{11} & \tilde{a}_{12} & \dots & \tilde{a}_{1(y_v-1)} & \tilde{a}_{1y_v} & \tilde{a}_{1(y_v+1)} & \dots & \tilde{a}_{1(m-1)} & \tilde{a}_{1m} \\ \tilde{a}_{21} & \tilde{a}_{22} & \dots & \tilde{a}_{2(y_v-1)} & \tilde{a}_{2y_v} & \tilde{a}_{2(y_v+1)} & \dots & \tilde{a}_{2(m-1)} & \tilde{a}_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{a}_{(x_v-1)1} & \tilde{a}_{(x_v-1)2} & \dots & \tilde{a}_{(x_v-1)(y_v-1)} & \tilde{a}_{(x_v-1)y_v} & \tilde{a}_{(x_v-1)(y_v+1)} & \dots & \tilde{a}_{(x_v-1)(m-1)} & \tilde{a}_{(x_v-1)m} \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \end{array} \right\|, \quad (4)$$

on its basis. Low-level programming was chosen as the software environment, because this approach realizes one-to-one correspondence of language constructions to processor instructions, so allows the most rational use of resources by MCU. This program is more efficient than those generated by translators from high-level programming languages, *i.e.*, the developed programs are characterized by using fewer instructions and memory usage, which allows increasing the speed and reducing the size of the program.

The research of the software synthesis of bar graph IM in the form (2) showed that the main concentration of efforts to reduce the resource consumption of the developed solutions should focus on optimizing the blocks of programs that are executed during interrupts serving the input-output subsystem of MCU. An interrupt mechanism is best suited for handling events that occur asynchronously to program execution. Simultaneously, we solve one of the main tasks of the program optimization – to reduce the percentage of active processor time. In this approach to the construction of bi-cycle bar graph display software, the overall initialization of this subsystem occurs once at the start of the device with MCU. This function must include the general settings of the display interrupt handler and the corresponding variables.

For the effectiveness of the program and for solution of the problem of reducing MCU processor load, one of the main tasks, along with the choice of IM, is creation of an algorithm that will allow one to control the entire chain of processes from obtaining the data to displaying them in a scale, and, if possible, to minimize the number of executing blocks of the code [6].

At the first step of research, we have develop the generalized approach to realization of LED bar graph display control software. This universalization allows the developer to write a set of programs to excite the matrix of LED IA elements corresponding to bi-cycle IM without delving into the integrated mathematical apparatus and tricks used. So, we have a pattern that allows writing the program code, but does not permit one to properly analyze the nodes that can be optimized. At the second step, using this pattern one can obtain several practical code realizations that differ by technical characteristics. These programs can be analyzed in accordance with the selected optimization criteria, and developer will be able to select the best variant.

The generalized algorithm of the bar graph display data visualization service interrupt handler that form bi-cycle excitation of matrix of LED elements is presented in Fig. 1.

The first block provides initialization of the current variables of a particular point in time. The block 2 is a clock selector that is responsible for linking the current interrupt to the corresponding clock functions. The block 3 provides reception and storage in RAM of the current value of data to be visualized and coming from an external device.

The block 4 generates control codes (CC) for the higher bits of the LED matrix in the first and second cycles, respectively,

$$\mathrm{A_{1H}} = \left\{ \bigcup_{x=1}^{q} a_{xy} \left| \begin{array}{l} t = t_S + \tau_g - 0 \\ t = t_S + 0 \end{array} \right. \right\}$$

and

$$\mathrm{A_{2H}} = \left\{ \bigcup_{y=1}^{m} a_{xy} \left| \begin{array}{l} t = t_S + \tau_g - 0 \\ t = t_S + 0 \end{array} \right. \right\}.$$

The block 5 forms the CC for the lower bits of the matrix in the first and second cycles, respectively,

$$\mathrm{A_{1L}} = \left\{ \bigcup_{x=q+1}^{m} a_{xy} \left| \begin{array}{l} t = t_S + 2\tau_g - 0 \\ t = t_S + \tau_g + 0 \end{array} \right. \right\}$$

and

$$\mathrm{A_{2L}} = \left\{ \bigcup_{y=1}^{v-mq} a_{xy} \left| \begin{array}{l} t = t_S + 2\tau_g - 0 \\ t = t_S + \tau_g + 0 \end{array} \right. \right\}.$$

The block 6 is responsible for lock the BGI indication. It is necessary for correct formation of the image in IA when changing the cycle and data shown on the display. This unit eliminates the uncontrolled spurious illumination of the display.

The block 7 transmits previously formed CC at the blocks 4, 5 – $\mathrm{A_{1H}}$, $\mathrm{A_{2H}}$, $\mathrm{A_{1L}}$, $\mathrm{A_{2L}}$ – to the external ports of MCU to excite the respective sets of LEDs in the first and second cycles of output data to the scale. Using inertia of human vision and cyclically repeating the excitation of these two groups of elements with a frequency higher than 50 Hz, we can form a holistic visual image that corresponds to the desired symbol. The formed CC are fixed in the ports of MCU and provide permanent excitation of the elements of the matrix between serial interrupts.

The block 8 is responsible for unlocking indication and shows the new image in IA in accordance with the existing CC in the ports of MCU. Minimizing the runtime of the blocks 6 to 8 increases the brightness of the indicator.

The block 9 performs modification and storage in the memory program variables for further formation of new CC.

The block 10 exits the procedure of the display interrupt handler.

Applying the presented generalized approach to implementation of software support for bar graph information display, we can proceed to the second stage – practical software development.

5. Software implementation of the information model

The generalized approach to the bi-cycle IM software implementation, presented by the algorithm in Fig. 1, creates a functional platform for developing a practical display management code optimized for the used MCU resources. This code takes into account the architecture and functioning principles of MCU [6].

A detailed practical algorithm of the developed software support for controlling a bar graph display based on a matrix of LEDs is shown in Fig. 2. Excitation of the IA elements by the bi-cycle synthesis of IM by the two-character code "*YX*" applied to the matrix through MCU ports (where *X* and *Y* are appropriate MCU ports used for excitation of low- and high-order bits of the matrix or its rows and columns (1) respectively).

The block 1 coordinates the timing functions of the matrix excitation with the system time. The second block loads the current data for visualization in the program. The block 3 blocks the flow of new data in BGI until a full IM cycle completes the specified number of times. The block 4 checks the current cycle number. After that, the algorithm is split into 2 branches – for even and odd cycles (block 5). It allows us to form an arbitrary image in the dynamic bi-cycle mode from two disjoint sets of excited elements of IA. The cycles are formed by calculating the values of A_H (high-order bits of the matrix) and A_L (low-order bits).

In the first cycle, the values $A_H = E(A_H)$ (block 5) and $A_L = (A_L)_{max}$ (block 6) are calculated. Next A_L written to the variable A_{1L} (block 7) and A_H – to the variable A_{1H} (block 8). The block 9 generates control codes (CC) of the low-order bits, block 10 forms CC of the high-order bits of the matrix. In the block 11 all LEDs are switched off. This is necessary in order to avoid parasitical light when cycle is changing. Next, the received value of A_{1H} transmitted to port *Y* (block 12) and A_{1L} – to port *X* (block 13). In the matrix display, it looks like this for high-order bits $A_{1H} = \left| \tilde{a}_{x_v 1}\ \tilde{a}_{x_v 2} \ldots \tilde{a}_{x_v (y_v - 1)}\ 0 \ldots 0 \right|$ and so for low-order bits $A_{1L} = \begin{vmatrix} \tilde{a}_{11} \\ \tilde{a}_{21} \\ \vdots \\ \tilde{a}_{1(x_v+1)} \\ 0 \\ \vdots \\ 0 \end{vmatrix}$. This will allow one to excite in the first cycle only the necessary elements of IA in accordance to (4). The block 14 changes the variable of number of cycle.

In the second cycle, the values $A_H = E(A_H) + 1$ (block 15) and $A_L = 2^{A_L + 1}$ (block 16) are calculated. Next A_H written to the variable A_{2H} (block 17) and A_L – to the variable A_{2L} (block 18). The block 19 forms CC of the low-order bits, the block 20 forms CC of the high-order bits of the matrix. In the block 21 all LEDs are switched off. Next 2 blocks transmit previously formed pair of CC into the ports. So, A_{2H} transmitted to port *Y* (block 22) and A_{2L} – to port *X* (block 23). In matrix representation for high-order bits one can write $A_{2H} = \left| \tilde{a}_{11}\ \tilde{a}_{12} \ldots \tilde{a}_{(y_v+1)}\ 0 \ldots 0 \right|$ and for low-order bits as $A_{2L} = \begin{vmatrix} \tilde{a}_{11} \\ \tilde{a}_{21} \\ \vdots \\ \tilde{a}_{x_v y_v} \\ 0 \\ \vdots \\ 0 \end{vmatrix}$. Since both excited LED sets are disjoint, this avoids spurious illumination of IA elements. The next block 24 changes the variable of number of cycle.

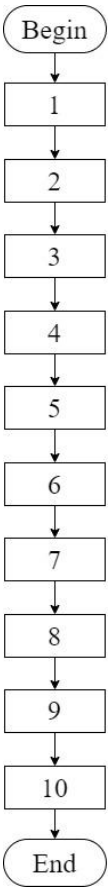

Fig. 1. The generalized algorithm of visualization service interrupt handler.

At the following step, the display will be unlocked (block 25), and visualization of the corresponding half of the image on IA begins, until the next interrupt appears. After that, in the block 26, the internal program variables are updated and they are stored in the system memory (block 27) for later use when forming new CC. In the block 27, the exit from the interrupt handler occurs.

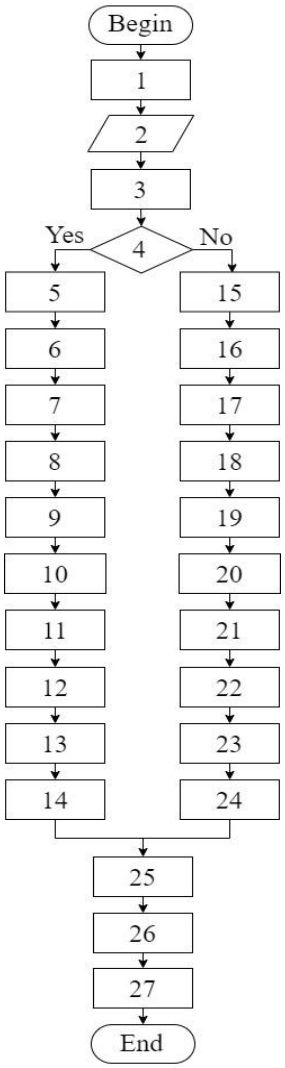

Fig. 2. The developed program algorithm for controlling a bar graph display based on a LED matrix.

The algorithm shown in Fig. 2 was implemented in assembly language for MCU Intel 8051 family and successfully tested as a part of software for built-in system that processes the measurement information for the operator console.

6. Conclusions

The relevance of implementation of display devices using MCU has been shown. In these technical solutions, the entire set of technical and economic parameters is primarily influenced by the selected type of IM and its software support.

Being based on examination of analytical representation of IM for a bar graph information display, a generalized approach to implementation of software support for LED scale in built-in systems based on MCU has been formulated. As a result, a generalized functional platform was created for developing a practical display control code optimized for the MCU resources used.

On this basis, the bar graph indication software for the built-in system with LED display based on MCU of the MCS-51 family has been developed, tested and presented as a practical algorithm. The developed algorithm allows minimizing the need for system resources when implementing the appropriate code in the cyclic interrupt service routine.

The results obtained in the work can serve as the basis for effective solutions to problems of improving the level of technical and economic indicators of serial and specialized devices, as well as simplifying their integration and implementation in advanced automated control tools for complex objects and processes.

References

1. *Handbook of Visual Display Technology*. Eds. J. Chen, W. Cranton, M. Fihn. Springer, Cham, 2016.
2. Bushma A.V. Information security for optoelectronic ergatic system. *Semiconductor Physics, Quantum Electronics and Optoelectronics.* 2010. **13**, No 2. P. 170–172. https://doi.org/10.15407/spqeo12.02.170.
3. Wu T., Sher C.-W., Lin Y. *et al.* Mini-LED and micro-LED: Promising candidates for the next generation display technology. *Appl. Sci*. 2018. **8**, No 9. P. 1557. https://doi.org/10.3390/app8091557.
4. Bushma A.V. Information redundancy of data visualization forms as a means to improve reliability of electronic equipment. *Radioelectronics and communications systems*. 2003. **46**, No 2. P. 5–10.
5. Bushma A.V., Sukatch G.A. Possible variants of two-cycle discrete-analog representation of information. *Radioelectronics and Communications Systems.* 2006. **49**, No 2. P. 11–17.
6. *Principles and Applications of Microcomputers*: 8051 Microcontroller Software, Hardware, and Interfacing. Ming-Bo Lin. CreateSpace Independent Publishing Platform, 2016.

Authors and CV

Aleksandr V. Bushma, Doctor of Sciences, Professor, Professor of the Department of Computer Science and Mathematics, Borys Grinchenko Kyiv University. The author of more than 250 publications and patents. His research interests include problems of optoelectronic systems, human–machine interaction, microcontroller systems, biosensors, data display and processing.

Andrii V. Turukalo, Postgraduate student of the Department of Computer Science at the National University of Life and Environmental Sciences of Ukraine. Field of scientific interests: optoelectronic systems, human–machine systems, built-in systems and microcontroller programming.

Програмне забезпечення для керування світлодіодним шкальним індикатором

О.В. Бушма, А.В. Турукало

Анотація. Робота присвячена програмній реалізації двотактних інформаційних моделей для світлодіодних шкальних індикаторів. Показано принципи побудови програмного забезпечення для вбудованих систем, які знижують споживання ресурсів мікроконтролера на вивід даних. Сформовано узагальнений підхід до реалізації програмної підтримки світлодіодної шкали у вбудованих системах на основі мікроконтролерів. Запропоновано детальний практичний алгоритм двотактного збудження багатоелементного світлодіодного шкального індикатора.

Ключові слова: світлодіод, шкальний індикатор, двотактне збудження, інформаційна модель, алгоритм, мікроконтролер, вбудована система.