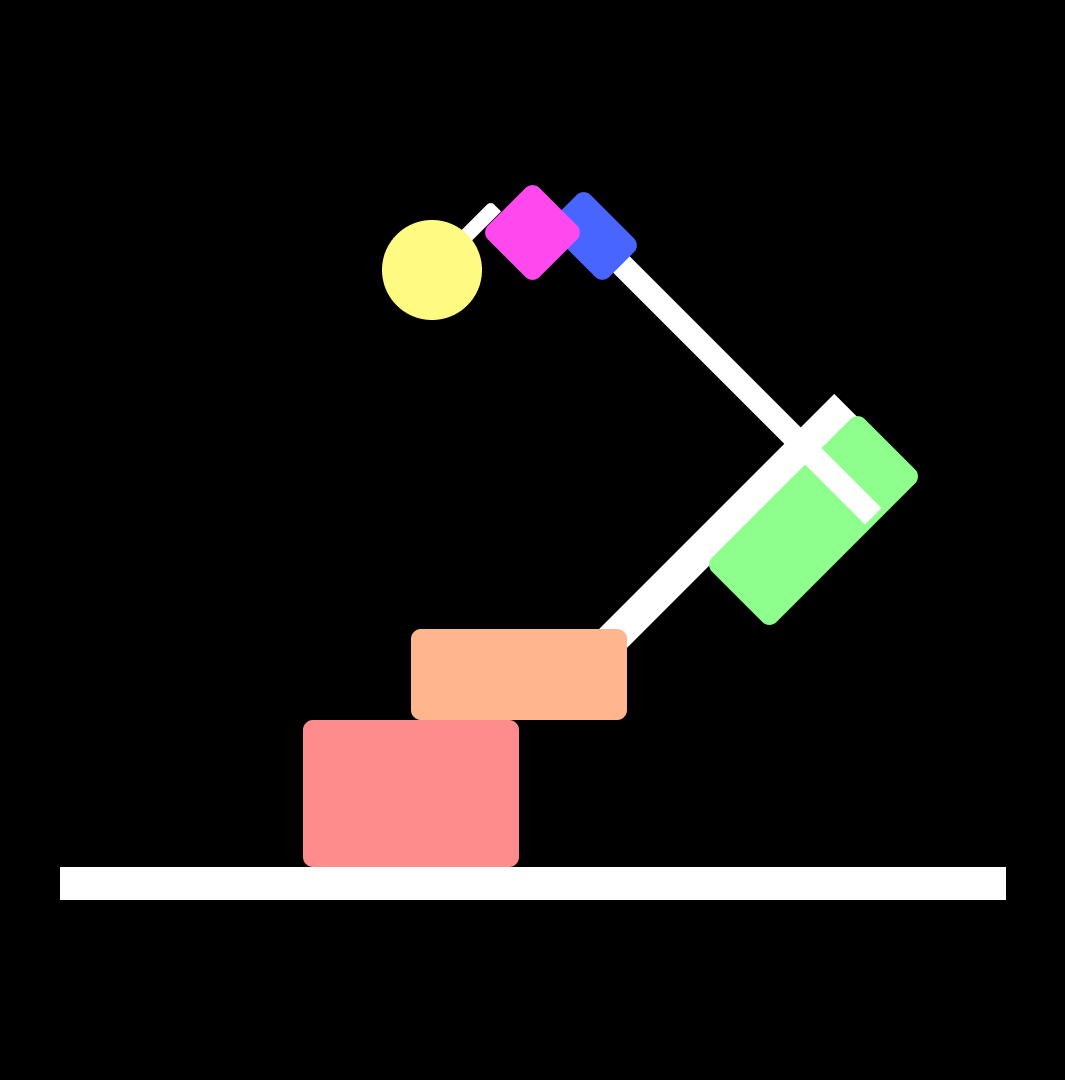
**Arduino Controlled Robotic Arm**

****

**Components**

* Arduino UNO
* HC-05 Bluetooth Module
* 5 servo motors
* Jumpers
* Power Source(Power Bank)
* Robot body parts

**Working in brief**

The Arduino is the main microcontroller here.

It receives commands from Bluetooth Serial Communication through the HC-05 Bluetooth Module.

The received command is being checked and the predefined actions are performed accordingly.

The Bluetooth Commands can be sent from any Bluetooth Client e.g. an Android device

**The Arduino Code**

#include <Servo.h>  //in-built Servo library

#include <string.h> //in-built String library

int delay\_time = 30; // default delay time for motor speed

int step = 1;        // delay step value for motor speed

Servo servos[5];                    // array of servo joints

int pos[5] = {10, 60, 40, 90, 100}; // array for holding joint servo's positions

int resetPin = 12; // resetPin connected at pin 12 of arduino

int baudRate = 9600; // default baudrate for HC-05

void setup()

{

    Serial.begin(baudRate);

    while (!Serial)

        ; // wait infinitely for Serial port to connected

    Serial.println("Serial port connected at: 9600");

    while (Serial.available() > 0)

        Serial.read(); // clear the Serial input stream

    Serial.flush(); // clear the Serial output stream

    servos[0].attach(3);

    servos[1].attach(5);

    servos[2].attach(6);

    servos[3].attach(9);

    servos[4].attach(10);

    for (int i = 0; i < 5; i++)

        servos[i].write(pos[i]); // setitng servos to their initial position...needed for speed calculation

    digitalWrite(resetPin, HIGH); // RESET pin configuration

    pinMode(resetPin, OUTPUT);

}

void loop()

{

    driver();    // driver function called

    delay(1000); // delay of 1s is provided after every command is performed.

}

void driver()

{

    String input\_command = "", input\_String = "";

    int input\_value = 0;

    if (Serial.available()) // checking if there is something in the Serial input stream

    {

        input\_command = Serial.readString();                // reads the whole command from the input

        input\_value = (input\_command.substring(1)).toInt(); // get the substring starting from index 1 to end and convert it to int

        switch (input\_command[0]) // the first letter of the input command is used for checking the cases.

        {

        case 'A':

            moveServo(input\_value, 0);

            break;

        case 'B':

            moveServo(input\_value, 1);

            break;

        case 'C':

            moveServo(input\_value, 2);

            break;

        case 'D':

            moveServo(input\_value, 3);

            break;

        case 'E':

            moveServo(input\_value, 4);

            break;

        case 'T':

            delay\_time = input\_value;

            Serial.println("Set delay to: " + String(delay\_time));

            break;

        case 'S':

            step = input\_value;

            Serial.println("Set step value to: " + String(step));

            break;

        case 'R':

            Serial.println("Resetting Arduino...");

            delay(1000);

            digitalWrite(resetPin, LOW);

            break;

        case 'P':

            Serial.println("Running Preprogrammed Movement...");

            while (!Serial.available())

            {

                moveServo(10, 4); // ROTATING

                moveServo(90, 0); // OPENING CLAWS

                moveServo(23, 3); // GOING DOWN

                moveServo(155, 1);

                moveServo(66, 2);

                moveServo(30, 0); // GRIPPING

                moveServo(60, 1); // GOING UP

                moveServo(75, 3);

                moveServo(169, 4); // ROTATE

                moveServo(23, 3); // GOING DOWN

                moveServo(155, 1);

                moveServo(90, 0); // RELEASING

                moveServo(60, 1); // GOING UP

                moveServo(75, 3);

                moveServo(10, 0); // CLOSING CLAWS

            }

            break;

        case 'M':

            Serial.println("Entering Manual Mode...");

            break;

        default:

            Serial.println("Wrong Command!");

        }

    }

}

void moveServo(int angle, int tag)

{

    if (angle < 10 || angle > 169)

    {

        Serial.println("Angle out of range!");

    }

    else

    {

        Serial.println("Turning joint " + String(tag) + " from " + String(pos[tag]) + " to " + String(angle) + " degree.");

        if (pos[tag] > angle)

        {

            for (int i = pos[tag]; i >= angle; i -= step)

            {

                pos[tag] = i;

                servos[tag].write(i);

                delay(delay\_time);

            }

        }

        else if (pos[tag] < angle)

        {

            for (int i = pos[tag]; i <= angle; i += step)

            {

                pos[tag] = i;

                servos[tag].write(i);

                delay(delay\_time);

            }

        }

        else

            ;

    }

}