

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Automatizálási és Alkalmazott Informatikai Tanszék

Beágyazott és irányító rendszerek specializáció – Számítógép-alapú rendszerek ágazat

**Mikrokontroller laboratórium**

Jegyzőkönyv

8. mérés

Document-View architektúra

|  |  |
| --- | --- |
| **Mérést végző hallgatók:** | Nyiri Levente ULYHQH |
| **Mérőcsoport:** | SDU4 |
| **Mérés időpontja:** | 2024.08.05 du |
| **Mérés helyszíne:** | BME AUT, labor: QB. 127 |
| **Mérésvezetők:** | Hideg Attila |
|  |  |
|  |  |

# Felhasznált eszközök

|  |  |  |
| --- | --- | --- |
| [eszköz neve] | [eszköz típusa] | [gyári v. leltári szám] |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Mérési feladatok

# Mérési értékek reprezentálása, saját dokumentum osztály, adatok mentése és betöltése

2.1:

namespace Signals

{

public class SignalValue

{

private double \_value = 0;

private DateTime timeStamp;

public double Value

{

get { return \_value; }

set { \_value = value; }

}

public DateTime TimeStamp

{

get { return timeStamp; }

set { timeStamp = value; }

}

}

}

2.2:

namespace Signals

{

public class SignalDocument : Document

{

private List<SignalValue> SignalValues;

public SignalDocument(string name) : base(name)

{

}

}

}

SignalDocument doc = new SignalDocument(form.DocName);

documents.Add(doc);

createView(doc, true);

2.3:

Konstruktorok:

public SignalValue(double value, DateTime timeStamp)

{

TimeStamp = timeStamp;

Value = value;

}

public SignalDocument(string name) : base(name)

{

SignalValues = new List<SignalValue>();

DateTime time = new DateTime(2024, 8, 4, 23, 58, 30, 999);

SignalValues.Add(new SignalValue(1, time));

SignalValues.Add(new SignalValue(4,new DateTime(2024, 8, 30, 13, 40, 32, 123)));

SignalValues.Add(new SignalValue(14, new DateTime(2024,9, 1,12,10,50)));

SignalValues.Add(new SignalValue(20, new DateTime(2024,9,4,15,30,700)));

SignalValues.Add(new SignalValue(10, new DateTime(2024,9,6,4,40,322)));

}

App. SaveActiveDocument:

public void SaveActiveDocument()

{

if (ActiveDocument == null)

return;

// Útvonal bekérése a felhasználótól a SaveFileDialog segítségével.

// http://msdn.microsoft.com/en-us/library/system.windows.forms.savefiledialog.aspx

// ..

SaveFileDialog saveFileDialog1 = new SaveFileDialog();

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

string path = saveFileDialog1.FileName;

// A dokumentum adatainak elmentése.

ActiveDocument.SaveDocument(path);

}

}

SaveDocument:

public override void SaveDocument(string filePath)

{

using (StreamWriter sw = new StreamWriter(filePath))

{

foreach (SignalValue q in SignalValues)

{

string dt = q.TimeStamp.ToUniversalTime().ToString("o");

//q.Value.ToString()

sw.WriteLine($"{q.Value.ToString()}\t{dt}");

}

}

}

# feladat

Feladat rövid megfogalmazása

[xyz]

Feladat megoldása

[xyz]

stb.

# Összefoglalás

[A mérés eredményeinek összefoglalása]

# Függelék

[Programlista, stb.]