



Development & Implementation of Client – Server Architecture for PLC Based Automation

Jagadeesh R¹ & Dr. R. Ranganatha²

¹M. Tech. Student, Dept. of Mechanical Engg., SJC Institute of Technology, (India)

²Professor, Dept. of Mechanical Engg., SJC Institute of Technology, (India)

ABSTRACT

Since its invention, Programmable Logical Controller (PLC) has been the industrial standard for control automated applications. For decades, it has been successful mainly due to its programmability and ability to work efficiently in harsh industrial environments. Today, PLCs have adopted an array of open standards to provide engineers a great deal of flexibility to create customized solutions.

The objective of this work was to develop a client – server architecture that will enable high level programming and networking of PLCs, integration of automation with Internet and Web technology, and incorporation of mobile devices as Human Machine Interfaces (HMI).

LabVIEW was used as the programming tool to develop the client – server architecture. The architecture was built using technologies, such as, DataSocket and Network Variables, Web Service, Remote Panels and Data Dashboard. They enabled the programming of PLC in LabVIEW and creation of virtual HMI to control and monitor PLC application through the Internet and World Wide Web.

For the demonstration of the architecture, a miniature pneumatic bending machine was fabricated. The client – server architecture was successfully deployed to yield outcomes, such as, use of mobile devices for wireless control and monitoring of processes, remote programming of PLC, real time data collection and online reporting.

Keywords – Client-Server model, industrial automation, LabVIEW, pneumatics, programmable logical controller (PLC)

I. INTRODUCTION

Automation can be defined as the synergy of technologies for the creation of a process that repeatedly runs with minimal or no human aid. Automation has been the key driving force behind mass manufacturing. Benefits of automation include increase in productivity, improvement in efficiency, reduction of manual labor, decrease in lead time and lower production cost.

At the center of every automated system is a controller. Programmable Logical Controller (PLC) has dominated this space since its invention in the late 1960s. A PLC is a real time computer optimized to handle automated operations in an industrial environment. The chief advantage that the PLC provides is programmability and stability. PLC allows creation of wide range of industrial applications with a simple change or modification the program in its memory. In the last decade, rapid development of microelectronics, computing devices and



programming software have greatly improved the capabilities of a PLC. In addition, they have also led to the emergence of Personal Computer (PC) based controllers and Programmable Automation Controllers (PAC).

With the use of appropriate interface cards, driver software and programming tools a traditional PC can be used to control industrial applications. The main benefits of a PC based controller are high level programming, increased computing power, superior Human Machine Interface (HMI) and networking. On the other hand, PAC combines the capabilities of PLC and PC into a single unit. PACs are programmed like a PC to carry out PLC applications in an industrial setting.

The difference between these controllers has been blurred with the adoption of open standards, in terms of both hardware and software. Controllers are, currently, built with hardware, such as, Ethernet, PROFI Bus and SERCOS. These standards enable networking between controllers and other programmable devices. On the software side, there has been a rise in general programming software due to open communication standard, like, OPC. OPC is short Object Linking & Embedding (OLE) for Process Control, it allows PLC to communicate with software operating Microsoft Windows.

Due to such high level of integration, engineers have complete freedom over how an automated application is developed. Today, it is possible to interface a PLC with programming software of the engineer's choice. The way users interact with a machine has also changed with the use of mobile devices as HMIs. Finally, every device in an automated system can communicate with each other over a wireless network.

II. LITERATURE SURVEY

Chang *et al.* [1] describe an architecture for a distant web controlled PLC lab. Authors note the benefits offered by the internet for engineers and scientists and its applicability for development of remote labs. Their architecture for the remote lab is built using internet tools, such as, Visual Basic, HTML, ActiveX and ASP. The software used for PLC programming is WinPreloader. In the architecture, the PLC communicates with the server PC via RS-232 cable.

Shyr *et al.* [2] have developed a system for remote learning of a Mechatronics system. The work described is similar to the one in the previous paper. However, instead of a PLC, users control a pneumatic powered pick and place robot. The mechatronics module is controlled by PLC, which communicates with server system through an RS-232 cable. On the server system, the data from the PLC is uploaded to the internet using TCP-IP. Clients can access the web page by entering the IP address in their web browser.

Jamal and Wenzel [3] explain the use of LabVIEW in real world applications. In the paper authors provide in depth history of LabVIEW programming language and its evolution since its creation. In addition, various programming features of LabVIEW are also presented. The authors have illustrated various benefits of LabVIEW, such as, debugging, instrument hierarchy, rapid prototyping, etc.

Kerry and Snyder [4] compare the use of LabVIEW with text based programming languages. In this paper, the authors write about the flexibility and reusability that is often necessary in modern applications. Illustrative comparisons are provided to between LabVIEW and text based programs in terms of productivity, code quality, longevity and scalability.

The literatures presented clearly emphasize that industrial internet, wireless connectivity and use of mobile devices for control and monitoring of equipment will widespread in the future. The work presented in this paper proposes a generalized client – server architecture for PLC based automation. This architecture will allow great deal of flexibility in terms of application development. It will enable remote development, control and monitoring of applications. Multiple deployment methods will support the access of applications on devices of clients’ choice.

II. OVERVIEW OF CLIENT – SERVER ARCHITECTURE

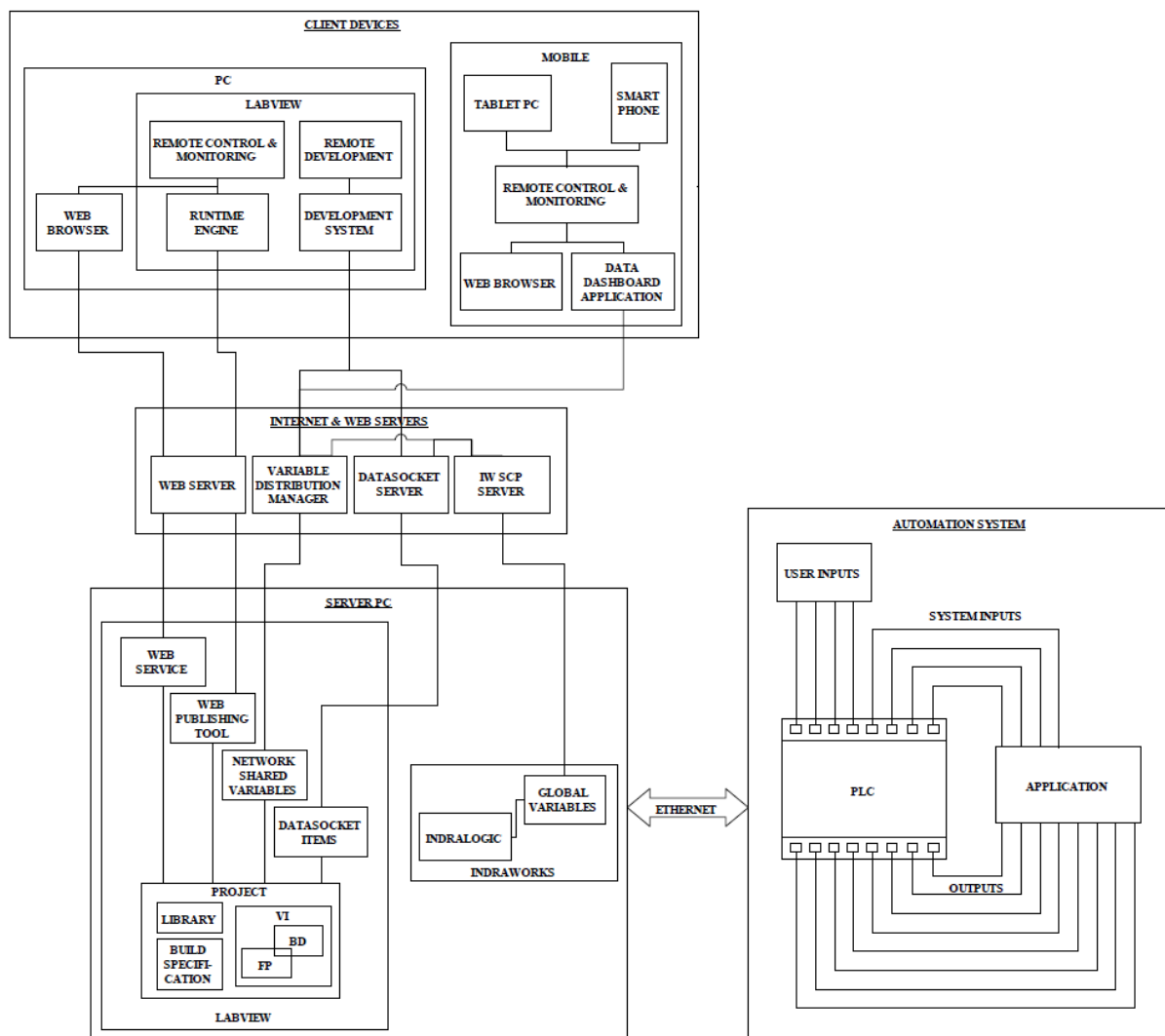


Figure 1: Block diagram of the client – server architecture

The principle, here, is to design a client – server system (Fig. 1) for the development, control and monitoring of PLC applications. In this system, clients are the users, who send request for the access of the application. Server is the PC that handles tasks, such as, connecting to the PLC, publishing data to Internet/Web and handling of clients.

The PLC, IndraControl L10, is physically connected to the server PC through an Ethernet cable. On the server PC, IndraWorks is used to configure the PLC and define the various inputs and outputs (I-Os). Using the programming environment of IndraLogic, the I-Os can be declared as global variables, which publishes them to OPC server. By using the OPC server it is possible to link IndraWorks with LabVIEW. Within, LabVIEW the PLC variables can be imported in two distinct ways, DataSocket and Shared Variables. DataSocket allows I-Os to be accessed with a simple URL and does not need any low-level programming.

Once the variables have been imported, applications can be developed in the LabVIEW environments. LabVIEW is a graphical language that allows creation of user interface as well as programming within a single environment. Here, the interface is built in the “Front Panel” and the programming is carried out in the “Block Panel”. After VI development, numerous LabVIEW technologies can be utilized for accessing the applications through Internet/Web. In this project, the various technologies used are Web Services, Web Publishing, Network Shared Variables and DataSocket. Here, each of technology is used for a certain purpose such as, simple monitoring, control and development across various client devices.

III. IMPLEMENTATION

1.1. Pneumatics

Pneumatics is a field of engineering where applications are developed by generation and control of compressed air. Pneumatic drives are often used in automated application due to ubiquitous availability of air, cleanliness, simple setup, repeatability and motion control. Within industrial automation, pneumatics is often utilized for applications, such as, material handling and machining operations.

1.2. Bending Machine – Physical Setup

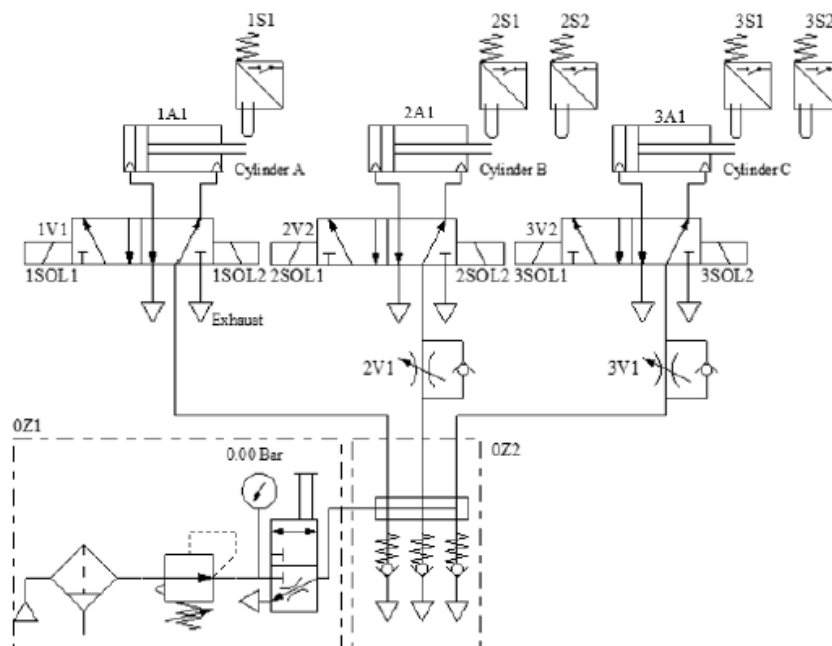


Figure 2: Pneumatic circuit for bending machine

The application that has been developed for the demonstration of the architecture is a miniaturized pneumatic bending machine. The purpose of this machine is to bend a work piece (sheet metal) as required. The process is as follows: An operator places a piece of sheet metal in the machine and starts the sequence; a pneumatic cylinder extends to clamp the work piece; two other cylinders extend to bend the sheet metal at right angles, and all three cylinders retract and the work piece is released.

The bending machine circuit (Fig. 2) begins with the compressor unit (OZ1), which provides the necessary pneumatic energy to drive the circuit. The outlet of the compressor is connected to a 6-fold distributor (OZ2). From this distributor, air can be supplied to the necessary components. Two flow control valves (2V1 & 3V1) are used to control the flow of air to the bending cylinders. Three independently operated directional valves (1V1, 2V2 & 3V2) are used to actuate the cylinders (1A1, 2A1, 3A1). Positions of the cylinders are indicated by the sensors (1S1 to 3S2).

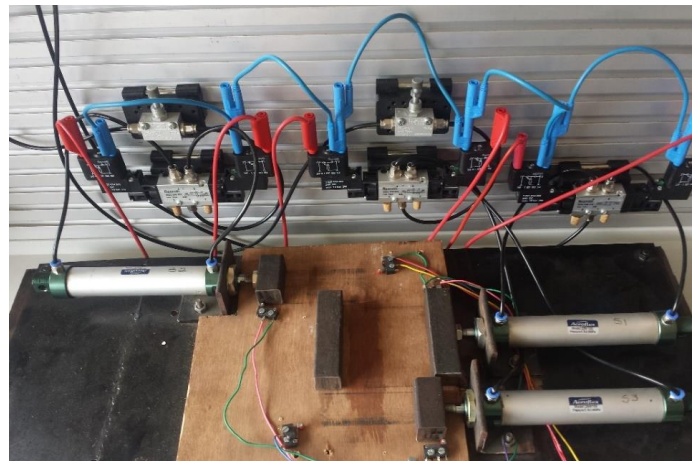


Figure 3: Physical setup of the bending machine

Fig. 3 shows the control and actuating setup of the bending machine. For pneumatic supply a 230V compressor has been utilized. It has an operating pressure of 6 - 8 bars and air quantity is 50 l/min. It also has certain accessories, such as pressure gauge and controller. The throttle check valve is a manually operated valve with a nominal flow of 135 l/min. The DCV used for actuation are 5/2 solenoid pulse valves. They consist of 5 ports and 2 positions for actuation. Each of the valves is triggered by a pulse signal from the PLC. The actuator, double-acting cylinder, has a 25mm diameter piston and 100mm stroke.

1.3. PLC Setup and Program

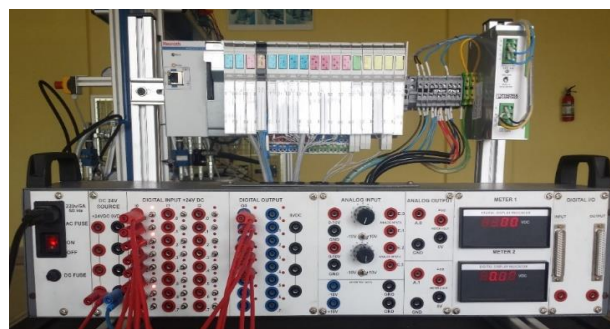


Figure 4: Rexroth L10 PLC

PLC utilized for this application is Rexroth IndraLogic L10 with custom inline input – output modules (Fig. 4). It can handle 24 digital inputs and provide 20 digital outputs. The bending machine consists of 5 digital inputs (limit switches) and 6 digital outputs (solenoids). The input limit switches have been connected to the PLC input module from address %IX1.1 to %IX1.5. Similarly, the outputs have been drawn from output ports %QX1.4 to %QX4.1. Two additional signals that have been drawn from the PLC are the 0V ground and 24V power supply.

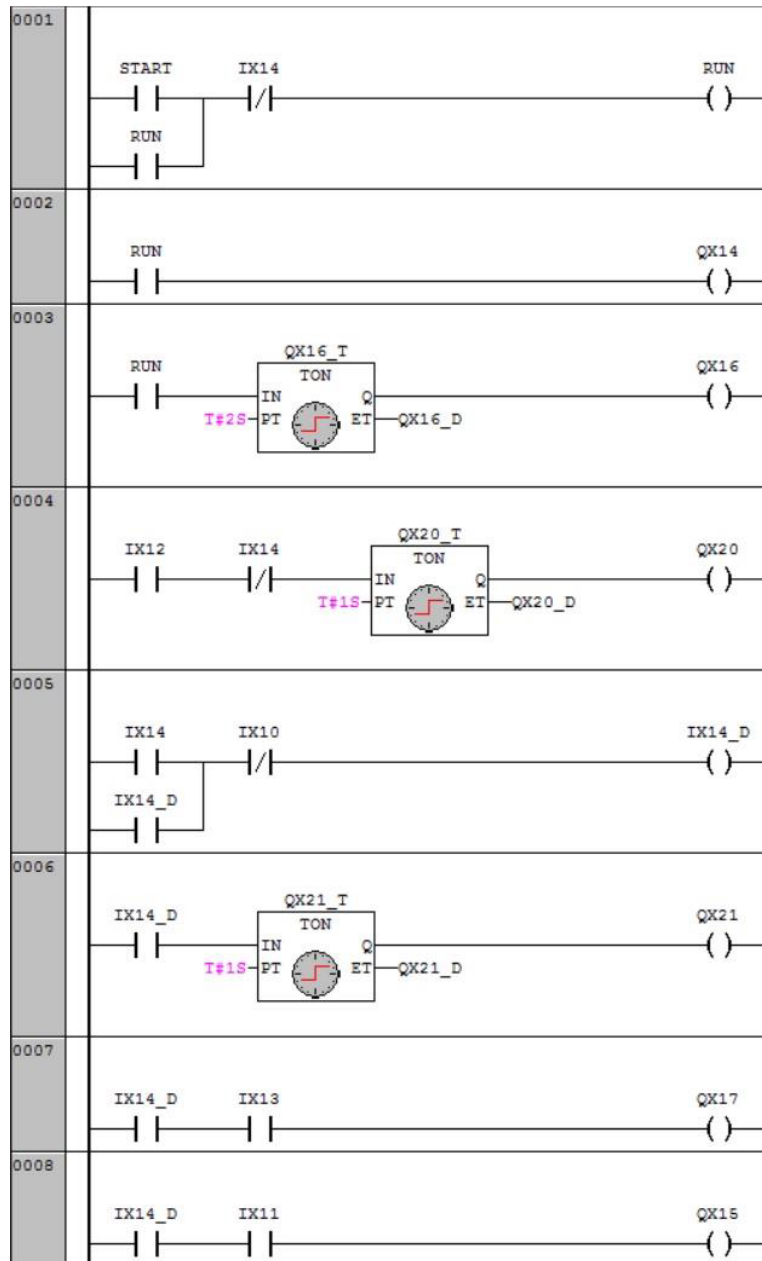


Figure 6: PLC Program

- Line 1: When user pushes the “START” button, the corresponding contact closes and coil “RUN” is activated and latched.
- Line 2: Contact “RUN” triggers the coil “QX14” which activates “1SOL1” and cylinder A extends.

- Line 3: Contact “RUN” triggers the ON delay timer “QX16_T”, after 2 second delay, coil “QX16” energizes and cylinder B extends.
- Line 4: When cylinder B completely extends, limit switch “2S2” is turned ON and its corresponding contact “IX12” closes. Timer “QX20_T” is triggered and after 2 second delay coil “QX20” activates and cylinder C extends.
- Line 5: Cylinder C extends to close the limit switch “3S2” and its corresponding contact “IX14”. This value is latched with coil “IX14_D”.
- Line 6: Contact “IX14_D” closes and after 1 second delay coil “QX21” is energized and cylinder C retracts.
- Line 7: Cylinder C retracts and closes limit switch “3S1” and contact “IX13”. This activates coil “QX17” and cylinder B retracts.
- Line 8: Cylinder B retracts to closes limit switch “2S1” and contact “IX13”. Coil “QX15” and cylinder A retracts. When cylinder completes retraction, limit switch “1S1” and contact “IX14” close and the entire sequence is reset.

4. 4. LabVIEW Interface

Methodology explained in the literature [3] and [4] have been utilized to link IndraWorks Engineering with LabVIEW via the OPC server. As LabVIEW executes the Virtual Instrument (VI) on the server PC and not the PLC, computing capabilities of the PC can be exploited to develop a VI that provides a graphical HMI, gathers data associated with the sequence and publishes it to the network for client access. The fully developed LabVIEW HMI is as shown in Fig. 7.

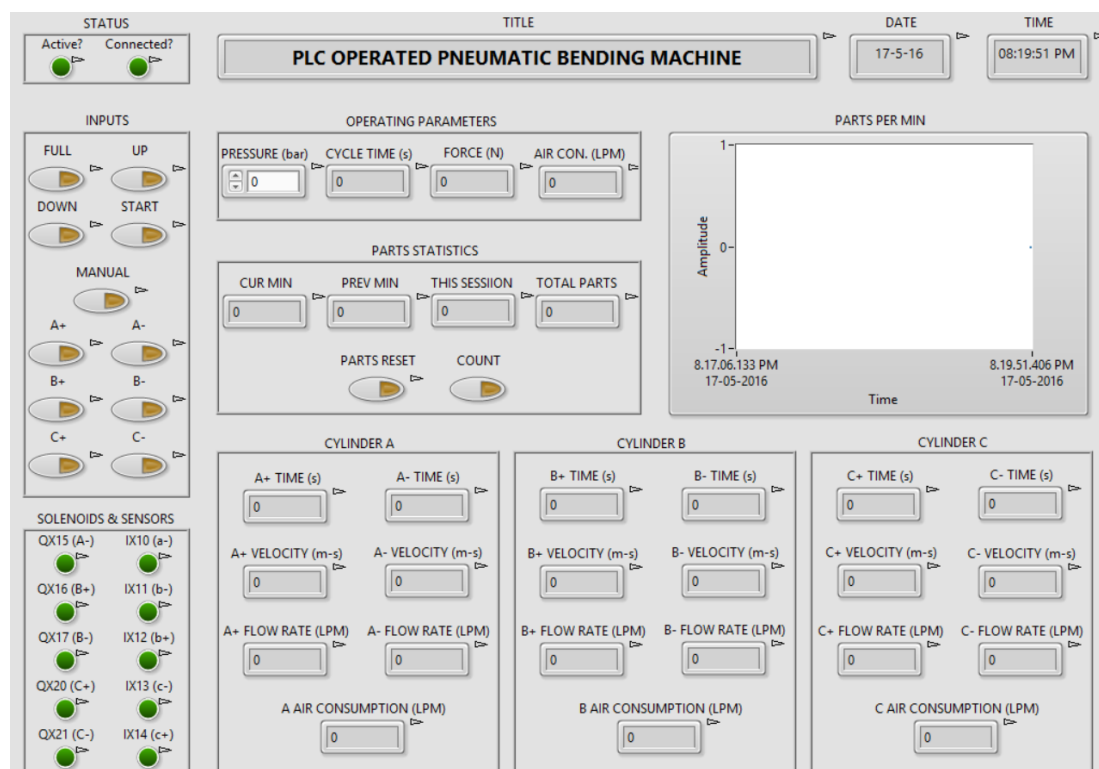


Figure 7: LabVIEW HMI for bending machine

For the bending machine, two types of client – server deployment methods have been chosen. First method is the Network Variable based Data Dashboard app (Fig. 8). Since the bending machine requires user inputs, smart phones and tablets can be utilized as an HMI to control the machine.

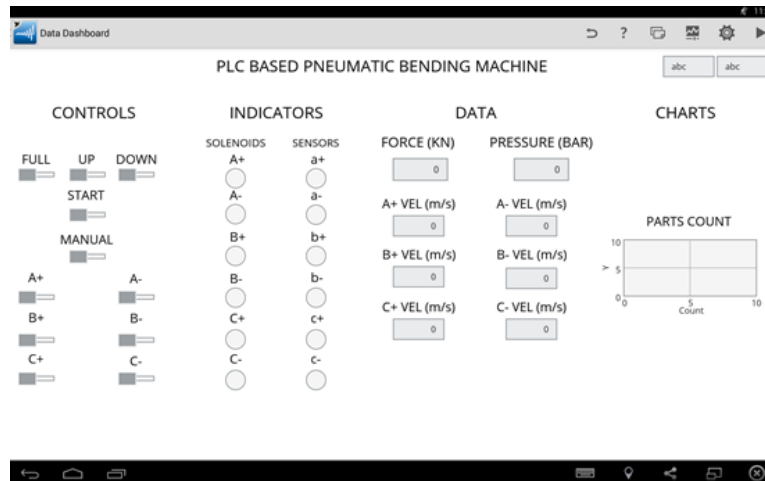


Figure 8: HMI dashboard

The second method is Web Service, which will be utilized for the monitoring of the machine. One of main objective of the application is to generate relevant information regarding the bending sequence and distribute to clients. For distribution of data, Web Service is most suitable as it has the widest reach and can be accessed on multiple devices.

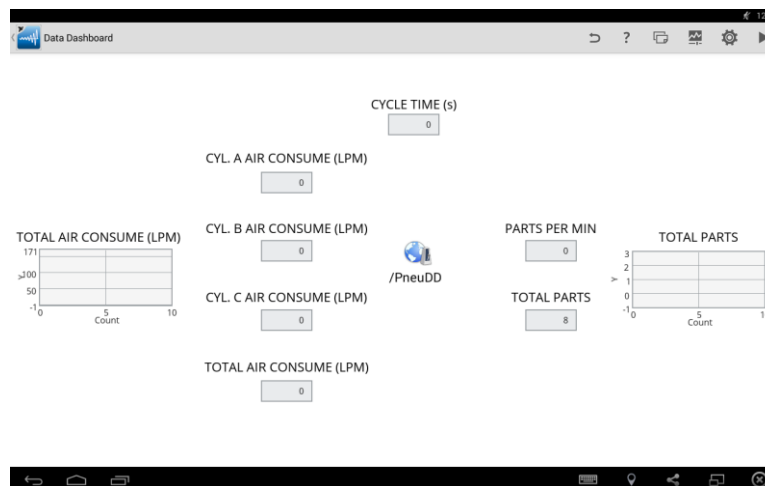


Figure 9: Web service dashboard

V. RESULTS

The execution of the client – server architecture mainly depends on the successful deployment of shared variables and web service. Deployment ensures that client devices can connect to the network and request access to the machine. Fig. 10 shows the deployment of shared variables for control of the sequence and web service for monitoring of the bending machine.

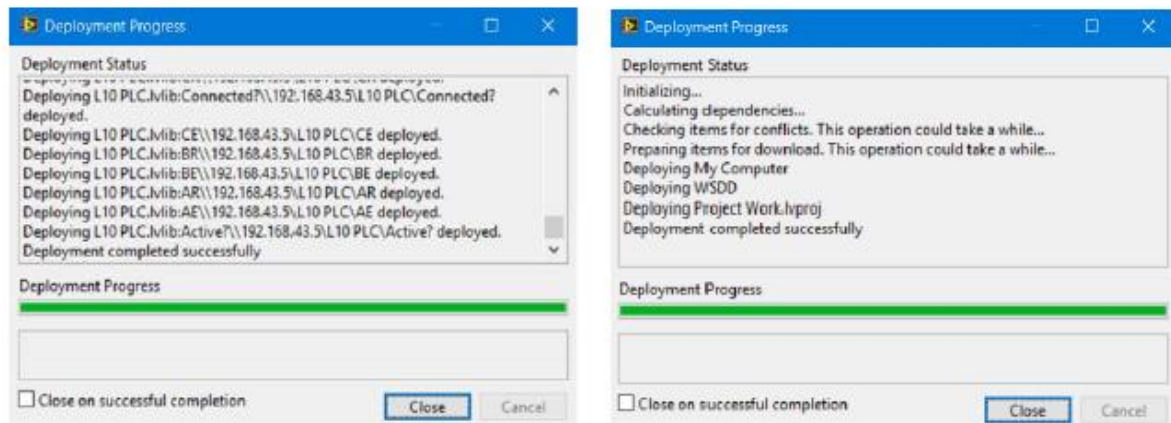


Figure 10: Deployment of shared variables and web service

With LabVIEW, the server system’s superior computing capabilities are utilized to provide a flexible and intuitive HMI for the operation of the bending machine. Fig. 11 shows the Dashboard HMI, which allows user to run the sequence in the required mode. In this particular instance, the machine is being operated in manual mode for testing purpose.

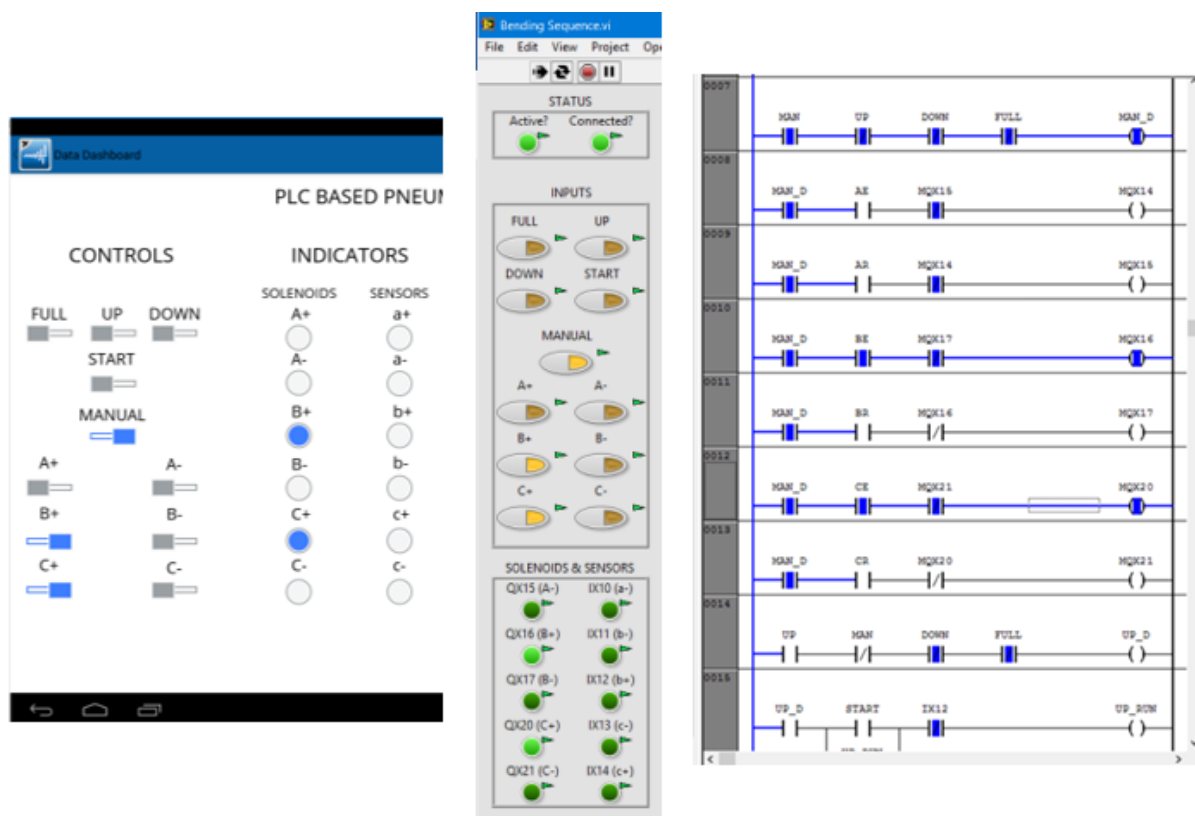


Figure 11: Successful connectivity between LabVIEW and IndraWorks

Fig.12 shows the complete view of the Dashboard in operation mode. In addition to manual operation, users can choose between full and half cycle based on requirements. In addition, to the controls and status indicators, user is also provided with data associated bending operation, such as, force, pressure and velocity. This data allows

user to utilize the machine more efficiently. Data about force and pressure allows the user to determine whether the required level of force is available for the bending process.

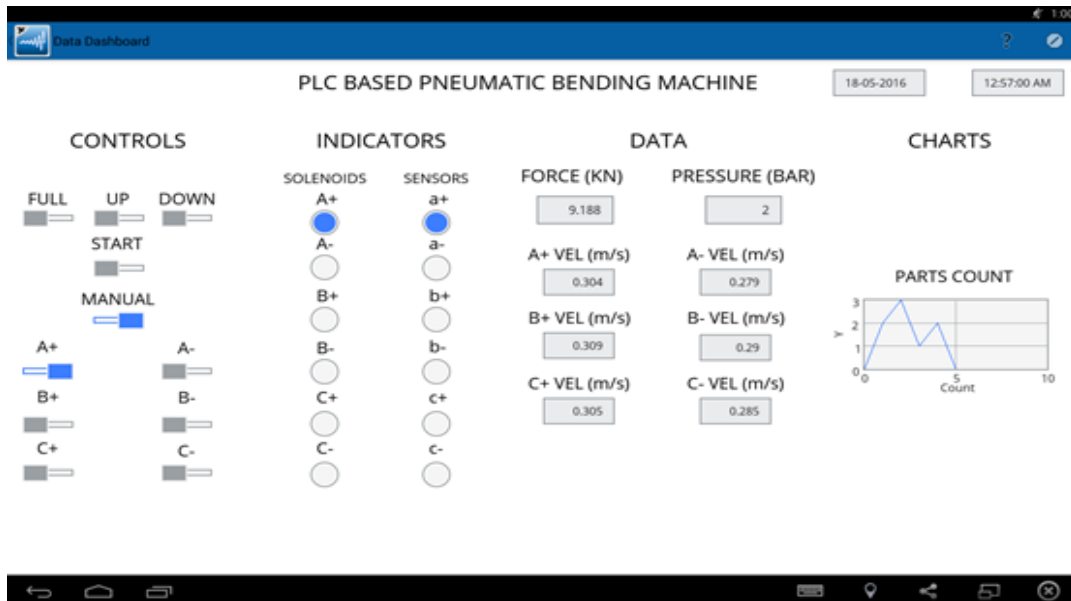


Figure 12: HMI dashboard during operation

Web services such as the one shown in Fig. 13 provides real data about the applications that are useful for documentation and analysis. As these services are distributed over the internet, multiple clients can access simultaneously. These Web Services can instantaneously provide data for something repetitive as velocity and flow rate of a cylinder stroke to long time trends in air consumption or history of parts produced. Since a web service is machine – to – machine communication, the data collection and distribution are completely automated and does not require any human work once they have been deployed. This often results in more productivity and time savings,

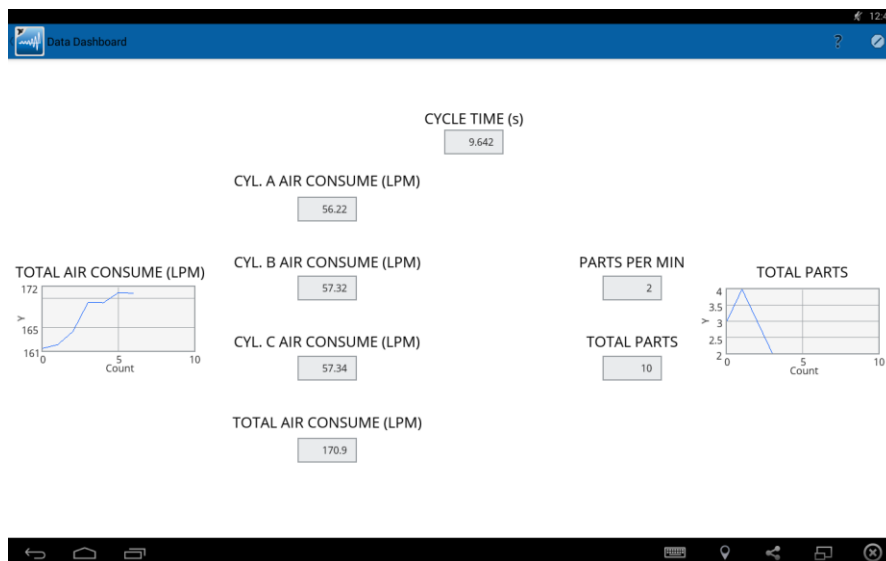


Figure 13: Web service dashboard showing information about the status of the bending machine

From Fig. 14, it is easy to understand how Web Service eliminates a number of manual tasks and reduce time taken for reporting and documentation. Since data collection and distribution is automated, personnel can spend their time on interpreting and analyzing the data. Another advantage of Web Service is that it always provides real time data, which can be used notice trends and downtimes. In addition, Web Service can also provide information related to fault and diagnosis.

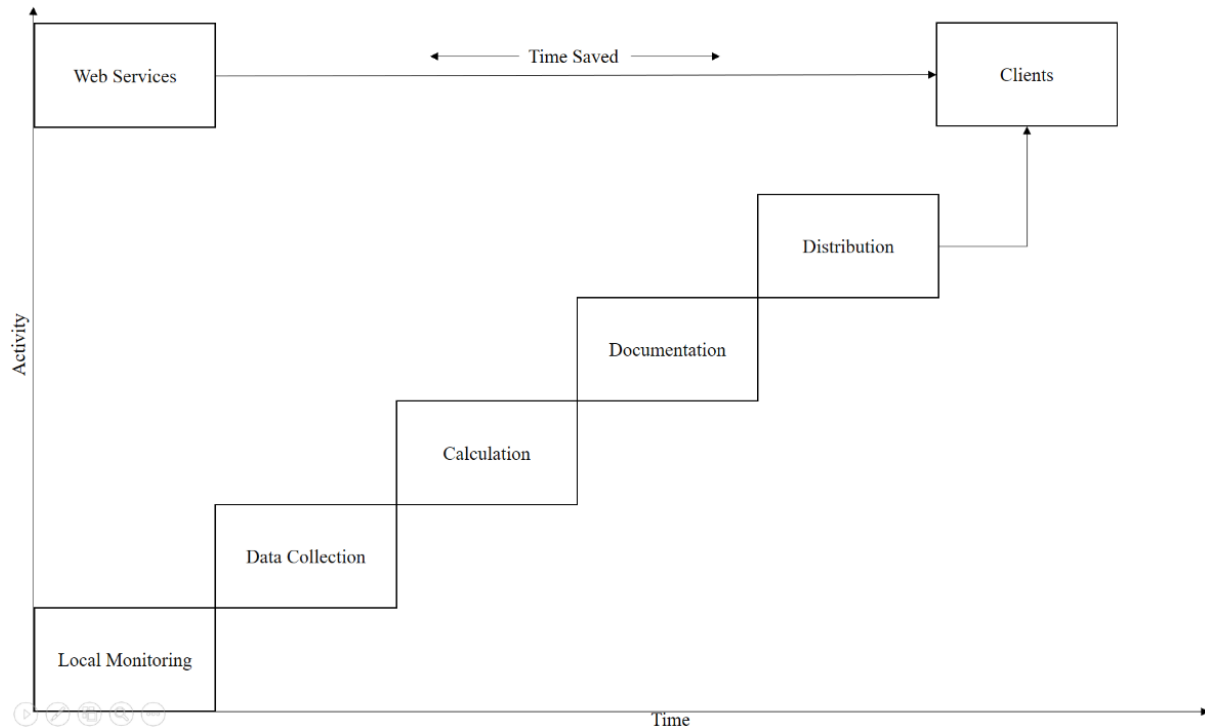


Figure 14: Time saved by employing web service for monitoring

VI. CONCLUSION

The Client – Server architecture for PLC based Automation was successfully developed and implemented. It showcased the benefits of high level programming language, such as, LabVIEW to carry out data collection, HMI creation and network connectivity. Control and monitoring of the bending machine through a mobile device illustrated the benefits of deploying a virtual/soft HMI. It was found that a virtual HMI can greatly reduce cost and application development. Similarly, Web Service automated reporting and documentation, thus enabling personnel to spend time on interpreting and analyzing the data. For future scope, the architecture can be extended to provide analysis and trend in the collected data. New deployment methods can be introduced, which have been built natively to enable user customizations. The architecture can be utilized for applications, such as, condition monitoring and analog systems that require extensive programming and interface development.



REFERENCES

- [1] Chang et al. “Design and Implementation of a Web-based Distance PLC Laboratory”, Proceedings of the 35th Southeastern Symposium on System Theory, Morgantown, 16-18 March 2005.
- [2] Sang-Gu Choi, Sang-Hyun Ryu and In-Ywan Park, “Development of Web-Based Control and Monitoring System for Facility in Shipbuilding Yard”, Proceedings of the 11th International Conference on Control, Automation and Systems, 2011.
- [3] Wen-Jye Shyr, Te-Jen Su and Chia-Ming Lin, “Development of Remote Monitoring and Control System Based on PLC and WebAccess for Learning Mechatronics”, International Journal of Advanced Robotic Systems: Mechatronics, 2015.
- [4] Rahman Jamal and Lothar Wenzel, “The Applicability of the Visual Programming Language LabVIEW to Large Real-World Applications”, IEEE, 1995.