DESIGN & IMPLEMENTATION OF AN AUTOMATED PRODUCT SORTING SYSTEM BASED ON SIZE, WEIGHT & MATERIAL USING A CONVEYOR BELT

A Project
by

Md. Mahadi Hassan	ME2201026169
Rezuwan Ahmed	ME2201026112
Farjana Islam	ME2201026089
Md. Maruf Hasan	ME2201026004
Md. Arafat Hossain	ME2201026335



Supervised By

Hasan Tareq Mahin

Lecturer

Department of Mechanical Engineering

Department of Mechanical Engineering

Sonargaon University (SU)

147/I, Green Road, Panthapath, Dhaka-1215, Bangladesh

September 2025

DESIGN & IMPLEMENTATION OF AN AUTOMATED PRODUCT SORTING SYSTEM BASED ON SIZE, WEIGHT & MATERIAL USING A CONVEYOR BELT

Submitted by

Md. Mahadi Hassan	ME2201026169
Rezuwan Ahmed	ME2201026112
Farjana Islam	ME2201026089
Md. Arafat Hossain	ME2201026335
Md. Maruf Hasan	ME2201026004

Supervised By

Hasan Tareq Mahin

Lecturer

Department of Mechanical Engineering

SUBMITTED TO THE:

Department of Mechanical Engineering

SONARGAON UNIVERSITY (SU)

In partial fulfillment of the requirement for award of the degree

of

Bachelor of Science in Mechanical Engineering

September 2025

DECLARATION

It is hereby solemnly declared that the work presented in this project report was carried out by us and has not been previously submitted to any university or organization for the award of any degree or certificate. It is hereby ensured that the work presented here does not breach any existing copyright. It is further undertaken that any loss or damage arising from a breach of the foregoing obligation shall be borne, and the university shall be fully indemnified.

[Authors]

Md. Mahadi Hassan
ME2201026169

Rezuwan Ahmed
ME2201026112

Farjana Islam
ME2201026089

Md. Arafat Hossain
ME2201026335

Md. Maruf Hasan
ME2201026004

ACKNOWLEDGEMENT

First, started in the name of almighty Allah. This thesis is accomplished under the supervision of **Hasan Tareq Mahin**, Lecturer, Department of Mechanical, Sonargaon University. It is a great pleasure to acknowledge profound gratitude and respect to the supervisor for this consistent guidance, encouragement, helpful suggestion, constructive criticism and endless patience through the progress of this work. The successful completion of this thesis would not have been possible without his persistent motivation and continuous guidance.

The authors are also grateful to **Professor Md. Mostofa Hossain**. Head of the Department of Mechanical Engineering and all respected teachers of the Mechanical Engineering Department for their co-operation and significant help for completing the thesis work successfully.

[Authors]

Md. Mahadi Hassan
ME2201026169

Rezuwan Ahmed
ME2201026112

Farjana Islam
ME2201026089

Md. Arafat Hossain
ME2201026335

Md. Maruf Hasan
ME2201026004

ABSTRACT

The Design and Implementation of an Automated Product Sorting System Based on Size, Weight, and Material Using a Conveyor Belt. Is an intelligent and fully automated solution designed to enhance industrial sorting operation. This system classifies products based on three key physical properties: size, weight, and material type (metal or non-metal), utilizing a conveyor belt system integrated with various sensors. While the ultrasonic sensor accurately measures their dimensions. A load cell with HX711 module evaluates the product's weight, and an inductive proximity sensor distinguishes between metallic and non-metallic items. An Arduino microcontroller serves as the brain of the system, processing real-time data from the sensors. Based on the analysis, the microcontroller activates servo motors to divert products into specific bins according to their category. This process is fast, precise, and eliminates the need for manual labor, thereby reducing human error and increasing operational efficiency. The system is particularly useful in packaging, logistics, recycling, and manufacturing industries, where accurate and high-speed sorting is critical. Its modular design allows for scalability and customization according to specific industrial needs. By automating the sorting process, this system provides a smart, efficient, and cost-effective approach to modern product handling.

TABLE OF CONTENTS

Declaration ii
Acknowledgement iii
Abstract iv
List of Figures vii
List of Table viii

CHAPTER 1 INTRODUCTION

1.1 Overview 1
1.2 Problem Statement 2
1.3 Objectives 3
1.4 Methodology 3
1.5 Work Flow Diagram 4
1.6 Report Overview 4

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction 5
2.2 Literature Review 5
2.3 Summary 8
2.4 List of Abbreviation……...8

CHAPTER 3 SYSTEM ARCHITECTURE

3.1 Introduction 9
3.2 Block Diagram 9
3.3 Circuit Diagram 10
3.4 Working Principle 10

CHAPTER 4 HARDWARE & SOFTWARE ANALYSIS

4.1 Introduction 11
4.2 Hardware & Software 11
4.3 Arduino Nano 11
4.4 Switch Mode Power Supply (SMPS) 15
4.5 Ultrasonic Sensor 17
4.6 Gear motor 29
4.7 Buck Converter 20
4.8 Servo Motor 21
4.9 Conveyor Belt 23
4.10 Proximity Sensor 25
4.11 Load Cell Module 27
4.12 LCD Display 28
4.13 Relay 30
4.14 Arduino IDE 32
4.15 Easy EDA 36
4.16 Final Project 37
4.17 Cost Analysis 38
4.18 Key Specification of the components in the project 39

CHAPTER 5 RESULT ANALYSIS & DISCUSSION

5.1 Steps to find results 40
5.2 Data Tables 41
5.3 Short Descriptions of the Quadratic Graphs 43
5.4 Discussion 45
5.5 Advantages 45
5.6 Applications 46
5.7 Limitations 46
5.8 Performance Analysis 46

CHAPTER 6 CONCLUSION

6.1 Conclusion.....47
6.2 Future Scope.....47

REFERENCE.....48
APPENDIX.....50

LIST OF FIGURES

Figure No. **Page No**

1.1 Product Sorting System.....1
1.2 Work Flow Diagram.....4
3.1 Block Diagram of our project.....9
3.2 Schematic Diagram of our project.....10
4.1 Arduino Nano.....13
4.2 Arduino Nano Schematic Diagram.....13
4.3 How Arduino Nano Looks like.....14
4.4 Microcontroller IC AT mega 328p.....15
4.5 SMPS.....16
4.6 SMPS Circuit.....17
4.7 Ultrasonic Sensor Working Procedure.....18
4.8 Ultrasonic Sensor Pinout.....19
4.9 DC Gear Motor.....20
4.10 DC-DC Buck Converter.....21
4.11 Servo Motor.....23
4.12 Schematic Diagram of Servo Motor.....24
4.13 Conveyor Belt.....25
4.14 Proximity sensor.....27
4.15 Load cell.....28
4.16 Load Cell Module.....29
4.17 16*2 LCD Display.....29
4.18 Pin out of 16*2 LCD Display.....30

4.19 16*2 LCD Display interfacing with Arduino 30
4.20 Relay 31
4.21 Pin Diagram of Relay Module 31
4.22 Main Voltage Connection 32
4.23 Arduino Software Interface IDE 34
4.24 Easy EDA Software Interface 37
4.25 Complete Project Prototype 38
4.26 Project prototype (Top View) 38
4.27 Project prototype (Front View) 39
4.28 Project prototype (Rear View) 39
5.1 Sample Product (Isometric View) 41
5.2 Height vs Detection Time Graph (Metal) 42
5.3 Weight vs Detection Time Graph (Metal) 42
5.4 Height vs Detection Time Graph (Non-Metal) 43
5.5 Weight vs Detection Time Graph (Non-Metal) 43

LIST OF TABLES

Table No. **Page No**

1 List of Components with Price 12
2 Pin Configuration of Ultrasonic Sensor 20
3 Key Specification of the Components Used In the Project 41
4 Specification of the Sample Product Used In the Project 41
5 Data Table for Metal Product Sorting 43
6 Data Table for Non - Metal Product Sorting 43
7 Data Table for faulty Product Sorting 44
8 Combination Table of sample Products (Real Time Data) 44

CHAPTER 1
INTRODUCTION

1.1 Overview

Manufacturing organizations are increasingly transitioning to industrial automation, which is becoming a global trend. Packaging and sorting procedures are among the most common in business. Automation of production is a result of control system development in industry. Production automation aims to run the entire system automatically and computerized, in addition to increasing productivity and lowering labor expenses. An industry's growth mostly hinges on how effectively it uses new, cutting-edge technologies. Every industry aspires to a higher rate of output. For this reason, every industry's key objective in order to advance in the cutthroat market is mass production. It is simple to make when the entire production process is totally automated. The industry is working to increase output rates in today's modern society. To achieve speedy production, everything is becoming automated. Economic growth, a more efficient division of labor, and a shift from relying on uncontrollable external forces to problem-solving through technical innovation are all characteristics of industrialization.



Figure 1.1: Product Sorting System

Every sector is experiencing rapid automation as industrial demands continue to evolve. Manufacturers now produce large volumes of products that may look similar but vary in Design and Implementation of an Automated Product Sorting System Based on Size, Weight, and

Material Using a Conveyor Belt type. In such cases, automated sorting plays a crucial role in accurately separating products based on specific physical parameters. This sorting not only improves the accuracy of grouping similar items but also enhances operational precision and consistency.

In traditional systems, sorting by hand is both time-consuming and prone to human error. Mechanical sorting methods, while somewhat faster, often suffer from delays and misclassification issues. To overcome these limitations, our project introduces a fully automated Design & Implementation of a Product Sorting System by size, weight & material using conveyor belt mechanism. This system utilizes sensors—such as ultrasonic for size measurement, load cell for weight evaluation, and inductive proximity sensor for material classification—to intelligently categorize each product.

Controlled by an Arduino microcontroller, the system processes real-time data and activates appropriate actuators to place items into their correct bins. As a result, the automated approach reduces sorting time, improves accuracy, and minimizes manual labor. This makes the system highly suitable for industries like packaging, manufacturing, and recycling, where precise sorting is essential for efficiency.

1.2 Problem Statement

In modern industries, sorting products manually based on size, weight, and material is time-consuming, labor-intensive, and prone to human error. Traditional mechanical systems often lack precision and adaptability, leading to inaccurate classification and reduced productivity. As production lines become faster and more diverse, the need for an automated, real-time sorting system becomes essential. Industries require a reliable solution that can detect and separate products with varying dimensions and materials efficiently. This project aims to develop an automated conveyor-based sorting system using sensors and microcontroller technology to enhance accuracy, reduce manual effort, and optimize industrial sorting operations.

1.3 Objectives

The objectives of this project are:

- To study, design and construct the Design and Implementation of an Automated Product Sorting System Based on Size, Weight, and Material Using a Conveyor Belt.
- To reduce human effort and minimize manual sorting errors.
- To use ultrasonic, load cell, and proximity sensors for accurate product detection.
- To improve sorting speed and precision in industrial environments.
- To categorize products into different bins automatically based on measured parameters.

1.4 Methodology

The project Methodology:

- Creating an idea for design and construction of a Design and Implementation of an Automated Product Sorting System Based on Size, Weight, and Material Using a Conveyor Belt. And designing a block diagram & circuit diagram to know which components we need to construct it.
- Collecting all the components and programming the micro controller to control the whole system.
- Setting up all the components in a PCB board & then soldering. Lastly, assembling all the blocks in a board and to run the system & for checking purposes.

1.5 Work Flow Diagram

The diagram the step-by-step development process of our automated sorting system project. It begins with the Study phase, where relevant technologies, sensors, and sorting mechanisms are researched. Next, in the Design stage, the circuit, mechanical setup, and flow logic are planned. Following this, the Components Collection step involves sourcing the required hardware such as Arduino, sensors, conveyor belt parts, and actuators. Then, in the Assembling phase, all components are interconnected and programmed. Finally, the Complete Project stage marks the successful integration and testing of the system, capable of sorting objects based on size, weight, and material.

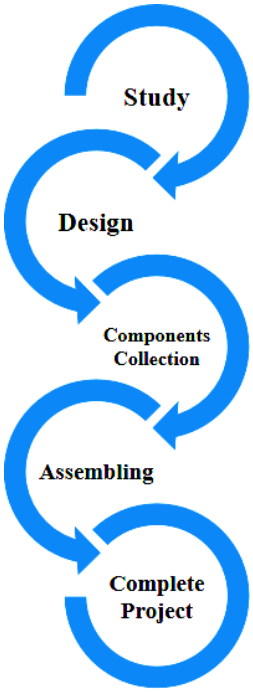


Figure 1.2: Work Flow Diagram

1.6 Report Overview

This project book consists of Six chapter. The first chapter contains the statement of the Introduction, Overview, Problem Statement, Work Flow Diagram, objectives of the study and Methodology. Chapter two contains system Literature e Review details. Chapter three System Architecture, Discussed in Block Diagram, Circuit Diagram, Working principle and Cost analysis. Chapter Four describes the hardware implementation with component details and the software which we have used for our work. Chapter Five deals with the result Analysis, shows the complete prototype, Advantage, Application and Limitation of the project that we have built. In the final chapter, we have discussed the, Future Scope, conclusion of the project.

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

In this chapter, the overview of the system is discussed. In this paper, the design and implementation of an automated product sorting system based on size, weight, and material using a conveyor belt are presented. In the following section, some previous literature is reviewed. From the existing literature, additional knowledge was obtained, which was then applied in the development of the project. As a result, the system was successfully completed.

2.2 Literature Review

Shen and Hassan [1] states that an approach for continuous recognition and the sorting of objects into their respective and desired location can be implemented as an image of colour processing that can attract an enormous attention leading to a possible widening scope of application in a different field in a modern technology. A colour-sorting robot is designed and developed using an Arduino Uno microcontroller, SG90 Tower Pro Servo Motor, TCS3200D colour sensor and several other electronic components. The system has the potential to sort the objects according to their colures into their relevant colour station in a less time. A distinct code for this system is developed.

According to Yunardi et al. [2], a 3D volume of the packed box is well quantified from the 2D images using the techniques of image processing. The 2D image consists of two images captured on the camera with a horizontal view and a vertical view. With the parameters, that is, the length, width and the height, a multiplication program is used to obtain the result of the volume. Consequently, contour-based object detection can be appertained to the automatic sorting system to measure the volume of an object in a computer-based vision.

According to Babita [3], sensors are embedded at several places that detects and senses the materials of various sizes that gets sorted at different stations based on their sizes. For sensing, the material used is an infrared sensor that is so sensitive. All the process is controlled and handled by a PLC.

Kulkarni et al. [4] has implemented the sorting of the boxes using a barcode, which is decoded by raspberry pi. This raspberry pi sends a signal to the motor driver to start and stop the motor accordingly. Proximity switches are implanted on every cylinder to sense the box position, which again sends back the signal to raspberry pi. Relay is used as an interface for the motor driver and motor. The relay used is of solid relay type. Raspberry pi uses cameras of five megapixels to capture the image, which will then be decoded

Sandesh Prajapati *et al.* [5] ,developed sorting of objects based on height, weight and color on a conveyor line. In the project, force resistive, ultrasonic, and TCS 230 color sensors were used. The entire apparatus was controlled by a microcontroller. S. V. Rautu *et al.* [6] have developed sorting of objects based on color, weight, and type on a conveyor line using PLC.
Here two types of sorting systems included, color and weight. The load cell is used for weight sorting and the TCS34725 color sensor is used for color detecting purposes. Overall process control by Siemens 300 series PLC.

Algitta *et al.* [7], developed a PLC-based automatic packaging process which is currently being employed in the manufacturing industry. This project's major objective is to develop a simple, compact conveyor belt system to automate the packaging of small cubic components. Data was provided to the controller using inductive and photoelectric sensors. Driven by electrical DC motors, the conveyor belt moves products. A ladder logic diagram is used to automate and manage the entire system using a programmable logic controller (PLC). S. N. Kulkarni *et al.* [8] put up a model for an automated system for sorting objects using a Raspberry Pi. This study uses a Raspberry Pi, a servo motor driver, a USB webcam, a MySQL database, a robotic arm, and an ultrasonic sensor. In this study, sorting operations are carried out using a Raspberry Pi.

Amin Nasiri *et al.* [9], developed an idea about sorting unwashed eggs using deep learning This paper suggests an innovative and accurate way for evaluating egg sorting using a deep convolutional neural network (CNN), a cutting-edge technique for performing classification problems in computer vision. Several layers, including a global average pooling layer, dense layers, a batch normalization layer, and a dropout layer, were added to the VGG16 architecture to help categorize photos of unwashed eggs. The modified model was trained using eggs that were clean, intact, bleeding, and fractured (having a break, crack, or hole in the eggshell). It was determined through 5-fold cross-validation that the CNN model outperformed conventional machine vision-based models.

Eied, simran *et al.* [10] developed an idea named development of automatic sorting conveyor belt using PLC. Here, the object is sort based on height and the sensor used for sensing the height is a diffused beam type photo-electric sensor. The development of a ladder logic program for the PLC that drives the pneumatic actuator used to sort the items is explained in the paper. Dezaki *et al.* [11] developed a pneumatic conveyor robot for color identification and sorting. Here, an autonomous, intelligent mechatronic conveyor system is developed to place and move circular goods in the manufacturing and packaging industries. The operation is run by a PLC, a color sensor, and an electronic switch. Ramachandran Natarajan *et al.* [12] developed the automatic weight sorting system by using a Load cell and Microcontroller. Here a feeding mechanism was used to feed the objects into the container according to their weight.

H. Hassan *et al.* [13] developed a prototype named A Low-Cost Automated Sorting Recycle Bin powered by an Arduino Microcontroller. Sorting is done in this location in order to recycle waste items like aluminum, paper, and plastic. A recycle bin is created that automatically separates various types of rubbish using an Arduino microcontroller. Varma *et al.* [14] an idea named Automatic Pneumatic Power Based Object Sorting System by Using MITSUBISHI PLC. Here, a MITSUBISHI PLC is used to create an automatic pneumatic power-based object sorting system. The primary goal of this project is to push the thing onto the conveyor belt once IR detectors determine its size. K. M. C. Babu *et al.* [15] developed a paper about Design and Development of a Cost-Effective Arduino based Object Sorting System. The Arduino NANO served as the hub of this study's color sorting system, bringing all the other parts together and allowing them to work together. A UV sensor is used to identify the object's presence. In this instance, objects are sorted using a TCS3200 sensor's color identification feature.

Harshita Borkar (2022) *et al.* [16] did the work on A Review on Arduino Based Color Sorting Machine. Here Arduino is the core element and, in this work, they showed by using Arduino a more feasible output is found easily. Nilima Bargal *et al*. [17] have developed a project about object sorting system using PLC. They have created an LCA system that uses DC geared motors to sort light things according to height difference. The Programmable Logic Controller (PLC) passes the object in front of the sensors, which subsequently detect it and decide the sorting logic. M.M. Sofu *et al.* [18] developed an idea name design of an automatic apple sorting system using machine vision. This study suggests a real-time processing-based automated apple sorting and quality checking system. Apple cultivars such as Granny Smith, Golden and Starking Delicious are divided into many classes based on their size, weight, and

color. Additionally, it can identify apples with rot, taint, and scab. The suggested system consists of conveyors for rollers, transporters, and classes along with an enclosed cabin that has load cell, control panel, and machine vision units.

2.3 Summary

From the literature discussed above, we have gained a lot of knowledge and inspiration to implement this project. The project has been successfully completed through the tireless efforts of all concerned.

2.4 List of Abbreviation

SMPS: Switch Mode Power Supply
SRAM: Static Random-Access Memory
EEPROM: Electrically Erasable Programmable Read-only Memory
LCD: Liquid Crystal Display
VCC: Voltage Common Collector

CHAPTER 3

SYSTEM ARCHITECTURE

3.1 Introduction

The project objectives, methods, literature evaluation, and other details were all clarified in the preceding chapter. The block diagram, circuit diagram, working principle, Project final Image and cost analysis will all be covered in this chapter.

3.2 Block Diagram

Here is the block diagram of the **Design and Implementation of an Automated Product Sorting System Based on Size, Weight, and Material Using a Conveyor Belt.** with all the essential components. All of the components are shown in below as a block in this diagram. Here we use Arduino Nano Micro-controller of our project. Other equipment's are attached with this micro-controller and work together for perform as our desire outcome.

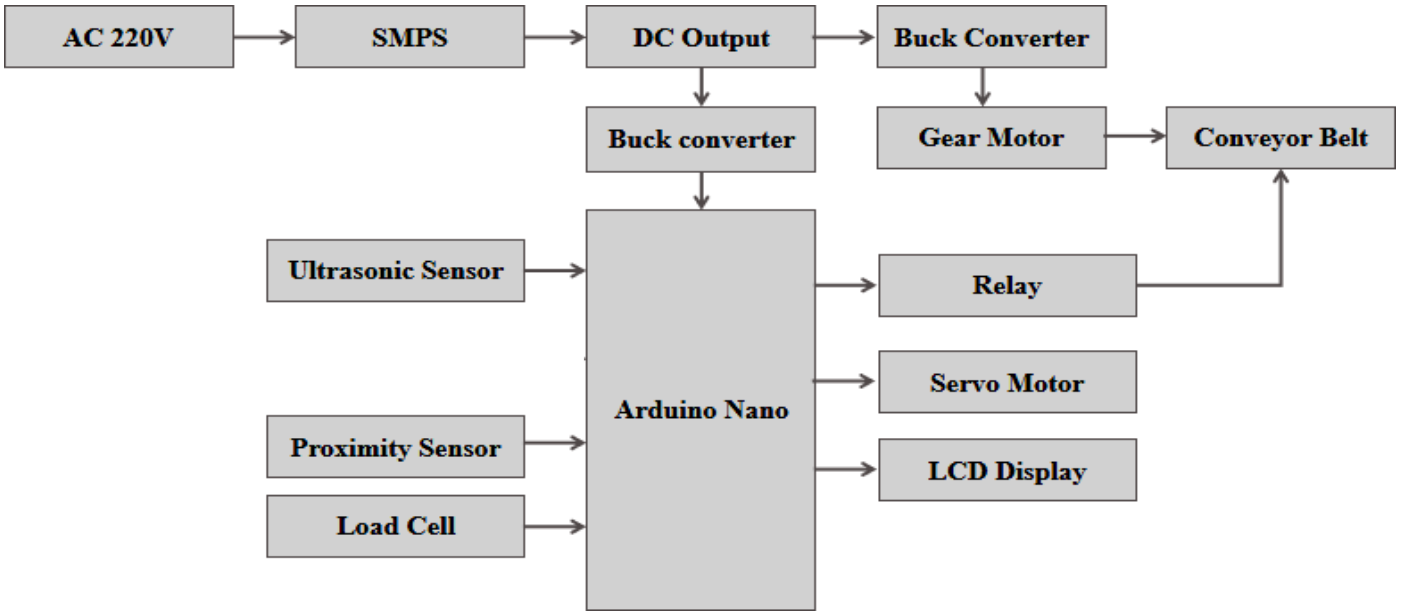


Figure 3.1: Block Diagram of the Project

3.3 Circuit Diagram

In this part show the project circuit design and connect out instrument through standard wire

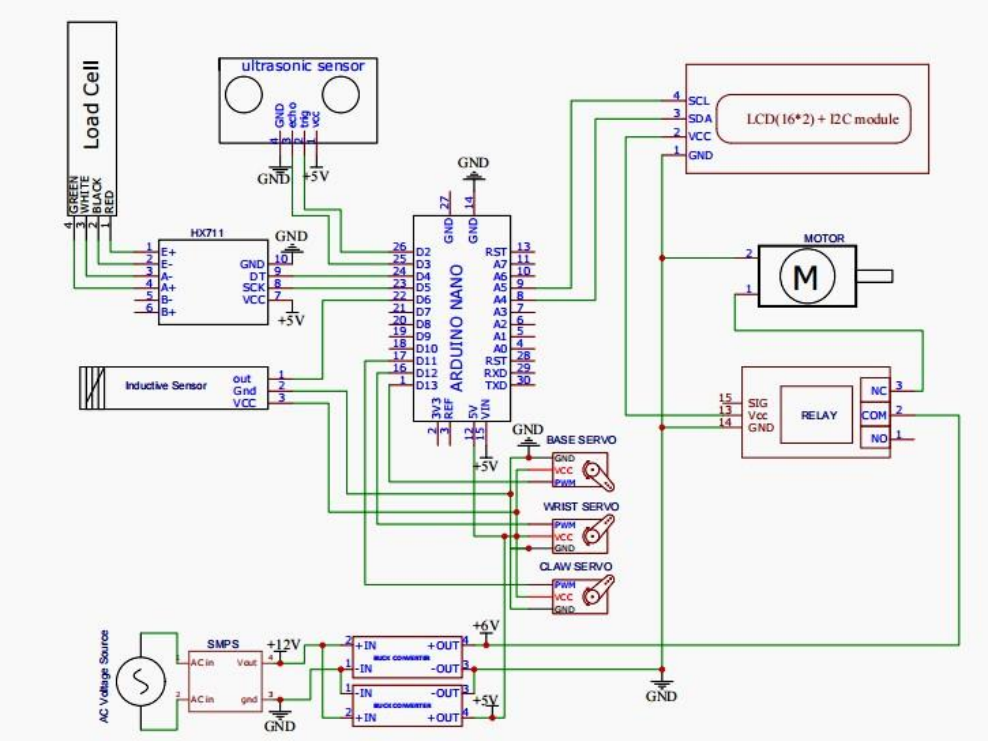


Figure 3.2: Schematic Design of The Project

3.4 Working principle

The system operates through a structured sequence of sensing, processing, and sorting. A SMPS supplies the main 12V DC power, while buck converters provide stable voltage to the Arduino Nano, sensors. When a product is placed on the conveyor belt (driven by a gear motor controlled via a relay), proximity sensor detects its arrival. The ultrasonic sensor measures the product's size, and the load cell module evaluates its weight. Simultaneously, the inductive proximity sensor identifies whether the object is metal or non-metal. All sensor data are processed by the Arduino Nano, which makes real-time sorting decisions.

Based on the readings, the servo motor is activated to direct the item into the correct bin. The system's status, like weight or material type, is shown on the LCD display. An additional buck converter ensures isolated power to the gear motor system for conveyor control, ensuring efficient and reliable sorting operations with precision and automation

CHAPTER 4
HARDWARE & SOFTWARE ANALYSIS

4.1 Introduction

In this section, an discuss elaborately about **Design and Implementation of an Automated Product Sorting System Based on Size, Weight, and Material Using a Conveyor Belt.** And the component description, features, working procedure of our all equipment. The system hardware fabricates composed of micro-controller unit, power unit, source unit, power store unites, sensor unit and many more related components.

4.2 Hardware & Software

Hardware

- Arduino Nano
- SMPS
- Buck Converter
- Gear Motor
- Ultrasonic Sensor
- Proximity Sensor
- Load Cell Module
- Relay
- LCD Display
- Servo Motor
- Conveyor Belt

Software

- Arduino IDE
- Easy EDA

4.3 Arduino Nano

Arduino is open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling Lights, motors,

.

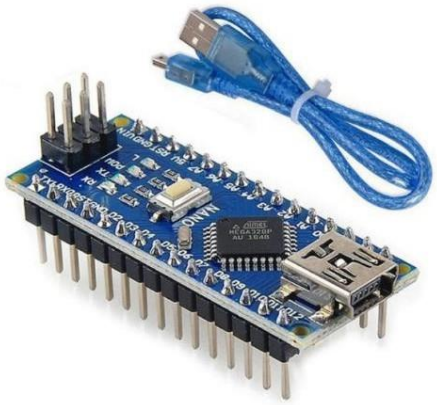


Figure 4.1: Arduino Nano

The micro-controller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, Maxims'). Arduino Nano is a surface mount breadboard embedded version with integrated USB. It is a small, complete, and breadboard friendly component. It has everything that Decimal/ Duemilanove has (electrically) with more analog input pins and onboard +5V AREF jumper. Physically, it is missing power jack. The Nano can automatically sense and switch to the higher potential source of power.

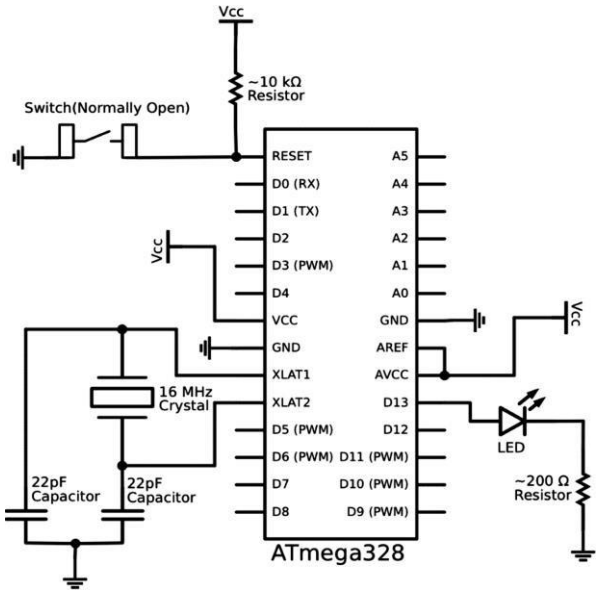


Figure 4.2: Arduino Nano Schematic Diagram

Nano's got the breadboard-ability of the Boarding and the Minibus with smaller footprint than either, so users have more breadboard space. It's got a pin layout that works well with the Mini or the Basic Stamp (TX, RX, ATN, and GND on one top, power and ground on the other). This new version 3.0 comes with ATMEGA328 which offer more programming and data memory space. It has two layers. That make it easier to hack and more affordable. One of the best features of Arduino Nano is, it's easy to use, compact and also small.

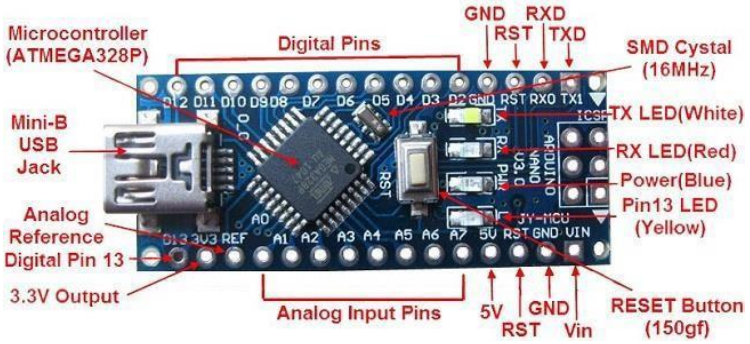


Figure 4.3: How Arduino Nano looks like

Specifications

- Micro-controller: Atmel ATmega328
- Operating Voltage (logic level): 5 V
- Input Voltage (recommended): 7-12 V
- Input Voltage (limits): 6-20 V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 8
- DC Current per I/O Pin: 40 mA
- Flash Memory: 32 KB (of which 2KB used by boot loader)
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Dimensions: 0.70" x 1.70"

Features

- Automatic reset during program download
- Power OK blue LED
- Green (TX), red (RX) and orange (L) LED

- Auto sensing/switching power input
- Small mini-B USB for programming and serial monitor
- ICSP header for direct program download
- Manual reset switch

Micro-controller IC ATmega328p

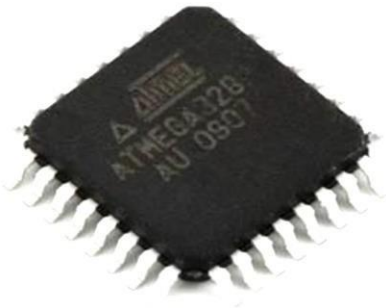


Figure 4.4: Micro-controller IC AT Mega 328p

The high-performance Microchip Pico Power 8-bit AVR RISC-based micro-controller combines 32KB ISP flash memory with read-while-write capabilities, 1024B EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, a 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. By executing powerful instructions in a single clock cycle, the device achieves throughput's approaching 1 MIPS per MHz, balancing power consumption and processing speed.

4.4 Switch Mode Power Supply (SMPS)

A switched-mode power supply (switching-mode power supply, switch-mode power supply, switched power supply, SMPS, or switcher) is an electronic power supply that incorporates a switching regulator to convert electrical power efficiently. Like other power supplies, an SMPS transfers power from a DC or AC source (often mains power) to DC loads,

such as a personal computer, while converting voltage and current characteristics. Unlike a linear power supply, the pass transistor of a switching-mode supply continually switches between low-dissipation, full-on and full-off states, and spends very little time in the high dissipation transitions, which minimizes wasted energy. A hypothetical ideal switched-mode power supply dissipates no power. Voltage regulation is achieved by varying the ratio of on-to-off time (also known as duty cycles). In contrast, a linear power supply regulates the output voltage by continually dissipating power in the pass transistor. This higher power conversion efficiency is an important advantage of a switched-mode power supply. Switched-mode power supplies may also be substantially smaller and lighter than a linear supply due to the smaller transformer size and weight.



Figure 4.5: SMPS

Switching regulators are used as replacements for linear regulators when higher efficiency, smaller size or lighter weight is required. They are, however, more complicated; their switching currents can cause electrical noise problems if not carefully suppressed, and simple designs may have a poor power factor. Switched-mode power supplies are classified according to the type of input and output voltages.

A basic isolated AC to DC switched-mode power supply consists of:

- Input rectifier and filter
- Inverter consisting of switching devices such as MOSFETs
- Transformer
- Output rectifier and filter
- Feedback and control circuit

The input DC supply from a rectifier or battery is fed to the inverter where it is turned on and off at high frequencies of between 20 KHz and 200 KHz by the switching MOSFET or power transistors. The high-frequency voltage pulses from the inverter are fed to the transformer primary winding, and the secondary AC output is rectified and smoothed to produce the required DC voltages. A feedback circuit monitors the output voltage and instructs the control circuit to adjust the duty cycle to maintain the output at the desired level.

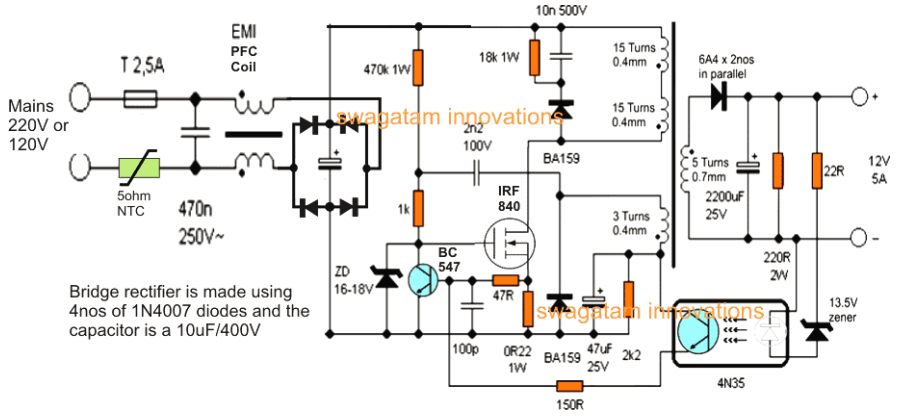


Figure 4.6: SMPS Circuit

Advantages of switched-mode power supplies:

- Higher efficiency of 68% to 90%
- Regulated and reliable outputs regardless of variations in input supply voltage
- Small size and lighter
- Flexible technology
- High power density

Disadvantages:

- Generates electromagnetic interference
- Complex circuit design
- Expensive compared to linear supplies

4.5 Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound.

HC-SR04 Ultrasonic Sensor - Working

As shown above the HC-SR04 Ultrasonic (US) sensor is a 4-pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications were measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that –

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

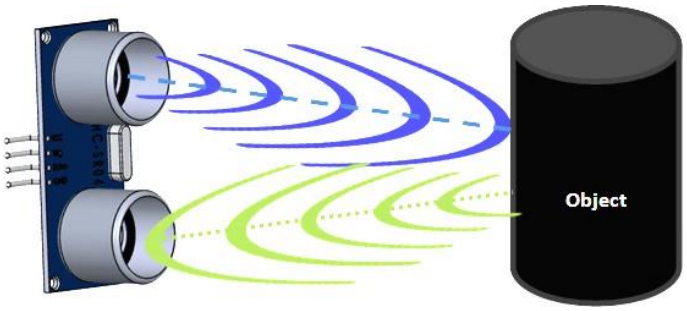


Figure 4.7: Ultrasonic Sensor Working Procedure

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave, we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a micro-controller or microprocessor.

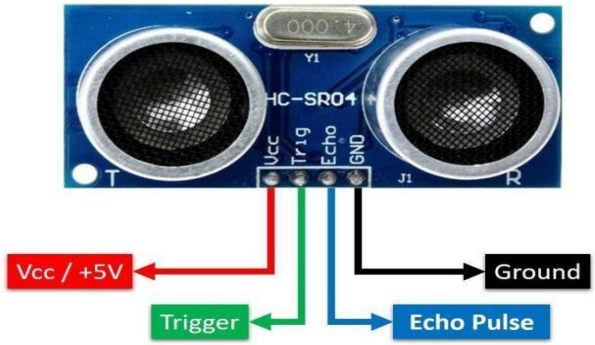


Figure 4.08: Ultrasonic Sensor Pin Out

Ultrasonic Sensor Pin Configuration

Table 2: Pin Configuration of Ultrasonic Sensor

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40H

4.6 Gear Motor

A DC motor is any motor within a class of electrical machines whereby direct current electrical power is converted into mechanical power. A 12v DC motor is small and inexpensive, yet powerful enough to be used for many applications.



Figure 4.09: DC Gear Motor

Specification

- Voltage: 12V DC
- Gear ratio: 1/31
- No-load speed: 200 RPM
- Rated Speed: 140 RPM
- Rated torque: 10 kg.cm
- Rated current: 2.5 Amp
- Length of Motor (including spindle): 106 mm/4.17″
- Diameter: 37 mm/1.45″
- Shaft length: 21 mm/0.82″
- Shaft diameter: 6 mm/0.24″

4.7 Buck Converter

A buck converter (step-down converter) is a DC-to-DC power converter which steps down voltage (while drawing less average current) from its input (supply) to its output (load). It is a class of switched-mode power supply (SMPS) typically containing at least two semiconductors (a diode and a transistor, although modern buck converters frequently replace the diode with a second transistor used for synchronous rectification) and at least one energy storage element,

a capacitor, inductor, or the two in combination. To reduce voltage ripple, filters made of capacitors (sometimes in combination with inductors) are normally added to such a converter's output (load-side filter) and input (supply-side filter). It is called a buck converter because the voltage across the inductor “bucks” or opposes the supply voltage.



Figure 4.10: DC -DC Buck Converter

DC-DC Buck Converter Step Down Module LM2596 Power Supply is a step-down (buck) switching regulator, capable of driving a 3-A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3 V, 5 V, 12 V, and an adjustable output version. The LM2596 series operates at a switching frequency of 150kHz, thus allowing smaller sized filter components than what would be required with lower frequency switching regulators.

Specifications of DC-DC Buck Converter Step Down Module LM2596 Power Supply

- Conversion efficiency: 92% (highest)
- Switching frequency: 150KHz
- Output ripple: 30mA9maximum)
- Load Regulation: ± 0.5%
- Voltage Regulation: ± 0.5%
- Dynamic Response speed: 5% 200uS
- Input voltage:4.75-35V
- Output voltage:1.25-26V(Adjustable)
- Output current: Rated current is 2A, maximum 3A (Additional heat sink is required)
- Conversion Efficiency: Up to 92% (output voltage higher, the higher the efficiency)
- Switching Frequency: 150KHz

- Rectifier: Non-Synchronous Rectification
- Module Properties: Non-isolated step-down module (buck)
- Short Circuit Protection: Current limiting, since the recovery
- Operating Temperature: Industrial grade (-40 to +85) (output power 10W or less)

4.8 Servo Motor

A servo motor is an electrical device that can push or rotate an object with great precision. If you want to rotate and object at certain angles or distances, you use servo motors. It is made by a simple motor which is driven by a servo mechanism. If the motor is DC driven then it is called DC servo motor and if it is AC driven motor then it is called AC servo motor. We can get a very high torque servo motor in a small and light weight packages. These are being used in various applications like toy cars, RC helicopters and planes, robotics, machines etc.



Figure 4.11: Servo Motor

Servo motors are rated at kg / cm (centimeters per kilogram). Most hobby servo motors are rated at 3 kg / cm or 6 kg / cm or 12 kg / cm. This kg / centimeter tells you how much weight your servo motor can lift at a certain distance. For example: A 6 kg / cm servo motor should be able to lift 6 kg if the load is suspended 1 cm away from the motor shaft, to be less than the carrying capacity. The position of the servo motor is determined by the electric pulse and its circuitry is placed next to the motor.

Servo Mechanism

It consists of three parts:

1. Controlled device
2. Output sensor
3. Feedback system

All motors have three wires. Of which two will be used for the supply (positive and negative) and one will be used for the signal transmitted from the MCU. The servo motor is controlled by PWM (Pulse with Modulation) which provides the control wires. Has minimum pulse, maximum pulse and repetition rate. The servo motor can rotate 90 degrees in two directions, creating its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the pulse length will determine how far the motor rotates. For example, a 1.5 mm pulse will turn the motor to 90 °, for example, if the branch is less than 1.5 mm shaft, it will move to 0 to, and if it is longer than 1.5 mm, the Servo will turn to 180 to.

The servo motor operates on the PWM (pulse width modulation) principle, meaning that the angle of rotation is controlled by the duration of the pulse applied to its control pin. Basically, the servo motor is made up of a DC motor which is controlled by a variable resistor (potentiometer) and some gears. The high-speed force of the DC motor is converted to torque by the gears. We know that work = force x DISTANCE, less in DC motor force and more in distance (speed) and in servo, forces are higher and distance is less. The pentameter server is connected to the output trench to calculate the angle and stop the DC motor at the required angle.

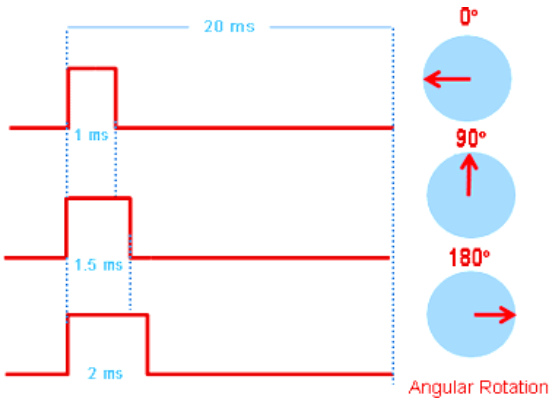


Figure 4.12: Schematic Diagram of servo motor

The servo motor can be rotated from 0 to 180 degrees but depending on the output it can go up to 210 degrees. This degree of rotation can be controlled by applying an electric pulse of appropriate width to its control pin. Servo examines pulses every 20 milliseconds. The 1 ms (1 millisecond) width pulse servo can rotate 0 degrees, 1.5 mm 90 degrees (neutral position) and 2 ms pulse it can rotate 180 degrees. All servo motors work directly with your + 5V supply

rails but we need to be careful about how much current the motor will use. If you plan to use more than two servo motors, you should design a suitable servo solder.

4.9 Conveyor Belt

A conveyor belt is the carrying medium of a belt conveyor system (often shortened to belt conveyor). A belt conveyor system is one of many types of conveyor systems. A belt conveyor system consists of two or more pulleys (sometimes referred to as drums), with a closed loop of carrying medium—the conveyor belt—that rotates about them. One or both of the pulleys are powered, moving the belt and the material on the belt forward. The powered pulley is called the drive pulley while the unpowered pulley is called the idler pulley.

Overview

Conveyors are durable and reliable components used in automated distribution and warehousing, as well as manufacturing and production facilities. In combination with computer-controlled pallet handling equipment this allows for more efficient retail, wholesale, and manufacturing distribution. It is considered a labor-saving system that allows large volumes to move rapidly through a process, allowing companies to ship or receive higher volumes with smaller storage space and with less labor expense.

Belt conveyors are the most commonly used powered conveyors because they are the most versatile and the least expensive. Products are conveyed directly on the belt so both regular and irregular shaped objects, large or small, light and heavy, can be transported successfully. Belt conveyors are also manufactured with curved sections that use tapered rollers and curved belting to convey products around a corner. These conveyor systems are commonly used in postal sorting offices and airport baggage handling systems.

Types of Conveyor Belt

“There are three different types of conveyor belts: the basic belt, snake sandwich belt and long belt. A basic belt conveyor consists of two or more pulleys that hold one continuous length of material. These types of belts can be motorized or require manual effort.

Used in the Project

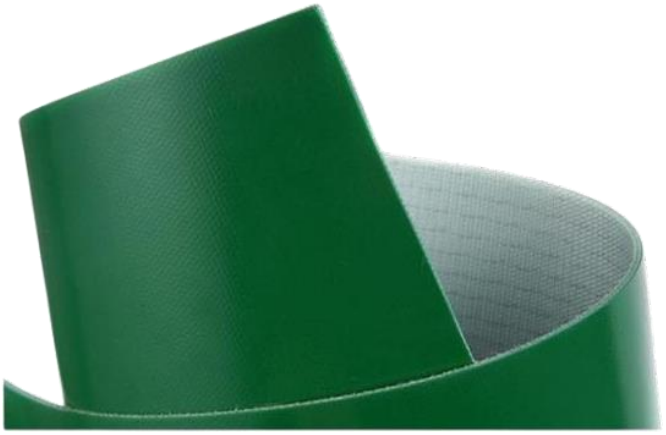


Figure 4.13: Conveyor Belt

Applications

Today there are different types of conveyor belts that have been created for conveying different kinds of material available in PVC and rubber materials. Material flowing over the belt may be weighed in transit using a belt weigher. Belts with regularly spaced partitions, known as elevator belts, are used for transporting loose materials up steep inclines. Belt Conveyors are used in self-unloading bulk freighters and in live bottom trucks. Belt conveyor technology is also used in conveyor transport such as moving sidewalks or escalators, as well as on many manufacturing assembly lines. Stores often have conveyor belts at the check-out counter to move shopping items, and may use checkout dividers in this process. Ski areas also use conveyor belts to transport skiers up the hill. Industrial and manufacturing applications for belt conveyors include package handling, trough belt conveyors, trash handling, bag handling,

coding conveyors, and more. Integration of Human-Machine Interface (HMI) to operate the conveyor system is in the developing stages and will prove to be an efficient innovation.

4.10 Proximity Sensor

A proximity sensor is a sensor able to detect the presence of nearby objects without any physical contact. A proximity sensor often emits an electromagnetic field or a beam of electromagnetic radiation (infrared, for instance), and looks for changes in the field or return signal. The object being sensed is often referred to as the proximity sensor's target. Different proximity sensor targets demand different sensors. For example, a capacitive proximity sensor or photoelectric sensor might be suitable for a plastic target; an inductive proximity sensor always requires a metal target. Proximity sensors can have a high reliability and long functional life because of the absence of mechanical parts and lack of physical contact between the sensor and the sensed object. Proximity sensors are also used in machine vibration monitoring to measure the variation in distance between a shaft and its support bearing. This is common in large steam turbines, compressors, and motors that use sleeve-type bearings.

Types of Proximity Sensor

- Inductive Proximity Sensor.
- Optical Proximity Sensor.
- Capacitive Proximity Sensor.
- Magnetic Proximity Sensor.
- Ultrasonic proximity Sensor.



Fig 4.14: Proximity Sensor

Proximity sensors are suitable for damp conditions and wide temperature range usage, unlike your traditional optical detection. Proximity sensors are also applicable in phones as well, be it your Android or IOS devices. It consists of simple IR technology that switches on and off display accordingly to your usage. Proximity sensors are used in phones, recycling plants, self-driving cars, anti-aircraft systems, and assembly lines. There are many types of proximity sensors, and they each sense targets in distinct ways.

- Model: LJ12A3-4-Z/BY.
- Wire Type - Cylindrical DC 3 Wire Type.
- Switch Appearance Type - Cylinder Type.
- Theory - Inductive Sensor.
- Output Type - PNP NO (Normal Open).
- Diameter of Head - 12mm.
- Detecting Distance - 4mm.
- Supply Voltage - DC 6-36V.
- Current Output - 300mA.
- Response Frequency - 0.5KHz.
- Detect Object - Iron.
- Operating Temperature -25°C to +55°C (Non-freezing Condition).
- Size 6.2 x 2cm/2.4" x 0.8" (L*Max. Dia).
- Cable Length 110cm/43.3".
- External Material Plastic, Alloy.
- Net Weight 46g.

4.11 Load Cell Module

A load cell is a force sensing module - a carefully designed metal structure, with small elements called strain gauges mounted in precise locations on the structure. Load cells are designed to measure a specific force, and ignore other forces being

applied. The electrical signal output by the load cell is very small and requires specialized amplification.



Figure 4.15: Load Cell

load cells are the kind most often found in industrial settings. It is ideal as it is highly accurate, versatile, and cost-effective. Structurally, a load cell has a metal body to which strain gauges have been secured. The body is usually made of aluminum, alloy steel, or stainless steel which makes it very sturdy but also minimally elastic. This elasticity gives rise to the term "spring element", referring to the body of the load cell. When force is exerted on the load cell, the spring element is slightly deformed, and unless overloaded, always returns to its original shape. As the spring element deforms, the strain gauges also change shape. The resulting alteration to the resistance in the strain gauges can be measured as voltage. The change in voltage is proportional to the amount of force applied to the cell, thus the amount of force can be calculated from the load cell's output.



Figure 4.16: Load Cell Module

Specifications

- Voltage range: 2.7-5V DC
- Nominal power consumption: 1.6mA
- Power consumption in sleep mode: <1uA
- 24-bit accuracy
- Possibility for two load cells
- Dimensions: 35mm x 21mm x 3mm
- Male pin headers included (10 pins)

4.12 LCD Display

The LCD (liquid crystal display) screen is an electronic display module and looks for various applications. A 16x2 LCD display is a very basic module and it is very commonly used in various devices and circuits. These modules are preferred over seven sections and many other segments.



Figure 4.17: 16*2 LCD Display

The reasons for having LCD are economic; Easily programmable, special and even custom characters (different in seven sections), there are no restrictions on displaying animations. A 16x2 LCD means it can display 16 characters per line and contains 2 lines. Each character on this LCD is displayed in a 5x7 pixel matrix.

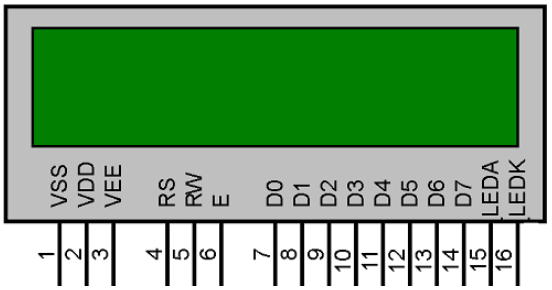


Figure 4.18: Pin out of 16*2 LCD Display

This LCD contains two articles called Command and Data. The command register stores the command instructions on the LCD. The command is a command given to the LCD to perform a predefined task such as starting it, clearing its screen, locating the cursor, controlling the display, etc.

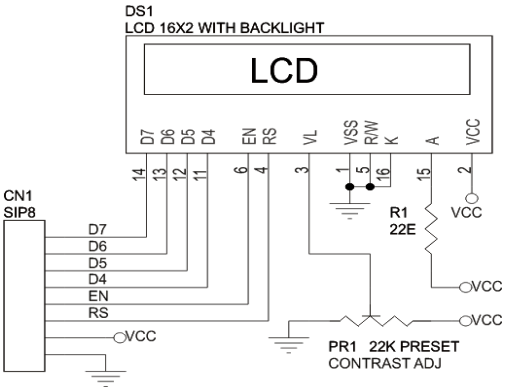


Figure 4.19: 16*2 LCD Display interfacing with Arduino

Feature

The features of this LCD mainly include the following.

- The operating voltage of this LCD is 4.7V-5.3V
- It includes two rows where each row can produce 16-characters.
- The utilization of current is 1mA with no backlight
- Every character can be built with a 5×8-pixel box
- The alphanumeric LCDs alphabets & numbers
- Is display can work on two modes like 4-bit & 8-bit
- These are obtainable in Blue & Green Backlight

4.13 Relay

A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers: they repeated the signal coming in from one circuit and re-transmitted it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

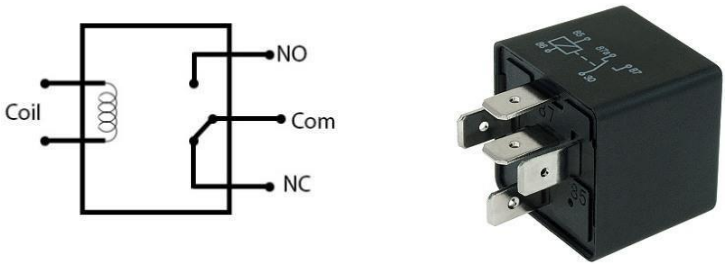


Figure 4.20: Relay

Mains voltage connections:

In relation to mains voltage, relays have 3 possible connections:

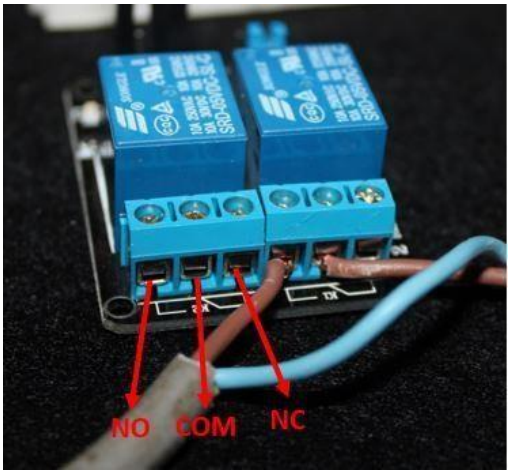


Figure 4.21: Pin diagram of Relay Module

COM: Common pin

NO (Normally Open): There is no contact between the common pin and the normally open pin. So, when you trigger the relay, it connects to the COM pin and supply is provided to a load

NC (Normally Closed): There is contact between the common pin and the normally closed pin. There is always connection between the COM and NC pins, even when the relay is turned off. When you trigger the relay, the circuit is opened and there is no supply provided to a load.

Pin wiring:

The connections between the relay module and the Arduino are really simple:

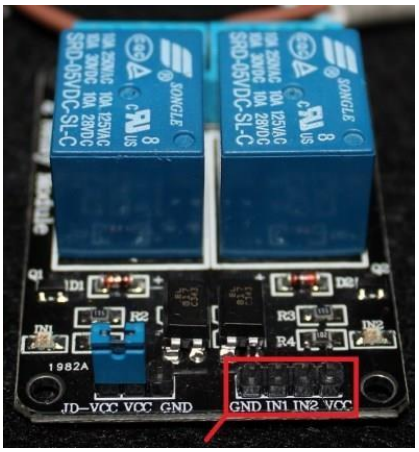


Figure 4.22: Main Voltage Connection

- **GND**: Goes to ground
- **IN1**: Controls the first relay (it will be connected to an Arduino digital pin)
- **IN2**: Controls the second relay (it should be connected to an Arduino digital pin if you are using this second relay. Otherwise, you don't need to connect it)
- **VCC**: Goes to 5V

SOFTWARE

4.14 Arduino IDE

The digital microcontroller unit named as Arduino UNO can be programmed with the Arduino software IDE. There is no any requirement for installing other software rather than Arduino. Firstly, Select "Arduino UNO from the Tools, Board menu (according to the microcontroller on our board). The IC used to name as ATmega328 on the Arduino UNO comes pre burned with a boot loader that allows us to upload new code to it without the use of an external hardware programmer.

Communication is using the original STK500 protocol (reference, C header files). We can also bypass the boot loader and programs the microcontroller through the ICSP (In Circuit Serial Programming) header. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. The Arduino UNO is one of the latest digital microcontroller units and has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL at (5V) with serial communication, which is available on digital pins 0 -(RX) for receive the data and pin no.1 (TX) for transmit the data. An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an .in file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board.

The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial Communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Uno digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. Arduino programs are written in C or C++ and the program code written for Arduino is called sketch. The Arduino IDE uses the GNU tool chain and AVR Lab to compile programs, and for uploading the programs it uses argued. As the Arduino platform uses Atmel microcontrollers, Atmel's development environment, AVR Studio or the newer Atmel Studio, may also be used to develop software for the Arduino.

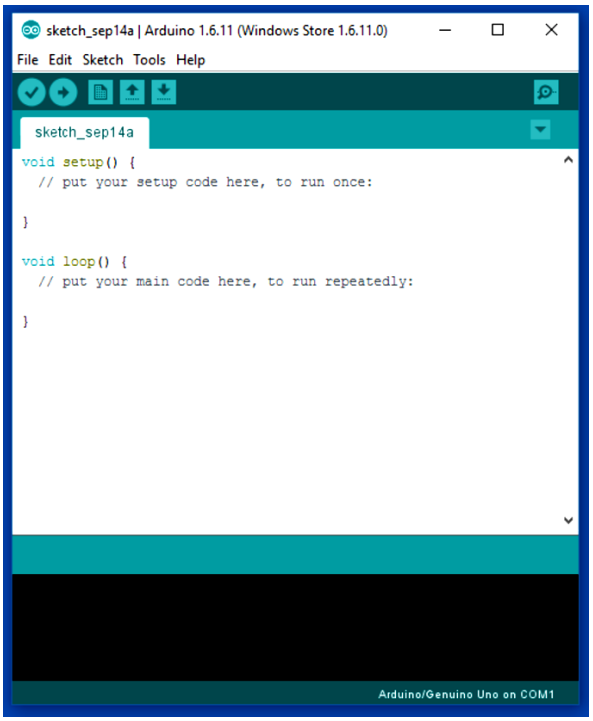


Figure 4.23: Arduino Software Interface IDE

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 Genuine hardware to upload programs and communicate with them.

Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension. 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.

Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino

software; it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog. Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the. pde extension. You may still open.pde named files in version 1.0 and later, the software will automatically rename the extension to.ino.

Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the sports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx, /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino boot loader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The boot loader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

Third-Party Hardware

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory To uninstall, simply delete its directory. For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd party Hardware specification.

Serial Monitor

This displays serial sent from the Arduino or Genuine board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to Serial. Begin in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

4.15 Easy EDA

Easy EDA is a web-based EDA tool suite that enables hardware engineers to design, simulate, share-publicly and privately-and discuss schematics, simulations and printed circuit boards. Other features include the creation of a bill of materials, Gerber files and pick and place files and documentary outputs in PDF, PNG and SVG formats. Easy EDA allows the creation and

editing of schematic diagrams, SPICE simulation of mixed analogue and digital circuits and the creation and editing of printed circuit board layouts and, optionally, the manufacture of printed circuit boards.

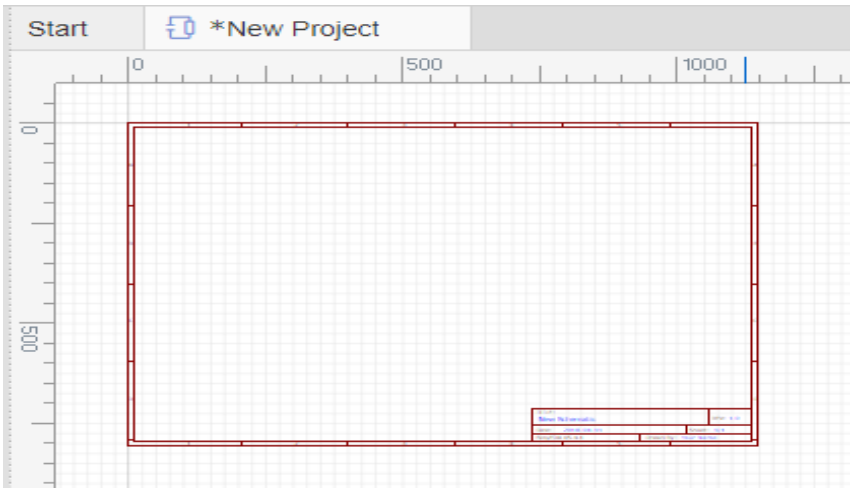


Figure 4.24: Easy EDA Software Interface

Subscription-free membership is offered for public plus a limited number of private projects. The number of private projects can be increased by contributing high quality public projects, schematic symbols, and PCB footprints and/or by paying a monthly subscription. Registered users can download Gerber files from the tool free of charge; but for a fee, Easy EDA offers a PCB fabrication service. This service is also able to accept Gerber file inputs from third party tools.

4.16 Final Project



Figure 4.25: Complete Project Prototype



Figure 4.26: Project Prototype (Top View)

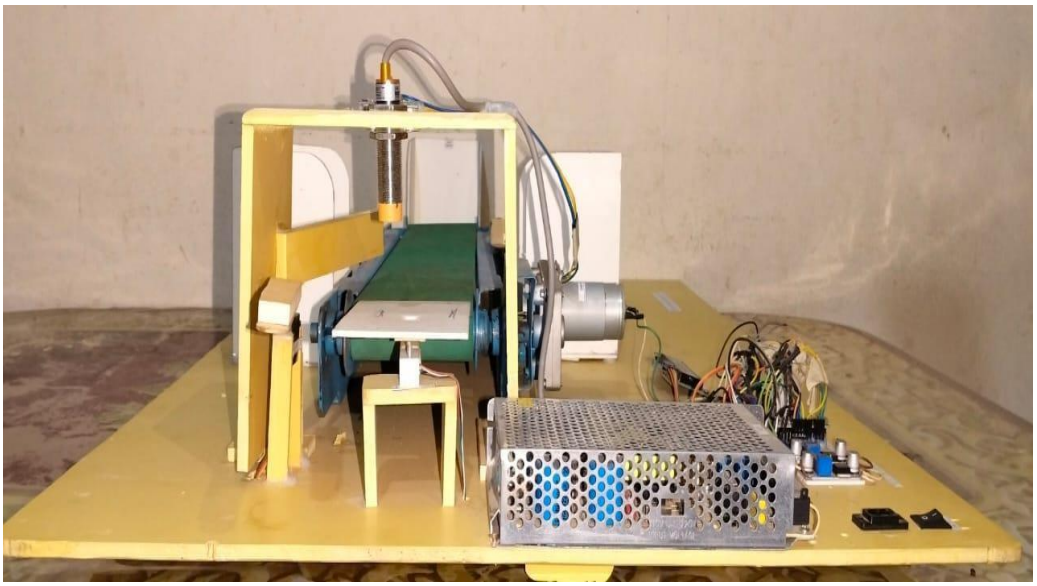


Figure 4.27: Project Prototype (Front View)

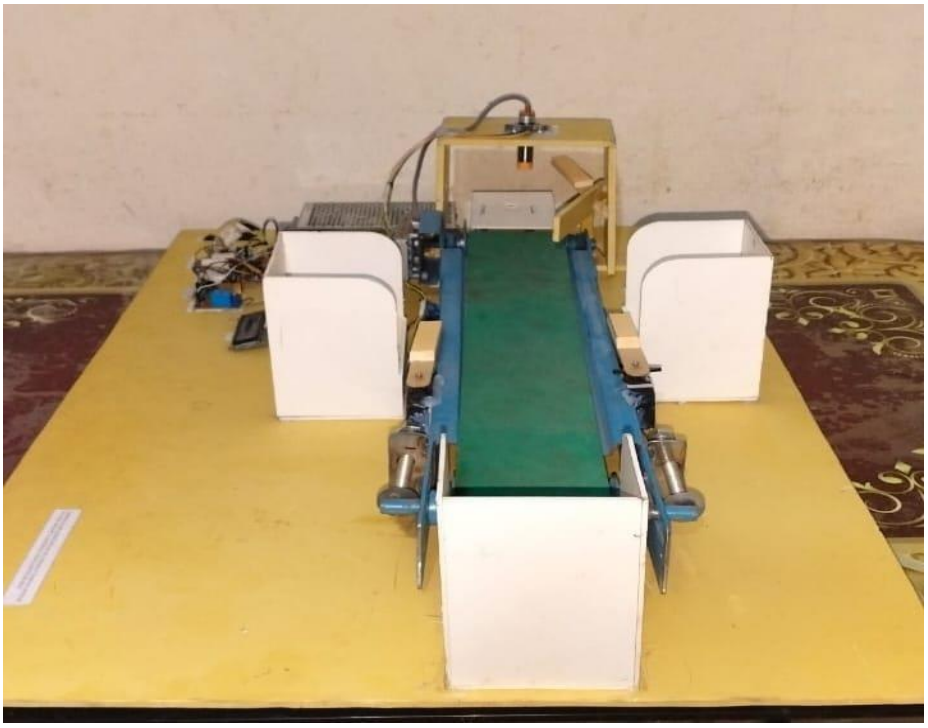


Figure 4.28: Project Prototype (Rear View)

4.17 Cost Analysis

Table 1: List of Components with Price

Sl. no	Particulars	Specification	Qty.	Unit Price (Taka)	Total Price (Taka)
1	Arduino Nano	ATmega328p	1	600	600
2	SMPS	12V	1	850	850
3	Buck Converter	LM2596	2	200	400
4	Gear Motor	775 DC	1	2000	2000
5	Servo Motor	MG90	3	300	900
6	Ultrasonic Sensor		1	350	350
7	Proximity Sensor	Inductive	1	550	550
8	Load Cell module	HX711	1	600	600
9	Relay	5V DC	1	150	150
10	LCD Display	16*2	1	400	400
11	Conveyor Belt			500	500
12	Others				3500
				Total	10,800/=

Table 3: Key Specification of the Components Used In the Project

Sl	Name	Qty (pc)	Specification
1	Conveyer belt	1	Length 58 cm X Width 10 cm
2	Load Cell	1	Max capacity 200 gm; Detecting time: 3 Second
3	Servo motor with arm	3	Max load capacity 200 gm/cm
4	Bin	3	Length 15 cm X Width 10 cm
5	Inductive Proximity Sensor	1	Detecting distance: 4 mm; Detecting time: 3 second
6	Ultrasonic Sensor	1	Detection time: 2 Second, Tolerance: 2 cm
7	Sample Product (Cubic shape)	11	Metal and non-metal with minimum height of 2 cm and maximum 5.1 cm. Minimum weight 6 gm and maximum is 99gm

Table 4: Specification of the Sample Product Used In the Project

Sl no	Sample Symbol	Type of Product	Height (cm)	Weight (gm)	Remarks
1	M1	Metal	4.8	76	Products higher than 3 cm, 35 gm and metal
2	M2	Metal	4.8	45	
3	M3	Metal	5.1	86	
4	M4	Metal	5	60	
5	NM1	Non-Metal	5	50	Products higher than 3 cm, 35 gm and non-metal
6	NM2	Non-Metal	4.7	63	
7	NM3	Non-Metal	4.5	83	
8	NM4	Non-Metal	3.5	99	
9	F1	Faulty	3	30	Products lower than 3 cm, 35 gm and metal/non-metal
10	F2	Faulty	2	51	
11	F3	Faulty	4.7	6	

CHAPTER 5

RESULT ANALYSIS & DISCUSSION

5.1 Steps to Find Results

Step 1: First we have to turn on the power switch after ensuring 220v AC power supplied to the project. The whole system will take about 5 seconds to be operational. All the servo motors will take their programmed position and all sensors will be activated automatically.

Step 2: After starting the system successfully, sample product has to be placed in the designated area for sorting. Then the sensors will sense its height, weight and detect if it's metal or non-metal. Sensor's signal will be processed by the Arduino chip and activate the servo motor accordingly.

Step 3: Each sample product is rectangular in shape. The LCD display will show 9.71 (Height between ultrasonic sensor and the platform) cm without any product and weight will be shown 00 gm. After placing any sample product in the designated area, the weight will be calculated at actual. But the height needs to be determined with an equation as follows:

h = (H-T)

Here,
H= Height displayed in the LCD
h = actual height of the sample
T= Tolerance of ultrasonic sensor is 1.2 cm

Step 4: After sorting any metal, non-metal or faulty product the delay for next product sorting is 3 seconds.

- Total time required to sort metal product is approximately – 18 seconds
- Total time required to sort non-metal product is approximately – 12 seconds
- Total time required to sort faulty product is approximately – 15 seconds

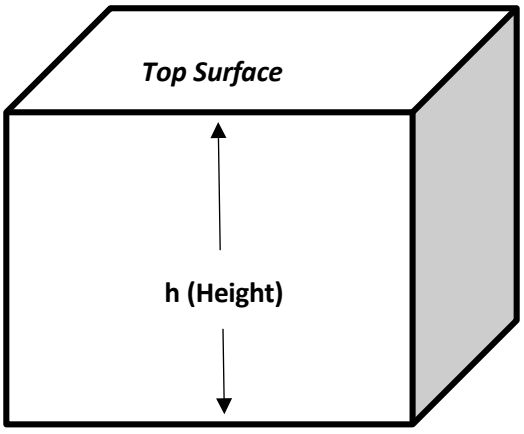


Figure: 5.1: Sample Product (Isometric View)

5.2 Data Tables

Table 5: Data Table for Metal Product Sorting

Sample Symbol	Weight (g)	Height (cm)	Time (Second)
M1	45	4.8	18.10
M4	60	5.00	18.18
M2	76	4.8	18.20
M3	86	5.1	18.29

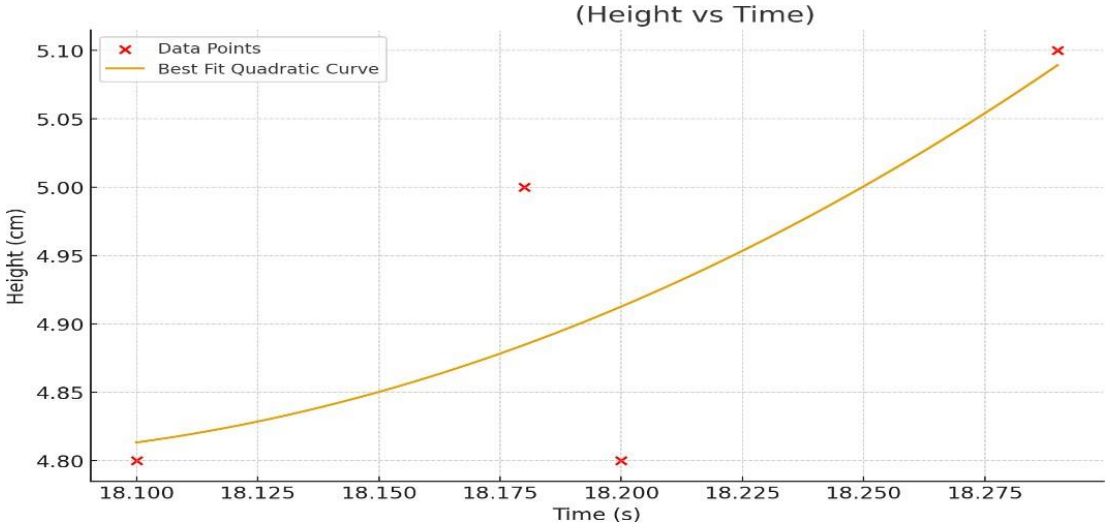


Figure: 5.2: Height vs Detection Time Graph (Metal)

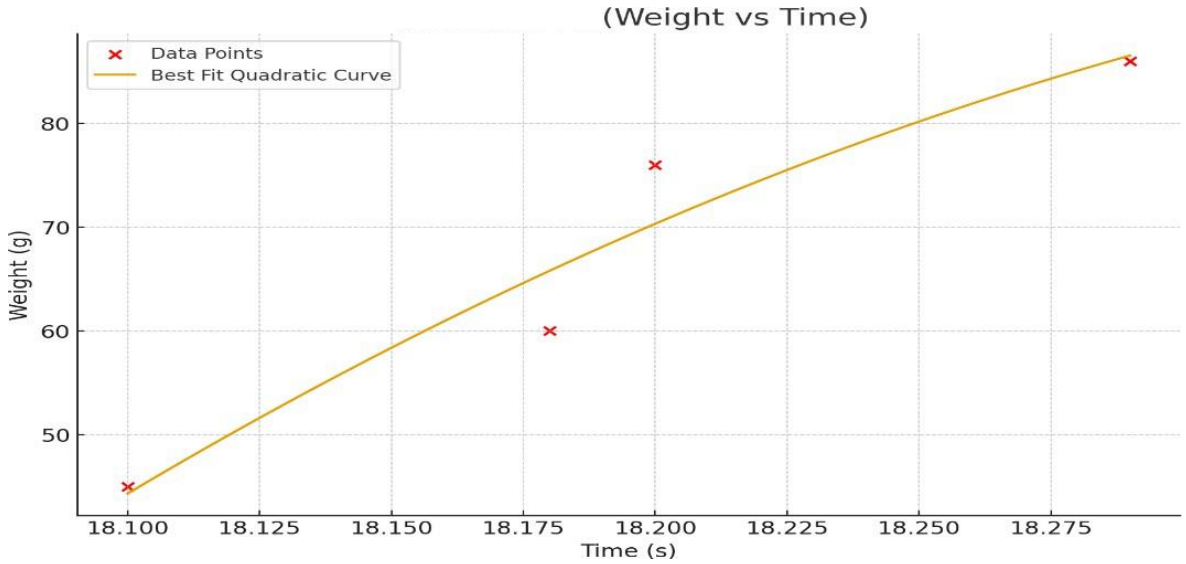


Figure: 5.3: Weight vs Detection Time Graph (Metal)

Table 6: Data Table for Non - Metal Product Sorting

Sample name	Weight (g)	Height (cm)	Time (Second)
NM1	50	5.00	12.95
NM2	63	4.7	13
NM3	83	4.5	13
NM4	99	3.5	13.12

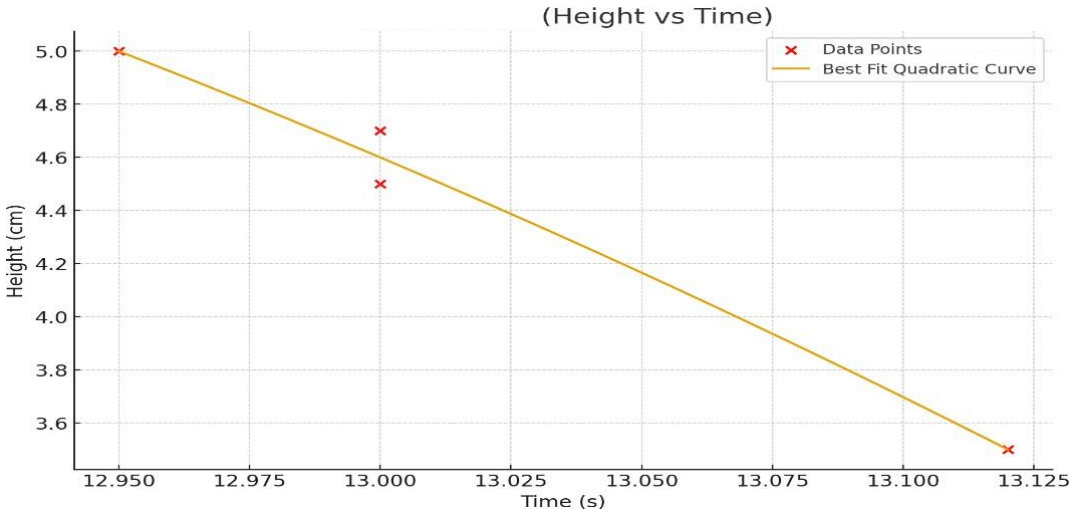


Figure: 5.4: Height vs Detection Time Graph (Non-metal)

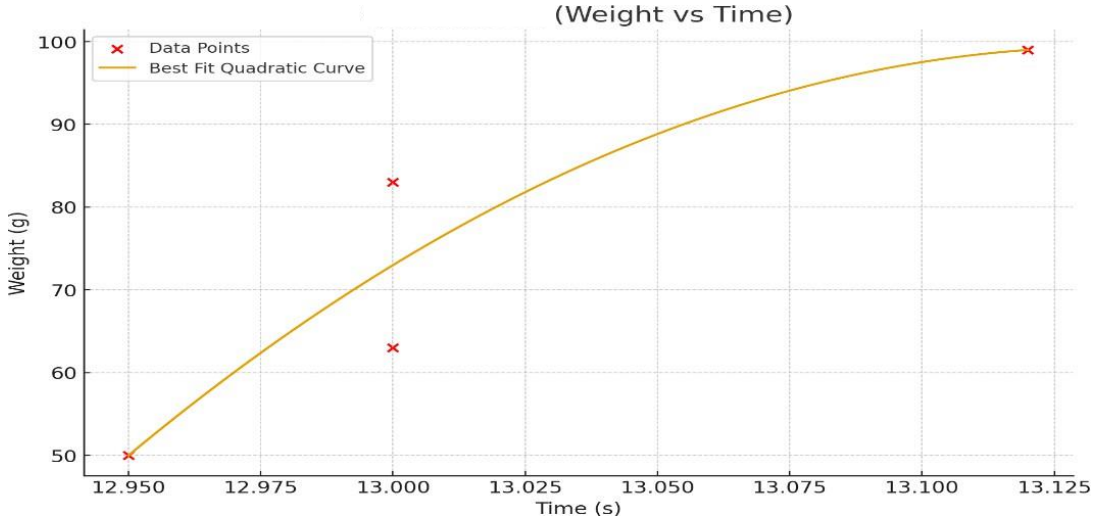


Figure: 5.5: Weight vs Detection Time Graph (Non-Metal)

5.3 Short Descriptions of the Quadratic Graphs

A. Height vs Time (Metal) – The quadratic fit shows how the height of the metal sample changes with time. The curve indicates a slight upward trend, suggesting an increase in height over time. Mentioned in Fig 5.2.

B. Height vs Time (Non-Metal) – The quadratic fit shows a downward trend in height with time, meaning the non-metal sample's height decreases as time progresses. Mentioned in Fig 5.4

C. Weight vs Time (Metal) – The quadratic fit shows a steady increase in weight with time, indicating the metal sample gains weight as time increases. Mentioned in Fig 5.3

D. Weight vs Time (Non-Metal) – The quadratic fit also shows an increase in weight with time for the non-metal sample, though the rise appears sharper compared to the metal. Mentioned in Fig 5.5

Table 7: Data Table for Faulty Product Sorting

Sample name	Weight (g)	Height (cm)	Time (Second)
F1	30	3	15.01
F2	51	2	14.98
F3	6	4.7	15.00

Note: Faulty metal or non-metal product sorting time is almost same and barely countable.

So, we did not use any graphical representation for this category.

Table 8: Combination Table of sample Products Sorting (Real Time Data)

Sl no	Sample Name	Type of Product	Height (cm)	Weight (gm)	Time (second)
1	M1	Metal	4.8	76	18.19
2	NM3	Non-Metal	4.5	83	12.88
3	F1	Faulty	3	30	15.19
4	F2	Faulty	2	51	15.20
5	M2	Metal	4.8	45	18.12
6	NM2	Non-Metal	4.7	63	13.02
7	NM4	Non-Metal	3.5	99	13.14
8	F3	Faulty	4.7	6	15.20
9	M3	Metal	5.1	86	18.10
10	NM1	Non-Metal	5	50	13.12
11	M4	Metal	5	60	18.11

5.4 Discussions

The automated product sorting system successfully demonstrated how weight, size, and material can be detected using a load cell, ultrasonic sensor, and induction proximity sensor integrated with a conveyor belt. The load cell provided accurate weight measurements, though minor fluctuations occurred due to belt vibrations, suggesting the need for better damping or filtering. The ultrasonic sensor measured object size reliably for regular shapes, but performance reduced with irregular or sound-absorbing materials. The induction proximity sensor effectively identified metallic objects but could not detect non-metallic ones, limiting its classification scope.

By combining these sensors, the system achieved multi-parameter sorting with good overall accuracy. However, challenges such as conveyor speed variation, sensor calibration, and mechanical misalignment occasionally affected performance. Despite these limitations, the prototype proved the feasibility of a low-cost, flexible, and scalable solution for industrial sorting applications.

5.5 Advantages

There are certainly many advantages of the project and some of the major ones have been given below -

- Automated Sorting Process.
- Saves time in product sorting.
- Reduces manual labor effort.
- Sorts by size, weight, and material.
- Increases sorting accuracy.
- Fast real-time sensor response.
- Low chances of human error.
- Suitable for industrial use.
- Easy to operate and monitor.
- Low power and energy efficient.
- Low Maintenance Requirements

5.6 Applications

Some of the application areas of the project has been pointed out below-

- Industrial packaging and product sorting automation.
- Pharmaceutical packaging and sorting operations.
- Logistics and warehouse inventory management systems.
- Mining industry for mineral sorting processes.

5.7 Limitations

- As this is a demo project, some limitations were identified. In the future, efforts will be made to reduce these types of limitations. The identified limitations are as follows:
- This project can now be only used for small scale purposes
- No energy backup.
- Cannot detect color or shape-based sorting.
- Not suitable for extremely fragile products.

5.8 Performance Analysis

After the project was completed, it was run and the output was observed. It was seen that the project was working well, as expected. Careful observations were made after the project was assembled, confirming that it functioned as desired. The output was delivered perfectly, and all equipment operated correctly. The performance was tested thoroughly, resulting in the expected and successful output.

- Finally, the project has been successfully completed, and it was checked to ensure it runs accurately according to the defined objectives.
- Initially, the system was started.
- Following are the outputs of the project.
- Successfully sorted products based on size, weight, and material type.
- Ultrasonic sensor measured object height with reliable precision.
- Load cell module provided correct weight data for classification.
- Proximity sensor efficiently identified metal and non-metal objects.
- Servo motors diverted products into correct bins based on combined parameters.
- LCD display showed live status of sorting results (weight, size, material).
- The system reduced manual effort and improved sorting speed and accuracy.

CHAPTER 6
CONCLUSION

6.1 Conclusion

The Size, Weight, and Material-Based Automatic Sorting System offers a smart and efficient approach to industrial sorting needs. By integrating ultrasonic, load cell, and proximity sensors with an Arduino Nano, the system accurately identifies and classifies products based on physical properties such as size, weight, and material type. The use of a conveyor belt and servo motors ensures smooth product movement and precise placement into the correct bins. Stable power delivery through SMPS and buck converters enhances system reliability.

This automation significantly reduces manual labor, improves sorting speed, and minimizes human error. It is particularly useful in industries like packaging, recycling, and manufacturing where precise product handling is critical. While the system has limitations with high-speed or shape-based sorting, its simplicity, cost-effectiveness, and adaptability make it a practical solution for small to mid-scale industrial applications. Overall, the project proves the effectiveness of sensor-based automation in enhancing operational efficiency and product classification accuracy.

6.2 Future Scopes

We are thinking about adding many features to our project in the future to get more desirable out comes. Some of the steps that we are thinking about taking are given below:

- In future development this project can be develop by more sensor & alarm system.
- In future, we will try it to apply in real field.
- In Future we will add extra power system.
- In future develop in IoT system Controlled our project
- We will add Counting Sensor.

REFENCES

[1] Lim Jie Shen & Irda Hassan, Design and Development of Colour Sorting Robot, Journal of Engineering Science and Technology EURECA 2014, (2015), 71-81. http://jestec.taylors.edu.my/eureca2014(1)_1_2015/eureca_14(1)_71_81.pdf

[2] Riky Tri Yunardi, Winarno and Pujiyanto, Contour-based object detection in Automatic Sorting System for a parcel boxes, International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA), (2015) 38-41, DOI: https://doi.org/10.1109/ICAMIMIA.2015.7507998

[3] Babita Nanda, Automatic Sorting Machine using Delta PLC, International Journal of Innovative Research in Advanced Engineering, 1(7) (2014) 235-242.

[4] Shreeya Vasant Kulkarni, Swati Rajendra Bhosale and Priyanka Pramod Bandewar, Automatic Box Sorting Machine, International Journal for Scientific Research & Development, 4(4) (2016), 57-58.

[5] S. Prajapati and S. Sutar, "Sorting of Object Based on Height, Weight and Colour on A Conveyor Line," *International Journal of Creative Research Thoughts (IJCRT)*, vol. 6, no. 2, pp. 176–178, 2018.

[6] S. V Rautu, A. P. Shinde, N. R. Darda, A. V Vaghule, C. B. Meshram, and S. S. Sarawade, "Sorting of Objects Based on Colour , Weight and Type on A Conveyor Line Using PLC," *IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)*, March, pp. 4–7, 2017, doi: 10.9790/168417010060407.

[7] A. A. Algitta, "Automated Packaging Machine Using PLC," *International Journal of Innovative Science Engineering and Technology (IJISET)*, vol. 2, no. 5, pp. 282–288, 2015.

[8] S. N. Kulkarni and S. Kumar Singh, "Object Sorting Automated System using Raspberry Pi," *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 2018, pp. 217-220, doi: 10.1109/CESYS.2018.8724056.

[9] A. Nasiri, M. Omid, and A. Taheri-garavand, "An automatic sorting system for unwashed eggs using deep learning," *Journal of Food Engineering*, vol. 283, February 2020, doi: 10.1016/j.jfoodeng.2020.110036.

[10] P. Patunkar, P. Amilkanthwar, S. Yadav, and T. Rahman, "Design and Development of Product Sorting Conveyor Belt Using PLC," *International Research Journal of Engineering and Technology (IRJET)*, vol. 8, no. 6, pp. 4337–4340, 2021.

[11] M. L. Dezaki, S. Hatami, A. Zolfagharian, and M. Bodaghi, “A pneumatic conveyor robot for color detection and sorting,” *Cognitive Robotics*, vol. 2, pp. 60–72, 2022, doi: 10.1016/j.cogr.2022.03.001

[12] R. Natarajan, B. Baskaran, S. Sivakumar, S. Mallikarjunan. R. Somasundaram, P. Raju, “Design and Development of Weight Based Material Sorting,” *International Journal of Pure and Applied Mathematics*, vol. 119, no. 12, pp. 2101–2108, 2018.

[13] H. Hassan, F. Saad and M. S. Mohd Raklan, "A Low-Cost Automated Sorting Recycle Bin powered by Arduino Microcontroller," *2018 IEEE Conference on Systems, Process and Control (ICSPC)*, Melaka, Malaysia, 2018, pp. 182-186, doi: 10.1109/SPC.2018.8704146.

[14] P. Choudesh Varma and G. Venkateswarlu, “Automatic Pneumatic Power Based Object Sorting System by Using MITSUBISHI PLC,” *CVR Journal of Science and Technology*, vol. 9, no. 1, pp. 91–97, 2015, doi: 10.32377/cvrjst0918.

[15] K. M. C. Babu and P. A. Harsha Vardhini, "Design and Development of Cost Effective Arduino based Object Sorting System," *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, pp. 913-918, 2020, doi: 10.1109/ICOSEC49089.2020.9215269.

[16] H. Borkar, “A Review on Arduino Based Color Sorting Machine,” *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 2, pp. 43–46, 2022, doi:10.22214/ijraset.2022.

[17] P. N. Bargal, A. Deshpande, R. Kulkarni, and R. Moghe, “PLC Based Object Sorting Automation,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 3, no. 7, pp. 103108, 2016.

[18] M. M. Sofu, O. Er, M. C. Kayacan, and B. Cetis, “Design of an automatic apple sorting system using machine vision,” *Computers and Electronics in Agriculture*, vol. 127, pp. 395–405, 2016, doi: 10.1016/j.compag.2016.06.030.

APPENDIX

Micro-Controller Programming Code :

```
#include <LiquidCrystal_I2C.h>
#include <HX711_ADC.h>
#include <Wire.h>
#include <Servo.h>

// --- LCD & LoadCell ---
LiquidCrystal_I2C lcd(0x27,16,2);
HX711_ADC LoadCell(4,5);

// --- Servos ---
Servo loadServo;       // Push to belt
Servo metalServo;      // Metal sorting
Servo nonMetalServo;   // Non-metal sorting

// --- Pins ---
const int trigPin = 6;       // Ultrasonic trigger
const int echoPin = 7;       // Ultrasonic echo
const int metalSensorPin = 3; // Metal detector

// --- Settings ---
float minWeight = 10.0;
float stableRange = 2.0;
unsigned long stableTime = 3000;
unsigned long receiptTime = 5000;
float triggerHeight = 7.0;
float sortHeightThreshold = 6.40;
float sortWeightThreshold = 35.0;

// --- Variables ---
float finalHeight = 0;
float finalWeight = 0;
bool isMetal = false;

enum MeasurementStage { MEASURE_HEIGHT, MEASURE_WEIGHT,
MEASURE_METAL, SHOW_RECEIPT, PUSH_PRODUCT };
MeasurementStage stage = MEASURE_HEIGHT;

unsigned long stableStartTime = 0;
float lastWeight = 0;

void setup() {
  Serial.begin(115200);

  // LoadCell setup
  LoadCell.begin();
  LoadCell.start(2000);
  LoadCell.setCalFactor(-769.27);
```

```
// LCD setup
lcd.init();
lcd.backlight();
```

```
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Sorting System");
  lcd.setCursor(0,1);
  lcd.print("Monitoring...");
  delay(2000);
  lcd.clear();

  // Pins
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(metalSensorPin, INPUT);

  // Servos
  loadServo.attach(9);
  metalServo.attach(10);
  nonMetalServo.attach(11);

  loadServo.write(0);
  metalServo.write(90);
  nonMetalServo.write(0);
}

void loop() {
  LoadCell.update();

  switch(stage) {
    case MEASURE_HEIGHT: {
      float height = getDistance();
      lcd.setCursor(0,0);
      if(height < 0) lcd.print("H: No Echo   ");
      else {
        lcd.print("H:"); lcd.print((float)height,2); lcd.print("cm   ");
        finalHeight = height;

        if(height < triggerHeight) {
          stage = MEASURE_WEIGHT;
          lastWeight = 0;
          stableStartTime = millis();
        }
      }
      break;
    }

    case MEASURE_WEIGHT: {
      float weight = LoadCell.getData();
      if(weight < 0) weight = 0;

      lcd.setCursor(0,1);
      lcd.print("W:"); lcd.print((int)weight); lcd.print("g   ");
```

```
      if(abs(weight - lastWeight) <= stableRange) {
        if(millis() - stableStartTime >= stableTime) {
        finalWeight = weight;
          stage = MEASURE_METAL;
        }
      } else {
        stableStartTime = millis();
        lastWeight = weight;
      }
      break;
    }

    case MEASURE_METAL: {
      isMetal = !digitalRead(metalSensorPin); // Inverted logic

      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("W:"); lcd.print((int)finalWeight); lcd.print("g");
      lcd.setCursor(0,1);
      lcd.print("H:"); lcd.print((float)finalHeight,2); lcd.print("cm ");
     delay(500);

// Only show receipt if metal AND meets height & weight conditions
if(isMetal && finalWeight > sortWeightThreshold && finalHeight <= sortHeightThreshold)
{
  stage = SHOW_RECEIPT;
} else {
  stage = PUSH_PRODUCT; // Skip receipt, go straight to pushing
}

      break;
    }

    case SHOW_RECEIPT: {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Receipt:");
      lcd.setCursor(0,1);
      if(isMetal) lcd.print("Metal Product");
      else lcd.print("Non-Metal");

      Serial.print("Height: "); Serial.print(finalHeight); Serial.print(" cm, ");
      Serial.print("Weight: "); Serial.print(finalWeight); Serial.print(" g, ");
      Serial.print("Type: "); Serial.println(isMetal ? "Metal" : "Non-Metal");

      delay(receiptTime);
      stage = PUSH_PRODUCT;
      break;
    }
```

```
    case PUSH_PRODUCT: {
      // Push product
      loadServo.write(150);
      delay(500);
      loadServo.write(0);

      // Sorting servos
      if(finalWeight > sortWeightThreshold && finalHeight <= sortHeightThreshold) {
        if(isMetal) {
          metalServo.write(0);
          delay(7000);
          metalServo.write(90);
        } else {
          nonMetalServo.write(90);
          delay(7000);
          nonMetalServo.write(0);
        }
      }

      Serial.println("Product pushed and sorted");

      // Reset for next product
      stage = MEASURE_HEIGHT;
      finalHeight = 0;
      finalWeight = 0;
      isMetal = false;
      lcd.clear();
      break;
    }
  }
}

// --- Ultrasonic distance ---
float getDistance() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin,HIGH);
delayMicroseconds(10);
digitalWrite(trigPin,LOW);

  long duration = pulseIn(echoPin,HIGH,30000); // 30ms timeout
  if(duration == 0) return -1; // No echo detected
  return duration * 0.034 / 2; // cm
}
```