DESIGN AND FABRICATION OF FAULTY PRODUCT DETECTION AND SEPARATION SYSTEM.

PROJECT REPORT

SUBMITTED BY

1. Wahid Dino Samo(18ME08)
2. Syed M. Hassan (18ME09)
3. Abdullah Memon(18ME107)
4. Shuja Ahmed(18ME155)
5. Rasheed(18ME24)
6. Zeeshan (18ME108)

Subject Teacher

Engr. Dur Muhammad Pathan

DEPARTMENT OF MECHANICAL ENGINEERING

Mehran University of Engineering & Technology, Jamshoro

Submitted for the Subject of Control Engineering as an Assignment for the Degree of Bachelor of Mechanical Engineering

21 February 2022

Learning Objectives

1. Selection of appropriate sensors.
2. Learn about the use of a microcontroller
3. Learn about the programming language.
4. Assemble the components to develop the project

TABLE OF CONTENTS

1. Introduction
1.1 Fundamental concept of the project.
1.2 Applications of project.
2. List of components
2.1 List of similar components available in the market (along with specification)
2.2 Selected components along with specification and justification
3. Circuit diagram of project.
4. Programming code of project
5. Working Mechanism of Project
6. Future extension of project
7. Output/Conclusion of project.

RUBRICS

Appearance	
Clear concept of selection of components	
Functionality of project	
Output result	
Clear concept of programming.	
Report writing	
Involvement of team members	

1. Introduction

Belt conveyor is widely used for material transportation over both short and long distances nowadays. All product manufacturing units need to have faulty product detection to maintain product quality and maintain a good reputation. So here we demonstrate such a system using a mini conveyer belt system. We propose to design and fabricate a faulty product detection system.

1.1 The fundamental concept of the project.

Each product is different and thus has different mechanisms to detect faulty products. Here faulty products are detected based on product size. A sensor is used to detect each product's height as products move over a conveyer belt. A defective product with a height higher than the maximum limit will be automatically detected as it moves on a conveyer belt and is separated by a servo arm. Here rollers and rubber belt is used to develop a mini conveyer belt mechanism. This mechanism is operated by a stepper motor. An ultrasonic sensor to detect product height and products with more than maximum height is detected as faulty products. The servo motor will remove the faulty product from the conveyor.

1.2 Applications of project.

It can be used in all those industries in which conveyor is used, and the product is judged by its height.

2. List of components

1. Arduino Nano
2. Breadboard
3. Jumper Wires
4. A4988 Stepper Motor Driver
5. ML17A3 NEMA 17 Bipolar Stepper Motor
6. Flexible Coupling
7. Rollers
8. Rexine Belt
9. Servo Motor
10. Ultrasonic Sensor HC-SR04
11. IR Sensor (MH Series)
12. Switch
13. Power Supply(12 Volt 2 Ampere)
14. LCD(16x2)

2.1 List of similar components available in market

- **LCD(16x2) with an I2C Connector**
- **DC Motor with relay**
- **Servo Motor(360)**
- **Rubber Belt**
- **Mini Breadboard**
- **Half Breadboard**
- **MG996R Metal Gear Servo Motor 360 Degree**
- **42BYGHW804 IM42HS60 NEMA17 Stepper Motor For 3D Printer And CNC**

  Specifications:

  Step Angle: 1.8
  Holding Torque: 4.8kg.cm
  Rated Voltage: 3.6Volt
  Rated Current: 1.2A/Phase
  Form Factor: NEMA 17
  Shaft Type D-Shaft
  Wire Connector JST PH male connector.
  2-Phase (4-Wires)
  Holding Torque: 48N.cm (4800g-cm)
  Resistance: 3.2Ω/phase
  Inductance: 5mH/phase
  Form factor: NEMA 17

- **28BYJ48 5V Stepper Motor**

  Details/ Specificatoins:

  Model : 28BYJ-48
  Number of Phase : 4
  Speed Variation Ratio : 1/64
  Stride Angle : 5.625° /64
  Rated voltage : DC [5v - 12v]
  DC resistance : 5v: 60Ω±7%(25℃); 12v: 130Ω±7%(25℃)

- **ULN2003 stepper motor driver**

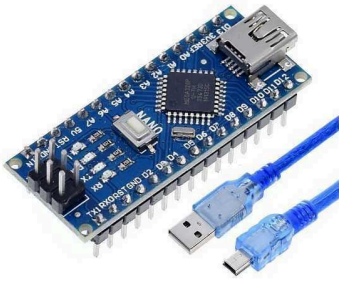
  Specifications
  On-board ULN2003A chipset
  5-12V power supply
  On-board 4-way indicator light
  On-board XH-5P socket
  Compatible with Arduino
  Stepper Motor Driver Test and DIY
  A, B, C, D LEDs indicate the shape of
  the four-phase stepper motor

- **Arduino UNO**

  Specifications
  Microcontroller ATmega328 DIP
  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 150 mA
  Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader
  SRAM
  2 KB (ATmega328) EEPROM
  1 KB (ATmega328)
  Clock Speed 16 MHz

2.2 Selected components along with specification and justification

1. Arduino Nano



The Arduino Nano is small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x) .It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one.

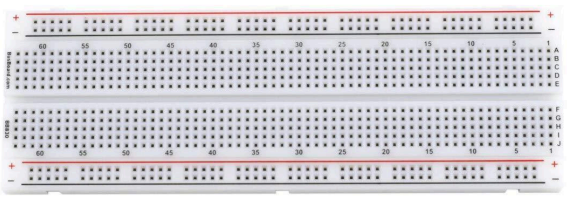
Specifications

- Microcontroller : ATmega328.
- Operating Voltage (logic level) : l5 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 : 16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by boot loader.
- SRAM : 1 KB (ATmega168) or 2 KB (ATmega328).
- EEPROM : 512 bytes (ATmega168) or 1 KB (ATmega328).
- Clock Speed : 16 MHz.
- Dimensions : 0.73" x 1.70".
- Length : 45 mm.

Justification For Selection of Arduino Nano

Arduino Nano is small in size as compared to Arduino UNO and is much cheaper. The Performance provided by both is the same. Also, Arduino Nano is easily fixable on Breadboard.

2. Breadboard



A breadboard, or protoboard, is a construction base for prototyping electronics.

Justification For Selection of a Breadboard

We could have used a Vero Board and Permanently fixed the Joints but instead, we use a Breadboard so that in case of any modification the Sensors are easy to remove.

3. Jumper Wires

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points without soldering. Jumper wires are typically used with breadboards and other prototyping tools to make it easy to change a circuit as needed. Fairly simple. It doesn't get much more basic than jumper wires.

4. A4988 Stepper Motor Driver

Specifications and Features :

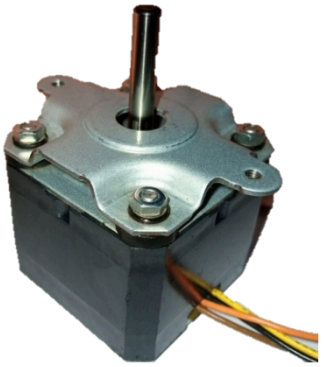
- Supply Voltage: 8-35 VDC.
- Current : 1.0A (no heatsink).
- Current : 2.0A (with heat-sinking).
- logic input : 3 – 5.5V.
- Automatic current decay mode detection/choice.
- Mixed with slow current decay mode.
- The low power dissipation of synchronous rectifier.
- Internal UVLO(ultra voltage lockout).
- Crossover current protection.
- Thermal shutdown circuit.
- Ground fault protection.
- Loading and short circuit protection.
- The optional five-step mode: full, 1/2, 1/4, 1/8 and 1/16.

Justification For the Selection of A4988 Stepper Motor Driver

A4988 is the best stepper motor available for controlling NEMA 17 Series Stepper Motors

5. ML17A3 NEMA 17 Bipolar Stepper Motor

Specifications and Features:



- Step Angle: 1.8°
- Phase Current: 1A
- Phase Resistance: 1.5Ohm
- Phase Inductance: 2.8mH
- Holding Torque: 10N.cm
- Braking Torque: 2.2N.cm
- Max. Temperature Rise: 80℃ (rated current, 2 phase on)
- Operating Temperature: -20℃~+70℃
- Insulation Resistance: 100 MΩ Min. , 500VDC

Justification For the Selection of ML17A3 Stepper Motor

Normal DC Motors of high torque can not be controlled much using Arduino but stepper motors are much more controllable than conventional DC Motors. ML17A3 is a Bipolar Stepper Motor that is fairly easier to use than a Unipolar Stepper Motor. ML17A3 belongs to NEMA 17 series which is a very common stepper motor whose tutorials are available on the internet and are fairly easier to set up. ML17A3 NEMA 17 is the cheapest motor available.

6. 5x6mm Flexible Coupling

5x6mm Flexible Coupling Shaft can connect driving shaft with the driven shaft. 5x6mm Flexible Coupling Shaft, it is very efficient in eliminating any misalignment to the possible extent. Minimum backlash is another great advantage of this Aluminium Flexible Coupling. This Flexible coupling with an outer diameter of 25mm and a length of 29mm is machined aluminum coupling. The first bore diameter is 5mm and the second bore diameter is 6mm.

Specifications and Features:

- Material: Aluminum Alloy
- Outer Diameter: 25mm
- Length: 29mm
- First bore diameter: 5mm
- Second bore diameter: 6mm
- Main Color: Silver Tone
- Torsional stiffness: Zero backlash
- Standard or Nonstandard: Standard
- Structure: Gear, Flexible or Rigid
- Absorption of parallel, angular misalignments and shaft end-play by spring action.
- Corrosion resistance and High Durability.
- For little torque, the shaft connects.
- One-piece construction with long-lasting performance.
- Low moment of inertia offering constant velocity.

7. TowerPro SG90 SG 90 180 Degree Servo Motor

Tiny and lightweight with high output power. The servo can rotate approximately 180 degrees (90 in each direction) and works just like the standard kinds but smaller.

Specification:

- Operating Voltage: 4.8~6.0V
- Operating Speed: 0.12sec/60 degree(4.8V)~0.1sec/60 degree(6.0V)
- Torque: 1.6kg/cm(4.8V)
- Dead Band Width: 5usec
- Temperature Range: -30~ 60?
- Cable Length: 25cm
- Servo Type: Analog Servo
- Brand Model: Tower Pro SG90

8. Ultrasonic Sensor HC-SR04

HC SR04 HC-SR04 Ultrasonic Sensor also called HC-SR04 Ultrasonic Range Finder. Ultrasonic Sensor module HC-SR04 provides 2cm – 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules include ultrasonic transmitters, receiver and control circuit.

Features :

- Power Supply: +5V DC
- Quiescent Current : <2mA
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance : 2cm – 400 cm/1″ – 13ft
- Resolution : 0.3 cm
- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS minimum
- Dimension: 45mm x 20mm x 15mm

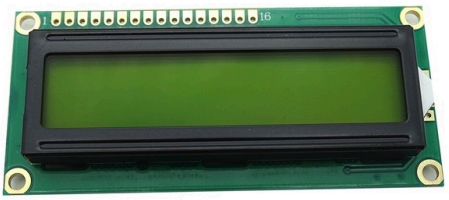
9. IR Sensor (MH Series)

Features and Specifications

- Main Chip: LM393
- Detection Angle: 35 °
- Board Size: 31 * 14mm / 1.22 * 0.55in
- Board Weight(1pc): 3g
- 5VDC Operating voltage
- I/O pins are 5V and 3.3V compliant
- Range: Up to 20cm
- Adjustable Sensing range
- Built-in Ambient Light Sensor
- 20mA supply current
- Mounting hole

10. LCD (16x2)

Features



- 16 Characters x 2 Lines
- Green Backlight
- 5x7 Dot Matrix Character + Cursor
- HD44780 Equivalent LCD Controller/driver Built-In
- 4-bit or 8-bit MPU Interface
- Standard Type
- Works with almost any Microcontroller

11. Rollers

Rollers are made with the help of PVC Pipe and Nut Bolts. To rotate the conveyer belt and complete the conveyer mechanism

12. Rexene Belt

For the Conveyer belt the material of Rexene is cheap and similar to that used in conveyers.

13. Switch

The Switch is used to turn on and off the power supply to the motor in order to avoid burning of Arduino nano and other sensors.

14. Power Supply

A power Supply is used to operate the Stepper Motor continuously.

Features

- Operating Voltage 220V AC
- Output Voltage: 12V DC
- Output Current: 2A

3. Circuit diagram of project.

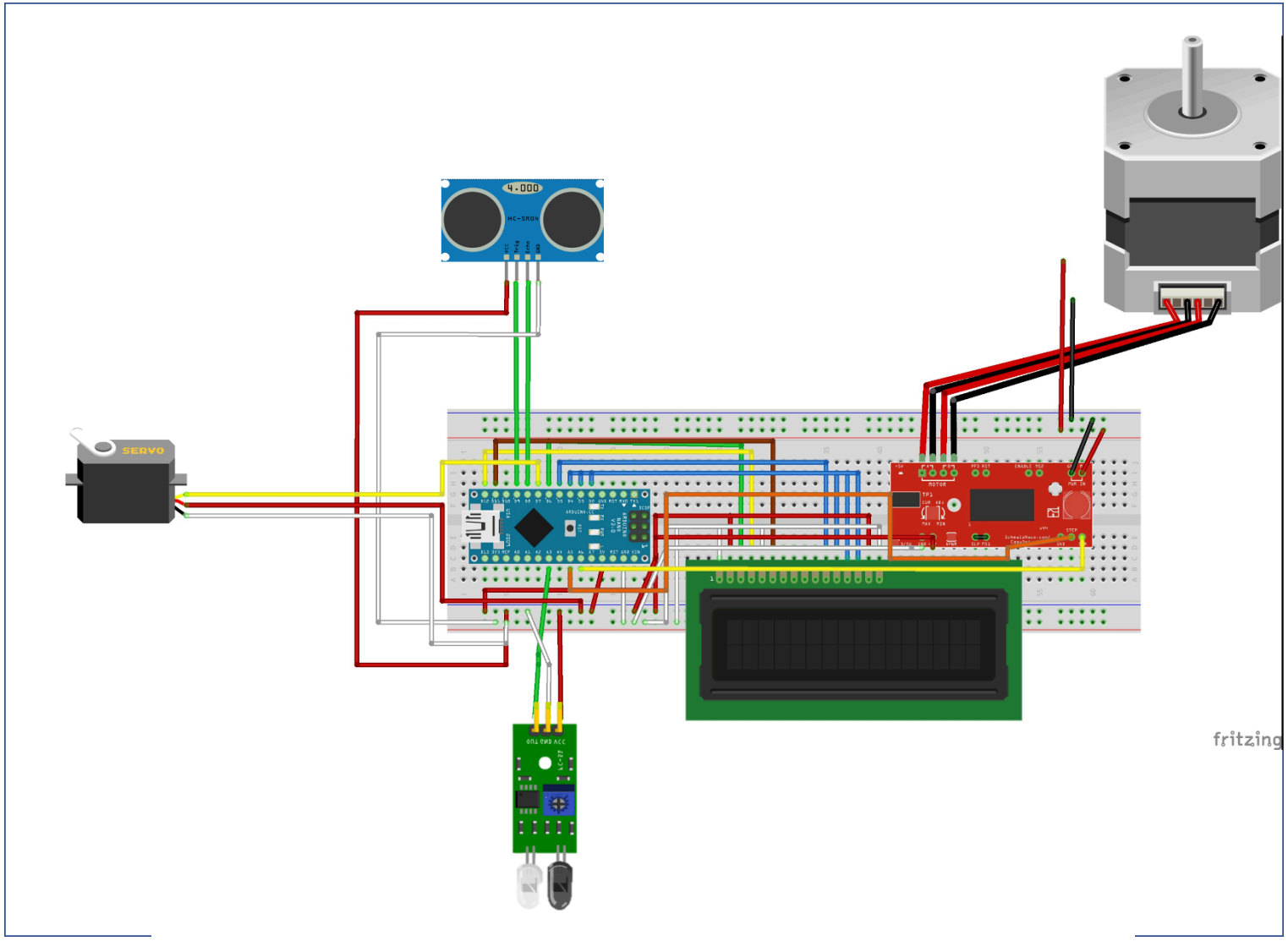


Fig: Circuit Diagram of The Project

LCD Connections
D4 ----5
D5 ----4
D6 ----3
D7 ----2
E ----11
RS ----12
A ----5V
VSS ----GND
VO ----6
RW ----GND

Stepper Motor Connections

A+ ---- Stepper Pin1
A- ---- Stepper Pin2
B+ ---- Stepper Pin 3
B- ---- Stepper Pin 4

A4988 wiring
VCC ----5V
GND ----GND
dirPin ---- A5
stepPin---- A4
PWR ---- +12V
GND ---- -12V
VCC ---- 5V
GND---- GND
SLP ---- RESET
RESET ---- SLP

IR Sensor Connections

IRSensor ---- A3

VCC ----5V

GND ----GND

Ultrasonic Sensor

trigPin ---- 9
echoPin ---- 8

4. Programming code of project

```
#include <LiquidCrystal.h>
#include <Servo.h>
#include <AccelStepper.h>
Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards


#define dirPin A5
#define stepPin A4
#define motorInterfaceType 1

AccelStepper stepper = AccelStepper(motorInterfaceType, stepPin, dirPin);


int pos = 0;    // variable to store the servo position

int IRSensor = A3;


int Contrast=0;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

const int trigPin = 9;
const int echoPin = 8;
long duration;
long distance;

int counter = 0;
int currentState = 0;
int previousState = 0;
int fault=0;
int f1 = 0;
int f2=0;


  void setup() {
      stepper.setMaxSpeed(1000);
      pinMode(stepPin, OUTPUT);
      pinMode(dirPin, OUTPUT);

        myservo.attach(7);   // attaches the servo on pin 7 to the servo
object

      analogWrite(6,Contrast);
      lcd.begin(16, 2);
```

```
      pinMode(trigPin, OUTPUT);
      pinMode (IRSensor, INPUT); // sensor pin INPUT
      pinMode(echoPin, INPUT);
      lcd.clear();

        }

    void loop()
       {
        // Set the speed in steps per second:
  stepper.setSpeed(200);
  // Step the motor with a constant speed as set by setSpeed():
  stepper.runSpeed();

        lcd.setCursor(0,0);


        int statusSensor = digitalRead (IRSensor);

        if (statusSensor == 1)      {
        currentState = 0;
        f1=0;
                                    }

        else
               {

          delay(1000);
          distance=readUltrasonicSensor();
          int d=18;
          int dist=d-distance;
          lcd.print("HEIGHT: ");
          lcd.print (dist);
          lcd.print("cm");
          currentState = 1;


            if(dist>5){

              delay(1000);
                  for (pos = 0; pos <= 180; pos += 180) { // goes from 0
degrees to 180 degrees
              // in steps of 1 degree
                  myservo.write(pos);               // tell servo to go to
position in variable 'pos'
                delay(1000);                          // waits 15ms for the
servo to reach the position
```

```
                                                    }
                for (pos = 180; pos >= 0; pos -= 180) { // goes from 180
degrees to 0 degrees
                 myservo.write(pos);               // tell servo to go to
position in variable 'pos'
               delay(1000);                          // waits 15ms for the
servo to reach the position
             f1=1;

                                                    }
                     }

           if(dist<5);{
            // Set the current position to 0:
            stepper.setCurrentPosition(0);
              // Run the motor forward at 200 steps/second until the motor
reaches 400 steps (2 revolutions):
             while(stepper.currentPosition() != 400)
                  {
                    stepper.setSpeed(200);
                    stepper.runSpeed();
                   }


                   }
             }


        if(currentState != previousState){
           if(currentState == 1){
              counter = counter + 1;


              lcd.setCursor(0,1);
              lcd.print("T:");
              lcd.print(counter);

                                 }

      }

      if(f1 != f2){
          if(f1 == 1){
             fault= fault + 1;
             lcd.print(" F:");
             lcd.print(fault);
```

```
                                }
        }


        }


long readUltrasonicSensor(){
  // Send 10µs pulse
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Read pulse duration
  duration = pulseIn(echoPin, HIGH);
  Serial.println(duration); //Convert and return value
  return duration/ 58;
}
```

5. Working Mechanism of the Project

The Conveyer Belt Mechanism Starts moving when we switch on the 12V Supply for the Stepper Motor. The Arduino Nano is connected to Laptop or any external supply to operate. As the object is placed on the conveyer the object moves forward. As the Object Approaches the IR Sensor the Conveyer belt stops for 1 second until that the ultrasonic sensor which is placed 18cm above the conveyer surface detects the height of the object. If the height of the object is greater than 5cm the Servo Motor with an arm connected rotates to 180 degrees striking off the object from the conveyer belt. The Stepper motor moves 400 steps so that the Object is moved away from the IR Sensor.

6. Future extension of project

The Servo Motor Actuator can be replaced with a hydraulic actuator. The Conveyer belt can be further modified to perfection. Sensors such as color sensor, pressure sensors to detect weight of the object can be used to further optimize the project. A Sensor can be attached so that it separates metallic and non-metallic objects. A claw can be placed at the end of the conveyer belt which places objects according to the color and object properties.

7. Conclusion

In this Project the usage of Arduino Nano, Stepper Motor, Stepper Driver, Servo Motor, Ultrasonic Sensor and IR Sensor was learned and how a microcontroller(Arduino Nano) is programmed.