**Contatore pezzi con sensore ad ultrasuoni**

Il seguente progetto conta i pezzi in blocchi di cinque. Quindi, dà in uscita sia il numero totale di pezzi che il numero di blocchi e, fa accendere un led ogni volta che si raggiunge il numero 5. La visualizzazione avviene su remoto, su un display lcd e su seriale.

**Programma arduino**

#include <LiquidCrystal.h>

#include <SPI.h>

#include <Ethernet.h>

const int TRIG = 8;

const int ECHO= 9;

int addr = 0;

#define RS 8

#define EN 9

#define D7 7

#define D6 6

#define D5 5

#define D4 4

LiquidCrystal lcd( RS, EN, D4, D5, D6, D7 ); // 'lcd' è una variabile di tipo LiquidCrystal \*/

boolean lettura=LOW;

byte mac[] = {

0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED

};

IPAddress ip(172, 16, 200, 243);

EthernetServer server(80);

int n=0, m=0, k=0,t=0;

const int led=4;

void setup() {

lcd.begin( 2, 16 );

Serial.begin(9600);

pinMode(TRIG,OUTPUT);

pinMode(ECHO,INPUT);

pinMode(led,OUTPUT);

Ethernet.begin(mac,ip);

server.begin();

}

void loop()

{delay(30);

digitalWrite(led,LOW);

lcd.clear();

lcd.setCursor( 0, 0 );

long durata, distanza;

digitalWrite(TRIG, LOW);

delayMicroseconds(2);

digitalWrite(TRIG, HIGH);

delayMicroseconds(10);

digitalWrite(TRIG, LOW);

durata = pulseIn(ECHO,HIGH);

distanza = durata / 29.1 / 2 ;

if (distanza <= 0){

Serial.println("Out of range");

}

else {

//Serial.print(distanza);

// Serial.println("cm");

//Serial.println();

if(distanza<15 && distanza>3){

m=m+1;

if(n==5){

n=0; k=k+1;

lcd.print( k );

digitalWrite(led,HIGH);

}

else{n=n+1;digitalWrite(led,LOW);}

}

}

Serial.println("pezzi");

Serial.println(m);

Serial.println("somma parziale");

Serial.println(n);

Serial.println("blocchi");

Serial.println(k); EthernetClient client = server.available();

if (client) {

Serial.println("new client");

boolean currentLineIsBlank = true;

while (client.connected()) {

if (client.available()) {

char c = client.read();

Serial.write(c);

if (c == '\n' && currentLineIsBlank) {

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("Connection: close");

client.println("Refresh: 5");

client.println();

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.println("<p />");

client.println("<H1>");

client.println("n=");

client.print(n);

client.println("</H1>");

client.println("<p />");

client.println("<H2>");

client.print("k= ");

client.println("</H2>");

client.println("<H1>");

client.print(k);

client.println("</H1>");

client.println("</html>");

break;

}

if (c == '\n') {

currentLineIsBlank = true;

}

else if (c != '\r') {

currentLineIsBlank = false;

}

Serial.println("ingressi\n");

}}

} }

**Programma in Visual basic per la visualizzazione dei dati**

Imports System

Imports System.ComponentModel

Imports System.Threading

Imports System.IO.Ports

Public Class frmMain

Dim myPort As Array

Delegate Sub SetTextCallback(ByVal [text] As String)

Private Sub frmMain\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

myPort = IO.Ports.SerialPort.GetPortNames()

cmbBaud.Items.Add(9600)

cmbBaud.Items.Add(19200)

cmbBaud.Items.Add(38400)

cmbBaud.Items.Add(57600)

cmbBaud.Items.Add(115200)

For i = 0 To UBound(myPort)

cmbPort.Items.Add(myPort(i))

Next

cmbPort.Text = cmbPort.Items.Item(0)

cmbBaud.Text = cmbBaud.Items.Item(0)

btnDisconnect.Enabled = False

End Sub

Private Sub btnConnect\_Click(sender As Object, e As EventArgs) Handles btnConnect.Click

SerialPort1.PortName = cmbPort.Text

SerialPort1.BaudRate = cmbBaud.Text

SerialPort1.Parity = IO.Ports.Parity.None

SerialPort1.StopBits = IO.Ports.StopBits.One

SerialPort1.DataBits = 8

SerialPort1.Open()

btnConnect.Enabled = False

btnDisconnect.Enabled = True

End Sub

Private Sub btnDisconnect\_Click(sender As Object, e As EventArgs) Handles btnDisconnect.Click

SerialPort1.Close()

btnConnect.Enabled = True

btnDisconnect.Enabled = False

End Sub

Private Sub btnSend\_Click(sender As Object, e As EventArgs) Handles btnSend.Click

SerialPort1.Write(txtTransmit.Text)

End Sub

Private Sub SerialPort1\_DataReceived(ByVal sender As Object, ByVal e As System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived

ReceivedText(SerialPort1.ReadExisting())

End Sub

Private Sub ReceivedText(ByVal [text] As String)

If Me.rtbReceived.InvokeRequired Then

Dim x As New SetTextCallback(AddressOf ReceivedText)

Me.Invoke(x, New Object() {(text)})

Else

Me.rtbReceived.Text &= [text]

End If

End Sub

Private Sub cmbPort\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cmbPort.SelectedIndexChanged

If SerialPort1.IsOpen = False Then

SerialPort1.PortName = cmbPort.Text

Else

MsgBox("Valid only if port is Closed", vbCritical)

End If

End Sub

Private Sub cmbBaud\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cmbBaud.SelectedIndexChanged

If SerialPort1.IsOpen = False Then

SerialPort1.BaudRate = cmbBaud.Text

Else

MsgBox("Valid only if port is Closed", vbCritical)

End If

End

Private Sub GroupBox1\_Enter(sender As Object, e As EventArgs) Handles GroupBox1.Enter

End Sub

End Class