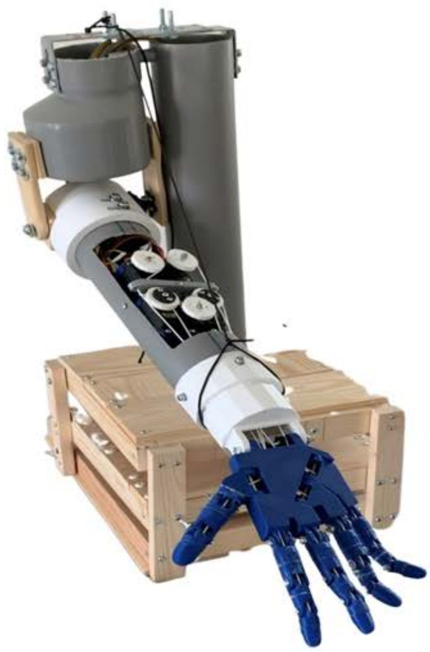
DEGREE PROJECT IN TECHNOLOGY,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2021

Wireless Control of a Robotic Arm

ALAN ISSA

CHRISTOS ANDREANIDIS



KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT

Wireless Control of a Robotic Arm

ALAN ISSA
CHRISTOS ANDREANIDIS

Bachelor's Thesis at ITM
Supervisor: Nihad Subasic
Examiner: Nihad Subasic

TRITA-ITM-EX 2021:42

Abstract

This paper looks at all aspects of developing a robotic arm and hand that consists of five fingers which is able to imitate human movements. The imitation ability, accuracy and factors affecting both points are studied. A project like this requires the interplay of various electrical components to achieve the desired results.

The prototype constructed measured the controller's movements of the fingers with the help of flex sensors. The movements in the elbow and wrist however were measured with the help of potentiometers. The flex sensors and potentiometers were connected to an Arduino Mega which then sent the values with the help of a transmitter. The robotic arm consists of an Arduino Uno, seven servomotors and a receiver that reads the messages sent from the transmitter. All values were converted into degrees that rotated the motor axles accordingly.

The prototype produced positive results, showing that it was able to copy all movements done by the controller. Tests were conducted to study the accuracy and imitation ability. The conclusion was that the factors affecting imitation and accuracy were mostly connected to the weight of the robot and the design of the hand.

Keywords
Mechatronics, Robotic Arm, Wireless Control.

Referat

Trådlös styrning av en robotarm

Denna uppsats behandlar olika aspekter i utvecklingen av en robotarm vars gripdon är en hand med fem fingrar, med syfte att kunna imitera mänskliga rörelser. Imitationsförmågan, noggrannheten samt vilka faktorer som påverkar dessa studeras. För att uppnå ett önskvärt resultat har det krävts styrning och samverkan mellan olika elektroniska komponenter.

I prototypen som presenteras mättes fingrarnas rörelse med hjälp av flexsensorer samt rörelsen i armbåge och handleden med hjälp av vridpotentiometrar. Flexsensorerna och potentiometrarna var anslutna till en Arduino Mega vars värden skickades med hjälp av en sändare. Elektronikkomponenterna som användes i robotarmen var en Arduino Uno, sju servomotorer och en mottagare, vars funktion var att läsa av meddelanden som skickades från sändaren. Alla värden omvandlades till grader och motoraxlarna roterade i enlighet med dessa.

Prototypen uppnådde ett önskvärt betteende då roboten hade förmågan att imitera alla rörelser som utfördes av styrenheten. Noggrannheten och imitationsförmågan undersöktes med olika tester. De mest betydelsefulla faktorer som påverkade imitationen och noggrannheten av prototypen var kopplade till vikten av roboten och designen av handen, enligt slutsatserna som har dragits.

Acknowledgements

This project would not be possible without the guidance of our supervisor Nihad Subasic. A special thanks to Amir Avdic for his assistance with 3D printing and handling of components. Finally, we would like to thank our classmates for their contributions during the seminars and their continuous feedback.

Alan Issa & Christos Andreanidis
Stockholm, May 2021

Contents

List of Figures

List of Tables

Nomenclature

3D	Three-dimensional
AC	Alternating current
CAD	Computer-aided design
CE	Chip enable
CSN	Chip select not
DC	Direct current
DOF	Degrees of freedom
Hz	Hertz
IRQ	Interrupt request
KTH	Royal Institute of Technology
MISO	Master in Slave out
MOSI	Master out Slave in
ms	Millisecond
PWM	Pulse Width Modulation
RF	Radio frequency
SCK	Serial clock
SPI	Serial peripheral interface
USB	Universal Serial Bus
VCC	Voltage Common Collector

Chapter 1

Introduction

1.1 Background

The idea that robots will one day help humanity has been fantasized about for as long as humans can remember. With the development seen in technology and the spread of information, what once was an idea is now looking more like reality. All branches are looking at finding ways to implement machines such as robots.

The human anatomy is incredibly complex and one of the body parts that can be used for countless tasks is the arm. Applying the knowledge of the functions of the arm and hand enables the performance of complicated, dangerous or high accuracy-based tasks. The development of control systems has allowed certain tasks to be performed at much faster speeds with a better precision than before [22]. The diverse functionality of a robotic arm has made it possible for it to be applied in multiple areas such as prosthetics, industrial assembly lines and manufacturing [11], thereby improving the human quality of life.

This project looks at the development of previous work in the area, namely the wireless control of a robotic arm with the help of an exoskeleton control system connected to the operator's body. This project looks at building a robust robotic arm which includes a robotic hand consisting of five fingers. The operator is to easily be able to control the arm without previous knowledge of programming[27].

1.2 Purpose

The purpose of this research is to look at creating a robotic arm with a design similar to the human arm so that the wireless control will be as user-friendly as possible.
The research questions to be investigated in this thesis are the following:

- How reliable is the robot's accuracy?
- What factors affect the accuracy?
- How well can the robot imitate human movements?

1.3 Scope

This report is a bachelor's thesis which means that there are both time and resource limitations. This in turn means that certain parts of the project, such as the design, are simplified. The robot's control system is dependent on the included electrical components which often do not entail the highest precision.

1.4 Method

The project was started through potential sketches as well as identifying the most important components of a robotic arm. A design adapted to the requirements was visualised with the help of a computer-aided design (CAD) program known as Solid-Edge [29]. A readily available design of the hand already existed online for public use [20], the model can be found in Appendix D.1. The rest of the robot was then designed to adapt to the hand. All parts were then created with the help of three-dimensional (3D) printers. In addition, the robot's movements were simulated in a software called Acumen[12], where the code for the simulation can be found in Appendix F. Once this step was completed, the focus was on the assembly of all parts the robot consists of together with all electronic components. Furthermore, the functionality of the components were tested where changes in the code were made to achieve desired behaviour. The source code can be found in Appendix E. Finally, it was time to investigate the research questions, which was done with the help of two tests. The first test was conducted by testing the grip function with six different grips. For the second test, the controller moved their arm at different speeds and looked at how accurately the robot was able to imitate all movements.

Chapter 2

Theory

2.1 Previous Research

The inspiration behind this project are two previous bachelor's theses that also looked at a robot arm and a robot hand. This project is a development on the work done by Miko Nore and Caspar Westerberg in their bachelor thesis "Robotic Arm controlled by Arm Movements". The purpose behind their work was the wireless control of a robotic arm that has the ability to hold certain items [27].

The key area for development is the hand, the previous work used a variant of a clamp to grab items. This means that the degrees of freedom (DOF) are restricted since the hand can only perform one action. This project looks at incorporating a robotic hand consisting of five fingers to help tackle the issue regarding the degrees of freedom.

The robotic hand consisting of five fingers was inspired by the bachelor's thesis "Robotic Hand Controlled by Glove Using Wireless Communication" written by Mehnaz Kazi and Michelle Bill. The purpose behind their project was to construct a glove that can signal their robot hand to move with the help of finger movements [23].

2.2 Physiology of the human arm and hand

The complexity of the human arm and hand depends on the number of bones, joints and tendons that they consist of. The arm itself consists of three larger bones that are connected to each other or to other body parts with the help of a total of three joints [31]. The hand in turn, which is directly connected to the wrist, consists of a total of 27 smaller bones [28].

2.2.1 Anatomy of the arm

The long bone in the arm, also known as the humerus is connected to the shoulder from one end, and the elbow at the other end. The elbow in turn, is a complex joint similar to the shoulder. Since the elbow is composed of a hinge joint and a pivot joint it allows movement with two DOF. Flexion and extension takes place in one direction, with an approximate angle of 150 degrees. Rotation, also known as supination and pronation, in the other direction, has an approximate angle of about 160 degrees [6].These movements can be visualized by Figure 3.5b. Finally, the forearm consist of two larger bones which are connected to the wrist. The anatomy and function of the wrist are explained in more detail in the next section.

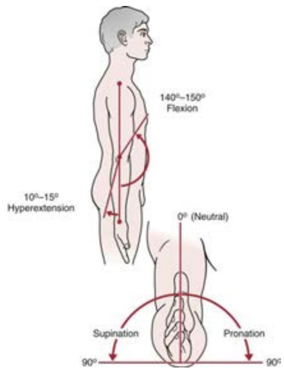
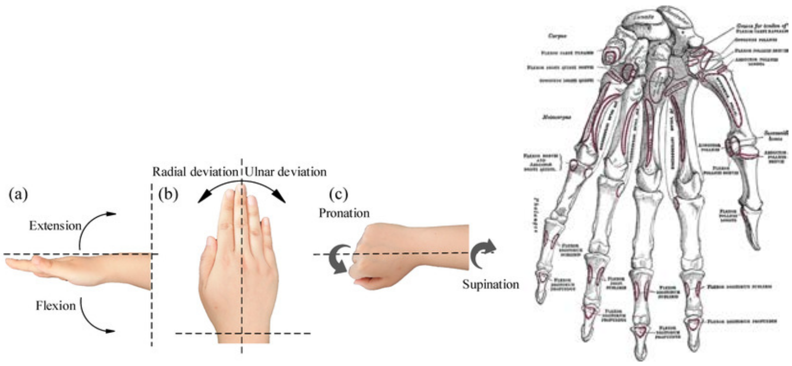


Figure 2.1: Freedom of rotation of the shoulder [7].

2.2.2 Anatomy of the hand

The main function of the hand is to grip, however its complicated structure including the fact that it consists of many small bones makes it a multi-functional tool. As previously mentioned, 27 smaller bones are included in the hand and to ease the understanding of its anatomy and physiology, the hand is usually divided into three main parts, namely the wrist, the palm and fingers [28].

Each part of the hand consists of bones of varying sizes and shapes, Figure 2.2b gives a good overview of the structure of the hand. The movements in the wrist can be summed up as extension, radial deviation, ulnar deviation, pronation and supination [19]. All movements can be seen seen in Figure 2.2a.



(a) Physiology of the wrist [19]

(b) The skeleton of the hand [28]

Figure 2.2: The anatomy and physiology of the hand.

The mobility of the fingers allows the hand to grip objects in different ways. The way the fingers bend for each grip gives a name to the grip itself, Figure 2.3 shows the most common grips. Each grip also involves a certain amount of force and precision depending on the way the fingers are used. This report will not look at the forces associated with each grip, instead the robots ability to imitate each grip is studied.

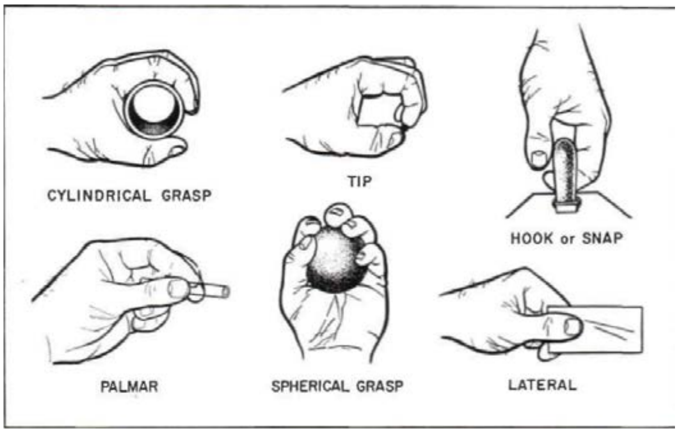


Figure 2.3: Six most common grips [30].

2.3 Components

Controlling a robotic arms' movements can be done with the help of a wide range of methods, however the main methods are hydraulic control, pneumatic control, and control with the help of electronic components. Hydraulics are characterized by large components that in turn can control larger forces. Pneumatics can take larger forces as well however cannot be used in feedback systems. Smaller electronic components are favored in feedback systems, such as the system used in this project, however the drawback is the ability to only take small forces.

Electronic actuators offer high precision and repeatability, all while still being noise free and free from environmental hazards such as fluid leaks. Their ability to be quickly reprogrammed and networked is the reason it is the favored method to build a robotic arm [18].

2.4 Potentiometer

A potentiometer is a variable voltage divider with a rotatable wiper in direct contact with a resistive path. A potentiometer has three terminals, as can be seen in Figure 2.4, where one is connected to the feeding voltage, one for ground and the third is directly connected to the resistance. Depending on the position of the wiper, different voltages are emitted by rotating the wiper. This allows the potentiometer to be used for reading different voltages as different signals. Most of these components have a wiper that is restricted to a 0-180° rotation which in turn means that the voltage will be between 0 V and the maximal value of the supply voltage [13].

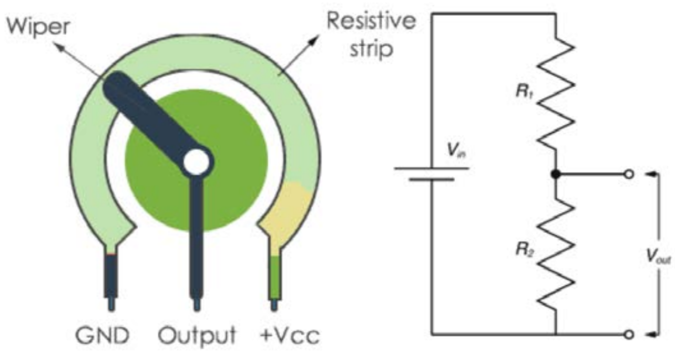


Figure 2.4: Visual representation of the terminals in a potentiometer [8].

2.5 Servomotors

Servomotor is a collective name for motors with feedback, which means that there are many different types of servomotors for different needs. What is common in the construction of all servomotors is that they consist of an alternating current (AC) or direct current (DC) motor and a potentiometer connected to a control circuit. It is a feedback system which means that the movement of the motor leads to rotation of the potentiometer where its resistance generates a certain signal in the form of a voltage which is finally compared to the input voltage. This allows the control of how many degrees the motor axle has rotated very easy [21].

The control of servomotors is done with the help of electrical pulses, also known as pulse width modulation (PWM). The control signal usually occurs every 20 milliseconds (ms), which corresponds to a frequency of 50 hertz (Hz). It is the duration of the pulse that determines the position of the servomotor. A pulse of 1 ms usually corresponds to 0°, while pulses of 1.5 and 2 ms usually corresponds to 90° and 180° respectively. Figure 2.5 depicts how PWM affects the servomotors position [10].

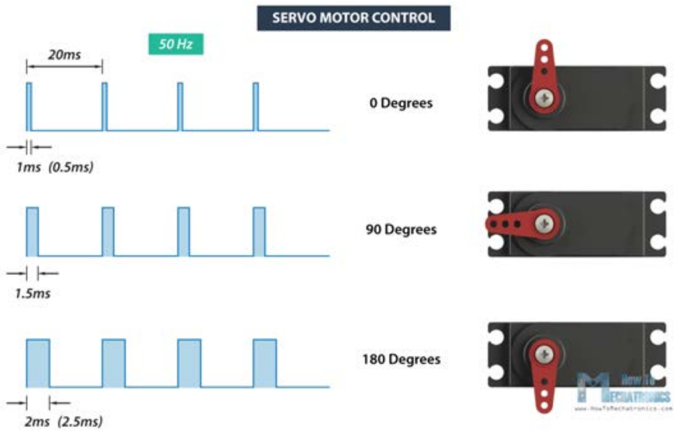


Figure 2.5: Correlation between PWM and angle [10].

2.6 Flex Sensor

To measure the amount of bending, also known as defection, a flex sensor can be used. To be put simply, a flex sensor is a variable resistor that increases in resistance the more it is bent. The bending is measured by measuring the angle created between the front and the back of the sensor. The sensor consists of two terminals, where the operating voltage required to activate the sensor can range anywhere between 0-5V DC [15]. A basic configuration of a flex sensor can be seen in Figure 2.6

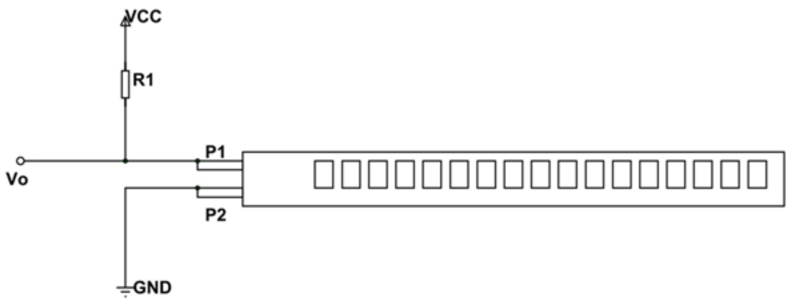


Figure 2.6: Circuit with a flex sensor showing how the two pins of the sensor can be configured [9].

2.7 Radio Frequency Module

A small sized component that is used to transmit or receive radio signals is known as a radio frequency (RF) module. An RF module can play a vital role in embedded systems where information needs to be sent wirelessly. For many applications, RF is the method of choice when transferring information wirelessly since there is no line-of-sight requirement [16].

The module of interest in this study is the nRF24L01+ Wireless Module. To communicate, the information is sent and received on a certain channel. The channel is the frequency the transceivers are operating at. The nRF24L01+ module operates at any frequency in the range of 2.4-2.525 GHz, also known as the frequencies in the 2.4 GHz ISM band. There are 125 different channels to choose from since each channel requires under 1 MHz as well as the 1 MHz used for spacing. A visual interpretation can be seen in Figure 2.7

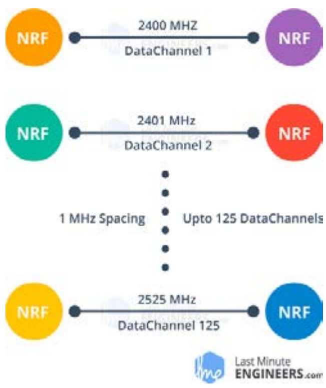


Figure 2.7: Frequency channels on the nRF24L01+ Wireless Module [25].

With the help of a 4-pin Serial Peripheral Interface (SPI), the nRF24L01+ module is able to communicate with maximum data rate of 10 megabits per second. With the help of the SPI interface, parameters such as output power, frequency channel and data rate can be configured.

As can be seen in Figure 2.8, there are 8 pins on the wireless module. The first pin is GND which stands for the ground pin. The power supply is connected to the Voltage Common Collector (VCC) pin which, as stated previously, can be between 1.9-3.3V. CE stands for Chip Enable and depending on the mode the module is in, it will either transmit or receive. The Chip select not (CSN) pin is used to listen and process data on its SPI. The SPI bus Master provides clock pulses which is then accepted by the serial clock (SCK) pin. Master out Slave in (MOSI) and Master in Slave out (MISO) are the SPI input and output. This means that in cases where the transceiver is used only as a transmitter and receiver respectively, only one of the pins needs to be used. Finally, when new data is available to be processed, the interrupt request (IRQ) pin will alert the Master[25].

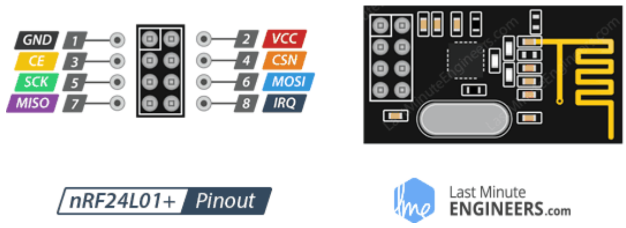


Figure 2.8: Pin configuration on the nRf24L01+ Wireless Module [25].

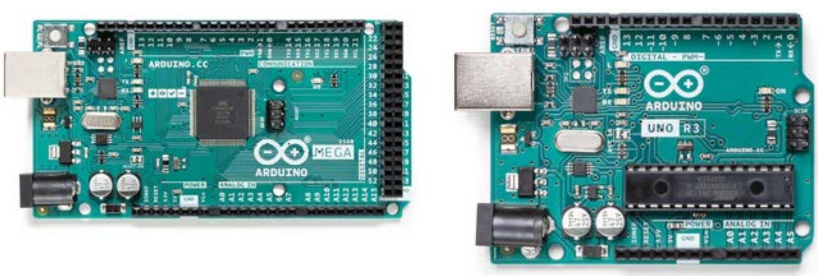
2.8 Microcontroller

It is of importance in this thesis that the control of the motors is regulated, values read from flex sensors and potentiometers are read and converted so that the motors react accordingly. This is done with the help of microcontrollers.

In general, interactions with electromechanical components are done with the help of a microcontroller which, simply put, is an integrated circuit. A microcontrollers memory, microprocessor unit and other peripherals help control an electrical system[24].

Arguably the most cost effective and popular microcontrollers are Arduino Nano/Uno/Mega. The Atmega 2560 microcontroller is the basis for the open-source microcontroller board. The board of interest is the Arduino Mega which includes 16 analog input pins, 54 digital input/output pins where 14 of them are PWM outputs, a power jack and Universal Serial Bus (USB) connection. The layout of an Arduino Mega board can be seen in Figure 2.9a

The boards of interest are the Arduino Uno and Mega. The difference between the two boards is simply the number of pins. Both boards have a set number of analog input pins, digital input/output pins where a percentage of them are PWM outputs as well, a power jack and USB connection. The layout of both boards can be seen in Figure 2.9.



(a) Arduino Mega [2] (b) Arduino Uno[3]

Figure 2.9: The layouts of the Arduino boards.

The board can be powered either with the help of a USB cable or by batteries. As previously stated, there are designated analog pins where each pin will read the voltage between 0-5V and translate it to a 10-bit value. It is important to state that all the analog pins can be used as digital input/output. SPI communication is done with the help of MISO, MOSI, SCK and SS where each have different designated pins on the two boards[14]. The digital pins can be configured as inputs or outputs depending on the user's needs. Digital pins can be seen as a less superior version of the analog pins where it is not able to map out the voltages. Here it is simply HIGH or LOW which translate to 5V or 0V[4]. To achieve voltages between 0V and 5V, PWM can be used. With the help of PWM one can control the output voltage, however this can only be achieved by the certain designated digital pins that have this increased functionality[1].

Chapter 3

Demonstrator

This chapter looks at the construction of the robotic arm both from a mechanical and electrical aspect. Once construction is completed, the prototype is then used to conduct tests in order to answer the research questions that were brought up at the beginning of the report.

3.1 Electrical Components

The robotic arm is to be controlled by a person wirelessly where the controller's movements in the arm are mimicked by the robotic arm. The construction can therefore be divided into two parts; the part that is used by the controller and the part which is the robotic arm itself.

3.1.1 The Controller's Components

The controller is to have the movements in their fingers, wrist and elbow measured so that the robotic arm can then mimic the movements based off of the values read by the flex sensors and potentiometers. A list of all the components and the amount can be seen in Table 3.1.

Table 3.1: The controller's electrical components

Component	Amount
Arduino Uno	1
Flex sensor	5
Potentiometer	2
Resistor 10kΩ	5
nRF24L01+	1
4x AA Batteries	1

Every finger is to be measured by a flex sensor, the elbow and wrist movement are to be measured by potentiometers where the rotation in the joints change the resistance which is then picked up by the Arduino Uno. Finally the data is then transmitted with the help of the nRF24L01+ transceivers. It is left to the controller how they want to power the Arduino Uno, a pair of batteries or a USB connection to a computer will both work, however this project used 4 AA batteries. The circuit can be seen in Figure 3.1.

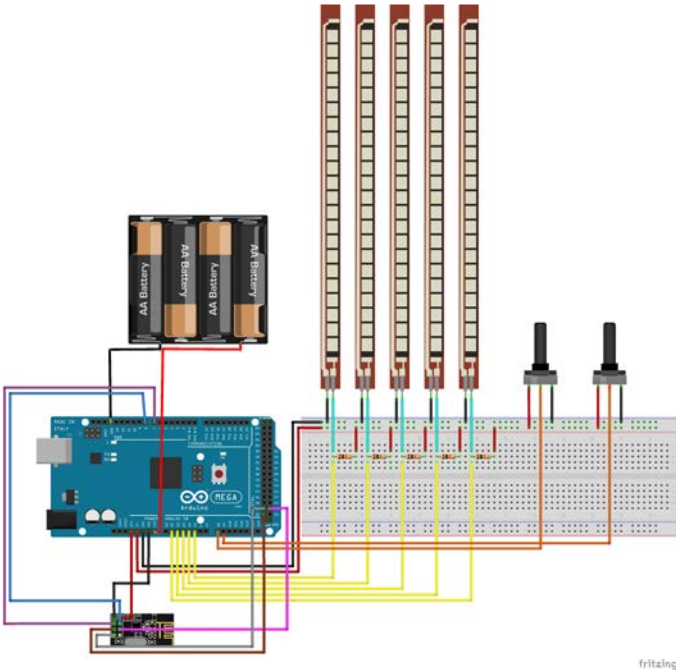


Figure 3.1: The controllers circuit created with Fritzing [17].

3.1.2 The Robotic Arm's Components

The robotic arm translates the data read from the controller into degrees the motor axle is to rotate. Table 3.2 shows a list of all the components required to allow the robotic arm to move.

Table 3.2: The robotic arm's electrical components

Component	Amount
Arduino Mega	1
MG995 Servo Motor	5
DS3225 Servo Motor	2
nRF24L01+	1
4x AA Batteries	4

The data is received by the nRF24L01+ transceiver and translated by the Arduino Mega. Each finger is then controlled by an MG995 servomotor. Due to the weight of the hand and the large torque required to rotate the wrist and elbow, the high torque DS3225 servo motors were used. The motors need to be operating at high torques which in turn means that each motor require high voltages. Therefore 4 sets of 4 AA batteries are used. The circuit for the robotic arm along with how each motor is power can be seen in Figure 3.2.

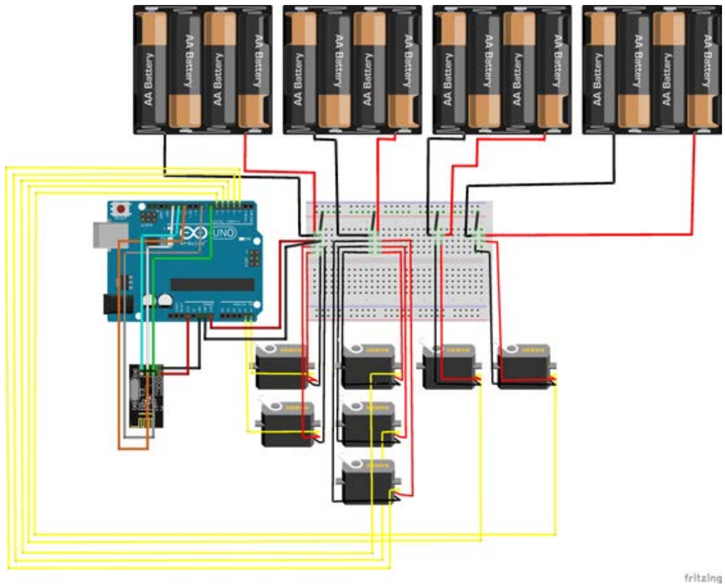


Figure 3.2: The robotic arms circuit created with Fritzing [17].

3.2 Software and Programming

The programming language C is often used when working with Arduino and this project is no exception. The code was written in Arduinos integrated development environment which is available for free on their website[5].

Two different codes were written; one for the transmitter and one for the receiver. The code regarding the transmitter simply read all the values from the flex sensors and potentiometers, created a vector containing all the data and finally sent it to the receiver. Once the receiver received the data, it converts the numbers to angles between 0-180 where finally the motors rotate accordingly. An overview of the code can be seen in Figure 3.3.

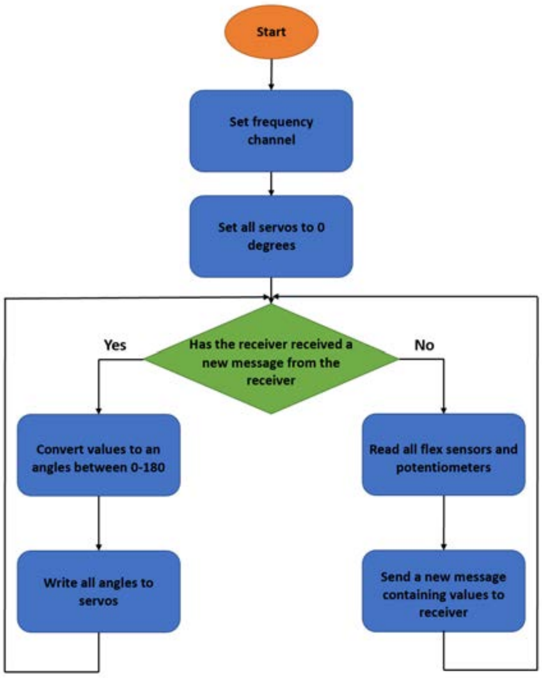


Figure 3.3: A software flow chart created by Microsoft PowerPoint [26].

3.3 Construction

The construction work of this project has been divided into two main parts, namely the robotic arm itself and a separate construction to control the robotic arm.

3.3.1 Design of the Robotic Arm

The robot arm consists of four larger parts, namely a hand, a forearm, an upper arm and a base, which in turn are made up of smaller components. The hand is printed with the help of a 3D printer and consists of a palm and five fingers, where each finger is made up of six smaller details. In addition, an M3 screw is used between each joint to represent the hinge joint between each part of the finger. The contraction and extraction of the fingers is done with the help of a thin rope and rubber band, which corresponds to the muscle function in the body. Each part has two smaller holes, one for the rope used to contract the finger and one for a rubber band for the extraction of the finger into its original position.

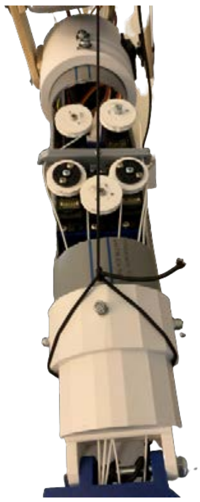
(a) Assembly of the hand

(b) Assembly of the thumb

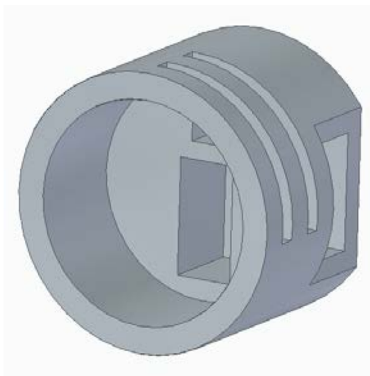
(c) Assembly of the index

Figure 3.4: Assembly of the hand, pictures taken by the authors.

The forearm consists of a 3D-printed wrist and elbow as well as a tube that forms the forearm itself. The tube has an opening adapted to the dimensions of a motor holder where the motors that pull the fingers are located, as you can see in Figure 3.5a. The elbow is designed in such a way that it has room for two more servomotors, one for rotation of the entire forearm and one for vertical movements, as shown in Figure 3.5b. All parts are assembled together with the help of screws and nuts of different sizes.



(a) Assembly of the forearm



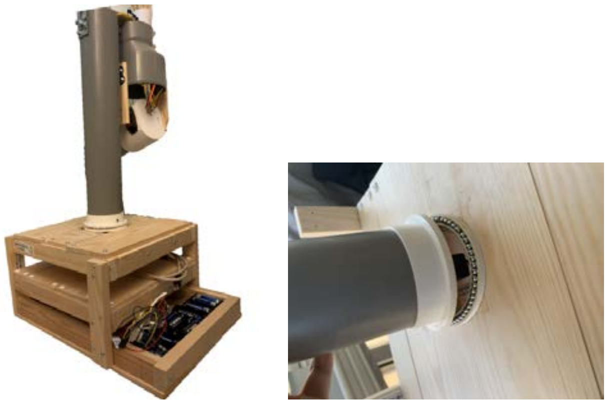
(b) CAD picture of the elbow

Figure 3.5: Figure of the forearm and the elbow designed in Solid-Edge [29].

In order for the forearm to be connected to the rest of the structure, it was necessary to design a type of fork that is screwed onto the upper arm and thus allows vertical movements, as shown in Figure 3.6. The upper arm is in turn attached to the base itself with the help of metal angles and screws. It has been designed to allow rotation that mimics the movements of the shoulder when adding another motor and removing certain screws. However, this leads to weakened stability.

Figure 3.6: The upper arm and it's fork, picture taken by the authors.

Finally, the base consists of a pipe that sits on a square construction made of smaller wooden beams and wooden boards assembled with screws. The square wooden construction consists of three ”floors”, as shown in Figure 3.7a. The floors are self-designed and the 3D-printed ball bearings, as shown in Figure 3.7b, facilitate the rotation of the robot around its own axis. This rotation has to be done manually in this project, however if desired one can place an extra motor there to do it wirelessly.



(a) Assembly of the base (b) Ball bearings

Figure 3.7: Figure of the base and its ball bearings, pictures taken by the authors.

3.3.2 Design of the controller

The control panel can, like the robot, be divided into four parts which in this case consist of wooden boards and are adapted to the user's body size. A board sits, as shown in the left corner of Figure 3.8, on the user's back and fulfills two functions, partly for convenience, partly since it is a suitable place to attach all electronic equipment such as the micro-controller and batteries. The board mounted on the user's forearm is attached to the back plate using a potentiometer that acts as an axis of rotation. Furthermore, the board that is to sit on the forearm is mounted using two potentiometers and a wooden-angle, as shown in the middle of Figure 3.8. This way the freedom of rotation is increased and the user has the opportunity to rotate the forearm vertically and horizontally. Finally, a glove is also mounted at the end of the forearm board, where the flex sensors sit with double-sided adhesive tape, as shown in the right corner of Figure 3.8.

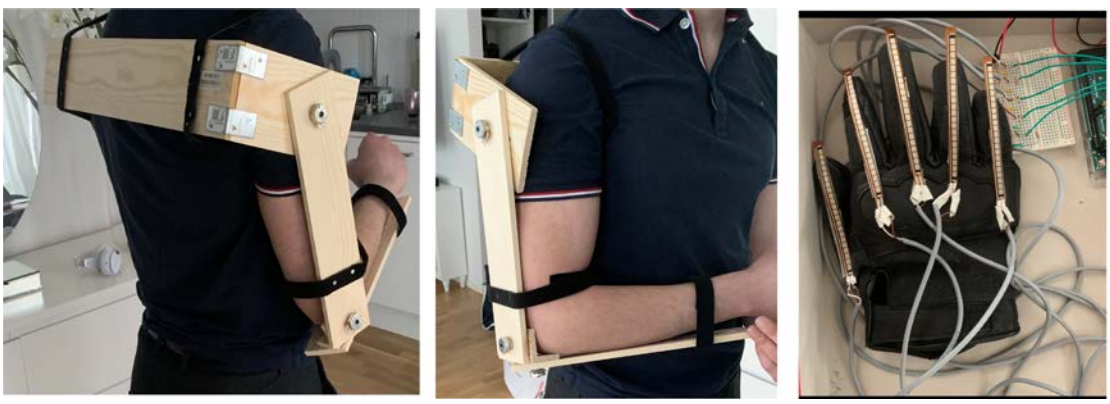


Figure 3.8: The control panel from two views and the glove with the flex sensors, pictures taken by the authors.

3.4 Testing

The research questions in this project are primarily about the robot's precision and imitation ability. Two different tests were conducted to answer the research questions.

3.4.1 Imitation tests

The imitation tests were used to examine whether the current construction of the hand can withstand different types of basic grips that were discussed in the theory section. To test the hand's ability to grip, different objects were used to see whether or not the robot was able to get a grip. The results consist of a simple pass or fail grade for each type of grip.

3.4.2 Accuracy test

Due to the weight of the robotic arm, all motors will be exposed to large amount of torque. It is therefore important to examine whether or not the construction is able to withstand sudden forces. The test is conducted by moving the fingers, wrist and elbow at the same time and separately at varying speeds.

Chapter 4

Results

This chapter presents the results achieved from testing as well as the overall performance of the robotic arm.

4.1 Overall Performance

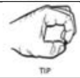
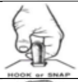
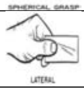
The robot is able to mimic all movements done by the controller, even if the controller is in a different room. However, the poor 3D printing quality sometimes causes the middle finger and thumb to get stuck during flexion. Furthermore, the motor in the elbow was not able to perform as well as what is specified in the data sheet which can be seen in Appendix A.2, therefore an elastic band was connected from the wrist to the elbow to help the motor lift the forearm and hand.

Overall the results are positive for a first prototype. Despite certain concerns with the rigidity of the base, the robot was able to lift objects weighing up to 100 grams and hold objects weighing up to 220 grams without lifting.

4.2 Imitation Test

When performing the imitation test, the tests were based on imitating the six basic grips discussed in the theory section. Table 4.1 shows the type of grip that was tested and whether or not the robot could successfully perform each type of grip.

Table 4.1: Results of the imitation tests

Imitation tests		
Grip type	Figure description	Result
Cylindrical Grasp		PASS
Tip		FAIL
Hook or Snap		PASS
Palmar		FAIL
Spherical Grasp		PASS
Lateral		FAIL

4.3 Accuracy test

After conducting multiple tests it was observed that the fingers and wrist could move as desired when moving each one at a time. With the help of the rubber band, the elbow was also able to move as desired as well. Without the rubber band the elbow would simply not move.

Issues arose when moving multiple parts at the same time. At times, the hand was able to move all fingers at the same time without any problems, however there were anomalies where some fingers would only move slightly. The wrist was always able to move as desired even if other parts were moving, however the same cannot be said for the elbow. The elbow requires a better power source because there were a few instances where the elbow would give up if the batteries powering the motor showed the slightest bit of aging. The largest issue regarding the accuracy of the elbow was the rigidity of the base. The base as it is currently is not stable enough to withstand the weight of the arm, so at times when moving the elbow, the ball bearings seen in Figure 3.7b slightly move out of place.

Chapter 5

Discussion and conclusions

5.1 Discussion

The robotic arm performed very well, however there are some points that can be further analyzed and discussed. A large part of the project was dedicated to designing a robotic arm that can imitate the movements done by a controller. This was done by focusing on some key grips that were introduced in section 2.2.2. Table 4.1 shows that the arm is able to imitate three grips, depending on the users needs these three grips can be enough. However, if one wants more flexibility, the hand can be redesigned to fit the users demands.

Regarding the accuracy of the robot, the results were once again positive for the most part. The robot was able to accurately copy the controllers' movements, however some irregularities arose when moving all parts at the same time. These irregularities were almost always connected to moving the arm vertically, meaning that the arm is simply too heavy. The problem regarding the weight was only exaggerated due to the ball bearings in the base. The project was constructed to show that, if need be, the entire robotic arm can rotate 360 degrees, to do that ball bearings and an extra motor are needed. However, the ball bearings were 3D printed and consists of two parts that can easily be separated. This means that moving the robotic arm rigorously can cause the two parts to separate a little.

The final research question was determining the factors that affect the robot's accuracy. It has already been discussed that the weight of the construction, 3D printing quality and stability of the base are the main factors to look at. The weight of the construction mostly came from the large number of screws that are in the hand. The weight of the screws along with the long forearm made the job of the motor in the elbow difficult. As for the stability of the base, this can simply be solved by removing the ball bearings or using ball bearings that are not 3D printed.

Many parts of the project were 3D printed or obtained from the warehouse available for all students at the Royal Institute of Technology (KTH). Despite this, the cost of the project exceeded the SEK 1000 budget set from the beginning. This is mostly due to the number of flex sensors and the number of servo motors the construction requires.

5.2 Conclusions

A prototype of a robotic arm was constructed that could be controlled from afar. The prototype was able to complete three of the six basic gripping techniques and was able to deliver a well approved accuracy for a first prototype. The factors affecting the accuracy mostly depended on the weight of the construction and rigidity of the base. Further development of the robot is possible by focusing on solving these factors.

Chapter 6

Recommendations and future work

The project has a lot of potential which with the recommendations that will be presented shortly, a more robust construction can be built that allows even more DOF with an improved accuracy.

Starting with the hand model, the one used in this project is not complex at all due to the budget limitations. For a wider variety of gripping techniques, one can look at more complicated designs. Another improvement regarding the hand would be to look at 3D printed screws for the hand. This will allow the weight of the construction to drop significantly.

The forearm consists of a single tube where the only improvement that can be done is shortening the tube so that there is just enough space for all the servo motors. By shortening the tube, the need for high torque motors is negated as well as removing the need to use the elastic band that is currently helping the motor in the elbow to operate.

Our ambition with the project from the beginning was that the upper arm would be able to rotate as well, however due to the budget limitations this was not feasible. The upper arm in the current construction is attached to the tube that is coming out of the base. By removing two of the screws seen in Figure 3.6 one can allow the robot to mimic the movements done at the shoulder.

The base is rigid, however due to the structure of the ball bearings there is some instability in that region. Firstly, one can add an extra servo motor inside of the ball bearings to rotate the tube coming out of the base allowing for an extra DOF. Secondly, the ball bearing should be more compact so that it does not consist of two parts like it currently does.

Bibliography

[1] Arduino. *"Analog Input Pins"*. URL: https://www.arduino.cc/en/Tutorial/Foundations/AnalogInputPins (visited on 04/01/2021).

[2] Arduino. *"ARDUINO MEGA 2560 REV3"*. URL: https://store.arduino.cc/arduino-mega-2560-rev3 (visited on 04/01/2021).

[3] Arduino. *"ARDUINO Uno REV3"*. URL: https://store.arduino.cc/arduino-uno-rev3 (visited on 04/30/2021).

[4] Arduino. *"Digital Pins"*. URL: https://www.arduino.cc/en/Tutorial/Foundations/DigitalPins (visited on 04/01/2021).

[5] Arduino. *"Software"*. URL: https://www.arduino.cc/en/software (visited on 02/15/2021).

[6] Ryan K. Card and Jason B. Lowe. "Anatomy, Shoulder and Upper Limb, Elbow Joint". In: *StatPearls* (Aug. 2020). DOI: https://www.ncbi.nlm.nih.gov/books/NBK532948/. (Visited on 02/02/2021).

[7] Clinical Gate. *"Elbow,"*. Mar. 2015. [Online]. URL: https://clinicalgate.com/elbow-3/ (visited on 02/02/2021).

[8] Codrey. *"Potentiometer: How it Works?"*. Aug. 2020. [Online]. URL: https://www.codrey.com/resistor/potentiometer-how-it-works/ (visited on 02/05/2021).

[9] Components101. *"Flex Sensor," Mar. 18, 2018*. URL: https://components101.com/sensors/flex-sensor-working-circuit-datasheet (visited on 02/08/2021).

[10] Dejan. *How Servo Motor Works & How To Control Servos using Arduino*. URL: https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/ (visited on 02/08/2021).

[11] Rahul Raj Devaraja, Rytis Maskeliūnas, and Robertas Damaševičius. "Design and Evaluation of Anthropomorphic Robotic Hand for Object Grasping and Shape Recognition". In: *Computers* 10.1 (2021). ISSN: 2073-431X. DOI: 10.3390/computers10010001. URL: https://www.mdpi.com/2073-431X/10/1/1 (visited on 01/24/2021).

[12] Effective Modeling Group (EMG). *Acumen*. Version 2016$_0$8$_3$0. Feb. 7, 2021. URL: http://www.acumen-language.org/.

[13] Rod Elliott. *Beginners' Guide to Potentiometers*. ESP. Jan. 2002. [Online]. URL: https://sound-au.com/pots.htm (visited on 02/05/2021).

[14] ELPROCUS. *"Arduino Mega 2560 Board"*. URL: https://www.elprocus.com/arduino-mega-2560-board/ (visited on 04/01/2021).

[15] ELPROCUS. *"Flex Sensor Working and Its Applications"*. URL: https://www.elprocus.com/flex-sensor-working-and-its-applications/?web=1&wdLOR=c1B5DFFDF-EA9B-42F7-B75C-2E794E01298A (visited on 02/08/2021).

[16] ELPROCUS. *"RF Module – Transmitter Receiver"*. URL: https://www.elprocus.com/rf-module-transmitter-receiver/ (visited on 02/08/2021).

[17] Fritzing. *Fritzing*. Version 0.9.6. Apr. 2, 2021. URL: https://fritzing.org/download/.

[18] Carlos Gonzalez. *What's the Difference Between Pneumatic, Hydraulic, and Electrical Actuators?* MachineDesign. Apr. 2015. [Online]. URL: https://www.machinedesign.com/mechanical-motion-systems/linear-motion/article/21832047/whats-the-difference-between-pneumatic-hydraulic-and-electrical-actuators (visited on 02/03/2021).

[19] Bo Huang et al. "Wearable Stretch Sensors for Motion Measurement of the Wrist Joint Based on Dielectric Elastomers". In: *Sensors* 17.12 (2017). ISSN: 1424-8220. DOI: 10.3390/s17122708. URL: https://www.mdpi.com/1424-8220/17/12/2708/htm (visited on 02/02/2021).

[20] InMoov. *"InMoov body parts library : Right-Hand.* URL: http://inmoov.fr/inmoov-stl-parts-viewer/?bodyparts=Right-Hand (visited on 01/19/2021).

[21] Jameco. *"How Servo Motors Work"*. URL: https://www.jameco.com/Jameco/workshop/Howitworks/how-servo-motors-work.html (visited on 02/05/2021).

[22] Haruhisa Kawasaki and Tetsuya Mouri. "Humanoid Robot Hand and its Applied Research". In: *Journal of Robotics and Mechatronics* 31 (Feb. 2019), pp. 16–26. DOI: 10.20965/jrm.2019.p0016. (Visited on 01/24/2021).

[23] Mehnaz Kazi and Michelle Bill. "Robotic Hand Controlled by Glove Using Wireless Communication". DEGREE PROJECT IN MECHANICAL ENGINEERING, FIRST CYCLE. Stockholm, Sweden: Royal Institute of Technology, KTH, 2020. URL: https://kth.diva-portal.org/smash/get/diva2:1462080/FULLTEXT01.pdf (visited on 01/19/2021).

[24] Robert Keim. *What Is a Microcontroller? The Defining Characteristics and Architecture of a Common Component*. All About Circuits. Mar. 2019. [Online]. URL: https://www.allaboutcircuits.com/technical-articles/what-is-a-microcontroller-introduction-component-characteristics-component/ (visited on 04/01/2021).

[25] Last Minute Engineers. *"How nRF24L01+ Wireless Module Works Interface with Arduino"*. URL: https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/ (visited on 04/01/2021).

[26] Microsoft. *Microsoft PowerPoint*. Version 16.0.13901.20312. Apr. 2, 2021. URL: https://www.microsoft.com/en-us/microsoft-365/powerpoint.

[27] Miko Nore and Caspar Westerberg. "Robotic Arm controlled by Arm Movements". DEGREE PROJECT IN TECHNOLOGY, FIRST CYCLE. Stockholm, Sweden: Royal Institute of Technology, KTH, 2019. URL: https://kth.diva-portal.org/smash/get/diva2:1373883/FULLTEXT01.pdf (visited on 01/19/2021).

[28] Surbhi Panchal-Kildare and Kevin Malone. "Skeletal Anatomy of the Hand". In: *Hand Clinics* 29.4 (2013). Management of Hand Fractures, pp. 459–471. ISSN: 0749-0712. DOI: https://doi.org/10.1016/j.hcl.2013.08.001. URL: https://www.sciencedirect.com/science/article/pii/S0749071213000565 (visited on 02/02/2021).

[29] Siemens. *Solid Edge*. Version ver.2020. 2020. URL: https://solidedge.siemens.com/en/ (visited on 05/01/2021).

[30] Craig L Taylor and Robert J. Schwarz. "The Anatomy and Mechanics of the Human Hand". In: *Academy of Sciences-National Research Council* (1955. [Online]). URL: http://www.oandplibrary.org/al/pdf/1955_02_022.pdf (visited on 04/22/2021).

[31] Glenn C. Terry and Thomas M. Chopp. "Functional Anatomy of the Shoulder". In: *Journal of Athletic Training* 35 (July 2000. [Online]). DOI: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1323385/. (Visited on 02/02/2021).

Appendix A

Servomotor datasheets

A.1 MG995 datasheet

31150-MP

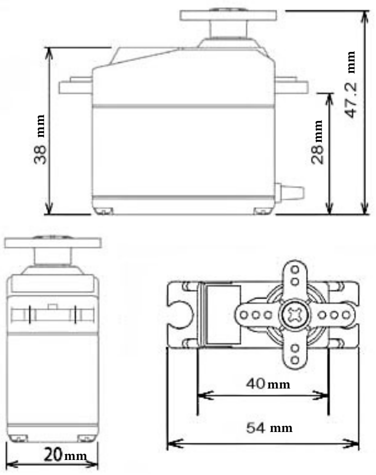
MG995 High Speed Servo Actuator

The unit comes complete with color coded 30cm wire leads with a 3 X 1 pin 0.1" Pitch type female header connector that matches most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

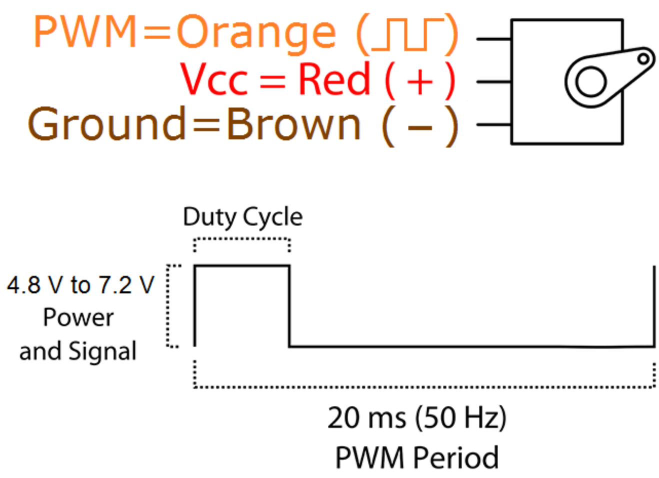
This high-speed servo actuator is not code dependant; You can use any servo code, hardware or library to control them. The MG995 Actuator includes arms and hardware to get started.

Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 8.5 kgf·cm (4.8 V), 10 kgf·cm (6 V)
- Rotation Angle: 120deg. (+- 60 from center)
- Operating speed: 0.2 s/60º (4.8 V), 0.16 s/60º (6 V)
- Operating voltage: 4.8 V to 7.2 V
- Dead band width: 5 µs
- Stable and shock proof double ball bearing design
- Metal Gears for longer life
- Temperature range: 0 ºC – 55 ºC



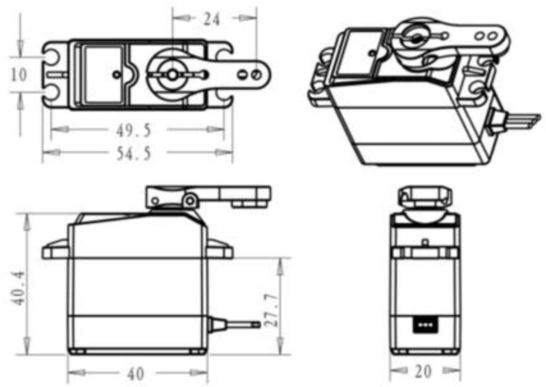
31150-MP
MG995 High Speed Servo Actuator



MARLIN P. JONES & ASSOC., INC.
P.O. Box 530400 Lake Park, Fl 33403
800-652-6733 FAX 561-844-8764
WWW.MPJA.COM

A.2 DS3225 datasheet

产品型号（Product Name）：DS3225
产品描述（Product Description）：6V 25kg RC Digital Servo
产 品 图（Drawing）



1. 使用环境条件 Apply Environmental Condition

No.	Item	Specification
1-1	存储温度 Storage Temperature Range	-30℃～80℃
1-2	运行温度 Operating Temperature Range	-15℃～70℃
1-3	工作电压范围 Operating Voltage Range	4.8-6.8V

2. 机械特性 Mechanical Specification

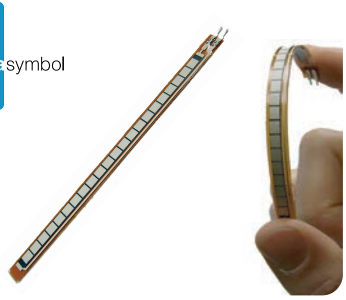
No.	Item	Specification
2-1	尺寸 Size	40*20*40.5mm
2-2	重量 Weight	60g
2-3	齿轮比 Gear ratio	275
2-4	轴承 Bearing	Double bearing
2-5	舵机线 Connector wire	300±5mm
2-6	马达 Motor	3-pole(s)
2-7	防水性能 Waterproof performance	IP66

Brand	Dsservo	
Product Name	DS3225	DS3225 Coreless Digital Servo
Torque	21 kg-cm @5V	22 kg-cm @5V
	24.5 kg-cm @6.8V	24 kg-cm @6V
		25.5 kg-cm @7.4V
Speed	0.15 sec/60° @5V	0.1 sec/60° @5V
	0.13 sec/60° @6.8V	0.08 sec/60° @6V
		0.07 sec/60° @7.4V
Note	High Torque Ensure that the BEC has enough output power Recommend using 6V	High Speed、High Efficiency Recommend using 7.4V

Appendix B

Flex sensor datasheet

spectrasymbol



FLEX SENSOR FS

Special Edition Length

Features

- Angle Displacement Measurement
- Bends and Flexes physically with motion device
- Possible Uses
  - Robotics
  - Gaming (Virtual Motion)
  - Medical Devices
  - Computer Peripherals
  - Musical Instruments
  - Physical Therapy
- Simple Construction
- Low Profile

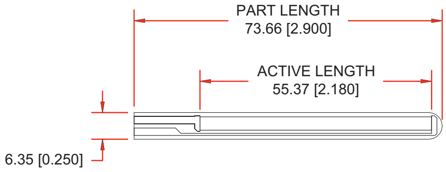
Mechanical Specifications

-Life Cycle: >1 million
-Height: ≤0.43mm (0.017")
-Temperature Range: -35°C to +80°C

Electrical Specifications

-Flat Resistance: 25K Ohms
-Resistance Tolerance: ±30%
-Bend Resistance Range: 45K to 125K Ohms (depending on bend radius)
-Power Rating : 0.50 Watts continuous. 1 Watt Peak

Dimensional Diagram - Stock Flex Sensor

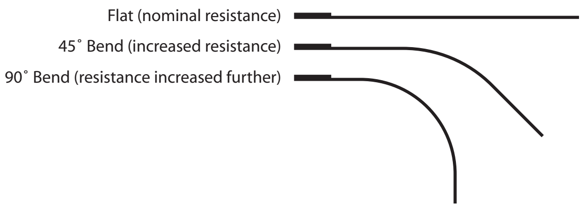


How to Order - Stock Flex Sensor

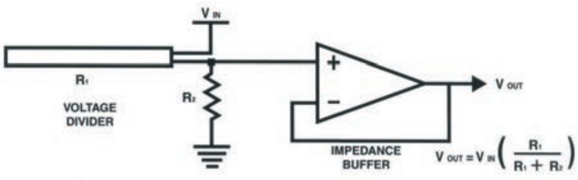
FS — L — 0055 — 253 — ST

Series	Model	Active Length	Resistance	Connectors
FS = Flex Sensor	L = Linear	0055 = 55.37mm	253 = 25K Ohms	ST = Solder Tab

How It Works



BASIC FLEX SENSOR CIRCUIT:

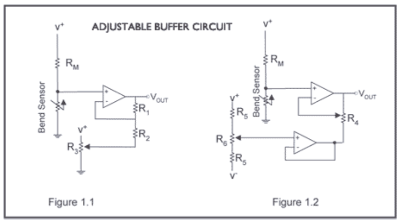


Following are notes from the ITP Flex Sensor Workshop

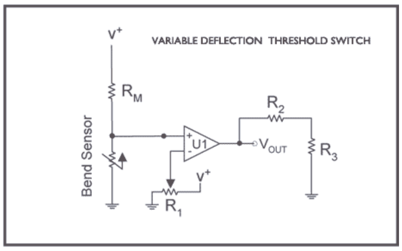
"The impedance buffer in the [Basic Flex Sensor Circuit] (above) is a single sided operational amplifier, used with these sensors because the low bias current of the op amp reduces errer due to source impedance of the flex sensor as voltage divider. Suggested op amps are the LM358 or LM324."

"You can also test your flex sensor using the simplest circut, and skip the op amp."

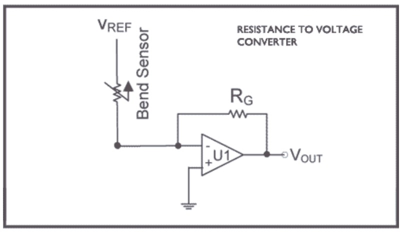
"Adjustable Buffer - a potentiometer can be added to the circuit to adjust the sensitivity range."



"Variable Deflection Threshold Switch - an op amp is used and outputs either high or low depending on the voltage of the inverting input. In this way you can use the flex sensor as a switch without going through a microcontroller."



"Resistance to Voltage Converter - use the sensor as the input of a resistance to voltage converter using a dual sided supply op-amp. A negative reference voltage will give a positive output. Should be used in situations when you want output at a low degree of bending."



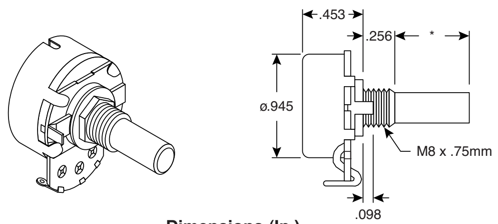
Appendix C

Potentiometer datasheet

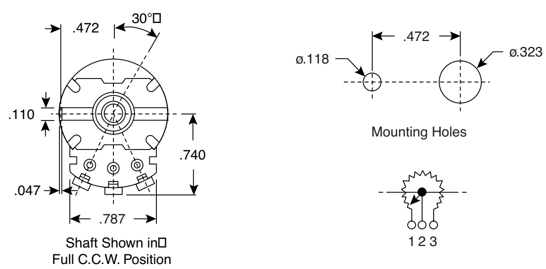
Date: 08/07/08 **Ref.#:** TW-700187

ALPHA

Potentiometer
31VA Series
31VC Series
31VJ Series



Dimensions (In.)
(except where noted)



Specifications:

- Resistance tolerance: ±20%
- Taper: linear(31VA series, 31VC series), audio(31VJ series)
- Maximum working voltage: 500V(linear taper) 250V(audio taper)
- Power Dissipation: .5W(linear taper) .25W(audio taper)
- Insulation resistance: > 100MΩ @ 500VDC
- Minimum resistance: if total R < 2K then ≤ 20Ω; if total R > 2K then ≤ 100Ω; if total R > 1m then ≤ 200Ω
- Rotation life: 15,000 cycles
- Resistive element: carbon composition
- Electrical rotation angle: 285° ± 5°
- Rotation angle: 300° ± 5°
- Rotation torque: 20~200 gf.cm
- Shaft stop strength: > 6Kgf.cm/min
- Terminals: solder lugs
- Shaft length*: .335"(31VA series, 31VJ series), 1.32"(31VC series)
- RoHS Compliant

MOUSER STOCK NO.	Value (Ω)
31VA301-F	1K
31VA302-F	2K
31VA303-F	2.5K
31VA305-F	5K
31VA401-F	10K
31VA403-F	25K
31VA405-F	50K
31VA501-F	100K
31VA503-F	250K
31VA505-F	500K
31VA601-F	1M
31VA602-F	2M
31VA605-F	5M

MOUSER STOCK NO.	Value (Ω)
31VC205-F	500
31VC301-F	1K
31VC302-F	2K
31VC303-F	2.5K
31VC305-F	5K
31VC401-F	10K
31VC403-F	25K
31VC405-F	50K
31VC501-F	100K
31VC503-F	250K
31VC505-F	500K
31VC601-F	1M
31VC605-F	5M

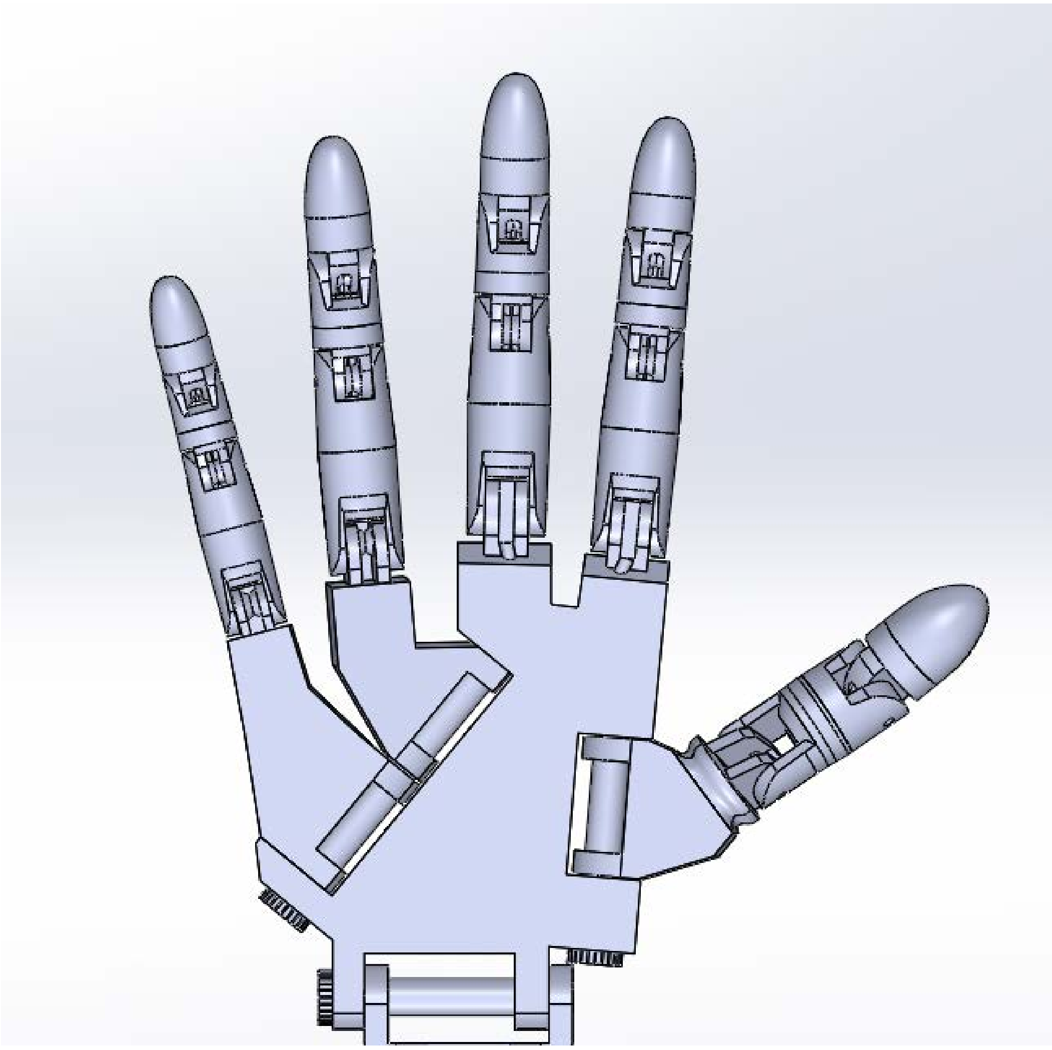
MOUSER STOCK NO.	Value (Ω)
31VJ301-F	1K
31VJ305-F	5K
31VJ401-F	10K
31VJ405-F	50K
31VJ501-F	100K
31VJ503-F	250K
31VJ505-F	500K
31VJ601-F	1M

Available from Mouser Electronics www.mouser.com (800) 346-6873

Appendix D

Computer Aided Design models

D.1 Model of the hand



D.2 Model of the elbow

Appendix E

Source code

E.1 Transmitter code

```
/*
 * Wireless Control of a Robotic Arm (Transmitter Code)
 *
 * Date: 06-05-2021
 * Written by: Alan Issa & Christos Andreanidis
 * Examinor: Nihad Subasic
 * TRITA-nr: 2021:42
 * University: Royal Institute of Technology, KTH
 * Course: Degree Project in Mechatronics, MF133X
 *
 * The arduino code written is used to transmit all values
    read from flex sensors and potentiometers.
 */

#include <SPI.h>
#include "RF24.h"

RF24 myRadio (7, 8);
byte addresses[][6] = {"0"};

int Vector[7];  //Vector conatining the values of the flex
   sensor

//Values read from analog read
int value1;    //(Thumb)
int value2;    //(Index)
int value3;    //(Middle)
int value4;    //(Ring)
```

```
int value5;     //(Little)
int value6;     //(Wrist)
int value7;     //(Elbow)

//Pin associated with each flex sensor and potentiometer (
    number after each pin indicates which body part defined
    above)
int flexpin1 = A5;
int flexpin2 = A4;
int flexpin3 = A3;
int flexpin4 = A2;
int flexpin5 = A1;
int wristpin = A8;
int elbowpin = A9;



void setup()
{
  Serial.begin(115200);
  delay(1000);

  //Set up RF module
  myRadio.begin();
  myRadio.setChannel(115);
  myRadio.setPALevel(RF24_PA_MAX);
  myRadio.setDataRate( RF24_250KBPS ) ;
  myRadio.openWritingPipe( addresses[0]);
  delay(1000);
}

void loop()
{
  // Read all the values from the flex sensor
  value1 = analogRead( flexpin1 );
  value2 = analogRead( flexpin2 );
  value3 = analogRead( flexpin3 );
  value4 = analogRead( flexpin4 );
  value5 = analogRead( flexpin5 );
  value6 = analogRead( wristpin );
  value7 = analogRead( elbowpin );

  // Enter all the value into the vector
  Vector[0] = value1;
```

```
  Vector[1] = value2;
  Vector[2] = value3;
  Vector[3] = value4;
  Vector[4] = value5;
  Vector[5] = value6;
  Vector[6] = value7;

  // Send the vector
  myRadio.write(&Vector, sizeof(Vector));

}
```

E.2 Receiver code

```
/*
 * Wireless Control of a Robotic Arm (Receiver Code)
 *
 * Date: 06-05-2021
 * Written by: Alan Issa & Christos Andreanidis
 * Examinor: Nihad Subasic
 * TRITA-nr: 2021:42
 * University: Royal Institute of Technology, KTH
 * Course: Degree Project in Mechatronics, MF133X
 *
 * The arduino code written is used to control all motors
    depending on the values sent from the transmitter.
 */

#include <SPI.h>
#include "RF24.h"
#include <Servo.h>

RF24 myRadio (7, 8); //7,8
byte addresses[][6] = {"0"};

//Define servomotors
Servo motor1;   //(Thumb)
Servo motor2;   //(Index)
Servo motor3;   //(Middle)
Servo motor4;   //(Ring)
Servo motor5;   //(Little)
Servo motor6;   //(Wrist)
Servo motor7;   //(Elbow)

//Define which pin it is connected to (Note that some are
   analog some are digital)
int motorpin1 = A5; //A5
int motorpin2 = A4;
int motorpin3 = 2;  //DIGITAL 2
int motorpin4 = 3;
int motorpin5 = 4;
int motorpin6 = 5;
int motorpin7 = 6;

int vector[7];  //Define my vector
```

```

//Values read from the receiver
int value1;
int value2;
int value3;
int value4;
int value5;
int value6;
int value7;

//Used to convert the values into angles
int angle1;
int angle2;
int angle3;
int angle4;
int angle5;
int angle6;
int angle7;




void setup()
{
  //Attach all motors
  motor1.attach( motorpin1 );
  motor2.attach( motorpin2 );
  motor3.attach( motorpin3 );
  motor4.attach( motorpin4 );
  motor5.attach( motorpin5 );
  motor6.attach( motorpin6 );
  motor7.attach( motorpin7 );

  //Make sure all motors are starting at 0
  motor1.write(0);
  motor2.write(0);
  motor3.write(0);
  motor4.write(0);
  motor5.write(0);
  motor6.write(115);
  motor7.write(0);

  Serial.begin(115200);
  delay(1000);
```

```

  //Set up RF module
  myRadio.begin();
  myRadio.setChannel(115);
  myRadio.setPALevel(RF24_PA_MAX);
  myRadio.setDataRate( RF24_250KBPS ) ;
  myRadio.openReadingPipe(1, addresses[0]);
  myRadio.startListening();
}


void loop()
{

  if ( myRadio.available()){  //If a message is available
      myRadio.read( &vector, sizeof(vector) );  //Retrieve the vector sent from the transmitter

      //Take out all the values sent from the transmitter
      value1 = vector[0];
      value2 = vector[1];
      value3 = vector[2];
      value4 = vector[3];
      value5 = vector[4];
      value6 = vector[5];
      value7 = vector[6];

      //Convert all values to angles
      angle1 = map(value1, 650,750,0,180);
      angle2 = map(value2, 620,720,0,180);
      angle3 = map(value3, 660,760,0,180);
      angle4 = map(value4, 650,700,0,180);
      angle5 = map(value5, 650,750,0,180);
      angle6 = map(value6, 150,360,115,25);
      angle7 = map(value7, 400, 700, 90, 120);


      if (angle6>115){  //Make sure that the wrist value isnt outside of our intervall
        angle6=115;
      }

      if (angle6<25) {  //Make sure that the wrist value isnt outside of our intervall
```

```
        angle6=25;
      }

      if (angle7>120) { //Make sure that the elbow value
         isnt outside of our intervall
        angle7=120;
      }

      if (angle7<90){ //Make sure that the elbow value isnt
         outside of our intervall
        angle7=90;
      }

      //Make all motors rotate according to the angles
      motor1.write(angle1);
      motor2.write(angle2);
      motor3.write(angle3);
      motor4.write(angle4);
      motor5.write(angle5);
      motor6.write(angle6);
      motor7.write(angle7);
    }
  }
```

Appendix F

Acumen code

```
//Program name: Robotic Arm Simulator
//Andra program- Simulering
//Date: 2021-03-03
//Authors: Alan Issa & Christos Andreanidis
//Examiner: Nihad Subasic
//Course: MF133X

model Main(simulator) =
initially
   alpha1=0, //The angle between the upper arm and z axis
   alpha1'=pi/4,
   rate1=pi/4,

   alpha2=0, //Angle between hand and y axis
   alpha2'=0,
   rate2=pi/2,

   alpha3=0, //Angle between the fingers and the xy plane
   alpha3'=0,
   rate3=pi/2,

   _3D=(),
   _3DView=((12,-5,9) ,(-8,10,-8) ), //Camera View

   //Three coordinates will be used the simulation,
   //there are 3 simulations
   //each simulation uses one set of coordinates
   i=1,  //The first
   j=0,  //The second
   k=0   //The third
```

```
always
   //Directly start the first simulation the shoulder rotation
   alpha1'=rate1,
   //Once the shoulder rotation is done, the wrist will begin rotating
   alpha2'=j*rate2,
   //Finally the fingers will start moving
   alpha3'=k*rate3,



   _3D = (   //Initiate all the shapes required for the arm

    //The base
    Box center=(0,1,3) size=(2,1,6) color=black

    //The base
    Box center=(0,0.5,5.5) size=(2,0.25,1) color=black

    //Shoulder joint
    Sphere center=(0,0,5.5) size=(0.5) color=blue

    //The upper arm
    //Which coordinates are in polar form dependent on alpha1
    Cylinder center=(2.75*sin(alpha1),0,(5.5-2.75*cos(alpha1)))
size=(5.5,0.25) color=red rotation=(pi/2,-alpha1,0)

    //Elbow joint that is dependent on the angle created by the upper arm
    Sphere center=(5.5*sin(alpha1),0,5.5-5.5*cos(alpha1))
size=(0.5) color=blue

    //The forearm which is also dependant on the angle alpha1
    Cylinder center=(5.5*sin(alpha1)+2.75*cos(alpha1),0,
5.5*(1-cos(alpha1))+2.75*sin(alpha1))
size=(5.5,0.2) color=red rotation=(-alpha1,0,pi/2)

    //The wrist
    Sphere center=(5.5*sin(alpha1)+5.5*cos(alpha1),0,
5.5*(1-cos(alpha1))+5.5*sin(alpha1)) size=(0.5) color=blue

    //The hand which is also dependant on the angle alpha1
    Box center=( 5.5*sin(alpha1)+6.25*cos(alpha1), 0,
5.5*(1-cos(alpha1))+6.25*sin(alpha1) ) size=(1.5,1.5,1)
color=yellow rotation=(alpha2,-alpha1,0)
```

```
    //Fingers:
    //The coordinates of the fingers will change depending on simulation.
    //A lot of trigonometry!

    //Index
    Cylinder center=
( (i+j-k)*(5.5*sin(alpha1)+7.5*cos(alpha1))+k*(7+0.5*cos(alpha3)),
(i+j)*(0.65*cos(alpha2)),
i*(5.5*(1-cos(alpha1))+7.5*sin(alpha1))+
j*(0.65*sin(alpha2))+k*(0.5*sin(alpha3)) )
size=(1,0.1) color=green rotation=(i*(-alpha1)+k*(-alpha3), alpha2, pi/2)

    //Middle
    Cylinder center=( (i+j-k)*(5.5*sin(alpha1)+7.5*cos(alpha1))+
k*(7+0.5*cos(alpha3)), (i+j)*(0.25*cos(alpha2)),
i*(5.5*(1-cos(alpha1))+7.5*sin(alpha1))+
j*(0.25*sin(alpha2))+k*(0.5*sin(alpha3)) )
size=(1,0.1) color=green rotation=(i*(-alpha1)+k*(-alpha3),alpha2,pi/2)

    //Ring
    Cylinder center=
( (i+j-k)*(5.5*sin(alpha1)+7.5*cos(alpha1))+k*(7+0.5*cos(alpha3)),
(i+j)*(-0.25*cos(alpha2)),
i*(5.5*(1-cos(alpha1))+7.5*sin(alpha1))+
j*(-0.25*sin(alpha2))+k*(0.5*sin(alpha3)) )
size=(1,0.1) color=green rotation=(i*(-alpha1)+k*(-alpha3),alpha2,pi/2)

    //Pinkie
    Cylinder center=
( (i+j-k)*(5.5*sin(alpha1)+7.5*cos(alpha1))+k*(7+0.5*cos(alpha3)),
(i+j)*(-0.65*cos(alpha2)), i*(5.5*(1-cos(alpha1))+7.5*sin(alpha1))+
j*(-0.65*sin(alpha2))+k*(0.5*sin(alpha3)) )
size=(1,0.1) color=green rotation=(i*(-alpha1)+k*(-alpha3),alpha2,pi/2)

    //Thumb:
    Cylinder center=( 5.5*sin(alpha1)+6.25*cos(alpha1),
(i+j-k)*(1.25*cos(alpha2))+k*(0.75+0.5*cos(alpha3)),
i*(5.5*(1-cos(alpha1))+6.25*sin(alpha1))+
j*(1.25*sin(alpha2))+k*(0.5*sin(alpha3)) )
size=(1,0.1) color=green rotation=(alpha2+alpha3,0,0)
    ),


    //The shoulder will continue rotating until the angle pi/3 is reached
```

```
 if (alpha1>pi/3) then
   rate1=-pi/4   //Start rotating in the opposite direction
 noelse,

 //Once it is back to starting position, stop rotation of the shoulder
 if (alpha1<0) then
    rate1=0
 noelse,

 //Now it is time to rotate the hand
 //Polar coordinates will be used once again
 //however as a function of a new angle
 if (rate1==0) then
    i=0,
    j=1   //Is used in "center" for the fingers.
          //Allows new coordinates to be used
          //also allows alpha2 to not be zero
 noelse,

 //Once the hand is flipped over
 if (alpha2>pi) then
    rate2=-pi/2  //Chage direction
 noelse,

 //Once the hand is back at its starting position
 if (alpha2<0) then
    rate2=0 //let the hand rest
 noelse,

 //Once hand is back at original position.
 //Start moving the fingers, this is the third and final simulation
 if (rate1==0 && rate2==0) then
    k=1 //used to change coordinates and activate alpha3
 noelse,

//Once the fingers are all inwards
 if (alpha3>pi/1.7) then
    rate3=-pi/2  //Change the rotation direction
 noelse,

 //Once we are back to starting position
 if (alpha3<0) then
    rate3=0   //Third and final simulation complete!
 noelse
```

TRITA TRITA-ITM-EX 2021:42