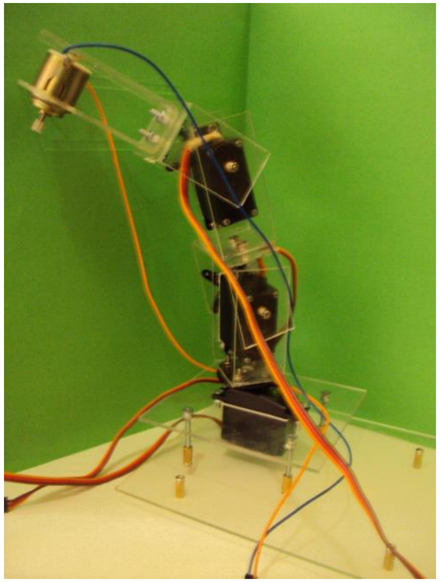


# Robotic Arm

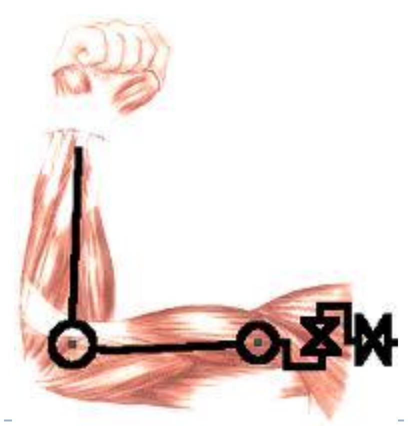
A project in COE Elective Robotics and Mechatronics

# Introduction

- The term robot comes from the Czech word ***robota***, generally translated as "forced labor." This describes the majority of robots fairly well. Most robots in the world are designed for heavy, repetitive manufacturing work. They handle tasks that are difficult, dangerous or boring to human beings.

# Introduction

- An industrial robot with six joints closely resembles a human arm -- it has the equivalent of a shoulder, an elbow and a wrist. Typically, the shoulder is mounted to a stationary base structure rather than to a movable body. This type of robot has six **degrees of freedom**, meaning it can pivot in six different ways. A human arm, by comparison, has seven degrees of freedom.



# Introduction

- Industrial robots are designed to do exactly the same thing, in a controlled environment, over and over again. To teach a robot how to do its job, the programmer guides the arm through the motions using a handheld controller. The robot stores the exact sequence of movements in its memory, and does it again and again every time a new unit comes down the assembly line.
- Most industrial robots work in auto assembly lines, putting cars together. Robots can do a lot of this work more efficiently than human beings because they are so precise.

# Introduction

# Statement of the Problem

- **Statement of the Problem**
- **1.1.1 General Problem**
- How to design and develop a robotic arm that can perform or simulate a task?
- 
- **1.1.2 Specific Problems**
- 1. How to design and develop the articulated jointed arm and base that will hold
- the end effector?

# Statement of the Problem

- 2. How to design and develop the end effector that can do a special task?
- 3. How to design and develop a circuit that can control the robotic arm's movement?
- 4. How to create a microcontroller program that can efficiently utilize the robotic arm?

# Project Rationale

**Project Rationale**
This project would be beneficial to:

- **Students**
- **Engineering Instructors**
- **Future Researcher**

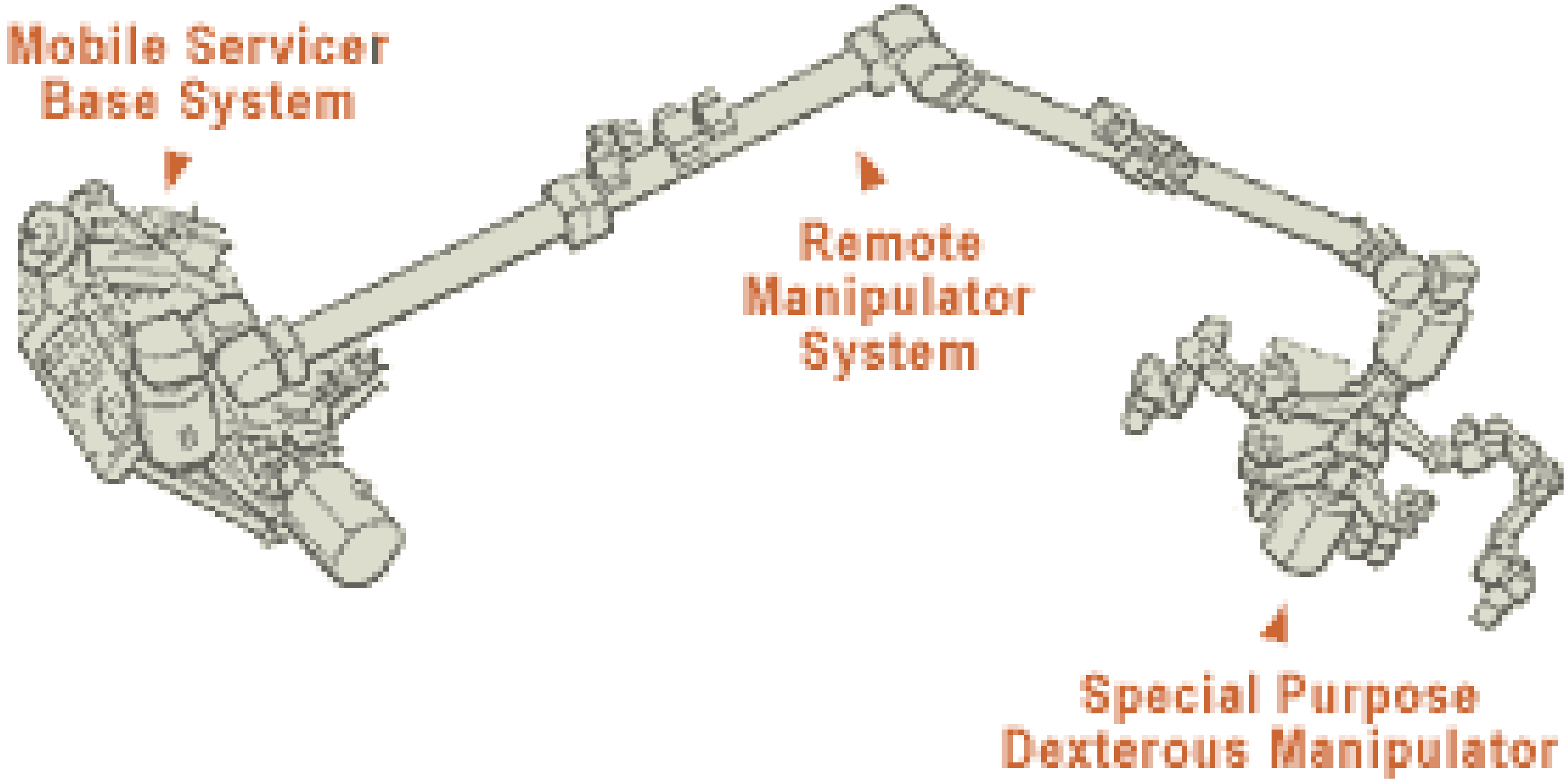
# Current State of Technology

- Robotic arms have been extensively used in the industry today. They can be found in medicine, space exploration, and commonly in manufacturing and production firms.

# Current State of Technology

- **Canadarm2**

  Launched on STS-100 (assembly flight 6A) in April 2001, the next generation Canadarm is a bigger, better, smarter version of the space shuttle's robotic arm. It is 17.6 meters (57.7 feet) long when fully extended and has seven motorized joints. This arm is capable of handling large payloads and assisting with docking the space shuttle. The Space Station Remote Manipulator System, or SSRMS, is self-relocatable with a Latching End Effector, so it can be attached to complementary ports spread throughout the station's exterior surfaces.



Mobile Servicer
Base System
Remote
Manipulator
System
Special Purpose
Dexterous Manipulator

# Current State of Technology

## Motoman UP6

- The Motoman UP6 is a high-speed robot that is compact and requires minimal installation space. Offering superior performance for welding and non-welding applications, its thin base and arm design allows for close placement for piece holding fixtures to improve part accessibility.

- The UP6 features collision avoidance that allows for two or three robots to work in conjunction with each other for increased productivity. The UP6 offers the widest work envelope in its class with a 1,373mm reach.

# Current State of Technology

# Current State of Technology

## Luke

- Robotic arms are also used by amputees. One of the latest advancements is “Luke” by Dean Kamen, which is an impressive, mind-controlled prosthetic robot arm he’s invented at D6 in Carlsbad. “Luke” is an incredibly sophisticated bit of engineering that’s lightyears ahead of the clamping "claws" that many amputees are forced to use today. The arm is fully articulated, giving the user the same degrees of movement as a natural arm, and is sensitive enough to pick up a piece of paper, a wineglass or even a grape without mishap.

# Current State of Technology

# Objectives

- **Objectives**

  - **General Objective**

This project aims to design and develop a robotic arm that can perform or simulate a task.

# Objectives

**Specific Objectives**

- Design and develop the articulated jointed arm and base that will hold the end effector.

  *The arm utilizes three servo motors to hold the end effector. The motors are connected by brackets.*

- Design and develop a cleaning tool as the end effector .

  *The arm utilizes a DC motor as a cleaning tool.*

`

# Objectives

- 3. Design and develop a circuit that can control the robotic arm's movement.

  *A circuit, with an MCU as its main chip, contains pushbuttons that can control the movement of each of the motor bracket clockwise and counter clockwise.*

- 4. How to create a microcontroller program that can efficiently utilize the robotic arm?

  *The MCU program efficiently sends pulses to the servo motor for every movement. The program is written and compiled using PICBASIC PRO.*

# Scope

- The project demonstrates controlling of mechanical devices, such as servo motors and DC motor
- The robotic arm provides control for the user through the push buttons
- The robotic arm simulates some industrial and commercial tasks, such as cleaning a car, shining and polishing a car, hooking light objects

# Scope

- The project employs the use of a PIC MCU that can be reprogrammed. The button functions and motor movement can be changed by reprogramming the MCU.

# Limitation

- The power of the robotic arm to lift is limited to the strength of the motors. The robotic arm may not carry heavy objects and may cause it to topple down.

- The robotic arm depends on a continuous DC voltage with high current capability, such as a wall outlet.

# Limitation

- The servo motor used is limited to a maximum of 180 degrees and is recommended to run it less than 180 to increase the motor’s life.

- The elevation and reach of the robotic arm is limited by the length of its brackets.

# Project Design

Robotic Arm

# Mechanical

- The servomotor bracket components are shown in Fig 3.1. Each of the fiber-glass U brackets that make up the assembly has multiple holes for connecting a servomotor horn as well as bottom and top holes for each servomotor bracket assembly consists of the following components: two fiber glass U brackets, labeled A and B, pivot assembly consisting of 1.5mm x 8mm screw with nut and gold binding head, four 3mm x 6mm screws with nuts, and two sheet metal screws for mounting the servomotor horn.

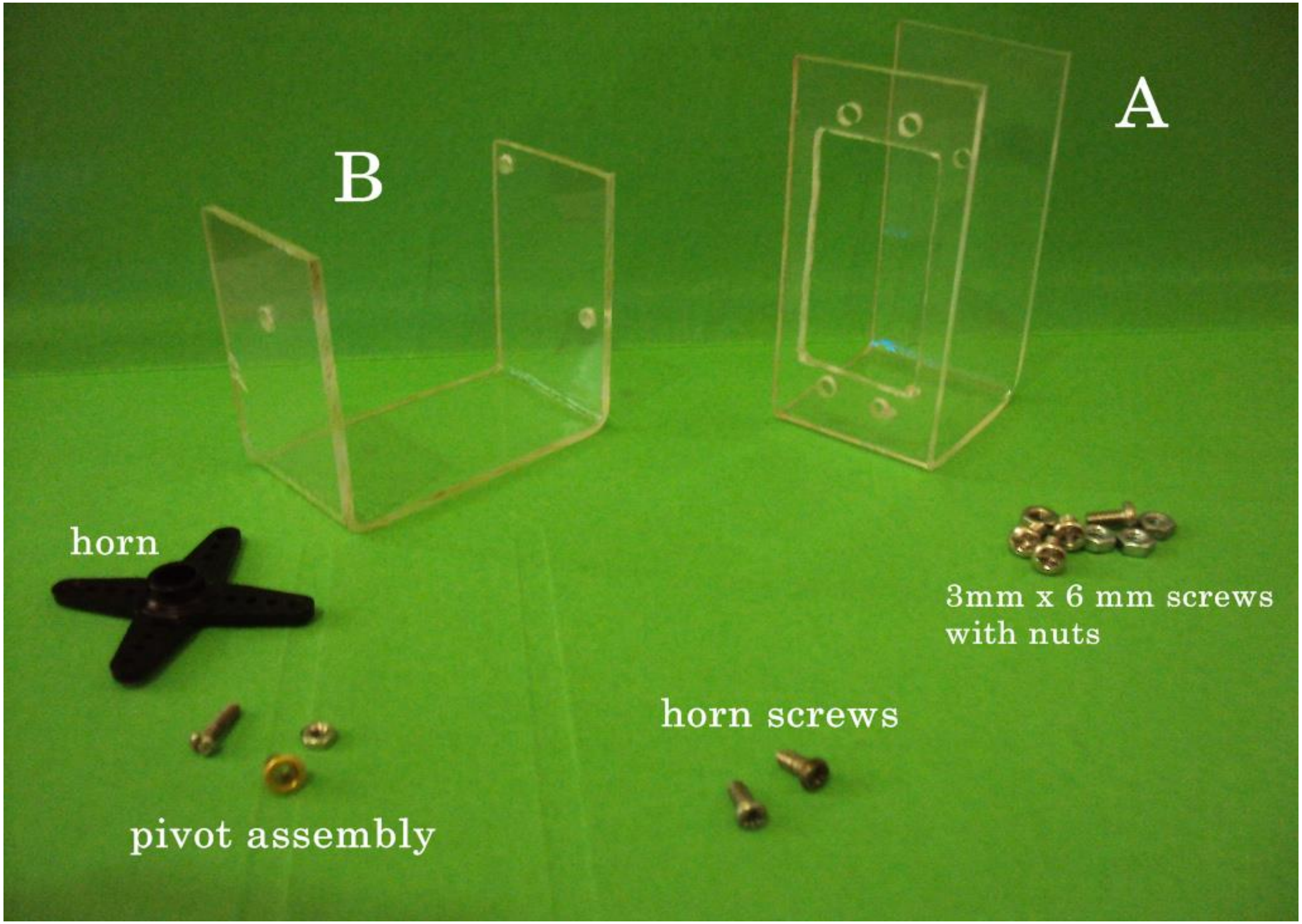



Figure 3.1 **Servomotor bracket kit**

# Mechanical

- When assembled with the servomotor (see Fig.3.2), the bracket becomes a modular component that may be attached to other brackets and components. The bracket allows the top and bottom components to swivel along the axis of the servomotor’s shaft (see Fig. 3.3).

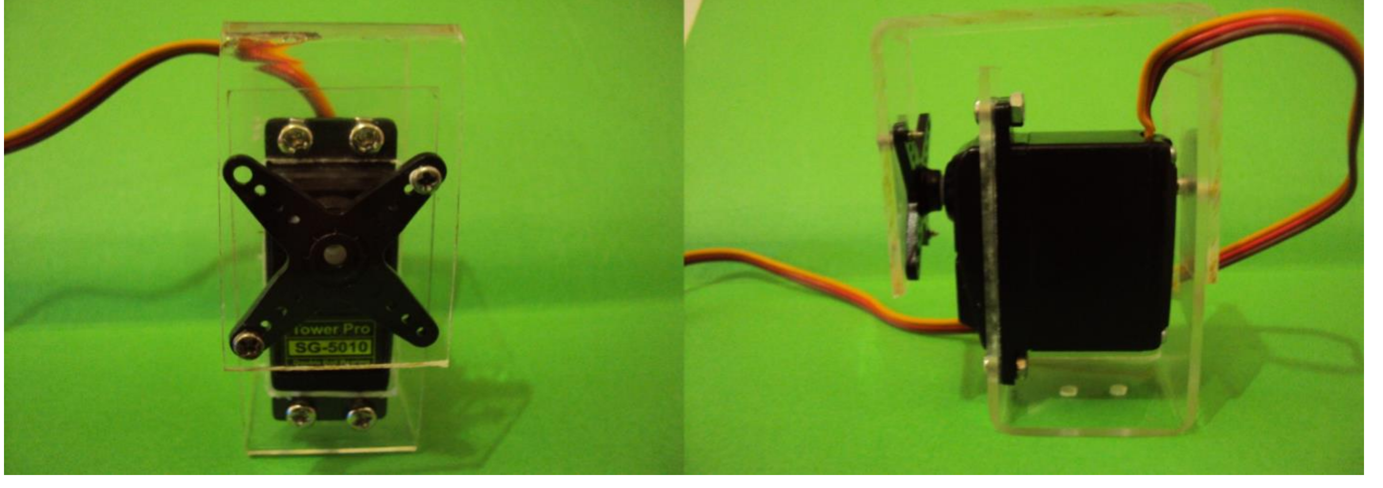



Figure 3.2 **Front and side views of servomotor bracket**

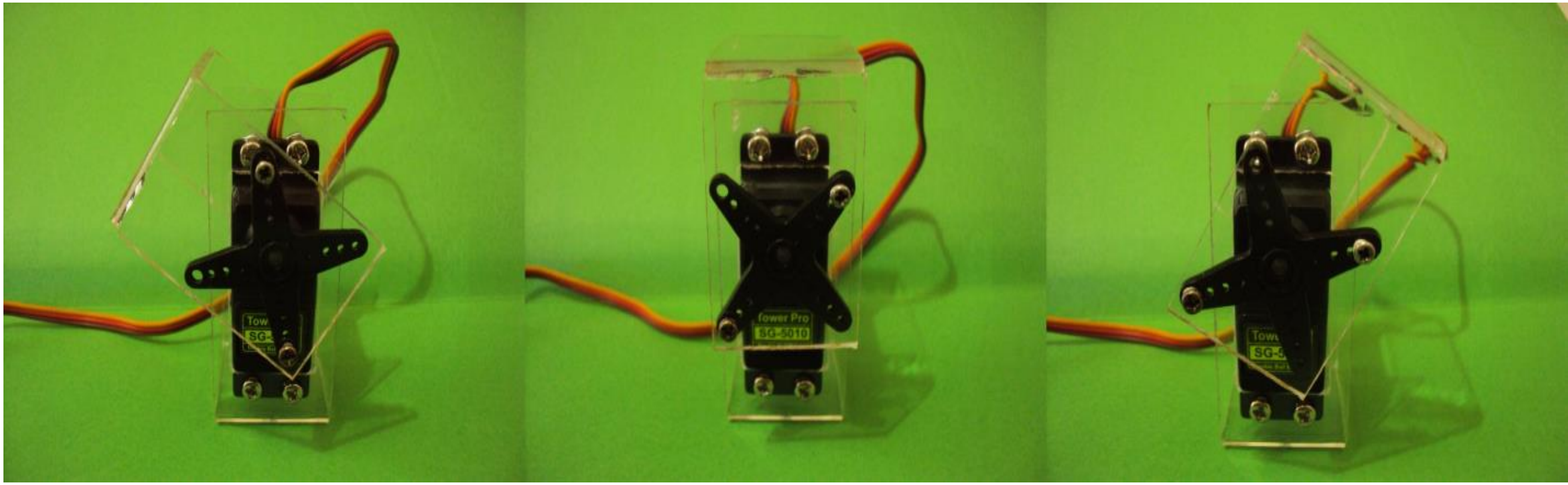



Figure 3.3 **Servomotor bracket travel (tilt left, center, tilt right)**

# Mechanical

- The individual servomotor bracket kit must be assembled first. The servomotor is placed into bracket A by securing the four screws with nuts(Fig.3.5). The pivot assembly screw must also be fixed (Fig. 3.4) and the servo horn must be fixed into bracket B (Fig.3.6) using the horn screws. Brackets A and B must be joined as shown in Fig 3.7.

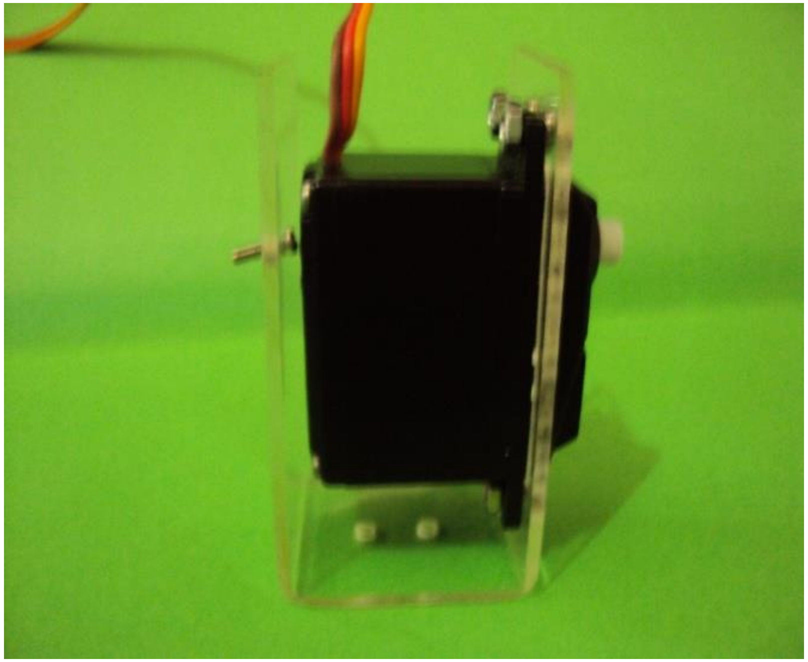


Figure 3.4 **Side view of placing servo in A Bracket**

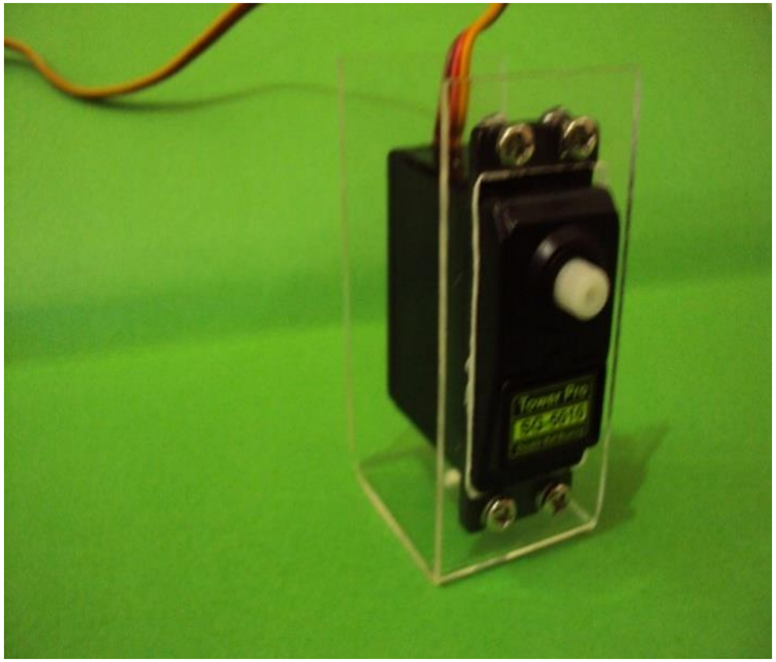


Figure 3.5 **A bracket with servo attached with screws and nuts**

Figure 3.6 **Servomotor horn attached in B bracket**

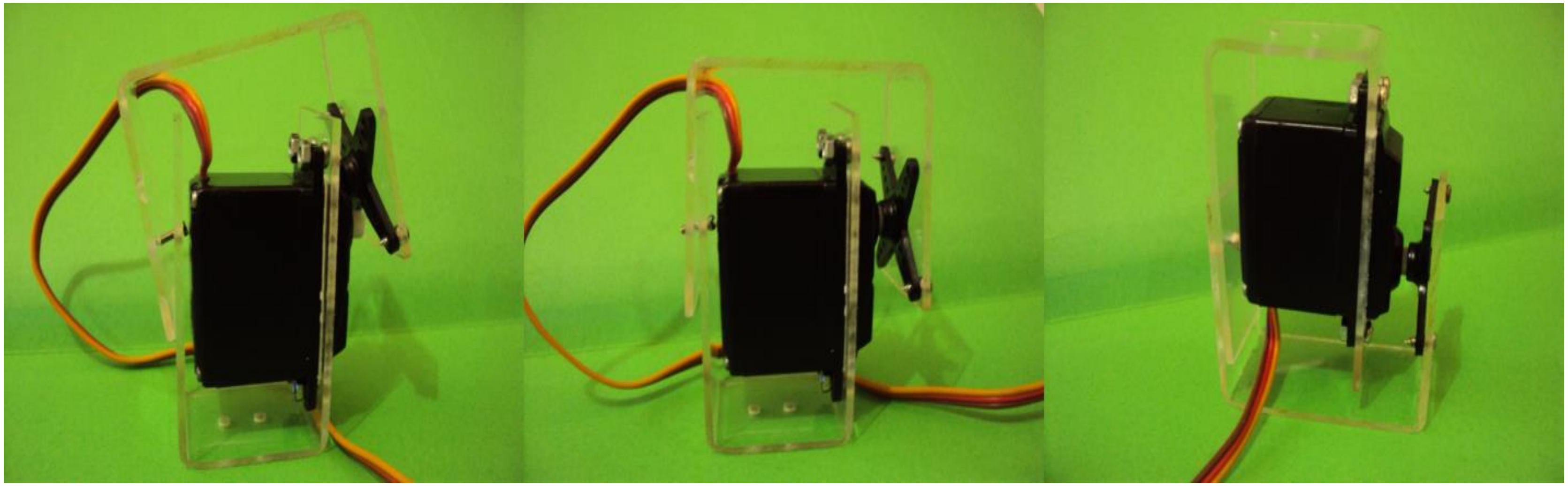


Figure 3.7 **Joining Bracket A and B, Horn assembly connected to servo**

# Mechanical

- With the design using multiple servomotor assemblies, it is essential to preplan how the servomotors will be connected. When two or more servomotors assemblies are connected, the connecting brackets of the joints should be pre-assembled. The DC motor bracket is then fixed using the screws in Fig 3.8 to become the one shown in Figures 3.9 and 3.10. The assembled forearm is then fixed to the bottom rotating servomotor as shown in Fig. 3.11. The complete robotic arm is shown in Figures 3.12


Figure 3.8 **Screws used in joining DC motor bracket to Bracket B of servo**

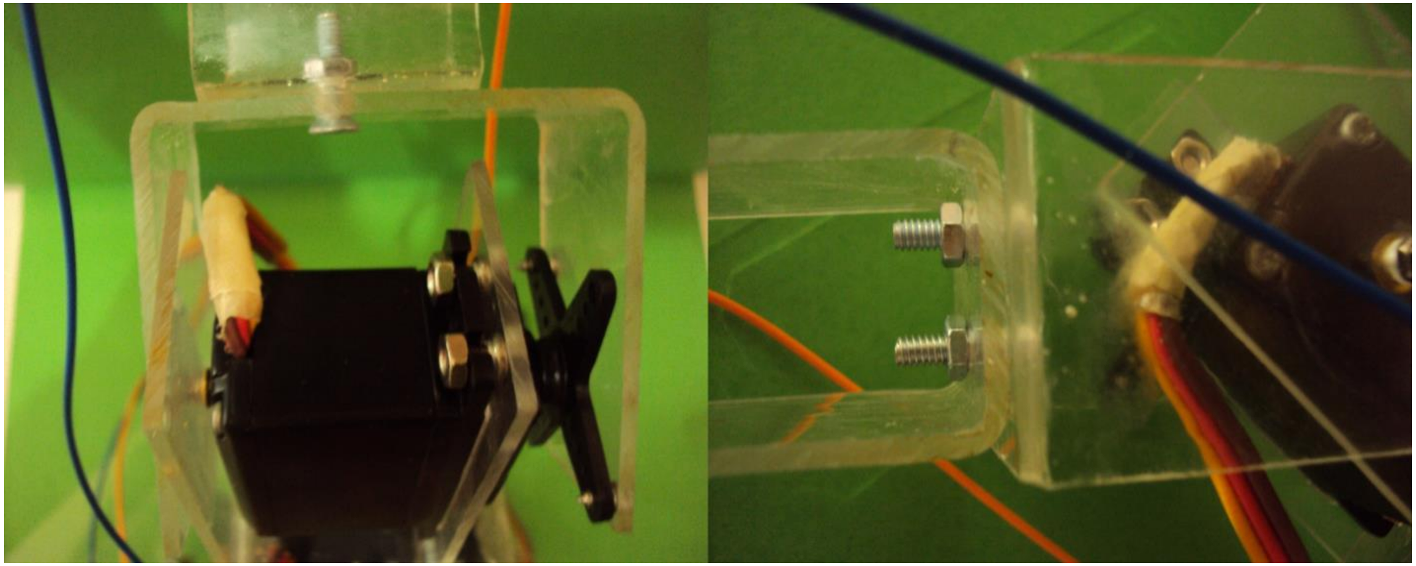


Figure 3.9 **DC motor bracket  connected to Bracket B of servo**

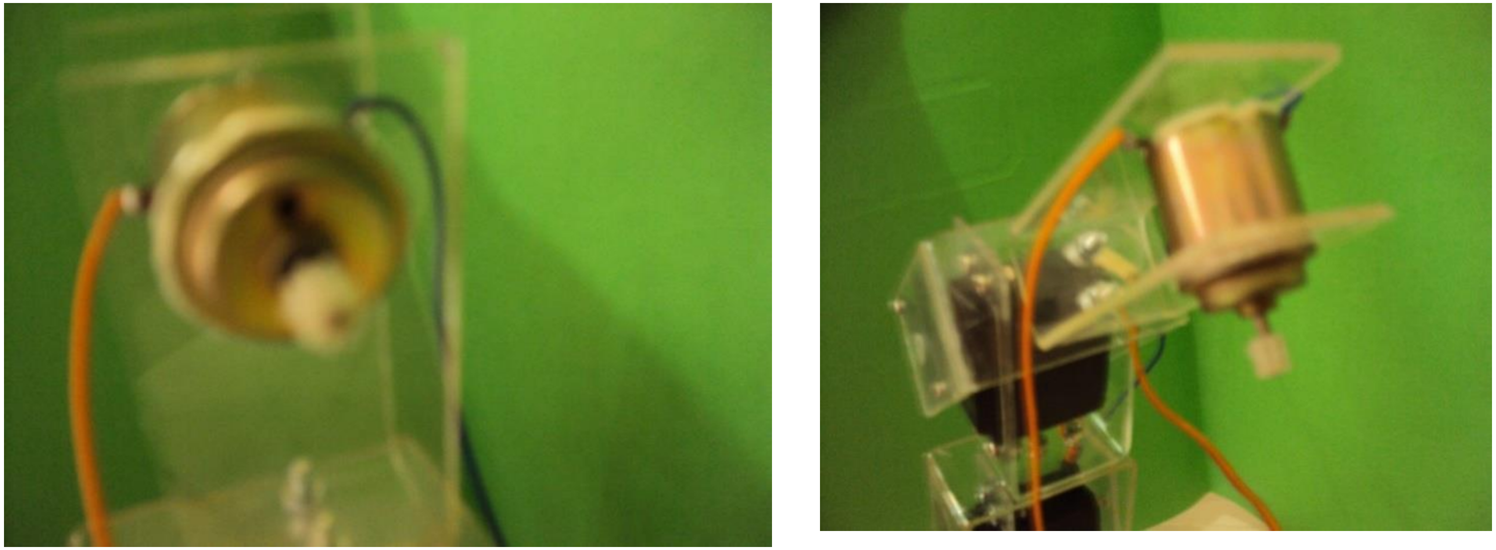


Figure 3.10 **DC motor bracket**

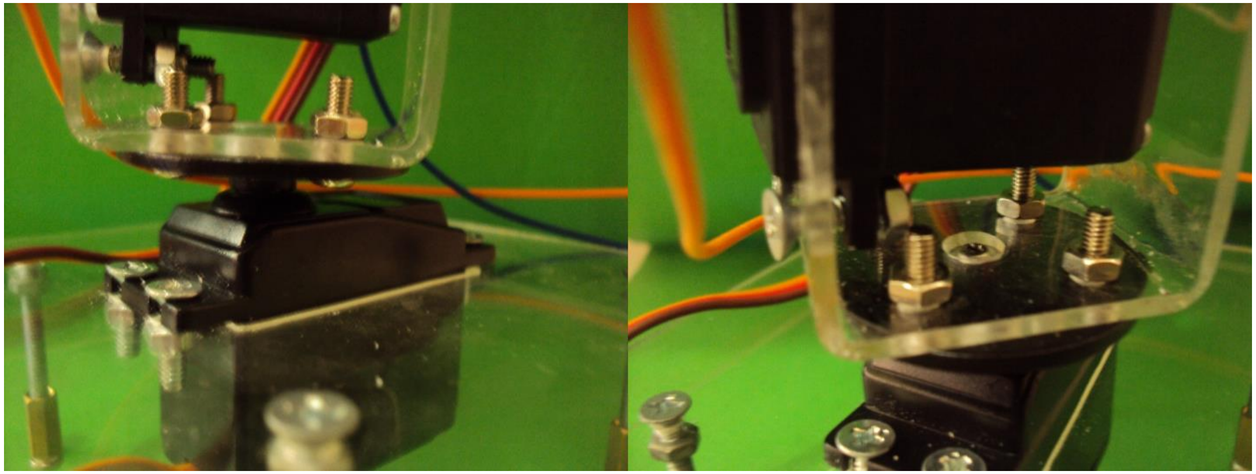


Figure 3.11 **Assembly of bottom servo to bracket A of middle servo**

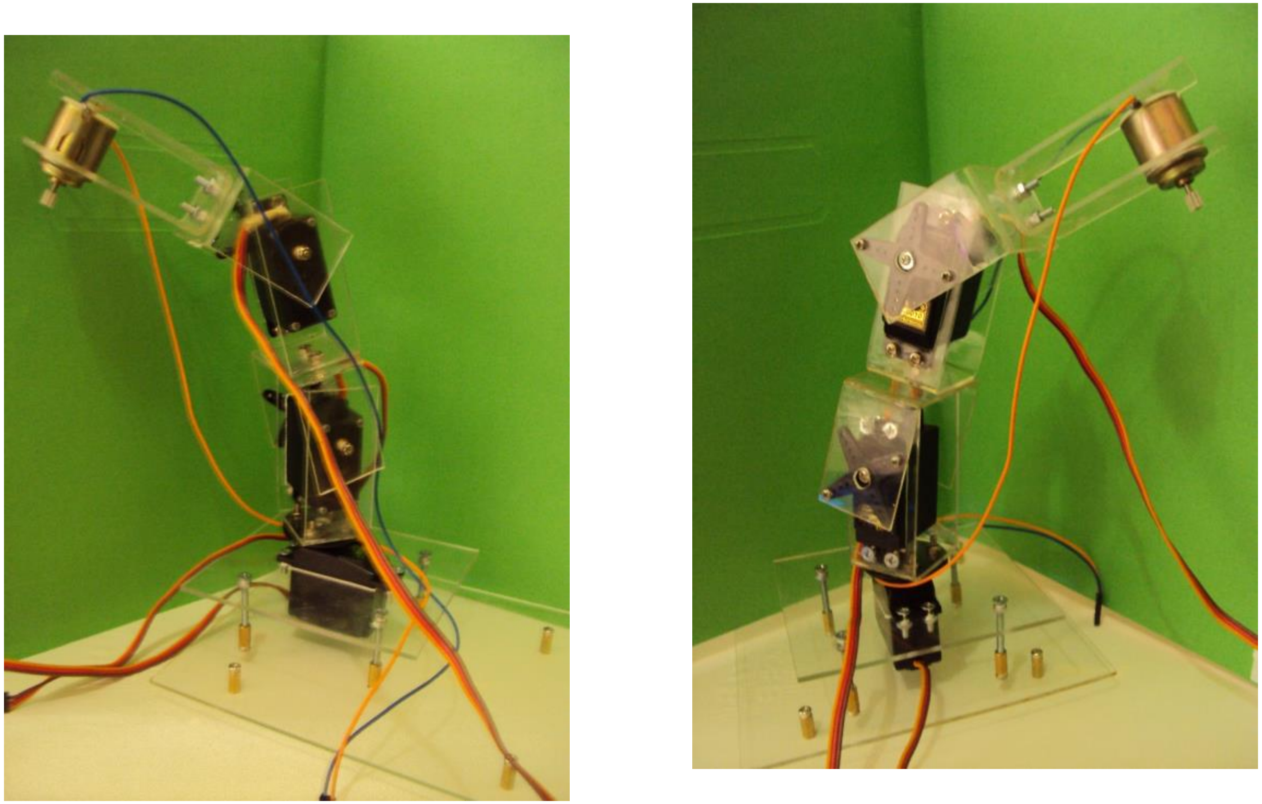


Figure 3.12 **4-motor Robotic Arm (3 servos, 1 DC motor)**

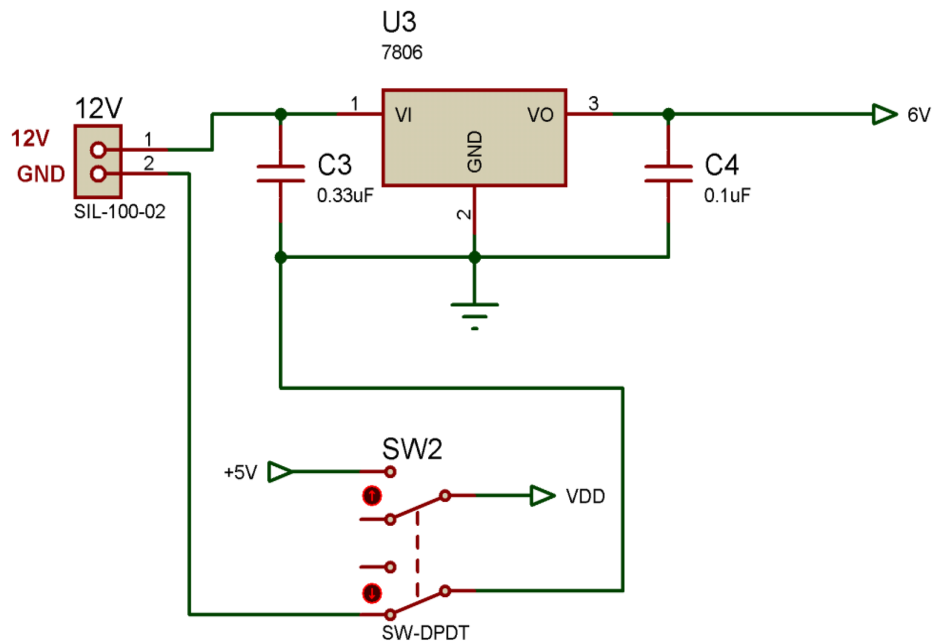
# Electronics

The project circuit includes several power supplies:

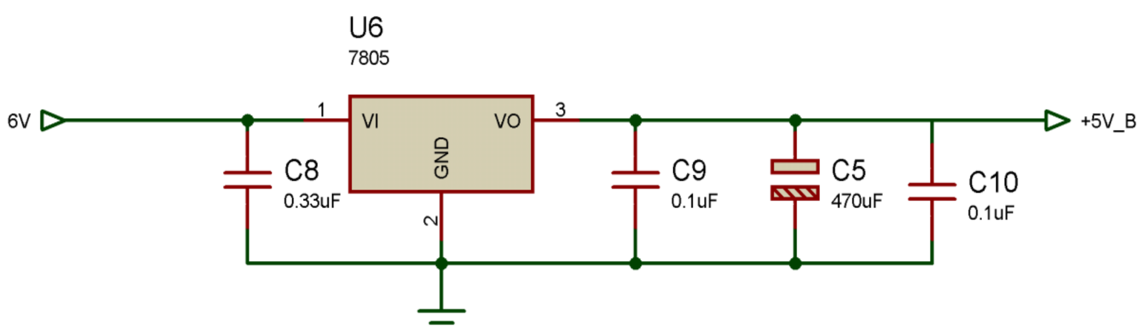
- +12V to +6V => 6V to power ServoMotors
  - Using 7806 voltage regulator
- +6V to +5V A => 5V for PIC MCU and pushbutton pull-ups
  - Using 7805 voltage regulator
- +6V to +5V B => 5V for DC Motor Driver L293D
  - Using 7805 voltage regulator
- +5V B to +3V => 3V for DC Motor
  - Using LM317 adjustable voltage regulator

- An overall circuit switch, a SPDT (Single Pole Double Throw) slide switch is also added that shorts or breaks the VDD +5V A supply for the PIC MCU and the GND from the main power supply. The main power supply is a 220V AC to 3V-12V DC selectable supply.

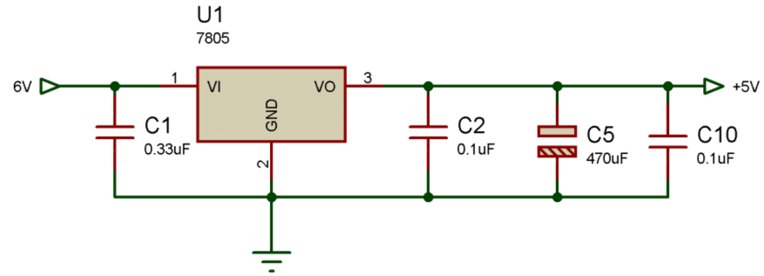
## +12V to +6V supply



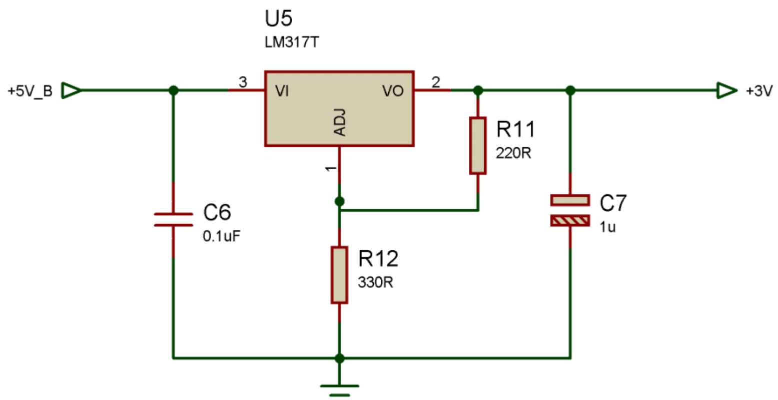

## +6V to +5V supply B



## +6V to +5V supply A



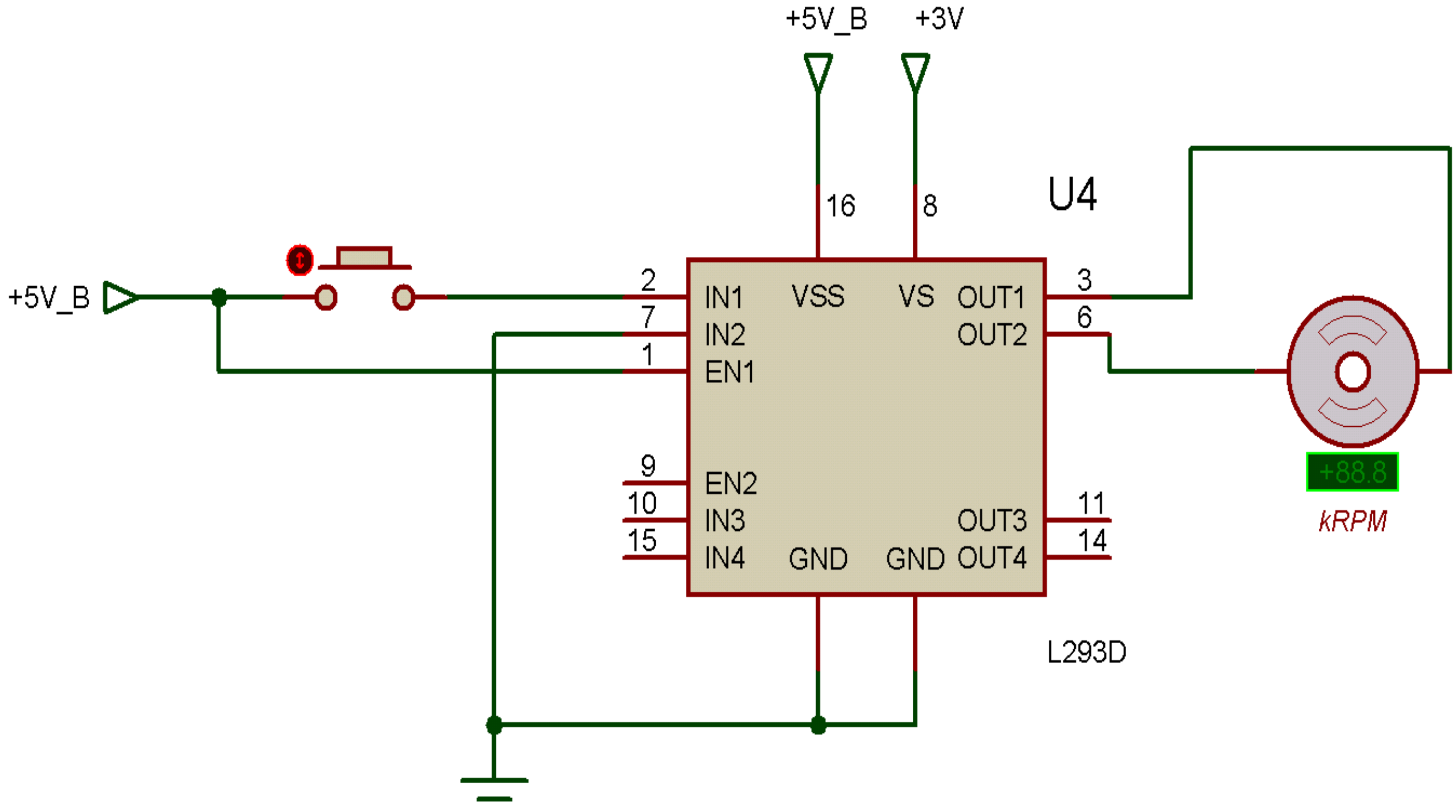

## +5V to +3V converter



# Electronics

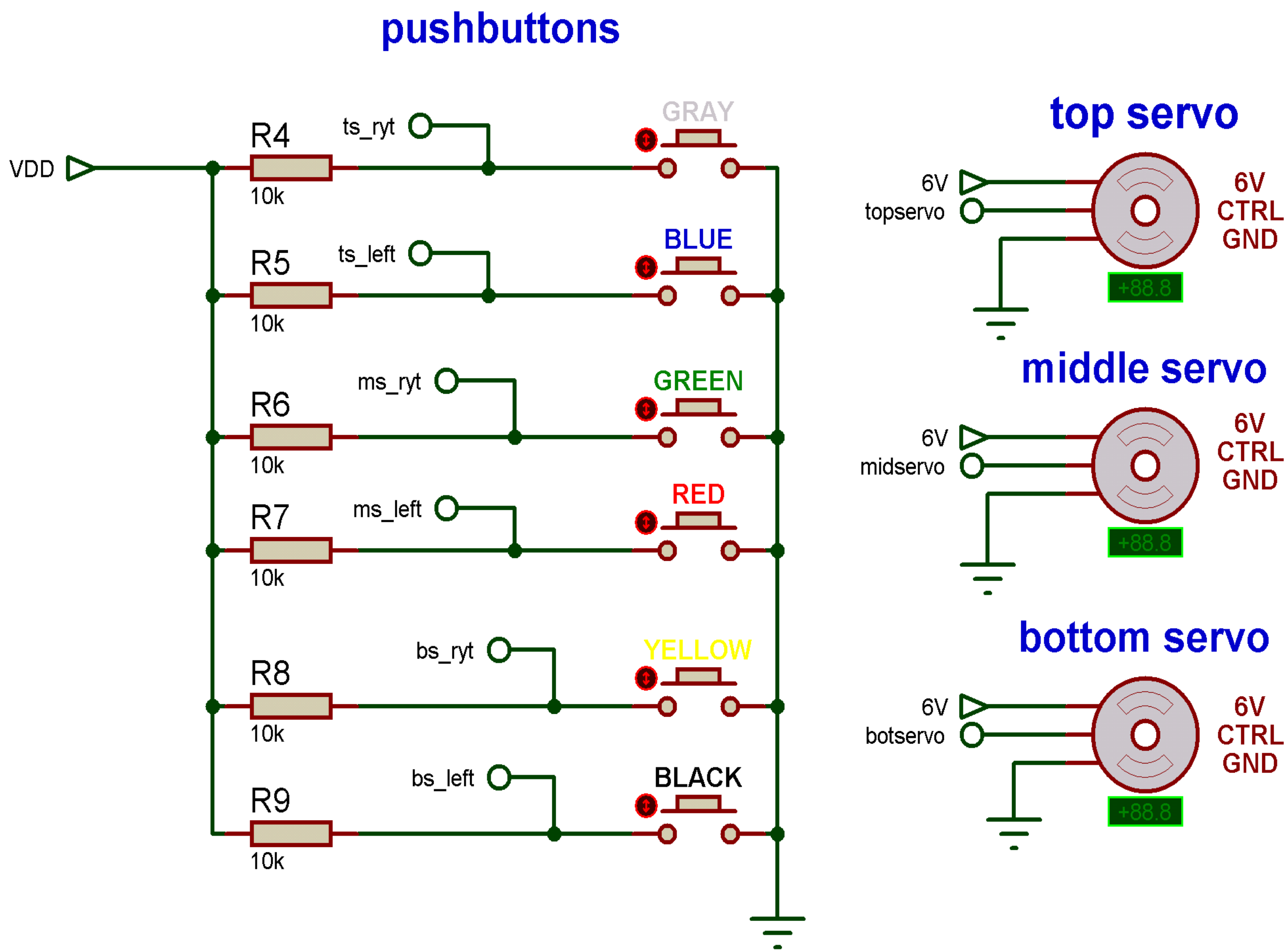
- A separate 5V supply is provided for the DC Motor circuit to reduce the electrical noise produced by the DC Motor. The driver used is L293D quad H-bridge that handles rotation of the motor and reduces electrical noise. 5V is supplied to the IC and the 3V is tapped as input for the motor. 1 pin of the motor is pulled to GND already and the positive supply remains open until shorted out by the pushbutton.

# DC Motor Control



# Electronics

- The user control circuitry consists of 7 pushbuttons (SPST momentary Normally Open push button) which are pulled high through a 10k resistor. All of the servo switches are connected to the input pins of the PIC MCU.
  - 1 switch for the momentary on/off  of DC motor actuator (SPST momentary NO -normally open- push button)
  - 2 switches for Clockwise and Counterclockwise rotation of Top Servo
  - 2 switches for Clockwise and Counterclockwise rotation of Mid Servo
  - 2 switches for Clockwise and Counterclockwise rotation of Bottom Servo

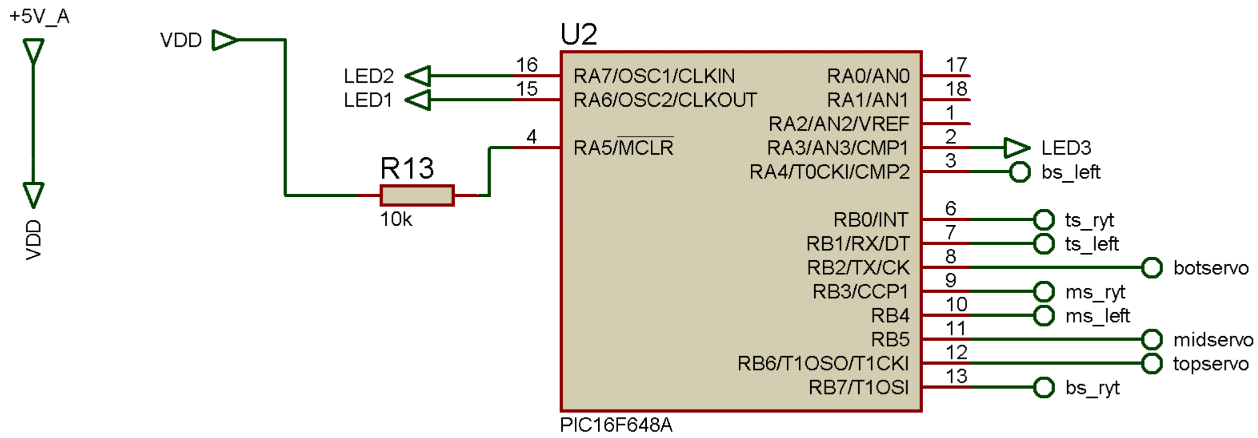


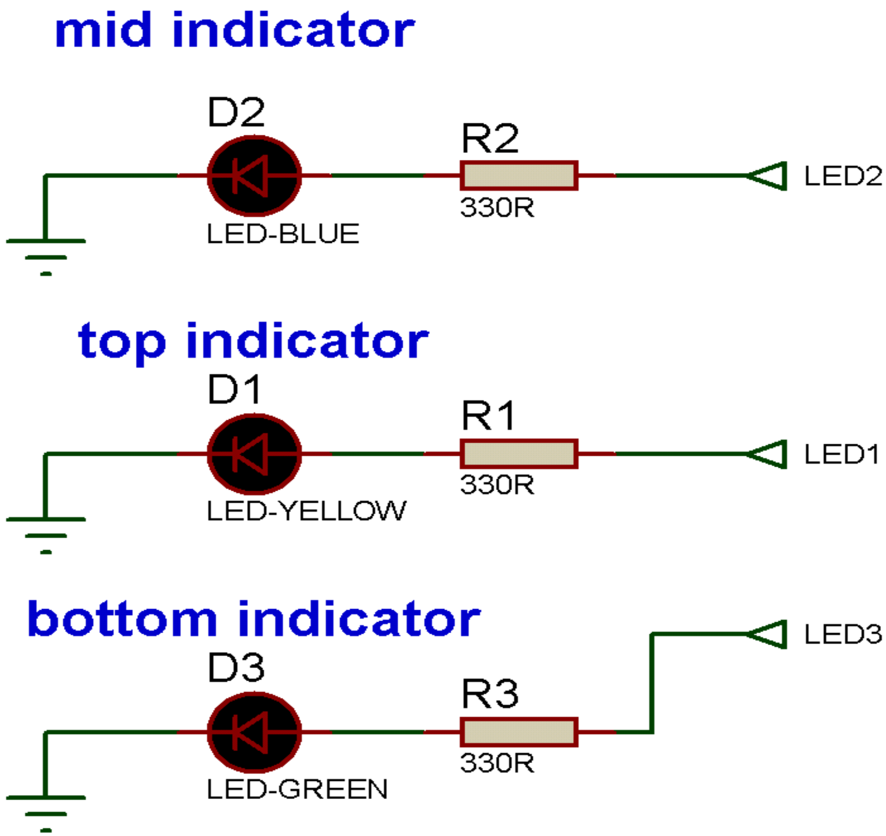
pushbuttons
VDD
R4
10k
ts_ryt
GRAY
R5
10k
ts_left
BLUE
R6
10k
ms_ryt
GREEN
R7
10k
ms_left
RED
R8
10k
bs_ryt
YELLOW
R9
10k
bs_left
BLACK
top servo
6V
topservo
6V
CTRL
GND
+88.8
middle servo
6V
midservo
6V
CTRL
GND
+88.8
bottom servo
6V
botservo
6V
CTRL
GND
+88.8

# Electronics

- The PIC MCU used is an 18-pin PIC MCU 16F819. It has a total of 15 I/O pins. 3 of the I/O pins outputs the pulse to the 3 servos. Three LEDs indicate which servo is activated.

Main Control (PIC MCU)




mid indicator
D2
R2
LED2
330R
LED-BLUE
top indicator
D1
R1
LED1
330R
LED-YELLOW
bottom indicator
LED3
D3
R3
330R
LED-GREEN

# Servo Motors and Pulse Width Modulation

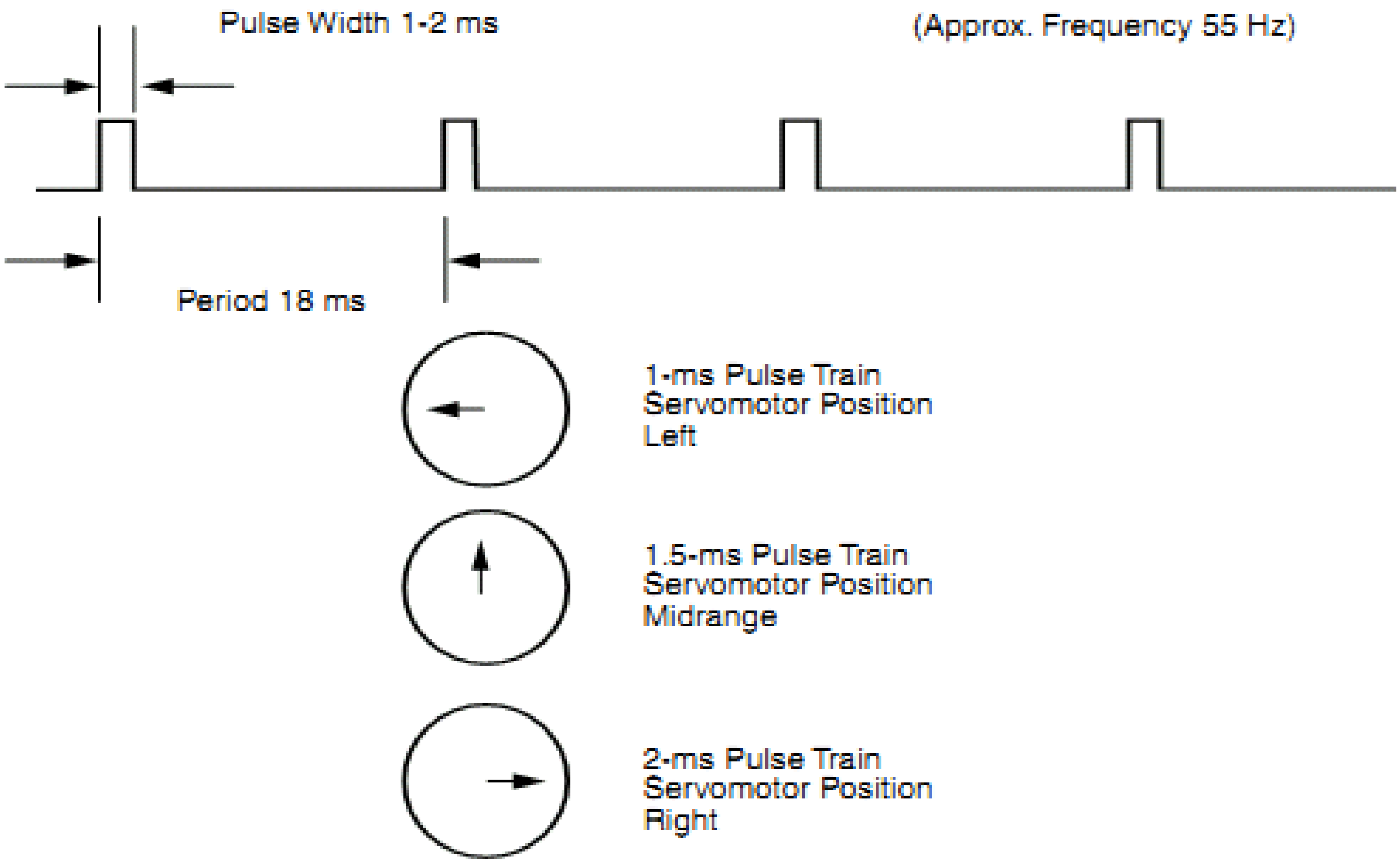
- Servomotors are geared dc motors with a positional feedback control that allows the shaft (rotor) to be rotated and positioned accurately. When a control signal is being fed to the servomotor, the servomotor’s shaft rotates to the position specified by the control signal. The positioning control is a dynamic feedback loop, meaning that if you forcibly rotate the servomotor’s shaft way from its control signal command position, the servomotor circuitry will read this as a position error and will increase its torque in an attempt to rotate the shaft back to its command position.

# Servo Motors and Pulse Width Modulation

- There are three wire leads to a hobby servomotor. Two leads are for power 15 V (red wire) and ground (black wire). The third lead (yellow or white wire) feeds a position control signal to the motor.

- The position control signal is a single variablewidth pulse. The pulse width typically varies between 1 and 2 ms. The width of the pulse controls the position of the servomotor shaft. Figure 12.26 illustrates the relationship of pulse width to servomotor position. A 1ms pulse rotates the shaft to the extreme counterclockwise (CCW) position (-45°). A 1.5ms pulse places the shaft in a neutral midpoint position (0°). A 2ms pulse rotates the shaft to the extreme CW position (+45°). The pulse width signal is sent to the servomotor approximately 55 times per second (55 Hz).

# Servo Motors and Pulse Width Modulation



# Programming

The servomotor controllers use the PicBasic Pro *puls out* command. The command format is as follows:

**pulsout pin, period**

The pulsout command generates a pulse on the pin specified for the period of time specified. The time is in 10us (microsecond) increments. So to send a 1.5 ms pulse out on port B pin 0, you could use one of the following command for the PicBasic Pro compiler:

**pulsout portb.0, 150**

# Programming

The angles of each servo is declared using its pulse counterpart

'============ PULSE widths for ANGLE ranges =============

'50 far left

'100 left

'150 center

'200 right

'250 far right

'min - lower limit of the angle pulse

'max - higher limit of the angle pulse

'the higher the range, the larger angle

'TOP

topmin CON **120**

topmax con **200**

'MID

midmin CON **120**

midmax con **180**

'BOTTOM

botmin CON **80**

botmax con **220**

# Programming

Each of the buttons are polled individually using an IF-END IF method.

```
'==================== TOP servo control ========================

IF PORTB.0 = 0 then              'TOP-right switch [gray]

    PORTA.6 = 1           'LED1 on
    PORTA.7 = 0           'LED2 off
    PORTA.3 = 0           'LED3 off

    'decrease pulse until it equals the lower limit pulse

    IF pulseWidth1 <= topmin Then
        pulseWidth1 = topmin
    Else
      pulseWidth1 = pulseWidth1 - 1
    Endif

ENDIF
```

# Hardware Resources

Robotic Arm

# Servomotor

TowerPro MG-995
Digital Servo

**Stall Torque:**
9.0kg/cm @ 4.8V
11kg/cm @ 6.0V
**Speed:**
0.17 seconds/60deg. @ 4.8V
0.13 seconds/60deg. @ 6.0V
**Dimensions:**
40.6 x 19.8 x 37.8mm
**Temperature Range:**
0c - 55c
**Operating Voltage:**
3.5V - 7.2V
**Weight:**
56grams

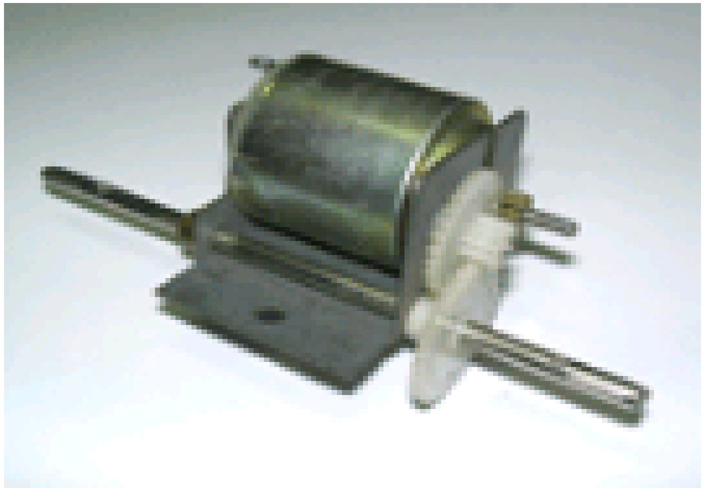
Metal Gear, Ball Bearing
**PHP 890.00**

TowerPro SG-5010

**Stall Torque:**
5.2kg/cm @ 4.8V 6.4kg/cm @ 6.0V
**Speed:**
0.2 seconds/60deg. @ 4.8V
or 0.16 seconds/60deg. @ 6.0V
**Dimensions:**
41mm x 20mm x 38mm
**Temperature Range:**
0c -55c
**Operating Voltage:**
4.8V - 6.0V
**Weight:**
41grams

Ball Bearing
**PHP 590.00**

# DC motor



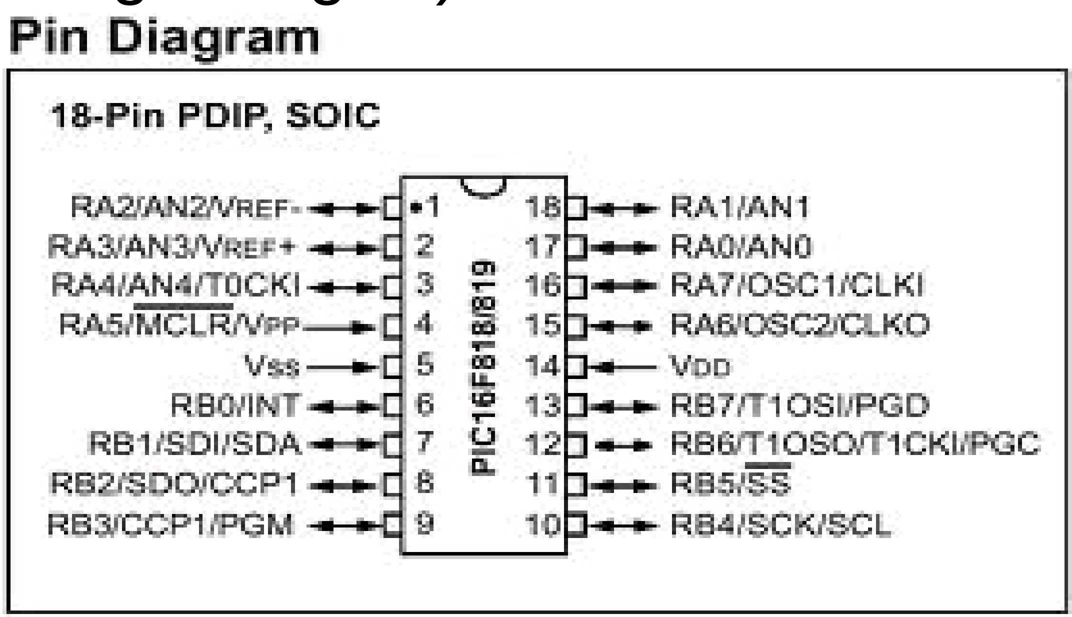
77010775

3V Drive assembly (used)

P25.00

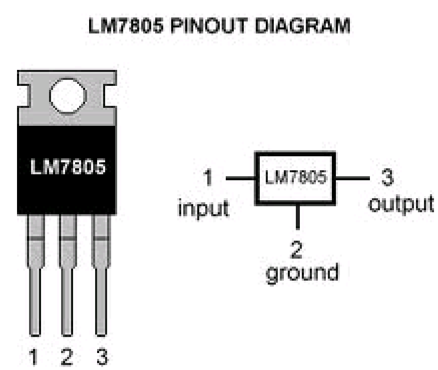
# PIC 16F819

- This powerful CMOS Flash-based 8-bit microcontroller packs Microchip's powerful PIC® architecture into an 18-pin package The PIC16F819 features 8MHz internal oscillator, 15 I/O pins, and ADC (analog-to-digital)



# 78xx 3-terminal Positive Voltage Regulators

- The 78xx (sometimes LM78xx) is a family of self-contained fixed linear voltage regulator integrated circuits. The 78xx family is commonly used in electronic circuits requiring a regulated power supply due to their ease-of-use and low cost.



# U-bracket

- The U brackets are fabricated to fit the servo motor and prepare the motor for assembly with other brackets.

# Software

**PIC BASIC PRO**

- The PICBASIC PRO™ Compiler (or PBP) makes it even quicker and easier for you to program Microchip Technology’s powerful PIC® microcontrollers (MCUs). The English-like BASIC language is much easier to read and write than assembly language.

# Software

**WinPic800**

- This free software serves as the burner that transfers the compiled HEX codes from the host PC through the MCU Programming device into the microcontroller. It allows you to program all types of serial PICs using Windows 95/98/NT/2000/ME/XP. It supports many PIC Microcontroller families and 24Cxx EEPROMs.

# Software

## Proteus PCB Design Package

- Proteus PCB design combines the ISIS schematic capture and ARES PCB layout programs to provide a powerful, integrated and easy to use suite of tools for professional PCB Design. It includes an integrated shape based autorouter and a basic SPICE simulation capability as standard.
- The developers use Proteus ISIS and ARES to schematic designs for the project hardware.

# Bill of Materials

| Materials | Quantity | Unit Price | Total Price |
|---|---|---|---|
| | | | |
| SG-5010 Servo Motor | 2 | P 590.00 | P 1180.00 |
| MG-995 Servo Motor | 1 | 890.00 | 890.00 |
| DC Motor | 1 | 25.00 | 25.00 |
| Fiber Glass | 1 | 30.00 | 30.00 |
| Resistors | 16 | 0.25 | 4.00 |
| Electrolytic Capacitors | 4 | 2 | 8.00 |
| Nonpolar Capacitors | 3 | 0.5 | 1.50 |
| 7805 | 2 | 5 | 10.00 |
| 7806 | 1 | 5 | 5.00 |
| L293D | 1 | 120 | 120.00 |
| LM317 | 1 | 5 | 5.00 |
| LED | 3 | 3 | 9.00 |
| 16F819 | 1 | 150 | 150.00 |
| 2- Pin Header | 2 | 2 | 4.00 |
| DPDT slide switch | 1 | 10 | 10.00 |
| Push Button | 7 | 10 | 70.00 |
| 3-pin cable | 1 | 15 | 15.00 |
| Wires (per m) | 3 | 5 | 15.00 |
| Screws | | | 40.00 |
| Travel Expenses | | | 500.00 |
| TOTAL | | | **P3091.50** |

# Demonstration