

C#과 아두이노를 이용한 컨베이어 벨트 자동 분류 및 분류 개수 확인 시스템

□ 아두이노 코드

```
#include <Servo.h>
#include <SimpleTimer.h>
#include <MsTimer2.h>

SimpleTimer timer;
Servo microServo; //서보모터 클래스
int echopin = 12;  //초음파
int trigpin = 13;  //초음파
int solenoid_pin=11;
int servoPin=3; //서보모터 핀
int switchpin = 6;
float duration, distance;
char inchar;

int xp = A1;
int yp = A0;

void Timer(){
    if(inchar=='1'){ //Solenoid
        delay(500);
        digitalWrite(solenoid_pin,HIGH); //핀에 HIGH 입력
        delay(500);
        digitalWrite(solenoid_pin,LOW); //핀에 LOW 입력
        delay(500);
        Serial.write('a');
    }
    if(inchar=='2'){ //LeftServo
        microServo.write(0);
        delay(500);
        microServo.write(90); //각도를 서보모터에 반영
        delay(500); //정확한 동작을 위해 딜레이
        Serial.write('b');
    }
}
```

```

    if(inchar=='3'){ //RightServo
        microServo.write(90); //각도를 서보모터에 반영
        delay(500); //정확한 동작을 위해 딜레이
        microServo.write(0); //각도를 서보모터에 반영
        delay(500); //정확한 동작을 위해 딜레이
        Serial.write('c');
    }
    inchar='0';
    delay(50);
}

void Timer2(){
    int Xval = analogRead(xp);
    int Yval = analogRead(yp);
    int mouseClick = digitalRead(switchpin);

    //Serial.print(mouseClick);

    if(mouseClick == 0){
        Serial.write('z');
    }

    if(Yval==1023 || Yval==1022) // 이 범위 일 때 마우스 아래로
    {
        Serial.write('d');
    }

    if(Yval== 0) // 마우스 움직임 위로
    {
        Serial.write('u');
    }

    if(Xval==1023 || Xval==1022) // 마우스 움직임 위로 왼쪽
    {
        Serial.write('l');
    }

```

```

}

if(Xval==0) // 마우스 움직임 오른쪽
{
    Serial.write('r');
}
}

void setup() {
    Serial.begin(115200);
    pinMode(switchpin, INPUT_PULLUP);
    pinMode(trigpin, OUTPUT);
    pinMode(echopin, INPUT);
    pinMode(solenoid_pin,OUTPUT);
    pinMode(servoPin,OUTPUT);
    microServo.attach(servoPin); //서보모터 핀으로부터 서보모터 작동
    MsTimer2::set(50,Timer2);
    MsTimer2::start();
}

void loop() {

    // 초음파를 보낸다. 다 보내면 echo가 HIGH 상태로 대기하게 된다.
    digitalWrite(trigpin, HIGH);
    delay(10);
    digitalWrite(trigpin, LOW);

    // echoPin 이 HIGH를 유지한 시간을 저장 한다.
    duration = pulseIn(echopin, HIGH);
    // HIGH 였을 때 시간(초음파가 보냈다가 다시 들어온 시간)을 가지고 거리를 계산
    한다.
    distance = ((float)(340 * duration) / 10000) /2;

```

```

delay(100);
Serial.print(distance);
Serial.println();

if(Serial.available()){
    inchar=Serial.read();
}

if(distance>=6.00 && distance <=8.00){
    timer.setTimeout(600,Timer);
}

timer.run();
}

```

□ C# 코드

```

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;
using Utilities;
using System.Runtime.InteropServices;
using System.Threading;

namespace WindowsFormsApplication7
{
    public partial class Form1 : Form
    {
        globalKeyboardHook gkh = new globalKeyboardHook();
        globalKeyboardHook gkh2 = new globalKeyboardHook();
    }
}

```

```
[DllImport("user32.dll", CharSet = CharSet.Auto, CallingConvention =
CallingConvention.StdCall)]
```

```
public static extern void mouse_event(uint dwFlags, uint dx, uint dy,
uint cButtons, uint dwExtraInfo);
```

```
//Mouse actions
```

```
private const int MOUSEEVENTF_LEFTDOWN = 0x02; //마우스 이벤트
```

활용

```
private const int MOUSEEVENTF_LEFTUP = 0x04;
```

```
private const int MOUSEEVENTF_RIGHTDOWN = 0x08;
```

```
private const int MOUSEEVENTF_RIGHTUP = 0x10;
```

```
string send_string = "";
```

```
byte[] send_buffer = new byte[1]; //보내는 버퍼 1바이트
```

```
byte[] recv_buffer = new byte[1]; //받는 버퍼
```

```
int cnt_Solenoid = 0; //솔레노이드 카운트 변수
```

```
int cnt_LeftServo = 0; //왼쪽분류 카운트 변수
```

```
int cnt_RightServo = 0; //오른쪽 분류 카운트 변수
```

```
byte[] readData = new byte[1];
```

```
SerialPort port = new SerialPort("COM4", 115200);
```

```
bool stop = false;
```

```
bool pressed = false;
```

```
bool clickHistory = false;
```

```
public void DoMouseClicked() //마우스 클릭
```

```
{
```

```
//Call the imported function with the cursor's current position
```

```
uint X = (uint)Cursor.Position.X; //커서위치 x,y좌표
```

```
uint Y = (uint)Cursor.Position.Y;
```

```
mouse_event(MOUSEEVENTF_LEFTDOWN | MOUSEEVENTF_LEFTUP, X,
```

```
Y, 0, 0);
```

```
}
```

```
public Form1()
```

```
{
```

```
InitializeComponent();
```

```
backgroundWorker1 = new BackgroundWorker();
```

```

        backgroundWorker1.DoWork += new
        DoWorkEventHandler(backgroundWorker1_DoWork);
        port.Open(); // 아두이노 포트와 연결

        gkh.HookedKeys.Add(Keys.LControlKey);
        gkh2.HookedKeys.Add(Keys.F12);
        gkh.KeyDown += new KeyEventHandler(gkh_KeyDown);
        gkh2.KeyDown += new KeyEventHandler(gkh2_KeyDown);
        gkh.KeyUp += new KeyEventHandler(gkh_KeyUp);

    }

    private void Form1_Load(object sender, EventArgs e)
    {
        backgroundWorker1.RunWorkerAsync();//백그라운드작업 시작
    }
    void gkh_KeyDown(object sender, KeyEventArgs e) //눌렀는지 확인하기
    위해
    {
        pressed = true;
    }

    void gkh_KeyUp(object sender, KeyEventArgs e)
    {
        pressed = false;
    }

    void gkh2_KeyDown(object sender, KeyEventArgs e)
    {
        if (pressed == true) //true일 때 멈추기
        {
            stop = true;
            this.Show();
        }
    }
}

```

```

private void backgroundWorker1_DoWork(object sender,
DoWorkEventArgs e) //백그라운드에 넣음
{
    while (true)
    {
        Thread.Sleep(10);
        port.Read(readData, 0, 1);

        if (readData[0] == 'z')
        {
            mouse_event(MOUSEEVENTF_LEFTDOWN |
MOUSEEVENTF_LEFTUP, (uint)Cursor.Position.X, (uint)Cursor.Position.Y, 0, 0);
        }
        if (readData[0] == 'u') // 조이스틱 위로 움직이기 위해 데이터
읽고 y-10으로 이동
        {
            Cursor.Position = new Point(Cursor.Position.X,
Cursor.Position.Y - 10);
        }
        else if (readData[0] == 'd') // 조이스틱 밑으로 움직이기 위해 데
이터 읽고 y+10으로 이동
        {
            Cursor.Position = new Point(Cursor.Position.X,
Cursor.Position.Y + 10);
        }
        else if (readData[0] == 'l') // 조이스틱 왼쪽으로 움직이기 위해 데
이터 읽고 x-10으로 이동
        {
            Cursor.Position = new Point(Cursor.Position.X - 10,
Cursor.Position.Y);
        }
        else if (readData[0] == 'r') // 조이스틱 오른쪽으로 움직이기 위
해 데이터 읽고 x+10으로 이동
        {
            Cursor.Position = new Point(Cursor.Position.X + 10,

```

```

Cursor.Position.Y);
    }

    else if (readData[0] == 'a') //Solenoid
    {
        cnt_Solenoid++;
        textBox1.Text = cnt_Solenoid.ToString();

    }
    else if (readData[0] == 'b') //LeftServo
    {
        cnt_LeftServo++;
        textBox2.Text = cnt_LeftServo.ToString();
    }
    else if (readData[0] == 'c') //RightServo
    {
        cnt_RightServo++;
        textBox3.Text = cnt_RightServo.ToString();

    }

}

}

private void notifyIcon1_MouseClick(object sender, MouseEventArgs e)//
마우스 클릭
{
    //notifyIcon1.Visible = false;
    //this.Show();
}

private void button1_Click(object sender, EventArgs e)
{
    send_buffer[0] = 49;    //버퍼에 '1' 입력
    port.Write(send_buffer, 0, 1);    //버퍼내용을 송신
}

```


}

```
private void button2_Click(object sender, EventArgs e)
```

```
{
```

```
    send_buffer[0] = 50;    //버퍼에 '2'입력
```

```
    port.Write(send_buffer, 0, 1); //버퍼내 송신
```

```
}
```

```
private void button3_Click(object sender, EventArgs e)
```

```
{
```

```
    send_buffer[0] = 51;    //버퍼에 '3'입력
```

```
    port.Write(send_buffer, 0, 1); //버퍼내용을 송신
```

```
}
```

```
}
```

```
}
```

□ 회로

