

# ● Report

**Team :**

**Yousef Mokhtar**

**Abdelkareem Hamdy**

**Ahmed Mohamed Fouly**

**Ali NourEldein**

**Basem zain**

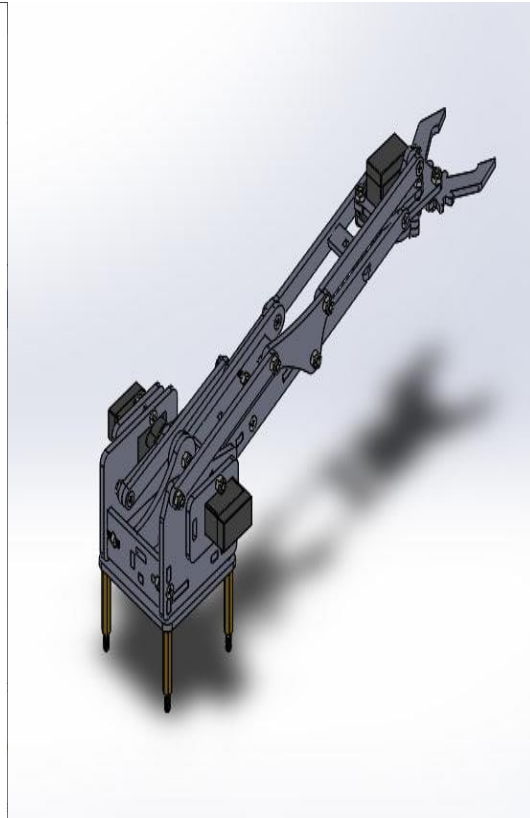
**Mostafa Mohsen saber**

**Ahmed Mostafa**

## • CAD Design

### Process:

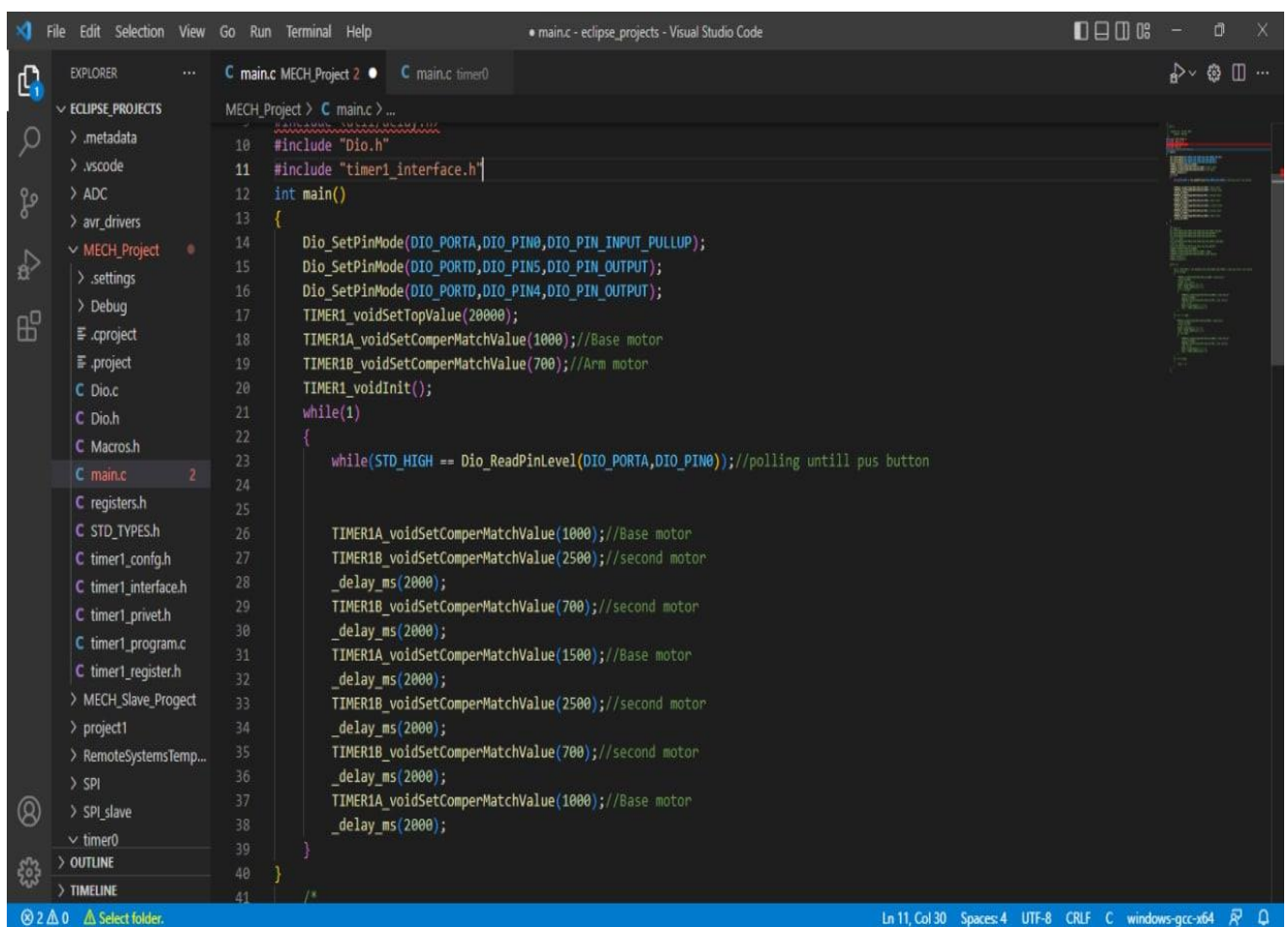
At this stage, we assembled the parts piece by piece and connected them together with screws.



# Embedded Code:

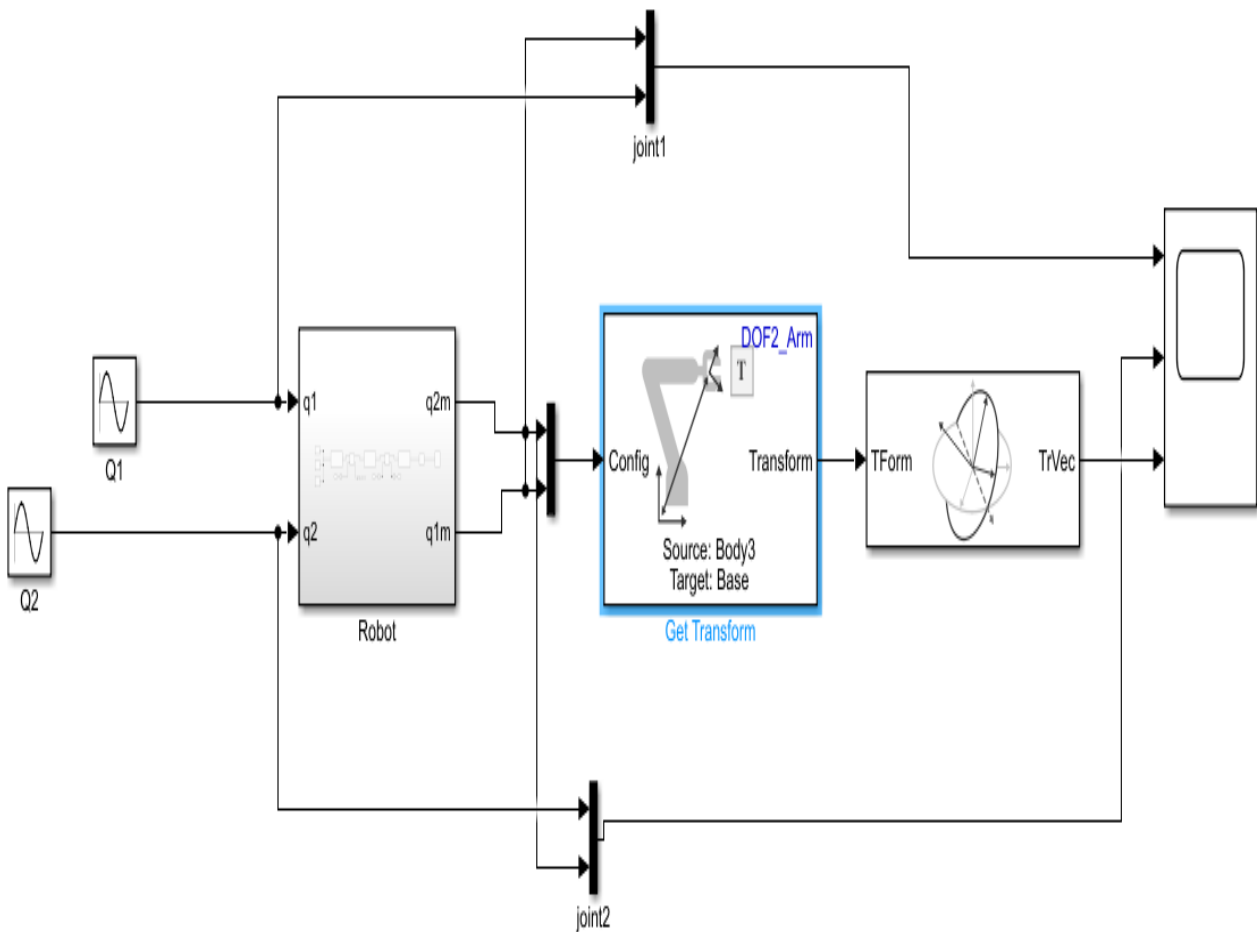
## Process

At this stage, we programmed a code using the C language to control the operating angles of the motor.

A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project structure with folders like .metadata, .vscode, ADC, avr\_drivers, and MECH\_Project. The MECH\_Project folder is expanded, showing files like .settings, Debug, .project, Dio.c, Dio.h, Macrosh, and main.c. The main.c file is selected and its content is displayed in the main editor area. The code is a C program for controlling a motor using timers and digital I/O. It includes headers for Dio.h and timer1\_interface.h. The main function sets up pin modes, initializes timers, and enters a while loop that polls a button and sets timer match values for two motors. The status bar at the bottom indicates the current line and column (Ln 11, Col 30) and the active language (C).

```
10 #include "Dio.h"
11 #include "timer1_interface.h"
12 int main()
13 {
14     Dio_SetPinMode(DIO_PORTA,DIO_PIN0,DIO_PIN_INPUT_PULLUP);
15     Dio_SetPinMode(DIO_PORTD,DIO_PIN5,DIO_PIN_OUTPUT);
16     Dio_SetPinMode(DIO_PORTD,DIO_PIN4,DIO_PIN_OUTPUT);
17     TIMER1_voidSetTopValue(20000);
18     TIMER1A_voidSetComperMatchValue(1000);//Base motor
19     TIMER1B_voidSetComperMatchValue(700);//Arm motor
20     TIMER1_voidInit();
21     while(1)
22     {
23         while(STD_HIGH == Dio_ReadPinLevel(DIO_PORTA,DIO_PIN0));//polling untill pus button
24
25
26         TIMER1A_voidSetComperMatchValue(1000);//Base motor
27         TIMER1B_voidSetComperMatchValue(2500);//second motor
28         _delay_ms(2000);
29         TIMER1B_voidSetComperMatchValue(700);//second motor
30         _delay_ms(2000);
31         TIMER1A_voidSetComperMatchValue(1500);//Base motor
32         _delay_ms(2000);
33         TIMER1B_voidSetComperMatchValue(2500);//second motor
34         _delay_ms(2000);
35         TIMER1B_voidSetComperMatchValue(700);//second motor
36         _delay_ms(2000);
37         TIMER1A_voidSetComperMatchValue(1000);//Base motor
38         _delay_ms(2000);
39     }
40 }
41 /*
```

# Matlab simulation Code



# Datasheet :

## Servo

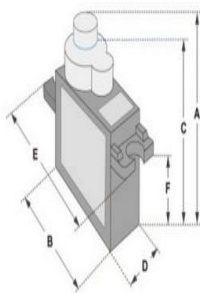
## Microcontroller

### SERVO MOTOR SG90

### DATA SHEET



Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.



#### Dimensions & Specifications

A (mm) : 32

B (mm) : 23

C (mm) : 28.5

D (mm) : 12

E (mm) : 32

F (mm) : 19.5

Speed (sec) : 0.1

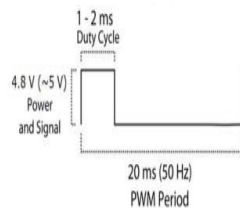
Torque (kg-cm) : 2.5

Weight (g) : 14.7

Voltage : 4.8 - 6

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "180" (~1ms pulse) is all the way to the left.

PWM=Orange (┐┐)  
Vcc=Red (+)  
Ground=Brown (-)



Block Diagram

Figure 2. Block Diagram

