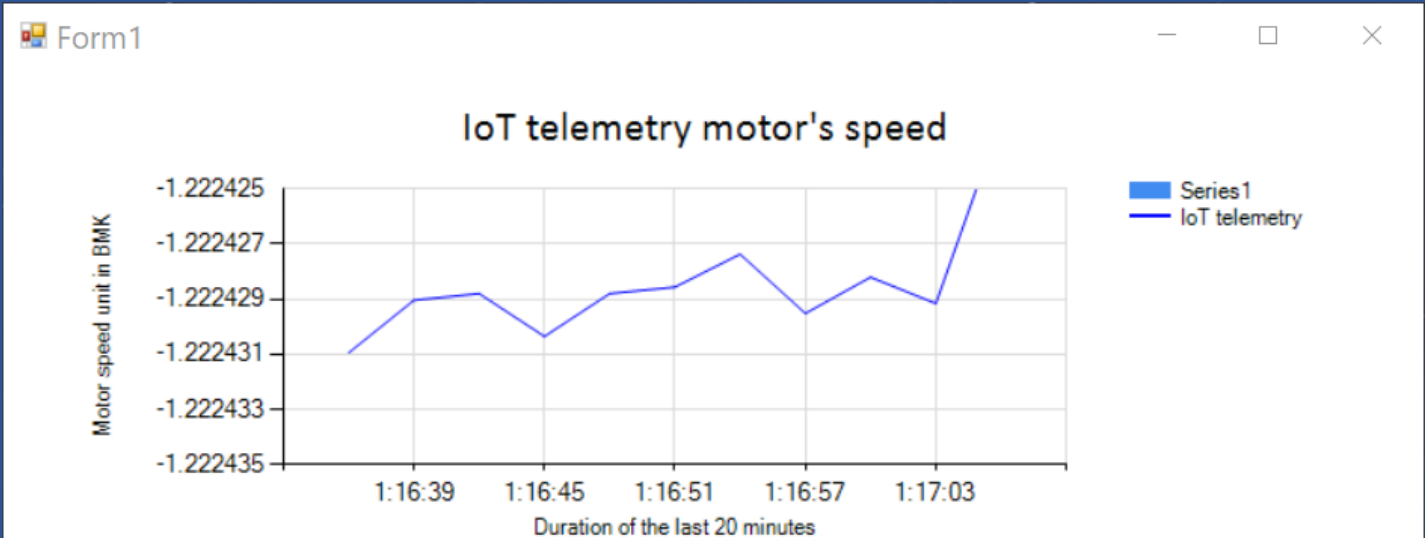
WinForm backend Line D2C

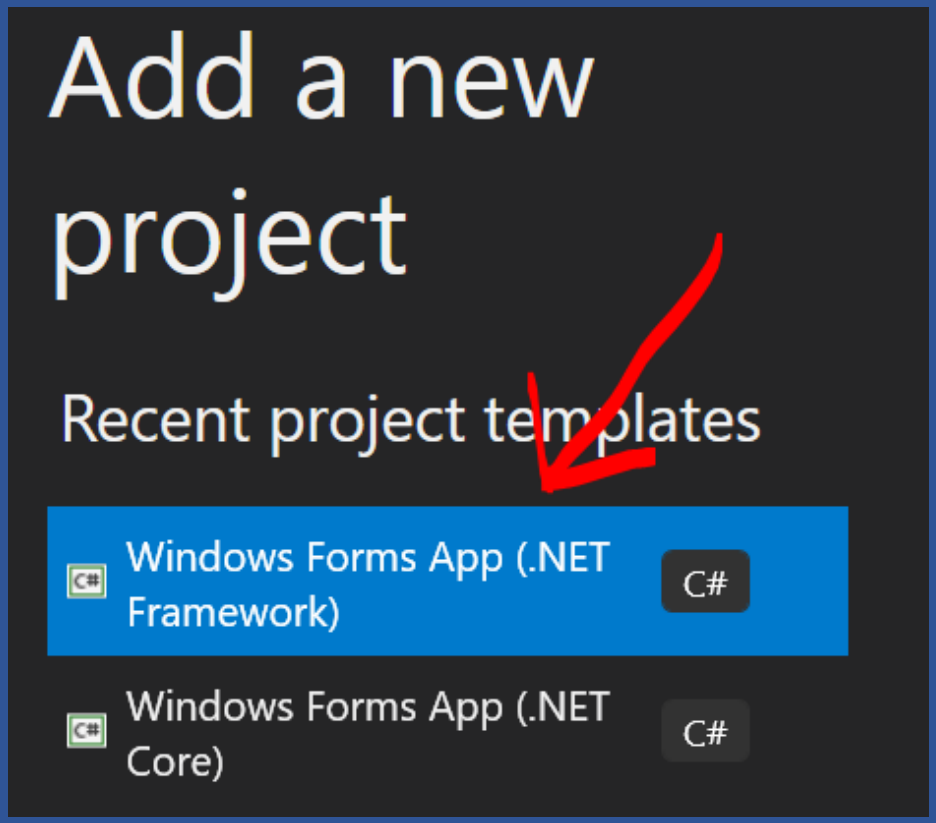
Loy Vanich (laploy@gmail.com 084 007 5544)

Create WinForm show a Line chart real-time D2C

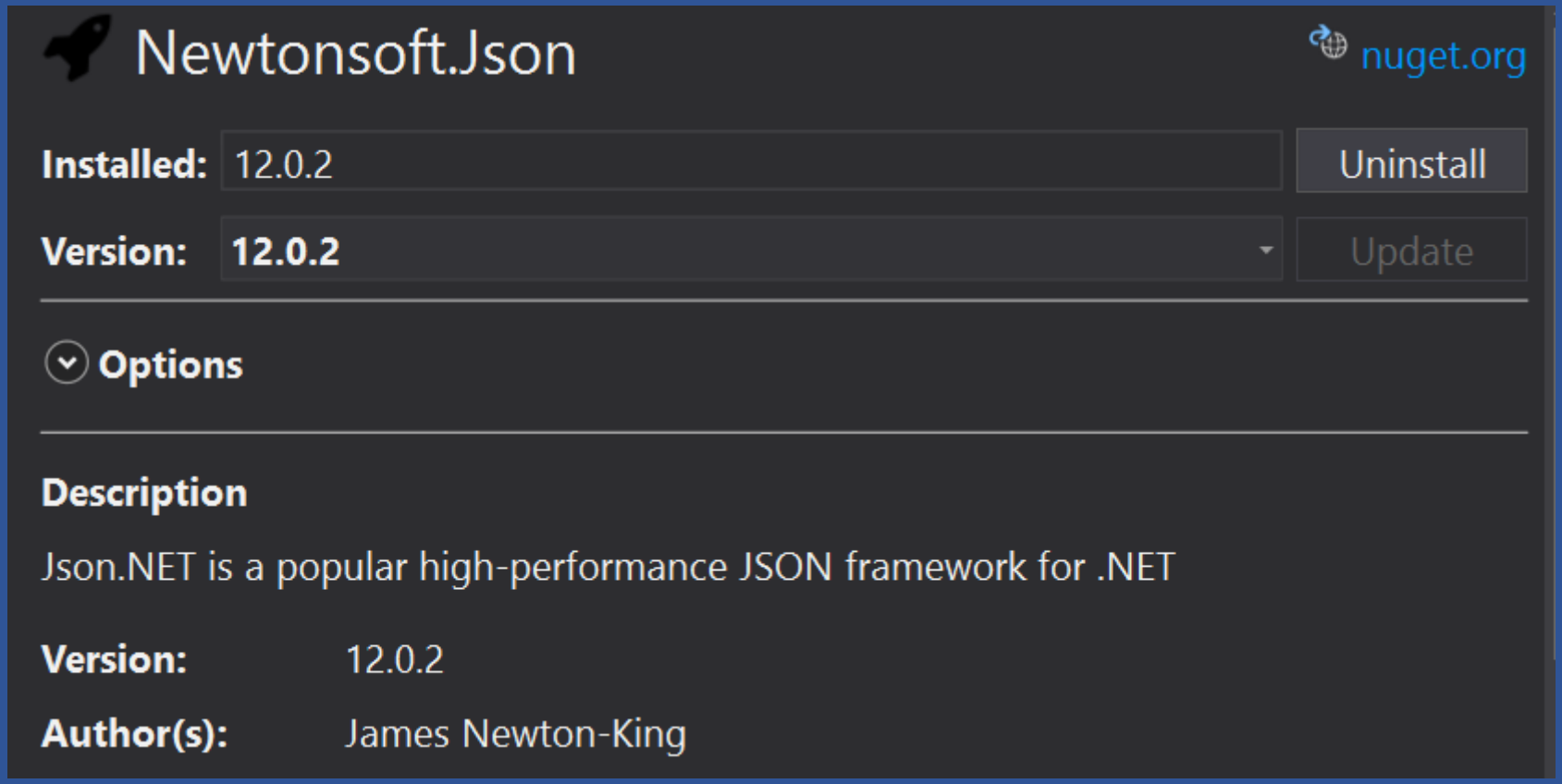


qtime	ambient	coolant	u_d	u_q	motor_speed	torque
1:16:36	-0.0395014733	1.43062127	0.330584645	-1.247891	-1.222431	-0.25563
1:16:39	-0.036167033	1.39542973	0.3308602	-1.24583161	-1.222429	-0.25563
1:16:42	-0.0358568765	1.35896814	0.331425577	-1.24944568	-1.2224288	-0.25563
1:16:45	-0.04257795	1.28793216	0.331577569	-1.246936	-1.22243035	-0.25563
1:16:48	-0.046077	1.20749342	0.3316244	-1.24791956	-1.2224288	-0.25563
1:16:51	-0.04590938	1.12158346	0.33131516	-1.24655354	-1.22242856	-0.25563
1:16:54	-0.0445864424	1.03432989	0.331142873	-1.24721479	-1.22242737	-0.25563
1:16:57	-0.0348029435	0.9407184	0.331035435	-1.24702883	-1.22242951	-0.25563

Open Visual Studio / Create C# .NET Framework WinForm



Add Newtonsoft.Json



Add new class file

```
3   public class Motor
4   {
5       public string qtime { get; set; }
6       public float ambient { get; set; }
7       public float coolant { get; set; }
8       public float u_d { get; set; }
9       public float u_q { get; set; }
10      public float motor_speed { get; set; }
11      public float torque { get; set; }
12      public float i_d { get; set; }
13      public float i_q { get; set; }
14      public float pm { get; set; }
15      public float stator_yoke { get; set; }
16      public float stator_tooth { get; set; }
17      public float stator_winding { get; set; }
18  }
19 }
```

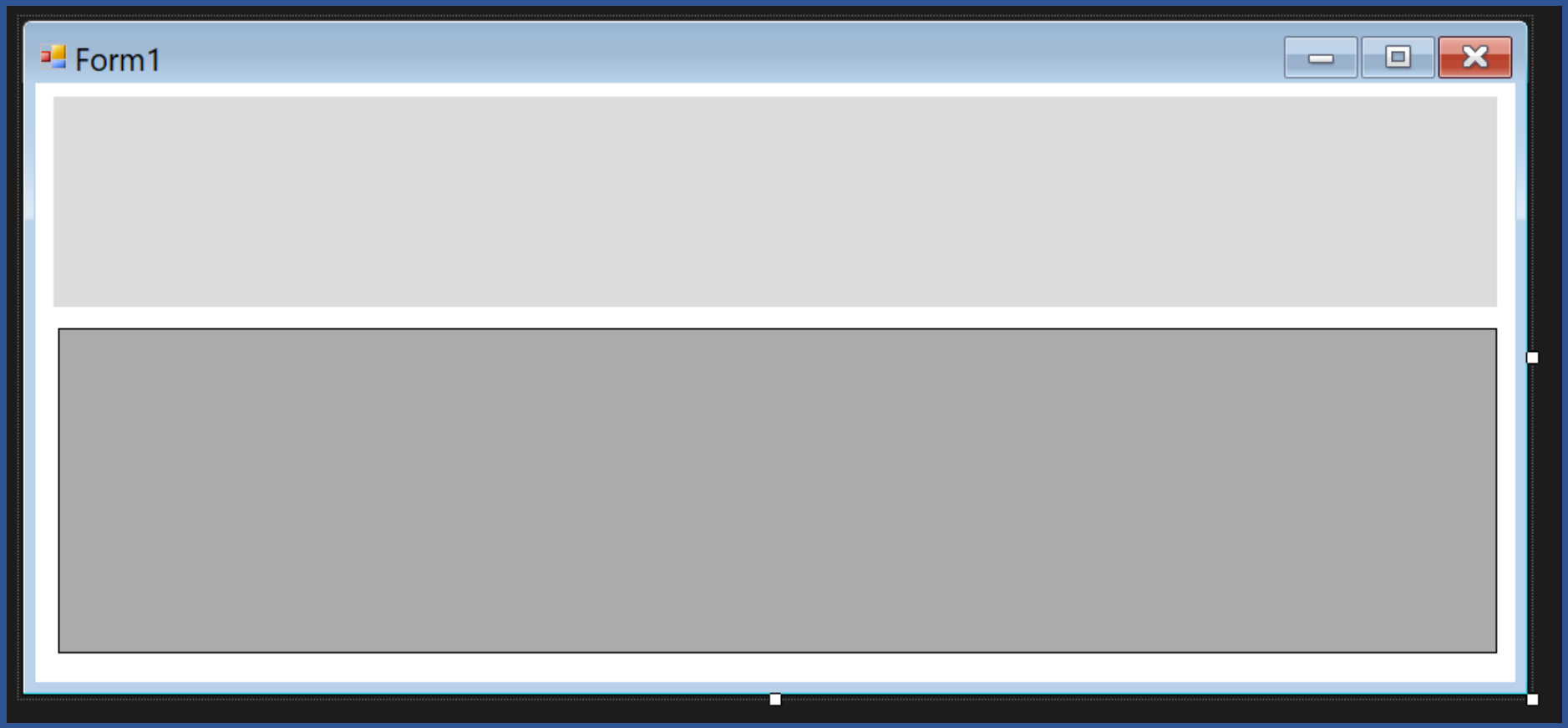
Add Namespace to Form1

```
1   using Microsoft.Azure.EventHubs;
2   using System;
3   using System.Collections.Generic;
4   using System.Text;
5   using System.Threading.Tasks;
6   using System.Windows.Forms;
7   using Newtonsoft.Json;
8   using System.ComponentModel;
9   using System.Windows.Forms.DataVisualization.Charting;
10  using System.Drawing;
```

Add class members

```
14      private Timer myTimer = new Timer();
15      private bool ready = true;
16      private List<Motor> datasetIoT = new List<Motor>();
17
18      private readonly string s_eventHubsCompatibleEndpoint =
19          "sb://ihsuprods[redacted].servicebus.windows.net/";
20      private readonly string s_eventHubsCompatiblePath =
21          "iothub-ehub-loyiot[redacted]4f3c";
22      private readonly string s_iotHubSasKey =
23          "KCyf3omKkmWncX[redacted]xBWRoWgmAw=";
24      private readonly string s_iotHubSasKeyName = "service";
25      private EventHubClient s_eventHubClient;
26      private PartitionReceiver eventHubReceiver;
```

Add a Label and a Chart control to Form1



Add method SetChartArea()

```
34      private void SetChartArea()
35      {
36          Title title1 = new Title();
37          title1.Font = new Font("Calibri", 16.2F, FontStyle.Regular,
38              GraphicsUnit.Point, ((byte)(0)));
39          title1.Name = "Title1";
40          title1.Text = "IoT telemetry motor's speed";
41          this.chart1.Titles.Add(title1);
42
43          chart1.ChartAreas[0].AxisX.MajorGrid.LineColor = Color.Gainsboro;
44          chart1.ChartAreas[0].AxisY.MajorGrid.LineColor = Color.Gainsboro;
45          chart1.ChartAreas[0].AxisY.Maximum = -1.2224250;
46          chart1.ChartAreas[0].AxisY.Minimum = -1.2224350;
47
48          chart1.ChartAreas[0].AxisX.Title = "Duration of the last 20 minutes";
49          chart1.ChartAreas[0].AxisX.TitleAlignment = StringAlignment.Center;
50          chart1.ChartAreas[0].AxisX.TextOrientation = TextOrientation.Horizontal;
51
52          chart1.ChartAreas[0].AxisY.Title = "Motor speed unit in BMK";
53          chart1.ChartAreas[0].AxisY.TitleAlignment = StringAlignment.Center;
54          chart1.ChartAreas[0].AxisY.TextOrientation = TextOrientation.Rotated270;
55
56          var speedSeries1 = new Series("IoT");
57          speedSeries1.ChartType = SeriesChartType.Line;
58          speedSeries1.Color = Color.Blue;
59          chart1.Series.Add(speedSeries1);
60          chart1.Series["IoT"].LegendText = "IoT telemetry";
61      }
```

Add Method ToTLine()

```
62      private void IoTLine()
63      {
64          List<double> yl = new List<double>();
65          List<string> xl = new List<string>();
66          foreach(var v in datasetIoT)
67          {
68              yl.Add(v.motor_speed);
69              xl.Add(v.qtime);
70          }
71          chart1.Series["IoT"].Points.DataBindXY(xl, yl);
72      }
```

Add method GetD2CMessage()

```
73  private async Task GetD2CMessage()
74  {
75      var events = await eventHubReceiver.ReceiveAsync(100);
76      if (events == null) { ready = true; return; }
77      foreach (EventData eventData in events)
78      {
79          string s = Encoding.UTF8.GetString(eventData.Body.Array);
80          Motor m = new Motor();
81          m = JsonConvert.DeserializeObject<Motor>(s);
82          m.qtime = DateTime.Now.ToString("h:mm:ss");
83          datasetIoT.Add(m);
84          var bindingList = new BindingList<Motor>(datasetIoT);
85          var source = new BindingSource(bindingList, null);
86          dataGridView1.DataSource = source;
87          ready = true;
88      }
89  }
```

Add code to Form1_Load

```
90      private void Form1_Load(object sender, EventArgs e)
91      {
92          SetChartArea();
93          myTimer.Enabled = true;
94          myTimer.Interval = 3000;
95          myTimer.Tick += MyTimer_Tick;
96
97          var connectionString = new EventHubsConnectionStringBuilder(
98              new Uri(s_eventHubsCompatibleEndpoint),
99              s_eventHubsCompatiblePath,
100             s_iotHubSasKeyName,
101             s_iotHubSasKey);
102         s_eventHubClient = EventHubClient.CreateFromConnectionString(
103             connectionString.ToString());
104         eventHubReceiver = s_eventHubClient.CreateReceiver(
105             "$Default",
106             "0",
107             EventPosition.FromEnqueuedTime(DateTime.Now));
108     }
```

Add code to myTimer_Tick

```
109     private async void MyTimer_Tick(object sender, EventArgs e)
110     {
111         if(ready)
112         {
113             ready = false;
114             await GetD2CMessage();
115             IoTLine();
116         }
117     }
```

What's next?