UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF MECHANICAL ENGINEERING

MANIPULATOR ROBOTIC ARM WITH SEVEN DEGRES OF FREEDOM FOR ARDUINO BOARD
ΡΟΜΠΟΤΙΚΟΣ ΒΡΑΧΙΩΝΑΣ ΧΕΙΡΙΣΤΗ ΓΙΑ ΤΗΝ ΠΛΑΤΦΟΡΜΑ ARDUINO

ALEXIOS MENEGAKIS

Submitted in partial fulfillment of the requirements for the degree of MSc in Supply Chain Management and Logistics in the Department of Mechanical Engineering at the University of Thessaly

Volos, 2024

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

MANIPULATOR ROBOTIC ARM WITH SEVEN DEGRES OF FREEDOM FOR ARDUINO BOARD

ALEXIOS MENEGAKIS

Submitted in partial fulfillment of the requirements for the degree of MSc in Supply Chain Management and Logistics in the Department of Mechanical Engineering at the
University of Thessaly

Volos 2024

Approved by the Committee on Final Examination:

Advisor	Dr. Ampountolas Konstantinos, Associate Professor, Department of Mechanical Engineering, University of Thessaly
Member	Dr. Georgios Saharidis, Associate Professor, Department of Mechanical Engineering, University of Thessaly
Member	Dr. Koukoumialos Stylianos Professor, Department of Business Administration, University of Thessaly

Date Approved: [September 25, 2024]

Acknowledgements

This thesis is devoted to my professors in the Department of Mechanical Engineering in University of Thessaly at Volos. Their guidance and their support enabled me to reach a concept and carry out this small task. Surely, their contribution was very crucial in shaping my professional career and inspiring me to continue with future developments.

MANIPULATOR ROBOTIC ARM WITH SEVEN DEGRES OF FREEDOM FOR ARDUINO BOARD

ALEXIOS MENEGAKIS

Department of Mechanical Engineering, University of Thessaly

Supervisor: Dr Konstantinos Ampountolas
Associate Professor of Operational Research

Abstract

The following paper presents the assembly and programming of a robotic arm manipulator of seven (7) degrees of freedom. The parts are customized, and the installation is based on drawings. Any modifications are due to the request for usage.

The manipulator will be programmed to carry out small tasks such as movements and when a pump is supplied to the main installation it will create vacuum that with a 24-voltage actuator will suck any object like a yoghurt cup or food and beverage packages.

The microcontroller of the installation is the Arduino Uno R3 board, and the motors used are controlled by the Sensor shield V5.0.

In the following image there is the final assembly.

The case study of controlling the manipulator will be represented in special Appendix

Key words: manipulator, Arduino, servo motor

MANIPULATOR ROBOTIC ARM WITH SEVEN DEGRES OF FREEDOM FOR ARDUINO BOARD

ALEXIOS MENEGAKIS

Department of Mechanical Engineering, University of Thessaly, 2024

Supervisor: Dr Konstantinos Ampountolas
Associate Professor of Operational Research

Contents

List of Figures

List of tables

Chapter 1. INTRODUCTION

In this chapter I represent my motivation to begin and assembly the robotic manipulator and the importance of this installation. At the next chapter I present some fundamental theory in robotics and motion. A separate chapter is dedicated to the electronic compartments of this robot. Finally, the thesis specializes in our equipment and does mention in the initialization of servo and motion control of the joints. I also include photos of the different phases of installation and final assembly. Not to forget, the special appendix includes a simple application of this robotic manipulator.

1.1 Motivation

The motivation arises from my workplace packaging equipment, where robot manipulators are used to carry out tasks like picking yoghurt cases and putting them into trays. Further, a bigger robotic arm takes the trays from the end of the conveyor.

1.2 Introduction to literature review

This installation is an assembly of parts that can be manufactured in machinery. There are some ads on like the vacuum pump or the silicone elastic hose. Also, tools greasing and modifications during the assembly were made by me.

Similar manipulators can be found easily with quick search because of the variety of applications and the widespread operation. To this it’s simple in mechanical equipment because each joint makes a movement through the servo motor and not a single axis, belts and gearing required to transmit motion. Also, controlling via a shield is very efficient since it does not need a rack or any space to fit. And one other important aspect is that its cost is effective and relatively cheap.

1.3 Thesis structure

The next of this thesis is separated into four other chapters.

More specifically, in chapter 2 we present some basic knowledge in robot kinematics. We define the parameters in the motion of a robotic arm and then we describe algorithm to find each one of them.

In chapter 3, we examine the electronic part of the assembly. Schematic electrical diagrams and short explanation of the Arduino R3 board used are also described.

In chapter 4, we present the initialization step a servo motor. It's driven by the shield.

In chapter 5, there is a part list with the tools and materials used. Also, some photos of the installation are given.

In chapter 6, there are the results of the study and some comments for future innovations.

Finally, there is an Appendix with a simple example of motion and carrying out small tasks.

In the photo below there is a photo with the final assembly.

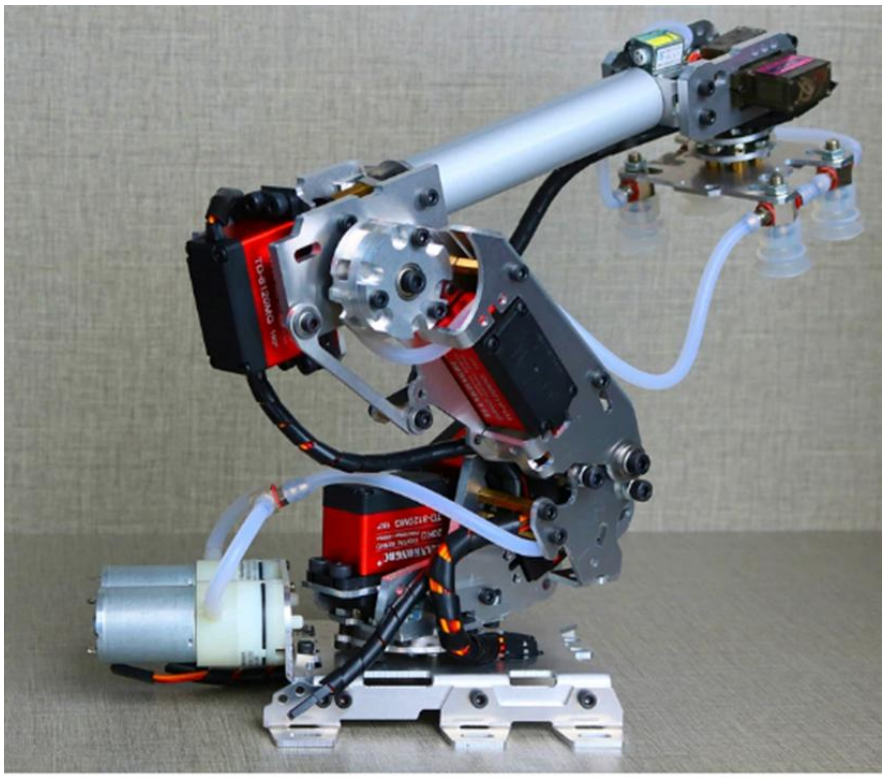


Figure 1: The 7 DOF manipulator after assembly

Chapter 2: ROBOT KINEMATICS

By means of kinematics we refer to the study of the motions in a manipulator without considering the forces that cause it. Kinematics is a property of the motion of an object.

2.1 What do we study?

- The Position, speed, acceleration and all derivatives of variables describing position (function of time or other variables).

- Robotic arms consist of ligaments/limbs (they are rigid elements), connected to each other by joints.

2.2 What is a robot manipulator?

A robot manipulator is an assembly of links that are connected at points called joints.

Joints

- Rotary or revolute: They are the ones that move rotary. Their movement is measured based on the angle (angle of joint) that form the moving members. Allows a relative rotation around a single axis.
  One degree of freedom: angle of rotation

- Prismatic: They are sliding joints. The measured difference in the position of the two states in this case is called displacement. Allows a linear motion along a single axis.

  One degree of freedom: amount of linear displacement

- Ball joint: spherical bearing

Figure 2: Two degrees of freedom spherical bearing. Ανακτήθηκε από τον ιστότοπο: https://www.timken.com/

- Socket joint:

  Two degrees of freedom

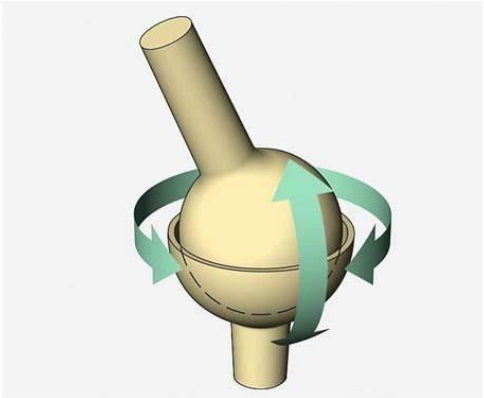


Figure 3: socket joint.
Ανακτήθηκε από τον ιστότοπο https://www.pinterest.com/

- Spherical wrist: Three degrees of freedom

Figure 4: Uniball spherical joint. Ανακτήθηκε από τον ιστότοπο https://www.desertcart.ae/

2.3 Analysis of Kinematic problem

For our analysis, we assume that each joint has one degree of freedom and the more one degree of freedom joints can be assuming as separate joints of one degree of freedom that their links have zero length. By this assumption, each joint behavior can be represented by a single real number.

In a rotational joint this real number is the angle of rotation θ_i. In a prismatic joint is the displacement.

A robot manipulator has n joints and n+1 links since from two points (joints) passes a straight line. So, in our analysis we will number 1-n for joints and 0-n for links, so that joint connects link i-1 to i. We also assume that joint i is fixed with respect to link i-1. When I joint is actuated, link i moves. Therefore link 0, which is the first link is fixed and does not move when the other joint does.

Each link must be referred to within a coordinate system. We assume that x_i, y_i, z_i system is attached to link i. By doing that, we assure that at every motion the robot executes the coordinates of each point on link are constant when expressed referring to the i^{th} coordinate frame. The coordinate system which is attached to the base of the robot is called inertial frame.

Degrees of Freedom

The number of degrees of freedom of a robotic arm is the number of independent position variables, which must be defined so that the position of all its constituent members can be determined. The number of degrees of freedom of arm equals the number of arm joints.

Forward and inverse kinematic problems

The two main categories can be distinguished in kinematics, direct kinematics and inverse kinematics.

Forward Kinematic Problem

We determine the position and orientation of the end effector, given the variables of the ligaments (joints). So, we should have the joint parameters.

The variables of the joints are the angles between the ligaments in the case of rotary joints and the elongation of the joint in the case of prismatic joints.

Inverse Kinematic Problem

We determine the variables of the joints given the position and the orientation of the end effector.

It is a more complex problem but used a lot in determining the motion of robot in industrial applications.

Parametric values
Both direct (forward) kinematics and inverse (backward) kinematics need to be parametrized with the same parametric values

2.4 Denavit-Hartenberg Algorithm

Denavit–Hartenberg (DH) parameters is a convention invented by robotics experts to mathematically describe the motion of a robot with only 4 parameters.

Step 1

Enumerate the joints and identify their parameters.

Step 2

Put axis z_i.

- Axis z_0 is in the basis at the direction of movement of the direction of the first joint.
- Axis zi is in the movement of the direction in the i+1 joint
- If joint is rotational, the direction of the movement is the axis of rotation.

Step 3

Put axis x_i

- Axis xi is in the common  between z_{i-1}, z_i with direction from z_{i-1} to z_i.
- If axis z_{i-1}, zi are crossed when we transfer the origin of coordinates system at that point.
- If axis z_{i-1}, zi are parallel, x_i is located explicitly with direction from z_{i-1} to z_i.

Step 4

We the parameters in D-H matrix according to the following rules.

θ_i is the angle of rotation of x_{i-1} related to the axis z_{i-1}, with tend to be parallel and at same direction with x_i

r_i is the minimum (perpendicular) distance that x_{i-1} needs to cross relative to z_{i-1} to meet x_i

d_i is the minimum (perpendicular) distance that z_{i-1} needs to cover with respect to x_i to meet z_i. For intersection axis z the prismatic joint is d_i=0

α_i is the angle of rotation of z_{i-1} with respect to xi in order to be parallel and at the same direction with z_i

Notations

Rotations are considered with the right finger rule
If joints are rotational. The parameter of joint should appear. r_i then is 0 or constant distance
If joints are prismatic, joint parameters should be appeared. θ_i is 0 or constant angle.
For intersection axis z_{i-1} related to x_i, to become parallel and at the same direction with z_i

Chapter 3. ELECTRONIC COMPONENTS OF THE INSTALLATION

3.1 Arduino Sensor Shield V5.0 sensor expansion board

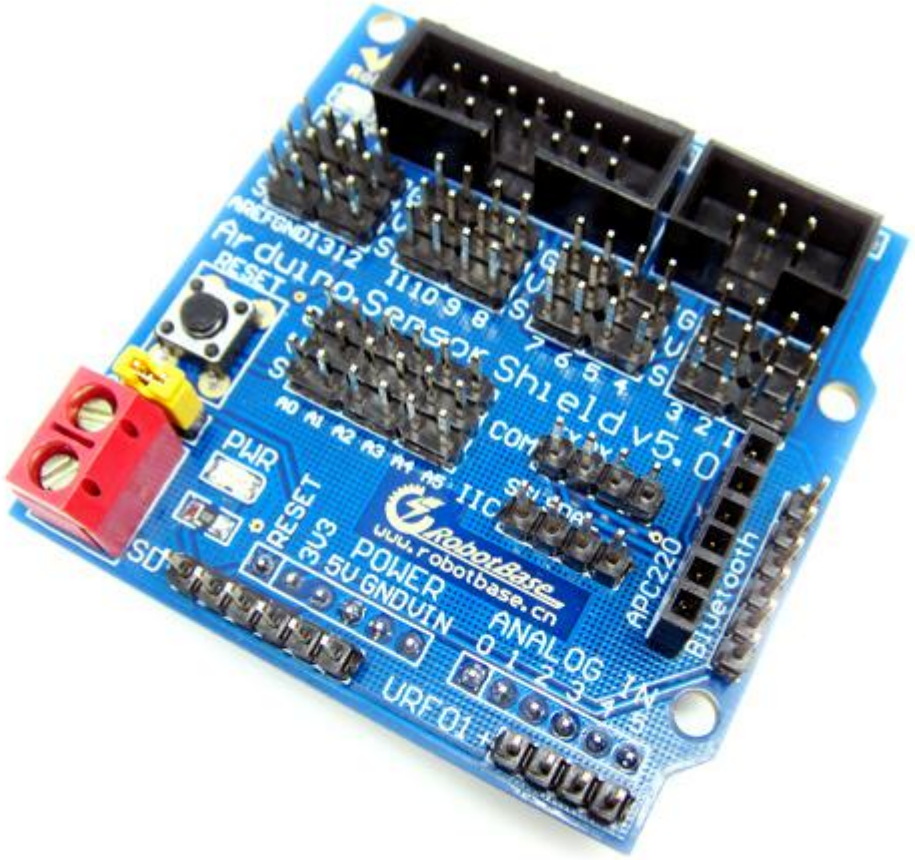
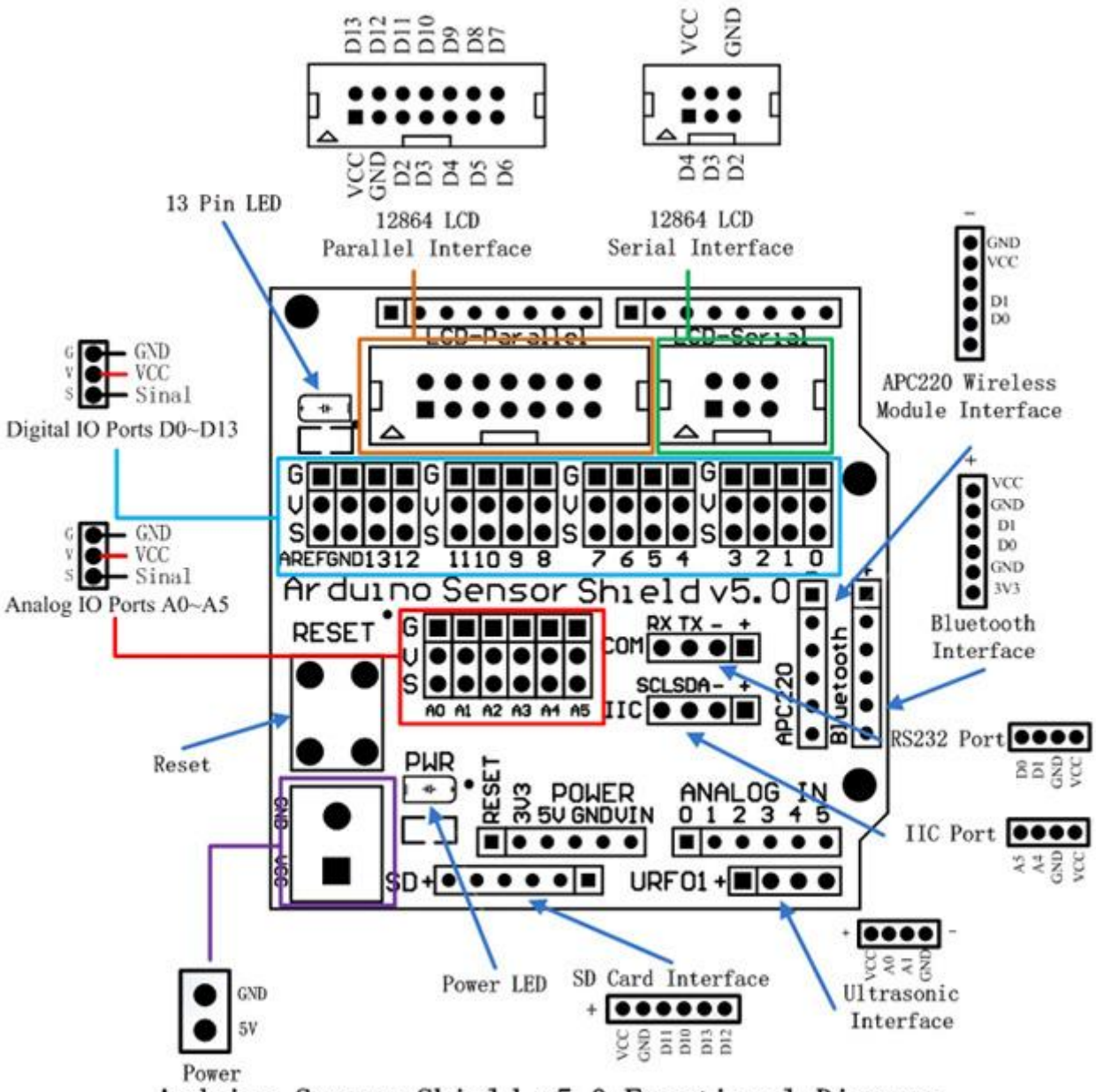


Figure 5: The Arduino Shield that is attached to the main board Arduino Uno R3.

Arduino Sensor Shield is compatible with the Arduino Uno R3. We connect to its pins servos, sensors and LCD. This board is connected with the Arduino Board using the jumper wires.

The latest Arduino Sensor Shield V5.0 sensor expansion board introduced in 2010 still adopts the laminated design and PCB immersion gold process based on retaining the advantages of the V4.0 version. The main board not only combines all the digital data of the Arduino Duemilanove 2009 controller with The analog interface is extended in the form of servo line sequence, and it is also equipped with IIC interface, 32-way servo controller interface, Bluetooth module communication interface, SD card module communication interface, APC220 wireless radio frequency module communication interface, RB URF v1.1 ultrasonic sensor Interface, 12864 LCD serial and parallel interface, independent expansion is easier to use.

This sensor expansion board really simplifies the circuit and can easily connect common sensors. A sensor only needs a universal 3P sensor connection line (regardless of digital connection line and analog connection line). After completing the circuit connection, we can write the corresponding Arduino program in IDE platform and download it to the Arduino Duemilanove controller to read the sensor data or receive the data back from the wireless module.



Arduino Sensor Shield v5.0 Functional Diagram

Figure 6: The Arduino Sensor Shield v5.0 and Functional Diagram

3.2 Technical Specifications of Arduino v5.0 and explanation of electronic diagram

Pins in this Arduino Shield are digital and analog.

Digital pins are in packs of 3. The top pin is the GND (0 Volt or -V), the middle is the Vcc which is the +V and the third is the signal.

G			GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND
V			Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
S	Aref	GND	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 1: Pinout of the digital pins in Arduino Shield

Analog pins are shown in Table 2, G is the letter used for Ground, Vcc is the V+ pin used to supply voltage and S is the signal of the sensors attached.

G	GND	GND	GND	GND	GND	GND
V	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
S	A0	A1	A2	A3	A4	A5

Table 2: Pinout of the Analog pins in the Arduino Shield

A binary signal takes only two states which are “Voltage present +24 V” and “Voltage not present 0 V”.

Analog signals can take any value between a defined range. An example of an analog sensor is a potentiometer. We adjust the position of the knob, so the resistance varies accordingly up to the max value.

In automation other examples of analog quantities are

- Temperature from -50 to +150 °C measured with temperature transducer
- The Flow rate in a pipe measured with a flow meter with range from 0 to 200 l/min
- Speed of a motor -500 to +50 rpm measured with encoder

3.3 Measuring transducers

The above quantities (temperature, flow rate, speed etc.) are converted to electrical proportional signals such as voltages, currents or resistances with the help of a measuring transducer. To measure speed for example, a range from 500 to 1500rpm will be converted into a voltage range from 0 to +10V with the use of measuring transducer. For instance, at a measured speed of 865 rpm, the measuring transducer would output a voltage value of +3.65 V.

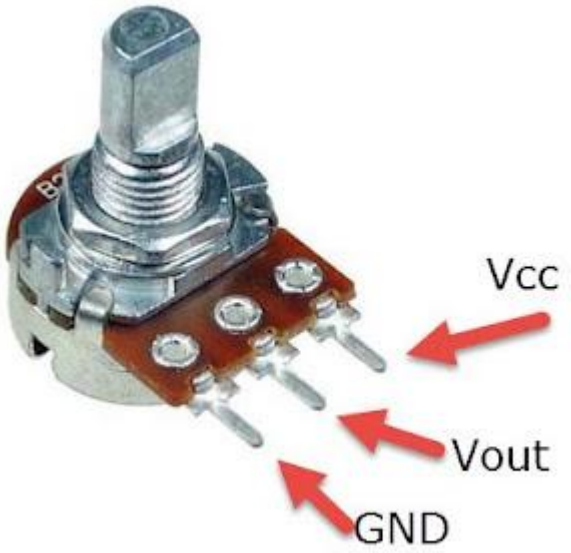


Figure 7: A typical potentiometer. With Vcc we denote the power supply voltage, with Vout the output voltage that ranges according to our regulation and with GND the ground pin.

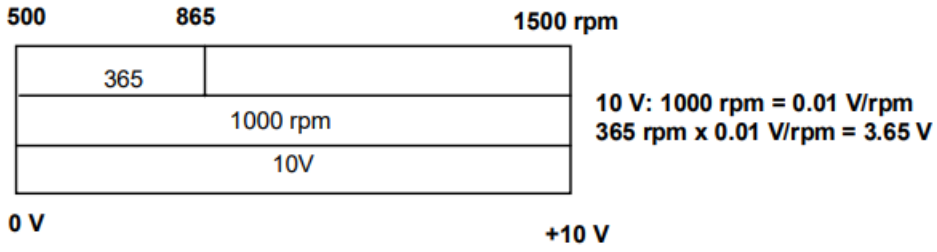


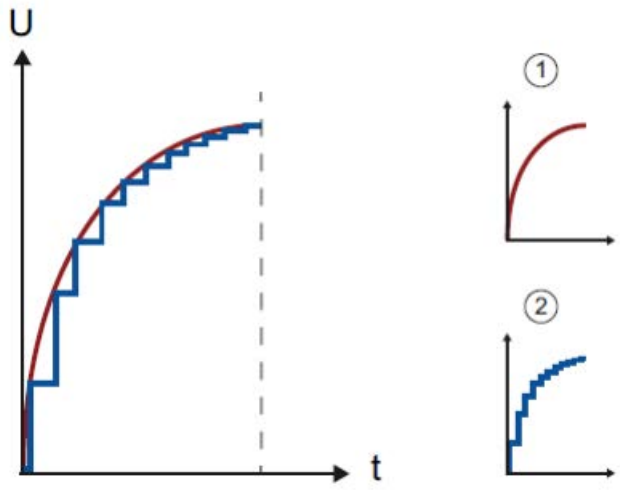
Figure 8: Analogue to digital scaling of velocity in a system

3.4 Analog modules – A/D converter

These electrical voltages, currents or resistances that are converted to analog signals, are connected to an analog module that digitizes this signal for further processing in the PLC. The digital information in the PLC is converted to a bit pattern. The conversion of the digital information to bit pattern is called analog-to-digital conversion (A/D conversion).

In our example the 3.65 Volt will be stored in PLC as a series of binary digits.

The analog input card where all the signals terminate, has embedded the ADC (analog-to-digital converter) which digitizes the analog signal being acquired and approximates its value in the form of a stepped curve. Basic parameters of this curve which is characteristic of the ADC are the resolution and the conversion rate.



1: Analog value

2. Digital value

Figure 9: Analog to digital conversion in Analog inputs Card

The resolution depends on the number of binary digits in the digitalization process. More binary digits mean better resolution in the analog input card.

For instance, if we had 1 bit for the representation of analog to digital for a voltage range 0 to +10 Volts, we would only know two if we are in the range 0 to +5 Volts with bit 0 or in the range +5 to +10 Volts with bit 1. Accordingly, if we used 2 bits we could separate the space in 4 different situations. 00 bit for the range 0 to 2.5 Volts, 01 bits for the range 2.5 to 5 Volts, 10 bits for the range 5 to 7,5 Volts and 11 for the range 7.5 to 10 Volts

Conventional A/D converters in control engineering use 8 bits, 11 bits or more for converting. With 8 bits you have 256 individual ranges, while 11 bits provide a resolution of 2048 individual ranges

RS232 Port

Serial communication

Serial data transmission

During serial data transmission, the individual bits of a character of information to be transmitted are sent successively in a defined sequence.

Bidirectional data traffic - operating mode

In the context of bidirectional data traffic, we distinguish between two operating modes for the communication module:

• Half-duplex operation

The data is exchanged between the communication partners in both directions alternately. In half-duplex operation one communication partner is sending and the other communication partner is receiving at the same time. In the process one line is alternately used for sending or receiving.

• Full-duplex operation

The data is exchanged between one or more communication partners in both directions simultaneously, which means you can send and receive data at the same time. This process requires one line for sending and one line for receiving.

Data transmission

The so-called time base synchronism (a fixed timing code used in the transmission of a fixed character string) is only upheld during transmission of a character. Each character to be sent is preceded by a synchronization impulse, also called start bit. The length of the start-bit transmission determines the clock pulse. The end of character transmission is formed by one or two stop bits.

In addition to start and stop bits, additional declarations must be made between the sending and receiving partners before serial transmission can take place. These include:

• Data transmission rate

• Frame start and end criteria (e.g., character delay time)

• Parity

• Number of data bits (7 or 8 bits/characters)

• Number of stop bits (1 or 2)

RS232 mode uses two lines for data transmission, one line for the send direction and one for the receive direction. The communication mode is full duplex enabling simultaneous send and receive.

RS232 signals

TXD: Output →Transmitted data, interface is transmitting

RXD: Input→Received data, interface is receiving

GND: Functional ground, isolated

DCD: Input→Data carrier detect, carrier signal when connecting a modem. The communication partner signals that it recognizes the incoming data

DTR: Output→ Data terminal ready,

DTR set to "ON": Communication module switched on, ready for operation DTR set to "OFF": Communication module not switched on, not ready for operation

DSR: Input→ Data set ready

DSR set to "ON": Communication partner signals readiness for operation DSR set to "OFF": Communication partner not switched on, not ready for operation

RTS: Output→ Request to send

RTS set to "ON": Communication module ready to send, signals to the communication partner that there is data ready to send
RTS set to "OFF": Communication module not ready to send

CTS: Input→ Clear to send; Communication partner can receive data from the communication module (response to RTS = ON of the communication module)

CTS set to "ON": Signals readiness to receive to the communication partner
CTS set to "OFF": Signals "Not ready to receive" to the communication partner

RI: Input→ Incoming call for connecting a modem (ring indicator)

After power on the communications module, the output signals are in the OFF state (inactive).

You configure the operation of the DTR/DSR and RTS/CTS control signals in the user interface of the communications module.

A simple possibility of data exchange is provided by serial communication via point-to-point connections.

RS232 is an interface of communication between partners through additional accompanying signals.

Handshaking procedure

Handshaking controls the data flow between two communication partners. The use of the handshaking method prevents data loss during transmission if the devices are operating at different speeds.

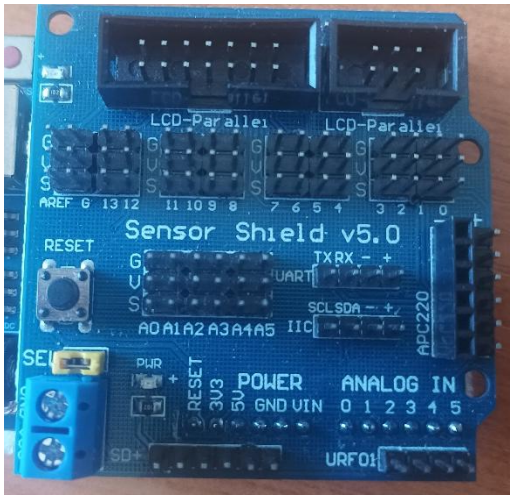


Figure 10: The Arduino Sensor Shield v5.0 we used in installation

APC220 Wireless module interface

APC220-43 is highly integrated semi-duplex low power transceiver module with high speed MCU and high capability RF IC。Using high efficiency forward error correction with Interleaving encoding technology, it can make anti-interference and sensitivity improved highly. It can have a good performance in strong interference circumstance as well, for example the industry field。The technique is advanced in data transfers area.

APC220-43 is a cost-effective and easy applied module that not only can transmit transparent data with large data buffer zone, but also can provide more than 100 channels. Its parameters easily setting and small size make the module an ideal for wireless data transfer application.

Arduino UNO R3

Power

Arduino Sensor Shield is compatible with the Arduino Uno R3. We connect to its pins servos, sensors and LCD. This board is connected with the Arduino Board using the jumper wires.

1. **13 Pin led**

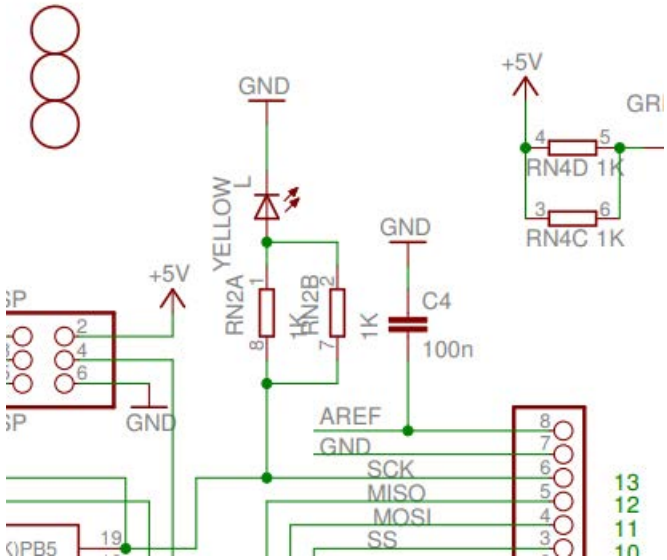


Figure 11: Explanation of diagram

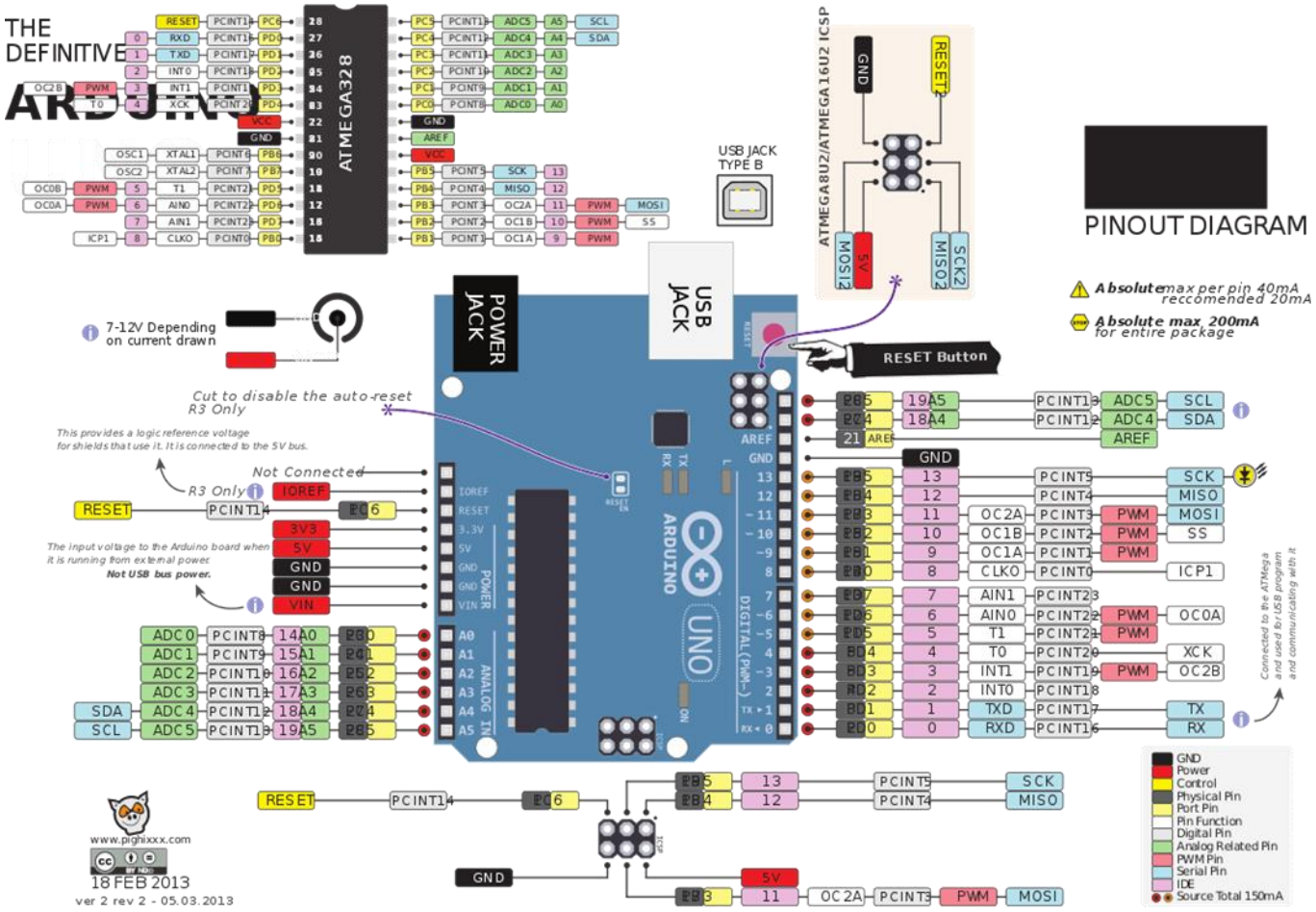


Figure 12: The Arduino R3 pinout with functions. Ανακτήθηκε από τον ιστότοπο https://arduino.stackexchange.com/

The Arduino Uno Board has two power sources as seen in the drawings. The USB power and the PWRIN power line.

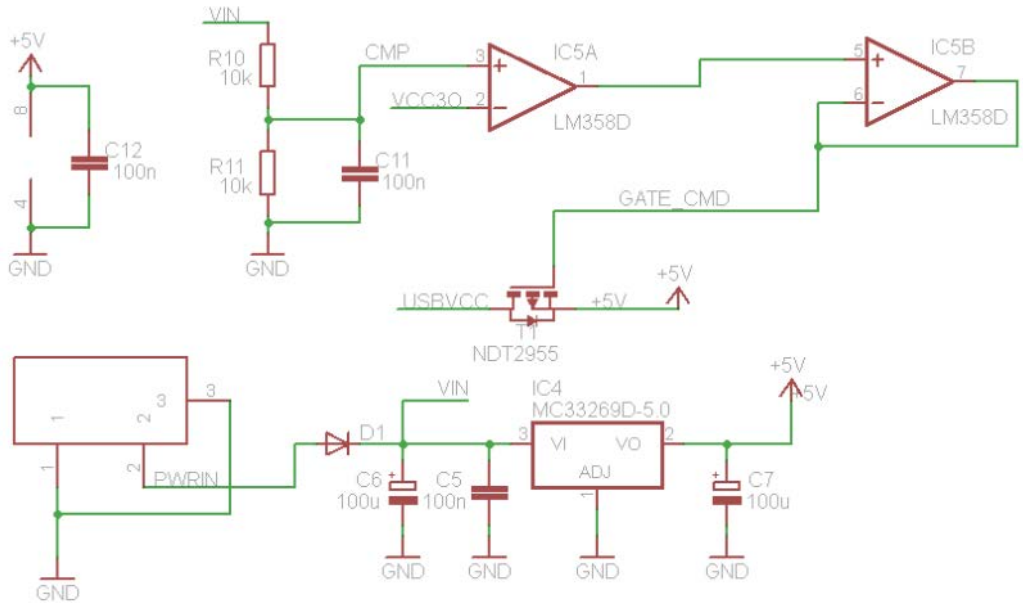


Figure 12: Explanation of diagram

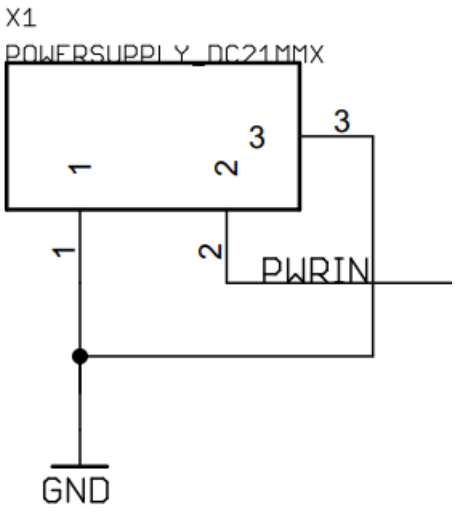


Figure 13: Explanation of diagram

This is a coaxial power connector with 2.1mm socket. With this barrel-shaped component we provide Arduino with an external power supply. Leg 2 which is the back pin is the V+ (PWRIN). The leg 1 is the front pin and it's the barrier and the side which is the pin 3 is connected to 1 as well presenting the status "connected". So, if we put a multimeter in 2 and 3, we can measure the power supply voltage to the Arduino board from external power other than USB.

Figure 14: D1M7 diode in diagram

Then the PWRIN passes through the D1M7 diode.

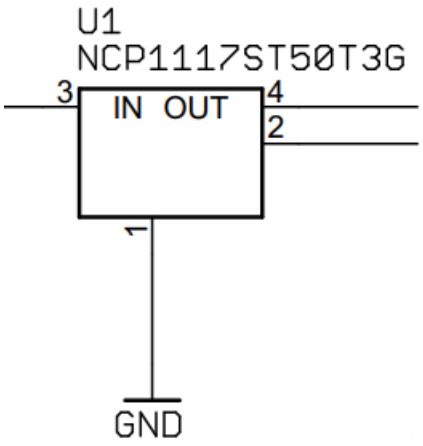


Figure 15: NCP1117 voltage regulator

This is NCP1117. It's a low dropout positive voltage regulator and is capable to provide an output current that is more than 1.0A with a maximum dropout voltage of 1.2 V at 800 mA overtemperature. By the term voltage regulator, we assume a component that gives a stable output voltage, no matter if the input voltage changes. Low dropout regulator is a type of DC linear voltage regulator circuit that can operate even when the input voltage is very close to the output voltage. On chip trimming, a process that is carried out after chip manufacture, can enclose the reference to output voltage within ±1.0% accuracy. The dropout voltage of a regulator is the minimum required to maintain +5Volt in output.

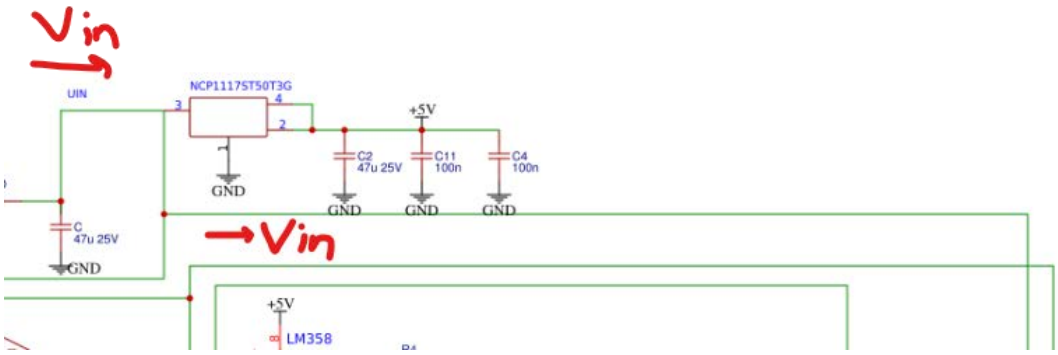


Figure 16: Explanation of diagram part

The one clone of Vin of the PS goes to the 7 I/O connectors in the Arduino Board. These are shown in the photo.

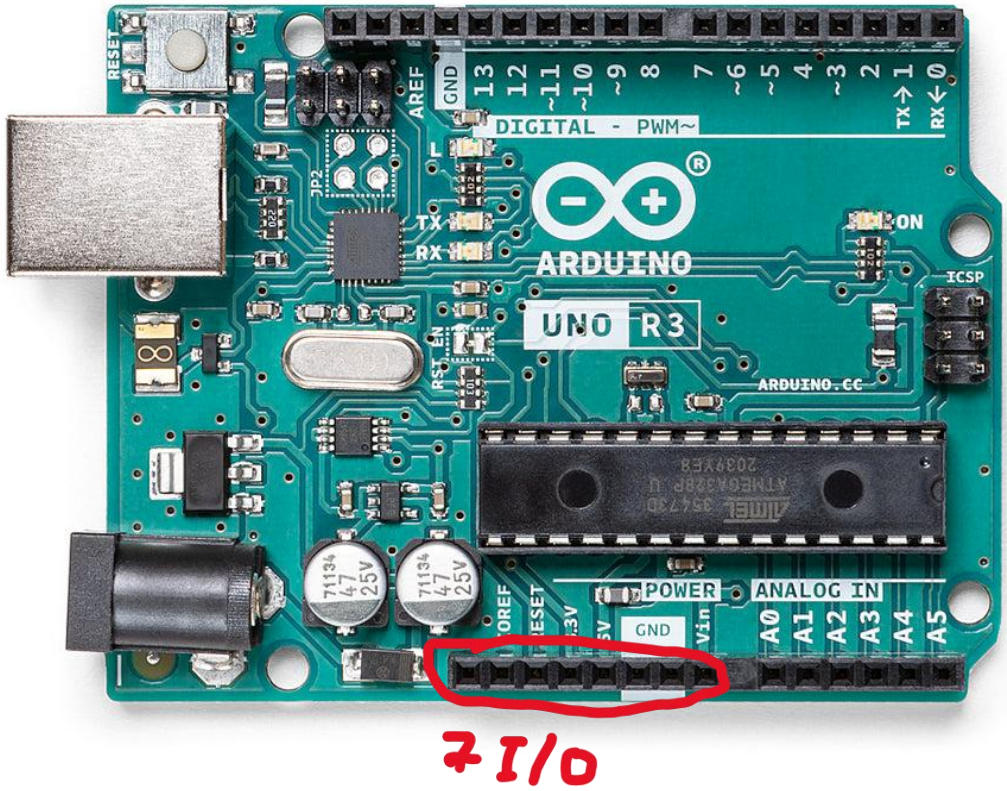


Figure 17: The Arduino Uno R3 Board. Ανακτήθηκε από τον ιστότοπο https://store.arduino.cc/

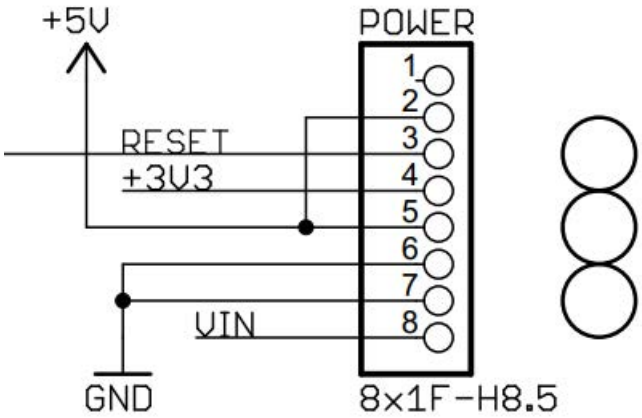


Figure 18: Explanation of diagram part

1→not used

2→+5Volt (IOREF). Used to show the reference voltage which the Arduino operates

3→RESET

4→3.3 Volt

5→+5Volt

6→GND

7→GND

8→+Vin it's the voltage before the regulator

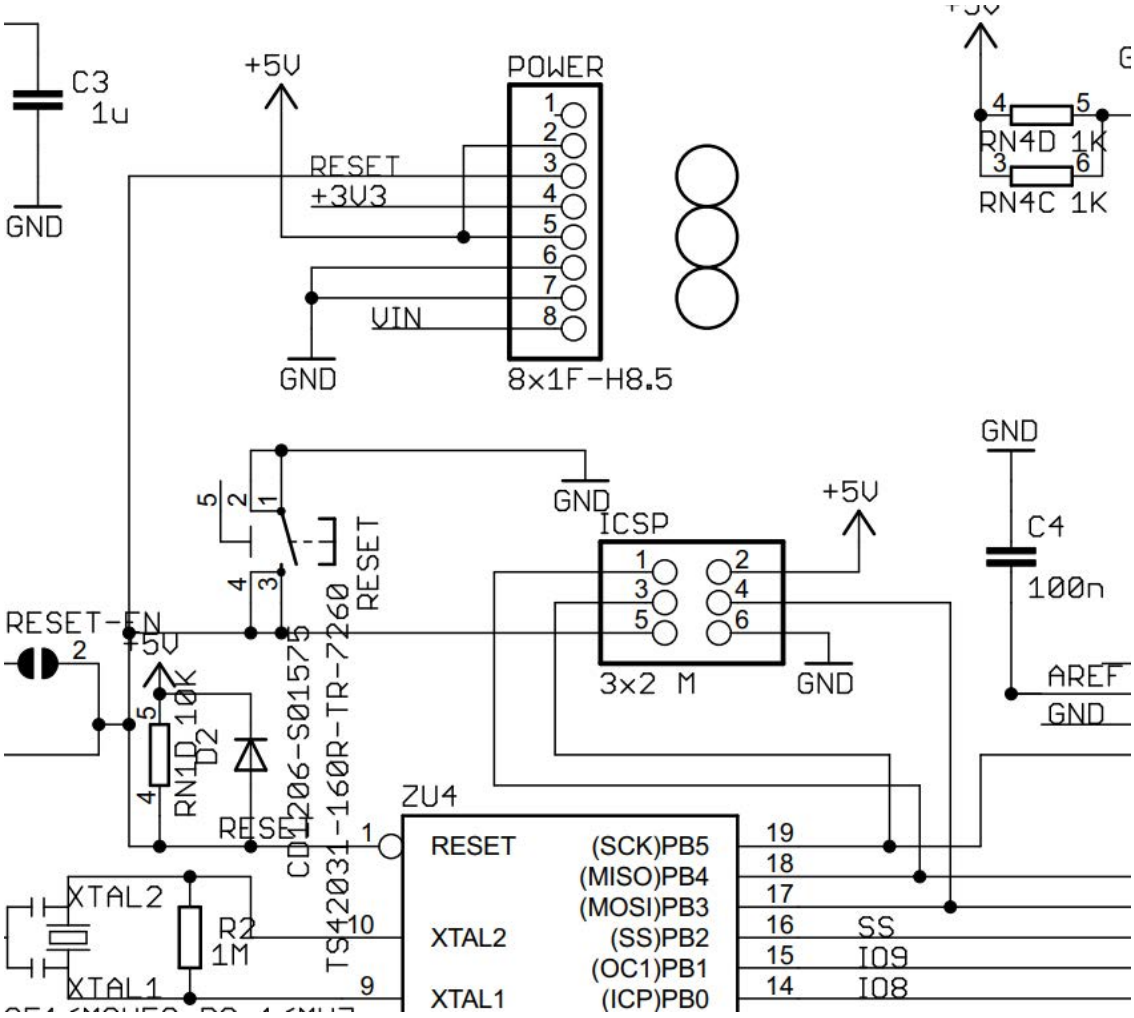


Figure 19: Explanation of diagram part

ICSP (In Circuit Serial Programming)

In-System Programming (ISP) technique requires that a device to be connected to PCB via a communication port. The In-Circuit Serial Programming (ICSP) that we use in our project is an ISP technique developed by Microchip's PICmicro® One-Time-Programmable (OTP) and FLASH RISC microcontrollers (MCU). In ICSP we use two I/O pins to input and output data.

Below there are abbreviations and meaning of the 6 ICSP pins.

1→MISO (Master In Slave Out, Data going from the master to the slave)

2→+5 Volt Vcc

3→SCK is the system clock

4→MOSI (Master Out Slave In, data going from the slave to the master)

5→RESET

6→GND

When we power the Arduino board through the DC jack, Dc voltage applies to the MC33269 regulator, and +5 Volt are generated. Then the 5 Voltage output goes to R10/R11 divider as VIN and to the + terminal of the comparator. The 3.3 Volt output of the FT232RL is fed to the – terminal. The differential is positive, and we characterize the output as high (+ 5Volt). The LM358D is used to decide whether the board will be supplied from USB or VCC30

In the upper drawing we see the Vin source. Then we have two resistances R10 and R11 of 10 KΩ each. VIN should be above VCC30 after being divided by the resistances R10 and R11. From the names, we can guess that it would be 6V, which could be the minimum IC4 requires to make reliable 5V out (I don't recognize the IC4 part number and didn't check). You are right, there is no purpose to IC5B. It is a unity gain buffer, but the output of IC5A should have the same impedance and drive capability.

Chapter 4. SERVO AND MOTION CONTROL

4.1 Test servos before installation

Before installation, please test whether the servo is operating normally and adjust the servo angle (as shown in Figure 1.5ms). That happens because after the installation we need the initialization procedure known as homing in automation. We set the mechanical position that the machine, the robot in our case, begins to measure the rotations for the programmed movements.

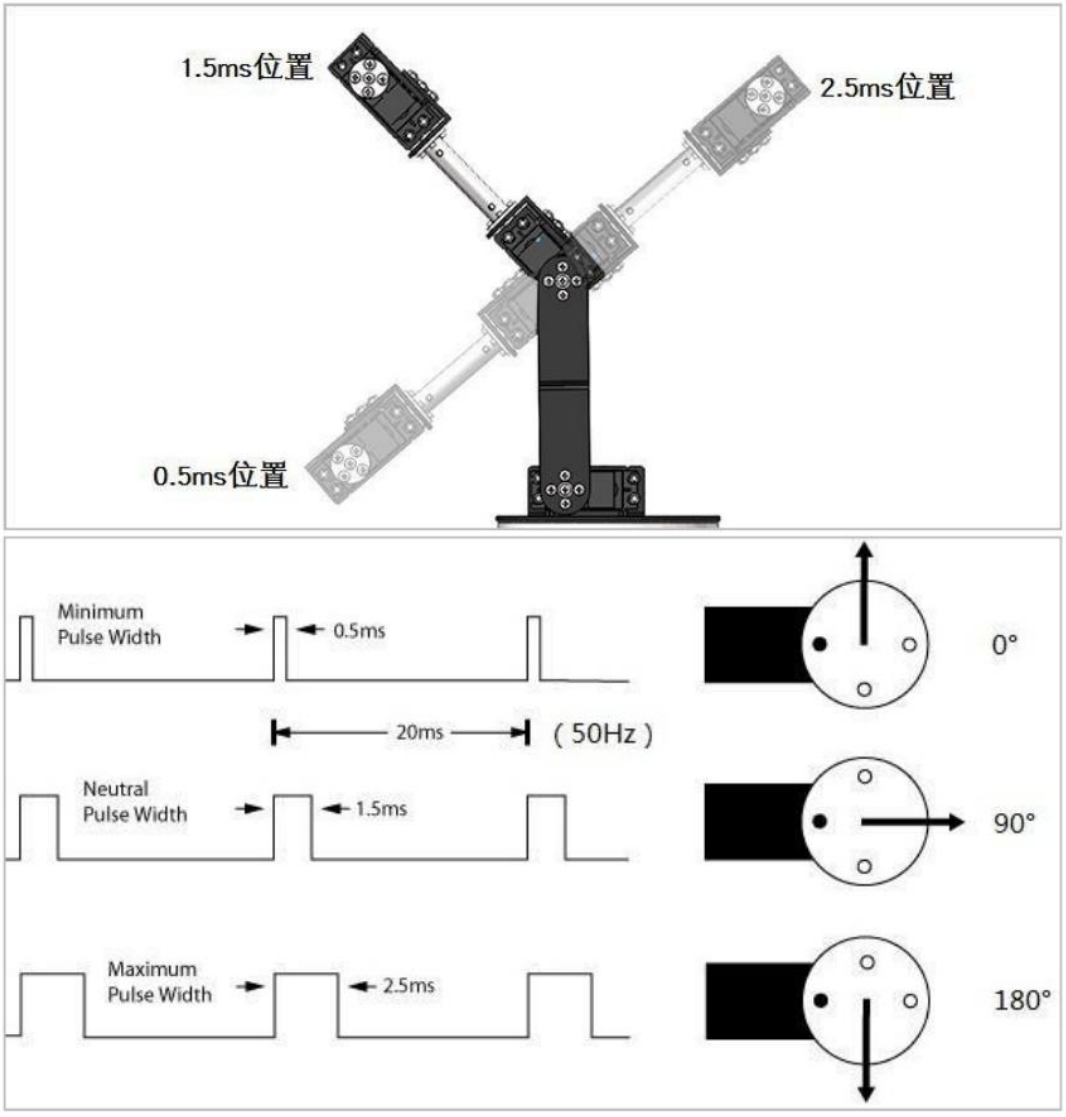


Figure 20: Set up the homing degrees of position in our servomotors

The 1.5ms (90°) position in the picture is the midpoint of the servo we need to find (no need to be particularly precise). For a 180-degree servo, it is the 90-degree position, so 0 degrees corresponds to the 0.5ms position, 180 degrees. The corresponding

position is 2.5ms; in this way, the installed robotic arm works exactly within the range of motion we need.

Therefore, before installation, it is very important to determine the midpoint of the servo (or 0-degree point or 180-degree point). There are usually two types of servos on the market, one with internal limit points: the other without internal limit points. Usually those without limits are digital servos.

The best way to judge is to install the rudder and turn it by hand. If it cannot turn after a certain angle, it means it has an internal limiter; if it can keep turning, it means it has no internal limiter.

For a servo with a limiter, it is at the 0-degree position when it cannot rotate counterclockwise, and it is at a 180-degree position when it cannot rotate clockwise; for a servo without a limiter, it must be powered on and tested to find the midpoint position of 1.5ms. You can write your own code for testing or use a specialized servo controller to implement it.

The controller can accurately set the initial angle of the servo to an initial value of 90 degrees (1.5ms), ensuring that subsequent servo operations will not be damaged due to the jamming of the servo due to the bracket limit.

The installation of the steering gear in any degree of freedom must follow the above process, otherwise unexpected failures will occur.

Chapter 5: INSTALLATION

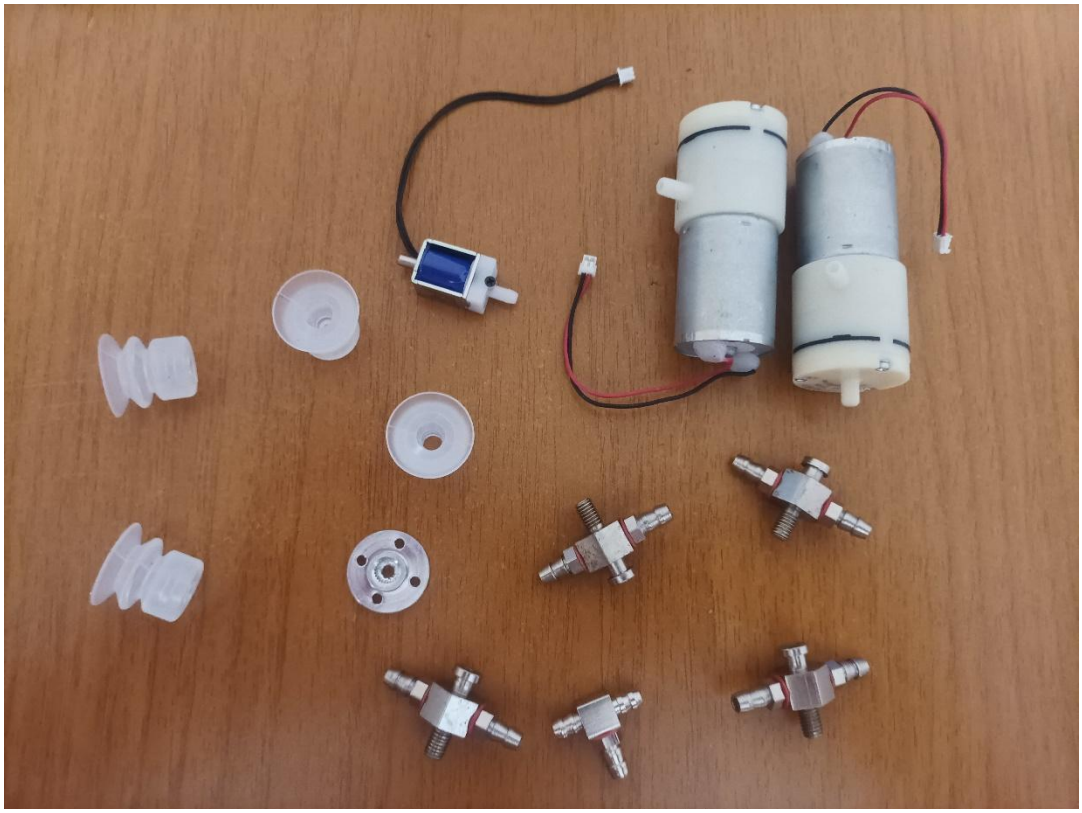
In this chapter we present photos of our installation during assembly.

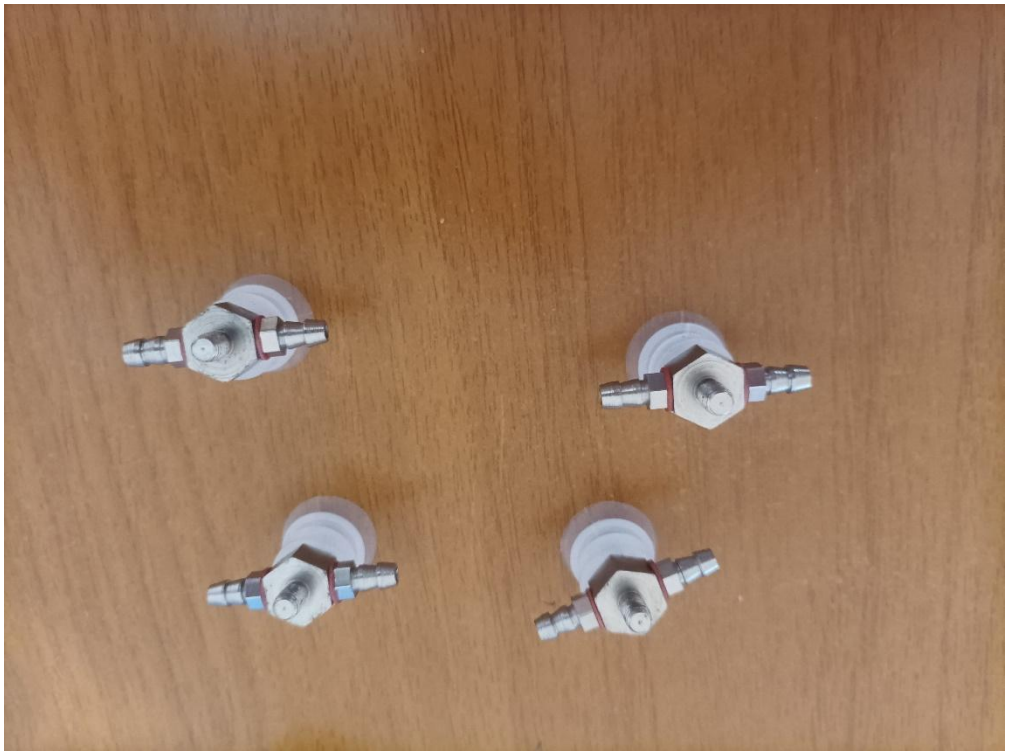
5.1 Materials used

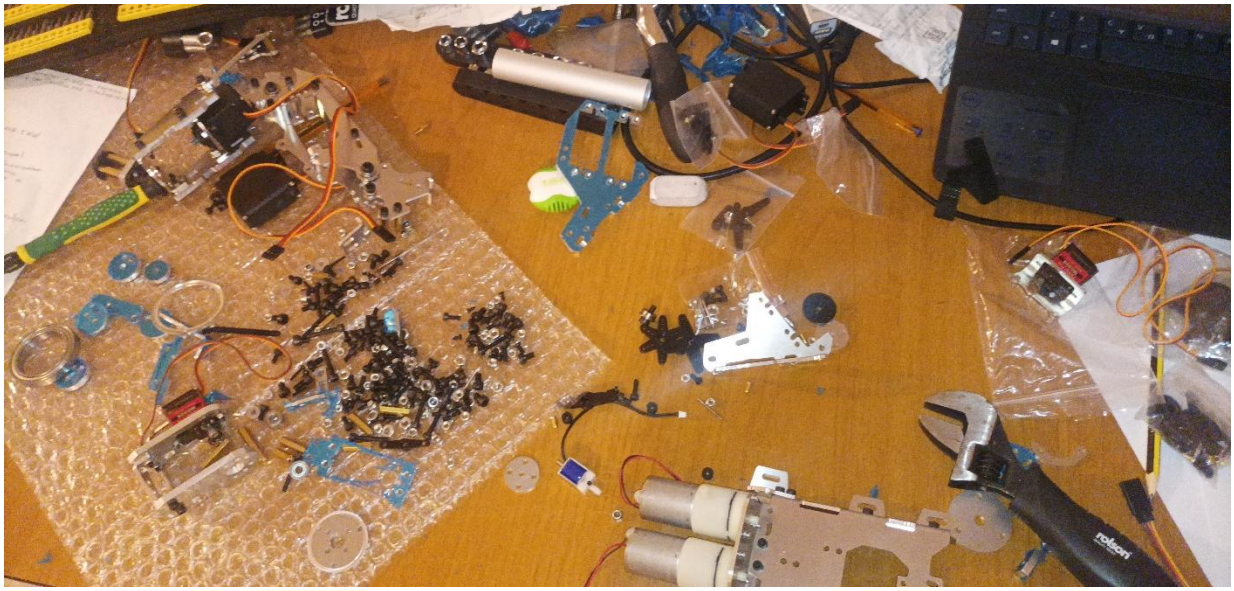
The materials used were

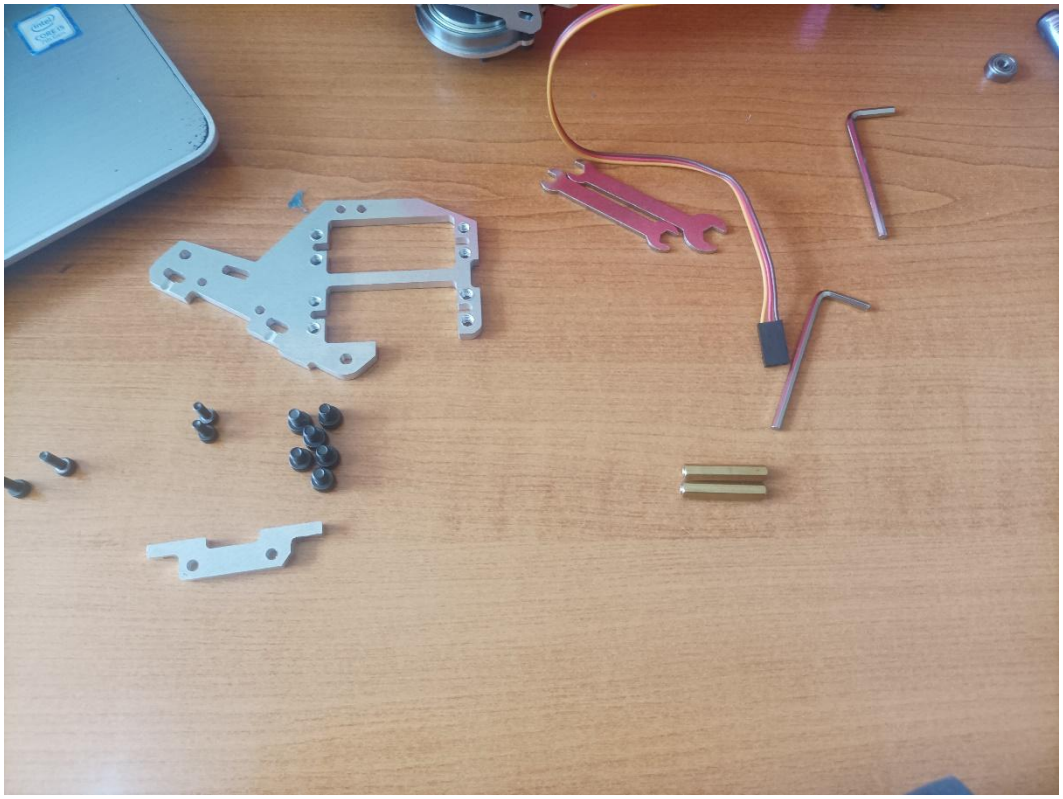
Bolts
Bearings
Metal parts
Suction Pump
Ventouza or air suction bellows
Silicon hose
Electrovalve
Cables
Arduino and shiels board
Grippers
Bushings
Rods
Bumpers
Bearing base
Nuts
Rodella

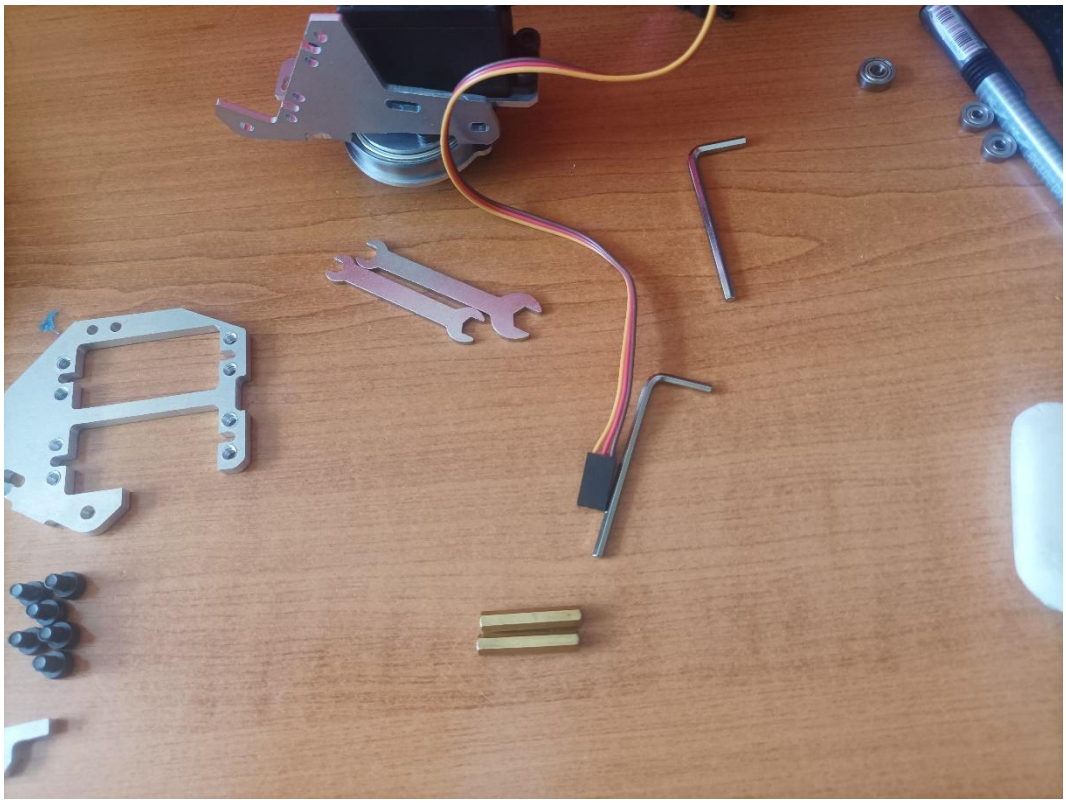
5.2 Installation photos

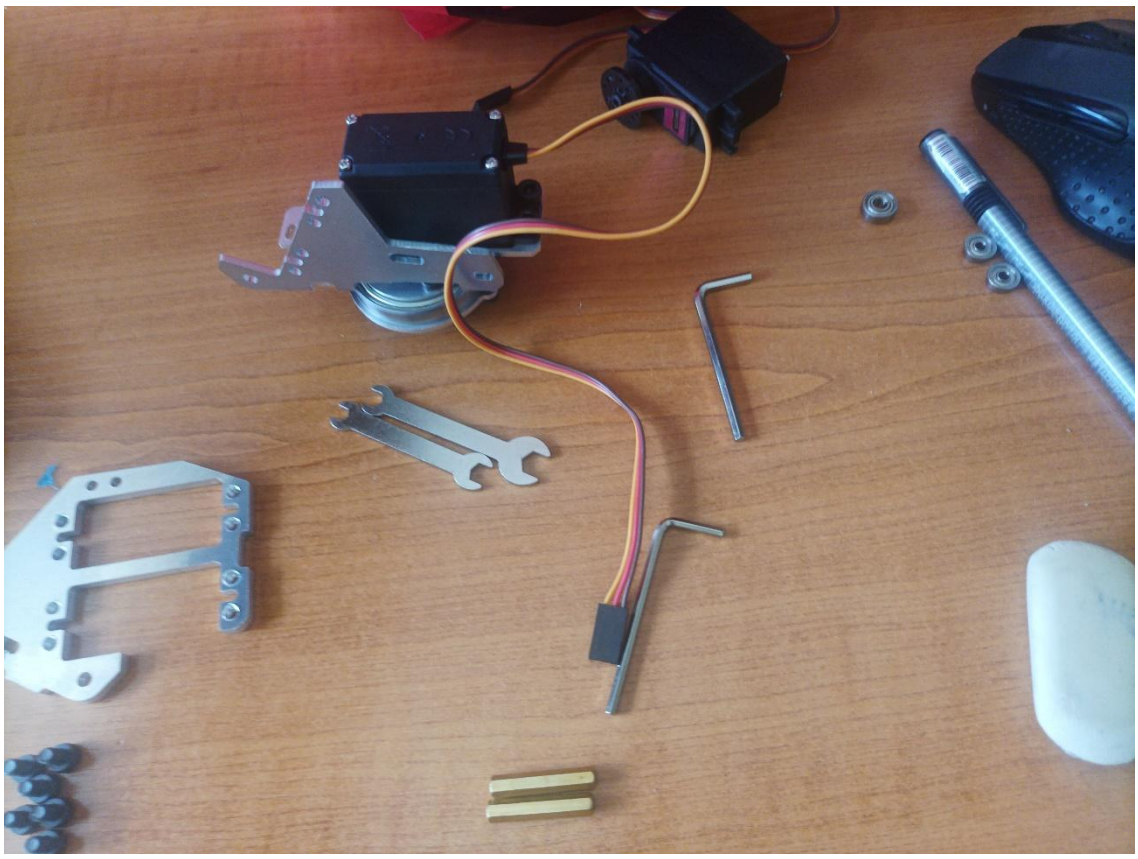


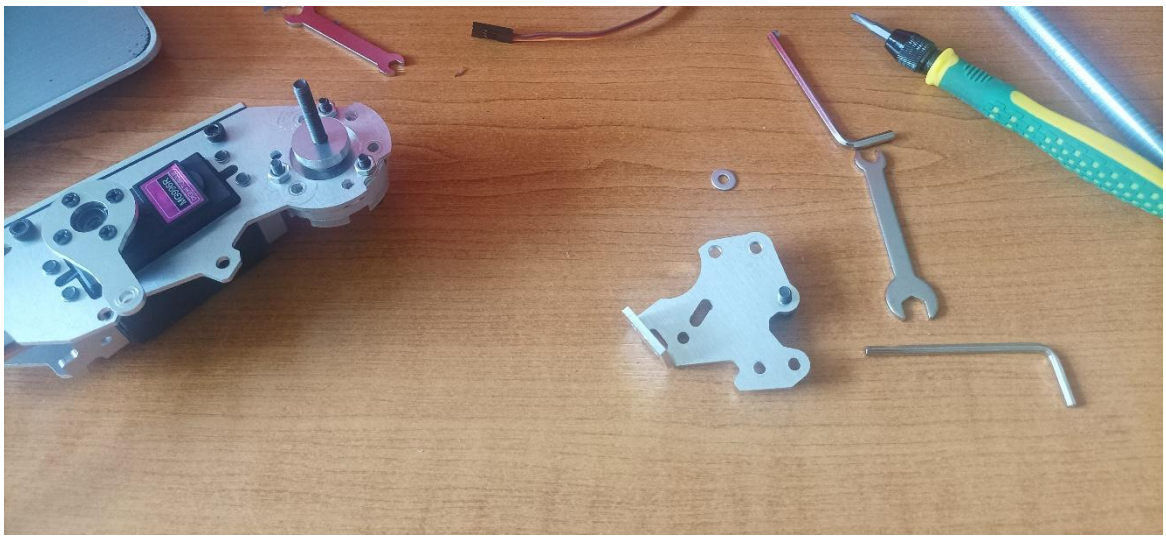












Chapter 6. RESULTS

6.1 Comments on further research

My next aim is to assemble a more complex manipulator with more mechanical compartments. There are similar solutions in the market or can be designed and produced in machinery. Each design should consider the loading conditions and the structural limits of each application.

The use of air vacuum pump is a further innovation in the present assembly and I hope I can develop soon. Requires the two small air suction pumps and the electrovalve.

Also, the programming of enabling the manipulator to do more movement will be important especially if on a larger scale can carry out a complex task. For example, small manipulators can be useful in palletizing yoghurt cups in a yoghurt production line.

The knowledge of the dynamic loads and capabilities of the manipulator can help us design their dimensions and materials for a variety of applications. A trend that was known even in the late 20^{th} century was the Delta Robot which is and easy and efficient way to pick small items from a conveyor using a servo motor and 4 arms in a Delta shape.

6.2 Future innovations and developments

The advance in robotics and the offer of more efficient, cheap and robust installations from the manufacturers has led to wider and wider use in applications. From the industry in cars to the beverage and food industry the advance in carrying out task formerly done by men is an important advance. Robots will not replace human even in Industry 4.0 but will change the needs of skilled professionals.

6.3 Difficulties and drawbacks during installation

During the assembly procedure I faced several difficulties that delayed the due time of the robotic arm. At first the servo motors at the end effector had cables that were smaller in length than the cylinder small column that connects the end effector with the rest assembly and as a result connection to the shield was difficult.

Also, the introduction of pump is a separate function that we leave for further development. The main point of attention on that is the suction hoses which need to be completely sealed from the environment, otherwise will not do the suction.

Another difficulty was that parts needed some kind of modification. So, I need to drill and open some holes then do the threads.

The robotic manipulator needs a fixed place otherwise dynamic loads applied to this will cause the assembly to lack robust. Also, on a larger scale application will lead to friction between part and fatigue which can cause damage. In industrial applications this damage is translated to increased maintenance costs and work hours and is something that preventive maintenance and industrial design deal with. The most critical parts are the joints which have bearings, and the belts used but this is another case of study.

APPENDIX

1.
In the Appendix 1 there is a program for simple motions in our installation.

Code for simple movements

```
#include <Servo.h>

Servo myservo0;
Servo myservo1;
Servo myservo2;
Servo myservo3;
Servo myservo4;
Servo myservo5;


void setup()
{
   myservo0.attach(2);
   myservo1.attach(3);
   myservo2.attach(4);
   myservo3.attach(5);
   myservo4.attach(6);
   myservo5.attach(7);

   myservo0.write(90);
   myservo1.write(90);
   myservo2.write(90);
   myservo3.write(90);
   myservo4.write(90);
   myservo5.write(90);

}
void loop()
{

   myservo0.write(20);
   myservo1.write(90);
   myservo2.write(90);
   myservo3.write(90);
   myservo4.write(90);
   myservo5.write(90);
```

```
delay(1500);

myservo0.write(160);
myservo1.write(90);
myservo2.write(90);
myservo3.write(90);
myservo4.write(90);
myservo5.write(90);
delay(1500);

myservo0.write(90);
myservo1.write(20);
myservo2.write(90);
myservo3.write(90);
myservo4.write(90);
myservo5.write(90);
delay(1500);

myservo0.write(90);
myservo1.write(160);
myservo2.write(90);
myservo3.write(90);
myservo4.write(90);
myservo5.write(90);
delay(1500);

myservo0.write(90);
myservo1.write(90);
myservo2.write(90);
myservo3.write(90);
myservo4.write(90);
myservo5.write(90);
delay(1500);

myservo0.write(90);
myservo1.write(90);
myservo2.write(0);
myservo3.write(90);
myservo4.write(90);
myservo5.write(90);
delay(1500);

myservo0.write(90);
myservo1.write(90);
```

```
myservo2.write(180);
myservo3.write(90);
myservo4.write(90);
myservo5.write(90);
delay(1500);

myservo0.write(90);
myservo1.write(90);
myservo2.write(90);
myservo3.write(90);
myservo4.write(90);
myservo5.write(90);
delay(1500);

myservo0.write(90);
myservo1.write(90);
myservo2.write(90);
myservo3.write(140);
myservo4.write(70);
myservo5.write(90);
delay(2000);

myservo0.write(90);
myservo1.write(90);
myservo2.write(90);
myservo3.write(90);
myservo4.write(90);
myservo5.write(90);
delay(1500);

myservo0.write(90);
myservo1.write(90);
myservo2.write(90);
myservo3.write(90);
myservo4.write(90);
myservo5.write(20);
delay(1500);

myservo0.write(90);
myservo1.write(90);
myservo2.write(90);
myservo3.write(90);
myservo4.write(90);
myservo5.write(160);
```

```
    delay(1500);

    myservo0.write(90);
    myservo1.write(90);
    myservo2.write(90);
    myservo3.write(90);
    myservo4.write(90);
    myservo5.write(90);
    delay(15000000);

}
```

REFERENCES

1. APC Series Transparent Transceiver Module APC220-43, SHENZHEN APPCON TECHNOLOGIES CO.LTD DVER 1.20

2. Σημειώσεις Ρομποτικής Τμήμα Μηχανικών Παραγωγής και διοίκησης Πολυτεχνείο Κρήτης.

3. MICROCHIP IN-CIRCUIT SERIAL PROGRAMMING™ GUIDE

4. Moebius official Store instructions

5. SIEMENS SIMATIC S7-1500 / ET 200MP / ET 200SP CM PtP - Configurations for point-to-point connections. Ανακτήθηκε από τον ιστότοπο https://cache.industry.siemens.com/dl/files/093/59057093/att_896038/v2/s71500_cm_ptp_function_manual_en-US_en-US.pdf

6. SIEMENS SIMATIC S7-1500, ET 200MP, ET 200SP, ET 200AL, ET 200pro, ET 200eco PN Analog Value processing

7. https://www.desertcart.ae/

8. https://www.pinterest.com/

9. https://arduino.stackexchange.com/ Μόνο το διάγραμμα έχω πάρει από αυτό site.

10. https://store.arduino.cc/

11. https://www.timken.com/