«ЎЗБЕКИСТОН ТЕМИР ЙЎЛЛАРИ» АКЦИЯДОРЛИК ЖАМИЯТИ

ТОШКЕНТ ТЕМИР ЙЎЛ МУҲАНДИСЛАРИ ИНСТИТУТИ

УДК 656.25

**ИРМУХАМЕДОВ ИБРОХИМ АБДУЛЖАМИЛ ЎҒЛИ**

# COMPUTER SOFTWARE PROGRAMMING FOR NMIIP BOX IN BOXED RELAY ROUTE INTERLOCKING SYSTEM

5А311002 – Темир йўл транспортида автоматика ва телемеханика

Магистр академик даражасини олиш учун ёзилган диссертация

Илмий раҳбар:

. т.ф.н., проф. АзизовА.Р.

Ташкент-2018

# ANNOTATION

Nowadays introduction of more reliable, fault-tolerant devices and systems for the "Automation and Telemechanics" branch has become one of the urgent tasks. This is especially true for high-speed traffic. It's no secret that high-speed traffic safety requires devices with the highest level of reliability.

There are different reasons for the need to introduce microprocessor and relay-processor centralization systems in place of existing ones. Advantages in the introduction of microprocessor and computer technology can be attributed to the ease of adaptation of the system, ease of linking with other systems, reduced dimensions of equipment.

The thesis is devoted to solving the actual task of developing the algorithm and software for the block of the set-up group of the BRRI system in the field of railway automation and telemechanics.

It solved the task of creating an algorithm for the existing NMIIP block of the set-up group of the BRRI system, software for this block and a computer model in the MicrosoftVisualstudio software environment were developed.

The developed software for the NMIIP computer model can be used as part of a new microprocessor centralization system. In this case, all other boxes of the set group should be developed in a similar way. It is necessary to take into account the correspondence of input and output data of different blocks of the BMRT set-up group.

The remaining units (the executive group) remain on the relay-contact basis. This system is considered to be a relay-contact centralization (RPM) system, in its processor part, the logical functions of the dial-up group will be performed using a computer, with the help of a microprocessor module, commands will be sent to the relay group executive group.

The computer model of the NMIIP block is also recommended for use in conducting laboratory exercises, in order to consolidate the acquired knowledge in the study of the BRRI recruitment group.

# АННОТАЦИЯ

Ҳозирги кунда бузилишларга бардошли энерго тежамкор қурилма ва тизимларни “Автоматика ва Телемеханика” тизимларига тадбиқ қилиш долзарб масалалардан бири бўлиб турубди. Бу айниқса, юқори тезликда ҳаракатланувчи темир йўл транспорти учун жорий қилиниши лозим. Ҳеч кимга сир эмас юқори тезликда ҳаракатланувчи поездларнинг хавфсизлигини таъминлаш тизимларига юқори ишончлилик талаблари қўйилади.

Темир йўл автоматика ва телемеханика тизимларига микропроцессорли тизимларни жалб қилишнинг бир неча сабаблари мавжуд, бу айниқса станцияларда стрелка ва сигналларни марказлаштириш тизимларида намоён бўлади. Микропроцессорли ва компьютер технологияларини автоматика ва телемеханикага тадбиқ қилишнинг авзаллиги шундаки, улар бошқа тизимлар билан онсон боғланади, хажм ва габаритнинг камайиши ва энерготежамкорлик.

Диссертация блокли маршрутли релели марказлаштириш тизимининг НМІІП блокининг ишлаш алгоритми ва дастурини тузиб чиқишга бағишланган.

Унда БМРМ тизимининг мавжуд НМІІП блокининг ишлаш алгоритмини такомиллаштирган ҳолда Microsoft Visual studio дастурида C# тили асосида блокнинг дастурий.

Бундан ташқари НМІІП блокининг компьютер моделини амалиёт дарсларида интерактив таълим тизимини ташкил қилган ҳолда ҳам БМРМ тизимининг териш гуруҳини ўрганиш мақсадида ҳам фойдаланиш мумкин.

## CONTENT

# INTRODUCTION

The head of Uzbekistan ShavkatMirziyoyevMiromonovich in the 18th meeting of the Council of Heads of State of the SCO member states held on June 10 noted «Since the beginning of this year, the volume of trade between Uzbekistan and the SCO states, observers and partners has grown by more than a third. And this is not the limit of our capabilities.

In our opinion, it is important to simplify the procedures for export-import operations in order to ensure further growth in trade turnover.

We also support the formation of agro-industrial clusters based on the principles of public-private partnership and expansion of "green corridors" network in supply of agricultural products", said the President of Uzbekistan. The need for effectively using the transport and transit potential of the SCO space, relevance of implementing interregional transport projects within the framework of "One Belt, One Road" initiative was noted.

It was emphasized that Uzbekistan supports the construction of Mazar-i-Sharif – Herat, China – Kyrgyzstan – Uzbekistan railway lines, as well as development of trans-regional corridors of Central Asia – Persian Gulf, North – South and East – West»

Theseare a great responsibility for the railway workers in every sphere. railroad raises, railway traffic safety, passenger and cargo security should improve the conditions for servicing passengers and customers. for this, technology should improve, especially at stations.

The systems of electric centralization (EC) of arrows and signals are intended for controlling the movement of trains at railway stations. They provide the necessary capacity and safety of movement. The first systems of centralization at stations appeared in the middle of the XIX century (1856, England) and were mechanical. In them, to move the switches of the strelok and the wings of semaphores, the human effort was applied, which was transferred to the transfer mechanisms by means of arrow and signal levers and rigid or flexible (wire) rods. Blocking relationships between arrows and semaphores

were also mechanically performed in special crates of dependencies and with the help of mechanical locks.

Although relay centralization with success can easily last for another 80 years, morally it has long been outdated. To replace the electrical centralization of the relay type, developed back in the 60-80 years of the last century, came the microprocessor centralization (MPC).

Microprocessor centralization on the railroad is an urgent need to update the technological process of management both by rail transportation and the work of various structural units of all railway transport, which includes not only the infrastructure of UTY, but also access roads of non-public use in the conduct of industrial enterprises.

MPC on railways is the link between the objects of the SATS JAT, the rolling stock, etc., which are primary sources of information, and the management system of the entire higher-level transportation process. It is the MPC JAT that allows you to link the entire set of these sources together without using additional settings, which is practically not realistic under relay centralization.

Equipment of the station with MPC devices allows avoiding paralysis, albeit not prolonged, of its operation. The point is that in the MPC of the SCB uninterruptible power supplies are used, which are not used in the centralization of the relay type.

## 1. COMPILE NECESSERY INFORMATION ABOUT BRRI SYSTEM .

### 1.1Explore features of the BRRI system

When several trains are in use on a railway network it is necessary to secure the travel of each train, in order to prevent collisions and derailings.

Security in a railway network is ensured with an *interlocking system* which controls signals and points.

As technology has evolved so has the type of interlocking systems and hence different technologies have been used.

Different Technologies - At the current time, four major types of interlocking systems have been implemented in the Danish railway network, namely the *mechanical interlocking system*, the *electro mechanic interlocking systems*, the *relay interlocking system*, and the *electronic interlocking system*. The different types of systems will be described briefly in the following sections.

Mechanical interlocking systems - Towards the end of the 19$^{th}$ century mechanical interlocking systems were used. The system created inter-dependency between signals and points using wire pulls. At a given time the operations that would lead to a collision were mechanically blocked.

Electro mechanic interlocking systems - Electro mechanic interlocking systems were implemented in the railway network from around 1915 until 1950. The interlocking system is a mix of the mechanical interlocking system and the relay interlocking system (see section2.1.4). External components such as points and signals were now controlled by relays instead of wire pulls. The part of the interlocking system that prevented trains from colliding, were still mechanically controlled. The control was now made by a mechanical registry consisting of steel beams with holes in.

Relay interlocking systems - In the middle of the 1990ies, relay interlocking systems of the type DSB 1953 and 1954 were introduced. Wire pulls were replaced by pushbuttons on an operators panel and, among other things, electrical monitoring was possible using track sections. Electronic interlocking systems - The newest technology is an interlocking system controlled by microprocessors. Instead of an operators panel the monitoring takes place on a computer screen.

**Relay Interlocking Systems.** In this thesis the focus will be on relay interlocking systems. From this point on a ”relay interlocking system” will be referred to as an interlocking system. An interlocking system is a quite extensive system and thus a complete description is considered outside the scope of this project. For this reason it is only the most essential part of the interlocking

system that is described in this section.

One type of interlocking system covers the open track between stations, while another covers the station itself. The systems at the stations are of the most interest since they are by far the most complicated. The goal of an interlocking system is, as previously mentioned, to secure the travel of a train. This is obtained by creating a complex circuit where each component in the circuit translates into a logic condition, e.g. ”if there is a train at the station, let the entry signal be red” (this condition requires several components though). The most important component in the circuit and thus in the interlocking system, is an electrical component called a *relay*. Besides relays a station is composed by *track sections, points, buttons, fuses, resistors, signals* and *lamps.*

In the following section the mechanical functionality of a relay will be explained. Next, the previously mentioned components on a station will be introduced,whereafter the structure in a circuit will be defined. After that, abstract concepts such as *train route* and *train route table* will be clarified. Finally the process of translating abstract conditions into a circuit of electrical components will be explained and a typical procedure for a train entering a station will be examined.

Relays - There are two mechanical types of relays in a circuit, namely *regular relays* and *steel core relays*. First, the functionality of a regular relay will be explained and thereafter the distinctive functionality of the steel core relay will be examined.

Regular relays - A regular relay consists of a coil, an electromagnet, an armature, a pole with horizontal conductive bars and a number of contacts, typically 6, 10 or 20. The electromagnet is placed inside the coil and each end of the coiling is connected to a pin. When no current is applied to these pins, the electromagnet is demagnetized and the state of the relay will be as on figure2.1. Each contact consists of two pins, to which wires can be connected. The lower contacts on figure2.1 are said to be *closed,* since current can pass from one pin on the contact to the other pin on the contact, via the horizontal bar. The upper

contacts of the relay are said to be *open*, since the horizontal bar through which the current can pass, is not in contact with the pins.

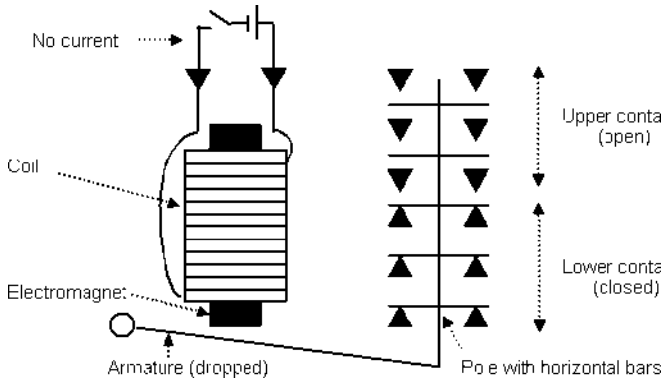


Figure 1.1: *The relay only allows current to pass through the lower contacts, given that the armature is dropped*

When current is applied to the coil pins the electromagnet will carry current and magnetize. The magnetized electromagnet draws the armature which in turn pushes the pole upwards. This will invert the state of the contacts so that the upper contacts are closed and the lower contacts are open, as seen on figure

When no more current is applied, the electromagnet will demagnetize, making the armature, and thus the pole, drop. The relay has now returned to the original state on figure1.1.

The pins are the only externally accessible parts on the relay since the other components are protected from dust and wear by a black box as seen on figure

The pins on a relay can be numbered in one of two ways. The coil pins are the uppermost pins as seen. In general it is the relay that is said to be *dropped* and *drawn*, even though it is in fact the armature and pole that are dropped and drawn respectively.

**Steel core relays**. Steel core relays mechanically differ from regular relays in that there instead of an electromagnetic core, is a core of heat-treated steel. The heat-treated steel core causes the core to remain magnetized, even when the supply of current is stopped. The coil in which the steel core is placed, has two

coilings; a magnetizing coiling and a demagnetizing coiling. Initially the steel core relay is magnetized. When current is applied to the demagnetizing coiling, the steel core will demagnetize and remain demagnetized until current is applied to the magnetizing coiling. When the steel core is magnetized, it will remain magnetized until current is applied to the demagnetizing coiling. The pin with ID 12 or 14 (see the pin's positions on the relay in figure1.4) is not externally connectable, since it is connected internally as regards the coilings.

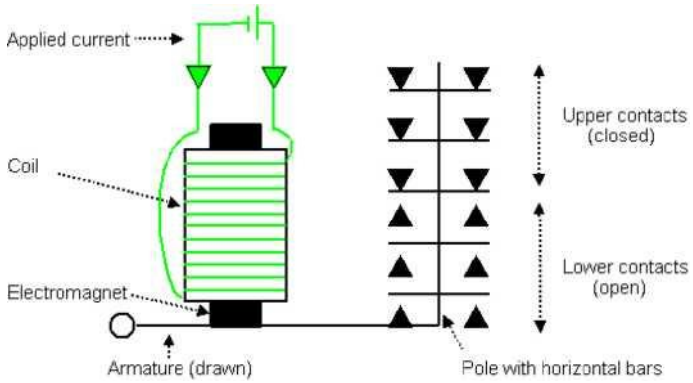


Figure 1.2: *The relay only allows current to pass through the upper contacts, given that the armature is drawn (green illustrates current).*

*Figure 1.3:* The coil, armature, electromagnet and the pole with horizontal bars are protected from dust and wear by a black box.

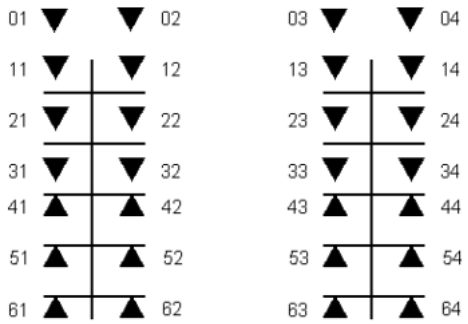


*Figure 1.4:* The pins on the relay are numbered in one of two ways.

Track Sections - A track has the ability to carry current. A track section is a piece of a track that is isolated so that the current does not spread from one track section to another. This means that track sections can carry current independently of each other. Each track section is connected to a relay so that the relay is drawn when the track section is free, see figure 2.5. The wheels and the axles of the train are conductive. This means, that when the wheels of the train come in contact with the track section, the circuit shorts out, see figure 2.6. The only external influences on the track section that will affact the state of the relay is a train or other conductive components. On the operators panel track sections are identified by a short text. E.g. in figure 2.8 the leftmost track section is denoted by ”01” just beneath the track section.

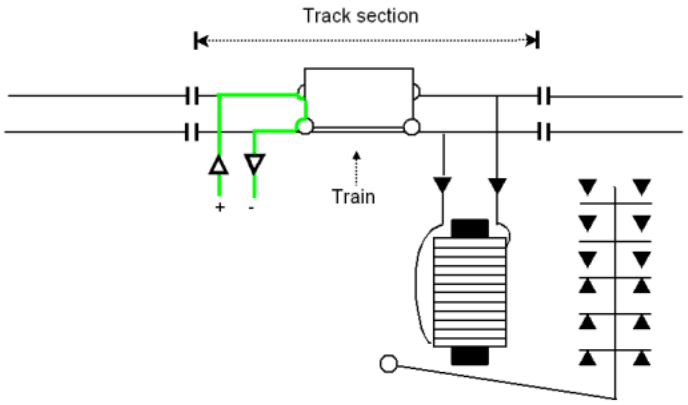


Figure 1.5: The relay is dropped, giving the track section is occupied.

Points - Some track sections contain a branching of the track. These track sections are called points.

Besides the functionality of a track section, the point has the functionality of being switched to one of two directions. A point can thus be in one of three states; switched to the plus direction, switched to the minus direction, or switched in an intermediate state. The intermediate state corresponds to the situation where the point has not been completely switched, i.e. the situation where the point is neither switched to the plus direction nor the minus direction. The plus direction is the direction to the right and the minus direction is the direction to the left, as seen on figure 2.7. Besides being connected to the relay specified in the previous section, a point is connected to two relays: A plus and

a minus relay. These relays indicate which position the track is switched to.

When the point is switched to the plus direction the plus relay is drawn minus relay is dropped, when the point is switched to the minus direction the minus relay is drawn and the plus relay is dropped. And finally when the point is in the intermediate state both relays are dropped. This means that only one of the plus and minus relays can be drawn at the same time.

Signals and Lamps - A signal is similar to a traffic signal, in that it is a set of lamps. On a station there are different types of signals for instance entry, exit and platform signals. A lamp consists of one or two filaments of which the second filament is a spare filament. Usually each filament is connected to one relay, so that when the filament is live the relay is drawn, and when the filament is not live, the relay is dropped. The filament is conductive when it is intact and non-conducting if it is burned out.

Signals are the final indication of an interlocking system, in the sense that the interlocking system interprets the current state of the station and uses the signals to indicate to the driver which actions he/she should take. If all conditions for letting a specific train enter/exit the station at its current state are

fulfilled the relevant signals will show ”go”, indicating to the driver that it is safe to enter/exit the station.

Circuit structure The structure in a circuit is created by connecting terminals with wires. ”Terminal” is a common term for the ”power in” and ”power out” components and the pins on the electrical components. The ”power in” and ”power out” components on the power source will from now on be referred to as the positive pole and the negative pole.

The following statements apply for a functional circuit:

• Connections are made between terminals.

• A pin can be connected to two terminals at the most.

• There is no upper limit on how many pins the positive and negative poles can be connected to.

• There exists at least one path from the positive pole to the negative pole in a circuit. A path consists of terminals and wires.

• Each component has a certain amount of internal resistance. If there are several branching paths with different internal resistance, most of the current will only go through the branching with the least internal resistance. If e.g. a circuit has two branches, one with coil pins and one with a contact the contact will have the least amount of internal resistance. Most of the current will pass through the contact and too little current will pass through the coil pins to make the relay draw.

Train Routes - A train route is an abstract concept describing a set of assembled track sections. There is one set of train routes for entering a station and one set of train routes for leaving a station. A simple station on a single tracked stretch will typically have 4 train routes per track on the station, since there is an enter and exit train route per driving direction through the station. On the schematic layout of the operators panel in figure 2.8 the enter train routes would be:

1. Entering at A using plus direction: {01, 02, 03}

2. Entering at A using minus direction: {01, 02, 04}

3. Entering at B using plus direction: {06, 05, 04}

4. Entering at B using minus direction: {06, 05, 03}

and the exit train routes would be:

1. Exiting towards A from 04: {04, 02, 01}

2. Exiting towards A from 03: {03, 02, 01}

3. Exiting towards B from 04: {04, 05, 06}

4. Exiting towards B from 03: {03, 05, 06}

On larger stations or stations on double tracked stretches there are several trainroutes. When an operator wants to allow a train to enter the station, he/she will initiate a locking of a train route. When a specific train route, e.g. for entering the station is locked, the track sections in the train route are reserved for the trainand points are switched correctly. This means the train can safely enter the station. The locking of the train route prevents train routes that might lead to collisions (so-called conflicting train routes) from being locked. Once the train route is locked and the train is at a certain place on the train route (specified in the train route table) there will be a release of the train route. The release of the train route cancels the reservation of the train route allowing train routes that were previously conflicting to be locked.

### 1.2 Entirely exploring working algorithm of the BRRI system

The systems of electric centralization (EC) of arrows and signals are intended for controlling the movement of trains at railway stations. They provide the necessary capacity and safety of movement. The first systems of centralization at the stations appeared in the middle of the 19th century. (1856, England) and were mechanical. In them, to move the switches of the arrows and wings of semaphores, a man's muscular effort was required, which was transferred to the transfer mechanisms with the help of arrow and signal levers and rigid or flexible (wire) rods. Blocking dependencies between arrows and semaphores were also implemented mechanically in special crates of dependencies and with the help of mechanical locks. In Russia, the first systems

of mechanical centralization were built on the St. Petersburg-Moscow line in the 1970s. XIX century.

With the advent of electric motors and electromagnetic contact relays, the need to use human muscle forces to control remote objects has disappeared. At the end of XIX century. Electromechanical and electrical centralizations (1891, USA, 1893, Austria) are beginning to be applied. In Russia, the first electromechanical centralization was built at the Vitebsk station of the Rigo-Oryol road (1909). At the station.Gudermes in 1934 was put into operation the first purely relay EC.

XX century.passed under the sign of the introduction and operation of relay systems of centralization of arrows and signals. At the same time, block route-relay centralization (BRRI) developed at the Giprotranssignalsvyaz institute (GTSS, Russia) in 1961 was the most widely used. It had good technical and operational characteristics. The principles laid down in the BISC were also used in other EC developments.

Relay systems had important advantages over mechanical ones. First of all, they significantly reduced the route setting time, which, together with the use of the principle of dividing routes into sections and sectional opening, made it possible to dramatically increase the capacity of the station necks. In principle, the problem of safety was solved in a new fashion in relay systems. Locking dependencies are realized in them in a schematic, logical way, which makes the EC algorithms more flexible and perfect [1].

In the 60's. In the past century, in connection with the development of semiconductor technology, work begins on the creation of the first electronic systems of EC, built on transistors and other non-contact elements. These works were carried out in three directions: the implementation of EC in the form of an electronic block system on safety elements or elements with symmetric failures; solving the problems of EC on general-purpose computers; Construction of an EC based on a specialized computer. At this time there are experimental

installations of electronic systems in England, Germany, France, Japan and other countries.

The first Russian non-contact station automation system is the contactless route dialing system (BMN) developed at the Leningrad Institute of Railway Engineers. transport. It was built on transistor OR-NE elements and was used in conjunction with the relay EC executive relay group. The use of transistors has made it possible to reduce the dimensions of EC units, to reduce the cost of maintaining equipment, and to increase the reliability of the operation of devices. The BMN system was introduced at the stations of Rezekne (1968) and Obukhovo (1969). Experience in the operation of these systems (BMN system at ObukhovoOktyabrskaya Railway Station for more than 30 years, practically without maintenance) showed that non-contact technology makes it possible to create highly reliable, unattended systems.

Electronic systems of EC in the 1960s.have not received wide distribution on the network of domestic and foreign roads, since the element base on which they were built was not promising. Such an element base was placed at the disposal of engineers in the mid-1970s. After the emergence of Intel's microprocessors. The advantages and capabilities of microprocessor technology in comparison with the relay are so great that it determines precisely at the end of the 20th and at the beginning of the 21st century. ways of development of means of automation in industry and transport. The main advantages include: higher reliability due to the absence of mechanical movements of parts and short-circuiting contacts, which allows the creation of unattended equipment; small sizes of integrated circuits, which allow to provide significant redundancy for giving the equipment the properties of fault tolerance and control of suitability; rapid dynamics of the development of integrated technology, which leads to a continuous improvement of its characteristics and a decrease in cost.

Although relay centralization with success can easily last for another 80 years, morally it has long been outdated. To replace the electrical centralization

of the relay type, developed back in the 60-80 years of the last century, came the microprocessor centralization (MPC).

MPC on railways is the link between the objects of the SATS JAT, the rolling stock, etc., which are primary sources of information, and the management system of the entire higher-level transportation process. It is the MPC JAT that allows you to link the entire set of these sources together without using additional settings, which is practically not realistic under relay centralization.

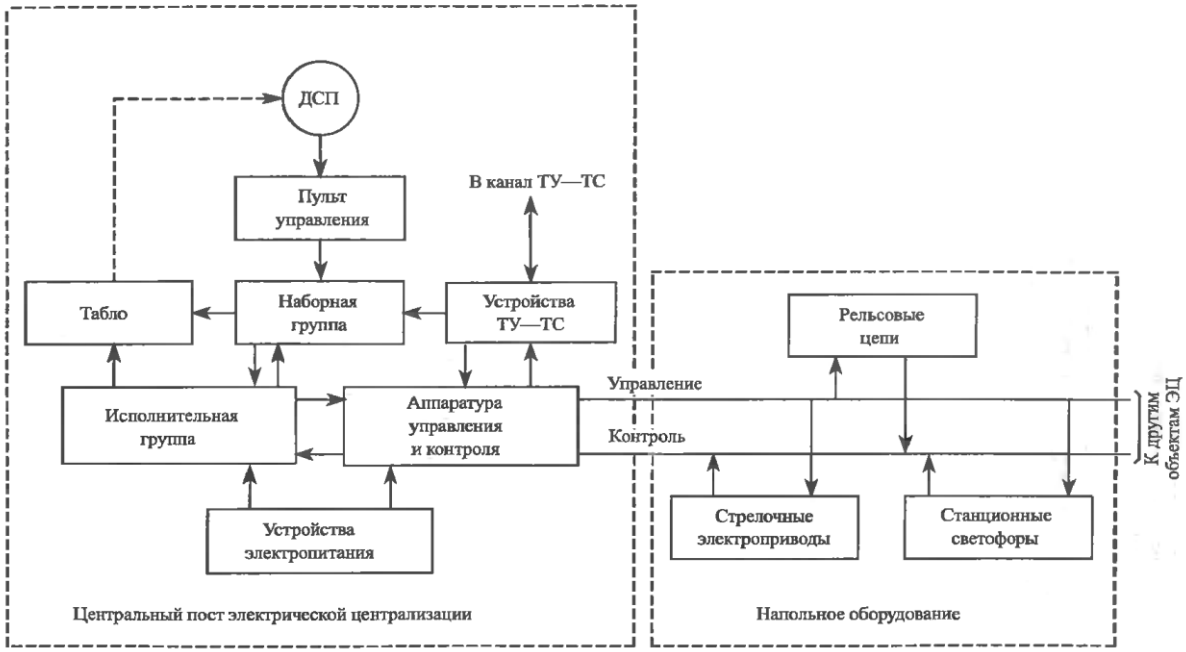


**Figure 1.6. Block diagram of electrical centralization**

Equipment of the station with MPC devices allows avoiding paralysis, albeit not prolonged, of its operation. The point is that in the MPC of the SCB uninterruptible power supplies are used, which are not used in the centralization of the relay type.

The first microprocessor centralization of JZH-850 by Ericsson was commissioned at the station. Gothenburg (Sweden) in 1978. Since then, for a quarter of a century, microprocessor centralization systems (MPCs) of various firms are built on hundreds of stations in many countries around the world. In this direction, the companies Ericsson, Bombardier (Sweden), Siemens, AEG

(Germany), Alcatel (Austria), Westinghouse (England), DSI (Denmark), JNR (Japan) and others are actively working in this direction.

Station stations have a complex track development. They are characterized by a large train and shunting work. To accelerate train and shunting work, as well as ensure the safety of train traffic at these stations, all train and shunting movements are routed. By the condition of the shunting work, only in some cases, non-routed movements are provided with the transfer of arrows to the local control from the shunting columns.

To increase the operational performance at the precinct stations, instead of the relay centralization system with separate control, both at intermediate stations, the routing and relay centralization (MRC) has been developed and widely implemented. In this system, to accelerate the installation of routes, the arrows and routes are not transferred separately sequentially, but simultaneously (all the arrows entering the route). Route control is carried out using the buttons on the control panel along the boundaries of the train and shunting routes. Sequential pressing of the buttons along the route boundaries, according to the "from-to-where" principle, includes starting circuits for simultaneous translation of the arrows entering the route. With the route control, the total time for setting the most complicated route consists of the time of pressing the buttons, the time of parallel transfer of the single arrows entering the route, and the consecutive transfer of the paired arrows, which is approximately 5-10 s. With the successive translation of the arrows entering the route (about 15 arrows), the time to set the route will be 4.5x15 = 67.5 s. Due to the shortening of the time for setting up routes with the route control, the throughput of the station neck is increased by 15-20%. The MRC system also improves the productivity and culture of the plant's operational staff.

The relay equipment of the route-relay centralization is divided into a set-up and an executive group. The dial-up group is called a routing set and is used to form the trigger control arrows. The executive group establishes the installation and closure of routes, the management of traffic lights of train and

shunting routes, as well as the opening of routes. The dial-up group does not perform dependencies for ensuring the safety of train traffic, therefore the relay of the route set takes the II class of reliability of the CRTT type. The executive group fulfills all the requirements for ensuring the safety of train traffic, therefore relays I of the reliability class NMS and KMSh are used in this group. The schemes of the executive group are unified, and they can be used for separate and route control.

Depending on the design configuration of the equipment, the MRC system can be non-block and block types (BRRI). Blocks are typical products manufactured at the factory.

At the beginning of the implementation, the route-relay centralization was projected and built on a non-block type. The set-up group was performed on the open-loop CCTV relay, the executive group - in the form of standard circuit nodes on the NMSH relay also with open mounting. Then came the block system, in which blocks of a closed type were used only for the executive group, the dialing group was still performed on open-mounted equipment. Such a system was built before 1966, after this period, the block structure of the route set was applied, and simultaneously the starting block blocks began to be used. A completely block-based route-relay centralization of BRRI appeared. Only a small part of the relay equipment is not assembled into blocks and is mounted with open mounting.

The BRRI system allows: to produce 70% of relay equipment at the plant, using typical circuit blocks, which significantly reduces the amount of installation work at the construction sites; check and adjust the units on a special stand, which improves the quality of installation work; reduce by 30-35% the time for designing relay centralization, as well as reduce the amount of project documentation by 40%.

Design BRRI is reduced to the set and connection of typical circuit blocks located on the path development of a given station. In design organizations, to speed up the design work, the circuit blocks are made on standard letterheads,

and the designers select the schemes by selecting and gluing the circuit blocks. A number of design documents are carried out using a computer and a plotter, which, according to a given program, produces drawings for the design of a functional block diagram and cable networks.

The BRRI system allows to significantly reduce the amount of installation work during construction and accelerate the introduction of centrally-controlled devices. Relay blocks have a plug-in connection to the operating circuit, which allows for quick replacement of the faulty unit without damage to the centralization. The BRRI system is widely used on the main and industrial transport network. In recent years, in order to unify the block system, not only precinct, but also intermediate stations were equipped.

The electronic systems of the EC in the 1960s were not widely used on the railway network, since the element base on which they were built was not promising. Such an element base was placed at the disposal of engineers in the mid-1970s after Intel's microprocessors appeared on the market. The advantages and capabilities of microprocessor technology in comparison with the relay are so great that it is precisely at the end of the 20th and the beginning of the 21st century that it determines the ways of development of automation means in industry and transport. The main advantages include higher reliability due to the absence of mechanical movements of parts and short-circuiting contacts, which makes it possible to create highly reliable, unattended equipment; the small size of the integrated circuits makes it possible to practically introduce significant redundancy to make the equipment fail-safe and maintainable; the rapid dynamics of the development of integrated technology leads to a constant improvement of its characteristics and a decrease in cost.

The operation of the BIS system when installing, for example, the reception route, the main path starts with the dial-up group. Pressing the start button of the route determines the direction and category of the route, after which the set of the route of the other direction and category is excluded. For the given direction and category of the route, when the NN button is pressed, the

beginning of the route from the traffic light is determined. Pressing the end button of the NC route determines the end of the route. After this, within the established boundaries, all the arrows entering the route are routed. After the transfer of the arrows by a special correspondence scheme, the correctness of the set and position of the translated arrows is checked. If there is a match, the start relay N and the end of the dialed train train are switched on, and the transition to the executive group takes place.

The work of the executive group begins with setting the Route Setup mode. Depending on the established boundaries of the dialed route, the waypoints and the section sections that are included in this route are selected. After this, all the conditions for the correctness of the established route are monitored with the help of the control-sectional relays. If the route is set correctly and the KS relay of all sections of the route is activated, then the closing relays 3 of these sections are switched off, the sections are closed. With the control of the closure of the route, the signal relay C opens, which opens the traffic light. If it is necessary to immediately stop the traffic light, press the start button of the NN route of this traffic light.

After the route is ready and the traffic light is opened, the route is opened in the modes of automatic sectional disconnection, cancellation of the route and artificial disconnection. Automatic disconnection occurs in the process of trains on the sections of the route. When the train enters the section 1SP (position 1), the traffic light is closed, and the section is prepared for opening. After the complete release of the sections / C / 7 and the occupation of the SST section (position 2), the section US / 7 is opened. Similarly, the sections SCH and SSP are opened.

If it is impossible to use the route, it is canceled. To do this, press the group cancel button OGK and the button H of the beginning of the route to close the traffic light. After the traffic light is closed, the route opens with a time delay of 5 s when the approach section is free; in case of occupancy of this site, with a

time delay of 3 minutes. Similarly, the shunting route with a time delay of 5 s with a free approach segment is canceled; 1 min when busy.

Artificial opening is performed by pressing the IRK buttons of the track and arrow sections entering the route, the group button of the artificial GIRK trim, the start button of the NN route to close the traffic light. After pressing all the buttons, the route opens with a time delay of 3-4 minutes.

**1.3.Investigate working algorithm of relays' K, KN, AKN, MP, VKM, VP in boxes NMIIP and NMIIAP**

The first microprocessor centralization of JZH-850 by Ericsson was commissioned at Gothenburg station in Sweden in 1978. Since then, for a quarter of a century, MPC systems of various firms have been built in hundreds of stations around the world. In this direction, the companies Ericsson, Bombardier (Sweden), Siemens, AEG (Germany), Alcatel (Austria), Westinghouse (England), DSI (Denmark), JNR (Japan) and others are actively working in this direction.

If we miss the stage of creating and developing train traffic control equipment, based on the use of mechanical means to implement the dependencies between the arrows and signals, the initial history of the creation of FTIAT devices implemented on electrical principles dates back to the 1930s and 1940s. These works laid the foundation for the development of optimal STS devices, which made it possible to provide one of the leading places in the implementation of safety requirements for train traffic.

Advantages of the microprocessor centralization system:

1. Installation MPC-I allows you to create on the site a single center for remote control of train traffic and provides integration with DC, STDM and radios; to integrate inactive stations into a single control post without the need to equip central posts and line points and with the preservation of local control.

2. Due to the use of client-server architecture, it is possible to organize an information management system of any complexity and configuration, with the possibility of further addition and development of functions.

3. The microprocessor system MPC-I allows dividing large railway stations into an unlimited number of control zones (permanent and seasonal), and at stations with shunting work, create temporary local control areas (control from a switch post or the organization of an additional work station) .

4. Multi-level hierarchical microprocessor centralization of arrows and signals by the type of "zone-station-section-road" is possible, providing, if necessary, operational transfer of control to the appropriate level.

The basis of the EBILock 950 MPC is a central processing unit (CPU) and a system of centralized or distributed object controllers.

The MPC of the MPC EBILock 950 collects information on the status of various floor objects, processes centralization data and sends orders to the relevant object controllers, which in turn control the floor objects.

The data transfer system provides the transfer of orders from the CPU to object controllers and status messages about the state of the floor objects in the CPU through redundant channels.

MPC EBILock 950 provides safety and control of the movement of trains at stations and hauls of all sizes, configurations and destinations, including stations for connecting different types of traction. The system integrates automatic (ABTC-E) and semi-automatic locking functions, remote management of areas and stations parks, as well as remote monitoring and integration with top-level systems (dispatching centralization and control).

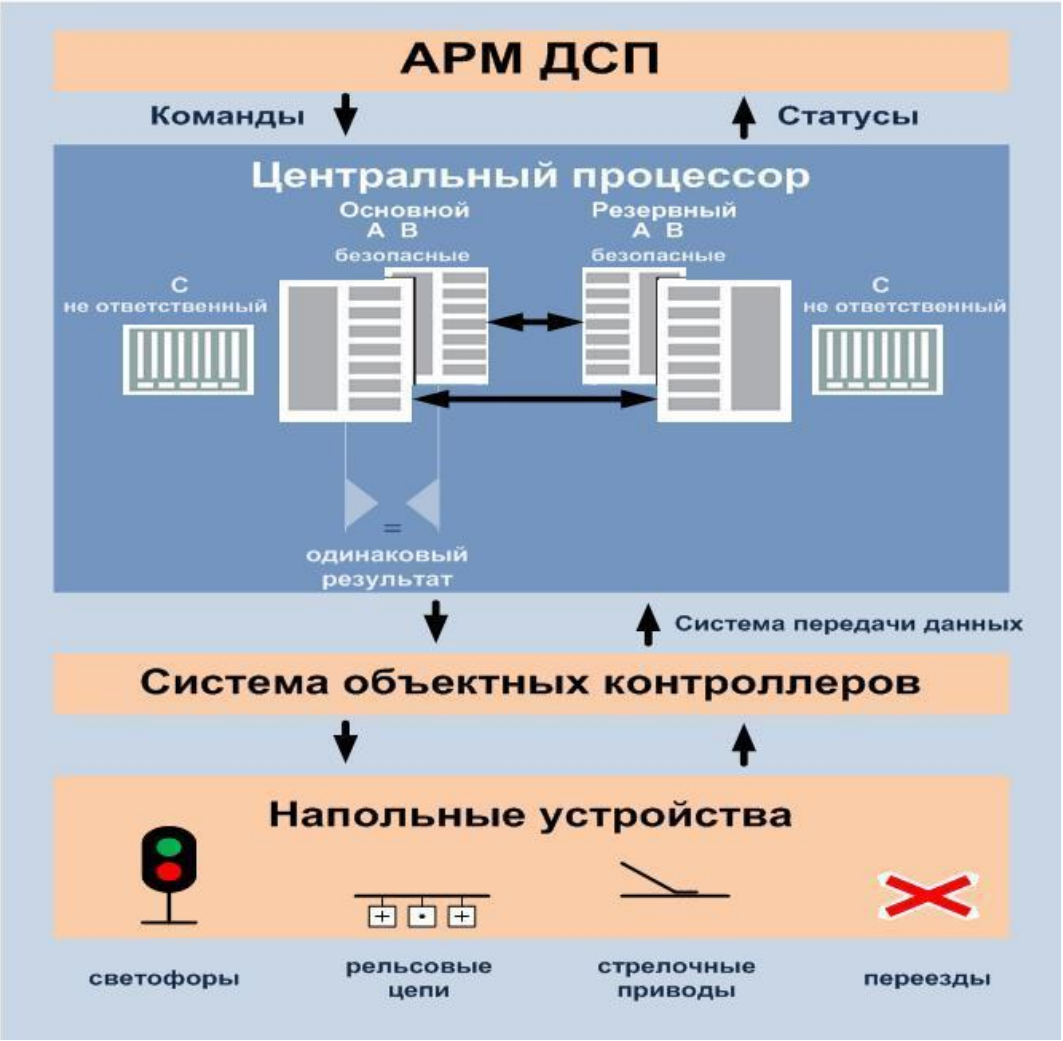


**Figure 1.7. Structural diagram of the MPIC EBILock 950**

Advantages of MPC EBILock 950:

• Full compliance with European and world (CENELEC SIL4) safety standards.

• Non-contact control of arrows and signals based on intelligent object controllers.

• Redundancy of the main components of the system.

• Organization of communication on the loop principle, reservation of the communication channel.

• Advanced diagnostics of the system, allowing to detect the preventative conditions of the equipment.

• Possibility of centralized or decentralized equipment placement.

• High level of availability: application of standard industrial modules, testing of the software and hardware complex is carried out in the factory, the object is supplied with fully tested and debugged equipment.

• Modular construction principle, the ability to increase the number of managed objects.

The computer system Ebilock-950 is developed by specialists of the Swedish company ABB Signal AB to control arrows, signals and other objects at the station. Implemented at more than 100 stations in Sweden, Norway, Poland, Germany, Turkey, Spain and other countries. The Ebilock-950 system, adapted to the conditions of Russian railways, is the main link of the MPC. Outdoor equipment and relay equipment in the MPC are used in Russia.

The MPC is one of the systems of the latest generation and can be applied either as a local system or as part of the ACS by the movement of high-level trains.

The MPC belongs to the class of decentralized systems and is built on a modular basis, where the logic and safety of the trains are carried out in the center, and the equipment for direct control of the objects is placed in the neck of the stations.

The use of the MPC allows, as a rule, to abandon the construction of new technical and technical buildings, to save cable products and energy resources, to link with the automated systems of the highest level, and to reduce operating costs.

The structure of the system includes: control and monitoring equipment; a central processing unit (CPU); modules of container type with concentrators and object controllers (IOC); outdoor equipment STSB.

Management MPC is carried out with ARM DSP, created on the basis of a standard PC.

The operation of the MPC is controlled by the display of objects on the display of the workstation, the control is carried out from the ARM keyboard.

Diagnostics of the MPC and control of technical parameters is carried out from the ARM ShN. The same workstation maintains a protocol of actions of the EAF and the work of the MPC.

The core of the system is a central processing unit that safely implements all the interdependencies accepted for EC arrows and signals. It also interacts with control and monitoring systems (ARM DSP and ARM ShN) and the OK system, which directly control electric drives of arrows, signals, interface relays. Through these relays, the status of the RC state and all systems linked to computer centralization is transmitted, as well as the status of the OK communication system with the central computer and the power supply system.

The MCC consists of two computers that provide the logic of the operation of the MPC and train safety devices. One computer is constantly at work, the second - in a hot standby. Due to the continuous transmission of information from the main computer to the backup operation of the backup computer in the event of a failure, the main computer immediately proceeds.

Both computers are connected through two loops with concentrators located in the IOC. When switching computers, the connection loops are automatically switched from one computer to another.

The main purpose of the MSC is to process the data in such a way as to prevent the execution of dangerous commands from the control system.

The MCP provides:

"The transformation of commands from the control system into orders that are safely transferred to the arrows, signals and other devices;

»Closure of objects in the route;

»Artificial and automatic route opening.

The composition of the MSC and the structural diagram of the interrelationships between the modules are shown in Figure 2.

The MSC of the system consists of two computers. Each computer consists of two hardware processing channels - Processor A and Processor B, which implement the program A and B, respectively. All functions, to which

security requirements are imposed, are realized in two independent logical channels. Functions related to providing an interface with external devices are implemented by the service processor.

A computer that is in a "hot standby" state constantly updates the data, so the existing availability of the system to switch to a backup channel in the event of failures or failures in the main information processing channel.

In the MPC "Ebilock-950" system, the use of diversified programs, that is, programs equivalent in functionality but different in structure, is fundamental in ensuring security.

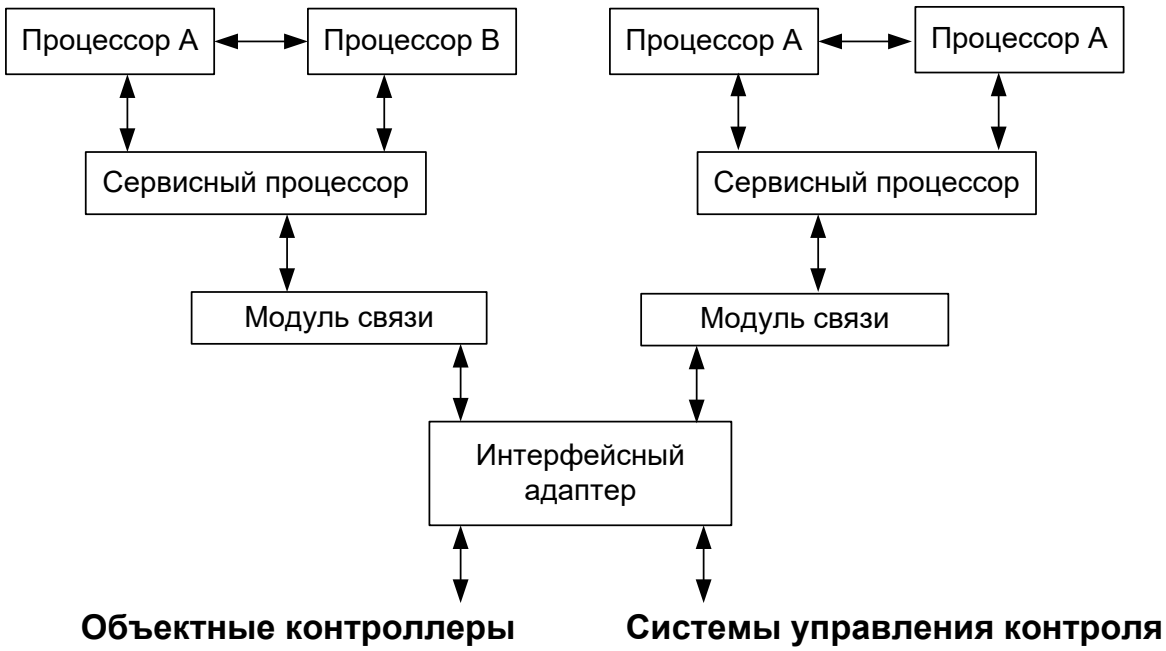


**Figure 1.8. Structural diagram and composition of the CPU**

The diversion of executable programs presupposes, first of all, various reactions of the programs to hardware failures and failures, and as a result, a difference in the calculated intermediate and final results.

In the IOC are concentrators, object controllers, relay equipment of the RC, coding station tracks, blowing arrows, linking with relays and other devices and systems, as well as power supply devices.

To ensure the safe operation of the software and hardware of the PMC in the OC, the following basic principles are implemented:

"Use of diversified programming; »Periodic comparison of input, output and intermediate results; "Management of the program flow;

"Control of time parameters of the computing process;

Memory testing; "Checksums of data, etc.

The system of microprocessor centralization of electrical centralization of the MPC-IPC is a new development in the family of computer systems based on microcomputers of programmable controllers. It is designed to control and control the devices of railway automation at stations using computer technology developed by the PGUPS (St. Petersburg).

The IPC-IGC in 2012 was put into permanent operation at the Khonykh station of Krasnoyarsk Railways.

Information exchange between the components of the system is based on standard protocols of computer systems and local networks. The use of modern standard computer facilities for input and display of information does not require the manufacture of specialized controls and controls.

A distinctive feature of the system from analogs is a secure contactless control and monitoring interface for objects, which is designed on a fundamentally new approach of functional signal transformation.

The equipment of the central computer system (VCS) is 100% redundant and consists of two parallel and independently functioning secure computer sets - "main" and "backup", included in the local computer network. Each of the kits consists of two PC-compatible industrial controllers and a circuit for monitoring the functioning of the kit. Normally, both sets are connected to the code lines of communication with the interface equipment with the control and monitoring facilities of the MPC. One of the kits is active and implements the control influence on the objects and transmits information about the status of the monitored objects via the communication channel to the AWP DSP, and the second set of the DSC is passive and is in the "hot" reserve. The additional function controllers are also reserved.

The workstation on duty at the station is designed to organize a user interface for managing and controlling objects of microprocessor centralization at the station. AWP in the minimal configuration is executed on the basis of two PCs (sets A and B), united by a local network. This network also includes ARM electromechanics, and, if necessary, other users of information on the movement of trains at the station (operator's workstation, shunting, station dispatchers, etc.) can be included. For the departure of the economic train and the pusher for the distillation in the hardware DSP, a switch-key guard is installed. In addition, the ARM chipboard can be equipped with remote plasma panels.

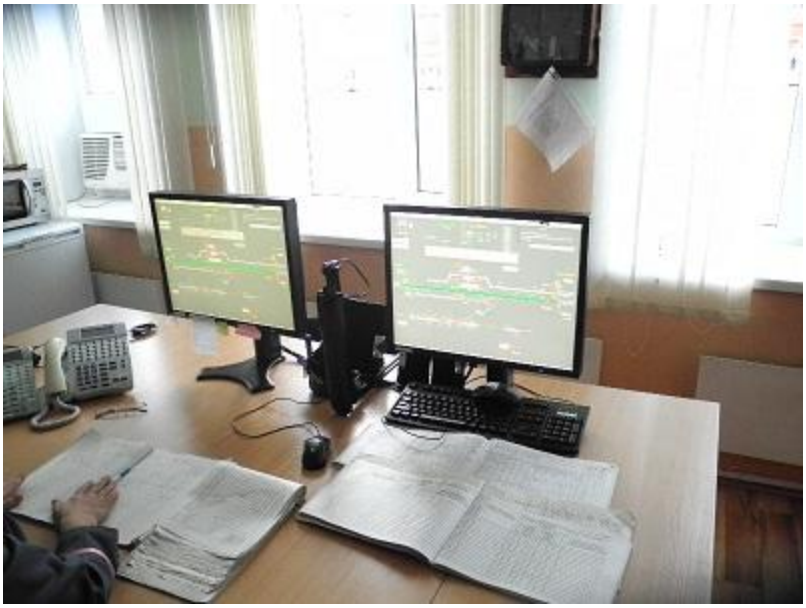

**Figure 1.9. Automated workstation DSP on microprocessor centralized system**

ARM DSP equipment has 100% reserve and consists of two parallel and independently functioning sets - "A" and "B", included in the local computer network. One of the kits is active and implements the control influence on the objects and receives information about the status of the monitored objects via the communication channel from the CCC of the Criminal Code. The second set of AWS ARM is passive, it is used only for displaying current information and is in the "hot" reserve. Both sets in the process of exchanging information among themselves on the LAN.

The system of relay-processor centralization of arrows and traffic lights RPTs-E was developed by LLC Bombardier Transportation (Signal).

The ROC-E system is designed to partially upgrade existing stations with any number of arrows equipped with electrical centralization, both with the preservation of the executive group (all existing standard albums for design), and with the construction of a new executive team, performed on the album MRTS-10BN. The system allows you to keep existing floor equipment in full.

Also, the ROC-E is easily integrated with the EBILock 950 MPC, for example, when building a new park and its equipment with microprocessor centralization devices. At the same time, the DSP receives a single workplace, and the operator manages the devices of the MPC and the EC in the same way.

ROC-E consists of automated workstations DSP and SHN, having all the functions realized in microprocessor centralizations, server RPTS-E, implemented on industrial computers, as well as from distributed USO. The latter are based on industrial controllers in a design that allows them to be placed both on the front and on the mounting side of the cabinet with access to the existing installation. The system has a hot redundancy of all components [5].

In the course of modernization, the set-up group is dismantled (if there is one), and the existing remote control panel. The station is equipped with automated workplaces. The system provides for linking with other systems via data transmission channels. In 2012, the ROC-E was put into permanent operation at the Abakan station of the Krasnoyarsk railroad. (114 points).

**Figure 1.10. RPTs-E equipment**

### 1.4 Review of the shortcomings in the algorithm of the BRRI system

Schemes of the set-up group BRRI are intended for realization of a mar-grunt way of controlling arrows and traffic lights. Therefore, the relays located in the dial-up blocks record the actions of the duty officer at the station on the control panel, automate the transfer of the running and guard arrows along the route route and the opening of one or more traffic lights. The following standard blocks are used in the set group:

• NPM-69-block; board traffic lights with shunting readings (weekend, route); in addition, it is used to control the traffic light and shunting light from the path section behind this entrance traffic light; It is also used for the final train button;

• HMI-control unit for single maneuvering traffic lights, located on the border of two arrow isolated areas; is also used for the variant button;

• NMDD-additional block for six blocks of type HMI: contains six push-button relays of buttons of the control panel;

• NMIIP-control unit of the maneuvering traffic light, permitting movement from the non-centralized zone, as well as for one of the two traffic lights from the road section or for one of the two traffic lights installed in the

alignment (on one ordinate);

• NMIIAP-control unit of the second traffic light from the road section or traffic lights in the alignment; is used in conjunction with the NMNP unit;

• HCOx2-control unit with two single arrows;

• NCC-control unit for paired arrows;

• HH-direction block, fixing the view and direction of the specified routes;

• NPC-block, which controls the consecutive transfer of arrows in the main supply of the working circuits of the BOT; contains three sets of control equipment;

• The BDH-20-unit, intended for the inclusion of angular push-button relays in HCC units, contains diode isolation circuits.

The blocks of the route set are connected in accordance with the topology of the track development of the station neck by four electric circuits.

Table 1

| | Designationinthes cheme | Name | Appointment |
|---|---|---|---|
| | KH, HKH | Circuit of push-button relays | Deactivation of push-button relays: in the blocks of the initial buttons when the PU or MU of the first arrow is triggered, and in the end button blocks - when the relay of the PU or MU of the last arrow |
| | AKH | Circuit of automatic push-button relays | Enabling of automatic push-button relays in intermediate HMI and HIIPAP units, if the buttons of these |

| | | | |
|---|---|---|---|
| | | | blocks were not pressed |
| | ПУ, МУ | Switch-controlrelaycircuit | Switching the control relay switches on the route elements (relay PU and MU between two adjacent buttons) |
| | СС | MatchingChart | Checking the correct execution of the control commands (contacts of the relay PU or MU) for the transfer of the running guard arrows (PC or MK relay contacts) |

**Table 1.Name of the main circuits of the route set**

He routing set is basically intended for the day of automation of the translation of the arrows in the established route, therefore in the blocks of the route set the code relays of the KDR type are used. Let's consider their designations, types, performed functions, p also the moments of switching on and off of these relays.

Pushbutton switches K (KDR1- 280) -rele-repeaters of buttons; They are installed in the blocks НМ1Д, NMIIP, НМIIАП; triggered when the corresponding button is pressed to check the power supply in the PC pole (SPB-K); disable when the button is released.

Push-button relays ICH (KDR1 280) are installed in NPM-69 and HMI blocks; in blocks NPM-69 turn on when pressing the train buttons; in blocks HMI - when pressing buttons as initial or variant; if the HMI block is intermediate and the button of this block is not pressed, then the CK relay is switched on via the contact of the DCS relay; the relay switches off when the relay is activated.

Push-button relays KN (KDR1-280) are located in blocks NPM-69, HMI, NMPP, NMPA; in blocks NPM-69 are switched on when pressing the manevrovye buttons; in blocks HMI-when pressing buttons as final or variant; in blocks NMIIP and NMIIAP-by pressing the corresponding buttons as initial, variant or final; Relays CN turn off when the relay PU, MU.

Automatic push-button relays AKN (KDR1 M-3,8) are installed in the HMI and NMPA units; are intended for the inclusion of button relays in HMI, NMIIP and NMIIAP blocks when specifying routes containing more than one element if the buttons of these blocks were not pressed; turn off when disconnecting the button relays ICH and CN start the route.

Angular pushbutton switches UK (KDR1 M-435) are located in the blocks of the NSS; Are intended for a choice of a route of the basic route at presence of variants of movement; are switched on and off by the contacts of the button relays NKN and CN in the blocks of the beginning and end of the route.

The direction relay P, O, ПМ, ОМ (КДР1 М-120) are installed in the block of direction HH; are used to fix the beginning, type and direction of movement in the train and shunting routes; Relays P and O are switched on via the contacts of the button relays of the IC of the start-of-route block, and the relay PM and OM-through the contacts of the auxiliary relays VPM and PTO; Turn off when all button switches are turned off.

Auxiliary relays of the direction of the RPM and PTO (KDR1-280) are installed in the LV direction block; are designed to activate the PM and OM relays when the initial shunt button is pressed, and to turn off the PC power pole (SPB-K) when pressing the option button or with auxiliary control; turn off when the corresponding button relay is turned off.

Reverse relay OP, PP, MP (KDR1 M-280) are located in the blocks NPM-69, HMI, NMPP, NMPA; They are designed to activate alarm relays, exclude spontaneous reopening of the traffic light after using the route; are included in the initial button block from the corresponding direction bus; are switched off by

the rear contact of the alarm relay after the opening of the traffic lights; After switching off the relays, the routing set comes to the initial state.

Auxiliary terminal relays VK, VKM (KDR1 M-435) are located in the blocks NPM-69, HMI, NMP, NMIIAP; are intended to determine the end of the route; The VKM relay includes the final shunting relays of the CM of the executive group; are included in the end button block; turn off by the contact of the closing relay 3 of the last section of the route.

The auxiliary intermediate relays of the VP (KDR1 M-435) are located in the blocks HMI, NNPP, NMPA; They are intended for closing the circuits of the relay PU and MU by the elements of the route; turn on the contacts of the button relays of the intermediate blocks from the direction buses; turn off by the contact of the closing relay Z.

The control relay arrows PU, MU (KDR1 M-3,8) are installed in the blocks of HNS and HCC; They are used to launch running and guard arrows along the route; are switched on by the relay contacts OP, MP, VC, VCM, VP by the route elements and are switched off when they are switched off

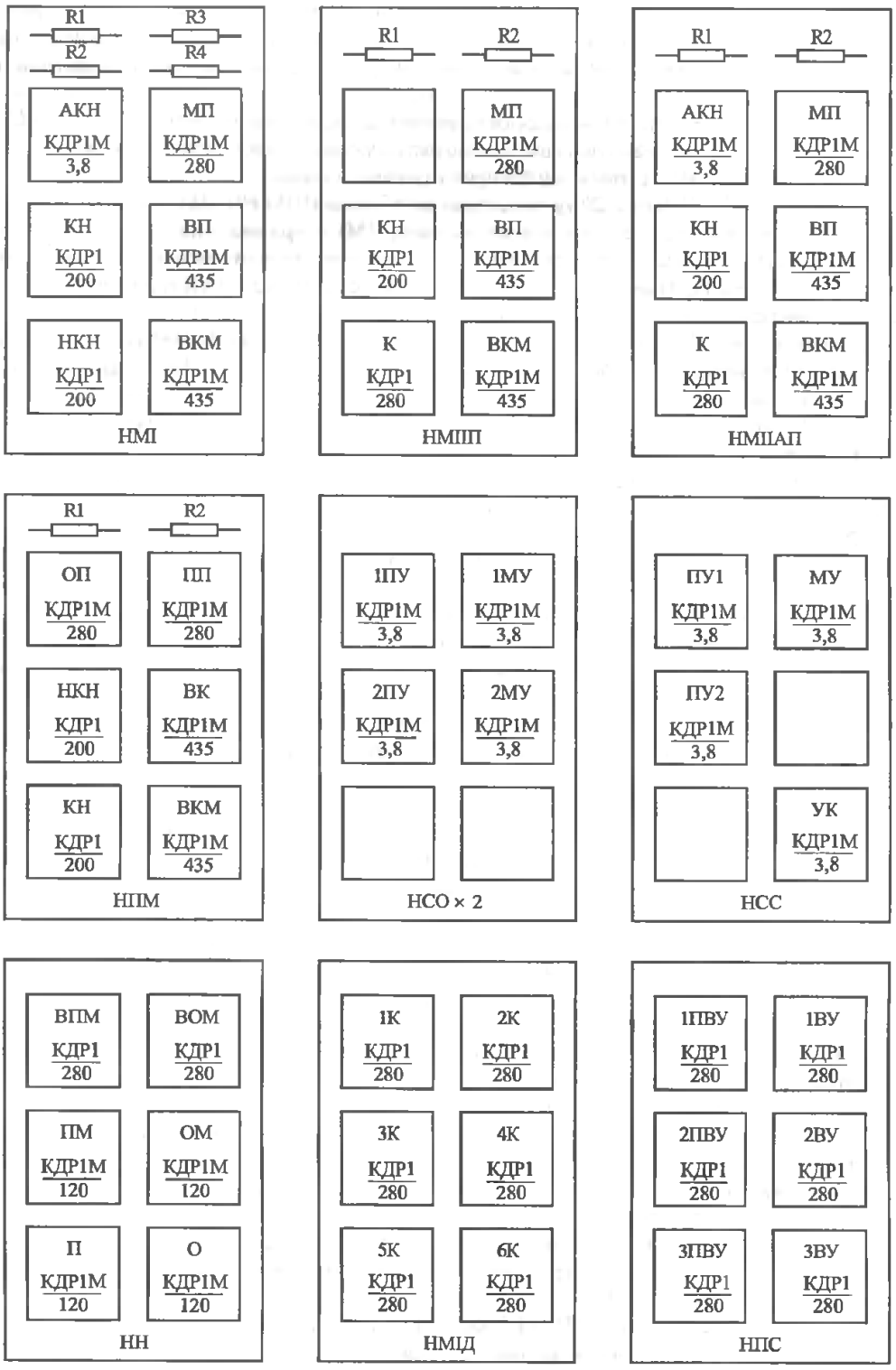


**Figure 1.11. The arrangement of devices in the blocks of the dial-up group**

In the typical blocks of the dial-up group, the following relays are used to dial the route in the key-relay relay scheme: the repeater of the traffic light K, the push-button KN and ICH; counterparts PP, OP, MP; an auxiliary finite

shunting VKM; auxiliary final train VC; auxiliary intermediate VP; automatic button AKN.

Reverse relays determine the beginning of the route and its genus in the route dialing schemes. These include:

- PP - train the opposite relay;
- OP is the common opposite relay;
- MP - shunting relay. The contacts of the opposite relays participate in the inclusion of the initial ones (the scheme of correspondence is the fourth chain of the route set), the control section (the first circuit of the executive circuits) and the signal relays (the second circuit of the executive circuits).

When setting the shunting route, pressing the button will turn on the button relay KN, the PM direction relay, which supplies power to the FM bus. The OP relay in the NPM block is triggered from this bus. After switching off the button relays, the OD and PP remain on via the rear contact of the button relays and its own front-line contact: The auxiliary relay contacts VKM, VK and the contacts of the opposite relays MP, OP and the PP from the common for the whole station route dialing scheme allocate a part related to the set route.

To set routes by the main variant, only two buttons (the beginning and the end of the route) are used to apply the scheme of automatic button relays. The circuit of switching on the AKN relay forms the second circuit of the general scheme of the route set. The AKN relay is located in the blocks NMI and NMIIAP. The AKN circuit passes through terminals 1-2 and 2-2 of the route block blocks.

To supply the circuit of the DCS relay to all signal blocks it is customary to send the MI pole from the even direction, and from the side of the odd one, P.

Power to the AKN circuit is supplied through the contacts of the button switches of the beginning and the end of the route, the opposite relay at the beginning of the route and the auxiliary end relay at the end.

The delay in the activation of the ACK relay before the operation of the relay OP and VKM provides the configuration of the circuit by the contacts of the relay UK.

The AKN relay switches off after de-energizing the button relays.

The correspondence scheme is the fourth circuit of the route set and serves to include the initial relays in the executive group and to verify the correspondence between the actual position of the arrows and the command for their translation. Compliance is checked by closing the contacts PU - PC and MU - MK [7].

The need for a matching scheme is caused by the fact that the task of translating arrows and the task of opening a traffic light are issued simultaneously. Therefore, in the absence of this scheme, the route could be established according to the positions of the arrows that were preserved from the previous route.

The beginning of the route in the match chain is determined by the contact of the opposite relay, the end of the train route by the VC contact, and the end of the shunting route by the VKM contact.

At the end of the shunting route, the contact of the VKM relay switches on the KM final shunt relay, which determines the end of the route in the circuits of the relay executive group. In the KM relay circuit, the closing relay of the last section in the route is entered.

## 2. SOFTWARE DEVELOPMENT FOR BUTTON RELAYS

### 2.1 Examined technical task specific software through the MS WPF study based on the C # language

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as web sites, web apps, web services and mobile apps. Visual Studio uses Microsoft

software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML and CSS. Support for other languages such as Python, Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

### 2.1.1 Architecture of Visual Studio

Visual Studio does not support any programming language, solution or tool intrinsically; instead, it allows the plugging of functionality coded as a VSPackage. When installed, the functionality is available as a Service. The IDE provides three services: SVsSolution, which provides the ability to enumerate projects and solutions; SVsUIShell, which provides windowing and UI functionality (including tabs, toolbars and tool windows); and SVsShell, which deals with registration of VSPackages. In addition, the IDE is also responsible for coordinating and enabling communication between

services.[9] All editors, designers, project types and other tools are implemented as VSPackages. Visual Studio uses COM to access the VSPackages. The Visual Studio SDK also includes the Managed Package Framework (MPF), which is a set of managed wrappers around the COM-interfaces that allow the Packages to be written in any CLI compliant language. However, MPF does not provide all the functionality exposed by the Visual Studio COM interfaces. The services can then be consumed for creation of other packages, which add functionality to the Visual Studio IDE.

Visual Studio supports running multiple instances of the environment (each with its own set of VSPackages). The instances use different registry hives (see MSDN's definition of the term "registry hive" in the sense used here) to store their configuration state and are differentiated by their AppId (Application ID). The instances are launched by an AppId-specific .exe that selects the AppId, sets the root hive and launches the IDE. VSPackages registered for one AppId are integrated with other VSPackages for that AppId. The various product editions of Visual Studio are created using the different AppIds. The Visual Studio Express edition products are installed with their own AppIds, but the Standard, Professional and Team Suite products share the same AppId. Consequently, one can install the Express editions side-by-side with other editions, unlike the other editions which update the same installation. The professional edition includes a superset of the VSPackages in the standard edition and the team suite includes a superset of the VSPackages in both other editions. The AppId system is leveraged by the Visual Studio Shell in Visual Studio 2008.

### 2.1.2. Debugger

Visual Studio includes a debugger that works both as a source-level debugger and as a machine-level debugger. It works with both managed code as well as native codeand can be used for debugging applications written in any language supported by Visual Studio. In addition, it can also attach to running processes and monitor and debug those processes. If source code for the running

process is available, it displays the code as it is being run. If source code is not available, it can show the disassembly. The Visual Studio debugger can also create memory dumps as well as load them later for debugging. Multi-threaded programs are also supported. The debugger can be configured to be launched when an application running outside the Visual Studio environment crashes.

The debugger allows setting breakpoints (which allow execution to be stopped temporarily at a certain position) and watches (which monitor the values of variables as the execution progresses). Breakpoints can be conditional, meaning they get triggered when the condition is met. Code can be stepped over, i.e., run one line (of source code) at a time. It can either step into functions to debug inside it, or step over it, i.e., the execution of the function body isn't available for manual inspection. The debugger supports Edit and Continue, i.e., it allows code to be edited as it is being debugged. When debugging, if the mouse pointer hovers over any variable, its current value is displayed in a tooltip ("data tooltips"), where it can also be modified if desired. During coding, the Visual Studio debugger lets certain functions be invoked manually from the Immediate tool window. The parameters to the method are supplied at the Immediate window.

## 2.2 **Design of K relaycircuitwork sequnce software**

Relay K – repeater of traffic switch button. It serves to multiply contacts. To change its state to under the current, get power from the bus PC through its own tail contact and self-blocking through the power jumper P, till releasing the traffic light button.

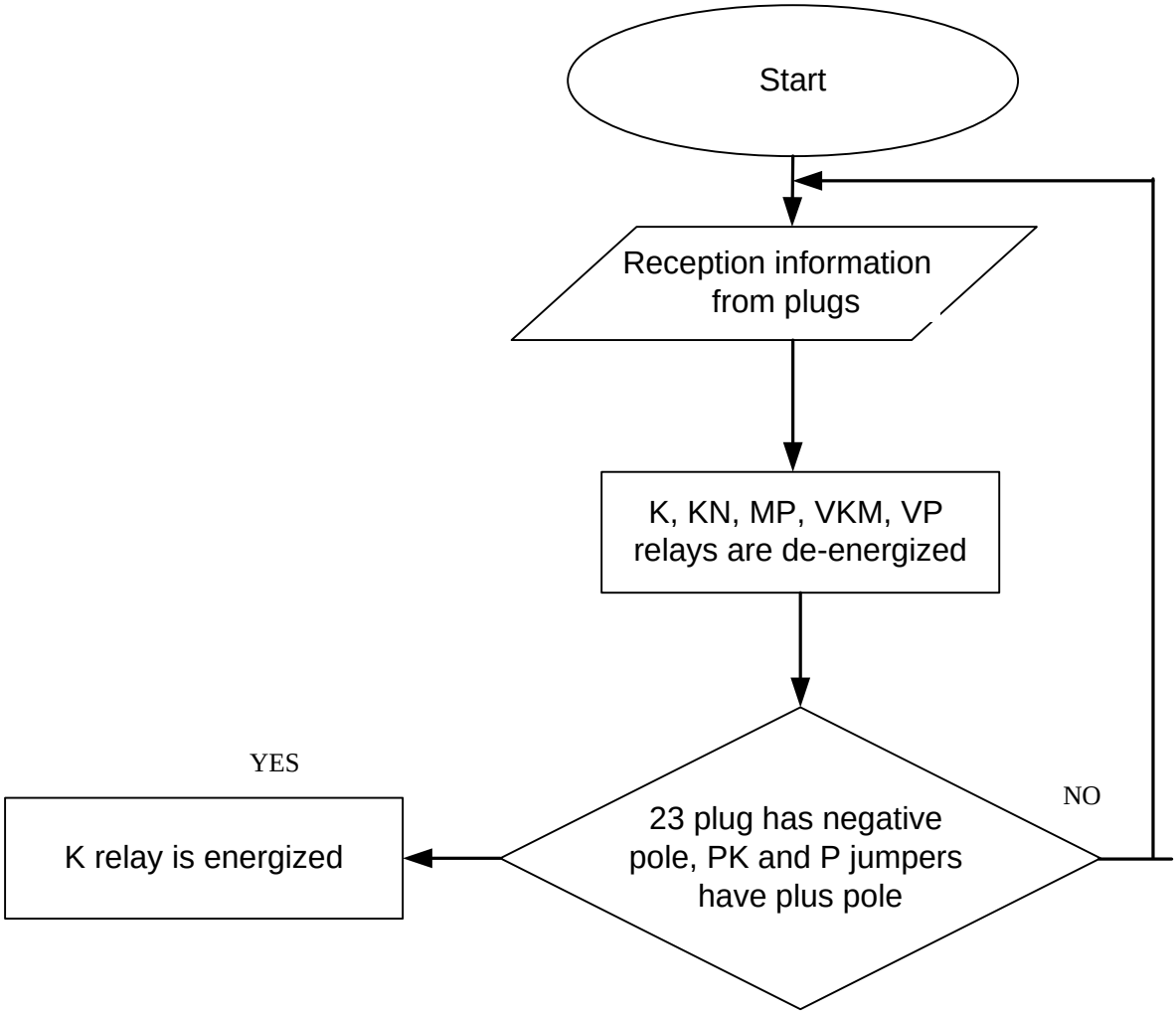


**Figure 2.1.Algorithmof energizing relay K**

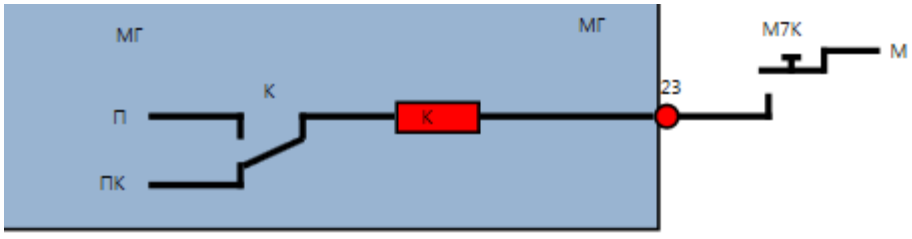


**Figure 2.2.Immitation of relay K in Microsoft Visual Studio**

```
{

if ((button_m7k.IsPressed == true&&Check_pk.IsChecked ==
true&&Check_p.IsChecked == true&&Check_m.IsChecked == true)
              )
              {
k_blok.Fill = green;
            }
else
              {
k_blok.Fill = red;
                ;   }
```

There was given fragment of code in language C#. These are conditions of energizing relay K .

### 2.3 Design of KN relaycircuitwork sequnce software

Pushbutton relays serve to fix the pressing of the button on the control panel.

Pushbutton relays are normally located without current and are switched on:

- from pressing the buttons;
- through contacts of automatic push-button relays (ACN).

Push-button relays are powered by the first interconnectionand are switched off after the operation of the control relays PU or MU or the alarm relay. The contact of the alarm relay is necessary in order to turn off the button relay when the signal is reopened, since the repeated opening is by pressing one button and the relay PU and MU do not work.

Power PC turns off after pressing two buttons of the route and thus completes the collection of the elementary route. Pressing the third buttondoes not lead to the activation of the button relay until the moment of de-energizing all KN relays between the two first buttons. This is necessary so that relay K does not turn off when the PC power is turned off if the button remains pressed. Otherwise, the relay K would work again after the power supply of the PC appeared and would switch on the relay KN of the beginning of the route through this traffic light.

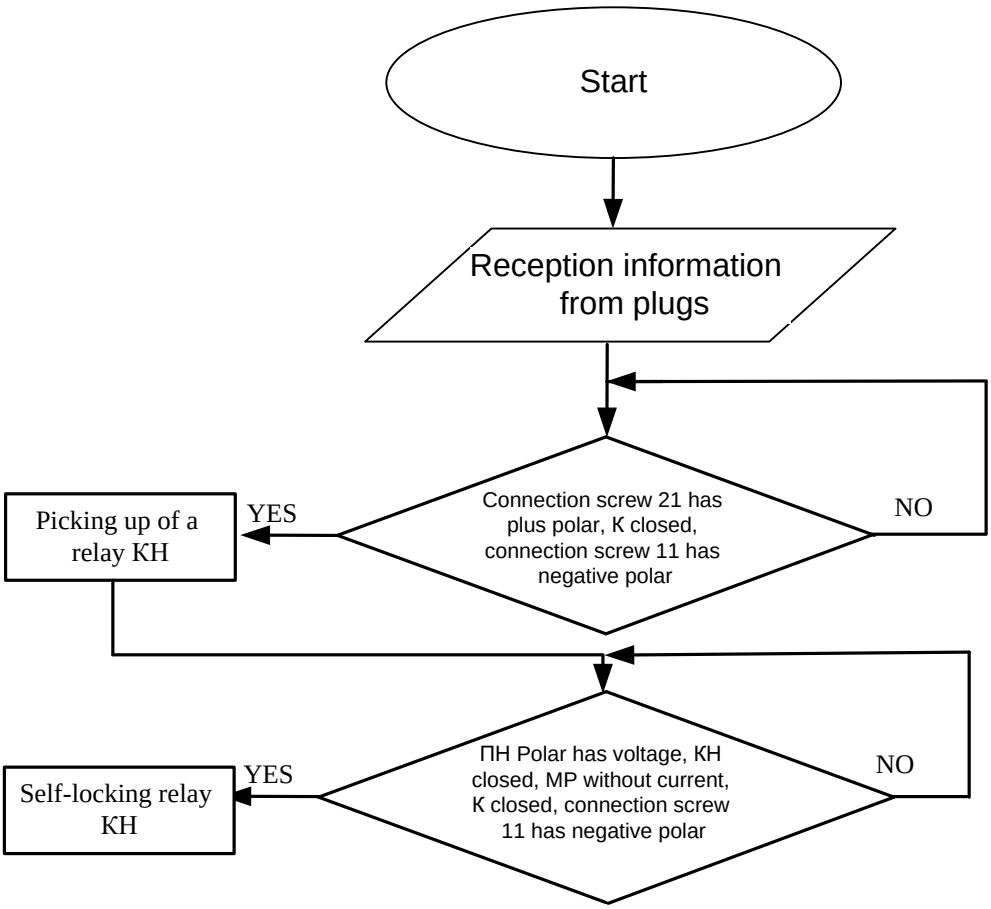


**Figure 2.3.Algorithm of energizing and self blocking relay KN**

```
if ((k_blok.Fill == green && Check_11.IsChecked ==
true&&Check_tchm.IsChecked == true)
```

```
                || (Check_11.IsChecked == true&&Check_pn.IsChecked ==
true&& kn_p1.Visibility == Visibility.Visible&& mp_m2.Visibility ==
Visibility.Visible)
                || (Check_28.IsChecked == true&& Check_11.IsChecked ==
true&& k_m3.Visibility == Visibility.Visible))
            {
kn_blok.Fill = green;

            }

else
            {
kn_blok.Fill = red;

                }
```

There was given fragment of code in language C#. These are conditions of energizing relay KN .

# 3. COMPUTER SOFTWARE PROGRAMMING FOR RELAYS THAT DETERMINES POSITION OF SHUNTING OPERATION

## 3.1. Design of MP relaycircuit work sequnce software

Anti-reverse (anti-repetative) relays determine the beginning of the route and its genus in the route dialing schemes. These include:

-PP - train the opposite relay;

-OP - common opposite relay;

-MP is a shunt relay.

Anti-repetitive relays MP are installed in blocks of maneuvering traffic lights HMI, HMIIП, HMIIAП.

The contacts of the anti-repetitive relays are involved in the inclusion of the initial relays (the correspondence scheme is the fourth chain of the route set), the control circuits (the first circuit of the executive circuits) and the signal relays (the second circuit of the executive circuits).

In box NMIIP MP relay energizes in this way:

$$ПГ \rightarrow \overline{КН} \rightarrow \underline{С} \rightarrow \overline{КН} \rightarrow \overline{МП} \rightarrow \left|КН\right| \rightarrow \underline{ПУ} \rightarrow М\,,$$

There are tail contact of signal relay might be for safety, if traffic switch opened and front contact of signal relay is closed – anti-repetetive relay wouldn’t be energized.

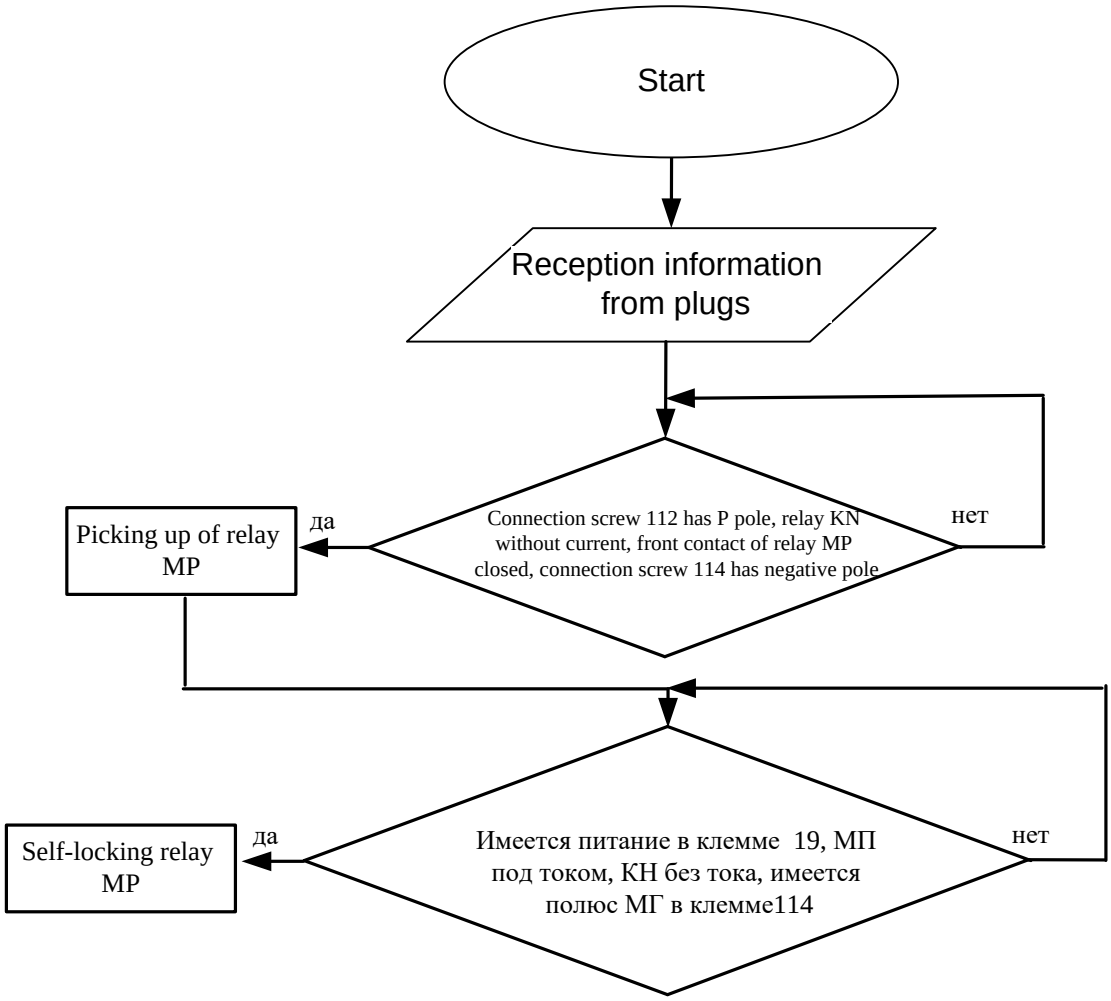


**Figure 3.1.Alorithm of picking up and self-lockin relay MP**

Part of code for NMIIP box writed in language C# for picking up and self-locking relay MP

```
if ((kn_p3.Visibility == Visibility.Visible&& Check_214.IsChecked == 
true&&Check_mg.IsChecked == true)
                || (Check_19.IsChecked == true&& mp_p1.Visibility == 
Visibility.Visible&& kn_m3.Visibility == 
Visibility.Visible&&Check_mg.IsChecked == true))
            {
mp_blok.Fill = green;


            }
else
```

```
            {
zamedleniya.Start();
                    }
```

And code for it contacts

```
 {
if (mp_blok.Fill == green)
                {
                   mp_p1.Visibility = Visibility.Visible;
                   mp_p2.Visibility = Visibility.Visible;
                   mp_m1.Visibility = Visibility.Hidden;
                   mp_m2.Visibility = Visibility.Hidden;
                }
else
                {
                   mp_p1.Visibility = Visibility.Hidden;
                   mp_p2.Visibility = Visibility.Hidden;
                   mp_m1.Visibility = Visibility.Visible;
                   mp_m2.Visibility = Visibility.Visible;
                }
                    }
```

### 3.2 Design of VKM relaycircuit  work sequnce software

The auxiliary end relays determine:

-VK - the end of the train route;

-VKM is the end of the shunting route.

The VKM relay is installed in the blocks of shunting signals HMI, HMIIП and HMIIAП, and in the NPM block – relay VK and VKM. These relays turn on when the button is not pressed first, and serve to turn on the circuit of automatic push-button relays (ACN) and switch control relays (PU and MU), and in the

executive group, the end-of-line relay switches are activated by the contacts of the VKM relay.

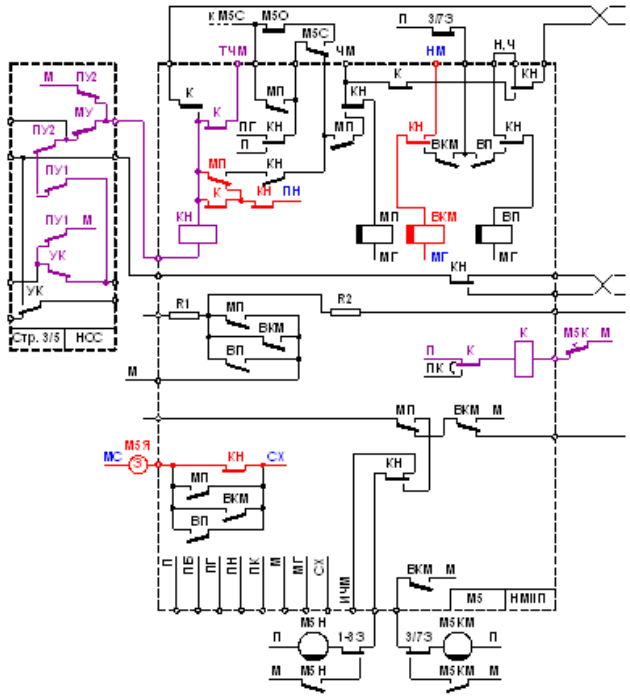

**Figure 3.2. Block NMIIP at the moment of excitation of the VKM relay.**

For imitation slowdown to the dropout relay armature used operator EvenHandler. There are will be shown code of simulation slowdown to the dropout relay armature:

```
publicMainWindow()
    {
InitializeComponent();

tik.Tick += newEventHandler(sikl);
tik.Interval = newTimeSpan(0, 0, 0, 0, 100);
tik.Start();

zamedleniya.Tick += newEventHandler(sekinlash);
zamedleniya.Interval = newTimeSpan(0, 0, 0, 0, 100);
```

```
            zamedleniya_2.Tick += newEventHandler(sekinlash_2);
            zamedleniya_2.Interval = newTimeSpan(0, 0, 0, 0, 10);


        }
privatevoid sekinlash_2(object sender, EventArgs e)
        { }
privatevoidChs_Activated(object sender, EventArgs e)
        {
tik.Tick += newEventHandler(sikl);
tik.Interval = newTimeSpan(0, 0, 0, 0, 100);
tik.Start();
        }
publicstaticbool btn_1flag, btn_2flag;


int n;
intsek_vaqt = 100;
publicvoidsekinlash(object sender, EventArgs e)
        {
            n = n + 1;
if (n == sek_vaqt)
            {


mp_blok.Fill = red;
vkm_blok.Fill = red;
vp_blok.Fill = red;
                n = 0;
zamedleniya.Stop();
            }       }
```

**3.3. Design of VP relaycircuit  work sequnce software**

If a train route or a shunting route is passed by the maneuvering signal, an auxiliary intermediate relay of the VP is switched on in the block of this traffic light (the relay does not work in the unit and in the VMM). In the circuit of relay VP, the switching of the button relays of both elementary paths is checked.contacts of the relay VP switch on the relay AKN and switch control relays of both ends of the elementary routes.

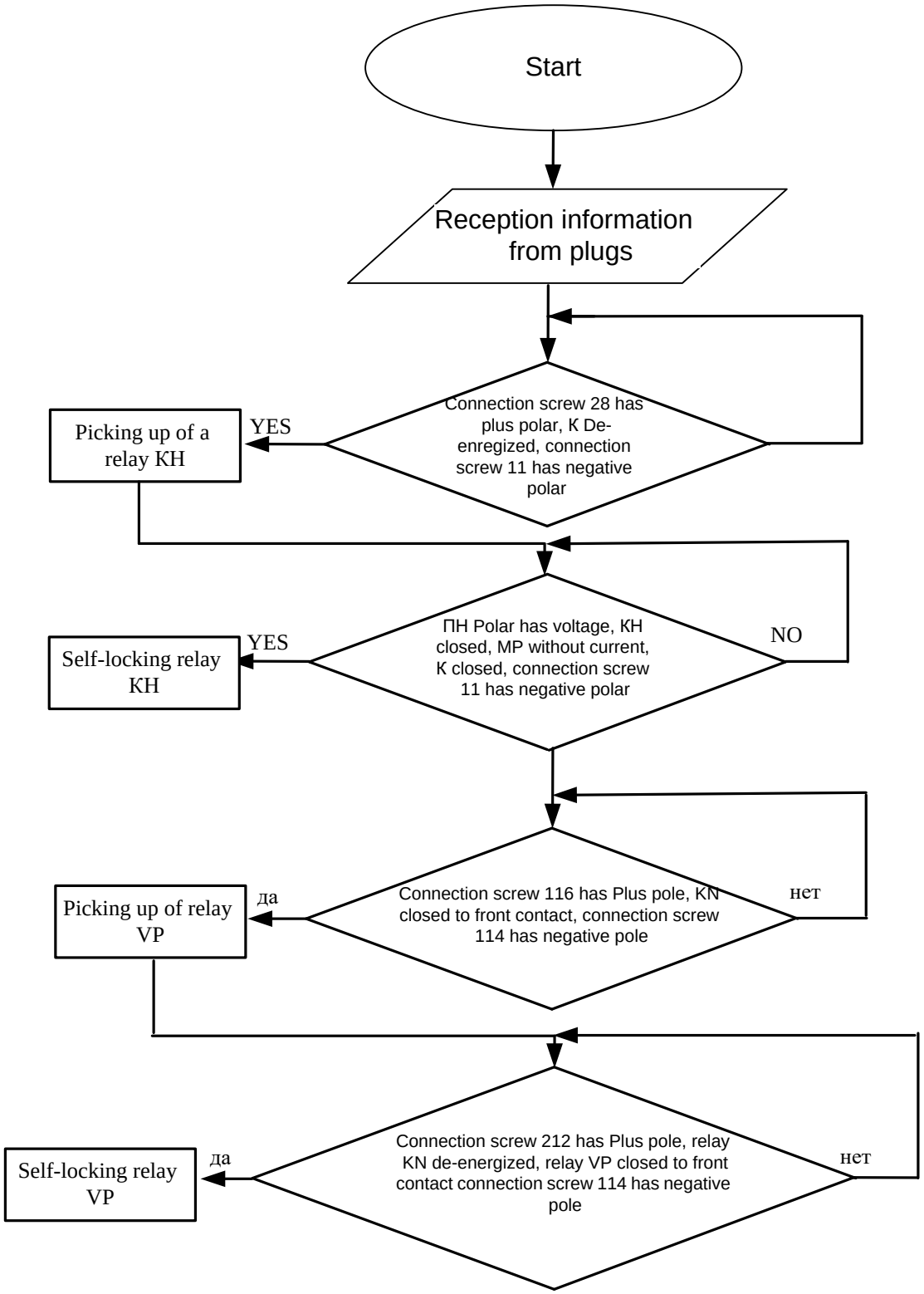


**Figure 3.3. Algorithm of picking up and self-locking of relay VP**

### 3.3 Studying and analyzing developed software

When an application is developed for simulation purposes, a thorough test becomes especially important. If a simulation is not conducted properly the application becomes completely useless.

Tests have been performed on the model and presentation throughout the project and thus defects have been corrected continuously. The tests in this section is concerned with the test performed on the final version of the application, i.e. tests performed after the implementation phase is completed.

It is a well known rule of thumb that defects in an application is cheaper to correct the sooner they are discovered in a development process. Most often this guideline is used for development projects that stretches over several years and the rule is followed to take out financial costs. The rule can still be applied in this project though since the time spent can be reduced if defects are identified as soon as possible.

The test strategy chosen for the application combines structural and functional testing with emphasis on the tests performed on the model.

In the following sections the specific test strategies for the model and presentation are described.Tests should be planned and executed as early as possible in a development project.

It is definitely an advantage to automate tests so they can be performed as often as needed. JUnit has been used to automate tests in the model. For each method in every class in the model a structural test has been performed. This means all branches in the code has been tested to assure that the code does what it is expected to do.

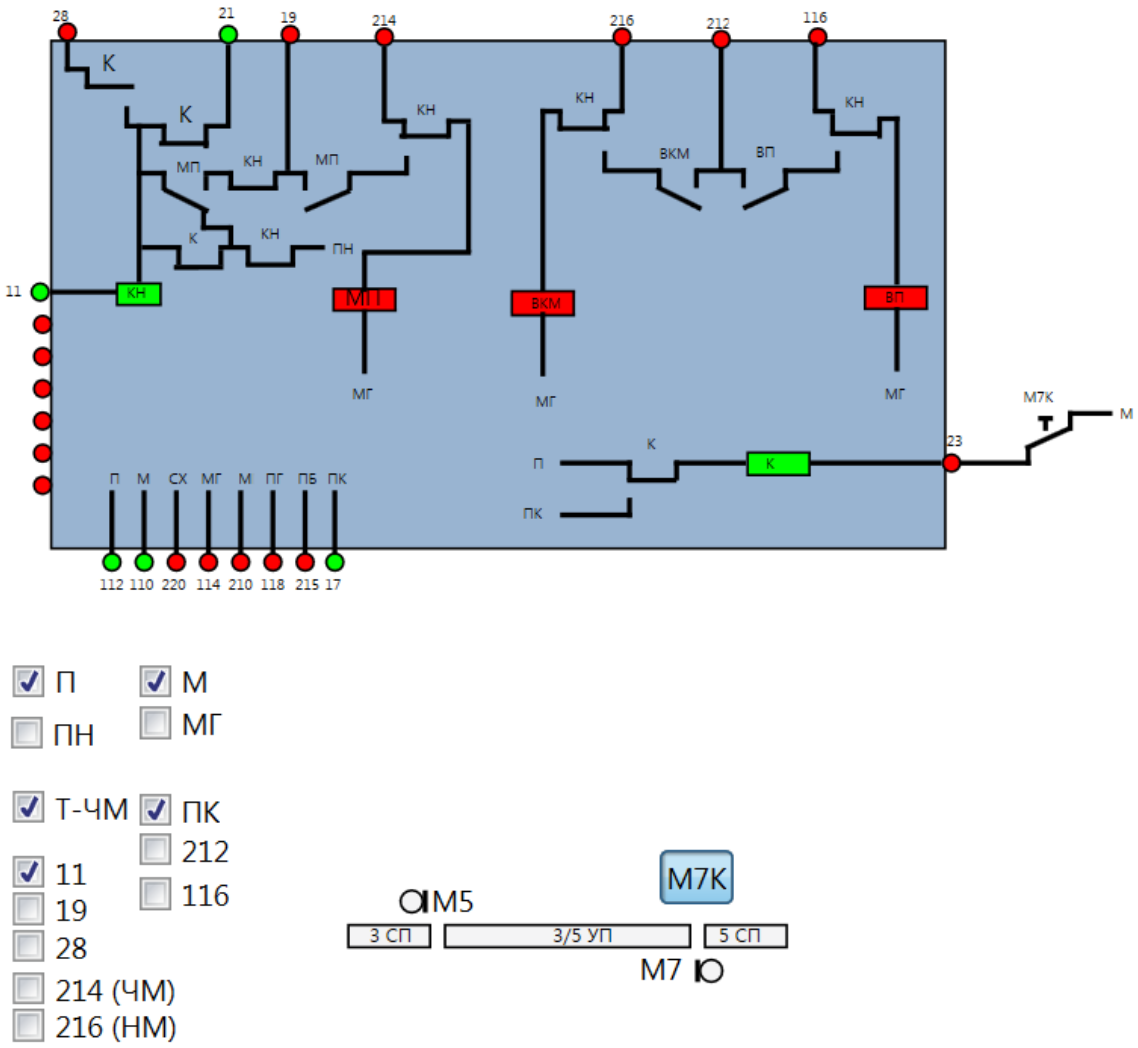


**Figure 3.4. Main window of simulator NMIIP box**

## CONCLUSION

It is well-known that today, the fast and safe delivery of large quantities of cargo and passengers to the address is one of the highest goals of JSC "Uzbekiston Temir Yullari". Therefore, security is one of our priorities in ensuring the effective use modern and advanced technologies in the process of operation and transportation. To do so, we need to equip our railways with equipment and systems that meet modern requirements.The introduction of microprocessor devices to the development of the railroad automation and telemechanics system opens great opportunities. The aim of the master's research is the introduction microprocessor systems torailway automation and telemechanics. Information about for first chapter - Dissertation collected from the department, libraries and the Internet. Based on the information gathered:

- explored features of the boxed relay route interlocking system;
- entirely explored working algorithm of the BRRI system;
- Investigate working algorithm of relays' K, KN, MP, VKM, VP in boxes NMIIP;
- Improvement working algorithm of NMIIP box in the BRRI system;
- As a result of these works, one scientific article was prepared, published at the scientific - practical conference and published the thesis.

In the second and third parts of the dissertation have carried out researches on algorithms and software of push-button relays and relays that determines position of shunting operation in NMIIP block. At the end of these studies, the following results were achieved:

- designed working sequence of K relay software;
- designed working sequence of KN relay software;
- designed working sequence of MP relay software;
- designed working sequence of VKM relay software;
- designed working sequence of VP relay software.

I would consider it necessary to list the works carried out at the time during working on dissertation of master degree:

- examined four current chains of dial group in boxed relay route interlocking system;
- made algorithm of sequence of NMIIP box which in set group;
- developed software for K, KN, MP, VKM, VP relays in language C# with simulation window.

## Bibliography


1. Горбунов В.Л., Панфилов Д.И. Микропроцессоры: Лабораторный практикум: Учеб. Пособие для вузов / Под ред. Л.Н. Преснухина. М.: Высшая школа, 1984. 104с.
2. Сапожников Вл.В. и др. Микропроцессорные системы централизации. – М.: ГОУ «Учебно-методический центр по образованию на железнодорожном транспорте», 2008. – 398 с.
3. Си Шарп: Создание приложений для Windows/ В. В. Лабор.— Мн.: Харвест, 2003. -384 с
4. Арипов М.М., Мухаммадиев Ж.У. Информатика, информацион технологиялар. –Т.: Тошкент давлат юридик институти, 2004. - 278 б.
5. Кондратьева Л.А. Устройства железнодорожной автоматики и телемеханики. – М.: Транспорт, 1983. – 232с.
6. Валиев Ш.К. Изучение и исследование схем блочной маршрутной централизации. – Екатеринбург.:УрГУПС, 2009. – 140 с.
7. Кононов В.А. Изучение наборной группы блочной маршрутно-релейной централизации. Методические указания по выполнению лабораторных работ. – Санкт-Петербург.: ПГУПС, 2007. – 28 с.
8. Микропроцессорные комплекты интегральных схем: Состав и структура: Справочник / В.С. Борисов, А.А. Васенкова и др.; Под ред. А.А. Васенкова, В.А. Шахнова. М.: Радио и связь, 1982. 192с.
9. Проектирование цифровых систем на комплектах микропрограммируемых БИС / С.С. Булгаков, В.М. Мещеряков, В.В. Новоселов, А.А. Шумилов; Под ред. В.Г. Колесникова. М.: Радио и связь, 1984. 240с.
10. Белов А.В. Конструирование устройств на микроконтроллерах. – СПб.: Наука и Техника, 2005. – 256 с.: ил.
11. Николаенко. М.Н. Самоучитель по радиоэлектронике. – М.: НТ Пресс, 2006. – 224 с.: ил
12. Гуртовцев А.Л., Гудыменко С.В. Программы для микропроцессоров:

Справ. Пособие. – Мн.: Выш.шк., 1989. – 352с.: ил.

13. Бродин В.Б., Калинин А.В. Системы на микроконтроллерах и БИС программируемой логики – М.: Издательство ЭКОМ, 2002 – 400 с.: илл.
14. Майоров С.А. и др. Введение в микроЭВМ - Л.: Машиностроение. Ленингр. отд-ние, 1988.-304 с.: ил.
15. Шпак Ю.А. Программирование на языке С для AVR и PIC микроконтроллеров. – К.: «МК-Пресс», 2006. – 400 с.
16. Системы телеуправления на железнодорожном транспорте. Брижака Е.П. – М.: Маршрут, 2005
17. Язык программирование C# и платформа .NET 4. Эндрю Троелсен, Я.Волкова, А.Моргунова, Н.Мухина. Вильямс – 2010 год.
18. C# 4.0. Полное руководство. Герберт Шилдт, И.Берштейн. Вильямс – 2013 год.
19. translate.google.ru
20. http://dssp.petrsu.ru/~IVK/zhirin/inf/inf/read12
21. https://ru.wikipedia.org/wiki/
22. https://www.microsoft.com/
23. http://nullpro.info/2013/samouchitel-po-c-dlya-nachinayushhix-01-osnovy-yazyka-peremennye-logika-cikly/
24. http://informaks.narod.ru/algo_baz.htm/
25. https://ru.wikipedia.org/wiki/
26. m.translate.ru
27. www.scbist.com

APPLICATION FOR DISSERTATION WORK OF MASTER'S DEGREE
STUDENT IRMUKHAMEDOV IBROKHIM ABDULJAMIL

```
<Window x:Class="WpfApplication5.MainWindow"
 xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
 xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
 Title="MainWindow" Height="350" Width="525">
<Viewbox>
<Grid Width="667" >
<Grid.ColumnDefinitions>
<ColumnDefinition Width="0*"/>
<ColumnDefinition/>
</Grid.ColumnDefinitions>
<Rectangle Grid.ColumnSpan="2" HorizontalAlignment="Left" Height="206"
Margin="33,22,0,0" Stroke="Black" StrokeThickness="1"
VerticalAlignment="Top" Width="362" Fill="{DynamicResource {x:Static
SystemColors.ActiveCaptionBrushKey}}"/>
<Canvas x:Name="station" Height="100" Margin="150,325,328,6"
Grid.ColumnSpan="2">
<Canvas x:Name="svetofor_N" Height="24" Width="50" Canvas.Left="25"
Canvas.Top="58">
<Ellipse Fill="#FFF4F4F5" Height="10" Canvas.Left="-1" Stroke="Black"
Canvas.Top="-18" Width="10"/>
<Rectangle Fill="#FFF4F4F5" Height="2" Canvas.Left="5" Stroke="Black"
Width="9" Canvas.Top="-14" RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Label Content="M5" Canvas.Left="7" Canvas.Top="-27" Height="24"/>

</Canvas>
<Rectangle Fill="#FFF4F4F5" Height="10" Canvas.Left="42" Stroke="Black"
Width="100" Canvas.Top="54"/>
<Canvas x:Name="svetofor_m" Height="26" Width="49" Canvas.Left="117"
Canvas.Top="60">
<Ellipse Fill="#FFF4F4F5" Height="10" Canvas.Left="28" Stroke="Black"
Canvas.Top="8" Width="10"/>
<Rectangle Fill="#FFF4F4F5" Height="2" Canvas.Left="23" Stroke="Black"
Width="9" Canvas.Top="12" RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
```

```
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Label Content="M7" Canvas.Left="-1" Canvas.Top="-1"/>

</Canvas>
<Rectangle Fill="#FFF4F4F5" Height="10" Canvas.Left="3" Stroke="Black"
Width="34" Canvas.Top="54"/>
<Rectangle Fill="#FFF4F4F5" Height="10" Canvas.Left="147" Stroke="Black"
Width="34" Canvas.Top="54"/>
<Label Content="3/5 УП" FontSize="8" Canvas.Left="81" Canvas.Top="48"/>
<Label Content="3 СП" FontSize="8" Canvas.Left="7" Canvas.Top="48"/>
<Button x:Name="button_m7k" Content="M7K" Canvas.Left="129"
Canvas.Top="24" Width="30" />
<Label Content="5 СП" FontSize="8" Canvas.Left="148" Canvas.Top="48"/>
</Canvas>
<Canvas x:Name="Bigger" Width="220" Height="250" Margin="1,1,446,180"
Grid.ColumnSpan="2">

<Path x:Name="kn_blok" Stroke="Black" Fill="Red" StrokeThickness="1"
 Data=" M 0,0 25,0 25,45 0,45 z" Stretch="Fill" Canvas.Top="118.993"
Canvas.Left="58.531" Height="9.726" Width="18.592"/>



<Label Content="28" Foreground="Black" FontSize="6" Canvas.Left="28"
Canvas.Top="2" Height="20" Width="22"/>
<Label Content="21" Foreground="Black" FontSize="6" Canvas.Left="95"
Canvas.Top="1" Height="20" Width="22"/>
<Label Content="19" Foreground="Black" FontSize="6" Canvas.Left="120"
Canvas.Top="3" Height="20" Width="19"/>
<Label Content="214" Foreground="Black" FontSize="6" Canvas.Left="157"
Canvas.Top="4" Height="20" Width="23" RenderTransformOrigin="-
0.015,0.479"/>

<Label Content="212" Foreground="Black" FontSize="6" Canvas.Left="292"
Height="20" Width="19" Canvas.Top="5"/>
<Ellipse x:Name="lampa_28" Fill="Red" Height="8" Canvas.Left="35"
Stroke="Black" StrokeThickness="1" Canvas.Top="14" Width="8"/>
<Ellipse x:Name="lampa_21" Fill="Red" Height="8" Canvas.Left="100"
Stroke="Black" StrokeThickness="1" Canvas.Top="15" Width="8"/>
```

```
<Ellipse x:Name="lampa_19" Fill="Red" Height="8" Canvas.Left="125"
Stroke="Black" StrokeThickness="1" Canvas.Top="15" Width="8"/>
<Ellipse x:Name="lampa_214" Fill="Red" Height="8" Canvas.Left="163"
Stroke="Black" StrokeThickness="1" Canvas.Top="16" Width="8"/>
<Ellipse x:Name="lampa_212" Fill="Red" Height="8" Canvas.Left="299"
Stroke="Black" StrokeThickness="1" Canvas.Top="17" Width="8"/>
<Ellipse x:Name="lampa_116" Fill="Red" Height="8" Canvas.Left="338"
Stroke="Black" StrokeThickness="1" Canvas.Top="16" Width="8"/>
<Ellipse Fill="Red" Height="8" Canvas.Left="25" Stroke="Black"
StrokeThickness="1" Canvas.Top="145" Width="8"/>
<Ellipse Fill="Red" Height="8" Canvas.Left="25" Stroke="Black"
StrokeThickness="1" Canvas.Top="132" Width="8"/>
<Ellipse Fill="Red" Height="8" Canvas.Left="25" Stroke="Black"
StrokeThickness="1" Canvas.Top="158" Width="8"/>
<Ellipse Fill="Red" Height="8" Canvas.Left="25" Stroke="Black"
StrokeThickness="1" Canvas.Top="171" Width="8"/>
<Ellipse x:Name="lampa_11" Fill="Red" Height="8" Canvas.Left="24"
Stroke="Black" StrokeThickness="1" Canvas.Top="119" Width="8"/>
<Ellipse Fill="Red" Height="8" Canvas.Left="25" Stroke="Black"
StrokeThickness="1" Canvas.Top="184" Width="8"/>
<Label Content="11" Foreground="Black" FontSize="6" Canvas.Left="9"
Canvas.Top="113" Height="20" Width="19"/>
<Ellipse Fill="Red" Height="8" Canvas.Left="25" Stroke="Black"
StrokeThickness="1" Canvas.Top="197" Width="8"/>
<Path x:Name="vp_blok" Stroke="Black" Fill="Red" StrokeThickness="1"
      Data=" M 0,0 25,0 25,45 0,45 z" Stretch="Fill" Canvas.Top="120.286"
Canvas.Left="360.571" Height="10.062"/>
<Path x:Name="vkm_blok" Stroke="Black" Fill="Red" StrokeThickness="1"
      Data=" M 0,0 25,0 25,45 0,45 z" Stretch="Fill" Canvas.Top="122.132"
Canvas.Left="218" Height="10.736"/>
<Path x:Name="mp_blok" Stroke="Black" Fill="Red" StrokeThickness="1"
      Data=" M 0,0 25,0 25,45 0,45 z" Stretch="Fill" Canvas.Top="121.162"
Canvas.Left="146" Height="9.838"/>
<Label Content="КН" Foreground="Black" FontSize="6" Canvas.Left="58"
Canvas.Top="114" Height="20" Width="20"/>
<Label Content="ВП" Foreground="Black" FontSize="6" Canvas.Left="364"
Canvas.Top="116" Height="20" Width="20"/>
<Label Content="ВКМ" Foreground="Black" FontSize="6" Canvas.Left="221"
Canvas.Top="118" Height="18" Width="22"/>
<Label Content="МП" Foreground="Black" FontSize="6" Canvas.Left="78"
Canvas.Top="63" Height="20" Width="20" RenderTransformOrigin="-
0.073,0.592"/>
<Ellipse x:Name="lampa_p" Fill="Red" Height="8" Canvas.Left="53"
Stroke="Black" StrokeThickness="1" Canvas.Top="228" Width="8"/>
```

```
<Ellipse x:Name="lampa_m" Fill="Red" Height="8" Canvas.Left="66"
Stroke="Black" StrokeThickness="1" Canvas.Top="228" Width="8"/>
<Ellipse x:Name="lampa_cx" Fill="Red" Height="8" Canvas.Left="79"
Stroke="Black" StrokeThickness="1" Canvas.Top="228" Width="8"/>
<Ellipse x:Name="lampa_mg" Fill="Red" Height="8" Canvas.Left="92"
Stroke="Black" StrokeThickness="1" Canvas.Top="228" Width="8"/>
<Ellipse x:Name="lampa_pn" Fill="Red" Height="8" Canvas.Left="105"
Stroke="Black" StrokeThickness="1" Canvas.Top="228" Width="8"/>
<Ellipse x:Name="lampa_pg" Fill="Red" Height="8" Canvas.Left="118"
Stroke="Black" StrokeThickness="1" Canvas.Top="228" Width="8"/>
<Ellipse x:Name="lampa_pb" Fill="Red" Height="8" Canvas.Left="131"
Stroke="Black" StrokeThickness="1" Canvas.Top="228" Width="8"/>
<Ellipse x:Name="lampa_216" Fill="Red" Height="8" Canvas.Left="259"
Stroke="Black" StrokeThickness="1" Canvas.Top="16" Width="8"/>

<Label Content="П" Foreground="Black" FontSize="6" Canvas.Left="51"
Canvas.Top="189" Height="19" Width="14"/>
<Label Content="М" Foreground="Black" FontSize="6" Canvas.Left="62"
Canvas.Top="189" Height="19" Width="18"/>
<Label Content="СХ" Foreground="Black" FontSize="6" Canvas.Left="75"
Canvas.Top="189" Height="19" Width="18"/>
<Label Content="МГ" Foreground="Black" FontSize="6" Canvas.Left="88"
Canvas.Top="189" Height="19" Width="19"/>
<Label Content="МПН" Foreground="Black" FontSize="6" Canvas.Left="103"
Canvas.Top="189" Height="19" Width="16"/>
<Label Content="ПГ" Foreground="Black" FontSize="6" Canvas.Left="114"
Canvas.Top="189" Height="19" Width="18"/>
<Label Content="ПБ" Foreground="Black" FontSize="6" Canvas.Left="127"
Canvas.Top="189" Height="19" Width="19"/>

<Label Content="216" Foreground="Black" FontSize="6" Canvas.Left="253"
Height="20" Width="22" Canvas.Top="4"/>
<Label Content="116" Foreground="Black" FontSize="6" Canvas.Left="331"
Height="20" Width="22" Canvas.Top="3"/>
<Ellipse x:Name="lampa_pk" Fill="Red" Height="8" Canvas.Left="143"
Stroke="Black" StrokeThickness="1" Canvas.Top="228" Width="8"/>
<Label Content="ПК" Foreground="Black" FontSize="6" Canvas.Left="139"
Canvas.Top="189" Height="19" Width="19"/>
<Label Content="МП" Foreground="Black" FontSize="6" Canvas.Left="134"
Canvas.Top="60" Height="20" Width="20" RenderTransformOrigin="-
0.073,0.592"/>
<Label Content="КН" Foreground="Black" FontSize="6" Canvas.Left="175"
Canvas.Top="40" Height="20" Width="20" RenderTransformOrigin="-
0.073,0.592"/>
```

```
<Label Content="КН" Foreground="Black" FontSize="6" Canvas.Left="105"
Canvas.Top="61" Height="20" Width="20" RenderTransformOrigin="-
0.073,0.592"/>
<Label Content="КН" Foreground="Black" FontSize="6" Canvas.Left="239"
Canvas.Top="35" Height="20" Width="20" RenderTransformOrigin="-
0.073,0.592"/>
<Label Content="ВКМ" Foreground="Black" FontSize="6" Canvas.Left="273"
Canvas.Top="58" Height="20" Width="24" RenderTransformOrigin="-
0.073,0.592"/>
<Label Content="ВП" Foreground="Black" FontSize="6" Canvas.Left="312"
Canvas.Top="57" Height="20" Width="20" RenderTransformOrigin="-
0.073,0.592"/>
<Label Content="КН" Foreground="Black" FontSize="6" Canvas.Left="348"
Canvas.Top="37" Height="20" Width="20" RenderTransformOrigin="-
0.073,0.592"/>
<Label Content="МГ" Foreground="Black" FontSize="6" Canvas.Left="149"
Canvas.Top="155" Height="20" Width="20" RenderTransformOrigin="-
0.073,0.592"/>
<Label Content="МГ" Foreground="Black" FontSize="6" Canvas.Left="223"
Canvas.Top="158" Height="20" Width="20" RenderTransformOrigin="-
0.073,0.592"/>
<Label Content="МГ" Foreground="Black" FontSize="6" Canvas.Left="364"
Canvas.Top="155" Height="20" Width="20" RenderTransformOrigin="-
0.073,0.592"/>
<Label Content="112" Foreground="Black" FontSize="6" Canvas.Left="47"
Canvas.Top="232" Height="19" Width="26"/>
<Label Content="110" Foreground="Black" FontSize="6" Canvas.Left="59"
Canvas.Top="232" Height="19" Width="26"/>
<Label Content="220" Foreground="Black" FontSize="6" Canvas.Left="72"
Canvas.Top="232" Height="19" Width="26"/>
<Label Content="114" Foreground="Black" FontSize="6" Canvas.Left="86"
Canvas.Top="232" Height="19" Width="26"/>
<Label Content="210" Foreground="Black" FontSize="6" Canvas.Left="99"
Canvas.Top="232" Height="19" Width="26"/>
<Label Content="118" Foreground="Black" FontSize="6" Canvas.Left="112"
Canvas.Top="232" Height="19" Width="26"/>
<Label Content="215" Foreground="Black" FontSize="6" Canvas.Left="126"
Canvas.Top="232" Height="19" Width="26"/>
<Label Content="17" Foreground="Black" FontSize="6" Canvas.Left="138"
Canvas.Top="232" Height="19" Width="26"/>
<Path x:Name="k_blok" Stroke="Black" Fill="Red" StrokeThickness="1"
      Data=" M 0,0 25,0 25,45 0,45 z" Stretch="Fill" Canvas.Top="187.344"
Canvas.Left="313.185" Height="10.062"/>
<Label Content="K" Foreground="Black" FontSize="6" Canvas.Left="316"
Canvas.Top="183" Height="20" Width="16"/>
```

```
<Label Content="К" Foreground="Black" FontSize="6" Canvas.Left="268"
Canvas.Top="175" Height="20" Width="16"/>
<Label Content="П" Foreground="Black" FontSize="6" Canvas.Left="222"
Canvas.Top="183" Height="20" Width="16"/>
<Label Content="ПК" Foreground="Black" FontSize="6" Canvas.Left="218"
Canvas.Top="203" Height="20" Width="20"/>
<Ellipse x:Name="lampa_23" Fill="Red" Height="8" Canvas.Left="392"
Stroke="Black" StrokeThickness="1" Canvas.Top="188" Width="8"/>
<Label Content="23" Foreground="Black" FontSize="6" Canvas.Left="389"
Canvas.Top="174" Height="19" Width="26"/>
<Label Content="М" Foreground="Black" FontSize="6" Canvas.Left="459"
Canvas.Top="163" Height="20" Width="20" RenderTransformOrigin="-
0.073,0.592"/>
<Label Content="М7К" Foreground="Black" FontSize="6" Canvas.Left="420"
Canvas.Top="156" Height="20" Width="26" RenderTransformOrigin="-
0.073,0.592"/>
<Label Content="К" Foreground="Black" FontSize="6" Canvas.Left="83"
Canvas.Top="92" Height="20" Width="20" RenderTransformOrigin="-
0.073,0.592"/>
<Label Content="ПН" Foreground="Black" FontSize="6" Canvas.Left="141"
Canvas.Top="96" Height="20" Width="20" RenderTransformOrigin="-
0.073,0.592"/>
<Label Content="КН" Foreground="Black" FontSize="6" Canvas.Left="112"
Canvas.Top="90" Height="20" Width="20" RenderTransformOrigin="-
0.073,0.592"/>

</Canvas>

<Canvas Margin="208,-9,123,197" Cursor="" Grid.ColumnSpan="2">
<Canvas x:Name="k_vozb" Width="37" Height="9" Canvas.Top="2"
Canvas.Left="-16">
<!-- vozbujdenie K================  -->
<Label Content="" FontSize="9" Canvas.Left="125" Canvas.Top="35"
RenderTransformOrigin="0.5,0.5">
<Label.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform AngleY="1.286"/>
<RotateTransform/>
<TranslateTransform Y="-0.202"/>
</TransformGroup>
</Label.RenderTransform>
</Label>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-97" Stroke="Black"
StrokeThickness="1" Canvas.Top="63" Width="10"/>
```

```
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-100"
Stroke="Black" StrokeThickness="1" Canvas.Top="68" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-155"
Stroke="Black" StrokeThickness="1" Canvas.Top="94" Width="64"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-117"
Stroke="Black" StrokeThickness="1" Canvas.Top="66" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-159"
Stroke="Black" StrokeThickness="1" Canvas.Top="130" Width="27"/>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-104"
Stroke="Black" StrokeThickness="1" Canvas.Top="47" Width="34"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
```

```
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-158"
Stroke="Black" StrokeThickness="1" Canvas.Top="35" Width="12"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-153"
Stroke="Black" StrokeThickness="1" Canvas.Top="40" Width="10"/>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-148"
Stroke="Black" StrokeThickness="1" Canvas.Top="44" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-129"
Stroke="Black" StrokeThickness="1" Canvas.Top="63" Width="17"/>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-132"
Stroke="Black" StrokeThickness="1" Canvas.Top="59" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
```

```
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-101"
Stroke="Black" StrokeThickness="1" Canvas.Top="101" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-90" Stroke="Black"
StrokeThickness="1" Canvas.Top="85" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-72" Stroke="Black"
StrokeThickness="1" Canvas.Top="85" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-68" Stroke="Black"
StrokeThickness="1" Canvas.Top="82" Width="13"/>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-97" Stroke="Black"
StrokeThickness="1" Canvas.Top="82" Width="12"/>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-99" Stroke="Black"
StrokeThickness="1" Canvas.Top="86" Width="6"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
```

```
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-117"
Stroke="Black" StrokeThickness="1" Canvas.Top="85" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-124"
Stroke="Black" StrokeThickness="1" Canvas.Top="82" Width="12"/>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-60" Stroke="Black"
StrokeThickness="1" Canvas.Top="85" Width="8"
RenderTransformOrigin="0.279,5.646">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform X="-8.524" Y="-12.06"/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-42" Stroke="Black"
StrokeThickness="1" Canvas.Top="85" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
```

```
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-38" Stroke="Black"
StrokeThickness="1" Canvas.Top="82" Width="23"
RenderTransformOrigin="1.414,11.398"/>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-39" Stroke="Black"
StrokeThickness="1" Canvas.Top="46" Width="30"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-19" Stroke="Black"
StrokeThickness="1" Canvas.Top="79" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-20" Stroke="Black"
StrokeThickness="1" Canvas.Top="64" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-25" Stroke="Black"
StrokeThickness="1" Canvas.Top="61" Width="10"/>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-2" Stroke="Black"
StrokeThickness="1" Canvas.Top="64" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
```

```
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="1" Stroke="Black"
StrokeThickness="1" Canvas.Top="61" Width="10"/>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-16" Stroke="Black"
StrokeThickness="1" Canvas.Top="88" Width="52"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-40" Stroke="Black"
StrokeThickness="1" Canvas.Top="122" Width="16"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-33" Stroke="Black"
StrokeThickness="1" Canvas.Top="114" Width="44"
RenderTransformOrigin="1.414,11.398"/>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="43" Stroke="Black"
StrokeThickness="1" Canvas.Top="61" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
```

```
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="61" Stroke="Black"
StrokeThickness="1" Canvas.Top="61" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="64" Stroke="Black"
StrokeThickness="1" Canvas.Top="57" Width="9"/>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="59" Stroke="Black"
StrokeThickness="1" Canvas.Top="44" Width="26"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="39" Stroke="Black"
StrokeThickness="1" Canvas.Top="57" Width="9"/>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="3" Stroke="Black"
StrokeThickness="1" Canvas.Top="93" Width="74"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
```

```
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="64" Stroke="Black"
StrokeThickness="1" Canvas.Top="81" Width="23"
RenderTransformOrigin="1.414,11.398"/>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="61" Stroke="Black"
StrokeThickness="1" Canvas.Top="77" Width="8" RenderTransformOrigin="-
0.129,5.139">
<Rectangle.RenderTransform>
<TransformGroup>
</Rectangle>
<Rectangle Fill="#FFBFBFBF" Height="2" Canvas.Left="-11" Stroke="Black"
StrokeThickness="1" Canvas.Top="-9" Width="8"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="90"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>

</Canvas>
<Canvas x:Name="VV_voz_samob" Width="180" Height="70"
Canvas.Top="256" Canvas.Left="269">
<!-- Vozbujdenie OP ================  -->
<Rectangle x:Name="vp_p1" Fill="#FFBFBFBF" Height="2" Canvas.Left="-
166" Stroke="Black" StrokeThickness="1" Canvas.Top="-166" Width="20"
Visibility="hidden"/>
<Rectangle x:Name="vp_m1" Fill="#FFBFBFBF" Height="2" Canvas.Left="-
165" Stroke="Black" StrokeThickness="1" Canvas.Top="-162" Width="20"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="-24.599"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle x:Name="kn_p5" Fill="#FFBFBFBF" Height="2" Canvas.Left="-
129" Stroke="Black" StrokeThickness="1" Canvas.Top="-188" Width="20"
Visibility="hidden" />
```

```
<Rectangle x:Name="kn_m5" Fill="#FFBFBFBF" Height="2" Canvas.Left="-
129" Stroke="Black" StrokeThickness="1" Canvas.Top="-183" Width="22"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="-19.943"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle x:Name="k_p2" Fill="#FFBFBFBF" Height="2" Canvas.Left="-
211" Stroke="Black" StrokeThickness="1" Canvas.Top="-48" Width="20"
Visibility="hidden"/>
<Rectangle x:Name="k_m2" Fill="#FFBFBFBF" Height="2" Canvas.Left="-
210" Stroke="Black" StrokeThickness="1" Canvas.Top="-44" Width="20"
RenderTransformOrigin="0.5,0.5">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="-24.599"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>
<Rectangle x:Name="m7k_p" Fill="#FFBFBFBF" Height="2" Canvas.Left="-
52" Stroke="Black" StrokeThickness="1" Canvas.Top="-69" Width="20"/>
<Rectangle x:Name="m7k_m" Fill="#FFBFBFBF" Height="2" Canvas.Left="-
51" Stroke="Black" StrokeThickness="1" Canvas.Top="-65" Width="20"
RenderTransformOrigin="0.5,0.5" Visibility="hidden">
<Rectangle.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform Angle="-24.599"/>
<TranslateTransform/>
</TransformGroup>
</Rectangle.RenderTransform>
</Rectangle>

</Canvas>
</Canvas>
```

```
<CheckBox x:Name="Check_p" Content="П" HorizontalAlignment="Left"
Margin="18,273,0,0" VerticalAlignment="Top" Grid.ColumnSpan="2"/>
<CheckBox x:Name="Check_tchm" Content="Т-ЧМ"
HorizontalAlignment="Left" Margin="18,324,0,0" VerticalAlignment="Top"
Grid.ColumnSpan="2" />
<CheckBox x:Name="Check_m" Content="М" HorizontalAlignment="Left"
Margin="69,273,0,0" VerticalAlignment="Top" Grid.ColumnSpan="2"/>
<CheckBox x:Name="Check_11" Content="11" HorizontalAlignment="Left"
Margin="18,350,0,0" VerticalAlignment="Top" Grid.ColumnSpan="2"/>
<CheckBox x:Name="Check_pk" Content="ПК" HorizontalAlignment="Left"
Margin="69,325,0,0" VerticalAlignment="Top" Grid.ColumnSpan="2"/>
<CheckBox x:Name="Check_pn" Content="ПН" HorizontalAlignment="Left"
Margin="17,294,0,0" VerticalAlignment="Top" Grid.ColumnSpan="2" />
<CheckBox x:Name="Check_214" Content="214 (ЧМ)"
HorizontalAlignment="Left" Margin="18,397,0,0" VerticalAlignment="Top"
Grid.ColumnSpan="2"/>
<CheckBox x:Name="Check_19" Content="19" HorizontalAlignment="Left"
Margin="18,365,0,0" VerticalAlignment="Top" Grid.ColumnSpan="2"/>
<CheckBox x:Name="Check_mg" Content="МГ" HorizontalAlignment="Left"
Margin="69,290,0,0" VerticalAlignment="Top" Grid.ColumnSpan="2" />
<CheckBox x:Name="Check_28" Content="28" HorizontalAlignment="Left"
Margin="18,380,0,0" VerticalAlignment="Top" Grid.ColumnSpan="2"/>
<CheckBox x:Name="Check_216" Content="216 (НМ)"
HorizontalAlignment="Left" Margin="18,412,0,0" VerticalAlignment="Top"
Grid.ColumnSpan="2"/>
<CheckBox x:Name="Check_212" Content="212" HorizontalAlignment="Left"
Margin="69,341,0,0" VerticalAlignment="Top" Grid.ColumnSpan="2" />
<CheckBox x:Name="Check_116" Content="116" HorizontalAlignment="Left"
Margin="69,359,0,0" VerticalAlignment="Top" Grid.ColumnSpan="2" />
<Rectangle Fill="#FFBFBFBF" HorizontalAlignment="Left" Height="100"
Margin="220,225,0,0" Stroke="Black" StrokeThickness="1"
VerticalAlignment="Top" Width="100"/>




</Grid>
</Viewbox>


</Window>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfApplication5
{
///<summary>
/// Interaction logic for MainWindow.xaml
///</summary>
publicpartialclassMainWindow : Window
   {
publicstaticColor rang = Color.FromRgb(0, 0, 255);
publicstaticSolidColorBrush kok = newSolidColorBrush(Color.FromRgb(0, 0,
255));
publicstaticSolidColorBrush green = newSolidColorBrush(Color.FromRgb(0,
255, 0));
publicstaticSolidColorBrush red = newSolidColorBrush(Color.FromRgb(255, 0,
0));
publicstaticSolidColorBrush sariq = newSolidColorBrush(Color.FromRgb(255,
255, 0));
publicstaticSolidColorBrush fon = newSolidColorBrush(Color.FromRgb(240,
230, 140));
publicstaticSolidColorBrush oq = newSolidColorBrush(Color.FromRgb(255,
255, 255));
publicstaticSolidColorBrush qora = newSolidColorBrush(Color.FromRgb(0, 0,
0));

        System.Windows.Threading.DispatcherTimer tik =
newSystem.Windows.Threading.DispatcherTimer();
        System.Windows.Threading.DispatcherTimer zamedleniya =
newSystem.Windows.Threading.DispatcherTimer();
        System.Windows.Threading.DispatcherTimer zamedleniya_2 =
newSystem.Windows.Threading.DispatcherTimer();
```

```
public MainWindow()
        {
InitializeComponent();

            tik.Tick += newEventHandler(sikl);
            tik.Interval = newTimeSpan(0, 0, 0, 0, 100);
tik.Start();

            zamedleniya.Tick += newEventHandler(sekinlash);
            zamedleniya.Interval = newTimeSpan(0, 0, 0, 0, 100);

            zamedleniya_2.Tick += newEventHandler(sekinlash_2);
            zamedleniya_2.Interval = newTimeSpan(0, 0, 0, 0, 10);

        }
privatevoid sekinlash_2(object sender, EventArgs e)
        { }
privatevoid Chs_Activated(object sender, EventArgs e)
        {
            tik.Tick += newEventHandler(sikl);
            tik.Interval = newTimeSpan(0, 0, 0, 0, 100);
tik.Start();


///tik.Start();
        }
publicstaticbool btn_1flag, btn_2flag;


int n;
int sek_vaqt = 100;
publicvoid sekinlash(object sender, EventArgs e)
        {
            n = n + 1;
if (n == sek_vaqt)
            {

                mp_blok.Fill = red;
                vkm_blok.Fill = red;
                vp_blok.Fill = red;
                n = 0;
zamedleniya.Stop();
            }

        }
```

```
publicvoid sikl(object sender, EventArgs e)
      {

          {
SolidColorBrush myBrushR = newSolidColorBrush(Colors.Red);
SolidColorBrush myBrushB = newSolidColorBrush(Colors.Blue);
//ns_p1.Visibility = Visibility.Visible;
if (Check_p.IsChecked == true)
              {

                 k_p2.Fill = myBrushB;
              }
else
              {
                 k_p2.Fill = myBrushR;
              }
          }



          {
if (Check_pk.IsChecked == true)
              {
                 lampa_pk.Fill = green;

              }

else
              {
                 lampa_pk.Fill = red;

              }

          }
          {
if (Check_p.IsChecked == true)
              {
                 lampa_p.Fill = green;

              }

else
```

```
            {
               lampa_p.Fill = red;

            }

         }




         {
if (Check_tchm.IsChecked == true)
            {
               lampa_21.Fill = green;

            }
else
{ lampa_21.Fill = red; }

         }


         {
if (Check_m.IsChecked == true)
            {
               lampa_m.Fill = green;

            }
else
{ lampa_m.Fill = red; }

         }







         {
if (Check_11.IsChecked == true)
            {
               lampa_11.Fill = green;

            }
```

```
else
{ lampa_11.Fill = red; }
        }

        {
if (Check_pn.IsChecked == true)
            {
               lampa_pn.Fill = green;

            }
else
{ lampa_pn.Fill = red; }
        }

        {
if (Check_19.IsChecked == true)
            {
               lampa_19.Fill = green;

            }
else
{ lampa_19.Fill = red; }
        }

        {
if (Check_214.IsChecked == true)
            {
               lampa_214.Fill = green;

            }
else
{ lampa_214.Fill = red; }
        }

        {
if (Check_mg.IsChecked == true)
            {
               lampa_mg.Fill = green;

            }
else
{ lampa_mg.Fill = red; }
        }
        {
if (Check_28.IsChecked == true)
```

```
            {
               lampa_28.Fill = green;

            }
else
{ lampa_28.Fill = red; }
         }
         {
if (Check_216.IsChecked == true)
            {
               lampa_216.Fill = green;

            }
else
{ lampa_216.Fill = red; }
         }
         {
if (Check_212.IsChecked == true)
            {
               lampa_212.Fill = green;

            }
else
{ lampa_212.Fill = red; }
         }

if (Check_116.IsChecked == true)
         {
            lampa_116.Fill = green;

         }
else
{ lampa_116.Fill = red; }




//нажатие M7K
         {
if (button_m7k.IsPressed == true)
            {
               m7k_p.Visibility = Visibility.Hidden;
               m7k_m.Visibility = Visibility.Visible;
```

```
            }
else
            {
               m7k_p.Visibility = Visibility.Visible;
               m7k_m.Visibility = Visibility.Hidden;
            }
         }

//ВключениерелеК
         {


if ((button_m7k.IsPressed == true&& Check_pk.IsChecked == true&&
Check_p.IsChecked == true&& Check_m.IsChecked == true)
             )
            {
               k_blok.Fill = green;




            }

else
            {
               k_blok.Fill = red;
               ;
            }

//контактыК
            {
if (k_blok.Fill == green)
               {
                  k_p2.Visibility = Visibility.Visible;
                  k_m2.Visibility = Visibility.Hidden;
                  k_p1.Visibility = Visibility.Visible;
                  k_m1.Visibility = Visibility.Hidden;
                  k_p3.Visibility = Visibility.Visible;
                  k_m3.Visibility = Visibility.Hidden;
                  k_p4.Visibility = Visibility.Visible;
                  k_m4.Visibility = Visibility.Hidden;



               }
```

```
else
                {
                   k_m2.Visibility = Visibility.Visible;
                   k_p2.Visibility = Visibility.Hidden;
                   k_m1.Visibility = Visibility.Visible;
                   k_p1.Visibility = Visibility.Hidden;
                   k_m3.Visibility = Visibility.Visible;
                   k_p3.Visibility = Visibility.Hidden;
                   k_m4.Visibility = Visibility.Visible;
                   k_p4.Visibility = Visibility.Hidden;


                }
            }
//ВключениерелеКНисамоблокировка

if ((k_blok.Fill == green && Check_11.IsChecked == true&&
Check_tchm.IsChecked == true)
                || (Check_11.IsChecked == true&& Check_pn.IsChecked ==
true&& kn_p1.Visibility == Visibility.Visible && mp_m2.Visibility ==
Visibility.Visible)
                || (Check_28.IsChecked == true&& Check_11.IsChecked ==
true&& k_m3.Visibility == Visibility.Visible))
            {
                kn_blok.Fill = green;

            }

else
            {
                kn_blok.Fill = red;

            }
//контактыКН
            {
if (kn_blok.Fill == green)

                {
                   kn_p1.Visibility = Visibility.Visible;
                   kn_m1.Visibility = Visibility.Hidden;

                   kn_p3.Visibility = Visibility.Visible;
                   kn_m3.Visibility = Visibility.Hidden;
                   kn_p4.Visibility = Visibility.Visible;
                   kn_m4.Visibility = Visibility.Hidden;
```

```
                kn_p5.Visibility = Visibility.Visible;
                kn_m5.Visibility = Visibility.Hidden;
                kn_p6.Visibility = Visibility.Visible;
                kn_m6.Visibility = Visibility.Hidden;
            }
else
            {
                kn_m1.Visibility = Visibility.Visible;
                kn_p1.Visibility = Visibility.Hidden;

                kn_m3.Visibility = Visibility.Visible;
                kn_p3.Visibility = Visibility.Hidden;
                kn_m4.Visibility = Visibility.Visible;
                kn_p4.Visibility = Visibility.Hidden;
                kn_m5.Visibility = Visibility.Visible;
                kn_p5.Visibility = Visibility.Hidden;
                kn_m6.Visibility = Visibility.Visible;
                kn_p6.Visibility = Visibility.Hidden;
            }
        }

//ВключениерелеМП
if ((kn_p3.Visibility == Visibility.Visible && Check_214.IsChecked ==
true&& Check_mg.IsChecked == true)
                || (Check_19.IsChecked == true&& mp_p1.Visibility ==
Visibility.Visible && kn_m3.Visibility == Visibility.Visible &&
Check_mg.IsChecked == true))
        {
            mp_blok.Fill = green;

        }
else
        {
zamedleniya.Start();
        }


// kontakty mp
        {
if (mp_blok.Fill == green)
            {
                mp_p1.Visibility = Visibility.Visible;
                mp_p2.Visibility = Visibility.Visible;
                mp_m1.Visibility = Visibility.Hidden;
                mp_m2.Visibility = Visibility.Hidden;
```

```
                }
else
                {
                    mp_p1.Visibility = Visibility.Hidden;
                    mp_p2.Visibility = Visibility.Hidden;
                    mp_m1.Visibility = Visibility.Visible;
                    mp_m2.Visibility = Visibility.Visible;
                }
            }

//vozbujdenie i samoblokirovka VKM
            {
if ((Check_mg.IsChecked == true&& Check_216.IsChecked == true&&
kn_p4.Visibility == Visibility.Visible)
                    || (Check_mg.IsChecked == true&& Check_212.IsChecked ==
true&& kn_m4.Visibility == Visibility.Visible && vkm_p1.Visibility ==
Visibility.Visible))
{ vkm_blok.Fill = green; }
else
{ zamedleniya.Start(); }
            }

            {
if (vkm_blok.Fill == green)
                {
                    vkm_p1.Visibility = Visibility.Visible;
                    vkm_m1.Visibility = Visibility.Hidden;
}
else
                {
                    vkm_p1.Visibility = Visibility.Hidden;
                    vkm_m1.Visibility = Visibility.Visible;
                }
            }
//vozbujdenie i samoblokirovka VKM

            {
if ((Check_mg.IsChecked == true&& Check_116.IsChecked == true&&
kn_p5.Visibility == Visibility.Visible)
                    || (Check_mg.IsChecked == true&& Check_212.IsChecked ==
true&& kn_m5.Visibility == Visibility.Visible && vp_p1.Visibility ==
Visibility.Visible))

{ vp_blok.Fill = green; }
```

```
else
{ zamedleniya.Start(); }
            }
if (vp_blok.Fill == green)
            {
               vp_p1.Visibility = Visibility.Visible;
               vp_m1.Visibility = Visibility.Hidden;
            }
else
            {
               vp_p1.Visibility = Visibility.Hidden;
               vp_m1.Visibility = Visibility.Visible;
}
        }      }    }  }
```