An Implementation of Database on LNC Controller

Cyan[1], Astra[2], Kevin[3], Albert[4]

Prof. Meng-Shiun Tsai

ME 7007: Software and Hardware System Development under Industry 4.0

M.E., National Taiwan University

June 18, 2019

[1] R07522843
[2] R07522840
[3] R07522802
[4] R07522803

Contents

- Introduction
  - Primary Scenario
  - Hardware
    - LNC controller
  - Software
    - Socket
    - MongoDB
- Basic Scenario
  - Ethernet Connection
  - Data Extraction
  - Controller
  - Machining Process
  - Data Transmission
  - Primary Scenario Demo
- Application Scenario
  - G-code Payload R values
  - Dynamically Updating Plot
  - Find data
  - Application Scenario Demo
- Conclusion

Introduction

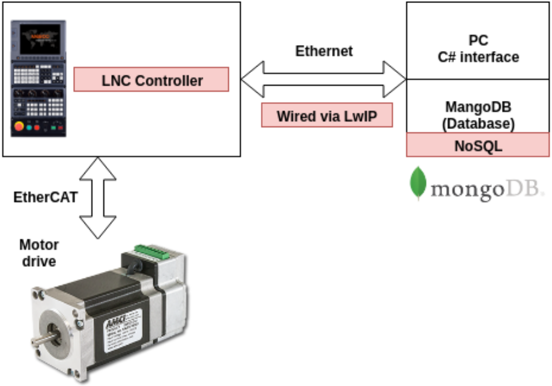


Figure 1: Framwork of experiment.

Primary Scenario

Outline of experiment environment

- Setting up GUI for controller in C# (provided by LNC Tech.)
- Connecting to controller via wired Ethernet.
- Uploading G-code for controller in C# via File Transfer Protocol.
- Starting up motor drive by running uploaded G-code.
- Receiving encoder data and sending to server (.py) via socket.
- Listening to any client to be connected.
- Creating MongoClient, connecting to server and receiving data simutaneously.
- Storing data into database and collection.

Hardware

LNC controller

- 10.4" TFT LCD.
- Control 9+6 axis.
- Support MII/RTEX/ EtherCAT communication protocol.
- High-speed, high- precision and wiring saving.
- Provides interpolations to satisfy the requirements of high level of turn-milling.
- Various intelligent functions.
- Particle-button and support USB drive.

Source: `https://www.lnc.com.tw/en-us/products/651310d3-07e8-4065-96e2-e9cee323a966/t6800d-m(vertical_pbo)/mod_4b83b157-94cd-45ad-a36e-0a85483f4009`

A mechanism for allowing communication between processes where running on same/different computers connected on a network.

socket.io

Widely used applications:

- Instant messaging and chat.
- Real-time analytics by pushing data to clients that get represented as real-time counters, charts or log.
- Documentation collaboration.

Software

MongoDB

- Easy to install and set up.
- A BSON (a JSON-like format) to store data.
- Easy to map the document objects to application code.
- Highly scalable and available, and includes support for out-of-the-box replication.
- Support MapReduce operations for condensing a large volume of data into useful aggregated results.
- Free and open source.

Software

MongoDB - NoSQL

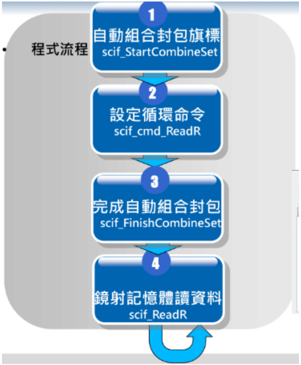
- Flexible data models (Schema Free)
    - NoSQL databases more relaxed in structure of data
- Clusters of cheap commodity servers to manage the data and transaction volumes
- NoSQL are still implementing their basic feature set

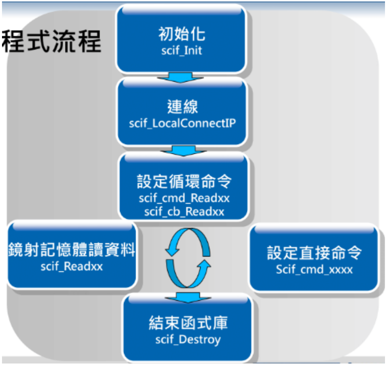
Why MongoDB?

- Simple queries
- Functionality provided applicable to most web applications
- Easy and fast integration of data
- Not well suited for heavy and complex transactions systems

Ethernet Connection

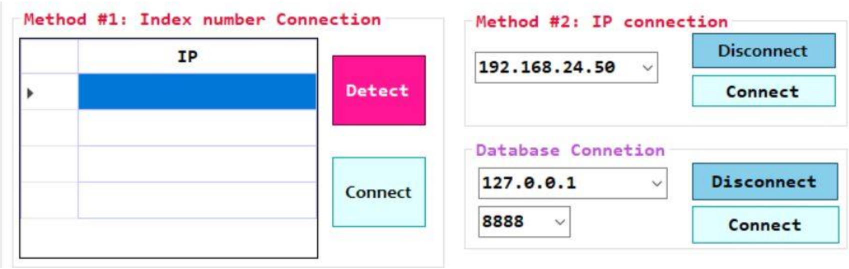
TCPIP connection between LNC controller to PC.





Ethernet Connection

TCPIP connection between LNC controller to PC.



Method #1: Index number Connection

	IP
▸	

Method #2: IP connection

192.168.24.50

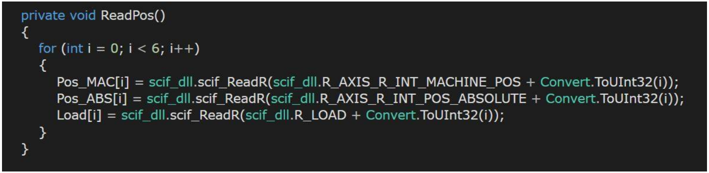
Database Connetion

127.0.0.1

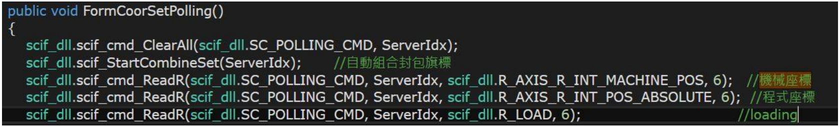
8888

Data Extraction

- Extract data (eg. coordinates, loading, error) stored in R values.
- Get polling of the mirror memory for important R values.



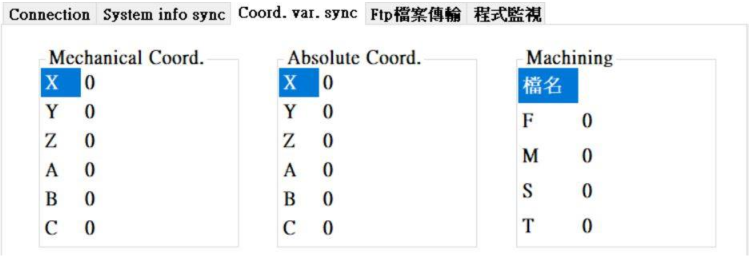
```
private void ReadPos()
{
    for (int i = 0; i < 6; i++)
    {
        Pos_MAC[i] = scif_dll.scif_ReadR(scif_dll.R_AXIS_R_INT_MACHINE_POS + Convert.ToUInt32(i));
        Pos_ABS[i] = scif_dll.scif_ReadR(scif_dll.R_AXIS_R_INT_POS_ABSOLUTE + Convert.ToUInt32(i));
        Load[i] = scif_dll.scif_ReadR(scif_dll.R_LOAD + Convert.ToUInt32(i));
    }
}
```



```
public void FormCoorSetPolling()
{
    scif_dll.scif_cmd_ClearAll(scif_dll.SC_POLLING_CMD, ServerIdx);
    scif_dll.scif_StartCombineSet(ServerIdx);      //自動組合封包旗標
    scif_dll.scif_cmd_ReadR(scif_dll.SC_POLLING_CMD, ServerIdx, scif_dll.R_AXIS_R_INT_MACHINE_POS, 6);   //機械座標
    scif_dll.scif_cmd_ReadR(scif_dll.SC_POLLING_CMD, ServerIdx, scif_dll.R_AXIS_R_INT_POS_ABSOLUTE, 6);  //程式座標
    scif_dll.scif_cmd_ReadR(scif_dll.SC_POLLING_CMD, ServerIdx, scif_dll.R_LOAD, 6);                     //loading
```

Data Extraction

- Extract data (eg. coordinates, loading, error) stored in R values.
- Get polling of the mirror memory for important R values.



Mechanical Coord.

X	0
Y	0
Z	0
A	0
B	0
C	0

Absolute Coord.

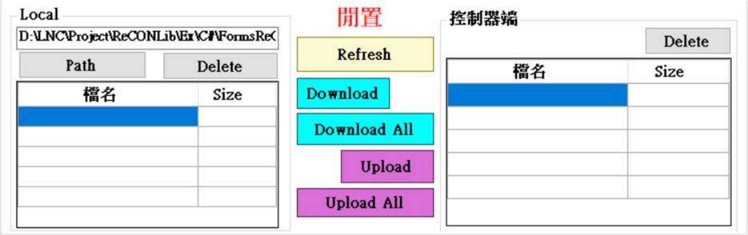
X	0
Y	0
Z	0
A	0
B	0
C	0

Machining

檔名	
F	0
M	0
S	0
T	0

Controller

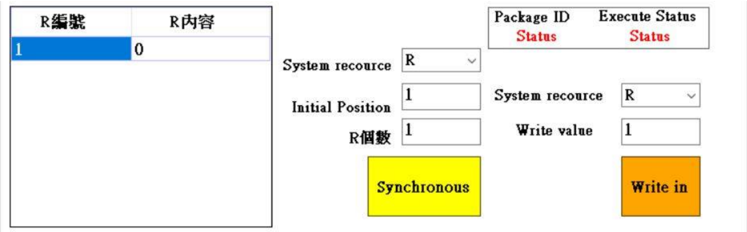
- Using G-code to control the hardware
- Using Ftp file transmission to upload G-code



D:\LNC\Project\ReCONLib\Ex\C#\FormsRe(

Machining Process

- Similar to typical factory application.
- G-code allows users to select machining process.
- Two R Values for G-code
  - $\Phi R290100$: Return to **0** when machining ends, set to **1** to start working
  - $\Phi R290101$: Set **0** to go straight path to desired points; **1** to go curve path; **2** to stop machining



R编號	R内容
1	0

System recource: R

Initial Position: 1

R個數: 1

Package ID: Status

Execute Status: Status

System recource: R

Write value: 1

Data Transmission

- LNC controller API function to Read R register value from Memory.
- **Socket** client(TCP) is created in same C# program.
- Socket client connects to local host (127.0.0.1).
- Waiting for server's acceptance/response.
- Sending Position and Payload data to socket server every second counts.

Data Transmission

- Create a python socket server and **listen** to any connection.
- Socket server **connect**s to local host (127.0.0.1).
- Receive (**recv**) data from client.
- **PyMongoClient** gets data from server also on local host.
- Database is created simutaneously.

Demo

Primary Scenario Demo

VIDEO

Application idea

Initially, we use R value of **Payload** as our main focus of optimization control. Therefore, the acceleration/deceleration of escalators reminds us of one of the key of the motion is depending on the magnitude of payload it endures.

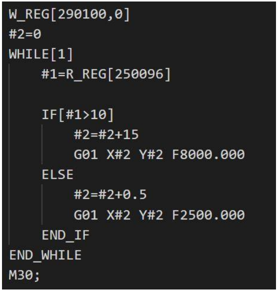
As to realize this scenario, we've extended the basic scenario futhermore.

Application Scenario

- Setting up GUI for controller in C# (provided by LNC Tech.)
- Connecting to controller via wired Ethernet.
- Uploading G-code for controller in C# via File Transfer Protocol.
  - G-code reads loading R values and controls rotating speed.
- Starting up motor drive by running uploaded G-code.
- Receiving encoder (& payload) data and sending to server (.py) via socket.
- Listening to any client to be connected.
- Creating MongoClient, connecting to server and receiving data simutaneously.
  - Dynamically updating plot in matplotlib.
- Storing data into database and collection.
  - Find data from collection w/ or w/o query.

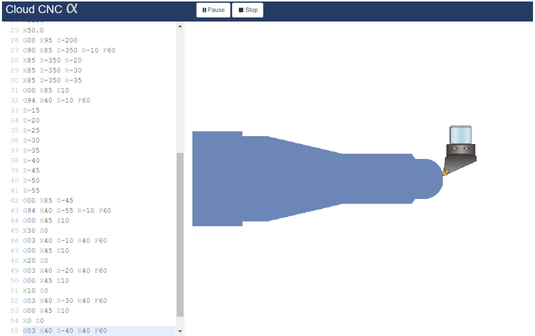
G-code Payload R values

G-code simulation.



```
W_REG[290100,0]
#2=0
WHILE[1]
    #1=R_REG[250096]

    IF[#1>10]
        #2=#2+15
        G01 X#2 Y#2 F8000.000
    ELSE
        #2=#2+0.5
        G01 X#2 Y#2 F2500.000
    END_IF
END_WHILE
M30;
```



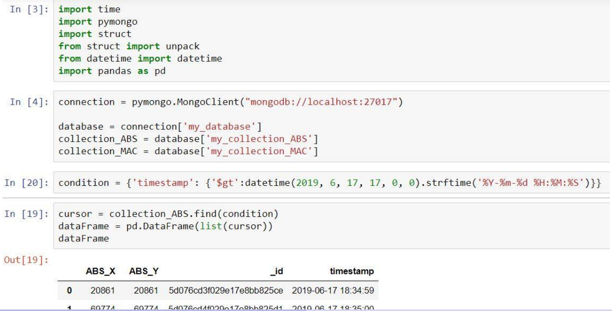
```
X50.0
G00 X95 Z-200
G90 X85 Z-350 R-10 F60
X85 Z-350 R-20
X85 Z-350 R-30
X85 Z-350 R-35
G00 X85 Z10
G94 X40 Z-10 F60
Z-15
Z-20
Z-25
Z-30
Z-35
Z-40
Z-45
Z-50
Z-55
G00 X85 Z-45
G94 X40 Z-55 R-10 F60
G00 X45 Z10
X30 Z0
G03 X40 Z-10 R40 F60
G00 X45 Z10
X20 Z0
G03 X40 Z-20 R40 F60
G00 X45 Z10
X10 Z0
G03 X40 Z-30 R40 F60
G00 X45 Z10
X0 Z0
G03 X40 Z-40 R40 F60
```

Dynamically Updating Plot

Use **Matplotlib**, update figure and axis range once receiving data from database dynamically.

Find Data

From another point of view, once the database is built, other clients can access to the database and look for data. For our case, we hope that user can find data by giving some information of time, and investigate the process situation during that time interval.



In [3]:
```
import time
import pymongo
import struct
from struct import unpack
from datetime import datetime
import pandas as pd
```

In [4]:
```
connection = pymongo.MongoClient("mongodb://localhost:27017")

database = connection['my_database']
collection_ABS = database['my_collection_ABS']
collection_MAC = database['my_collection_MAC']
```

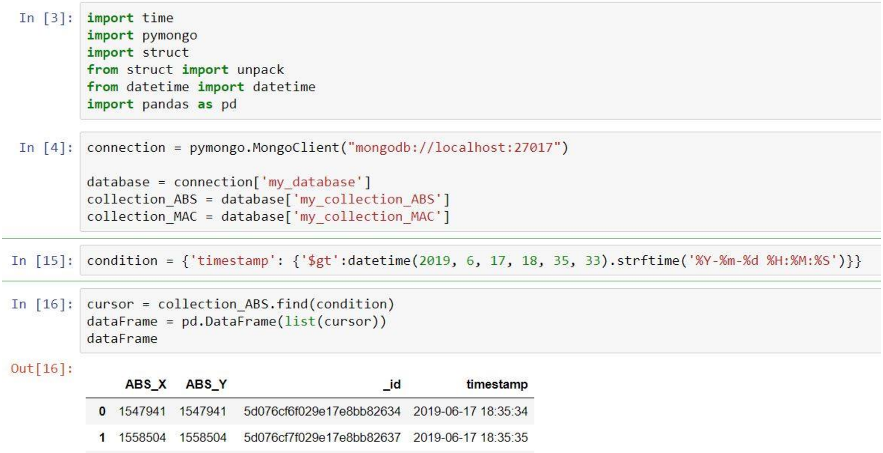
In [20]:
```
condition = {'timestamp': {'$gt':datetime(2019, 6, 17, 17, 0, 0).strftime('%Y-%m-%d %H:%M:%S')}}
```

In [19]:
```
cursor = collection_ABS.find(condition)
dataFrame = pd.DataFrame(list(cursor))
dataFrame
```

Out[19]:

	ABS_X	ABS_Y	_id	timestamp
0	20861	20861	5d076cd3f029e17e8bb825ce	2019-06-17 18:34:59
1	69774	69774	5d076cd4f029e17e8bb825d1	2019-06-17 18:35:00

Find Data



In [3]:
```
import time
import pymongo
import struct
from struct import unpack
from datetime import datetime
import pandas as pd
```

In [4]:
```
connection = pymongo.MongoClient("mongodb://localhost:27017")

database = connection['my_database']
collection_ABS = database['my_collection_ABS']
collection_MAC = database['my_collection_MAC']
```

In [15]:
```
condition = {'timestamp': {'$gt':datetime(2019, 6, 17, 18, 35, 33).strftime('%Y-%m-%d %H:%M:%S')}}
```

In [16]:
```
cursor = collection_ABS.find(condition)
dataFrame = pd.DataFrame(list(cursor))
dataFrame
```

Out[16]:

	ABS_X	ABS_Y	_id	timestamp
0	1547941	1547941	5d076cf6f029e17e8bb82634	2019-06-17 18:35:34
1	1558504	1558504	5d076cf7f029e17e8bb82637	2019-06-17 18:35:35

Demo

Application Scenario Demo

VIDEO ▶

Discussions

- We use python for database operation and C# for Controller data synchronization. How to implement Inter Program Communication(IPC)?
  - By Socket TCP, we send dataFrame from C# interface to python server and bind local Host as IP address.
- Simulating the process[5] can help users more easily monitor the entire machining process.
- Socket sends data stream in Bytes array. Thus, we take advantages of Concept of Union (The same memory address) and transform **Int** to **Bytes** array (1 Int = 4 Bytes in C#). While receiver decodes Bytes array back into Int.

[5] https://cnc-lathe-simulator.appspot.com/

Conclusion

Through C# GUI to give commands to controller, and store data into the database at the same time is achievable. Data visualization and its noSQL structure can better improve data storage and management. Those applications can greatly optimize control in different scenarios.

Thank you! Any Question?