|  |  |
| --- | --- |
| HÜ 12 | Abgabetermin: 8.12.2021 23:59 |
| Name: Johannes Werner | Aufwand in h: ca. 8h |
|  | Punkte: |

## UDP - Temperatur 16 Punkte

Entwerfen Sie am Arduino einen Sketch um einen Temperatursensor auszulesen. Senden Sie den Wert per UDP an die Broadcast Adresse des Netzwerkes.

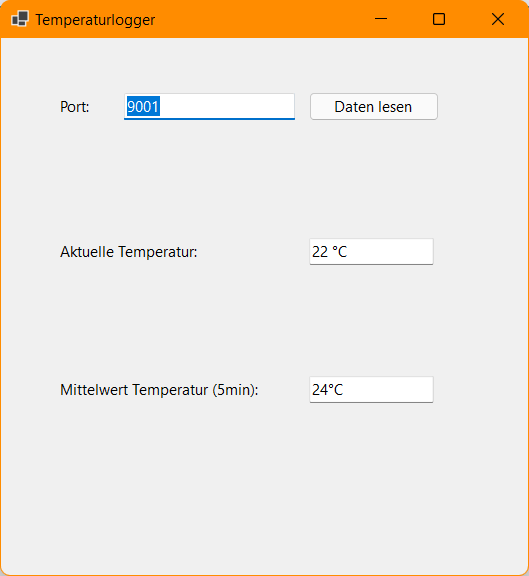
Entwickeln Sie eine grafische Benutzeroberfläche, die den Wert vom Netzwerk liest. Dabei soll der aktuelle Wert und ein Mittelwert der letzten 5 Minuten angezeigt werden.

Hinweis: Verwenden Sie zum Entwickeln eine entsprechend kürzere Zeitspanne (z.B.: 20 sec). Sie können den Temperatursensor mit Ihrer Handwärme relativ einfach erwärmen.

Erstellen Sie dazu eine Klasse Messwert mit zwei Properties: Zeitpunkt und Messwert. In einer generischen Liste können Sie Objekte der Klasse Messwert speichern. Neue Messwerte werden der Liste hinzugefügt und Messwerte, deren Zeit abgelaufen ist, können aus der Liste entfernt werden.

Zusätzlich soll ein Textfeld zur Eingabe des verwendeten Ports zur Verfügung stehen. Der Port soll zu Beginn spezifiziert werden können und nach Klick des Buttons „Daten lesen“, sollen die Messwerte gelesen werden. Danach soll der Button deaktiviert werden.

Freiwillige Zusatzübung: „Daten lesen“ soll während der Laufzeit mehrere Male gedrückt werden können, damit der Port gewechselt werden kann.



Sie dürfen das Design der grafischen Benutzeroberfläche natürlich gerne optisch ansprechender gestalten :-)

Testen Sie Ihr Programm ausführlich. Achten Sie auch darauf, dass im Netzwerk auch andere „Daten“ als die Messwerte per UDP gesendet werden können.

# Lösungsidee

Ich habe mir zu aller erst alle relevanten und hilfreichen Codestücke aus den Übungen des letzten Jahrs zusammen kopiert. Danach habe ich die GUI entworfen und die Codestücke zu einem funktionierenden Code zusammengeführt. Danach probierte ich noch die Zusatzübung zu lösen, wo ich jedoch durch ein Problem es nicht lösen konnte, weshalb ich den Code wieder zurück gebaut habe.

# Code

Arduino

#include <SPI.h>

#include <Ethernet.h>

#include <EthernetUdp.h>

#include <String.h>

#include "DHT.h"

#define DHTPIN 2

#define DHTTYPE DHT11

static byte mac[] = { 0x12,0x34,0x56, 0x78, 0x90, 0x22};

static const int port = 8888;

static IPAddress ip(255, 255, 255, 255);

DHT dht(DHTPIN, DHTTYPE);

EthernetUDP udp;

void setup() {

  Ethernet.init(10);

  Ethernet.begin(mac);  // IP. Gateway, Subnet, DNS vom DHCP

  udp.begin(port);

  Serial.begin(9600);

  while(!Serial){

    // warten bis der serielle Kommunikationspartner verfügbar

  }

  // Fehlerbehandlung

  if(Ethernet.hardwareStatus() == EthernetNoHardware){

    Serial.println("Kein Ethernet Shield vorhanden");

    while (true){

      delay(100); // ohne Ethernet Shield, machen wir nichts

    }

  }

  if(Ethernet.linkStatus() == LinkOFF){

    Serial.println("Kein Ethernet Kabel verbunden");

  }

  Serial.print("IP: ");

  Serial.println(Ethernet.localIP());

  Serial.print("Sending port: ");

  Serial.println(port);

}

void loop() {

  float temperature = dht.readTemperature();

  if (!isnan(temperature)) {

    String message = "-temperature measurement system werner-" + String(temperature , 2) + "###";

    Serial.println(message);

    udp.beginPacket( ip, port);

    udp.print(message);

    udp.endPacket();

  }

  delay(3000);

}

C#

Measurment Class

namespace Hue01\_Csharp

{

internal class Measurment

{

public DateTime Time { get;}

public double Value { get;}

public Measurment(DateTime time, double value)

{

this.Time = time;

this.Value = value;

}

}

}

GUI Code

using System.Net.Sockets;

using System.Net;

using System.Text;

using System.Diagnostics.Metrics;

namespace Hue01\_Csharp

{

public partial class Form1 : Form

{

private UdpClient udpClient;

private List<Measurment> measurments = new List<Measurment>();

private String serverIdentifier = "-temperature measurement system werner-";

private IPEndPoint receiveAdr;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void btReadData\_Click(object sender, EventArgs e)

{

receiveAdr = new IPEndPoint(IPAddress.Any, int.Parse(this.nud\_port.Value.ToString()));

udpClient = new UdpClient(receiveAdr);

this.ReceiveAsync();

this.btReadData.Enabled = false;

this.nud\_port.ReadOnly = true;

}

private async void ReceiveAsync() // async bedeutet, dass die Methode im Hintergrund ausgeführt wird

{

while (true)

{

UdpReceiveResult result = await udpClient.ReceiveAsync();

string value = Encoding.UTF8.GetString(result.Buffer);

if (value.Contains(serverIdentifier))

{

value = value.Substring(value.IndexOf(serverIdentifier) + serverIdentifier.Length, value.IndexOf("###") - value.IndexOf(serverIdentifier) - serverIdentifier.Length).Replace(".", ",");

this.tbCurrentTemperature.Text = value;

measurments.Add(new Measurment(DateTime.Now, double.Parse(value)));

double averageMeasurment = 0;

List<Measurment> newMeasurments = new List<Measurment>();

measurments.ForEach(measurment =>

{

if (measurment.Time.CompareTo(DateTime.Now.Subtract(new TimeSpan(0, 5, 0))) > 0)

{

newMeasurments.Add(measurment);

averageMeasurment += measurment.Value;

}

});

measurments = newMeasurments;

averageMeasurment = averageMeasurment/measurments.Count;

this.tbAverageTemperature.Text = averageMeasurment.ToString("0.00");

}

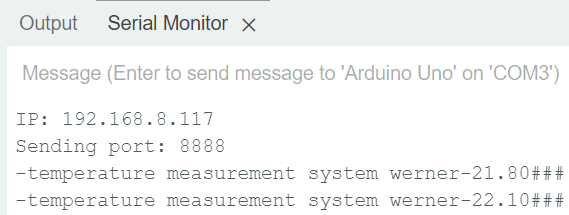
}

}

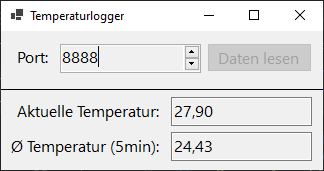
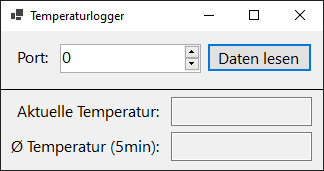
}

}

# Ergebnis



Info: der zusätzliche Text ist hier um mögliche andere UDP packeges raus zu filtern.



Link zu einen Video:

https://youtube.com/shorts/RrgvDk\_3-8A