International Journal of Innovative Research and Reviews, 5(1) 31-37

International Journal of Innovative Research and Reviews
ISSN: 2636-8919
Website: www.injirr.com
doi:
Research paper, Short communication, Review, Technical paper

RESEARCH ARTICLE

Design and Manufacturing of an Industrial Autonomous Controlled Separation System

Efe ÇEtin YILMAZ[1, *], Barış CANGÜLEÇ, Uğurcan ACAR[1]

[1] *Kilis 7 Aralık University Engineering Faculty Department of Control Systems Electrical and Electronic Engineering / Kilis / Turkey*

** Corresponding author E-mail: efecetinyilmaz@msn.com*

HIGHLIGHTS

- In this study design and manufacturing of an industrial autonomous controlled separation system, which consists of two parts, will load the products on the transport truck after separating them and transmit the products to their warehouses autonomously.
- Within the scope of this study, packaging and transportation services will be accelerated with the autonomously controlled robot sorter designed and produced, and thus production efficiency will be increased.
- This study scope separates the objects coming from the production line according to their color and transmits them to the transport car, and we transfer each product to its warehouse with this transport car, so we aim to accelerate production and respond to human needs more quickly and to minimize accidents due to heavy loads.

ARTICLE INFO

Received : 06.10.2021
Accepted : 07.05.2021
Published : 07.15.2021

Keywords:
Autonomous control,
System design and programming,
Microprocessor,
Color Sensor

ABSTRACT

Today, the human population is constantly increasing and production facilities and factories are constantly being built to meet the needs of people. Factories work 24 hours a day using manpower and autonomous systems to meet human needs. Although autonomous systems are used in factories, manpower is still dominant. Because manpower is at the forefront and the lack of autonomous systems during product packaging processes in the factories in the production area, production slows down and workers who have to carry heavy products can be overcome from time to time. In this study scope, we have designed our smart factory project, which will reduce manpower and bring technological power to the fore to increase production speed, prevent human errors and human diseases. This study scope separates the objects coming from the production line according to their color and transmits them to the transport car, and we transfer each product to its warehouse with this transport car, so we aim to accelerate production and respond to human needs more quickly and to minimize accidents due to heavy loads. In this study design and manufacturing of an industrial autonomous controlled separation system, which consists of two parts, will load the products on the transport truck after separating them and transmit the products to their warehouses autonomously. The number of products packed life will be transmitted to the engineer controlling the warehouse information via PC via C#, and it can be intervened in case of any develop a fault thanks to the impression of C# over the PC.

Contents

Cite this article Yilmaz EÇ, Cangüleç B, Acar U. Design and Manufacturing of an Industrial Autonomous Controlled Separation System. *International Journal of Innovative Research and Reviews (INJIRR)* (2021) 5(1) 31-37
Link to this article: http://www.injirr.com/article/view/68

1. Introduction

Autonomous control is an approach to dealing with increasing complexity in production and logistics. Autonomous control methods are characterized by decentralized coordination of logistic objects in a hierarchical organizational structure. Autonomous objects themselves are capable of processing information to make and execute decisions on their own. So far, research in this area has mainly focused on the effects of autonomous control methods that ignore their integration into existing planning systems. Therefore, it is currently not possible to give a verified recommendation that the combination of planning and autonomous control methods achieves sufficient logistics target success in production systems of different complexity [1]. Manufacturing today is facing the effects of several megatrends: globalization and growing market volatility are putting strong pressure on manufacturing companies. The world economy and, in particular, markets are characterized by growing dynamics and instability [1]. With a dynamic product lifecycle and a simultaneous increase in the number of products, deeper integrated technologies present complex problems for planning and control of production [2]. In such an environment, the concepts of collaborative production planning and control (PPC) are pushed to their limits . On the contrary, autonomous management methods are considered a promising approach to combat the growing dynamics and complexity of logistics processes [3]. The concept of autonomous control includes decentralized coordination of autonomous logistics objects within a hierarchical structure [1]. Predictive approaches create a single static program that tries to account for each deviation or deterioration through forecasting and forecasting, while fully reactive planning strategies select the next job for the near future from the current set of jobs for the current machine under consideration. The hybrid approach is called predictive-reactive planning and describes a two-step solution for dynamic programming. The first step creates a forward-looking, long-term and global program at the beginning of the period and allows a second step to adjust the program locally and reactively in case of unforeseen disruptions [4, 5]. Production scheduling is one of the most researched and complex problems in production research [6]. Many different solution approaches have been developed over time since the first trial in [6]. because of its importance in terms of production performance. The agent-based approach naturally solves these problems of flexibility, dynamics, and autonomy, providing an efficient decentralized planning system [7]. In addition, agent-based scheduling systems offer additional features such as modularity, configurability, scalability, or reliability [8]. In particular, the autonomous and self-controlled nature of the described scheduling system can be achieved using an agent-based approach, due to the suitability of multi-agent systems for purely reactive use approaches to planning [4].

The packaging part comes after the production stage of the factories. Here, people usually do the manual packaging and sorting process. Afterward, the products are moved to the warehouse to wait for the delivery time, and all of them are human power. Due to this situation, people may perform the wrong sorting or storage process while performing the sorting and storage process. In some cases, they can even crash and hurt themselves. This slows down the production of the factory and can make it unable to meet people's needs. Thanks to the in this study we designed, the sensors on the conveyor belt will make a quick decision, separate the products and load them into the vehicle at the end of the conveyor. Our vehicle will take its energy from the electrical wires that pass over it completely on the rail system and after carrying the products to the places where it needs to go in the warehouse, it will come back to the conveyor belt. Our aim here is to reduce human errors and accidents to zero, to avoid using batteries and diesel in the vehicle, and to prevent costs in terms of high battery and diesel engine. The process will be completed by speeding up production and responding to human needs immediately.

2. Materials and Methods

2.1. Electronic and Mechanic Design

In the industrial autonomous controlled separator system designed and manufactured within the scope of this study; Seven Segment Adjustable 3 A Voltage Regulator Board LM2596-ADJ (1) Arduino UNO (1) Arduino Mega (1) 10 mm LDR (3) Laser Sensor - 650 nm, 5 V, 5 mW (3) TCS3200 Color Sensor Card - Sensor Slot (1) Tower Pro SG90 RC Mini (9gr) Servo Motor (1) 12V 25mm 60 RPM Motor (1) 6 V 500 RPM Micro DC Geared Motor (2) L298N Motor Driver (1) Coupling & Bearing (1 & 1) parts were used. In this study, we fixed laser sensors and LDRs on the side base of the boxes that we describe as blue, red, and green. When the light from the laser falls on the LDR, the internal resistance of our LDR changes. We take this change digitally and process it. The system detects the arrival of the product when it exceeds the parameters we have determined. In this way, the incoming product is counted and we can instantly see the fullness of the boxes. Figure 1 shows, fixed laser sensors, and LDRs on the side base of the boxes that we describe as blue, red, and green.

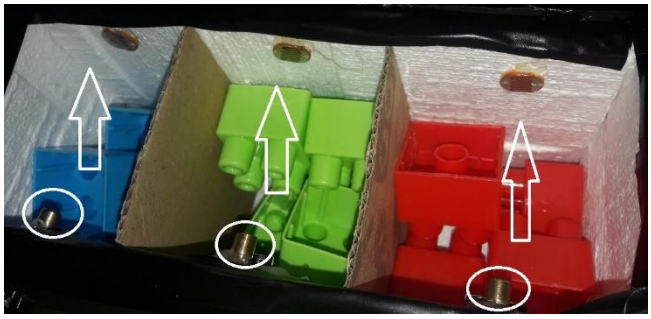

Figure 1 Fixed laser sensors and LDRs on the side base of the boxes that we describe as blue, green, and red (respectively from left to right)

We can say that the TCS3200 provides data flow to us with the help of photodiodes and a current-frequency converter. The sensor produces a square wave at a frequency that is directly proportional to the intensity of the light coming to them, and we process the square wave signals coming to us to determine the color of the object. In addition, in this study, after the products coming on the conveyor belt are recognized by our TCS3200 sensor, the angle signal is sent to our servo motor through the card. This signal changes with the color of the product that comes instantly. Different signals go for Green, Red, and Blue. In this way, we transmit our products to different areas with a single servo motor. After sorting the products, we used a 6V 500 rpm motor for the movement of our transport trolley, which we will use to transport them to their warehouses. When choosing our engine, a motor that can meet these needs has been selected by considering the product load it will carry, the speed values it should be and the energy it uses. In this study, we use the L298N driver card for the control of the motors in the conveyor belt and transport trolley. This is because our Arduino supports up to 100 mA. This means that it cannot give the power to operate our motors, and if we connect our motors directly to our Arduino board, our motors will want to draw the current demand from our board and our board may be damaged as a result. Thanks to the motor driver, we can control our motors by adjusting how the motor driver will power which motor with an external power supply and signals from Arduino. The connection of our drive roller with the motor is provided with a 6mm - 6mm coupling. Coupling is a mechanical apparatus used to transmit a motion to another piece of equipment. There is a roller bearing at the connection points of our drive and tail rolls. The reason we use it is to ensure that our rolls rotate in a medium with low friction. In this way, it performs the rotation process more comfortable and healthy. Figure 2 shows a) Positions of Motor and Driver in the process b) Coupling Connection to Motor Shaft c) Conveyor Belt Bearing.

Figure 2 a) Positions of Motor and Driver in the process b) Coupling Connection to Motor Shaft c) Conveyor Belt Bearing.

It is a known fact for mechanic process drive and tail drums are the rolls that help our belt to rotate. It is our drive roller that provides the first movement to our belt and keeps the movement. We used PPRC as the drive roll and rubber-coated on the outside so that the tape holds well. There is a roller bearing at the connection points of our drive and tail rolls. Thanks to the bearing, it can rotate more easily through the process. We used a 12 V 1.5 A Adapter (2.5 / 2.1 mm Jack) to energize our band and we used a 7 Segment Adjustable 3 A Voltage Regulator Card to adjust the speed of our band, so we can adjust the speed of our band in analogy. Figure 3 shows the conveyor belt used for the separation process in this study-designed system.

Figure 3 Conveyor belt used for the separation process

2.2. Separation and Transportation System Design

In the system designed in this study scope, a moving slide is mounted at the end of the belt to separate the products moving from the conveyor belt. This slide system is controlled by the Servo motor and changes the position of the slide using real-time information with the information produced by the sensor and the decision made by Arduino. Figure 4 shows the systematic slide system used separation process.

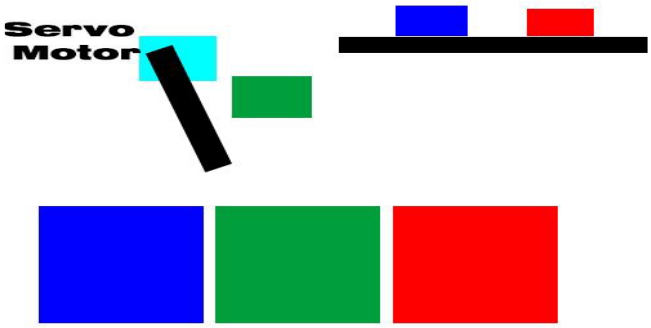

Figure 4 Systematic of slide system used separation process

After the separation process is completed, the transport car is designed as shown in Figure 5. 6 V 500 Rpm Geared Dc Motor is used for its movement. Our vehicle will go to the warehouses on the road where its wheels are buried. Afterward, it will come next to our conveyor belt thanks to the round rail track.

Figure 5 Transport Car

2.3. Product Recognition and Separation System Design

The products flowing on the conveyor belt will be separated by color sensors and servo motors. Numerical information will be produced by the reflection of the light sent to the object by the color sensor used. Product recognition will take place after the generated numerical information is processed on the Arduino card base. After the product recognition process, the servo motor on the belt will give way to the products and load them into the vehicle. As a result, a period

of product flowing over the belt will be separated and transported. The products flowing from the conveyor belt will be filled in 3 different categories in the baskets placed on the trolley. Baskets will be used to separate Blue, Red, and Green products. The control system of the transport cart will be provided with the Arduino UNO card. When one of the baskets is full, our vehicle will go to the warehouse areas on the road with the rail system we specially designed for it. The occupancy and space conditions in the boxes inside the transport vehicle will be provided by the laser placed on the LDR. In this way, the engine control of the vehicles will be provided. It will be ensured that the empty boxes will not stop in their storages and continue on their way. Figure 6 shows the connection circuit of the product recognition and separation system and connection circuit of the transport vehicle respectively.

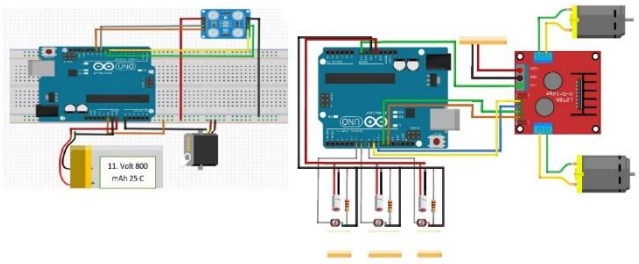

Figure 6 Connection circuit of the product recognition and separation system and connection circuit of the transport vehicle

2.4. Design of System Software

The color detection and separation software of the industrial autonomous controlled separation system designed and produced within the scope of this study is shown in Appendix 1. We did the remote monitoring part with C #, which is from the same language family as Arduino because C # and Arduino use C / C ++ languages, so we did not have any compatibility problems. Pity our code on the Visual Basic and C # Windows Forms Application part. Appendix 2 shows the software of the control system of the designed the industrial autonomous controlled separation system based on remote instant real-time data.

3. Conclusions

This study scope separates the objects coming from the production line according to their color and transmits them to the transport car, and we transfer each product to its warehouse with this transport car, so we aim to accelerate production and respond to human needs more quickly and to minimize accidents due to heavy loads. In this study design and manufacturing of an industrial autonomous controlled separation system, which consists of two parts, will load the products on the transport truck after separating them and transmit the products to their warehouses autonomously. The number of products packed life will be transmitted to the engineer controlling the warehouse information via PC via C#, and it can be intervened in case of any develop a fault thanks to the impression of C# over the PC.

Funding

This study was funded by The Scientific and Technological Research Council of Turkey Project Number: TUBITAK 2209-A 1919B011904279)

Conflict of Interest

The authors declare that they have no competing interests.

Appendices

Appendix 1: The color detection and separation software of the industrial autonomous controlled separation system

Appendix 2: The Software of the control system designed the industrial autonomous controlled separation system based on remote instant real-time data.

References

[1] Grundstein S, Schukraft S, Scholz-Reiter B, Freitag M. Evaluation system for autonomous control methods in coupled planning and control systems. *9th Cirp Conference on Intelligent Computation in Manufacturing Engineering - Cirp Icme '14* (2015) **33**:121–126.

[2] Windt K, Philipp T, Bose F. Complexity cube for the characterization of complex production systems. *International Journal of Computer Integrated Manufacturing* (2008) **21**(2):195–200.

[3] Scholz-Reiter B, Kolditz J, Hildebrandt T. Engineering autonomously controlled logistic systems. *International Journal of Production Research* (2009) **47**(6):1449–1468.

[4] Aytug H, Lawley MA, McKay K, Mohan S, Uzsoy R. Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research* (2005) **161**(1):86–110.

[5] Vieira GE, Herrmann JW, Lin E. Rescheduling manufacturing systems: A framework of strategies, policies, and methods. *Journal of Scheduling* (2003) **6**(1):39–62.

[6] Foerster F, Mueller D, Scholz D, Michalik A, Kiebler L. Integration of condition based maintenance orders into the decision-making of autonomous control methods. *52nd Cirp Conference on Manufacturing Systems (Cms)* (2019) **81**:216–221.

[7] Shen WM, Hao Q, Yoon HJ, Norrie DH. Applications of agent-based systems in intelligent manufacturing: An updated review. *Advanced Engineering Informatics* (2006) **20**(4):415–431.

[8] Shen WM, Wang LH, Hao Q. Agent-based distributed manufacturing process planning and scheduling: A state-of-the-art survey. *Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews* (2006) **36**(4):563–577.

Appendix 1

```
#define s0 8
#define s1 9
#define s2 10
#define s3 11
#define sensorOut 12
#include <Servo.h> //
Servo motor; //
int K,Y,M; //
void setup() {
```

```
pinMode(s0, OUTPUT); //.
pinMode(s1, OUTPUT);
pinMode(s2, OUTPUT);
pinMode(s3, OUTPUT);
pinMode(sensorOut, INPUT);
digitalWrite(s1,LOW);
digitalWrite(s0,HIGH);
motor.attach(13);
Serial.begin(9600);
}
void loop() {
digitalWrite(s2, LOW); //
digitalWrite(s3, LOW);
K = pulseIn(sensorOut, LOW);
Serial.print("Kırmızı= ");
Serial.print(K); //
Serial.print(" ");
delay(50); //
digitalWrite(s2, HIGH); //
digitalWrite(s3, HIGH);
Y = pulseIn(sensorOut, LOW); //
Serial.print("Yeşil= ");
Serial.print(Y); //
Serial.print(" ");
delay(50); //
digitalWrite(s2, LOW); //
digitalWrite(s3, HIGH);
M = pulseIn(sensorOut, LOW); //
Serial.print("Mavi= ");
Serial.print(M); //
Serial.println();
delay(50); //
if(K<26 && Y<32 && M<=51)
{ Serial.println("3"); /
motor.write(20); // }
else if(K<=50 && Y>60 && M>=50)
{ Serial.println("2");
motor.write(60); // motor 60 dereceye git.}
else if(K>=88 && Y>78 && M>=45)
{ Serial.println("0");
motor.write(80); //
else
Serial.println("1 ");
```

Appendix 2

```
using System; //
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;/*
namespace konveyor_ile_arduino
{public partial class Form1 : Form
{private string data;
public Form1()
{InitializeComponent();
button1.Location = new Point(100, 101);
}
int i = 0; //
int c = 0;
int b = 0;
private void button1_Click(object sender, EventArgs e)
{ try
{ 20

serialPort1.PortName = comboBox1.Text;
serialPort1.BaudRate = 9600;/*
serialPort1.Open();//Seri portu aç
baslat.Enabled = true; //
baslat.Enabled = false;//
} catch (Exception ex)
{MessageBox.Show(ex.Message, "Hata"); //
}}
private void durdur_Click(object sender, EventArgs e)
{ serialPort1.Close(); //Seri Portu kapa
baslat.Enabled = false; //
baslat.Enabled = true; //
}
private void button3_Click(object sender, EventArgs e)
{ //
```

```
textBox1.ResetText(); //textBox1'i sıfırla
listBox1.Items.Clear(); // depo listelerini temizle
listBox2.Items.Clear();
listBox3.Items.Clear();}
private void Form1_Load(object sender, EventArgs e)
{
textBox1.ReadOnly = true;//textBox1'i yalnızca okunabilir şekilde ayarla
string[] ports = SerialPort.GetPortNames();Seri portları diziye ekleme
foreach (string port in ports)
comboBox1.Items.Add(port);//Seri portları comboBox1'e ekleme
serialPort1.DataReceived+=newSerialDataReceivedEventHandler(SerialPort1_DataReceived);
//DataReceived eventini oluşturma }
private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
if (serialPort1.IsOpen) serialPort1.Close();//Seri port açıksa kapat
}
private void SerialPort1_DataReceived(object sender,SerialDataReceivedEventArgs e)
{
data = serialPort1.ReadLine();//Veriyi al
this.Invoke(new EventHandler(displayData_event));
}
private void displayData_event(object sender, EventArgs e)
{
label1.Text = "RENK = " + data;
int m = Convert.ToInt16(data);
if (m == 0)// mavi olma durumu
{
button1.BackColor = Color.Blue; //
timer1.Start();
timer2.Start(); // timer 1 ve 2 başla
}
else if (m == 3)// yesil olma durumu
{ 21

button1.BackColor = Color.Green; /
timer1.Start();
timer2.Start();// timer 1 ve 2 başla }
else if (m == 2) //kırmızı olma durumu
{button1.BackColor = Color.Red; //
timer1.Start();
timer2.Start();// timer 1 ve 2 başla
}
else if (m == 1) //renk olmama durumu
{
button1.BackColor = Color.Black; //
button1.Location = new Point(100, 101);
// sonra buton eski koordinatına dönsün
}}
private void timer1_Tick(object sender, EventArgs e)
{
button1.Location = new Point(100, 101); //
if (button1.BackColor == Color.Blue)
{ for (t = 0; t<= 36; t++)
.*/
{ Thread.Sleep(25);
button1.Location = new Point(button1.Location.X + 10, 100);
}
button1.BackColor = Color.Black; //Kutunun rengini Siyah yap
button1.Location = new Point(100, 101); /*X=100,Y=101 konumuna gel Kutunun rengini Siyah yap*/
}
if (button1.BackColor == Color.Green) // Yeşil Ürün ise
{for (y = 0; y <=36; y++)
{ Thread.Sleep(25);
button1.Location = new Point(button1.Location.X + 10, 100);
}
button1.BackColor = Color.Black; //Kutunun rengini Siyah yap
button1.Location = new Point(100, 101); /
}
if (button1.BackColor == Color.Red) // Yeşil Ürün ise 22
{ for (p = 0; p <= 36; p++)
{ Thread.Sleep(25);
button1.Location = new Point(button1.Location.X + 10, 100);
}
button1.BackColor = Color.Black; //Kutunun rengini Siyah yap
button1.Location = new Point(100, 101);
} } }
```

```
private void button2_Click(object sender, EventArgs e)
{
timer1.Start();
timer2.Start();
}
private void timer2_Tick(object sender, EventArgs e)
{
if (button1.Location.X == 100 && button1.BackColor ==
Color.Blue)
{
i += 1;

 listBox1.Items.Add(i); //Mavi depo listesine Ekle
}
if (button1.Location.X == 100 && button1.BackColor ==
Color.Green)
{
c += 1;
listBox3.Items.Add(c); //Yeşil depo listesine Ekle
}
if (button1.Location.X == 100 && button1.BackColor ==
Color.Red)
{
b = b + 1;
listBox2.Items.Add(b); //
}}
```