

```

using namespace System::IO; // fájlkezelés
using namespace System::Windows::Forms::DataVisualization::Charting; // chart
using namespace System::Collections::Generic; //lista

// Szűrő osztály készítése
ref class Filter
{
private:
    int window; // szűrő ablak mérete
    List<double> ^ data; // nyers adatok

public:
    // konstruktor
    Filter(int size)
    {
        window = size;
        data = gcnew List<double>();
    }

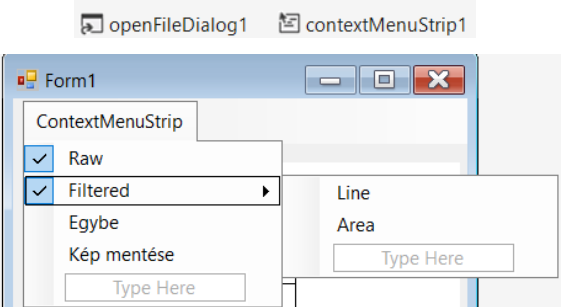
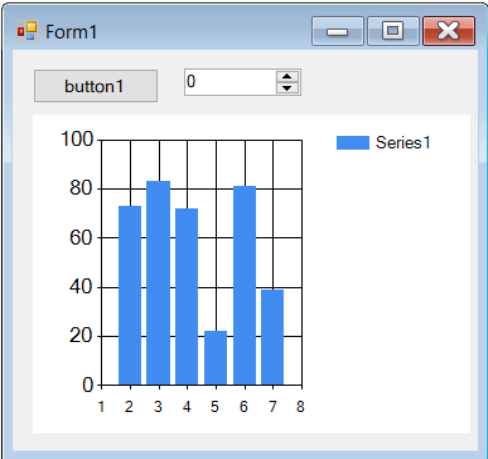
    // mozgó átlag abszolút értékkel
    double AbsAVG(double adat)
    {
        data->Add(Math::Abs(adat));
        double value = 0;
        int size = data->Count;
        if (size < window) return Double::NaN;
        if (size == window)
        {
            for (int i = 0; i < size; i++)
                value += Math::Abs(data[i]);
            data->RemoveAt(0);
        }
        return value/window;
    }

    // RMS szűrő
    double RMS(double adat)
    {
        data->Add(adat);
        double value = 0;
        int size = data->Count;
        if (size < window) return Double::NaN;
        if (size == window)
        {
            for(int i=0; i<size; i++)
                value += data[i] * data[i];
            data->RemoveAt(0);
        }
        return Math::Sqrt(value / window);
    }
};

// ablak betölt
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
    this->Text = "Grafikon";
    button1->Text = "Megnyitas";
    // szűrő időablakának beállítása [100, 500] tartományra
    numericUpDown1->Minimum = 100;
    numericUpDown1->Maximum = 500;
    // chart inicializálása
    chart_init();
    // chart lehoronyzása az ablakhoz, az órán a properties felületen állítottuk be
    chart1->Anchor = System::Windows::Forms::AnchorStyles::Top
        | System::Windows::Forms::AnchorStyles::Bottom
        | System::Windows::Forms::AnchorStyles::Left
        | System::Windows::Forms::AnchorStyles::Right;
    // contextMenuStrip1 hozzárendelése a chart1-hez
    chart1->ContextMenuStrip = contextMenuStrip1;
    // a contextMenuStrip1 elemeit kézzel vittük fel
}

// chart inicializálása

```



```

private: void chart_init()
{
    // eddigi sorozatok és területek törlése
    chart1->Series->Clear();
    chart1->ChartAreas->Clear();
    // hátterek átlátszóra állítása
    chart1->BackColor = Color::Transparent;
    chart1->Legends[0]->BackColor = Color::Transparent;

    // új terület(ek) létrehozása
    ChartArea ^charea = chart1->ChartAreas->Add("Lap1");
    charea->AxisX->Title = "Time [ms]";
    charea->AxisY->Title = "EMG [mV]";
    //charea->AxisX->LabelStyle->Format = "{0.00}ms"; //tengely formázása
    charea->AxisX->Interval = 1000;
    charea->AxisX->Minimum = 0;
    charea->AxisX->Maximum = 9000;
    charea->BackColor = Color::Transparent;

    charea = chart1->ChartAreas->Add("Lap2");
    charea->AxisX->Title = "Time [ms]";
    charea->AxisY->Title = "EMG [mV]";
    charea->AxisX->Interval = 1000;
    charea->AxisX->Minimum = 0;
    charea->AxisX->Maximum = 9000;
    charea->BackColor = Color::Transparent;

    // Sorozatok hozzáadása
    Series ^seriesRaw = chart1->Series->Add("Raw");
    seriesRaw->ChartArea = "Lap1";
    seriesRaw->ChartType = SeriesChartType::Line;
    seriesRaw->Color = Color::Orange;

    Series ^seriesRMS = chart1->Series->Add("Filtered");
    seriesRMS->ChartArea = "Lap2";
    seriesRMS->ChartType = SeriesChartType::Line;
    seriesRMS->Color = Color::Red;
    seriesRMS->BorderWidth = 3;
}

// megnyitás gomb
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    // chart inicializálása
    chart_init();
    // contextMenuStrip1 elemeinek visszaállítása
    filteredToolStripMenuItem->Checked = true;
    rawToolStripMenuItem->Checked = true;
    egybeToolStripMenuItem->Text = "Egybe";
    // file open
    openFileDialog1->FileName = "";
    openFileDialog1->Filter = "EMG data|*.emg";
    if (openFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK)
    {
        // szűrő létrehozása ablakmérettel
        Filter ^ filter = gnew Filter((int)numericUpDown1->Value);
        // bináris fájl megnyitása
        BinaryReader^ br = gnew BinaryReader(File::Open(openFileDialog1->FileName,
        FileMode::Open));

        // fájl végéig
        while (br->BaseStream->Position != br->BaseStream->Length)
        {
            // bináris adatok beolvasása |int|double|double|
            int time = br->ReadInt32();
            double ch1 = br->ReadDouble();
            double ch2 = br->ReadDouble();

            // pont hozzáadása a sorozathoz
            chart1->Series["Raw"]->Points->AddXY(time, ch1);
            // szűrt érték hozzáadása a sorozathoz
            chart1->Series["Filtered"]->Points->AddXY(time, filter->RMS(ch1));
        }
    }
}

```

```

    // adott elem elérése: 5. elem az elejéről
    br->BaseStream->Seek(5 * (sizeof(int) + 2 * sizeof(double)), SeekOrigin::Begin);
    int time = br->ReadInt32();
    double ch1 = br->ReadDouble();
    double ch2 = br->ReadDouble();

    // fájl bezárása
    br->Close();
}
}

// mért érték ki/be kapcsolása
private: System::Void rawToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    chart1->Series["Raw"]->Enabled = rawToolStripMenuItem->Checked;
}

// szűrt érték ki/be kapcsolása
private: System::Void filteredToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    chart1->Series["Filtered"]->Enabled = filteredToolStripMenuItem->Checked;
}

// egybe/külön diagramok
private: System::Void egybeToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    if (chart1->ChartAreas->Count == 2) // ha 2 lap van
    {
        // szűrt érték áthelyezése az 1. lapra
        Series ^sr = chart1->Series["Filtered"];
        sr->ChartArea = "Lap1";
        // 2. lap kitörlésre kerül, hogy ne foglaljon helyet
        chart1->ChartAreas->RemoveAt(1);
        // szöveg átírása
        egybeToolStripMenuItem->Text = "Külön";
    }
    else
    {
        // 2. lap újbóli létrehozása
        ChartArea ^ charea = chart1->ChartAreas->Add("Lap2");
        charea->AxisX->Title = "Time [ms]";
        charea->AxisY->Title = "EMG [mV]";
        charea->AxisX->Minimum = 0;
        charea->AxisX->Interval = 1000;
        charea->AxisX->Maximum = 9000;
        charea->BackColor = Color::Transparent;
        // szűrt adatok áthelyezése a 2. lapra
        chart1->Series["Filtered"]->ChartArea = "Lap2";
        // szöveg átírása
        egybeToolStripMenuItem->Text = "Egybe";
    }
}

// szűrt adatok megjelenítése vonalasan
private: System::Void lineToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    chart1->Series["Filtered"]->ChartType = SeriesChartType::Line;
}

// szűrt adatok megjelenítése kitöltéssel
private: System::Void areaToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    chart1->Series["Filtered"]->ChartType = SeriesChartType::Area;
}

// mentés képként
private: System::Void képMentéseToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    chart1->SaveImage("../chart.png", ChartImageFormat::Png);
}

```