**Arduino Source Code**

//Quartz Mini Robotic ARM - Testing code

//Joystick 1 and 2 is connected to A0,A1, A2 and A3

//SERVO CONNECTIONS

//Neck sero - D10 - pos is saved at EEPROM 3

//Front and Back - D11 - pos is saved at EEPROM 2

//UP and Down - D9 - pos is saved at EEPROM 1

//Gripper - D6 - pos is saved at EEPROM 0

//created by: Aswinth Raj

//created for: quartzcomponents.com

#include <Servo.h>

#include <EEPROM.h>

//All the gripper positions will be saved and read from eeprom to resume same positon

int gripper\_pos = EEPROM.read(0);

int updown\_pos = EEPROM.read(1);

int frontback\_pos = EEPROM.read(2);

int neck\_pos = EEPROM.read(3);

// Create a servo object - one each for 4 servos

Servo Gripper\_servo;

Servo UpDown\_servo;

Servo FrontBack\_servo;

Servo Neck\_servo;

//Function the control the servo based on joystick position

void control\_servo (Servo &current\_servo, int current\_pos, int EEPROM\_addr)

{

    //positon should alwasy be between 0 to 180

    if (current\_pos>=180)

    current\_pos=175; //jitter at maximum limit

    if (current\_pos<=0)

    current\_pos=10; //jitter at minimum limit

    current\_servo.write(current\_pos); //update servo postion

    EEPROM.write(EEPROM\_addr, current\_pos); //save position in EEPROM

    Serial.print(EEPROM\_addr); Serial.print(" = "); Serial.println(current\_pos); //for debugging on serial monitor

}

void setup() {

  Serial.begin (9600);

  Gripper\_servo.attach(6);

  Gripper\_servo.write(gripper\_pos);

  UpDown\_servo.attach(9);

  UpDown\_servo.write(gripper\_pos);

  FrontBack\_servo.attach(11);

  FrontBack\_servo.write(gripper\_pos);

  Neck\_servo.attach(10);

  Neck\_servo.write(gripper\_pos);

}

void loop() {

  /\*//USE FOR DEBUGGING

  Serial.println ("Gripper, UpDown, FrontBack, Neck");

  Serial.print(gripper\_pos);Serial.print(",");

  Serial.print(updown\_pos);Serial.print(",");

  Serial.print(frontback\_pos);Serial.print(",");

  Serial.print(neck\_pos);Serial.println(".");\*/

  delay(50); //predefined  delay to make the servo move slower

//A0 to control Gripper Servo

  int Joy\_value\_X1 = analogRead (A0);

  if (Joy\_value\_X1 > 700){

    gripper\_pos = gripper\_pos + 1;

    control\_servo (Gripper\_servo, gripper\_pos, 0);

  }

  if (Joy\_value\_X1 < 300){

    gripper\_pos = gripper\_pos - 1;

    control\_servo (Gripper\_servo, gripper\_pos, 0);

  }

//A1 to control UpDown Servo

    int Joy\_value\_Y1 = analogRead (A1);

  if (Joy\_value\_Y1 > 700){

    updown\_pos = updown\_pos + 1;

    control\_servo (UpDown\_servo, updown\_pos, 1);

  }

  if (Joy\_value\_Y1 < 300){

    updown\_pos = updown\_pos - 1;

    control\_servo (UpDown\_servo, updown\_pos, 1);

  }

//A2 to control FrontBack Servo

      int Joy\_value\_X2 = analogRead (A2);

  if (Joy\_value\_X2 > 700){

    frontback\_pos = frontback\_pos + 1;

    control\_servo (FrontBack\_servo, frontback\_pos, 2);

  }

  if (Joy\_value\_X2 < 300){

    frontback\_pos = frontback\_pos - 1;

    control\_servo (FrontBack\_servo, frontback\_pos, 2);

  }

//A3 to control Neck Servo

        int Joy\_value\_Y2 = analogRead (A3);

  if (Joy\_value\_Y2 > 700){

    neck\_pos = neck\_pos + 1;

    control\_servo (Neck\_servo, neck\_pos, 3);

  }

  if (Joy\_value\_Y2 < 300){

    neck\_pos = neck\_pos - 1;

    control\_servo (Neck\_servo, neck\_pos, 3);

  }

}