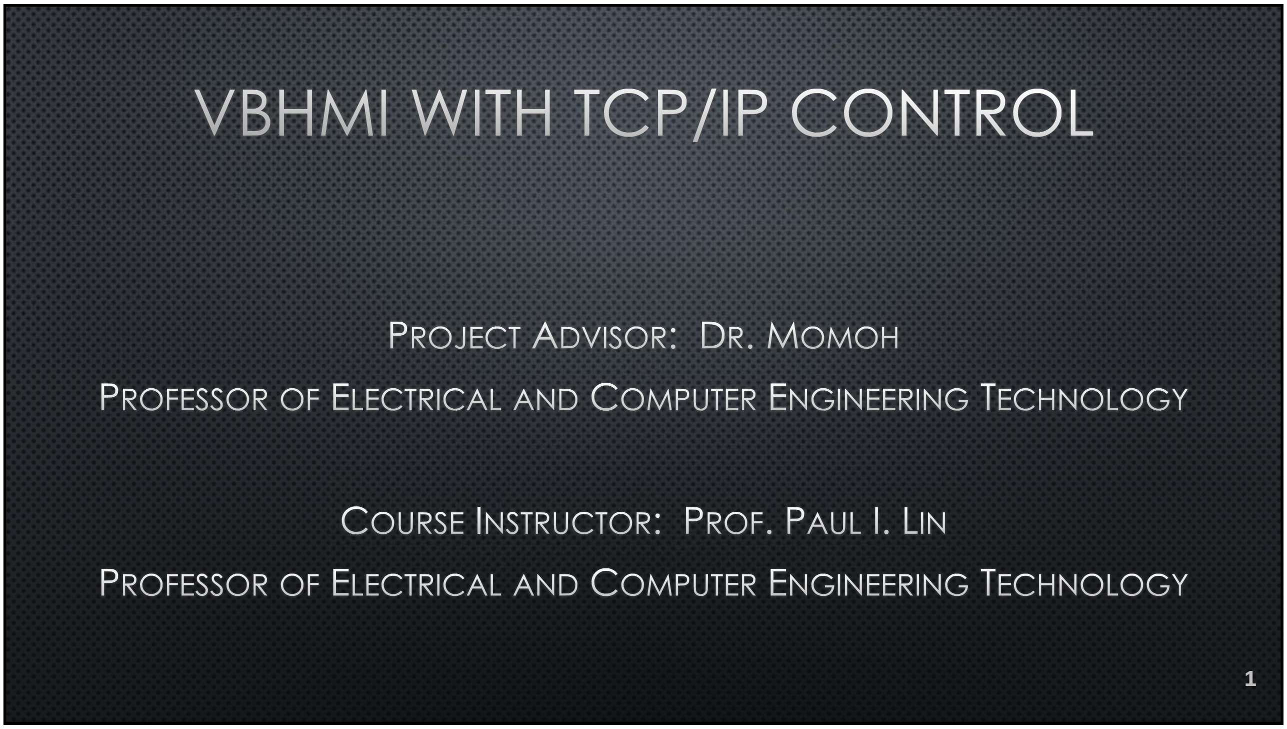


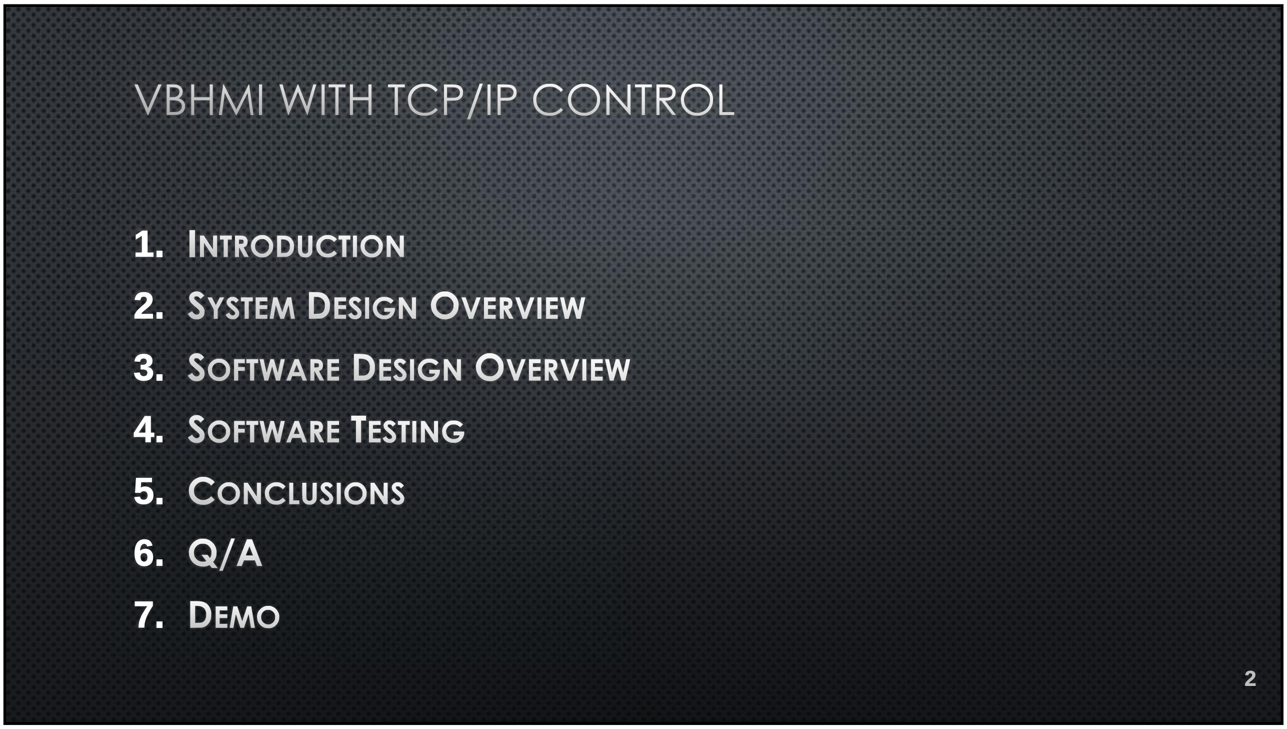
VBHMI WITH TCP/IP CONTROL

Project Advisor: Dr. Momoh

Professor of Electrical and Computer Engineering Technology

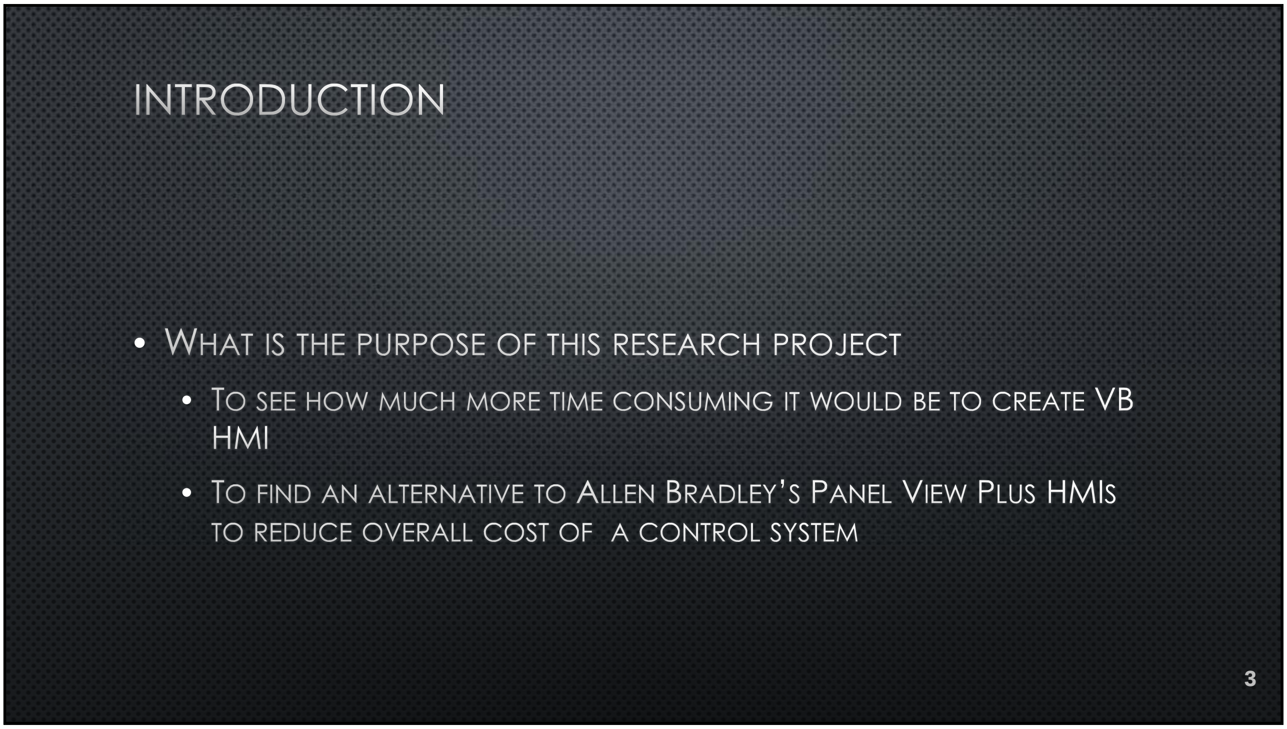
Course Instructor: Prof. Paul I. Lin

Professor of Electrical and Computer Engineering Technology



VBHMI WITH TCP/IP CONTROL

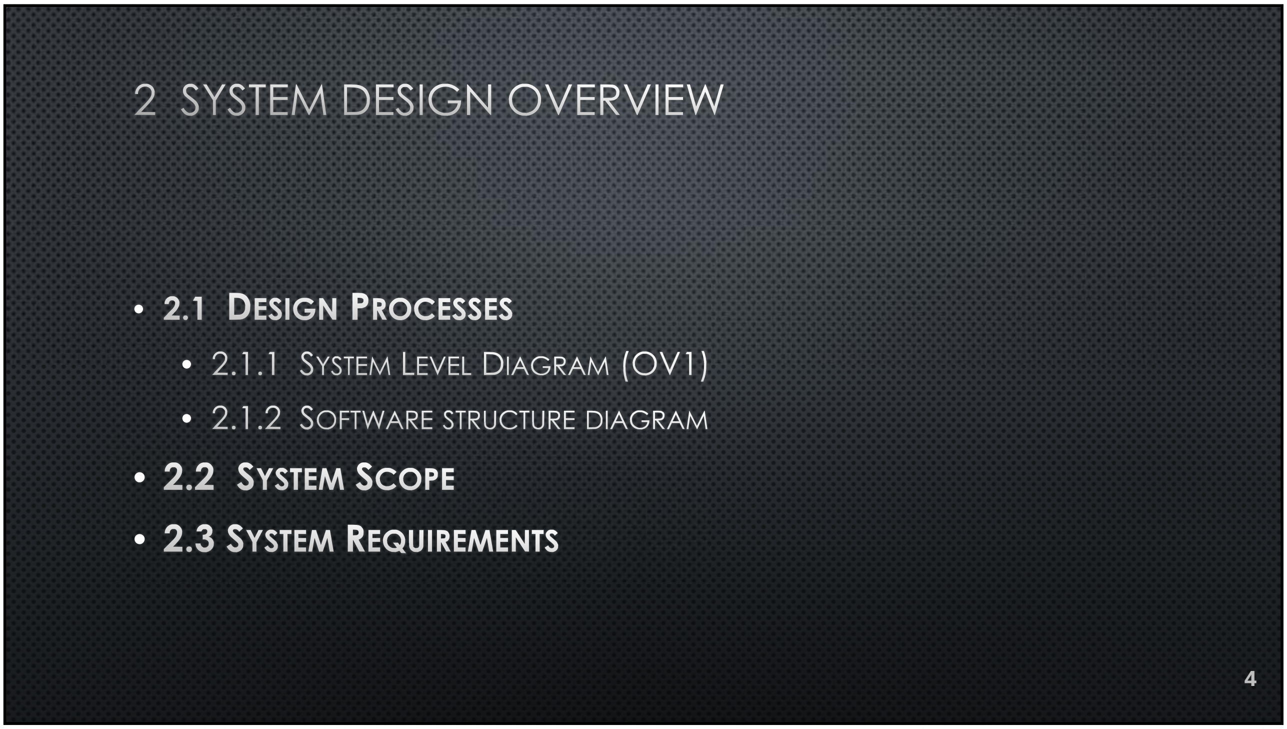
1. Introduction
2. System Design Overview
3. Software Design Overview
4. Software Testing
5. Conclusions
6. Q/A
7. Demo



INTRODUCTION

- What is the purpose of this research project
  - To see how much more time consuming it would be to create VB HMI
  - To find an alternative to Allen Bradley's Panel View Plus HMIs to reduce overall cost of a control system

3

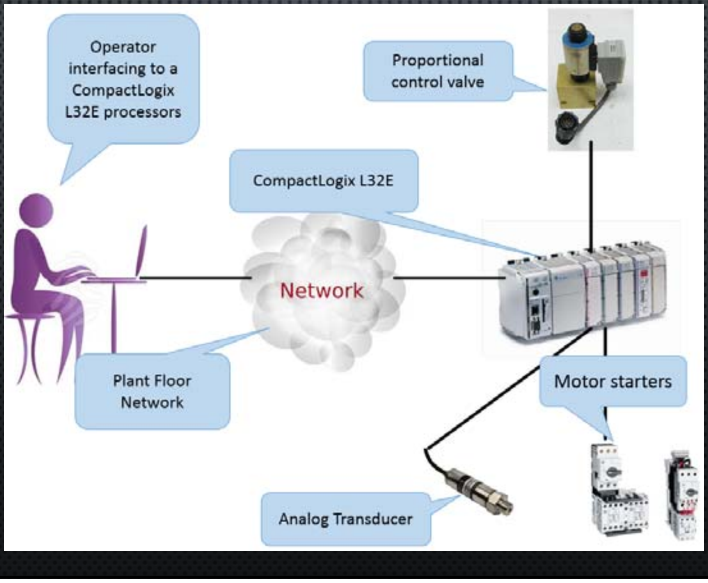


2 SYSTEM DESIGN OVERVIEW

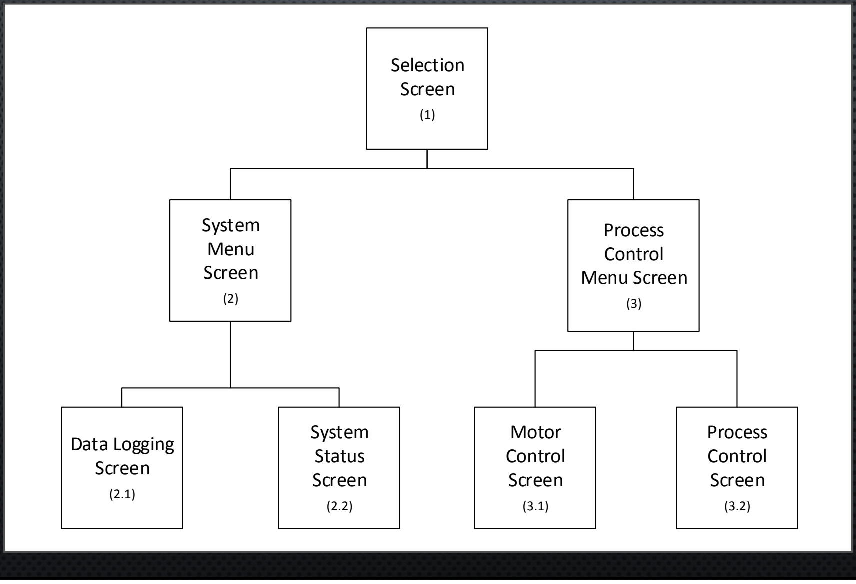
- **2.1 Design Processes**
  - 2.1.1 System Level Diagram (OV1)
  - 2.1.2 Software structure diagram
- **2.2 System Scope**
- **2.3 System Requirements**

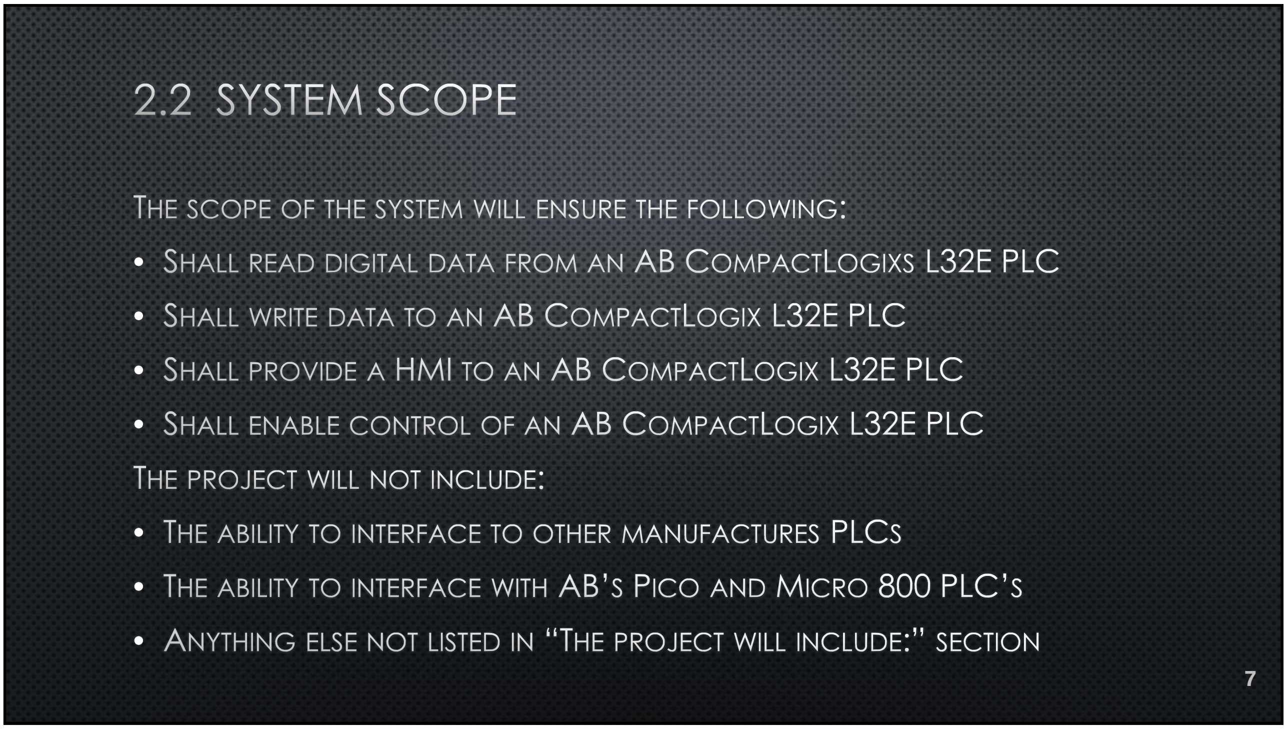
4

2.1.1 DESIGN PROCESSES


Operator interfacing to a CompactLogix L32E processors
Proportional control valve
CompactLogix L32E
Network
Plant Floor Network
Motor starters
Analog Transducer

2.1.2 DESIGN PROCESSES


Selection Screen (1)
System Menu Screen (2)
Process Control Menu Screen (3)
Data Logging Screen (2.1)
System Status Screen (2.2)
Motor Control Screen (3.1)
Process Control Screen (3.2)



2.2 SYSTEM SCOPE

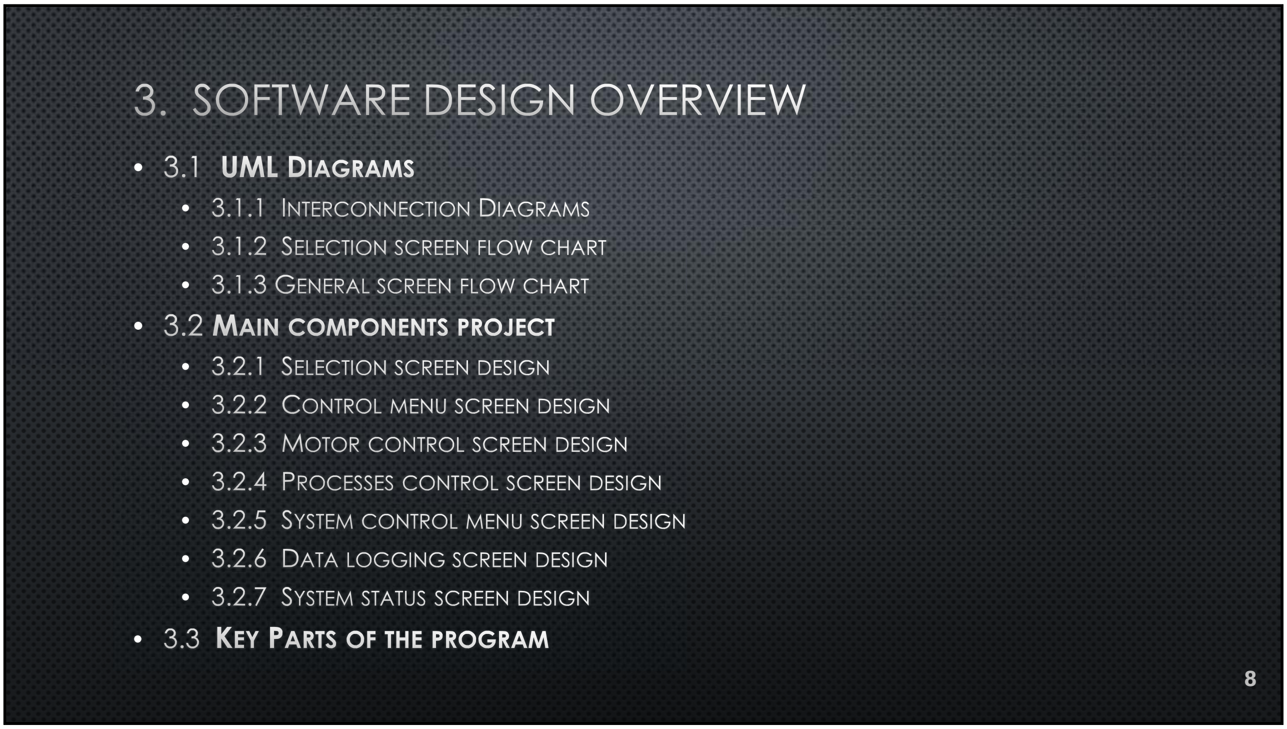
The scope of the system will ensure the following:

- Shall read digital data from an AB CompactLogixs L32E PLC
- Shall write data to an AB CompactLogix L32E PLC
- Shall provide a HMI to an AB CompactLogix L32E PLC
- Shall enable control of an AB CompactLogix L32E PLC

The project will not include:

- The ability to interface to other manufactures PLCs
- The ability to interface with AB's Pico and Micro 800 PLC's
- Anything else not listed in "The project will include:" section

7

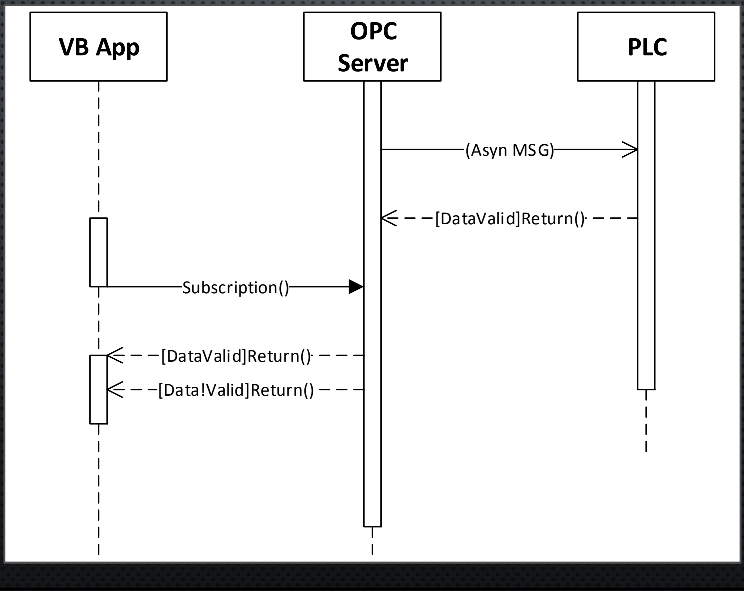


3. SOFTWARE DESIGN OVERVIEW

- 3.1 **UML Diagrams**
  - 3.1.1 Interconnection Diagrams
  - 3.1.2 Selection screen flow chart
  - 3.1.3 General screen flow chart
- 3.2 **Main components project**
  - 3.2.1 Selection screen design
  - 3.2.2 Control menu screen design
  - 3.2.3 Motor control screen design
  - 3.2.4 Processes control screen design
  - 3.2.5 System control menu screen design
  - 3.2.6 Data logging screen design
  - 3.2.7 System status screen design
- 3.3 **Key Parts of the program**

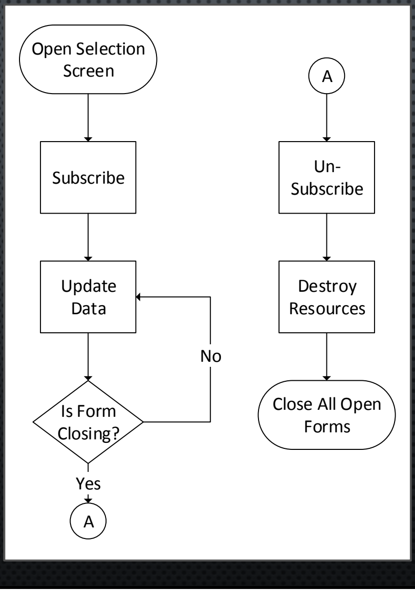
8

3.1.1 INTERCONNECTION DIAGRAM



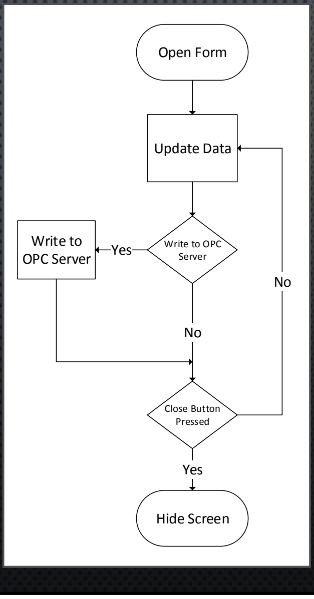
9

3.1.2 SELECTION SCREEN FLOW CHART



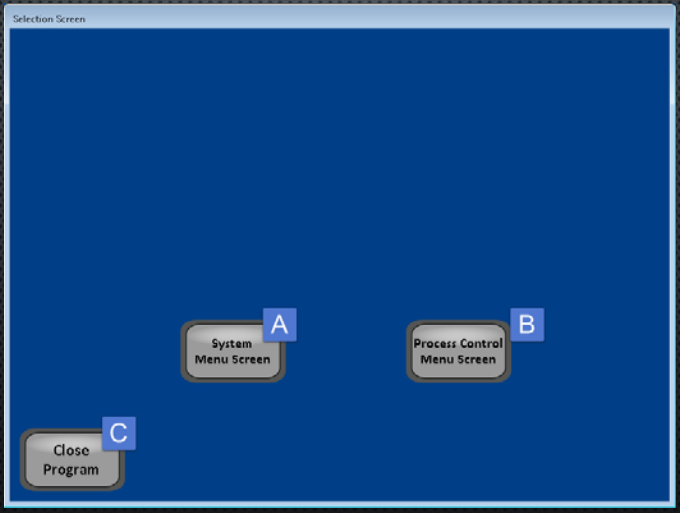
10

3.1.3 GENERAL SCREEN FLOW CHART



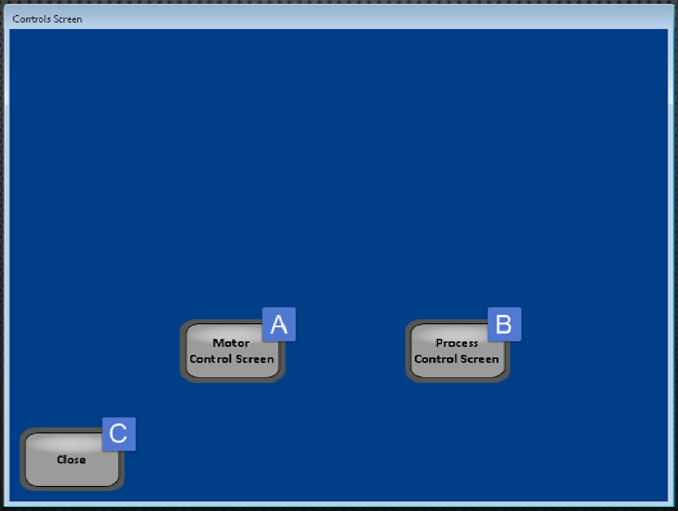
11

3.2.1 SELECTION SCREEN DESIGN

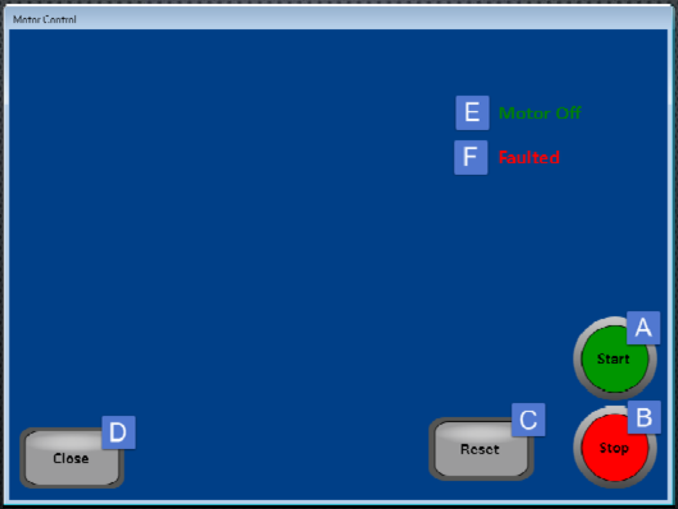


12

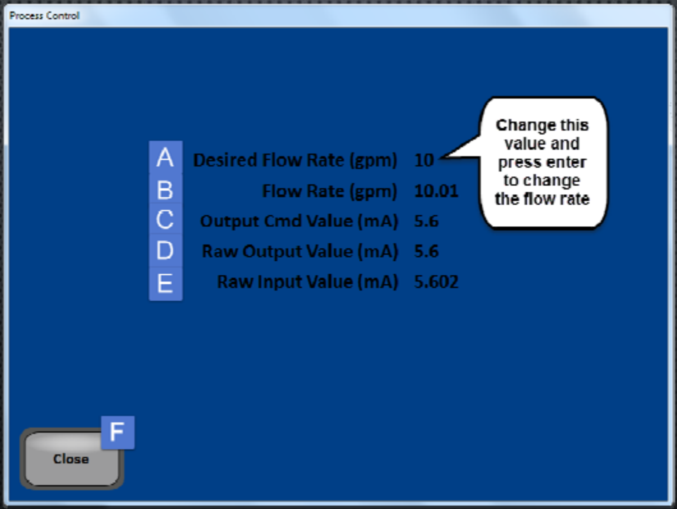
3.2.2 CONTROL MENU SCREEN DESIGN



3.2.3 MOTOR CONTROL SCREEN DESIGN



3.2.4 PROCESS CONTROL SCREEN DIAGRAM

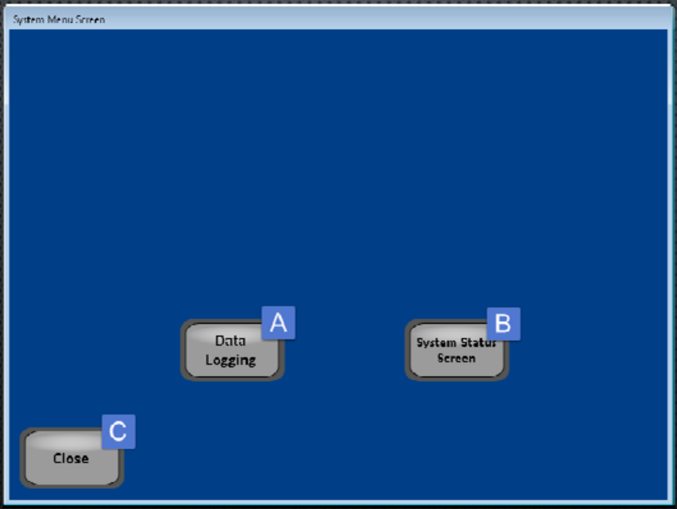


A	Desired Flow Rate (gpm)	10
B	Flow Rate (gpm)	10.01
C	Output Cmd Value (mA)	5.6
D	Raw Output Value (mA)	5.6
E	Raw Input Value (mA)	5.602

Change this value and press enter to change the flow rate

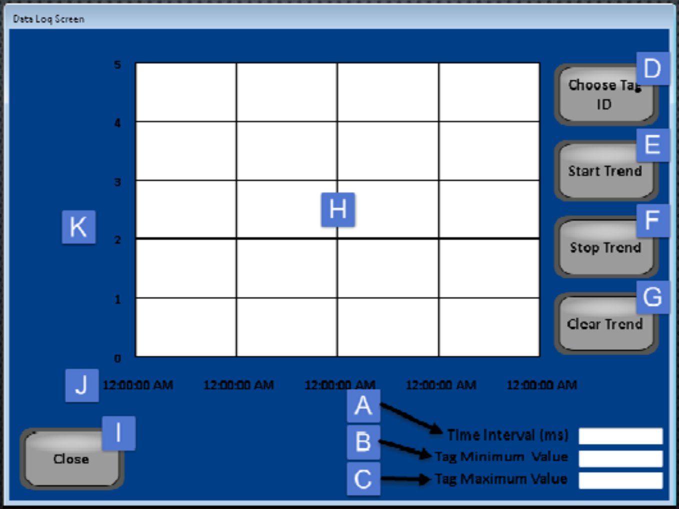
15

3.2.5 SYSTEM CONTROL MENU SCREEN DESIGN



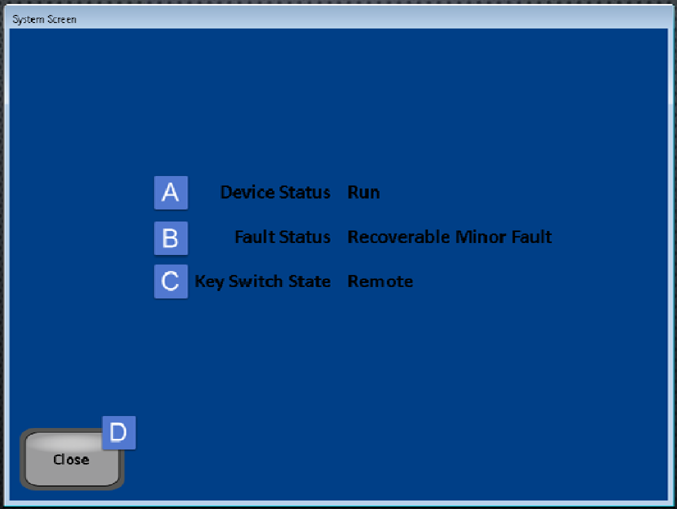
16

3.2.6 DATA LOGGING SCREEN DESIGN



17

3.2.7 SYSTEM STATUS SCREEN DESIGN



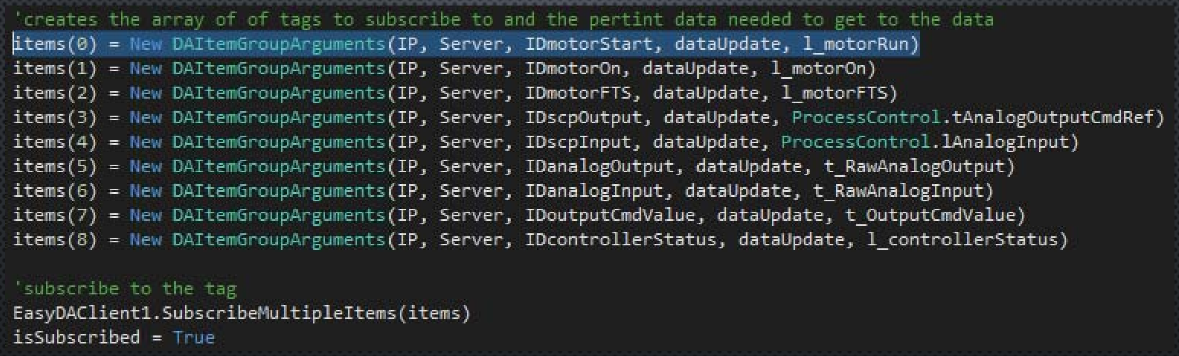
Device Status Run

Fault Status Recoverable Minor Fault

Key Switch State Remote

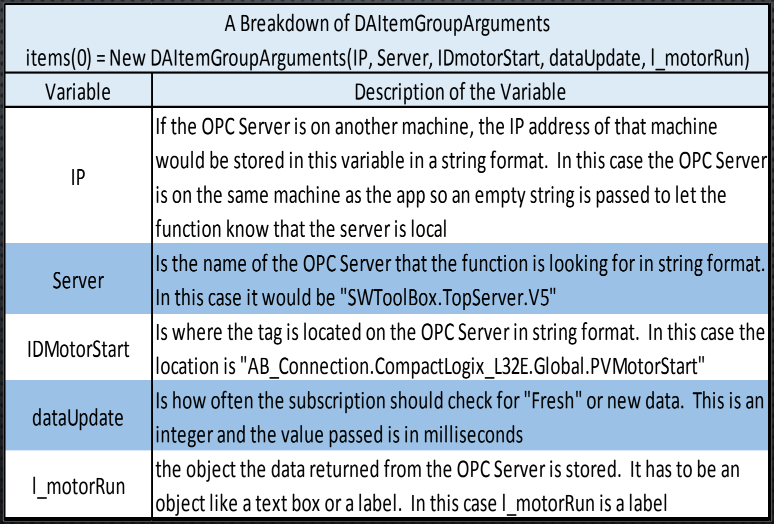
18

3.3 KEY PARTS OF THE PROGRAM



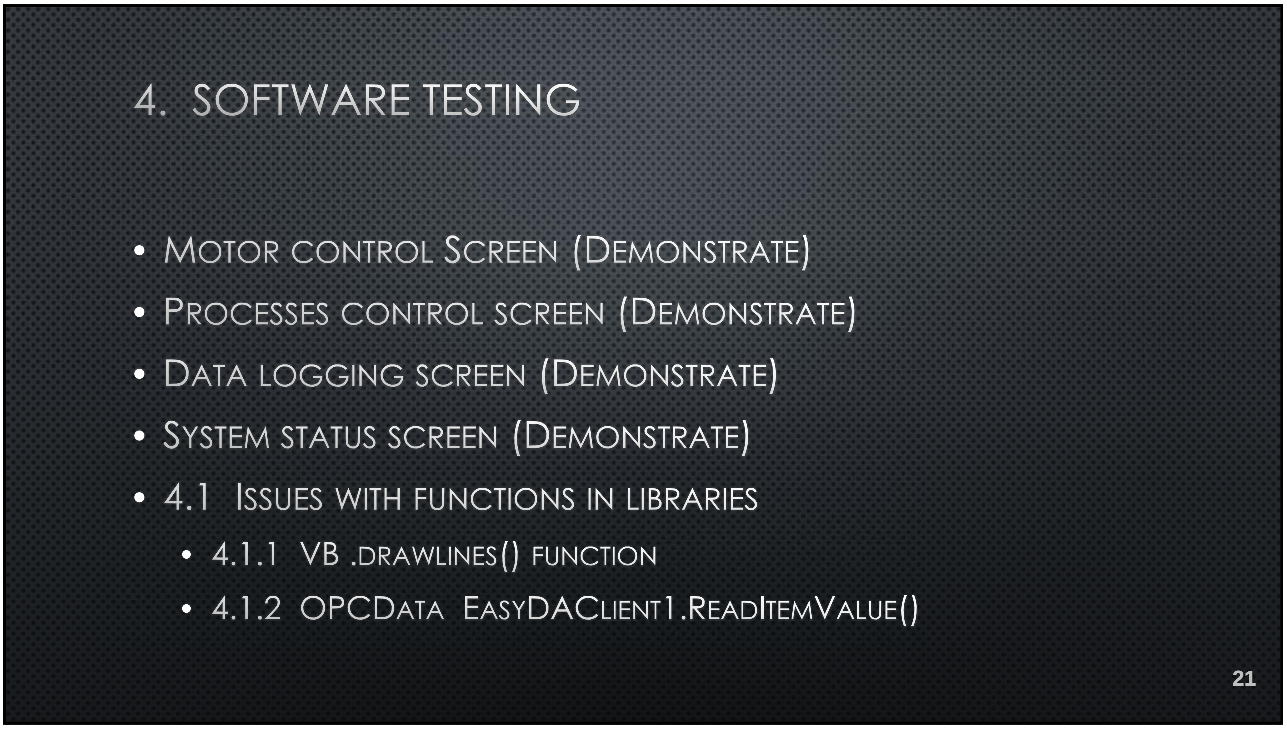
```
'creates the array of of tags to subscribe to and the pertint data needed to get to the data
items(0) = New DAItemGroupArguments(IP, Server, IDmotorStart, dataUpdate, l_motorRun)
items(1) = New DAItemGroupArguments(IP, Server, IDmotorOn, dataUpdate, l_motorOn)
items(2) = New DAItemGroupArguments(IP, Server, IDmotorFTS, dataUpdate, l_motorFTS)
items(3) = New DAItemGroupArguments(IP, Server, IDscpOutput, dataUpdate, ProcessControl.tAnalogOutputCmdRef)
items(4) = New DAItemGroupArguments(IP, Server, IDscpInput, dataUpdate, ProcessControl.lAnalogInput)
items(5) = New DAItemGroupArguments(IP, Server, IDanalogOutput, dataUpdate, t_RawAnalogOutput)
items(6) = New DAItemGroupArguments(IP, Server, IDanalogInput, dataUpdate, t_RawAnalogInput)
items(7) = New DAItemGroupArguments(IP, Server, IDoutputCmdValue, dataUpdate, t_OutputCmdValue)
items(8) = New DAItemGroupArguments(IP, Server, IDcontrollerStatus, dataUpdate, l_controllerStatus)

'subscribe to the tag
EasyDAClient1.SubscribeMultipleItems(items)
isSubscribed = True
```

ITEM(0)=DAITEMGROUPARGUMENTS(IP, SERVER, IDMOTORSTART, DATAUPDATE, L_MOTORRUN)



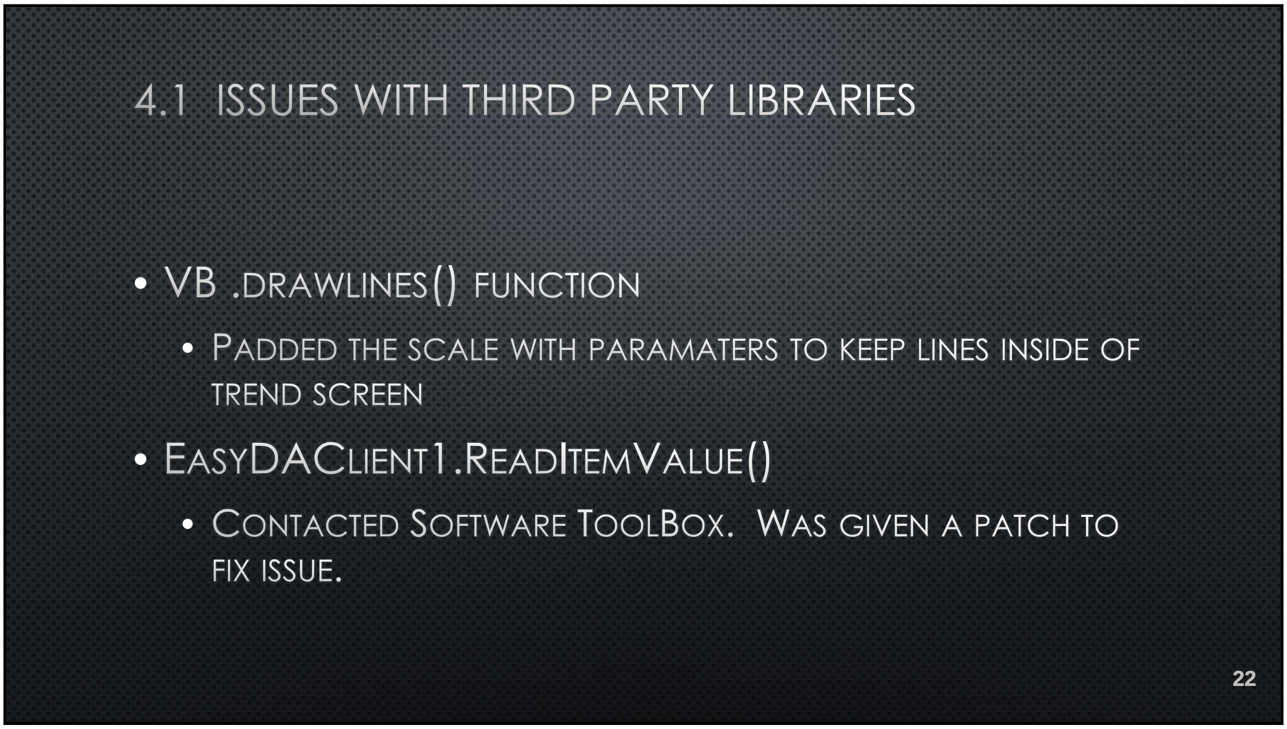
A Breakdown of DAItemGroupArguments
items(0) = New DAItemGroupArguments(IP, Server, IDmotorStart, dataUpdate, l_motorRun)

Variable	Description of the Variable
IP	If the OPC Server is on another machine, the IP address of that machine would be stored in this variable in a string format. In this case the OPC Server is on the same machine as the app so an empty string is passed to let the function know that the server is local
Server	Is the name of the OPC Server that the function is looking for in string format. In this case it would be "SWToolBox.TopServer.V5"
IDMotorStart	Is where the tag is located on the OPC Server in string format. In this case the location is "AB_Connection.CompactLogix_L32E.Global.PVMotorStart"
dataUpdate	Is how often the subscription should check for "Fresh" or new data. This is an integer and the value passed is in milliseconds
l_motorRun	the object the data returned from the OPC Server is stored. It has to be an object like a text box or a label. In this case l_motorRun is a label



4. SOFTWARE TESTING

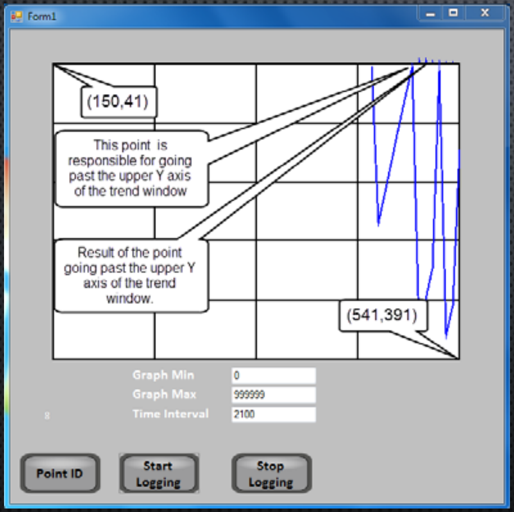
- Motor control Screen (Demonstrate)
- Processes control screen (Demonstrate)
- Data logging screen (Demonstrate)
- System status screen (Demonstrate)
- 4.1 Issues with functions in libraries
  - 4.1.1 VB .drawlines() function
  - 4.1.2 OPCData EasyDAClient1.ReadItemValue()



4.1 ISSUES WITH THIRD PARTY LIBRARIES

- VB .drawlines() function
  - Padded the scale with paramaters to keep lines inside of trend screen
- EasyDAClient1.ReadItemValue()
  - Contacted Software ToolBox. Was given a patch to fix issue.

4.1.1 VB .DRAWLINES() FUNCTION



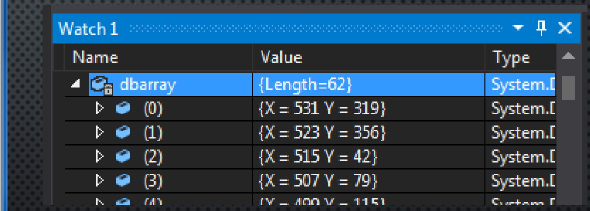
(150,41)

This point is responsible for going past the upper Y axis of the trend window

Result of the point going past the upper Y axis of the trend window.

(541,391)

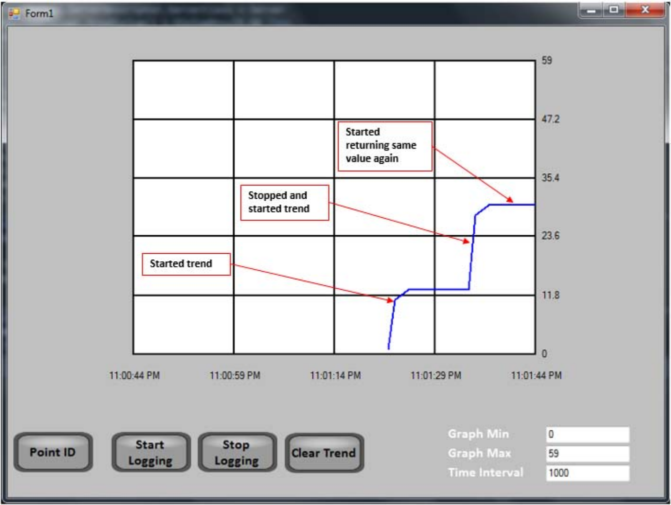
Graph Min	0
Graph Max	999999
Time Interval	2100



Watch 1

Name	Value	Type
dbarray	{Length=62}	System.D
(0)	{X = 531 Y = 319}	System.D
(1)	{X = 523 Y = 356}	System.D
(2)	{X = 515 Y = 42}	System.D
(3)	{X = 507 Y = 79}	System.D
(4)	{X = 499 Y = 115}	System.D

4.1.2 OPCDATA EASYDACLIENT1.READITEMVALUE()

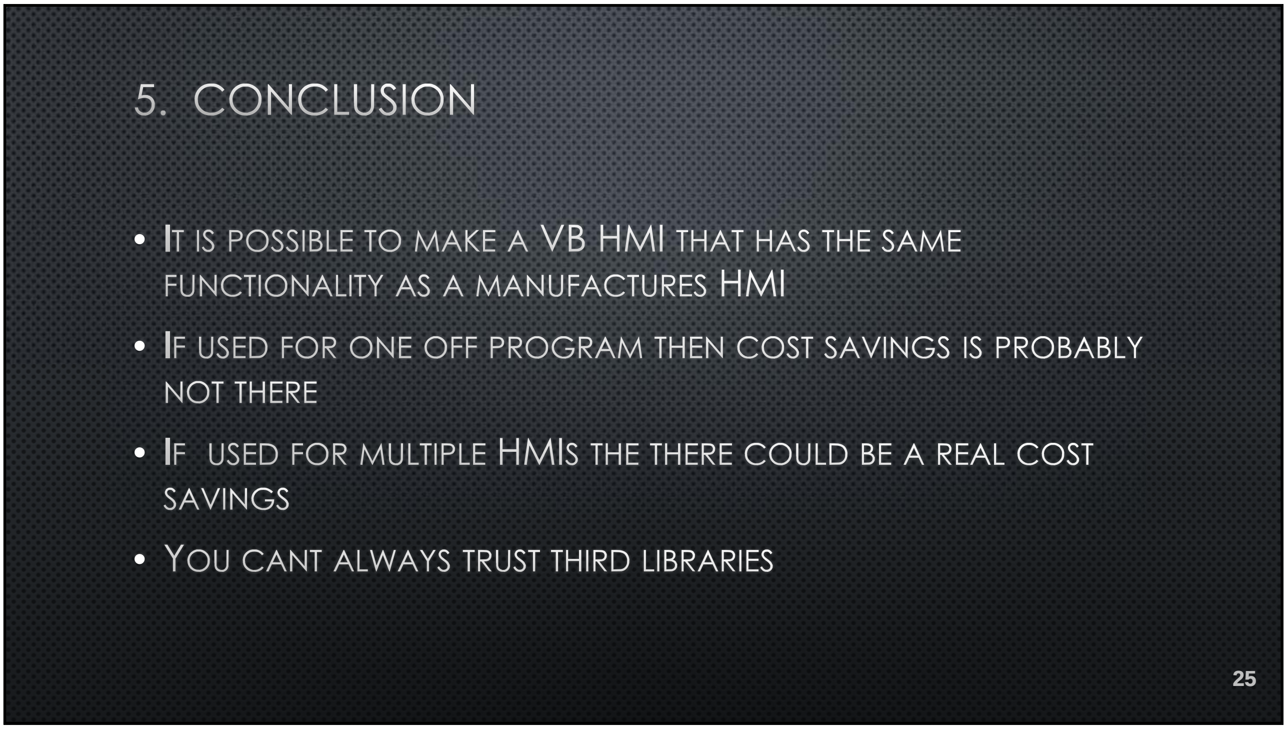


Started returning same value again

Stopped and started trend

Started trend

Graph Min	0
Graph Max	59
Time Interval	1000



5. Conclusion

- It is possible to make a VB HMI that has the same functionality as a manufactures HMI
- If used for one off program then cost savings is probably not there
- If used for multiple HMIs the there could be a real cost savings
- You cant always trust third libraries

25

Q/A

26

DEMO