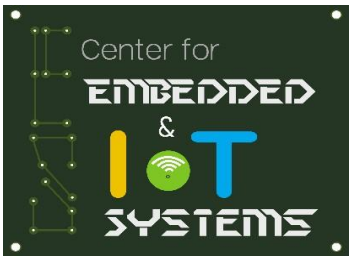
PICK AND PLACE ROBOT



A project report submitted in partial fulfillment of requirement for the course

On

Smart System Design

By

PERLA SINDHU	2205A42035
MANTRI HARSHINI	2205A42013

Under the guidance of

Mr. Rajeshwar Rao Arabelli

Asst. Prof. & Director, Centre for Embedded Systems and IoT,

Department of ECE

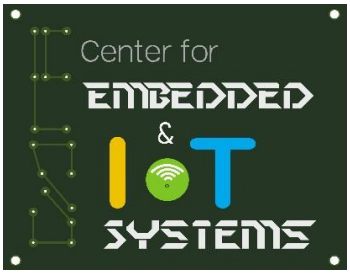
&

Dr. Ch.Rajendra Prasad

Asst. Prof., Department of ECE

Center for Embedded Systems and Internet of things

SR UNIVERSITY



CERTIFICATE

This is to certify that the course project entitled "Pick and Place" is the bonafide work carried out by PERLA SINDHU (2205A42035), MANTRI HARSHINI (2205A42013)**, in the partial fulfillment of the requirement for the award of course Smart System Design** during the academic year 2022-2023 under our guidance and Supervision.

Mr. Rajeshwar Rao Arabelli

Asst. Prof. & Director, Centre for Embedded Systems and IoT,

Department of ECE

CONTENTS

INTRODUCTION

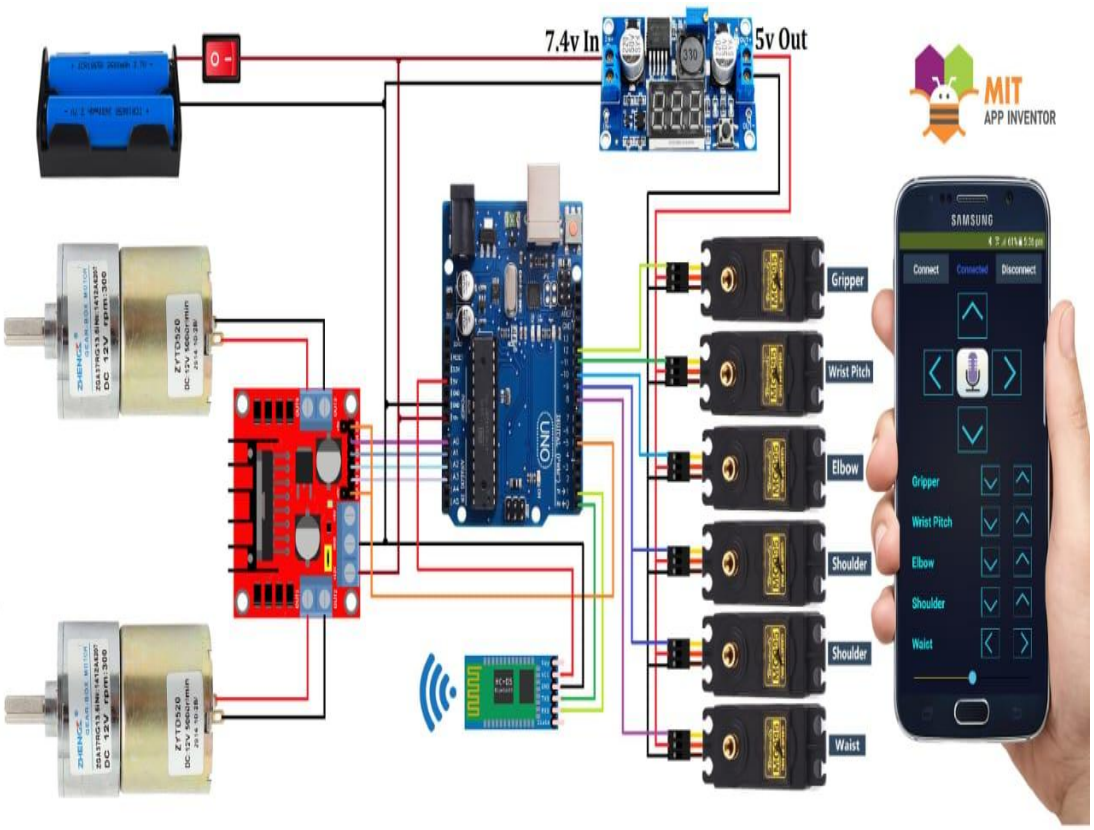
Homes of the 21st century will become more and more self-controlled and automated due to the comfort it provides, especially when employed in a private home. A home automation system is a means that allow users to control electric appliances of varying kind.Many existing, well-established home automation systems are based on wired communication. This does not pose a problem until the system is planned well in advance and installed during the physical construction of the building. But for already existing buildings the implementation cost goes very high. Home automation is the process of controlling home appliances automatically using various control system techniques. The electrical and electronic appliances in the home such as fan, lights, outdoor lights, fire alarm, kitchen timer, etc., can be controlled using various control techniques. There are various techniques to control home appliances such as home automation under Wi-Fi through android apps from any smartphone, Arduino based home automation, home automation by android application based remote control, home automation using digital control, RF based home automation system and touch screen based home automation.

CHAPTER 2

Project description

2. 1 BLOCK DIAGRAM OF THE PROJECT

As shown in the above schematic diagram it mainly consists of an arduino, one motor shield 2 servo motors ., hc05 bluctooth module is used to control the bot wirless using mobile.These de motors are connected to Arduino The block diagram of the project is shown in fig



2.2 HARDWARE DESCRIPTION

2.2.1Arduino UNO

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). Ithas 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs. a16 MHZ ceramic resonator, a USB connection, a power jack, an ICSP header, and a resetbutton.

It contains everything needed to support the microcontroller; simply connect it to acomputer with a USB cable or power it with a AC-to-DC adapter or battery to get started. TheUno difers from all preceding boards in that it does not use the FTDI USB-to-serial driverchip. Instead, it fcatures the. Atmega1 6U2 (Atmega8U2 up to version R2) programmed as aUSB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB lineto ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features: 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes. Stronger RESET circuit. Atmega 16U2 replace the 8U2. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)

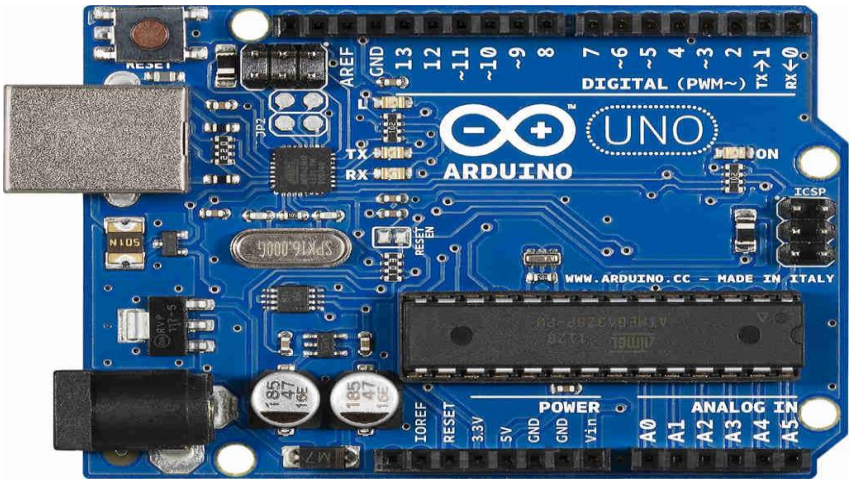


Fig. 2.2 Arduino Uno

Applications:

- Xoscillo, an open-source oscilloscope
- Arduinome, a MIDI controller device that mimics the Monome
- OBDuino, a trip computer that uses the on-board diagnostics interface found in most modern cars
- Gameduino, an Arduino shield to create retro 2D video games
- ArduinoPhone, a do-it-yourself cellphone
- Water quality testing platform
- Automatic titration system based on Arduino and stepper motor
- Low cost data glove for virtual reality applications
- Impedance sensor system to detect bovine milk adulteration
- Homemade CNC using Arduino and DC motors with close loop control by Homofaciens
- DC motor control using Arduino and H-Bridge

2.2.2 HC-05 Bluetooth Module

Ever wanted, to control your Mechanical Bots with an Android Phone or design the robotswith custom remote, here in this tutorial we will learn about a Bluetooth Module HC-05 usedfor the above mentioned and many other cases. Here we will be understanding the connectionand working of a HC-05 module and also its. interfacing with custom android app.

Basics

Wireless communication is swiftly replacing the wired connection when it comestoelectronics and communication. Designed to replace cable connections HC-05 uses serialcommunication to communicate with the electronics. Usually, it is used to connect smalldevices like mobile phones using a short-range wireless connection to exchange files. It usesthe 2. 45GHZ frequency band. The transfer rate of the data can vary up to 1Mbps and is in.The HC-05 module can be operated within 4-6V of power supply. It supports baud rate of9600, 19200, 38400, 57600, etc. Most importantly it can be operated in Master-Slave modewhich means it will neither send or receive data from external sources

Enable - This pin is used to set the Data Mode or and AT command mode (set high),

VCC- This is connected to +5V power supply.

Ground - Connected to ground of powering system.Tx (Transmitter) - This pin transmits the received data Serially.

Rx Receiver) - Used for broadcasting data serially over bluetooth.

State-Used to check if the bluetooth is working properly

Features:

- HC05 follows the "Bluetooth V2.0+EDR" protocol (EDR stands for EnhancedData Rate).
- Its operating frequency is 2.4 GHZ ISM Band
- HC05 uses CSR Bluecore 04-External single-chip Bluetooth system with CMOS technology.
- This module follows the IEEE (Institute of Electrical and Electronics Engineers) 802. 15.1 standard protocol.
- Dimensions of HC-05 are 12.7mmx27mm
- Its operating voltage is 5V
- It sends and receives data by UART, which is also used for setting the baud rate
- It has -80dBm sensitivity
- This module also uses (FHSS), a technique by which a radio signal is sent at different frequency levels
- This module has the ability to work as a master/slave mode
- This module can be easily connected with a laptop or mobile phone via Bluetooth

2.2.3 Servo Motor

A servomotor is a rotatetory actuator or linear actuator that allows for precise controlof angular or linear position, velocity, and acccleration. It consists of a suitable motor coupledto a sensor for position feedback. It also requires a rclatively sophisticated controller, often adedicated module designed specifically for use use with with servomotors.

Fig.2.4 ServoMotor

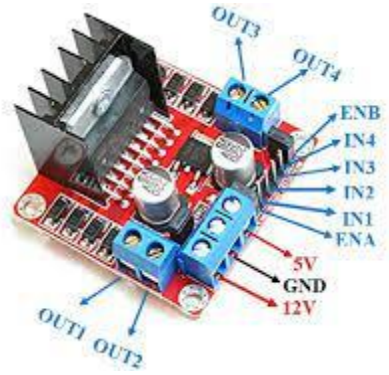
Mechanism

A servomotor is a closed servo motor mechanisam that uses position feedback to control itsmotion and final position. The input to its control is a signal (either analog or digital)representing the position commanded for the output shaft.

The motor is paired with some type of position encoder to provide position andspeed feedbackIn the simplest case, only the position is measured. The measured position of the output iscompared to the command position, the external input to the controller. If the output positiondiffers from that required, an error signal is generated which then causes the motor to rotate ineither direction, as needed to bring the output shaft to the appropriate position. As the positions approach, the error signal reduces to zero, and the motor stops.The very simplest servomotors use position-only sensing via a potentiometer and bang bang of their

motor; the motor always rotates at full speed (or is stopped). This type of servomotor isnot widely used in industrial motion control, but it forms the basis of the simple andcheap servos used for radio controlled servos

2.2.4 Drive module



The Drive module brings together several functions within a lightweight and high- strength aluminum structure. This is the basic body, complete with the suspension, crash element, energy storage device and drive unit. Weighing around 250 kg and withdimensions similar to those of a child's mattress, the energy storage system is the driving element of the integrative and functional design of the Drive module. The initial priority in the conception of the Drive module was therefore to integrate the battery – the largest and heaviest factor in the electric vehicle in terms of construction – into the vehicle structure so that it would be operationally reliable and safe in a crash

2.2.5 Dc motor

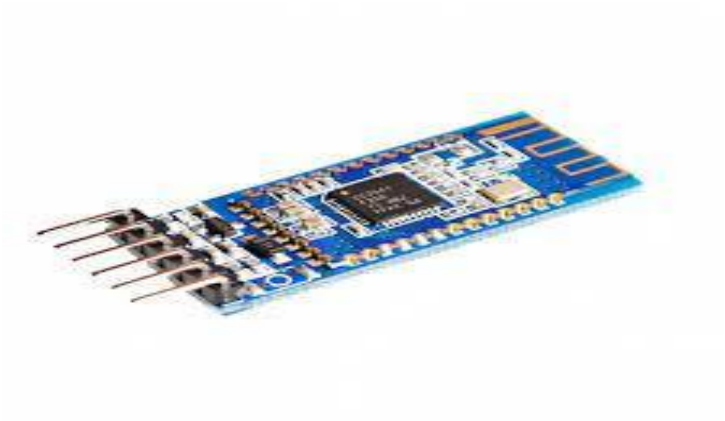
A DC motor is an electric motor that runs on direct current (DC) power. It is designed to convert

electrical energy into mechanical energy and is the most common type of motor used in everydayapplications. DC motors can be used for a wide range of applications, from small toys to large industrial machines. They are used in cars, robots, appliances, and many other devices.DC motors are relatively simple to operate, and they are highly efficient and reliable.

Mechanism

A DC motor consists of a stator, rotor, and an armature. The stator is a stationary part of the motor which houses the permanent magnets. The rotor is the rotating part of the motor and is attached to an armature which is a coil of wire. When a current is passed through the coil, a magnetic field is created, causing the rotor to rotate. The speed and direction of the motor can be controlled by changing the current, voltage, or the number of poles

2.2.6 Bluetooth module



A Bluetooth module is a type of wireless communication module that utilizes Bluetooth technology to enable short-range communication. These modules can be used to establish a wireless connection between two computers or devices, allowing them to exchange data. Bluetooth modules are most commonly found in mobile phones, headsets, and other consumer electronics. They are also used in industrial applications such as robotics, medical devices, and security systems. Bluetooth modules are typically small and have a low power consumption. They are designed to be reliable and robust, with the ability to establish a stable connection even in noisy or crowded environments.

Mechanism

Bluetooth modules use radio frequency signals to communicate with each other. The frequency

used by the module is determined by the Bluetooth version employed. The modules must be in close proximity to each other to establish a connection. Once a connection is established, the data is sent using a low-power radio signal that is transmitted in a packet format. The receiving module decodes the packet and sends an acknowledgment to the sending module

2.1.7 BATTERIES:

This is an original 2600mAh 18650 battery, 18650 battery is a Li-ion rechargeable battery with a 1200 mAh Battery Capacity. This is not a standard AA or AAA battery but is very useful for applications that require continuous high current.

3 SOFTWARE DESCRIPTION

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Writing Sketches:

Programs written using Arduino Software (IDE) are called sketches. Thessketches are written in the text editor and are saved with the file extension ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom

righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

NB:

Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension pde. It is possible to open these files with version 1.0, you will be prompted to save the sketc with the ino extension on save.

Verify

Checks your code for errors compiling it.

Upload

Compiles your code and uploads it to the configured board. See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"

New

Creates a new sketch.

Open

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead.

Save

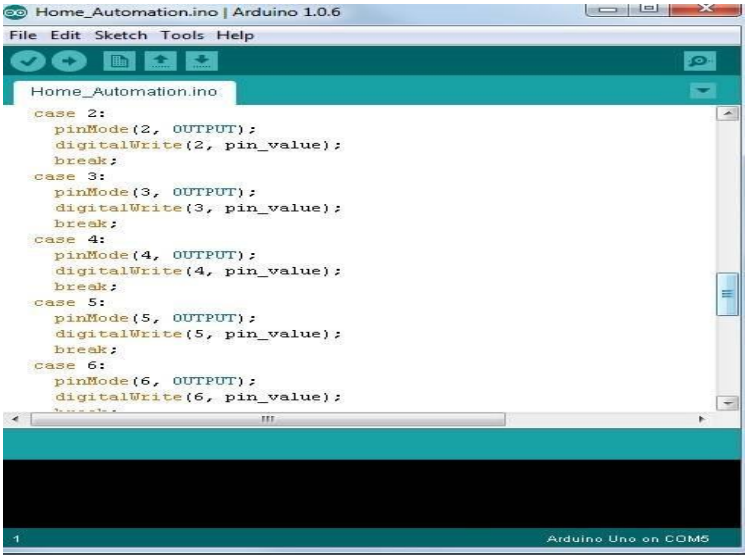
Saves your sketch.

Serial Monitor

Opens the serial monitor.

- Additional commands are found within the five menus: File, Edit, Sketch, Tools,and help.

Programming on arduinouno



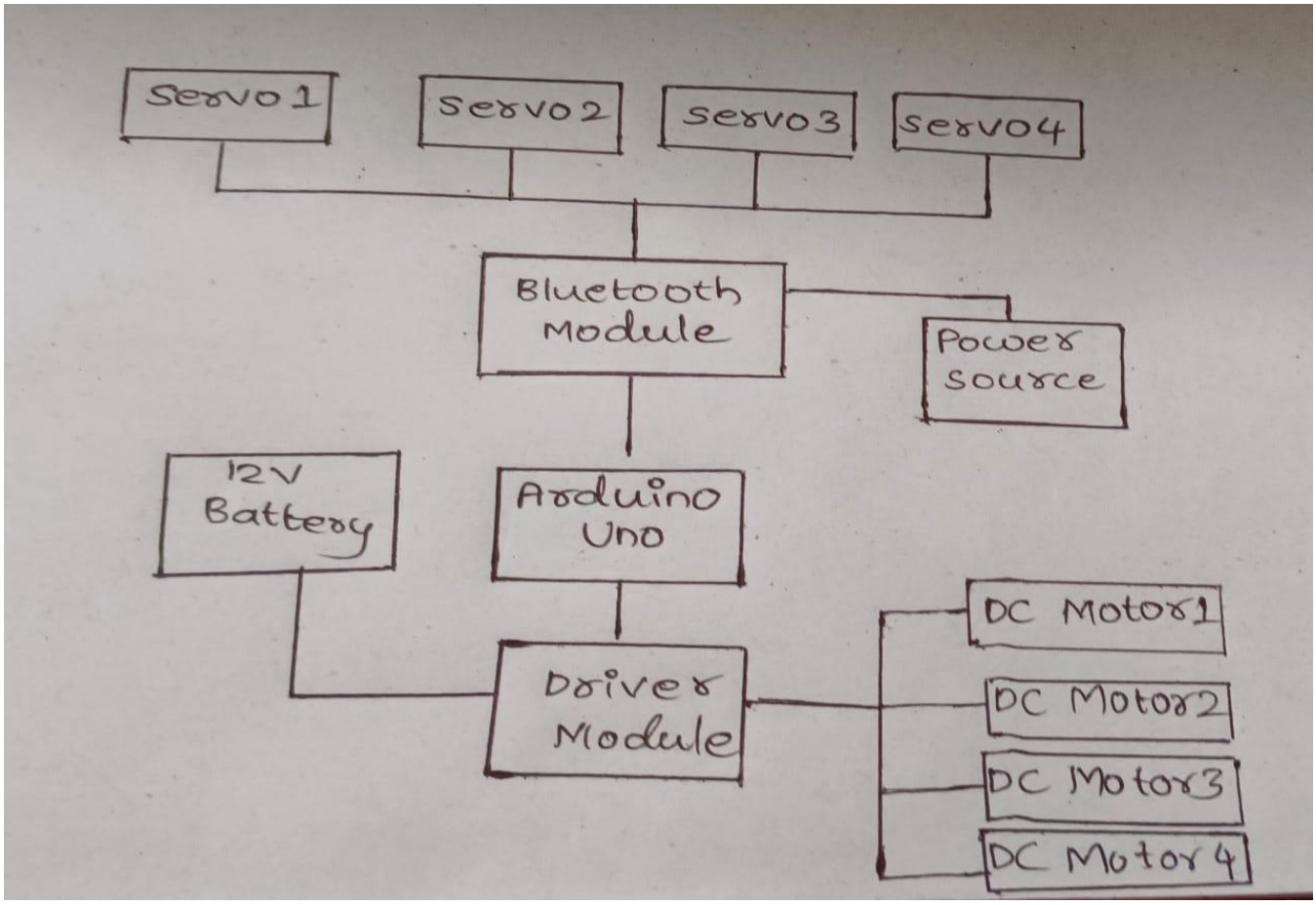
In order for the Arduino-Uno board to be able to interact with the application used in this project certain program (code) needs to be uploaded to the Arduino-Uno. Arduino Company provides user friendly software which allows writing any code for any function wanted to be performed by the Arduino-Uno and upload it to the board.Refer to appendix A for the full source code of the Arduino-Uno board.

CHAPTER 3

CIRCUIT DIAGRAM AND DESCRIPTION

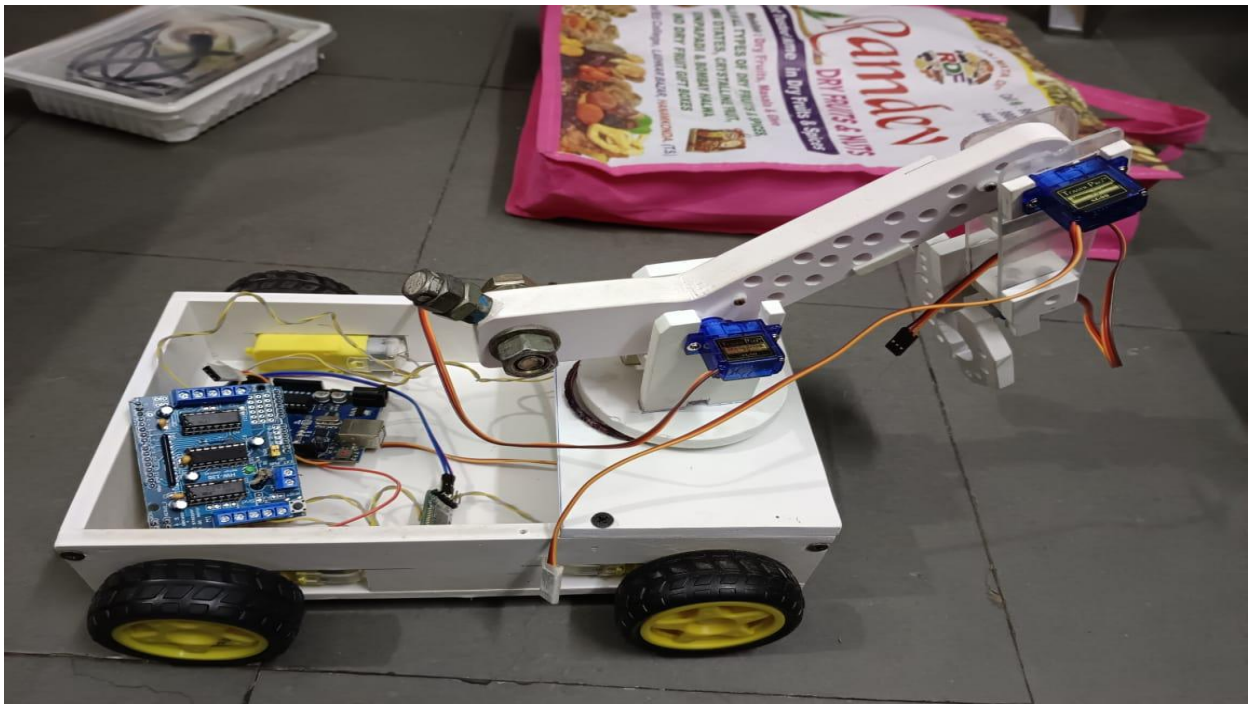
3.1 Working

Typically mounted on a stable stand, pick and place robots are positioned to reach differentareas to perform work. They use advanced vision systems to identify, grasp and move objectsfirom one place to another . With a variety of design options available, pick and place robots canbe configured with various end-of-arm tooling options for use in different applications, suchas assembly, packaging or bin picking. For example, pick and place robot systems may be usedto pick up items for an order and place them in a box for packaging, or they may be used topick up parts needed for assembly and move them to the next location



3.2 RESULT

The experimental result is as shown in below fig. 3.2



3.2 Experimental result

Managed to successfully apply the PICK AND PLACE ROBO and it was user tiendlyand cost effective. User friendly as in anyone can use just a click of a button on an androidscreen and

everything works. And it is cost effective as in it will cost exactly as the projectrequires (optimum price).

3.3 ADVANTAGES

- They are faster and can get the work done in seconds compared to their humancounterparts.
- They are flexible and have the appropriate design
- They increase the safety the working environment and actually never get
- They are accurate.

3.4DISADVANTAGES

- Human errors.
- Reliability.
- System compatibility.

CHAPTER 4

CONCLUSION

4.1 CONCLUSION

In today's dynamic manufacturing landscape, pick and place robots have become an essential driver of efficiency, hygiene, and consistency—especially in industries like food packaging. With advanced solutions like PWR Pack's high-speed robotic systems, companies can confidently meet rising consumer demands while maintaining the highest standards of quality and safety. By replacing manual processes with intelligent automation, businesses not only eliminate production bottlenecks but also unlock greater productivity, reliability, and growth opportunities.

5 REFERENCE

https://www.pwrpack.com/

3D Vision-Guided Pick-and-Place Using KUKA LBR iiwa Robot

https://arxiv.org/abs/2102.10710

Toward Fast and Optimal Robotic Pick-and-Place on a Moving Conveyor

https://arxiv.org/abs/1912.08009

Arduino Pick N Place Android Robot (Instructables)

https://www.instructables.com/Arduino-Pick-N-Place-Android-Robot/

Pick and Place Robotic Arm Using Arduino (ResearchGate)

https://www.researchgate.net/publication/332565132_Pick_and_Place_Robotic_Arm_Using_Arduino

code

```
#include <Servo.h>

Servo motor_1;
Servo motor_2;
Servo motor_3;
Servo motor_4;

#define enA 4  //Enable1 L298 Pin enA
#define in1 6 //Motor1  L298 Pin in1
#define in2 7 //Motor1  L298 Pin in1
#define in3 8 //Motor2  L298 Pin in1
#define in4 9 //Motor2  L298 Pin in1
#define enB 5  //Enable2 L298 Pin enB

int servo1 = 90;
int servo2 = 110;
int servo3 = 40;
int servo4 = 50;

int state=0;
void setup() {
  pinMode(enA, OUTPUT); // declare as output for L298 Pin enA
pinMode(in1, OUTPUT); // declare as output for L298 Pin in1
pinMode(in2, OUTPUT); // declare as output for L298 Pin in2
pinMode(in3, OUTPUT); // declare as output for L298 Pin in3
```

```
pinMode(in4, OUTPUT); // declare as output for L298 Pin in4
pinMode(enB, OUTPUT); // declare as output for L298 Pin enB


motor_1.attach(10);
motor_2.attach(11);
motor_3.attach(12);
motor_4.attach(13);

motor_1.write(servo1);
motor_2.write(servo2);
motor_3.write(servo3);
motor_4.write(servo4);

  Serial.begin(9600); // Default communication rate of the Bluetooth module
}



void loop() {
  if(Serial.available() > 0){ // Checks whether data is comming from the serial port
    state = Serial.read(); // Reads the data from the serial port
 Serial.println(state);
 }

 digitalWrite(enA, HIGH); // Write The Duty Cycle 0 to 255 Enable Pin A for Motor1 Speed
digitalWrite(enB, HIGH); // Write The Duty Cycle 0 to 255 Enable Pin B for Motor2 Speed


 if (state == '1') {
  forword();
  Serial.println("LED: OFF"); // Send back, to the phone, the String "LED: ON"
  state = 0;
 }
 else if (state == '2') {
 backword();
  Serial.println("LED: ON");;
  state = 0;
 }
 else if(state == '3'){turnLeft();state = 0;}  // if the bt_data is '3' the motor will turn left
else if(state == '4'){turnRight();state = 0;} // if the bt_data is '4' the motor will turn right
else if(state == '5'){Stop();state = 0; }      // if the bt_data '5' the motor will Stop

else if(state == '6'){turnLeft(); state = 0; delay(400);  state = 5;}
else if(state == '7'){turnRight(); state = 0;delay(400);  state = 5;}
```

```
else if (state == '8'){
if(servo1<180){servo1 = servo1+1;}
motor_1.write(servo1);  state = 0;
}
else if (state == '9'){
if(servo1>0){servo1 = servo1-1;}
motor_1.write(servo1);  state = 0;
}

else if (state == '10'){
if(servo2>0){servo2 = servo2-1;}
motor_2.write(servo2);state = 0;
}
else if (state == '11'){
if(servo2<180){servo2 = servo2+1;}
motor_2.write(servo2);  state = 0;
}

else if(state == '12'){
if(servo3>0){servo3 = servo3-1;}
motor_3.write(servo3);state = 0;
}
else if (state == '13'){
if(servo3<180){servo3 = servo3+1;}
motor_3.write(servo3);state = 0;
}

else if (state == '14'){
if(servo4<180){servo4 = servo4+1;}
motor_4.write(servo4);state = 0;
}
else if(state == '15'){
if(servo4>0){servo4 = servo4-1;}
motor_4.write(servo4);state = 0;
}
}

void forword(){  //forword
digitalWrite(in1, HIGH); //Right Motor forword Pin
digitalWrite(in2, LOW);  //Right Motor backword Pin
digitalWrite(in3, LOW);  //Left Motor backword Pin
digitalWrite(in4, HIGH); //Left Motor forword Pin
}
```

```
void backword(){ //backword
digitalWrite(in1, LOW);  //Right Motor forword Pin
digitalWrite(in2, HIGH); //Right Motor backword Pin
digitalWrite(in3, HIGH); //Left Motor backword Pin
digitalWrite(in4, LOW);  //Left Motor forword Pin
}

void turnRight(){ //turnRight
digitalWrite(in1, LOW);  //Right Motor forword Pin
digitalWrite(in2, HIGH); //Right Motor backword Pin
digitalWrite(in3, LOW);  //Left Motor backword Pin
digitalWrite(in4, HIGH); //Left Motor forword Pin
}

void turnLeft(){ //turnLeft
digitalWrite(in1, HIGH); //Right Motor forword Pin
digitalWrite(in2, LOW);  //Right Motor backword Pin
digitalWrite(in3, HIGH); //Left Motor backword Pin

digitalWrite(in4, LOW);  //Left Motor forword Pin
}

void Stop(){ //stop
digitalWrite(in1, LOW); //Right Motor forword Pin
digitalWrite(in2, LOW); //Right Motor backword Pin
digitalWrite(in3, LOW); //Left Motor backword Pin
digitalWrite(in4, LOW); //Left Motor forword Pin
}
```