**Hướng dẫn sử dụng MongoDB Local để làm data logger và quản lý dữ liệu Database.**

**Tác giả: Trần Quốc Đông**

**Đơn vị công tác: Becamex SMIC**

**Lởi mở đầu:**

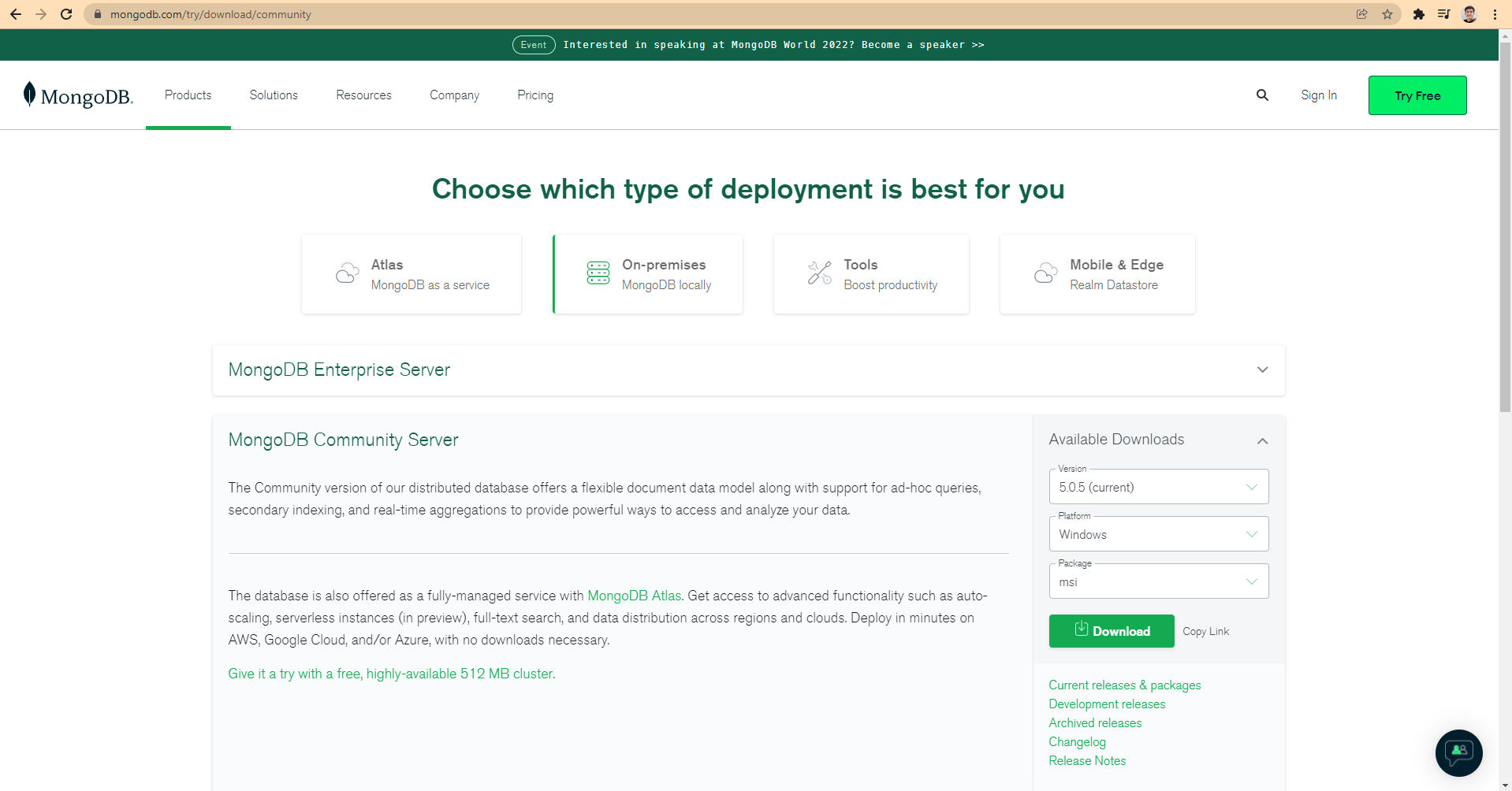
Cở sở dữ liệu phi quan hệ hay NoSQL database đang ngày càng được ứng dụng rộng rãi trên nhiều lĩnh nhờ sự dễ dàng sử dụng, dễ dàng tùy biến và dễ dàng mở rộng của nó. Không nằm ngoài xu hướng trên lĩnh vực tự động hóa công nghiệp trong những năm gần đây cũng tích cực đổi mới và thích nghi chung với xu hướng này. Trong bài viết tác giả sẽ hướng dẫn việc sử dung MongoDB – một CSDL NoSQL phổ biến hiện nay để lưu trữ và quản lý dữ liệu từ nhà máy thông qua một ứng dụng SCADA/HMI được xây dựng trên nền tảng Winform c# (Chương 1,2).

Đồng thời với việc các ứng dụng IIoT ngày càng phát triển thì việc lưu dữ liệu IoT vào MongoDB tại server cũng rất quan trọng do đó tại chương 3 tác giả sẽ hướng dẫn cách kết nối và lưu trữ dữ liệu lên MongoDB từ các thiết bị chạy Winform c#.

**Chương 1: Hướng dẫn cài đặt MongoDB trên Windows.**

**Bước 1: Download MongoDB Community.**

**Link dowload:** [**https://www.mongodb.com/try/download/community**](https://www.mongodb.com/try/download/community)



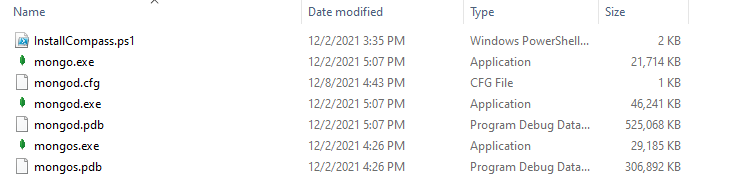
**Chọn cấu hình và nhấn Download.**

**Bước 2: Cài đặt MongoDB trên máy tính windows.**

Sau khi dowload xong thì tiến hành chạy file MSI và tiến hành cài đặt theo hướng dẫn của nhà cung cấp phần mềm.

Sau khi cài đặt xong tại địa chỉ dưới sẽ có các file sau.

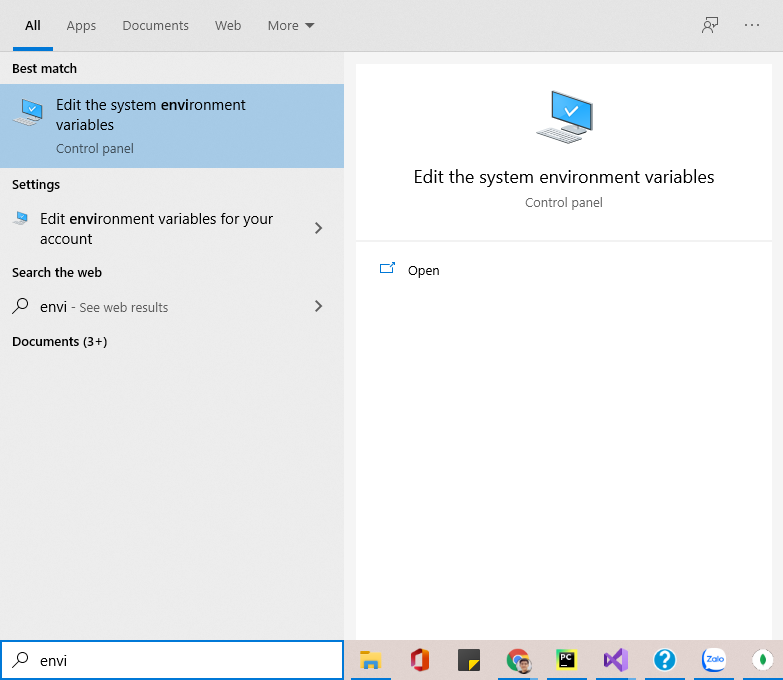
C:\Program Files\MongoDB\Server\5.0\bin



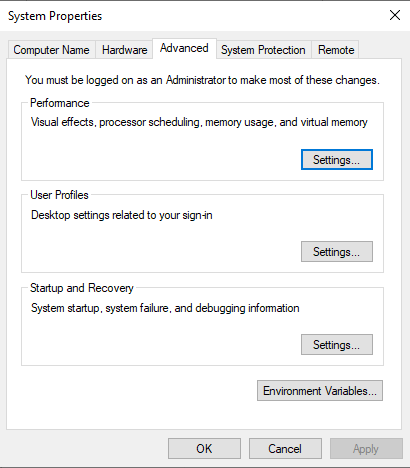
Sau khi cài đặt xong tại ổ C:/ tạo thư mục “data” trong thư mục data tạo thư mục “db”.

**Bước 3: Khởi chạy MongoDB bằng command Prompt.**

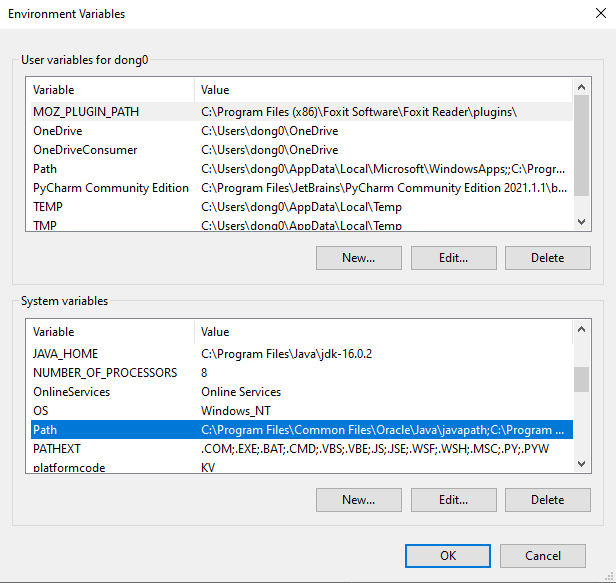
Thiết lập biến môi trường cho MongoDB

****

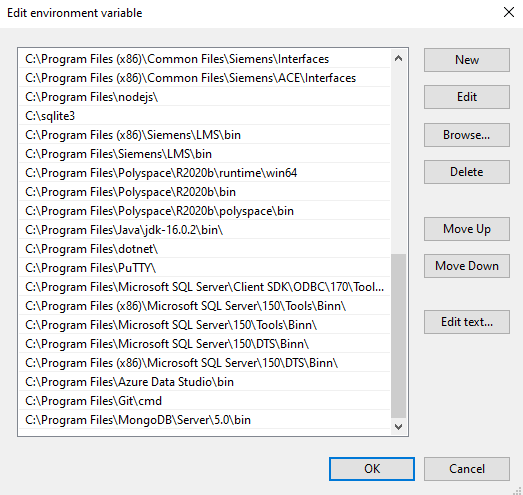
Tại Control Panel tìm đến Edit the system enviroment variables



Tại tab Advanced chọn Enviroment Varibales…



Tại tab System variables chọn Path và nhấn vào Edit..



Chọn new và thêm đường dẫn này: C:\Program Files\MongoDB\Server\5.0\bin

Vào và nhấn OK.

Bật command Prompt:

* Gõ mongod
* Gõ mongo

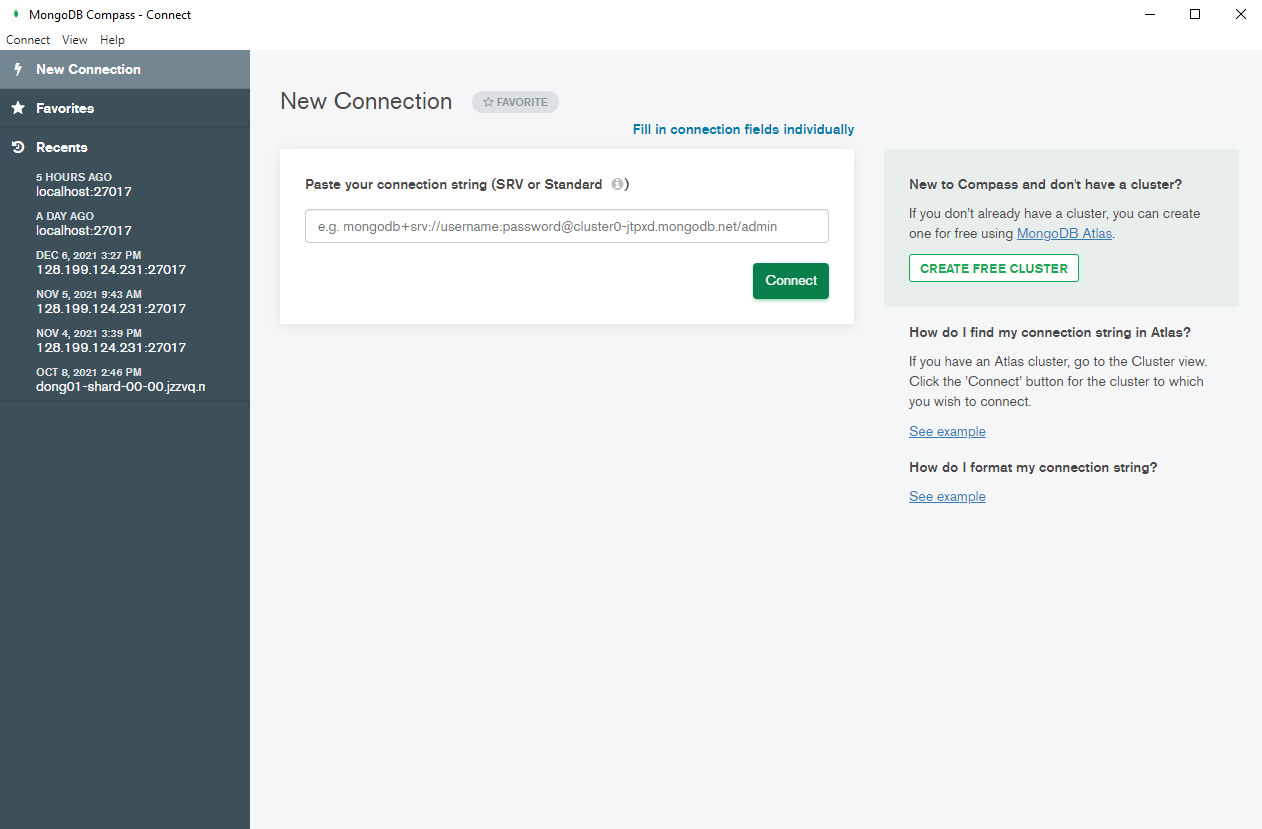
Để kiểm tra mongoDB trên máy

Làm theo turorial này: <https://www.youtube.com/watch?v=3wqzr-GJoS0&t=6s>

**Bước 4: Làm việc với MongoDB bằng MongoDB Compass.**

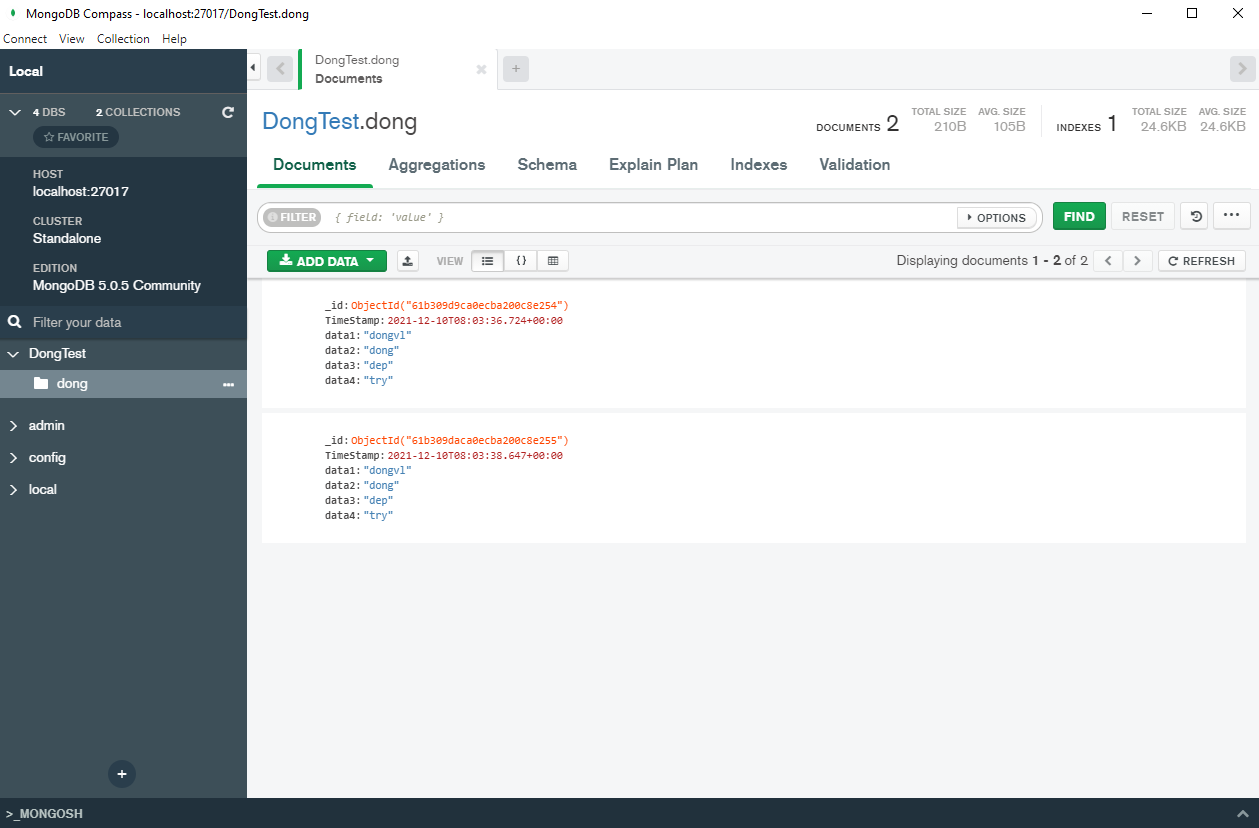
Sau khi cài đặt xong MongoDB nếu chưa có MongoDB Compass thì tải về theo link này:

<https://www.mongodb.com/try/download/compass>



Nhấn vào Connect để kết nối đến MongoDB chạy tại Local Host.

Tại đây có thể thêm mới Database và Collection mới.



Database và Collection được tạo bằng MongoDB Compass.

**Chương 2: Hướng dẫn sử dụng Winform C# để Insert, Delete và Query MongoDB.**

Trong ví dụ tác giả sử dụng thư viện MongoDB của .NET như sau:

using MongoDB.Bson;

using MongoDB.Driver;

Thư viện sau khi viết lại:

using System;

using System.ComponentModel;

using System.Linq;

using System.Windows.Forms;

using MongoDB.Bson;

using MongoDB.Driver;

namespace MongoTest

{

public class MongoDB\_Config

{

public string ConnectionString;

public string Database;

public string Collection;

//

public class Mydata

{

public object Id { get; set; }

public DateTime TimeStamp { get; set; }

public string data1 { get; set; }

public string data2 { get; set; }

public string data3 { get; set; }

public string data4 { get; set; }

}

public MongoDB\_Config(string connectionstring,string database,string collection)

{

ConnectionString = connectionstring;

Database = database;

Collection = collection;

}

//hien thi du lieu dang Bindinglist hien thi ra datagridview dataGridview.DataSource = return this.

public BindingList<Mydata> MongoQuery(DateTime fromdate, DateTime todate)

{

try

{

var client = new MongoClient(ConnectionString);

var database = client.GetDatabase(Database);

var collec = database.GetCollection<Mydata>(Collection);

BindingList<Mydata> doclist = new BindingList<Mydata>();

var min = fromdate;

var max = todate;

foreach (var deger in collec.Find(x => x.TimeStamp > min && x.TimeStamp < max).ToList())

{

doclist.Add(deger);

Application.DoEvents();

}

return doclist;

}

catch

{

Console.WriteLine("Error when query database");

return null;

}

}

//Insert du lieu sang MongoDB duoi dang Bson document.

public void MongoInsert(BsonDocument document)

{

try

{

var client = new MongoClient(ConnectionString);

var database = client.GetDatabase(Database);

var collec = database.GetCollection<BsonDocument>(Collection);

collec.InsertOneAsync(document);

}

catch

{

Console.WriteLine("Error when insert Bson doc to MongoDB");

}

}

//xoa du lieu theo ngay dươc dua vao

public void MongoDelete(DateTime fromtime,string TimeStampKey)

{

try

{

var client = new MongoClient(ConnectionString);

var database = client.GetDatabase(Database);

var collec = database.GetCollection<BsonDocument>(Collection);

var dataQuery = new BsonDocument

{

{TimeStampKey,new BsonDocument {

{"$lt",fromtime}

} }

};

var resule = collec.DeleteMany(dataQuery);

}

catch

{

Console.WriteLine("Error when remove Bson doc from MongoDB");

}

}

}

}

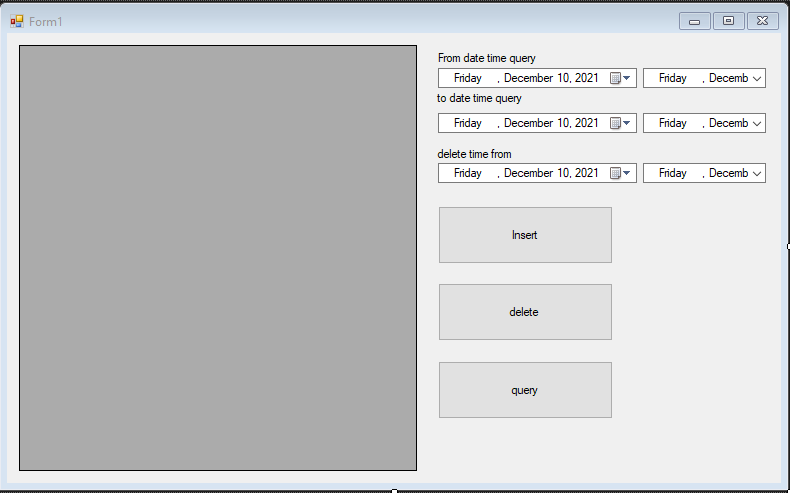
Cách sử dụng:

**Bước 1: Tạo mới class và thêm đoạn source code trên hoặc liên hệ tác giả để lấy source code mấu.**

namespace MongoTest

Chỉnh sửa lại namespace cho giống với tên của Project.

Thêm các thư viện còn thiếu bằng cách add refrence vào project.



Tạo form bào gồm 3 nút nhấn, 6 datetimePicker và 1 DataGridview như hình.

using System;

using System.Windows.Forms;

using MongoDB.Bson;

namespace MongoTest

{

public partial class Form1 : Form

{

//Khai báo thông tin của MongoDB bao gồm Connection string, database, collection

public static string ConnectionString = "mongodb://localhost:27017";

public static string Database = "DongTest";

public static string Collection = "dong";

public Form1()

{

InitializeComponent();

//Cấu hình cài đặt cho các datetimepicker để lấy dữ liệu thời gian

dtpk\_time\_delete.Format = DateTimePickerFormat.Time;

dtpk\_time\_delete.ShowUpDown = true;

dtpk\_time\_from.Format = DateTimePickerFormat.Time;

dtpk\_time\_from.ShowUpDown = true;

dtpk\_time\_to.Format = DateTimePickerFormat.Time;

dtpk\_time\_to.ShowUpDown = true;

}

MongoDB\_Config mongoDB = new MongoDB\_Config(ConnectionString, Database, Collection);

private void bt\_Insert\_Click(object sender, EventArgs e)

{

//Tạo BsonDocument để Insert vào MongoDB.

var doc = new BsonDocument

{

{"TimeStamp",DateTime.Now},

{"data1","dongvl" },

{"data2","dong"},

{"data3","dep"},

{"data4","try"}

};

mongoDB.MongoInsert(doc);

}

private void bt\_Delete\_Click(object sender, EventArgs e)

{

//Xóa dữ liệu MongoDB từ thời gian được chọn trở về trước

var datetime = dtpk\_date\_delete.Value.Date + dtpk\_time\_delete.Value.TimeOfDay;

mongoDB.MongoDelete(datetime,"TimeStamp");

}

private void bt\_Query\_Click(object sender, EventArgs e)

{

//Tìm kiếm và hiển thị dữ liệu từ MongoDB về DataGridview.

DateTime fromtime = dtpk\_date\_from.Value.Date + dtpk\_time\_from.Value.TimeOfDay;

DateTime totime = dtpk\_date\_to.Value.Date + dtpk\_time\_to.Value.TimeOfDay;

dataGridView1.DataSource = mongoDB.MongoQuery(fromtime, totime);

}

}

}

Trong đó lưu ý: Bson Document có thể đưa các tag của PLC vào, Class Mydata trong MongoDB\_Config phải được để theo giống định dạng của Bson Document.

var doc = new BsonDocument

{

{"TimeStamp",DateTime.Now},

{"data1","dongvl" },

{"data2","dong"},

{"data3","dep"},

{"data4","try"}

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class Mydata

{

public object Id { get; set; }

public DateTime TimeStamp { get; set; }

public string data1 { get; set; }

public string data2 { get; set; }

public string data3 { get; set; }

public string data4 { get; set; }

}

**Link code: https://github.com/dong050199/CSharp-Winform-MongoDB.git**

**Chường 3: Hướng dẫn dùng Node-Red kết nối winform và MongoDB thông qua giao thức MQTT.**

**Chưa làm :DDD**