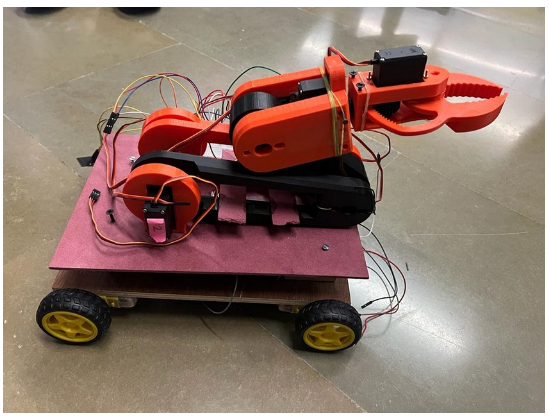
**EMBEDDED SYSTEMS x SENSORS**

**SECTION:**

**PROFESSOR SANKET PATEL**
**PROFESSOR VINOD MALL**

# MASC - 213

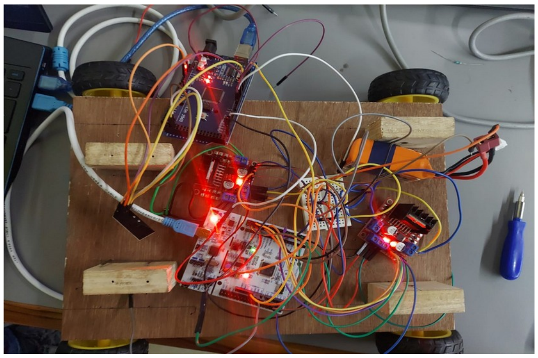
***“Mechanical Arm Sensor Controlled”***



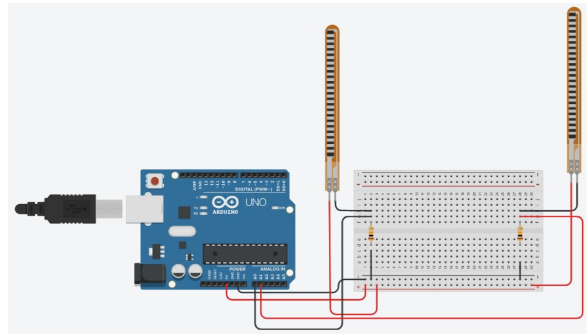
Group: 2 & 13

**Contents**

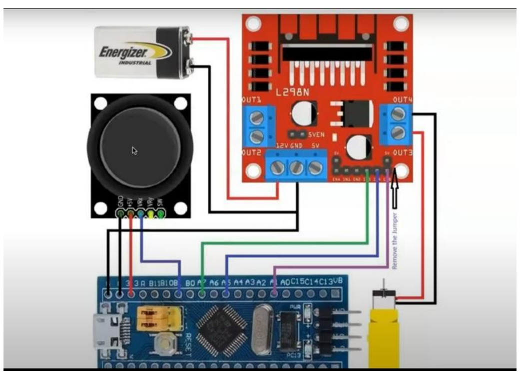
## 1. CIRCUIT DIAGRAMS:



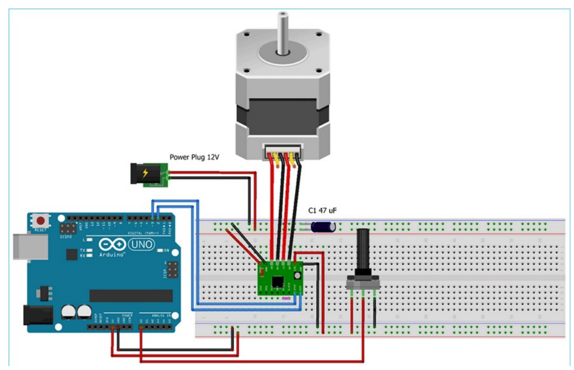
Circuit I: Base of the project




Circuit II: Flex Sensor Connections




Circuit III: DC Motor Connections




Circuit IV: Stepper Motor Connections

- https://www.tinkercad.com/things/0maIxxxCauJ?sharecode=FgTfrsfbIDEOnh93VmH6jxE5SqIfeZXtuGOsvpzANAI
- https://www.tinkercad.com/things/f2SHiW4SESk-spectacular-bruticus-tumelo/editel?sharecode=BHgrfbIRiTxJagMf_5178qIkD236uFWlzIVHj5jjqzk

## 2. COMPONENT LIST

**Delta Electronics:**

| Name | Qnty | Price |
|---|---|---|
| NEMA 17 stepper Motor | 1 | 767 |
| MG995 Servo | 1 | 354 |
| 18650 1- cell holder | 1 | 36 |
| Lithium ion 18650 2200mah | 2 | 307 |
| Battery charger | 1 | 354 |
| Cell Holder | 1 | 89 |
| PCA9685 16 Channel 12 BIT PWM Servo Motor Driver | 1 | 531 |
| Drawn Parts: Brushless Motor | 2 | 1062 |
| Lipo B3 charger | 1 | 413 |
| Lipo battery 850mah 11.1v | 1 | 826 |
| Slide Switch | 2 | 12 |
| Burgstrip-M | 1 | 12 |
| Hookup wire | 2 | 118 |
| Wheel 10cm | 4 | 472 |
| ESC 30A Simonk | 1 | 531 |
| MG995 Servo 180 Degree | 5 | 2065 |
| A4998 | 1 | 118 |
| L298 Driver Board | 2 | 354 |
| Jumper Cable | 2 | 142 |
| Motor: BO Motor | 4 | 402 |
| PB2 connector | 1 | 30 |
| Breadboard 840 points | 1 | 118 |

**Lakhi Electronics:**

| | | |
|---|---|---|
| Shaft | 4 | 187 |
| A4988 | 1 | 167 |

**A-one Bearings:**

| | | |
|---|---|---|
| Ball bearing 8x16 | 10 | 236 |

**Shantinath Electronics:**

| | | |
|---|---|---|
| Motor BLDC | 1 | 531 |
| ESC | 2 | 1040 |

**Techtonics:**

| | | |
|---|---|---|
| Arduino NANO V3.0 | 1 | 485 |
| GY-521 MPU6050- Triple Axis Gyro Accelerometer Module | 2 | 318 |
| PCA9685 16 -Channel 12-Bit PWM/Servo Driver I2C Interface | 1 | 425 |
| Flex Sensor 2.2 inch | 2 | 740 |
| NRF24L01 2.4 GHz Wireless Transceiver Module | 3 | 255 |
| Force Sensitive Resistor - 0.2" 5. 08mm Pressure Sensor | 1 | 325 |
| 9V Original HW HighQuality Battery - 5Pc | 1 | 25 |
| Fire or Flame Sensor Module (1 meter Range) | 3 | 147 |
| Breadboard 840 Tie Points Solderless DIY Project Circuit | 2 | 138 |

## 3. EXPLANATION:

***Stepper Motor:***

NEMA 17 Stepper Motor: One of the functionalities of the base is to move the whole structure left or right based on the direction of the object targeted to be picked. We decided to use the NEMA17 motor because it is a DC motor that works in discrete steps. Moreover, it provides accurate CONTROL and can be differentiated based on torque, steps per revolution, and input voltage. Stepper motors do not rotate, they step, and the NEMA17 motor has a step angle of 1.8 deg. means it covers 1.8 degrees in every step. We have used a stepper driver module- A4988 stepper motor driver to control the working of the stepper motor. It sends the current to the stepper motor through various phases. The direction pin of the motor is used to control the direction of the motor, i.e., clockwise or counter-clockwise. A step pin is used to manage the steps. Here in this project, we have used full-step micro-step resolution. To get enough torque to move the stepper smoothly, we gave an external voltage of 14.8 V(3.7*4). In this project, we control the stepper motor based on the value of X2 received from the MPU6050.

***STM32 and DC motors:***

STM32 will receive the data from the Arduino using I2C communication in the base. We have used I2C1 of STM32. In I2C, we have two pins: SCL and SDA, which are connected to the corresponding pins of the Arduino. The data received here is the X and Y values of the MPU of the glove transmitted to the Arduino using NRF24L01. Four DC motors are connected to the STM32 through the L298N motor driver using which we are giving PWM signals to run the motors. The DC motors are operated according to the X and Y values, and the whole base will move. If the hand glove is tilted down, then the Y values will be in the range of 270 to 360 degrees.

All the motors will be configured in the forward direction, and the base will move forward. If the hand is tilted up, then the Y values will be in the range of 0 to 90 degrees. All the motors will be configured in the reverse direction, and the base will move backward. If the hand is tilted left, then the X values will be in the range of 270 to 360 degrees. The two left motors will be configured in the reverse direction, and the two right motors will be configured in the forward direction. If the hand is tilted right, then the X values will be in the range of 0 to 90 degrees. The two left motors will be configured in the forward direction, and the two right motors will be configured in the reverse direction. The motors are operated using the PWM using the Timer1. Four motors are connected to the four channels of the Timer1.

***NRF:***

NRF is a transceiver module. It uses the 2.4 GHz band and it can operate with baud rates from 250 kbps up to 2 Mbps. If used in open space and with lower baud rate its range can reach up to 100 meters. The module can use 125 different channels which gives a possibility to have a network of 125 independently working modems in one place. Each channel can have up to 6 addresses, or each unit can communicate with up to 6 other units at the same time. Upon research we found that NRF is one of the best options when one needs wireless communication. Due to higher range, we have used NRF in this project to establish communication between arm and the glove.

***MPU6050:***

The **MPU6050** is a very popular accelerometer gyroscope chip that has six axes sense with a 16-bit measurement resolution. The MPU6050 is a Micro-Electro-Mechanical Systems (MEMS) that consists of a 3-axis Accelerometer and 3-axis Gyroscope inside it. This helps us to measure acceleration, velocity, orientation, displacement and many other motion-related parameters of a system or object. MEMS accelerometers are used wherever there is a need to measure linear motion, either movement, shock, or vibration but without a fixed reference. All accelerometers work on the principle of a mass on a spring, when the thing they are attached to accelerates, then the mass wants to remain stationary due to its inertia and therefore the spring is stretched or compressed, creating a force which is detected and corresponds to the applied acceleration. Specifically about this project, when the MPU6050 moves in any one of the axes, the arm must move accordingly.

***FLEX SENSOR***:

A flex sensor or bend sensor is a low-cost and easy-to-use sensor specifically designed to measure the amount of deflection or bending. A flex sensor is basically a variable resistor that varies in resistance upon bending. Since the resistance is directly proportional to the amount of bending, it is often called a Flexible Potentiometer. A segmented conductor is placed on top to form a flexible potentiometer in which resistance changes upon deflection. Based on the degree of bend of this plastic piece, the resistance changes, affecting the current. This measurement is then used to detect hand movement changes. Flex sensors are designed to flex in only one direction – away from ink. Bending the sensor in another direction may damage it. In this project, we have attached the flex sensor to the finger of the glove. This flex sensor is used to control the movement of gripper.

## 4. DISCUSSION:

Robots are now an essential component of many industries' manufacturing processes because of the accuracy they provide and their capacity to work for prolonged hours without fatigue. Taking inspiration from this, our group has tried to create gesture controlled mechanical arm integrated with all the functionalities required to pick up an object and move it to another place. In the discussion below, we have tried to explain the flow and working of our project.

When the glove is moved either left or right, NRF will send the values of x1, which were detected by the gyro accelerometer module. There are two MPUs used- the first on the wrist, which controls the gripper's movement rotating in the x-axis in the wrist joint2. The second MPU near the wrist controls the motion of the shoulder joint (stepper motor) - up and down in layman's terms. The elbow joint is governed by the value of y2 obtained from the second MPU (near the elbow). It also controls the base with the value of x2, which turns the direction of the base. The wrist_joint_1 is controlled with the values of y1 that are obtained from the MPU1(on the palm). Overall, the whole structure can be moved in the forward and backward direction by making it change to the desired direction. Also, the gripper can be rotated, and the object can be lifted. Finally, the claw part, flex sensors transfer the values that control the movement of the claw used to grab the objects.

## 5. TECHNICAL PROBLEMS FACED IN THE PROJECT:

MASC is a small scaled project to show the ability to synergize the robots with the help of enhanced robotic assistance. Sailing through the journey, we faced many technical problems while we got our hands deep into the project.

1) ***To tinker with the brains by tinkering with the sensors***

- MASC-213 is an amalgamation of some of the widely used sensors like the MPU6050 Sensor (Accelerometer) and the Flex Sensor. One of the primary issues we faced was how to format the data coming from the **MPU6050** and **Flex Sensors** and how we can use that data to find out measures like velocity, orientation, acceleration, displacement, and other motion-like features.

- Integrating these sensors with the best suitable microcontroller was also a crucial aspect to touch upon, as different Arduino boards have different functionalities. We wanted to choose the one which is the best, most effective, and most optimal board to go on with. In our case, after several connections and debugging, the microcontroller turned out to be **Arduino Nano**.

- While we were making the glove for MASC, we found out that though every analog pin has the same use, some of the analog pins do not give us the desired output. This means that some of the analog pins in Nano had some other hidden functionalities, which we would like to explore.

2) ***WIRELESS COMMUNICATION***

- As amazing as it sounds, now that we got our 3 axial directional data all polished and ready to go. We wanted to send that data to be implemented on the arm. Here, we brainstormed to choose something that has a really strong baud rate, is very user-friendly, and is interactive and responsive to the environment (as it is an embedded system).

- We found the **NRF24L01 Transmitter Receiver Module** to be the best option for the same. This module has a 2.4 GHz frequency range with a data rate of up to 2Mbps, with a very high communication range of 800+m line of sight.

- Though it has very few connections, establishing a CONNECTION was quite a task. Receiving the data perfectly to the other side required a lot of critical thinking and research into the NRF module from our side. But, by setting the right channels and activating the required wires and registers, we were able to transmit errorless precise data going in at a very good speed.

**3) *SERVO AND STATES***

- As we established wireless communication between the glove and the controller. It was high time we actually used the data. As discussed earlier, our Arm has 2 states. First, in which, we control the gestures of the arm and second in which, we control the base of the arm for its movement in all directions. The toughest task for us was to first divide the states, and then divide the divided states into further more divisible states (sounds confusing right?) and conquer each part and assemble it all together.

**First State**

- The First State, which uses 6 MG995 series servo motors. They are attached to the 4 joints of the arm. While we were trying to figure out the motion of the servos, we faced a lot of hardware issues. Initially, we planned to use the PCA9685 Servo driver for motion of servos, but that wasn't the right choice as it had battery shortage and leaking issues, and it had a very limited range of motion which we didn't want at all.

- We wanted the perfect setup of the servos for each joint such that they coordinate effectively and do not limit themselves to only one of two values given; hence we went with **Arduino Mega.** With the help of Arduino Mega, we were able to provide sufficient batteries for the servos. At the same time, we were able to access the motions of each servo which were then closely touched upon individually, set up the range of each joint, coordinate joints wherever applicable, and then assemble the whole arm in one unit.

**Second State**

- When the switch in the glove is pressed, MASC gets into its second state, which is the motion of the whole arm. The arm can move in all directions with the help of STM32F070RB. We have used L298N motor drivers to move the base with the help of BO Motors, so it can hold the weight of the arm. We faced issues with the cross-platform communication of NRF with STM. Hence, we tried sending the data of NRF to STM using UART communication, but that too refrained to send the data. Finally, I2C Communication helped us to send the appropriate data.

## 6. TECHNOLOGIES USED:

***Arduino IDE*:** The open-source software known as the Arduino IDE is used to create and upload code to Arduino boards. The programming languages C and C++ are supported.

Sketching is a common term for writing a program or piece of code in the Arduino IDE. To upload the sketch created in the Arduino IDE software, we must connect the Genuino and Arduino board with the IDE. The sketch has the “. ino" file extension.

***STM32cube IDE*****:** Through a step-by-step procedure, STM32CubeMX is a graphical tool that enables very simple configuration of STM32 microcontrollers and microprocessors, as well as the generation of the appropriate initialization C code for the Arm® Cortex®-M core or a partial Linux® Device Tree for the Arm® Cortex®-A core.

***Blender*****:** A free and open-source 3D modeling program. In comparison to other 3D creation programs, it also has comparatively low memory and drive needs. Its interface makes use of OpenGL to deliver a uniform user experience on all systems and hardware that are supported.

***Keil*****:** The whole software development environment for a variety of Arm Cortex-M-based microcontroller chips is called Keil MDK. The Arm C/C++ compiler, the Vision IDE and debugger, and necessary middleware parts are all included in MDK. It is simple to use and covers all silicon vendors with more than 9,500 devices.

## 7. POSSIBLE IMPROVEMENTS:

Following are the factors where there is a scope of improvement for our project

- Servo requires external higher power. So instead, we could use a powerful servo that can power itself.
- Data communication between gloves and arms can be improved.
- There is a scope to display the weight of the object picked on the app we created.
- We had to use multiple batteries to power the whole structure. We could optimize here and may use a single battery for the whole project.
- Presently, the arm can only hold very little weight, by improving structure and parts, we could enable it to hold heavier weight.

## 8. FUTURE SCOPE:

Robotic Arms have a wide scope of development. Shortly the arms will be able to perform every task as humans and in a much better way. The system can work in the same way as a human arm. This system would make it easier for men to unrivaled the risk of handling suspicious and heavy objects which could be hazardous in their present environment and workplace. Complex duties would be achieved faster and more accurately with this design.

## 9. OUTCOMES:

"It is not the destination; it is the journey." But the journey matters only when you reach the desired destination. All the efforts and mistakes were justified when we finally got the desired working of our MASC Project. In the final outcome, we observed the following:

- Able to attain wireless communication to send data
- Able to coordinate servos on the basis of their angles
- Able to interactively change the real-time data and send it
- Able to format and send 3 axial data using MPU6050
- Able to define the motion of wheels to move the structure.
- Able to move base in forward and backward directions along with the clockwise and counterclockwise rotation.

## 10. CONCLUSION:

After hours of research, weeks of hard work and thousands of mistakes, MASC is finally working. It is righteously addressing the problem statement we first started with. Moreover, we are able to integrate the working of 4 microcontrollers, 6 servo motors, 4 BO DC Motors, 1 stepper motor, MPU6050 Sensor and Flex Sensor to make this project work. To conclude, MASC-213 is able to demonstrate the working of a multipurpose mechanical arm used to gesture control the process of placing goods or products.

## 11.REFERENCES:


*5017-009 - mouser electronics*. (n.d.). Retrieved December 6, 2022, from https://www.mouser.com/datasheet/2/30/5017-009-845352.pdf

*Controlling Nema 17 stepper motor with Arduino and A4988 Stepper driver module*. Circuit Digest - Electronics Engineering News, Latest Products, Articles and Projects. (n.d.). Retrieved December 7, 2022, from https://circuitdigest.com/microcontroller-projects/controlling-nema-17-stepper-motor-with-arduino-and-a4988-stepper-driver-module

*Electronic Components Distributor - Mouser Electronics*. (n.d.). Retrieved December 6, 2022, from https://www.mouser.com/datasheet/2/297/nRF24L01_Product_Specification_v2_0-9199.pdf

*Hand gesture controlled robotic arm using Arduino Nano*. Circuit Digest - Electronics Engineering News, Latest Products, Articles and Projects. (n.d.). Retrieved December 7, 2022, from https://circuitdigest.com/microcontroller-projects/diy-hand-gesture-controlled-robotic-arm-using-arduino-nano

Last Minute Engineers. (2022, October 29). *In-depth: Interface MPU6050 Accelerometer & Gyroscope sensor with Arduino*. Last Minute Engineers. Retrieved December 7, 2022, from https://lastminuteengineers.com/mpu6050-accel-gyro-arduino-tutorial/

Thingiverse.com. (n.d.). *EEZYbotARM mk2 by Daghizmo*. Thingiverse. Retrieved December 7, 2022, from https://www.thingiverse.com/thing:1454048

*NRF24L01 interfacing with Arduino: Wireless Communication*. Arduino Project Hub. (n.d.). Retrieved December 7, 2022, from https://create.arduino.cc/projecthub/muhammad-aqib/nrf24l01-interfacing-with-arduino-wireless-communication-0c13d4


## 12. CONTRIBUTION:

- **Kathan Bhavsar-** Glove of the arm
- **Dev Patel-** Glove of the arm
- **Maulik Ranadive-** Base of the arm
- **Pratham Lalwani-** Base of the arm

- **Manav Shah-** Servo Motor Integration
- **Meet Patel-** Base of the arm
- **Aditi Vasa-** Servo Motor Integration
- **Devyash Shah-** Servo Motor Integration
- **Vandan Shah-** Servo Motor Integration
- **Aastha Gaudani-** Glove of the arm

**CATEGORIZATION OF TEAMS:**

1. **Glove of the Arm:**
   Integration of Two MPU which generates 3 axial coordinate data with Arduino nano and flex sensors which are sent into the NRF24L01 Transmitter.

2. **Base of the Arm:**
   Established I2C communication between STM32 and Arduino Mega. Configured DC Motors by giving PWM Signals through Timer1.

3. **Servo Motor Integration:**
   Receiving the data sent from the glove into the NRF24L01 receiver and the initializing the ranges and gestures of the servo according to the joints.
   Base Joint
   Shoulder Joint
   Elbow Joint
   Wrist Joint (2)
   Gripper Joint