

A dark blue vertical bar runs down the left side of the page. A teal arrow points to the right from the bar, containing the text '[Date]'. Below the bar, several thin, curved lines in dark blue and light grey sweep upwards from the bottom left towards the center of the page.

[Date]

[Document title]

[Document subtitle]

MUHAMMAD ABDULLAH BIN IMRAN
DILSHAD AHMED

INTRODUCTION:

Spot Welding is common process in our manufacturing industry, but unfortunately it is done manually by labors. So, this process take time to done manually and safety issues are also present there, due to which industry has to pay a heavy price. For this reason, we come up with the idea of making robotic arm which is capable of doing welding automatically.

DEVELOPMENT TOOLS:

- ARDUINO UNO:

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button.



- SERVO MOTOR:

A servo motor is a rotational or translational motor to which power is supplied by a servo amplifier and serves to apply torque or force to a mechanical system, such as an actuator or brake. This type of motor is associated with a closed-loop control system.



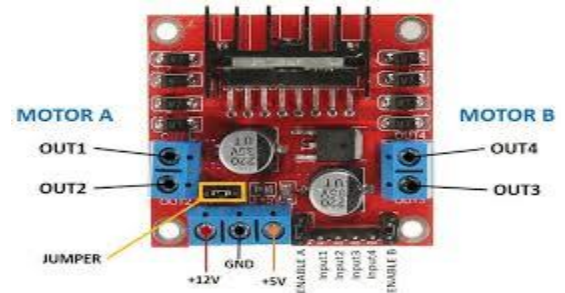
- BLUETHOOTH MODULUE(HC-05):

HC-05 Bluetooth Module is an easy-to-use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. ... HC-05 Bluetooth module provides switching mode between master and slave mode which means it able to use neither receiving nor transmitting data.



- **L298N MOTOR DRIVER MODULE:**

Motor drivers acts as an interface between the motors and the control circuits. Motor requires high amount of current whereas the controller circuit works on low current signals. So, the function of motor drivers is to take a low-current control signal and then turn it into a higher-current signal that can drive a motor.



- **CAR CHASSIS WITH DC GEAR MOTORS:**

car chassis connected with dc motors along with gears and wheels.it is used for the movement if needed in projects.



- **RELAYS MODULE:**

The relay is the device that open or closes the contacts to cause the operation of the other electric control. It detects the undesirable condition with an assigned area and gives the commands to the circuit breaker to disconnect the affected area through ON or OFF.



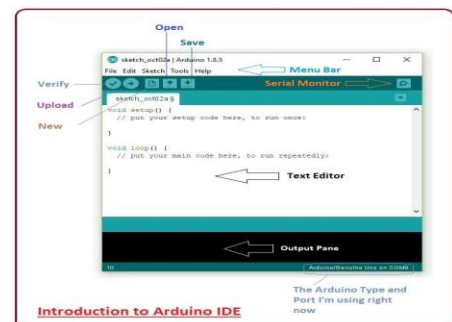
- **BATTERY AND WIRSE:**

Battery is used to power to entire circuit while jumper wires are used to connect the circuit from one end to another.

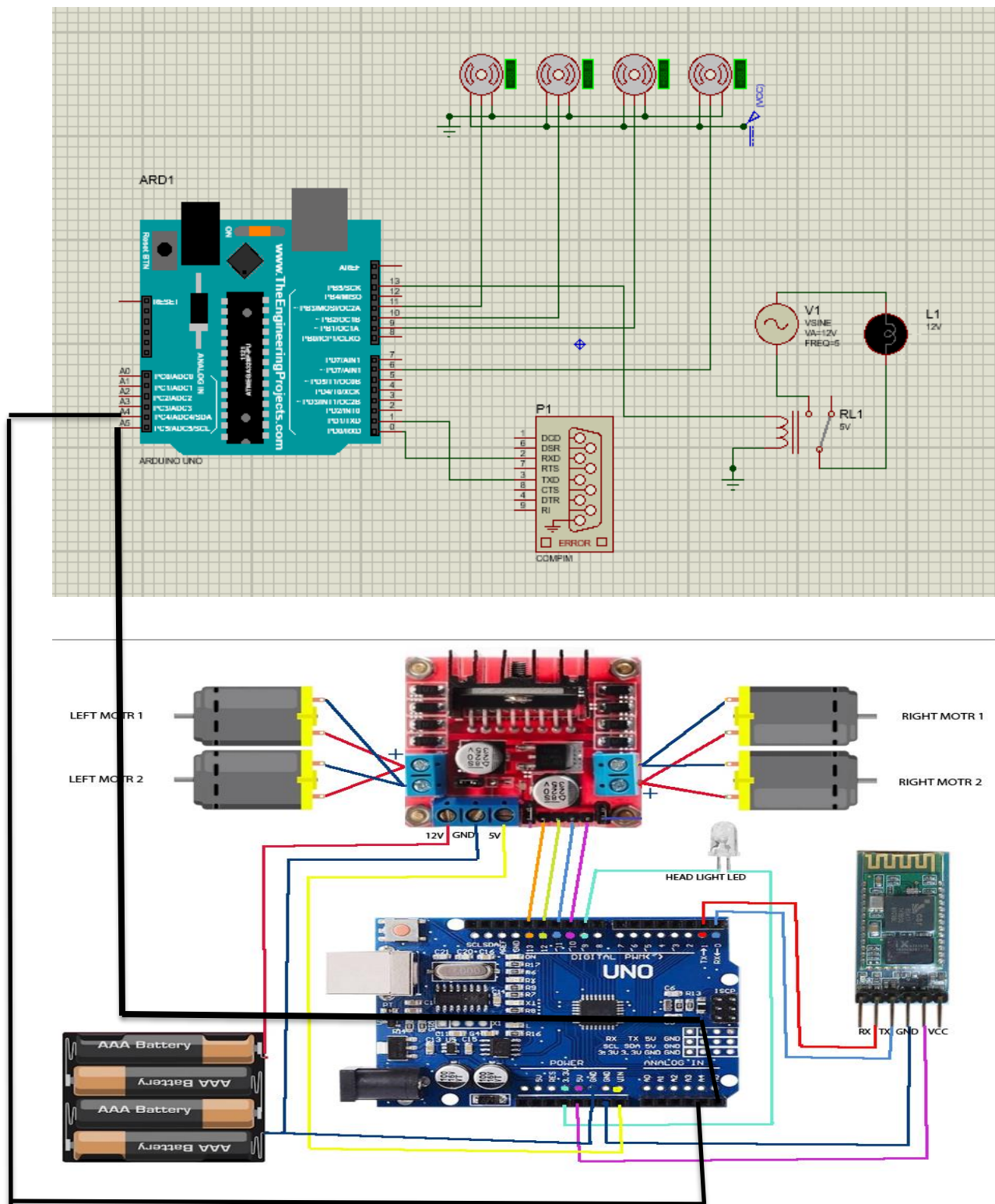


- **ARDUINO COMPILER:**

The Arduino IDE is a cross-platform application that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards.



- CIRCUIT DIGRAMS:



Arm Code

```
#include <SoftwareSerial.h>

#include <Servo.h>

#include <Wire.h>

Servo servo01;

Servo servo02;

Servo servo03;

Servo servo04;

Servo servo05;

Servo servo06;

SoftwareSerial Bluetooth(3, 4); // Arduino(RX, TX) - HC-05 Bluetooth (TX, RX)

int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos, servo6Pos; // current position

int servo1PPos, servo2PPos, servo3PPos, servo4PPos, servo5PPos, servo6PPos; // previous position

int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50], servo05SP[50], servo06SP[50]; // for storing
positions/steps

int speedDelay = 20;

int index = 0;

String dataIn = "";

void setup() {

    servo01.attach(5);

    servo02.attach(6);

    servo03.attach(7);

    servo04.attach(8);

    servo05.attach(9);

    servo06.attach(10);

    Bluetooth.begin(38400); // Default baud rate of the Bluetooth module

    Wire.begin();

    Bluetooth.setTimeout(1);

    delay(20);
```

```

// Robot arm initial position

servo1PPos = 90;

servo01.write(servo1PPos);

servo2PPos = 150;

servo02.write(servo2PPos);

servo3PPos = 35;

servo03.write(servo3PPos);

servo4PPos = 140;

servo04.write(servo4PPos);

servo5PPos = 85;

servo05.write(servo5PPos);

servo6PPos = 80;

servo06.write(servo6PPos);
}

void loop() {

    // Check for incoming data

    if (Bluetooth.available() > 0) {

        dataIn = Bluetooth.readString(); // Read the data as string


        // If "Waist" slider has changed value - Move Servo 1 to position

        if (dataIn.startsWith("s1")) {

            String dataInS = dataIn.substring(2, dataIn.length()); // Extract only the number. E.g. from "s1120" to "120"

            servo1Pos = dataInS.toInt(); // Convert the string into integer

            // We use for loops so we can control the speed of the servo

            // If previous position is bigger then current position

            if (servo1PPos > servo1Pos) {

                for ( int j = servo1PPos; j >= servo1Pos; j--) { // Run servo down

                    servo01.write(j);

                    delay(20); // defines the speed at which the servo rotates
                }
            }
        }
    }
}

```

```

    }

    }

    // If previous position is smaller then current position
    if (servo1PPos < servo1Pos) {

        for ( int j = servo1PPos; j <= servo1Pos; j++) { // Run servo up

            servo01.write(j);

            delay(20);

        }

    }

    servo1PPos = servo1Pos; // set current position as previous position
}

// Move Servo 2

if (dataIn.startsWith("s2")) {

    String dataInS = dataIn.substring(2, dataIn.length());

    servo2Pos = dataInS.toInt();

    if (servo2PPos > servo2Pos) {

        for ( int j = servo2PPos; j >= servo2Pos; j--) {

            servo02.write(j);

            delay(50);

        }

    }

    if (servo2PPos < servo2Pos) {

        for ( int j = servo2PPos; j <= servo2Pos; j++) {

            servo02.write(j);

            delay(50);

        }

    }

    servo2PPos = servo2Pos;

```

```

}

// Move Servo 3

if (dataIn.startsWith("s3")) {

    String dataInS = dataIn.substring(2, dataIn.length());

    servo3Pos = dataInS.toInt();

    if (servo3PPos > servo3Pos) {

        for ( int j = servo3PPos; j >= servo3Pos; j--) {

            servo03.write(j);

            delay(30);

        }

    }

    if (servo3PPos < servo3Pos) {

        for ( int j = servo3PPos; j <= servo3Pos; j++) {

            servo03.write(j);

            delay(30);

        }

    }

    servo3PPos = servo3Pos;

}

// Move Servo 4

if (dataIn.startsWith("s4")) {

    String dataInS = dataIn.substring(2, dataIn.length());

    servo4Pos = dataInS.toInt();

    if (servo4PPos > servo4Pos) {

        for ( int j = servo4PPos; j >= servo4Pos; j--) {

            servo04.write(j);

            delay(30);

        }

    }

}

```



```

if (servo4PPos < servo4Pos) {

    for ( int j = servo4PPos; j <= servo4Pos; j++) {

        servo04.write(j);

        delay(30);

    }

}

servo4PPos = servo4Pos;

}

// Move Servo 5

if (dataIn.startsWith("s5")) {

    String dataInS = dataIn.substring(2, dataIn.length());

    servo5Pos = dataInS.toInt();

    if (servo5PPos > servo5Pos) {

        for ( int j = servo5PPos; j >= servo5Pos; j--) {

            servo05.write(j);

            delay(30);

        }

    }

    if (servo5PPos < servo5Pos) {

        for ( int j = servo5PPos; j <= servo5Pos; j++) {

            servo05.write(j);

            delay(30);

        }

    }

    servo5PPos = servo5Pos;

}

// Move Servo 6

if (dataIn.startsWith("s6")) {

    String dataInS = dataIn.substring(2, dataIn.length());

```

```

servo6Pos = dataInS.toInt();

if (servo6PPos > servo6Pos) {

  for ( int j = servo6PPos; j >= servo6Pos; j--) {

    servo06.write(j);

    delay(30);

  }

}

if (servo6PPos < servo6Pos) {

  for ( int j = servo6PPos; j <= servo6Pos; j++) {

    servo06.write(j);

    delay(30);

  }

}

servo6PPos = servo6Pos;

}

if (dataIn.startsWith("CAR")) {

  String dataInS = dataIn.substring(3, dataIn.length());

  if (dataInS == "forward" || dataInS == "reverse" || dataInS == "right" || dataInS == "left" || dataInS == "stop" ) {

    Wire.beginTransaction(1);

    Wire.write(dataInS);

    Wire.endTransmission();

  }

}

// If button "SAVE" is pressed

if (dataIn.startsWith("SAVE")) {

  servo01SP[index] = servo1PPos; // save position into the array

  servo02SP[index] = servo2PPos;

  servo03SP[index] = servo3PPos;

  servo04SP[index] = servo4PPos;

}

```



```

    dataIn = Bluetooth.readString();

    if ( dataIn == "RESET") {

        break;

    }

}

}

}

}

// If speed slider is changed
if (dataIn.startsWith("ss")) {

    String dataInS = dataIn.substring(2, dataIn.length());

    speedDelay = dataInS.toInt(); // Change servo speed (delay time)

}

}

// Servo 1
if (servo01SP[i] == servo01SP[i + 1]) {

}

if (servo01SP[i] > servo01SP[i + 1]) {

    for ( int j = servo01SP[i]; j >= servo01SP[i + 1]; j--) {

        servo01.write(j);

        delay(speedDelay);

    }

}

if (servo01SP[i] < servo01SP[i + 1]) {

    for ( int j = servo01SP[i]; j <= servo01SP[i + 1]; j++) {

        servo01.write(j);

        delay(speedDelay);

    }

}

// Servo 2

```

```
if (servo02SP[i] == servo02SP[i + 1]) {  
    }  
if (servo02SP[i] > servo02SP[i + 1]) {  
    for ( int j = servo02SP[i]; j >= servo02SP[i + 1]; j--) {  
        servo02.write(j);  
        delay(speedDelay);  
    }  
}  
if (servo02SP[i] < servo02SP[i + 1]) {  
    for ( int j = servo02SP[i]; j <= servo02SP[i + 1]; j++) {  
        servo02.write(j);  
        delay(speedDelay);  
    }  
}  
// Servo 3  
if (servo03SP[i] == servo03SP[i + 1]) {  
    }  
if (servo03SP[i] > servo03SP[i + 1]) {  
    for ( int j = servo03SP[i]; j >= servo03SP[i + 1]; j--) {  
        servo03.write(j);  
        delay(speedDelay);  
    }  
}  
if (servo03SP[i] < servo03SP[i + 1]) {  
    for ( int j = servo03SP[i]; j <= servo03SP[i + 1]; j++) {  
        servo03.write(j);  
        delay(speedDelay);  
    }  
}
```

```

// Servo 4

if (servo04SP[i] == servo04SP[i + 1]) {

}

if (servo04SP[i] > servo04SP[i + 1]) {

    for ( int j = servo04SP[i]; j >= servo04SP[i + 1]; j--) {

        servo04.write(j);

        delay(speedDelay);

    }

}

if (servo04SP[i] < servo04SP[i + 1]) {

    for ( int j = servo04SP[i]; j <= servo04SP[i + 1]; j++) {

        servo04.write(j);

        delay(speedDelay);

    }

}

// Servo 5

if (servo05SP[i] == servo05SP[i + 1]) {

}

if (servo05SP[i] > servo05SP[i + 1]) {

    for ( int j = servo05SP[i]; j >= servo05SP[i + 1]; j--) {

        servo05.write(j);

        delay(speedDelay);

    }

}

if (servo05SP[i] < servo05SP[i + 1]) {

    for ( int j = servo05SP[i]; j <= servo05SP[i + 1]; j++) {

        servo05.write(j);

        delay(speedDelay);

    }

}

```

```
}  
  
// Servo 6  
  
if (servo06SP[i] == servo06SP[i + 1]) {  
  
}  
  
if (servo06SP[i] > servo06SP[i + 1]) {  
  
    for ( int j = servo06SP[i]; j >= servo06SP[i + 1]; j--) {  
  
        servo06.write(j);  
  
        delay(speedDelay);  
  
    }  
  
}  
  
if (servo06SP[i] < servo06SP[i + 1]) {  
  
    for ( int j = servo06SP[i]; j <= servo06SP[i + 1]; j++) {  
  
        servo06.write(j);  
  
        delay(speedDelay);  
  
    }  
  
}  
  
}  
  
}
```

Car code

```
#include <Wire.h>
```

```
const byte slaveId = 1;
```

```
void setup() {
```

```
    Wire.begin(slaveId);
```

```
    Wire.onReceive(receiveEvent);
```

```
    pinMode(9, OUTPUT); //left motors forward
```

```
    pinMode(10, OUTPUT); //left motors reverse
```

```
    pinMode(11, OUTPUT); //right motors forward
```

```
    pinMode(12, OUTPUT); //right motors reverse
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop() {
```



```
}
```

```
void receiveEvent(int howMany) {
```

```
    String inChar;
```

```
    while (Wire.available() > 0)
```

```
    {
```

```
        inChar = Wire.read();
```

```
        if (Serial.available()) {
```

```
            Serial.println(inChar);
```

```
        }
```

```
        if (inChar == "forward") { //move forward(all motors  
rotate in forward direction)
```

```
            digitalWrite(9, HIGH);
```

```
            digitalWrite(10, LOW);
```

```
            digitalWrite(11, HIGH);
```

```
            digitalWrite(12, LOW);
```

```
}
```

```
    else if (inChar == "reverse") {    //move reverse (all motors  
rotate in reverse direction)
```

```
        digitalWrite(9, LOW);
```

```
        digitalWrite(10, HIGH);
```

```
        digitalWrite(11, LOW);
```

```
        digitalWrite(12, HIGH);
```

```
    }
```

```
    else if (inChar == "right") {    //turn right (left side motors rotate  
in forward direction, right side motors doesn't rotate)
```

```
        digitalWrite(9, LOW);
```

```
        digitalWrite(10, LOW);
```

```
        digitalWrite(11, HIGH);
```

```
        digitalWrite(12, LOW);
```

```
    }
```

```
    else if (inChar == "left") {    //turn left (right side motors rotate  
in forward direction, left side motors doesn't rotate)
```

```
        digitalWrite(9, HIGH);
```

```
    digitalWrite(10, LOW);  
    digitalWrite(11, LOW);  
    digitalWrite(12, LOW);  
}  
  
else if (inChar == "stop") { //STOP (all motors stop)  
    digitalWrite(9, LOW);  
    digitalWrite(10, LOW);  
    digitalWrite(11, LOW);  
    digitalWrite(12, LOW);  
    } delay(100);  
}  
}
```