Ministry of Education and Science of Ukraine
Ternopil Ivan Puluj National Technical University

Faculty of Computer Information Systems and Software Engineering
(full name of faculty)
Computer Science Department
(full name of department)

QUALIFYING PAPER

For the degree of

Bachelors' degree
(degree name)

Topic: **Information Systems for remote control of the robot manipulator.**

Submitted by: fourth year student 4 , group ICH-42
specialty Computer Science
Computer Sciences 122
(code and name of specialty)

(signature) Agyemang Frederick Obeng (surname and initials)

Supervisor (signature) Nazarevych Oleg (surname and initials)

Standards verified by (signature) Shymchuk Gryhorii (surname and initials)

Head of Department (signature) Bodnarchuk Ihor (surname and initials)

Reviewer (signature) (surname and initials)

Ternopil
2021

Ministry of Education and Science of Ukraine
Ternopil Ivan Puluj National Technical University

Faculty Faculty of Computer Information Systems and Software Engineering
(full name of faculty)

Department Computer Science Department
(full name of department)

APPROVED BY
Head of Department
_____ Bodnarchuk I.O
(signature) (surname and initials)
« » 20___

ASSIGNMENT

for QUALIFYING PAPER

for the degree of Bachelors Degree
(degree name)

specialty 122 Computer Science
(code and name of the specialty)

student Agyemang Frederick Obeng
(surname, name, patronymic)

1. Paper topic **Information system for remote control of the robot manipulator**

Paper supervisor Nazarevych Oleh Bohdanovych, PhD, Assoc. Prof.
(surname, name, patronymic, scientific degree, academic rank)

Approved by university order as of «*01*» *01* 2021 № *4/7-62*

2. Student's paper submission deadline 26.06.2021

3. Initial data for the paper Technical task for development

4. Paper contents (list of issues to be developed)
Introduction 1. Analysis on Arduino and its operations and Selection of servo model. 2. Assembling of manipulator's parts. 3. Information technology that implements messaging protocol to the manipulator via smartphone, configuration of MQTT broker on Windows operating system and smartphone. . Conclusion, References.

5. List of graphic material (with exact number of required drawings, slides)

6. Advisors of paper chapters

Chapter	Advisor's surname, initials and position	Signature, date	
		assignment was given by	assignment was received by
Life safety			

7. Date of receiving the assignment January 25, 2021

TIME SCHEDULE

LN	Paper stages	Paper stages deadlines	Notes
1.	Familiarity with the qualification paper's assignment	25.01.2021	Completed
2.	Analysis on Arduino and its operations	26.01.2021 – 27.02.2021	Completed
3.	Selection of servo model	29.01.2021 – 01.02.2021	Completed
4.	Assembling of manipulator's parts	02.02.2021 – 08.02.2021	Completed
5.	Introduction of a messaging broker (MQTT)	09.02.2021 – 13.02.2021	Completed
6.	Configuration of MQTT broker on Windows operating system	14.02.2021 – 17.02.2021	Completed
	Execution of task assigned by sub-division: Life safety	15.06.2021	Completed
7.	Registration of qualification paper	07.06.2021	Completed
8.	Standardization control	08.06.2021	Completed
9.	Plagiarism check	12.06.2021	Completed
10.	Preliminary defense of qualifying paper	18.06.2021	Completed
11.	Defense of qualification paper	26.06.2021	Completed

Student _____ (signature) Agyemang Frederick Obeng (surname and initials)

Paper supervisor _____ (signature) Nazarevych Oleg (surname and initials)

ANNOTATION

Remote control of the robot manipulator// Qualification work of the educational level “Bachelor” // Agyemang Frederick Obeng // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software, Department of Computer Science, ICH-42 group // Ternopil, 2021.

This bachelor’s thesis, is devoted to the creation of a remote-control system for a robot manipulator, through smartphone control. The thesis consists of introduction, four sections and conclusion.

The first section is devoted to analytics of Arduino and its usefulness in this project.

The second section is about selection of various appropriate technologies for the robot manipulator.

The third section concentrates on designing of robot manipulator, programming, launch and configuring the manipulator.

Software tools for development and planning of remote-control system: Arduino, C/C++, Mqtt dashboard for transport of messages between devices.

Keywords: Arduino, servo drive, robot manipulator, MQTT protocol, Microcontroller ESP8266 Node MCU v3, mosquito broker, expansion board, Wi-Fi antenna, C/C++.

LIST OF SYMBOLS, SYMBOLS, UNITS, ABBREVIATIONS AND TERMS

A computer is an electronic computer

PC is a personal computer

PWM – pulse width modulation

PS – power supply

OS – operating system

IDE – Integrated Development Environment

SD (Secure Digital) – portable flash card

LCD – liquid crystal display

CoAP – Constrained Application Protocol

AMQP – Advanced Message Queuing Protocol, Extended Message Queue Protocol

MQTT – Message Queue Telemetry Transport (data exchange protocol)

HTTP – (HyperText Transfer Protocol) is a hypertext transfer protocol

UART – universal asynchronous receiver transmitter

API (A Programming Language) is an array-oriented programming language

XML – Extensible Markup Language

s – second

In – volts

kg · cm – kilogram per centimeter

Fig. – drawing

ma. – milliamperes

mm – millimeters

g – gram

CONTENTS

INTRODUCTION

No sector of instrument making today can do without automated systems. Automation is an important means of simplifying production, which can be achieved through the involvement in the production and final control of robotic systems. Nowadays, manipulator robots are widely used in production, which are divided into many varieties, classes and models, but not all of these robots are able to perform certain operations on their own and still require the intervention of a technical specialist.

Solving the problem of manipulator control as part of various systems and mechanisms is an urgent task, because manipulators are widely used in production. The movement of the manipulator is carried out by means of the servo drive. The vast majority of servos have a separate control unit. Namely, they are controlled by pulse-width modulation, so they are well suited for use in laboratory work both individually and as part of a robot manipulator.

Therefore, the need to create a training model of the robot manipulator and control it using remote control based on a modern programmable controller is quite an urgent task today.

1. ANALYTICAL PART

1.1 Arduino hardware computing platform

Arduino is a hardware computing platform for amateur design, the main components of which are a microcontroller board with I/O elements and a Processing/Wiring development language in a programming language that is a simplified subset of C/C ++. Arduino can be used to create stand-alone interactive objects, and connect to software running on a computer (for example: Processing, Adobe Flash, Max / MSP, Pure Data, SuperCollider) [1].

The appearance of the first microcontrollers, miniature semiconductor devices marked the beginning of a new era in the development of microprocessor technology. The fact that in a small case most system devices have made the microcontroller similar to a normal computer. In the literature they began to be called single-chip microcomputers. Practically with their appearance there was a desire to use microcontrollers as personal computers. But various factors restrained this desire. One of such factors is that the device on the microcontroller is quite difficult to assemble, the necessary knowledge of the basics of circuitry, to be able to program in assembler. Additional accessories and programmers are also required. After all, without a certain amount of knowledge and quite expensive equipment cannot do. This situation did not allow the use of microcontrollers in their needs. With the advent of devices that allow you to work with different microcontrollers without hardware and expensive equipment, everything has changed dramatically. An example is the Arduino project.

With the help of Arduino, you can implement almost any idea. It can be an automatic greenhouse watering control system, a web server, or even an autopilot for a Multicopter. Thus, the Arduino is a hardware computing platform for developing devices based on a microcontroller, in a simple and clear programming language in the Arduino integrated environment. This is a cheap

board that is freely available, it is compatible with all popular operating systems, as well as Windows, Mac and Linux, to which you can connect many different additional electronic components. By adding sensors, drives, speakers, additional modules and additional chips, we can use the Arduino as a "brain" for any control system. The sensors can receive information about the environment, as well as control various actuators connected to the Arduino board. It is difficult to even list everything that this platform is capable of, because the possibilities are limited only by our imagination.

The Arduino platform includes an electronic unit (developer fee) and software, as well as other electronic components that can be added to the board.

The Arduino board is analogous to the motherboard of a modern PC. It has a microcontroller and the minimum number of elements required for its operation. The microcontroller is programmed using a connector for communication with a computer. There are also connectors for connecting external devices. All you need to create a new electronic device is an Arduino board (of any type), a communication cable and a PC.

The second part of the Arduino project is software called IDE. It combines a development environment and a programming language, which is a simplified version of the C / C ++ language supplemented by libraries for microcontrollers. The Arduino IDE has been added elements that allow you to create programs without studying the hardware, so to work with Arduino enough knowledge of the basics of programming in the language in C / C ++.

1.2.1 What is the advantage of Arduino?

The user's work with Arduino is very similar to working with a PC. Arduino allows the user to focus on project development rather than studying the device and the principles of operation of individual elements.

The presence of a large number of ready-made modules and sample libraries allows non-professionals to create ready-made working devices to solve problems. And the options for using the Arduino are limited only by the capabilities of the microcontroller and the available version of the board, and, of course, the imagination of the developer. As a result, access to the development of microprocessor devices has been given not only to electrical professionals, but also to ordinary amateurs.

There are many other microcontrollers and microprocessor devices designed to program various hardware: Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard and many more. All these devices offer similar functionality and are designed to free the user from the need to delve into the small details of the internal device of microcontrollers, giving him a simple and user-friendly interface for their programming. Arduino also simplifies the process of working with microcontrollers, but unlike other systems provides a number of benefits for teachers, students and radio amateurs:

Low cost: Compared to similar hardware platforms, Arduino boards have a relatively low cost: ready-made Arduino modules cost no more than $ 50, and the ability to assemble the board manually allows you to save as much money and get an Arduino at the lowest price.

Cross-platform: Arduino software runs on Windows, Macintosh OSX, and Linux, while most of these systems are Windows-only.

Simple and user-friendly programming environment. The Arduino programming environment is clear and easy for beginners, but flexible enough for advanced users. It is based on the Processing programming environment, which can be convenient for teachers. Thanks to this, students who study programming in the Processing environment will be able to easily master Arduino.

Expandable open-source software: Arduino software is open source, so experienced programmers can modify and supplement it. Arduino capabilities

can also be extended with C ++ libraries. Because it is based on the AVR C language, advanced users who want to understand the technical details can easily switch from Arduino to C or insert sections of AVR-C code directly into Arduino programs.

Expandable open hardware: Arduino devices are based on Atmel ATmega8 and ATmega168 microcontrollers. Because all Arduino schematics are published under a Creative Commons license, experienced engineers and developers can create their own versions of devices based on existing ones. And even regular users can collect Arduino prototypes to better understand how they work and save money.

1.1.3 Hardware part

The Arduino board consists of an Atmel AVR microcontroller, as well as binding elements for programming and integration with other devices. Many boards have a linear voltage regulator + 5V or + 3.3V. Clocking is performed at a frequency of 16 or 8 MHz quartz resonator. The bootloader is written to the microcontroller, so an external programmer is not required.

At the conceptual level, all boards are programmed via RS-232 (serial connection), but the implementation of this method differs from version to version. Newer boards are programmed via USB, which is possible thanks to the USB-to-Serial FTDI FT232R converter chip. The Arduino Uno version uses the Atmega8 controller in an SMD case as a converter. This solution allows you to program the converter so that the platform is immediately recognized as a mouse, joystick or other device of the developer's choice with all the necessary additional control signals. In some embodiments, such as the Arduino Mini or the unofficial Boarduino, a separate USB-to-Serial board or cable must be connected to the controller for programming. [1]

1.2 Arduino Shields – expansion boards for Arduino

The key advantage of the Arduino platform is its popularity and ease of use. Manufacturers of electronic devices support the Arduino platform and release special versions of boards that extend the basic functionality of Arduino. Such boards are called expansion cards. Their other name is Arduino shield or shield, they perform various tasks and simplify the use of the board. Use Arduino shields with various Arduino devices: LCD screens (Shield LCD), motors (Shield of motor drivers), sensors (sensor shield), to obtain coordinates and time from GPS satellites (GPS module), SD-cards (data logger) and others.

The Arduino expansion card is a device designed to use certain functions and is connected using standard connectors on the body of the board. All necessary electronic components are placed on the Arduino shield, and interaction with the main board takes place through standard Arduino foams. Typically, power to the expansion card is supplied from an Arduino board, but power from other sources is possible. Other components are connected to the expansion board through the remaining free foam. You can connect several expansion cards at the same time (Fig. 1.1).

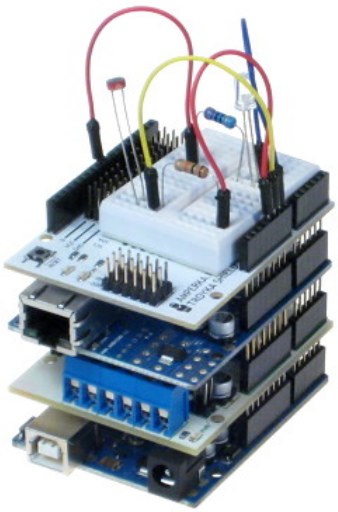

Figure 1.1 – Example of connecting expansion cards to Arduino

1.2.1 Why do I need expansion cards?

Expansion cards (Fig. 1.2) are designed to 1) expand Arduino functions, and 2) save time. Why spend your time designing, placing, soldering and adjusting what can be taken in an assembled, compact version, and immediately began to use in their projects? Especially since such boards are quite cheap, well thought out and assembled on high-quality equipment.

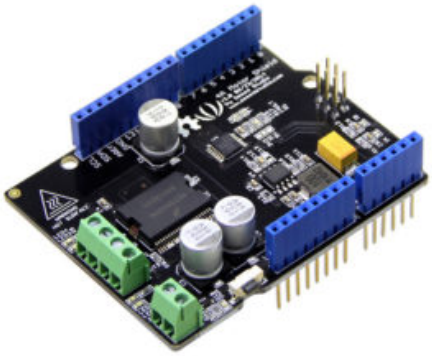

Figure 1.2 – Appearance of the expansion board

Despite the cheapness of such expansion boards, their final price will always be higher than the price of individual components, and you can make a similar option for such a sign cheaper. But the time spent on creating such a fee alone is not worth the money spent. Moreover, the cost of boards is constantly declining, so often the choice is made in favor of the use of ready-made devices.

1.2.2 Connecting and programming Arduino Shields

To connect the expansion board, you just need to "put" it on top of the Arduino board. The contacts of the expansion board have the type of comb (dad), they are easily installed in the connectors of the main board.

Expansion boards are designed for a specific version of the Arduino board, but larger shields for the Arduino Uno are suitable and work fine with Arduino Mega boards. The contact pins on the Arduino Mega board are made in such a way that the first 14 digital pins and foams on the opposite side of the board coincide with the location of the contacts on the Arduino Uno, so it easily becomes an expansion board from the Arduino Uno.

Programming circuits with installed signs is no different from the usual programming of an Arduino board, because from the controller's point of view, we simply connected the devices to its pins. In the program sketch, we initialize those pins that are not connected to the expansion board with the corresponding contacts on the main board. Typically, manufacturers of expansion cards indicate the conformity of the contacts on the Shield or in a separate instruction.

Reading or writing signals on the expansion board is also performed in the usual way: using the standard functions analogRead (), digitalRead (), digitalWrite() and others.

Arduino boards allow you to use a significant number of microcontroller pins as input / output contacts in external circuits. For example, the Decimila board has 14 digital inputs / outputs, 6 of which can generate a PWM signal, and 6 analog inputs.

1.2.3 Varieties of expansion boards

Arduino Sensor Shield is the most popular expansion card, because it usually comes in Arduino kits. The shield has a fairly simple task – to provide more convenient options for connecting to the main board. This is done through additional power and ground connectors, output to the board to each of the analog and digital pins. You can also find connectors on the board to connect an external power supply (you need to install jumpers to switch), there is also an LED and a restart button. Variants of the Arduino Sensor Shield are shown in (Fig. 1.3).

Figure 1.3 – Appearance of the Arduino Sensor Shield

There are many versions of the touch expansion card. They differ in the number and type of connectors. The most popular versions today are Sensor Shield v4 and v5.

Arduino Motor Shield (Fig. 1.4) – this type of expansion board is very important in robotic projects, because it allows you to connect to the board Arduino servo motors. The expansion board has all the necessary regulators, switches and fuses. In general, the Motor Shield has everything to ensure easy control of engines and to protect them. The main task of the Arduino Motor Shield is to provide control of devices that consume a high enough current for a conventional Arduino board. Also, such a shield can control engine power (using PWM) and change the direction of rotation. As a rule, the Arduino Motor Shield circuit has a powerful transistor through which the external load is connected, foam for connection to the Arduino, heat dissipation element, circuits for connecting external power supply, foam for connecting motors.

Figure 1.4 – Appearance of the Arduino Motor Shield

Arduino Ethernet Shield (Fig. 1.5) – is designed to ensure the independence of the Arduino project from a personal computer, as it allows you to connect the Arduino to a local network via Ethernet. Another advantage of such a sign is the presence of a slot for a MicroSD card. So, if there is a need to process a large amount of information, you can store data on the SD card. Using Ethernet Shield You can provide hosting for a web server.

Figure 1.5 – Appearance of the Arduino Ethernet Shield

Arduino LCD shield – This type of expansion card is used to work with LCD screens in Arduino. It outputs information from the Arduino not to a personal computer, but directly to the peripheral screen on the expansion board. Also, connecting an Arduino LCD shield, unlike connecting even the simplest 2-

satirical text screen, saves a few contacts, and how to connect a peripheral text ecran usually requires 7 or more contacts, and the Arduino LCD shield uses the I2C data protocol, ie to connect it, only 2 contacts are involved!

In the Arduino LCD shield (Fig. 1.6) in addition to the screen, the board has 4 "control" buttons and a "select" button. allowing you to organize both an external user interface and a direct connection to a PC can be avoided.

Figure 1.6 – Appearance of the Arduino LCD shield

Arduino Data Logger Shield (Fig. 1.7) – is designed to store data received from sensors, with time binding. This expansion card allows not only to store data and receive time from the built-in clock, but also to connect sensors by soldering or on the circuit board.

Figure 1.7 – Appearance of the Arduino Data Logger Shield

1.3 Mosquitto message broker

Eclipse Mosquitto is an open-source message broker (EPL / EDL licensed) that implements MQTT versions 3.1 and 3.1.1. Mosquitto is a fairly lightweight system, and as a result can be used on many types of devices: from simple single-board computers to full-fledged servers.

The MQTT protocol provides an easy method of messaging using a publish / subscribe model. This makes it suitable for messaging the Internet of Things, such as low-power sensors or mobile devices such as telephones, embedded computers or microcontrollers.

Mosquitto also provides a library in the C programming language, which allows you to implement MQTT clients, as well as quite popular console clients: mosquitto_pub and mosquitto_sub.

1.4 CoAP, AMQP, MQTT network exchange protocols

The exchange and transmission of data in the network is carried out using the accepted set of rules and agreements. Protocols determine the type and format of data transmitted, the general rules of interaction of different programs, the order of processing information packets, creates a single transmission space.

The interaction of the two terminals in the network is divided into several levels. These include transport, network, physical, application, etc.

Network communication protocol is a list of types and formats of transmitted data blocks, the rules of their processing and interaction of terminal computers at one level. An interface is a set of rules that determine the interaction of services at neighboring levels in one terminal.

Different protocols, taken together, form a stack of protocols and describe different aspects of the same type of communication. Protocols implement software indicated by the names "protocol" and "protocol stack".

CoAP – Constrained Application Protocol is a specialized protocol for Internet applications for limited devices. This allows those limited devices called "nodes" to communicate with the wider Internet using similar protocols. CoAP is intended for use between devices on the same restricted network (for example, low-power, loss-making networks), between devices and shared nodes on the Internet, and between devices on different restricted networks connected to the Internet. CoAP is also used through other mechanisms, such as SMS in mobile networks.

The CoAP protocol provides URIs for finding types of online media to describe them, as well as not preserving the state of setting correspondence to HTTP verbs. The protocol is compatible with existing IP infrastructure with UDP binding.

The CoAP protocol is designed for interoperability of simple devices, such as valves, switches, sensors that can control or monitor their readings remotely over the Internet. Information exchange between such devices is called machine interaction (M2M) or devices with limited resources. They got this name due to limited energy, low memory and low power, so when working with these devices it is important to ensure low power consumption and use the transmission of small messages. CoAP fulfills all the necessary requirements to ensure the interoperability of these devices. The network in which such devices operate is called a network with limited resources.

The structure of the CoAP protocol is designed according to REST. REST (Representational State Transfer) is a style of software architecture for systems such as the World Wide Web that is used to build web services. REST is a very simple information management interface without the use of additional internal layers.

The CoAP protocol has four types of messages: Confirmable, Non-confirmable, Acknowledgment, Reset. They are encoded in binary form.

AMQP protocol:

- Advanced Message Queuing Protocol
- Advanced Message Queuing Protocol (AMQP) is an open protocol for transmitting messages between system components with low latency and at a fairly high speed. AMQP can be customized to the needs of a specific project and provides a reliable transport protocol, such as TCP.

The idea of this protocol is to exchange messages arbitrarily through AMQP-broker, which performs routing, guarantees delivery, distribution of data flows, subscription to the desired types of messages.

The AMQP protocol provides communication by publishing / subscribing a message. The main advantage of AMQP is the function of storage and data transmission, which provides reliability even if the network fails.

The disadvantage of the AMQP protocol is the low level of delivery success at low bandwidth and increases with increasing bandwidth. And the advantage of comparing AMQP to REST is that AMQP can send more messages per second.

In the AMQP protocol, at the lowest level, the format of data encoding in binary form is determined, for transmission over a TCP connection, above the format of transmission of RPC requests between the server and the client. The semantics of the message protocol are described in the XML specification, which specifies the RPC interface of the server and client. XML is the latest protocol specification.

AMQP is based on three concepts:

1. Message – a unit of transmitted data, this part is not explained by the server.

2. Exchange point – messages are sent to it and messages are distributed in one or more queues. The messages are not saved at the exchange point.

3. Queue (queue) – works as a buffer, messages are stored here until it is picked up by the client from one or more queues.

MQTT protocol: Message Queue Telemetry Transport

MQTT or Message Queue Telemetry Transport is a lightweight, compact and open data exchange protocol designed for data transmission in remote locations where a small code size is required and there are bandwidth limitations.

There is also a version of the MQTT-SN protocol (MQTT for Sensor Networks), formerly known as MQTT-S, which is designed for embedded wireless devices without support for TCP / IP networks, such as Zigbee.

The main features of the MQTT message protocol are:

- asynchronous protocol.
- compact messages.
- Work in conditions of unstable communication on the data line.
- Support for multiple levels of service quality (QoS).
- Easy integration of new devices.

The MQTT protocol works at the application level on top of TCP / IP and uses the default 1883 port.

MQTT messaging takes place between a client, which can be a publisher or subscriber of messages, and a message broker (for example, Mosquitto MQTT).

The publisher sends the data to the MQTT broker, indicating in the message a certain topic, topic (topic). Subscribers can receive different data from many publishers depending on the subscription to the relevant topics.

MQTT devices use certain types of messages to interact with the broker, the following are the main ones:

- Connect – establish a connection with a broker.
- Disconnect – break the connection with the broker.
- Publish – publish data to the broker.

- Subscribe – subscribe to a broker's topic.
- Unsubscribe – unsubscribe from the topic.

The scheme of interaction between the subscriber, the publisher and the broker can be seen in Figure 1.8. [7]

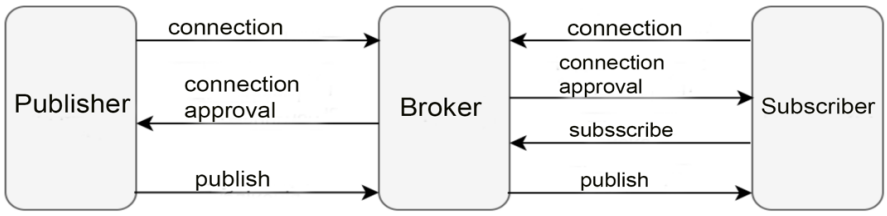

Figure 1.8 – Scheme of interaction between the subscriber, publisher and broker

The publisher sends the data to the MQTT broker, indicating in the message a certain topic, topic (topic). Subscribers can receive different data from many publishers depending on the subscription to the relevant topics.

2. TECHNOLOGICAL PART

2.1 Selection of the servo model

2.1.1 The concept of servo and its structure

Servo – a type of drive, the shaft of which can be rotated to a certain angle and hold this position or maintain rotation for a specified period. Drive control is based on the use of a negative connection where the signal at the input and output in a closed circuit does not match one or in some cases several system signals. This principle allows you to control the parameters of the movement. The servo has a sensor and a drive control unit that supports all the necessary parameters on the sensor in accordance with the specified value of the control system.

The most common servomotors are those that maintain a constant speed of rotation of the shaft and maintain a given angle of rotation.

The structure of the servo is shown in the figure below (Fig. 2.1).

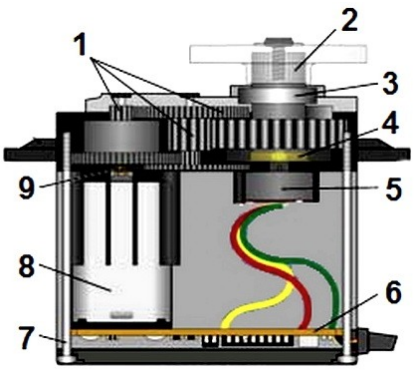

Figure 2.1 – The structure of the servo

1. Reduction gears.
2. Val.
3. Ball bearing.
4. Bushing.

5. Potentiometer.
6. Management board.
7. Housing screw.
8. Servomotor.
9. Motor gear.

An electric motor is a device that is needed to convert electrical energy into mechanical rotation of the output shaft. Usually the speed of the electric motor is too high for its practical use. Therefore, to reduce the speed using a gearbox.

The pontetiometer is used to change the current state of the mechanism. The change in resistance occurs when changing the angle of rotation of the slider potentiometer, which is proportional to the angle of rotation of the output shaft.

2.1.2 Internal interface of control signals. Servo control

To change the position of the servo, it is necessary to send a control signal to the drive. That is, pulses of constant frequency but variable width. An important parameter is the pulse length, as it determines the position of the servo. The pulse length is set manually in the program by selecting the angle of rotation or using the library. When the control circuit receives a signal, the pulse generator generates its own pulse and the potentiometer determines its duration. Then the pulse duration is compared and if it is different, the motor is turned on. The direction of rotation of the motor depends on which of the two pulses is longer, and if they are equal, the motor stops.

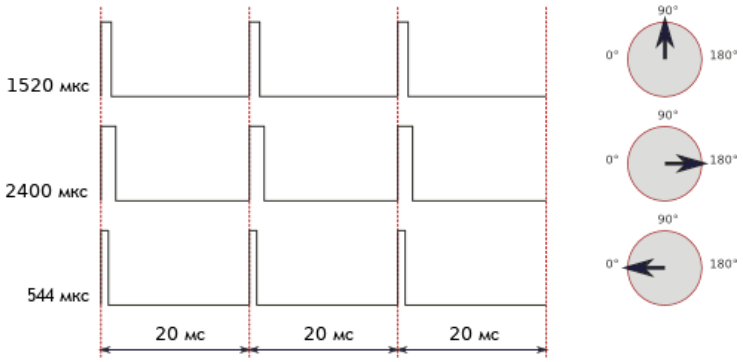

Figure 2.2 – Diagram of the dependence of the position of the servo on the pulse length

At a pulse frequency of 50 Hz, the pulse will be emitted once every 20 milliseconds. At these values, the pulse duration is 1520 μs and the servo occupies a position of 90°. Changing the pulse length changes the angle of rotation. The Arduino library has limits on the value of the pulse length at 0° – 544 microseconds (lower limit), at 180 ° – about 2400 microseconds (upper limit).

It is worth noting that sometimes the factory settings differ from the accepted standard settings. On some servos, the average position is a pulse duration of 760 microseconds.

Sometimes the method of controlling servos is called PWM. But this is not entirely correct, because the pulse length is important for its control and the frequency is not so important. A frequency of 50 Hz is considered normal, but the servo will work even with a small deviation. At high low frequencies, the servo will run jerkily, and at high (e.g., 150 Hz) the servo may overheat and, in some cases, fail.

By type of servos are divided into analog and digital. Externally there are no differences, they differ in the built-in control electronics. The analog servo has a chip, and the digital has a microprocessor that controls the motor. That is, the difference between them is only in the method of processing the input pulses

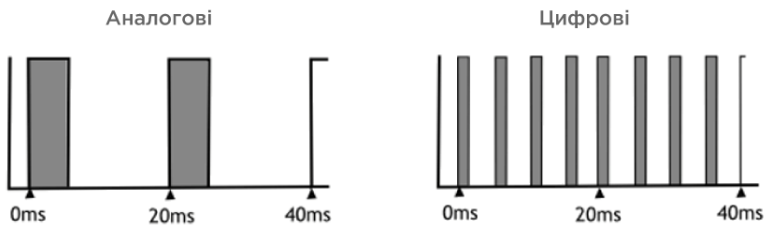

Figure 2.3 – Pulse width in analog and digital servos

Both types of servos receive the same input control pulses. The digital servo drive thanks to use of the microprocessor which sends a signal to the electric motor with a frequency of 200 Hz, reacts faster to external influences, develops necessary speed and torque, and much better holds the set position. However, it consumes less electricity and is easier to produce, so it is cheaper.

Gears and their material

Gears for servo drives are made of: metal, plastic, carbon. Their choice depends on the characteristics required in the installation, and on the specific task. Options for gears made of metal and plastic are shown in (Fig. 2.4).

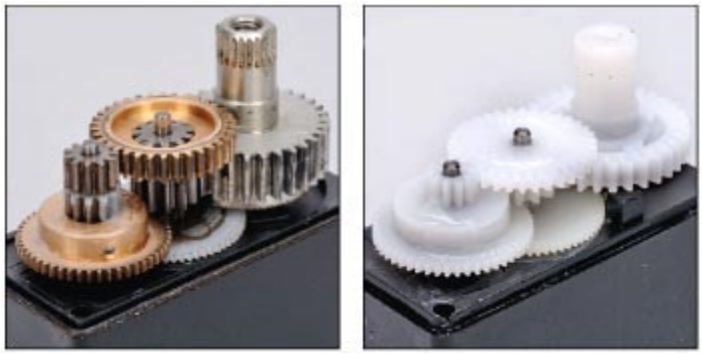

Figure 2.4 – Servo gears

Plastic (nylon) gears are a great choice if you do not have heavy loads on the servo, because they are prone to rapid wear. However, they are very light and most common to use because of their low price.

For more durable use, use servos with carbon gears, their disadvantage is the high price for them. Metal gears can withstand heavy loads, although they also wear out quickly.

Connecting servos to the Arduino

On any servo there are three wires which are painted in special colors.

- Brown or black wire – connects to ground.
- yellow or white wire is a signal connected to the digital foam of the Arduino board.
- The red wire is the power supply; it is connected to the +5 V contact or to the power supply.

You can use a special expansion card to connect the servo to the Arduino board or by connecting the servo wires directly to the Arduino pins. We see the standard connection scheme on (fig. 2.5).

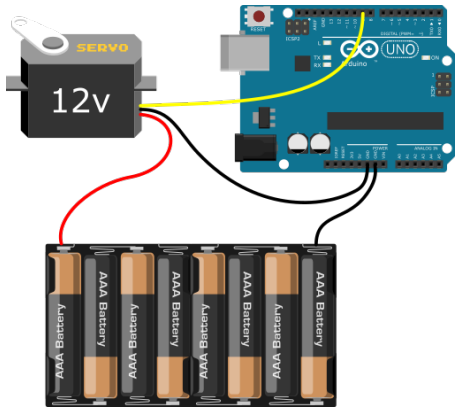

Figure 2.5 – Servo connection diagram

Characteristics of servo drives

The main characteristics of servos are: size, weight, speed, angle of rotation, torque on the shaft.

There are 3 classes of sizes: "small", "standard", "large". The size and weight of each class may vary slightly, you can see in table1.

Table 2.1 – Classification of servos by size and weight

Class	Weight	Linear dimensions
Small	8-25 g	22 × 15 × 25 mm
Standard	40-80 g	40 × 20 × 37 mm
Large	50-90 g	49 × 25 × 40 mm

The speed of the servo is the time during which the drive rotates at an angle of 60° at a voltage of 4.8V and 6V. For example, a servo with the specified parameter 0.26s / 60 ° at 6V rotates the shaft by 60 degrees for 0.26s at a supply voltage of 6V. This is a somewhat mediocre result. Some servos perform this movement for a time from 0.06 to 0.09s.

It is believed that the angle of rotation of the servo is 180 degrees, but this is not the case. Servo drives, depending on the model, can rotate 60, 90, 180, 270, 360 degrees. The angle of rotation of such servos is limited by internal electronics and mechanics. To change the operating range of the servo, you need to change its internal design.

Torque is an important characteristic for the development of various projects, in particular the hands of the manipulator.

The moment of force, or torque, is a vector physical quantity equal to the product of the radius vector drawn from the axis of rotation to the point of application of the force on the vector of this force. Characterizes the rotational effect of force on a solid body. [3]

That is, the moment of force (Fig. 2.6) is the weight of the load, which is measured in kilograms, which the servo holds motionless on a swing with a shoulder length of 1 cm. These figures indicate for the input voltage of 4.8V or 6V.

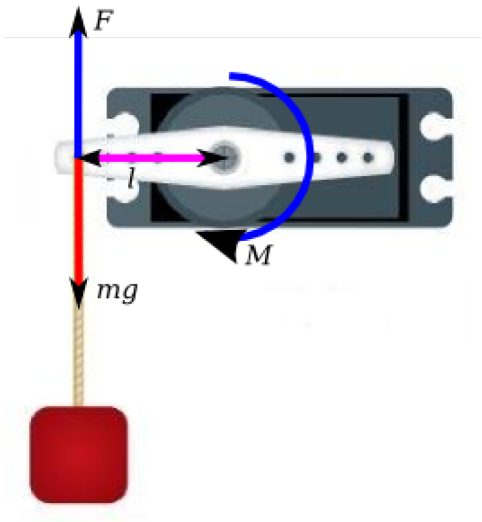

Figure 2.6 – Torque of the servo

- l – swing shoulder;
- M – torque;
- mg – force given by the load;
- F – counteracting force of the servo drive/

2.1.3 Servo selection

To develop a manipulator arm, consider 3 models of servos that best meet our needs.

The first option is a DS04-NFC servo

General Information

Constant rotation drive DS04-NFC (Fig. 2.7) – drive with a rotation angle of 360 ° and a force of 5.5 kg · cm

Figure 2.7 – DS04-NFC servo

It can be used in the construction of robots, various mechanisms, where you do not need to track the angle of rotation and need a drive capable of full rotation.

Features:

- Power supply: 4.8-6V;
- Current: up to 1000mA;
- Range of rotation: 360 °;
- Speed of rotation without loading at 4,8V: 60 ° for 0,22s;
- Speed of rotation without loading at 6B: 60 ° for 0,18s
- Moment of force at a voltage of 4.8V: 5.5 kg · cm;
- Overall dimensions: 41kh20kh40mm;
- Weight: 38g.

Connection:

- Black wire – GND;
- Red wire – 5V;
- White wire – Signal, connects to any digital output of the Arduino board;

More about the drive.

The main difference between a constant rotation drive and a servo drive is that it rotates infinitely long in the direction assigned to it, as well as that the drive does not have a potentiometer or optocoupler, so it is possible to programmatically set only the direction and speed of rotation of the drive.

From the point of view of writing sketches, this drive can be controlled using standard libraries that are part of the Arduino IDE. The main difference is that the function Servo.writeMicroseconds (PULSE_WIDTH) is used, which sets not the angle, but the direction and speed of rotation of the drive:

Servo drive MG90S

General Information

Microservoir MG-90S (Fig. 2.8) (with metal gears) – analog servomotor, which is used to control small light mechanisms, the angle of rotation of which is limited to the range from 0 to 180 degrees.

Figure 2.8 – MG90S servo

Features

- Voltage: 4.8 – 6V;
- Power supply 4.8V: 1.8kg · cm;
- Power supply 6B: 2.2 kg · cm;
- 60° rotation time:
- Power supply 4.8V: 0.10s;
- Power supply 6B: 0.08s;
- Material of gears: Brass, aluminum alloy;
- Bearings of an output shaft: 1 piece, from above;
- Electronics: Analog;

- Cable length: 0.26 m;
- Weight: 13.4 g;
- Dimensions: 22.8 x 12.2 x 28.5 mm;
- Connection;
- Brown wire – GND;
- Red wire – 5V.
- Orange wire – Signal, connects to any Arduino digital output;
- Servo drive MG996R-180 (with the increased torque)/

Servo MG996R-180 (Fig. 2.9) is an improved version of the popular digital servo, model MG995R, in accuracy and speed of positioning. Control is performed on an analog chip, not as in the original on a digital microcontroller. Which significantly reduces the price of the servo. Compared to previous models has increased torque. The drive is made of metal, and the redesigned board and control system makes the servomotor more accurate. That is why this model is often used for radio-controlled aircraft, gliders, helicopters, cars, boats, robot chassis and others. Thanks to the gearbox with metal gears on the output shaft develop a sufficient force for use in mobile robots. Double-row ball-like bearing and metal gears ensure stable and reliable operation of the drive without damage.

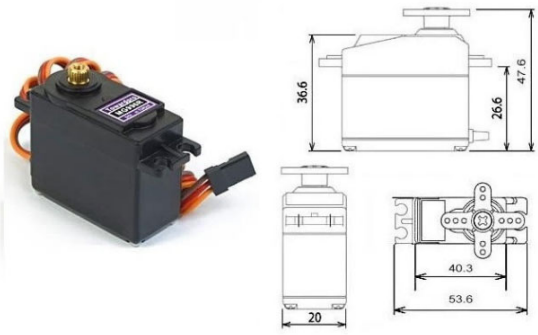

Figure 2.9 – Appearance of the MG995R servo

Servo characteristics:

- Weight: 55 grams;
- Dimensions: 40.7 x 19.7 x 42.9;

- Torque:
- 8.5 kg x cm (at 4.8 V);
- 10 kg x cm (at 6 V);
- Speed: 0.2 s / 60º (at 4.8 V), 0.16 s / 60º (at 6 V);
- Working power supply: 4.8 – 7.2 V;
- Dead zone width: 5 μs;
- Operating temperature range: 0 ºC – 55 ºC.

Connection:

- Black wire – GND (ground);
- Red wire – 5V (power supply);
- White wire – Signal (connects to any digital output of the Arduino board).

Having considered these three models of servo drives, comparing their characteristics, we conclude that the MG996R servo drive is the most suitable for designing the operation of the manipulator. In this model, the drive is made of metal, as well as a double-row ball bearing makes the servo more reliable, durable and protected from damage. The redesigned board and control system makes the servo more accurate than previous models, and the analog chip significantly reduces the price of the servo.

2.2 Selecting the layout of the robot manipulator

To implement the project, you need to choose the layout of the manipulator, which will be controlled through a broker using WiFi.

At the moment there are a large number of robot manipulators of different designs and shapes, consider 3 options for the manipulator.

In the first version (Fig. 2.10) we see the layout of the robot made of wood using a fairly accurate laser cutting. In this model, all mechanical parts are quite

versatile. The manipulator has 4 levels of freedom. Work weight 230g that allows you to use cheap servos. There is space on the back for the Arduino board. This model has very low design and mechanical characteristics, as well as a small working area.

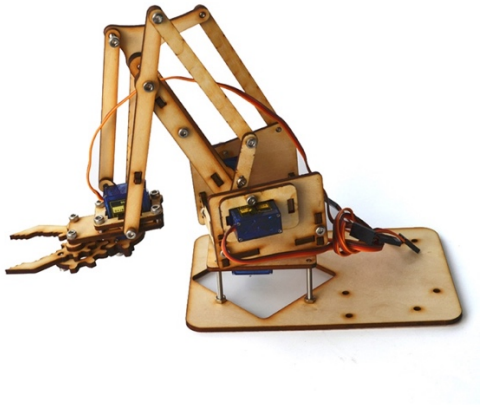

Figure 2.10 – View of the manipulator SNAM1500

In the second option we see the manipulator made of acrylic (Fig.2.11).

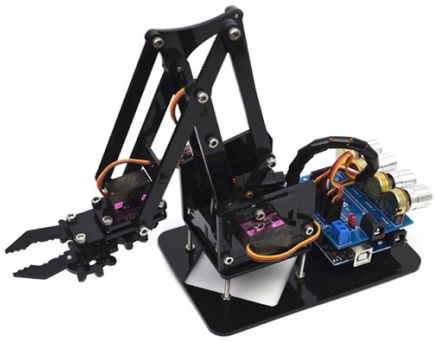

Figure 2.11 – View of the manipulator made of acrylic

This is a classic machine with 4 degrees of freedom, which has existed for a long time and is the inspiration of an industrially robotic manipulator. This is an economical, powerful and very easy to use kit for robotic learning based on Arduino. This could be a fairly simple way to control. Four knobs are used to

control the movement of the potentiometer. After connecting the robot to a computer via a USB port, you can send commands via the serial port to control the angle of rotation of the servo. This model is compatible with other software for managing it via a PC, and also supports programs for performing fixed and repeated actions.

This model is easy to assemble because it does not need to be soldered, there is no additional equipment and tools, and the default board is attached to the base.

- The system of rods, allows to place powerful servo drives, and also keeps capture parallel or perpendicularly to the basis of the manipulator;
- A simple set of components;
- Bearings are in most parts of the manipulator
- Easy to assemble.
- The grip position can be changed by 90 degrees
- Arduino compatibility.

The third option is a manipulator made of aluminum plates (Fig.2.12). This model has 5 levels of freedom. All mechanical parts of the robot are universal and can be used depending on the purpose that the manipulator will perform in future operations.

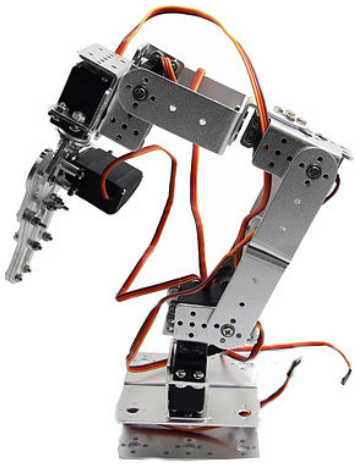

Figure 2.12 – Appearance of the manipulator made of aluminum plates

Having considered all models of robot manipulators, we choose the third option to implement the project. Next, consider in more detail its technical characteristics.

This manipulator can be considered as a demonstration platform with five levels of freedom. The power system of the manipulator consists of five servo drives which can carry out demonstration of capture, movement to the left, to the right, up and down. For stability of the manipulator, all materials are made of an aluminum plate 2 mm thick. All connections use a high-quality bearing for better performance.

The advantage of this model is also a large operating range. A schematic representation of the operating range of the manipulator is shown in (Fig. 2.13).

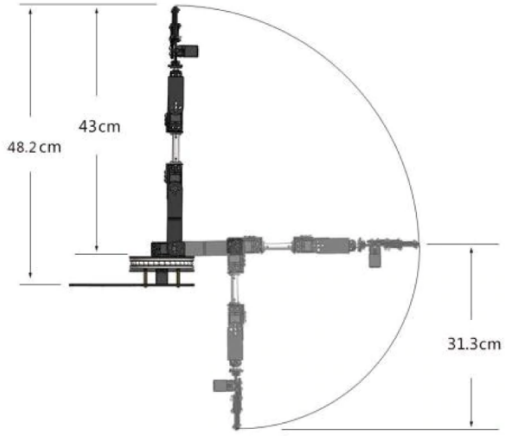

Figure 2.13 – Image of the operating range of the manipulator

The manipulator uses a metal gripper (Fig. 2.15) made of hard aluminum alloy. Servo drives of the following models are suitable for this type of capture: HS322 / HS422 / MG995 / MG996R / MG996 / MG996R-180 / MG946 / SG5010.

Capture characteristics for the manipulator:

- Weight of capture (without servo drive) – 68 g;
- The maximum step of opening of a paw – 50 mm;

- Length of capture in the closed condition – 118 mm;
- Length of capture in an open condition – 107 mm;
- Width of capture in the closed condition – 55 mm.

Figure 2.14 – Metal grip for the manipulator

3. DESIGN PART

3.1 Robot manipulator on Arduino

3.1.1 General description of the project

In the project of operation of the manipulator 5 servo drives are used. For the mechanical part, I used a set of "5 DOF robot manipulator" made of aluminum alloy 2 millimeters thick. Arduino UNO and ESP8266 microcontrollers are used to control the robot.

3.1.2 The main nodes for the project are the work of the manipulator

To assemble the manipulator, we need:

- 5 servo drives of the MG996R brand (in this model the increased torque, high service life);
- A set of metal brackets for assembling the robot. For their full composition, see in table 2;
- Working stand for the robot;
- Power Supply;
- Arduino UNO board;
- Microcontroller ESP8266 Node MCU v3;
- Arduino Motor Shield expansion board;
- The switch 220B;
- A set of wires;
- Wi Fi antenna;
- Cables of different types as well as some basic tools.

Table 3.1 – Set of components for hand manipulator

Number	Name of components
4	multifunction servo bracket
2	long u-shaped servo bracket
2	l-shaped servo bracket
1	U-bracket
2	U-shaped bracket Bracket
3	Deep groove ball bearings
1	Aluminum claw clamp
6	metal horns
1	flange rod
3	servo extension
1	winding belt
1	set of screws and nuts

3.2 Collecting the layout of the robot manipulator

Having all the components we can move on to design. Their complete set is shown in Figure 3.1.

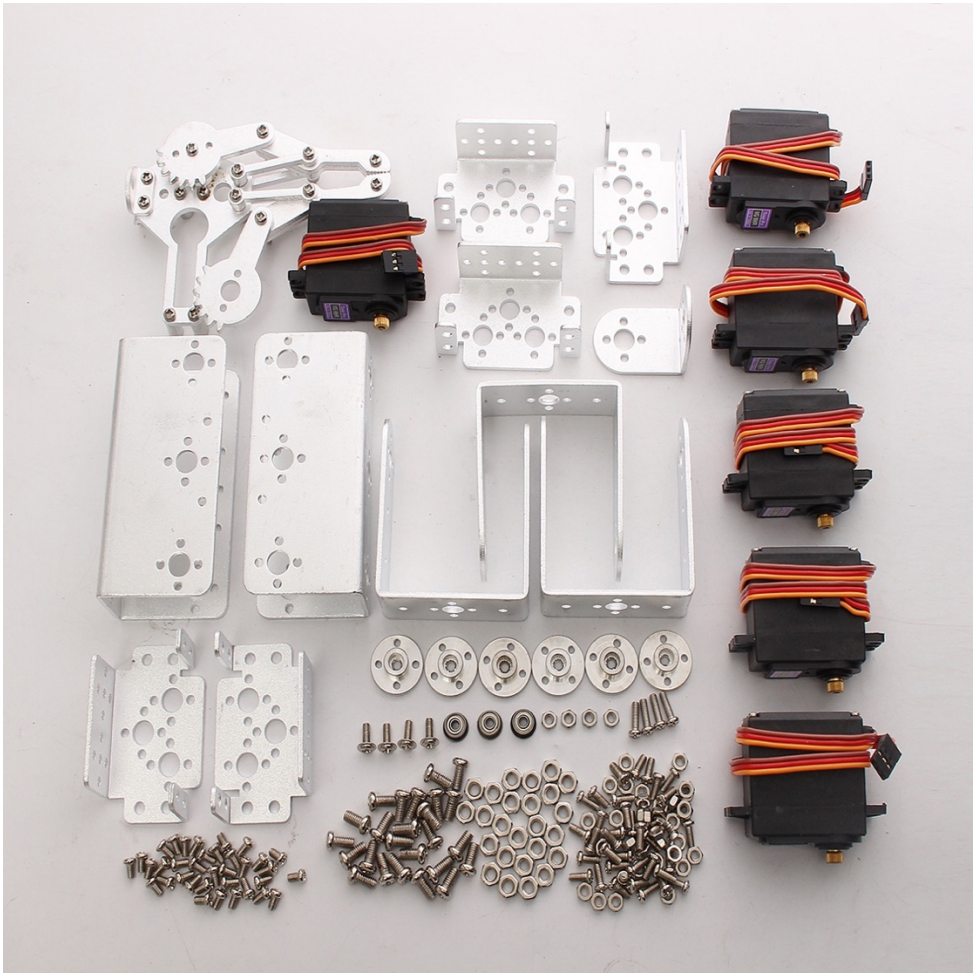

Figure 3.1 – A set of components of the robot manipulator

To begin with it is necessary to collect the manipulator. For this purpose, we use accessories from table 2 and servo drives MG996R. Assembly of the manipulator is carried out according to the instruction. I want to pay special attention to collecting the grip for the manipulator.

To assemble the gripper, it is necessary to shorten the swing from the servo drive to such a size that it fits the gripper and tighten it with bolts.

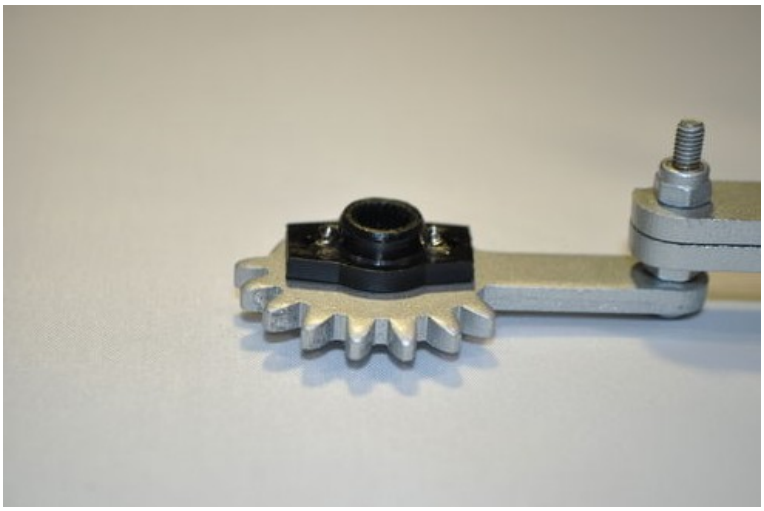

Figure 3.2 – Attaching the swing to the servo

After installing the servo, return the grip to the extreme position by pre-squeezing it (Fig. 3.3). You can now tighten the servo to the four bolts while holding it in the extreme position and with a compressed grip. To check the efficiency, connect to the Arduino board.

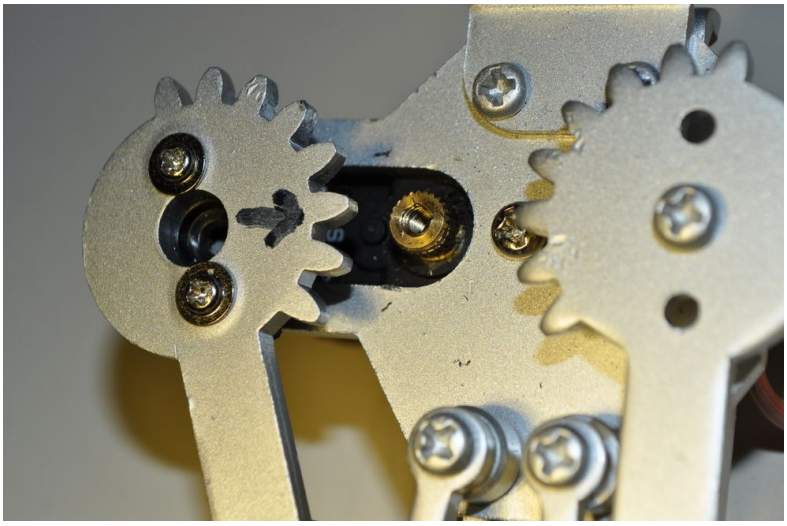

Figure 3.3 – Collecting the grip for the manipulator

The collected hand manipulator is seen in (Fig. 3.4).

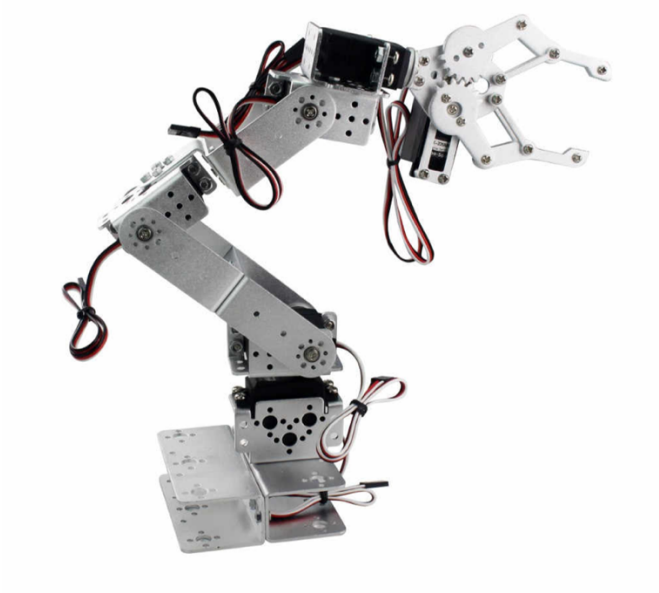

Figure 3.4 – Robot manipulator

Once the robot is assembled, attach it to the work stand. As a stand, I chose an old training stand. Attach the work in the middle of the stand.

A power supply can be attached to the back of the robot. As a power supply, I chose Gamemax GM-400 8CM (Fig. 3.5). This is a great choice for office PCs or entry-level gaming computers. The power of the BZ is 400 watts. Voltage on the lines of 3.3 V and 5 V. A fan with a diameter of 120 mm is used to cool the power units.

Figure 3.5 – Power supply

The next stage of assembly will be the assembly of the electrical circuit of all components.

Since there is a lot of free space in the middle of the power supply, all electrical components will be located there. First, attach the board Arduino UNO and ESP8266 Node MCU v3 to the power supply cover (Fig. 3.6). To do this, mark and drill holes in the lid. Attach the board with bolts and nuts. Since in the middle of the metal housing of the power supply will weaken the Wi-Fi signal to the board ESP8266, remove the antenna from the housing for the best signal.

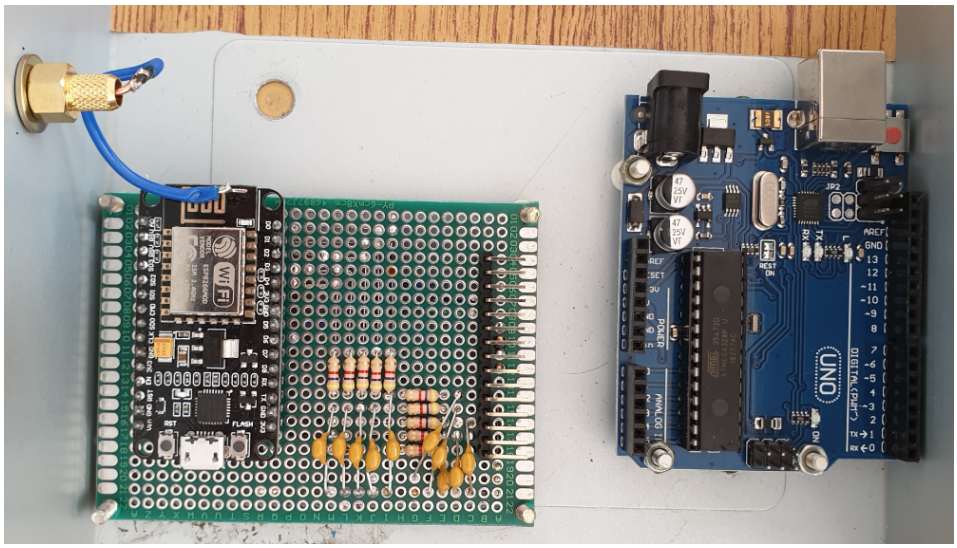

Figure 3.6 – Attached boards on the cover of the BZ

On top of the Arduino UNO install the expansion board shield sensor v4. Connect the servo drives and the ESP8266 Node MCU v3 microcontroller to this expansion board (Fig. 3.7). The outputs of the servos are connected to pins with pulse-width modulation of the Arduino board. Using the symbols on the board and servos, connect the servos as follows

- Black wire to the foam "G" – earth;
- Red wire to the foam "V" – 5V;
- Orange wire to the foam "S" – signal.

The 5 outputs of the ESP8266 Node MCU v3 are connected to the analog outputs on the expansion board. 1 more output to digital output. It will serve as a switch for ESP8266. At the same time, do not forget to correctly indicate in the program the names of the pins that we used when connecting the boards.

The red and black wires from the power supply and the ESP8266 board are also connected to the shield sensor v4.

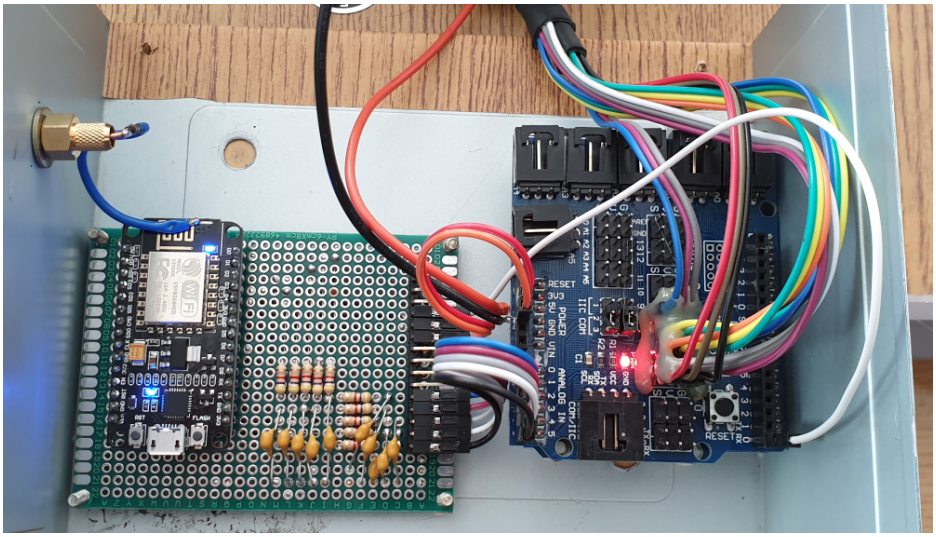

Figure 3.7 – connection of all components

Table 3.2 – Connecting the pins of the ESP8266 board to the expansion board

ESP8266 Node MCU v3	Shield sensor v4
D0	1 (ANALOG IN)
D1	2 (ANALOG IN)
D2	3 (ANALOG IN)
D3	4 (ANALOG IN)
D4	5 (ANALOG IN)
D5	0 (DIGITAL IN)
G	GND
VIN	5V

The complete electrical diagram can be seen in (Fig. 3.8).

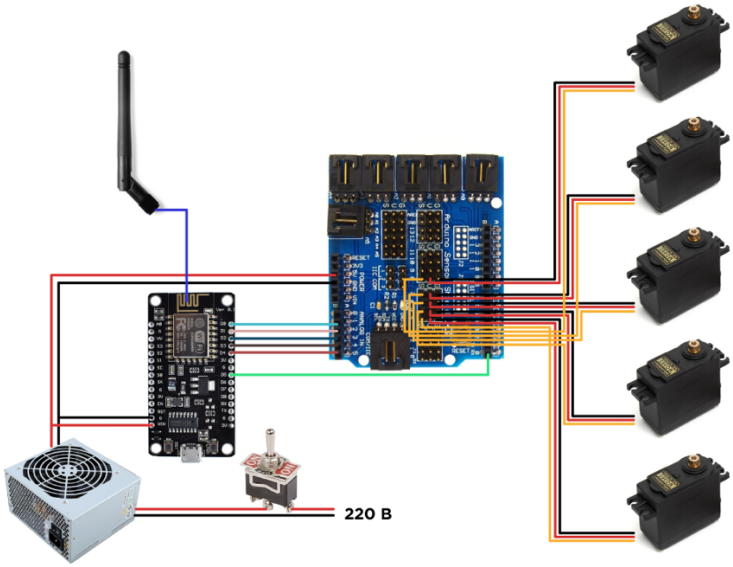

Figure 3.8 – Wiring diagram

Arduino UNO and ESP8266 Node MCU v3 must be flashed before installing the cover. We flash the Arduino UNO board with a program that will set the initial position of the manipulator arm when power is applied to the board. And on the board ESP8266 Node MCU v3 will be stitched program that will receive data from the broker and transfer them to the Arduino UNO.

Before connecting the Arduino board via a USB cable to a personal computer, turn off the power. During the test of the program, we also turn off the power of the hand. Even when the power is off, the Arduino will receive 5 V of current from the USB. Accordingly, the power from the PC transmitted via USB will go to the power supply and it will "sag" a bit.

After stitching the boards, install the cover on the power supply. For convenience in use of the robot we establish the switch "toggle switch" which will break power supply to the power supply unit from a network.

The robot manipulator in assembled form is shown in Figure 3.9.

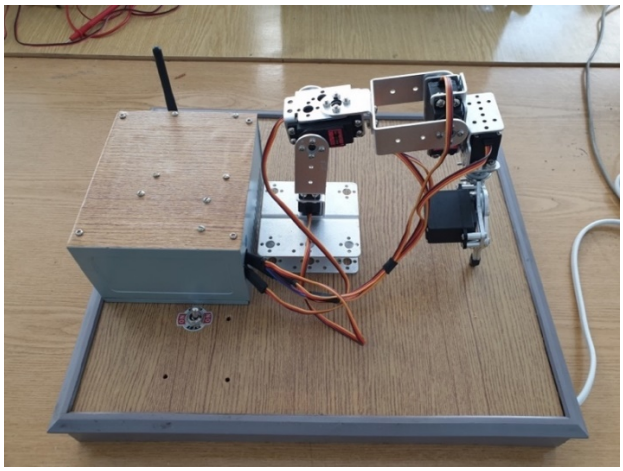

Figure 3.9 – Image of a fully assembled manipulator

3.3 Manipulator operation algorithm

Let's make a more detailed review of work of the manipulator hand designed by us. In Figure 3.10 we see the algorithm in the form of a block diagram.

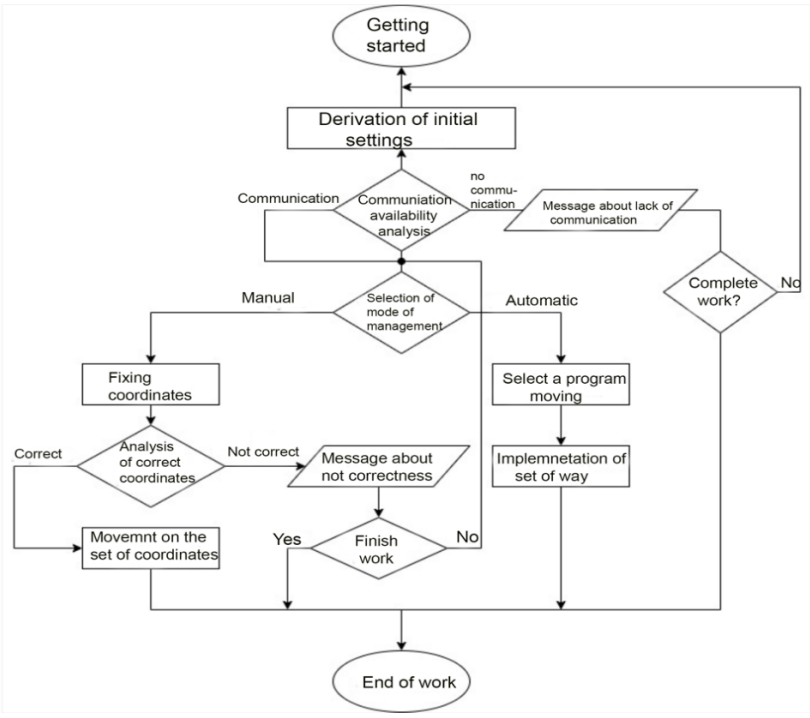

Figure 3.10 – Image of the block diagram with the algorithm of the manipulator

As you can see in Figure 3.10, the start of the manipulator begins with the inclusion, input of initial settings and self-testing of the robot. Once the settings are complete, the existing connection is analyzed. If there is no connection, the manipulator is re-analyzed or shut down. If there is a connection, the control mode is selected: automatic or manual. Automatic mode is the movement of parts of the robot on a certain trajectory, as pre-programmed in the firmware by program code. Choosing manual control, there is a fixation of coordinates and their analysis of correctness. When the coordinates are set correctly, the manipulator performs the specified movement, if the coordinates are not correct, the program terminates the work, or returns to the initial stage.

3.4 Data transmission via MQTT protocol

Reception and transfer of data from the module to the broker is carried out by means of the WIFI Internet network on the MQTT message protocol. The functional diagram of the connection of devices to the controller is shown in Figure 3.11.

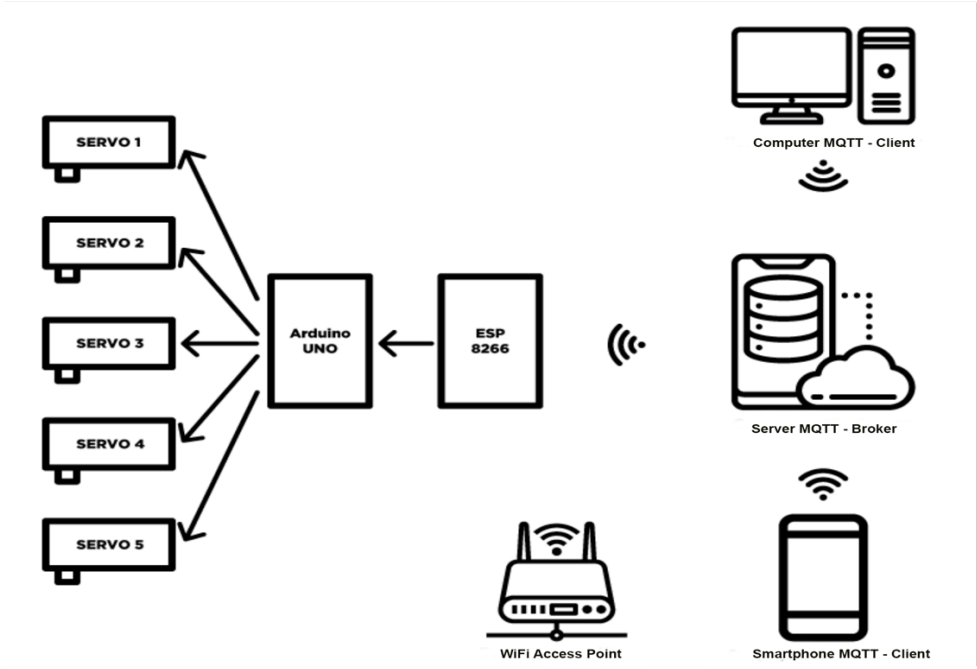

Figure 3.11 – Functional diagram of the connection of devices to the controller

List of main functions performed by the module:

- Device initialization;
- Receiving messages from the broker;
- Signal to the outputs of the module that controls the servos;
- Reading data;
- Data transfer to the broker;
- Periodic informing the broker about the state of operation of the devices.

3.5 Algorithm of mqtt protocol operation

When power is applied, the device is initialized. At this stage, the module is also connected to the available Wi-Fi network, and mqtt is started – the client, which sends the output status (default sending time interval 2c), the interval can be changed depending on the task.

When a command comes from a broker, it is executed and the device goes into standby mode for a new command. If the command does not come, the device continues to be in the previous mode. The algorithm, in the form of a block diagram is presented in Figure 3.12.

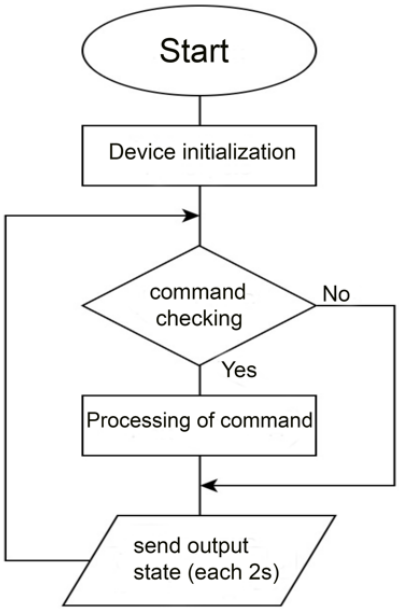

Figure 3.12 – Block diagram with the algorithm of the module

3.6 Launch, configure and send messages via Mosquitto broker on the Windows operating system

To start the broker, go to the command line and enter the cd and the directory in which mosquito is installed, in our case it is C: \ Program Files \ mosquitto (Fig. 3.13).

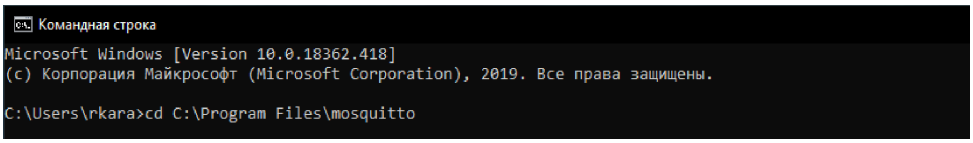

Figure 3.13 – Mosquito is installed

Next, write mosquitto and press Enter (Fig. 3.14).

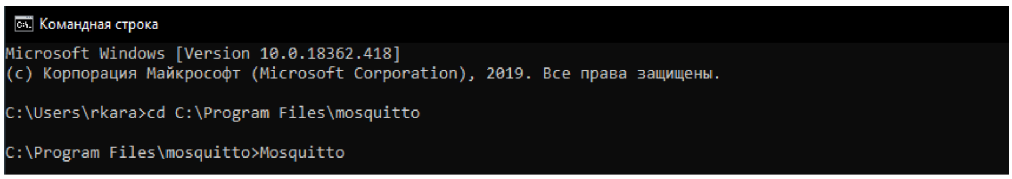

Figure 3.14– Write mosquitto and press Enter

Broker Mosquitt MQTT is ready to work.

We will test whether the transmission and receipt of messages in mosquitto. To do this, open two command lines. In the first command line, subscribe to the topic "mytopic" by entering the command mosquitto_sub -t mytopic (Fig. 3.15).

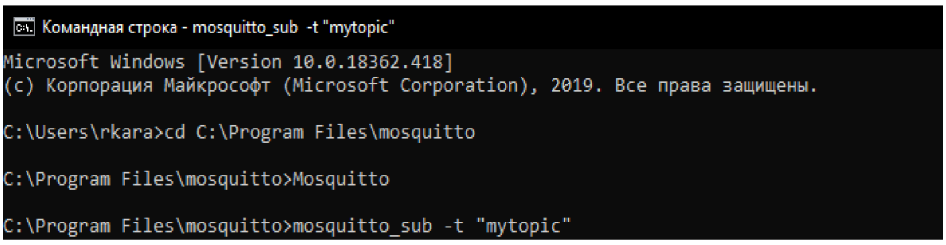

Figure 3.15 – Command mosquitto_sub -t mytopic

In the second command line, publish the topic "mytopic" and pass the value "1". To do this, enter the command mosquitto_pub -t mytopic -m 1 and press Enter (Fig. 3.16).

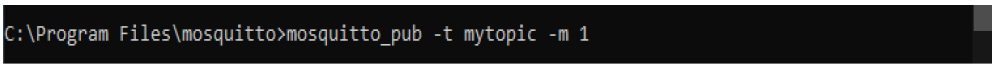

Figure 3.16 – Command mosquitto_pub -t mytopic -m 1

As you can see, the value "1" appeared in the first command line, so everything works fine (Fig. 3.17).

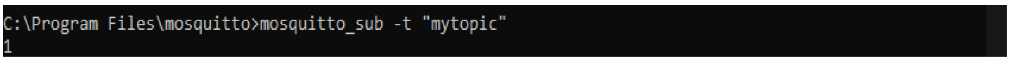

Figure 3.17 – Check for correctness

3.7 Description of the client application on the Android operating system

We use the MQTT dashboard application that allows you to connect to the MQTT broker Mosquitto and receive / send data to specific topics. The peculiarity of this application is that it is easily scalable and the average user can publish or receive messages without knowledge of programming.

We are creating a new project. Enter the project name, mqtt broker address and port (Figure 3.18).

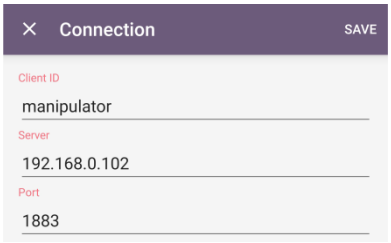

Figure 3.18 – Tab for creating a new project

On the subscribe tab we create 5 objects specifying the name of the topic (Fig. 3.19).

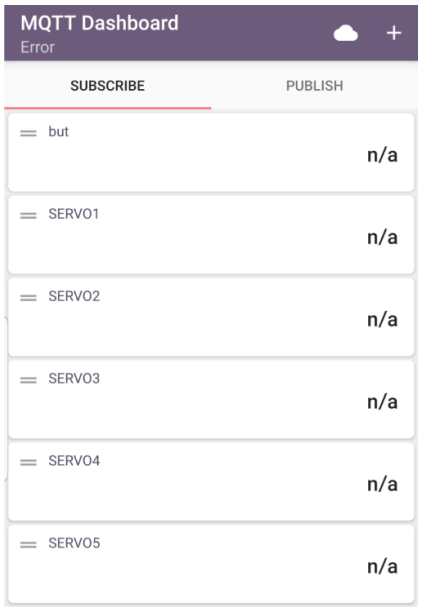

Figure 3.19 – Subscribe tab

Table 3.3 – The names of objects and their corresponding names of topics that will be accepted

Object name	Topic name
Servo1	inTopic1
Servo2	inTopic2
Servo3	inTopic3
Servo4	inTopic4
Servo5	inTopic5
But1	inTopic6

On the Publish tab, create 5 objects of the "slider" type, specifying the name of the topic and the minimum and maximum value that they will publish, as well as a switch with possible values of "1" and "0" (Fig. 3.20).

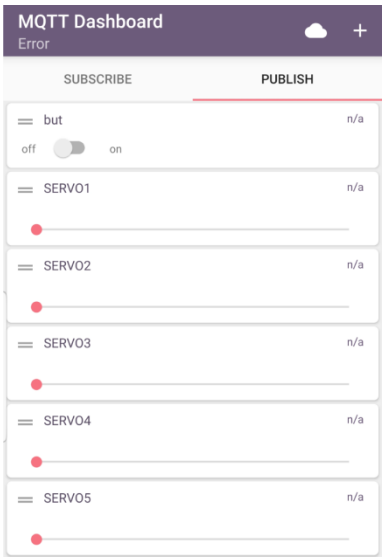

Figure 3.20 – Publish tab

Table 3.4 – The names of objects and their corresponding names of topics with maximum and minimum values to be issued

Object name	Topic name	Min value	Max value
Servo1	inTopic1	0	255
Servo2	inTopic2	0	255
Servo3	inTopic3	0	255
Servo4	inTopic4	0	255
Servo5	inTopic5	0	255
But1	inTopic6	0	1

3.8 ESP8266 microcontroller

When designing the manipulator arm, we decided to use the ESP8266 Node MCU v3 module as a controller (see Fig. 3.21), because: it contains a sufficient number of inputs and outputs; contains a built-in stack of TCP / IP protocols; contains several serial UART ports that will be required for system debugging;

provides satisfactory computing power. Also, a big advantage of the board is its low power consumption. They are often used in autonomous power circuits.

The platform has a modern API for hardware input and output. This reduces the number of actions when working with the equipment and when setting it up. With the NodeMCU firmware, you can use all the working potential for rapid development of the device.

Figure 3.21 – Appearance of the controller ESP8266 NodeMCU v3

3.8.1 Technical characteristics of the ESP8266 NodeMCU module:

The main technical characteristics of the board.

- Support for Wi-Fi 802.11 b / g / n protocol;
- Support for Wi-Fi mode – client, access point;
- Input voltage 3.7 V – 20 V;
- Operating voltage 3V-3.6V;
- Maximum current 220mA;
- Built-in TCP / IP stack;
- Range of working temperatures from -40C to 125C;
- 80 MHz, 32-bit processor;
- Time to wake up and send 22MS packets;
- Built-in TR switch and PLL;
- Availability of power amplifiers, regulators, power management systems.

There are several generations of NodeMcu boards – V1 (version 0.9), V2 (version 1.0) and V3 (version 1.0). Symbols V1, V2, V3 are used when selling in online stores. There is often confusion in the boards – for example, V3 is outwardly identical to V2. Also, all boards work on the principle of open-source, so they can be produced by any company.

Power supply of the ESP8266 NodeMcu v3 microcontroller

You can supply power to the module in several ways:

- Apply 5-18 volts through the pin "Vin";
- 5V via USB connector or VUSB contact;
- 3.3 through 3V output.

3.8.2 Advantages and disadvantages of the NodeMcu v3 module

Advantages of the module:

- The presence of the UART-USB interface on the board with a built-in micro-USB connector allows the user to easily connect the board to a personal computer.
- Availability of 4 MB of flash memory.
- Ability to update the firmware of the microcontroller via USB cable.
- It is possible to create scripts in the LUA programming language and save them in the file system.

Disadvantages of the module

The main disadvantage is the ability to execute only LUA scripts located in RAM. This type of memory is small, the volume is only 20 KB, so writing large scripts causes a number of difficulties. First of all, the whole algorithm will have to be divided into linear blocks. These blocks must be written to separate system files. All these modules are executed using the dofile statement.

When writing, you need to follow the rules – when exchanging data between modules, you need to use global variables, and when calculating within modules – local. It is also important to call the collectgarbage function at the end of each written script.

NodeMcu v3 temperature regulators

The V3 module has 11 general-purpose I / O contacts. In addition, some of the outputs have additional functions: (Fig. 3.22)

- D1-D10 – outputs with pulse width modulation;
- D1, D2- outputs for I²C / TWI interface;
- D5-D8 – outputs for the SPI interface;
- D9, D10 – UART;
- A0 – ADC input.

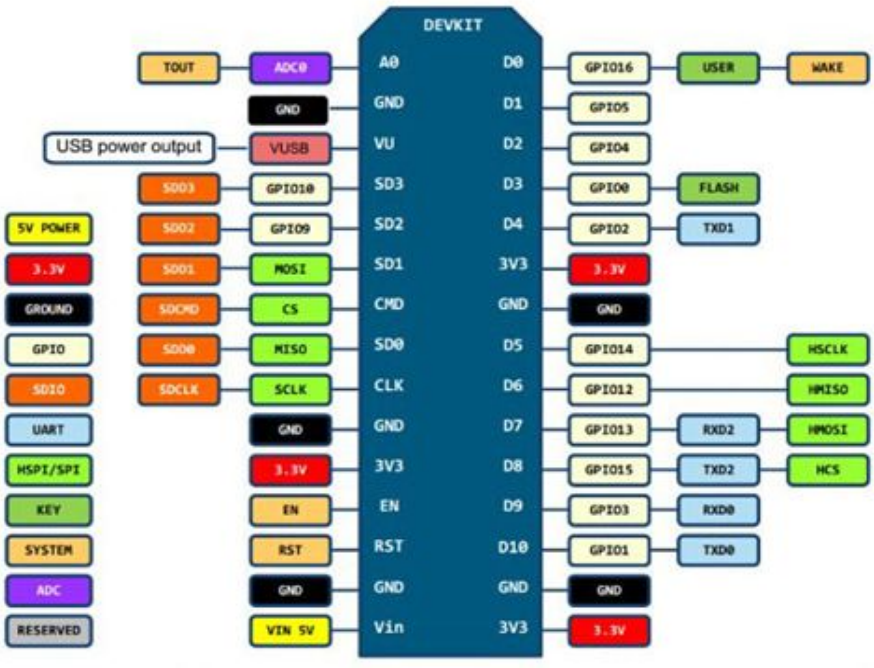

Figure 3.22 – Layout of pins ESP8266 NodeMCU v3

"The microcontroller does not have non-volatile memory for the user on the chip. Execution of the program is carried out from the external SPI ROM by dynamically loading the necessary intervals of the program in the cache of instructions. Downloading is performed hardware, transparent to the programmer. Up to 16 MB of external program memory is supported. Standard, Dual or Quad SPI interface is possible. The manufacturer does not provide documentation for the internal peripherals of the controller. Instead, it provides a set of libraries through which the programmer can access peripherals through an API. ”

The microcontroller is programmed in the Arduino IDE environment in the C programming language. The microcontroller is connected to the computer via microUSB, which provides its power, and through this connector it is programmed.

3.9 Control system programming

3.9.1 Description of the Arduino IDE programming environment

To create developments based on Arduino, you need software to write and download programs to the microcontroller. These functions are performed using the Arduino IDE programming environment. The Arduino programming environment consists of a text editor for writing program code, a toolbar, and a text output window. To download programs, you must connect the development environment to the hardware of the microcontroller. Any program written in the Arduino IDE is called a sketch. The sketch is written using a text editor, which contains tools that allow you to perform any actions on it, such as inserting / deleting, replacing / searching for text. When saving or compiling a sketch, error messages appear in the message area if the program code is misspelled. The

Arduino programming language is standard C ++, which has features that make it easier to write programs:

- Program files are processed by the Arduino preprocessor before compilation;
- The programmer is obliged to write two obligatory functions. This setup () is called at startup and loop () is repeated in an infinite loop;
- It is not necessary to write titles in the text of the program when using standard libraries;
- No pre-compiler settings.

The sketch is downloaded through the built-in bootloader (Bootloader), is a program that allows you to download program code without the use of any additional hardware. And can also work via RS-232, USB and Ethernet interfaces. The bootloader is active for a period of a few seconds when loading any sketch into the microcontroller. The operation of the bootloader is signaled by an LED built into the microcontroller board.

To find a solution to the problem of controlling the hand manipulator in the Arduino IDE, you must perform the following steps of configuration:

- Creation of the project;
- Equipment configuration;
- Connecting devices to the network;
- Microcontroller programming;
- Visualization settings;
- Download configuration data;
- Diagnosis of various functions of the equipment.

The program interface provides easy navigation of tasks and project data

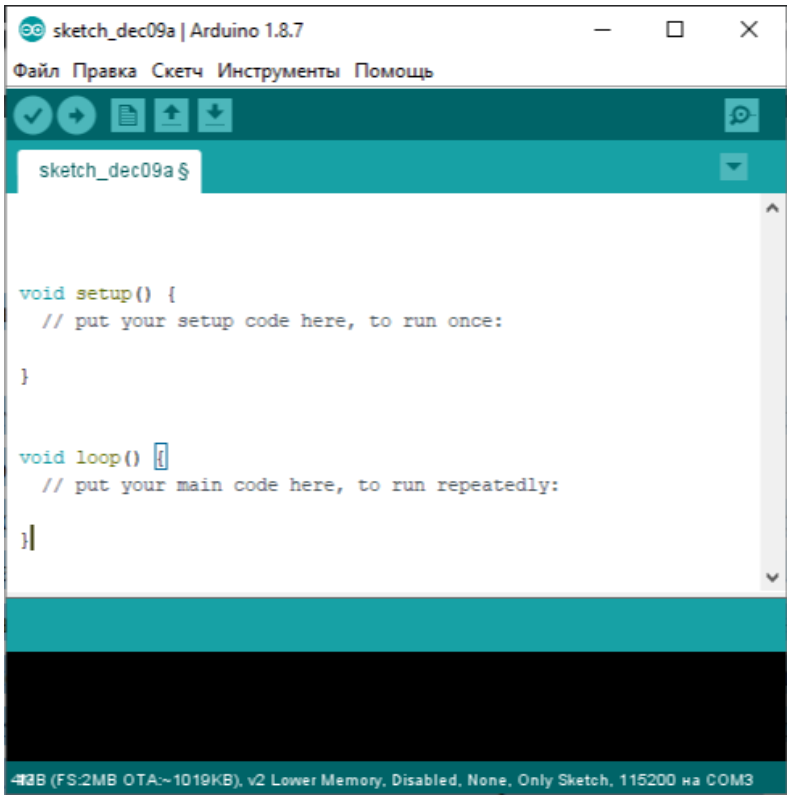

Figure 3.23-Picture of the structure of the interface for creating a new sketch

1) Toolbar buttons that allow you to check, download, create a program, open, close or save a sketch, as well as open port monitoring;

2) A text editor designed to write a sketch;

3) Console;

4) The area of messages with information about the selected in the current microcontroller output port.

3.9.2 Development of a hand-manipulator control program

Next, the stages of creating a project with servo control will be described in detail. The data and program code developed during the project creation

process are written in order. The I / O port numbers recorded below were used when working with the ESP-8266 microcontroller.

The first step is to connect the necessary libraries, initialize the Wi-Fi and MQTT client and specify the network name, password if the network is not open and the mqtt address of the client (Fig. 3.24).

The #include directive prompts the compiler to include another source file, the name of which is specified after the directive.

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

const char *ssid =  "Android";
const char *password =  "";
const char *mqtt_server = "192.168.43.205";

WiFiClient espClient;
PubSubClient client(espClient);
```

Figure 3.24 – Fragment of the written code of the program of connection of libraries, initialization of Wi-Fi and MQTT of the client

The next step is to initialize the output ports of the Arduino microcontroller. For more convenient orientation in the program code by means of the "#define" directive to each involved exit the names which more explicitly express their performed functions will be appropriated.

Table 3.5 – Table of the #define directive for the manipulator hand control sketch program.

Output name	Output number	Description
SERVO1	16	PWM output that supplies a control signal with a given pulse width
SERVO2	5	PWM output that supplies a control signal with a given pulse width
SERVO3	4	PWM output that supplies a control signal with a given pulse width
SERVO4	0	PWM output that supplies a control signal with a given pulse width
SERVO5	2	PWM output that supplies a control signal with a given pulse width
BUTTON1	14	Digital output that sends a signal to turn on / off the servos

```
#define SERVO1 16
#define SERVO2 5
#define SERVO3 4
#define SERVO4 0
#define SERVO5 2
#define BUTTON1 14
```

Figure 3.25 – Recording the initialization in the program inputs / outputs of the microcontroller directive "#define"

Pin 14 is a digital pin configured using the "pinMode ()" and "digitalWrite ()" functions, which acts as a signal to turn on / off the hand-operated servos.

The PWM outputs 16, 5, 4, 0, 2 supply a control signal to the driver with a corresponding pulse width to control the rotation angle of the servomotors using the "analogWrite ()" function.

Next, use the callback function to obtain topics from the broker. The function is very convenient because it writes common code that does not depend on the logic in the called function and can be reused with various inversions.

The function "Serial.print (topic)" displays the name of the topic in the series print, output the va§lue of the obtained data, get the name of the topic (Fig. 3.26)

```
void callback(char* topic, byte* payload, unsigned int length)
{
  Serial.println();
  Serial.print(topic);
  Serial.print(" => ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }

  String strTopic = String(topic);
```

Figure 3.26 – The function of obtaining topics from the broker

The block on (fig. 3.27) is used for initialization of servodrives. This block checks whether the data came from the desired topic. If so, the read values are read from it, converted to PWM, the values 0-255 are converted to the value 0-1023 and the PWM signal level is set. This program code is duplicated 5 times, because 5 servos are used to implement the manipulator arm. In each subsequent block, change the name of the topic to the corresponding name of the servo

```
if (strTopic == "inTopic1")
{
  payload[length] = 0;
  String strPayload = String((char*)payload);
  int val = strPayload.toInt();
  int stled = map(val, 0, 255, 0, 1023);
  analogWrite(SERVO1, stled);
  Serial.print(" => ");
  Serial.print(stled);
}
```

Figure 3.27 – Fragment of the program code for servo control

During the development of the project there was a need to create a switch that could, when changing the values of 1 and 0, turn on or off all servos. The implementation of such code is shown in (Fig. 3.28).

```
if (strcmp(topic, "inTopic6") == 0) {
  if ((char)payload[0] == '1') {
  digitalWrite(BUTTON1, HIGH);
 }else {
  digitalWrite(BUTTON1, LOW);
}
}
```

Figure 3.28 – Fragment of the program code for the implementation of the switch

The "setup ()" function, which is called when the sketch is started. Inside the curly braces is the code that initializes the variables as soon as the program starts, determines the mode of operation of the controller outputs, launches the libraries used. Starts only once after power is applied to the microcontroller.

By downloading the program, Arduino gives our code the opportunity to participate in the initialization of the system. To do this, we must specify to the microcontroller the commands that it executes at boot time and then forgets about them. And for this purpose, in our program with you we should allocate the block in which these commands will be stored. void setup (), or rather the space inside the curly braces of this function, and is such a place inside the Arduino sketch.

The pinMode function is used to initialize the output mode of the microcontroller board outputs. The pin entry initializes the input of the microcontroller to be used in operation. Mode sets the input or output operation mode.

In (Fig. 3.29) we see the initialization of variables responsible for servos and the button-switch. The mode of operation is set to the output.

```
void setup() {
  pinMode(BUTTON1, OUTPUT);
  pinMode(SERVO1, OUTPUT);
  pinMode(SERVO2, OUTPUT);
  pinMode(SERVO3, OUTPUT);
  pinMode(SERVO4, OUTPUT);
  pinMode(SERVO5, OUTPUT);
  Serial.begin(115200);
}
```

Figure 3.29 – Fragment of the program code of the function "setup ()" to initialize variables

The loop function () is repeated in an infinite loop. It initializes the initial values of variables, performs calculations in the program and responds to them. Starting with the first command, the microcontroller will reach the end and immediately jump to the beginning to repeat the same sequence. And so many times (as long as the board will be supplied with electricity).

In the middle of the block function "loop ()" write the program code that will be responsible for connecting the controller to the Wi-Fi network and the MQTT broker. Subscribe to topics using the "client.subscribe (" ");" function and specify the client id (must be unique in the network) using the function "client.connect (" ");" The last fragment of the program is responsible for checking the connection and starting the cycle of obtaining topics (Fig. 3.30).

```
void loop() {
  if (WiFi.status() != WL_CONNECTED)
  {
    WiFi.begin(ssid, password);
    if (WiFi.waitForConnectResult() != WL_CONNECTED)
      return;
  }
  if (WiFi.status() == WL_CONNECTED)
  {
    if (!client.connected())
    {
      client.setServer(mqtt_server, 1883);
      client.setCallback(callback);
      client.connect("ESP03led");
      client.subscribe("inTopic1");
      client.subscribe("inTopic2");
      client.subscribe("inTopic3");
      client.subscribe("inTopic4");
      client.subscribe("inTopic5");
      client.subscribe("inTopic6");
    }
    if (client.connected()) {
      client.loop();
    }
  }
}
```

Figure 3.30 – Fragment of the program code of the function "loop ()"

4. LIFE SAFETY

4.1 Safety rules when working with the manipulator

It is very important to follow the safety rules when working with a computer and a manipulator when performing a diploma project.

A personal computer differs from other electrical appliances in that it provides the possibility of continuous operation without disconnection from the mains. In addition to normal operation, the computer may be in low-power mode or sleep mode. The power supply unit of the manipulator works from a standard power supply network 220V. Due to the possibility of long-term operation of the computer and the manipulator without disconnection from the mains, special attention should be paid to the quality of power supply.

It is inadmissible to use low-quality and worn-out components in the power supply system, as well as their surrogate substitutes: sockets, extension cords, adapters, tees. It is not allowed to modify the sockets for connecting plugs that meet other standards. Electrical contacts of sockets should not experience the mechanical loadings connected with connection of massive components (adapters, tees, etc.). All power cords and wires should be located on the back of the computer and peripherals. Their placement in the user's work area is not allowed.

It is not allowed to work with any operations related to connecting, disconnecting or moving the components of the computer system without first turning off the power.

Also, the computer and the manipulator should not be installed near electric heaters and heating systems.

It is inadmissible to place foreign objects on the system unit, the power supply unit of the manipulator, the monitor and peripheral devices: books, sheets

of paper, napkins, covers for protection against dust. This leads to permanent or temporary blockage of the vents.

Features of the monitor power supply. The monitor has elements that can store high voltage for a long time after disconnection from the mains. Disclosure of the monitor by the user is not allowed under any circumstances. This is not only life-threatening, but also technically useless, as there are no organs inside the monitor that can be adjusted or adjusted by the user to improve its performance. Dissection and maintenance of monitors can be done only in special workshops.

All components of the system unit receive electricity from the power supply. A PC power supply is a stand-alone node

at the top of the system unit. Safety regulations do not prohibit opening the system unit, such as installing or upgrading additional internal devices, but this does not apply to the power supply. The computer power supply is a source of increased fire danger, so it is subject to disclosure and repair only in specialized workshops. The power supply has a built-in fan and vents.

In this regard, it inevitably accumulates dust, which can cause a short circuit. It is recommended to periodically (once – twice a year) use a vacuum cleaner to remove dust from the power supply through the vents without opening the system unit. It is especially important to do this operation before each transport or tilt of the system unit.

4.2 Workplace requirements

The requirements for the workplace include the requirements for the desktop, seat (chair, armchair), armrests and legs. Despite the seeming simplicity, to ensure the correct placement of elements of the computer system and the correct fit of the user is extremely difficult. A complete solution to the problem requires additional costs comparable in magnitude to the cost of individual

components of a computer system, so in everyday life and at work, these requirements are often neglected.

The monitor must be installed directly in front of the user and not require turning the head or body. The manipulator must be installed on the side of the user.

The desktop and seat should be so high that the user's eye level is slightly above the center of the monitor. The monitor screen should be viewed from top to bottom, not the other way around. Even short-term work with a monitor set too high leads to fatigue of the cervical spine.

The keyboard should be placed at such a height that the fingers are placed on it freely, without tension, and the angle between the shoulder and forearm was 100 ° – 110 °. Prolonged use of the keyboard may cause fatigue of the tendons of the wrist. Known severe occupational disease – carpal tunnel syndrome, associated with incorrect position of the hands on the keyboard. To avoid excessive loads on the brush, it is desirable to provide a work chair with armrests, the height of which, measured from the floor, coincides with the level of the height of the keyboard.

When working with the mouse, the hand should not be on the weight. The elbow of the hand or at least the wrist should have firm support. If it is difficult to provide the necessary location of the desk and chair, it is recommended to use a mouse pad with a special support roller. The duration of continuous work with a personal computer without regulated breaks should not exceed 2 hours.

Before starting work, make sure that there is a protective ground and connection of the screen conductor to the processor case; check the correct installation of the table, chair, screen angle, keyboard position, adjust the frequency, brightness, contrast, sound level.

It is forbidden to start work at detection of malfunction of the equipment. To reduce the impact of harmful factors, it is recommended to prepare the workplace so as to eliminate awkward postures and prolonged stress, set the

display so that the bottom of the screen is 20 cm below eye level and the manipulator so as not to touch any objects.

During work it is necessary to adhere to such sequence of switching off of computer equipment: to close all active tasks, control programs of the manipulator; turn off the power to all peripherals; turn off the power to the system unit (processor); turn off the power supply of the manipulator. After working on a personal computer, you need to inspect and organize the workplace. Put the hand of the manipulator in the initial working position. [8]

Before switching on the manipulator, it is necessary to visually check the power cord for mechanical damage. All electrical appliances must be securely grounded in accordance with the rules of the appliance. It is forbidden to work with electrical appliances with wet hands. Do not leave the manipulator and the personal computer (which acts as a server) unattended for a long time, after work check that all devices are turned off. It is strictly forbidden to perform any repair work yourself.

Saving the life of a person affected by an electric shock in many cases depends on the speed and correctness of the actions of caregivers. First of all, it is necessary to release the victim from the electric current as soon as possible. If it is not possible to disconnect the electrical equipment from the mains, the victim must be immediately released from the live parts without touching the victim.

4.3 General safety requirements when working with the manipulator

The source of danger to life when working with computers is a supply voltage of 220 – 380 V. To work with a computer and manipulator are allowed persons who:

a) read the instructions for working with a computer and received on-the-job training;

b) have mastered the appropriate practical course required to work on computers.

Safety requirements when working on computer remotes:

- do not switch on or off the cable connectors at the supplied supply voltage;
- do not leave computers unattended.

Before connecting the computer to the network, make sure:

- in the presence of grounding of devices;
- in serviceability of a power cord, a cord of communication of the keyboard with the power supply unit;
- turn on the power.

In the event of an electric shock, act immediately (see: Safety precautions when using electrical appliances).

In case of fire, it is necessary:

a) use all available fire extinguishers, except water and fire extinguisher OHP-10;

b) if necessary, call the fire brigade by phone 01.

It is strictly forbidden:

a) plug in a computer with a faulty power cord;

b) connect and disconnect cable connectors;

c) carry out any repairs when turning on the computer.

GENERAL CONCLUSIONS FOR THE THESIS

In the course of the work all the tasks were solved. Namely, a model of hand-manipulator operation was constructed, the data transmission and control of which was implemented using the MQTT message protocol. Also, for the robot model the program code by means of which control of the manipulator is carried out was written. The software was tested using a local broker implemented on a local PC and smartphone (MQTT – Mosquitto broker), MQTT – client, namely the Android application "mqtt dashboard" is installed on a personal smartphone.

A comparative description of the most popular models of robot manipulators was made, as well as a selection of servos. Given the fact that the use of manipulator robots is a way to automate processes in production, as well as their control, I think this topic is quite relevant. In addition to the above, this model of the manipulator is set up for further work and for study purposes.

REFERENCES

1. Arduino [Electronic resource] //Sitehttps://uk.wikipedia.org – Access mode: https://uk.wikipedia.org/wiki/Arduino

2. Servo drives [Electronic resource] // Website princeton.edu – Access mode: http://www.princeton.edu/~mae412/TEXT/NTRAK2002/292-302.pdf

3. Robotics: servos [Electronic resource] // Website https: // http://wiki.amperka.ru – Access mode: http://wiki.amperka.ru/ Robotics: servos

4. Wikipedia [Electronic resource] //Sitehttps://uk.wikipedia.org – Access mode: https://uk.wikipedia.org/wiki/ESP8266

5. Protocol of limited application [Electronic resource] // Website https://en.wikipedia.org – Access mode: https://en.wikipedia.org/wiki/Constrained_Application_Protocol

6. Arduino [Electronic resource] //Sitehttps://uk.wikipedia.org – Access mode: https://uk.wikipedia.org/wiki/Arduino

7. What is MQTT and why is it needed in IoT? Description of the MQTT protocol [Electronic resource] // Website https://ipc2u.ru/ – Access mode: https://ipc2u.ru/articles/prostye-resheniya/chto-takoe-mqtt/

8. Safety at work on the personal computer [Electronic resource] // Website https://studopedia.info – Access mode: https://studopedia.info/3-51426.html

9. Safety precautions when using electrical appliances [Electronic resource] // Website https://buklib.net – Access mode: https://buklib.net/books/30658/

10. Computer in the office and its environmental safety [Electronic resource] // Website http://ur.co.ua – Access mode: http://ur.co.ua/112/219-1-komp-yuter -v-ofise-i-ego-ekologicheskaya-bezopasnost.html

11. Energy [Electronic resource] // Website http://www.info-library.com.ua – Access mode: http://www.info-library.com.ua/books-text-2694.html