Journal of Engineering and Development, Vol. 16, No.2, June 2012 ISSN 1813- 7822

Distributed Real-Time Monitoring and Control of Industrial Drive Systems via Ethernet

Ammar Ibrahim Majeed
Department of Electrical Engineering,
College of Engineering, Al-Mustansiriya University
e-mail: ammar_19722@yahoo.com

Abstract

Real-time electronic distributed control systems are an important development of the technological evolution. Electronics are employed to control and monitor most safety-critical applications from flight decks to hospital operating rooms. As these real-time systems become increasingly prevalent and advanced, so does the demand to physically distribute the control in strict real-time. Thus, there is a need for control network protocols to support stringent real-time requirements. Real-time networks must provide a guarantee of service so they will consistently operate deterministically and correctly. Ethernet is nowadays being focused by various automation system developers over other field bus systems due to its cheap hardware availability, being wireless, straightforward integration to the Internet and support for the higher bandwidth requirements in the future *[1]*. ***It is also emerging strongly into the area of industrial communication. Ethernet, as defined in IEEE 802.3, is non-deterministic and thus, is unsuitable for hard real-time applications. The media access control protocol, CSMA/CD with its back off algorithm, prevents the network from supporting hard real-time communication as due to its random delays and potential transmission failures. This paper presents design and implementation based on an extensive research and development being carried out to enhance the possibilities of using standard TCP/IP Ethernet protocol for condition monitoring and distributed real-time control of industrial drive systems via Ethernet.***

Keywords: Distributed Monitoring and Controlling, Real Time, Ethernet, Visual Basic.

منظومات الزمن الحقيقي للمراقبة و السيطرة الموزعة للمشغلات الصناعية باستخدام الايثرنت

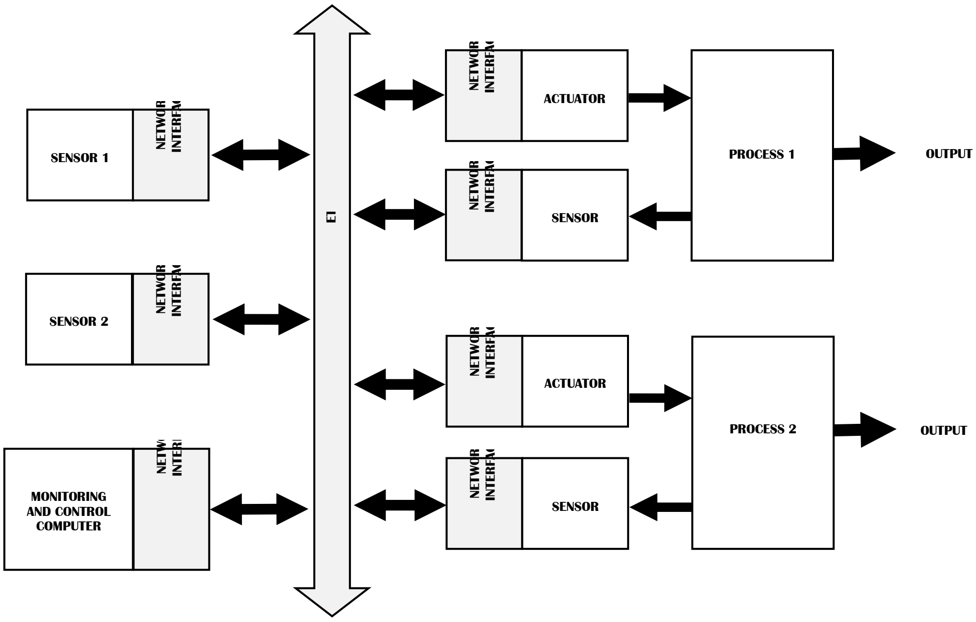
تعتبر منظومات الزمن الحقيقي للسيطرة الالكترونية الموزعة من المنظومات المهمة في التطور التكنولوجي الحالي, حيث تستخدم التقنيات الالكترونية في السيطرة والمراقبة لاغلب التطبيقات التي يتطلب انجازها الكثير من الدقة والعناية الفائقة كونها تتعلق بسلامة الافراد والمنشات والممتلكات ابتداءا من ادارة مدارج المطارات الى غرف العمليات

بالمستشفيات وما الى ذلك. وفي الاونة الاخيرة اصبحت هذه المنظومات اكثر استخداما وتطورا وانتشارا مما دعا الى الحاجة لتوزع السيطرة لتغطي منطقة جغرافية اوسع الامر الذي ادى الى الحاجة لوضع نظم لشبكات السيطرة لدعم متطلبات الزمن الحقيقي لهذه الشبكة. يجب ان تضمن شبكات الزمن الحقيقي الخدمة المطلوبة منها اثناء فترة استخدامها وان تعمل بطريقة محددة لها مسبقا وبشكل صحيح. اصبحت شبكات الايثرنت هذه الايام محط اهتمام الكثير من مطوري منظومات التحكم والمراقبة وذلك لرخص وتوفر اجهزتها ولكونها لاسلكية احيانا كثيرة وسهولة الربط مع الشبكة العالمية الانترنت ودعمها في المستقبل القريب لعرض حزمة واسع. هذا بالاضافة لاستخدامها المتزايد في شبكات الاتصال والتحكم الصناعي. ان شبكات الايثرنت وكما هو معرف في المقياس 802.3 بانها غير حتمية ولهذا فانها غير صالحة للتطبيقات الواسعة للزمن الحقيقي. ان نظام CSMA/CD لشبكات الايثرنت يمنع الشبكة من ان تدعم اتصالات الزمن الحقيقي الواسعة وذلك للتاخر العشوائي وحالات فشل ارساله المحتمل. هذا البحث يستعرض عملية تصميم وتنفيذ منظومة سيطرة ومراقبة موزعة باستخدام الزمن الحقيقي مبنية على عمليات بحث وتطوير متواصلة وشاملة لتحسين احتمالات استخدام بروتوكول شبكة الايثرنت TCP/IP لعمليات السيطرة والمراقبة لمنظومة زمن حقيقي موزعة للتحكم الصناعي باستخدام الايثرنت.

1. Introduction

Condition monitoring and closed-loop controls are essential and well-known techniques in any industrial environment. A controller or an observer receives information about the industrial drive system (or the process) to be controlled or observed from the sensors and in case of a controller it sends out driving signals to the actuator [2]. Condition monitoring done from a location remote to the place at which the particular industrial process is commissioned is at the tip of today's cutting edge technology (e.g. Control room of a factory). Control loops that are closed over a communication network, called Distributed Control Systems (DCS), also get more and more common as the hardware devices for network and network nodes become cheaper thanks to advanced cost effective silicon technology. One important feature of such a distribution is that, instead of hardwiring the control devices with point-to-point connections, sensors, actuators and controllers are all connected to the local area network (LAN) as nodes. Several advantages of this implementation include: reduced system wiring, plug and play devices, increased system agility and ease of system diagnosis and maintenance. In such a system, measurement and control signals are transmitted between process and controller/observer modules as encapsulated data packets. These types of industrial applications demand fast, flexible, secure, reliable and robust data communication at a reasonable cost. Employing a suitable fieldbus full-fills some of them. Profibus, ControlNet, DeviceNet, Ethernet, Suconet, and Interbus etc are among the commonly used field-buses. One major requirement of such a system regardless of the vendor, is its ability to connect any physical sensor or actuator to the network with minimum system administrative overhead and cost. In other words it is the interfacing of the sensor/actuator node to the communication network without much of a burden.

Being a versatile networking hardware and software solution developed over two decades, Ethernet has received a lot of attention from industry as the future industrial communication medium. One objective of this research is to address this problem of interfacing of the sensor/actuator node to the communication network, when the communication is done via Ethernet. The second objective is to investigate the possibility of using standard TCP/IP Ethernet for distributed real-time control of industrial drive systems [2],



(a) Condition Monitoring ***(b) Distributed Control***

Figure 1: Ethernet for industrial communication

The Ethernet based system topologies shown in Figure 1 (a) and (b) respectively would enable condition monitoring and distributed real-time control as depicted.

2. Ethernet

In the mid-1990s, standardization activities were started both in the United States and in Europe. While the U.S. activities (UCA 2.0—Utility Communication Architecture) primarily focused on standardization between the station and bay levels, the European approach (driven by IEC TC57, WG 10, 11, and 12) included the communication down to the time-critical process level from the beginning. In 1998, the two activities were merged to define one worldwide applicable standard: IEC 61850 [3]. Instead of debating between several competing fieldbuses, an agreement was reached to use Ethernet as a communication base for the station bus. This agreement was based on the fact that the Ethernet technology has evolved significantly. Starting out as a network solution for office and business applications, Ethernet today is applied more and more as a solution for high-speed communication backbone applications between PCs and industrial networks. The high-speed properties of current Ethernet technology, together with its dominant position in the Local Area Networks

(LAN), makes Ethernet an interesting communication technology condition monitoring and distributed control of industrial drive systems [4].

2.1. Traditional Ethernet

Traditional Ethernet, as defined in IEEE 802.3, is unsuitable for strict real time industrial applications because its communication is non-deterministic. This is due to the definition of its media access control (MAC) protocol, based on Carrier Sense Multiple Access/ Collision Detection (CSMA/CD). The implementation described in the standard uses a truncated binary exponential back off algorithm. With CSMA/CD, each node can detect if another node is transmitting on the medium (Carrier Sense). When a node detects a carrier, its Carrier Sense is turned on and it will defer transmission until determining the medium is free. If two nodes transmit simultaneously (Multiple Access), a collision occurs and all frames are destroyed. Nodes detect collisions (Collision Detection) by monitoring the collision Detect signal provided by the physical layer. When a collision occurs, the node transmits a jam sequence. When a node begins transmission there is a time interval, called the Collision Window, during which a collision can occur. This window is large enough to allow the signal to propagate around the entire network/segment. When this window is over, all (functioning) nodes should have their Carrier Sense on, and so would not attempt to commence transmission. Figure 4 below show the principles of CSMA/CD [5].

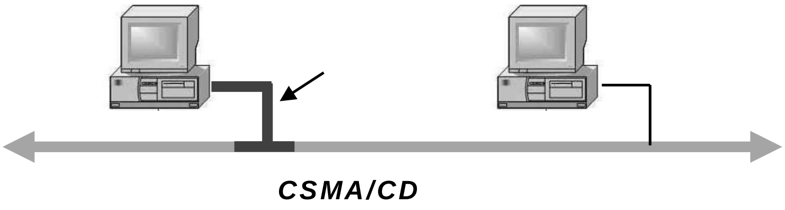


Figure 2: Carrier Sense

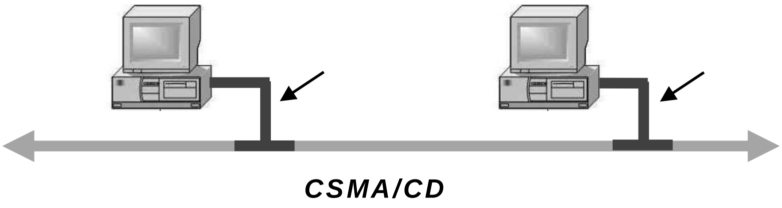


Figure 3: Multiple Access

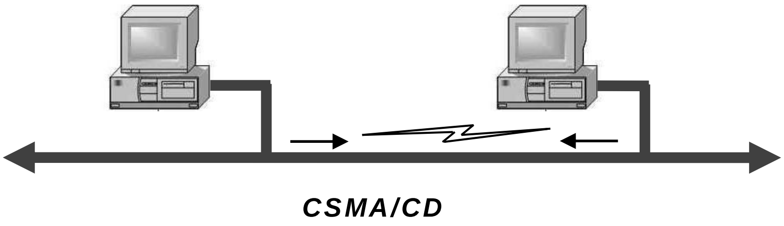


Figure 4: Collision Detection

2.2. Switched Ethernet

Switches are data-link layer hardware devices that permit single-collision domains through network segmentation. While a bridge operates like a switch, it only contains two ports compared to switches that have more than two with each port connected to a collision domain. Switches can operate in half duplex or full duplex mode [5]. When full duplex switches are used with full duplex capable nodes, no segment will have collisions. Today’s switches are more intelligent and faster and with careful design and implementation could be used to achieve a hard real time communication network using IEEE 802.3. Although switches are data-link layer devices, they can perform switching functions based on data from layers 3 and 4. Layer 3 switches can operate on information provided by IP - such as IP version, source/destination address or type of service. Layer 4 devices can switch by source/destination port or even information from the higher-level application. Further refinements to the IEEE 802 standards, specifically for switch operations, are 802.1p and 802.1Q. IEEE 802.1p (incorporated into IEEE 802.1D [6]) brings Quality of Service (QoS) to the MAC level and defines how these switches deal with prioritization — priority determination, queue management, etc. This is achieved by adding a 3-bit priority field to the MAC header, giving 8 (0-7) different priority levels for use by switches or hubs. As defined, 802.1p supports priorities on topologies compatible with its prioritization service, but for Ethernet, which has no prioritization field in its frame format, it uses 802.1Q. IEEE 802.1Q [7] defines an architecture for virtual bridged LANs, their services and the protocols and algorithms used by those services. 802.1Q allows Ethernet frames to support VLANs (Virtual Local Area Networks)—limiting broadcast domains and thereby reducing broadcast traffic on the entire LAN. This is achieved by inserting 4 bytes between the source address and length/type fields in the frame header, which among other identifiers, includes that of the originating VLAN. For a real time Industrial Ethernet application, an 802.1p/Q implementation has certain advantages: it introduces standardized prioritization, allowing control engineers up to eight different user-defined priority levels for their traffic. But these standards also have drawbacks including the extra hardware costs for the increased frame length (1522 bytes)—which introduces compatibility issues with legacy Ethernet networks. A real time implementation using 802.1 p/Q requires full duplex, switched Ethernet. IEEE 802.1p/Q are acceptable for certain applications of real time Ethernet in industry when switch ‘through’ time is predictable and an overload situation will not result in hard deadlines being missed. Although switches can certainly provide real time deterministic Ethernet communication and are the backbone of the Industrial Ethernet solutions available today, they have drawbacks. They are costly—a major influence on cost-conscious industries. They are powered devices capable of failure (a major factor for hard real time control operations). And sometimes the operational predictability is not guaranteed by the manufacturer. A study on switches for real time applications is available at [8].

2.3. TCP/UDP/IP for Real-Time Ethernet

With Industrial Ethernet, the trend is to define an application-layer environment along with the TCP/IP protocol, to realize an industrial automation networking solution. Some real time Ethernet solutions (e.g., EtherNet/IP) perform all their communication, real time included, through the TCP/UDP/IP stack. But most solutions, while providing TCP/IP compatibility, do not employ this protocol for real time communication. In a system like EtherNet/IP, TCP is used for initialization and configuration of explicit messages while UDP, with its reduced overhead, is used for real time I/O (implicit messaging). Typically, real time Industrial Ethernet applications are compatible with TCP/IP, but the protocol suite is bypassed for all real time communication. The ability of a real time Ethernet solution to intercommunicate with an office based system is paramount to achieve the Ethernet technology plant of the future [5]. Ethernet switches provide 10M, 100M, 1G bps or even 10 Gbps (under development) on each drop link. This represents a scalable and huge bandwidth increase compared to e.g. an Ethernet hub where the bandwidth is either 10 or 100 Mbps and shared between all users connected to the same network segment [2].

3. Distributed Controller

Figure 5 outlines the timing aspects of a distributed real-time control system [2].

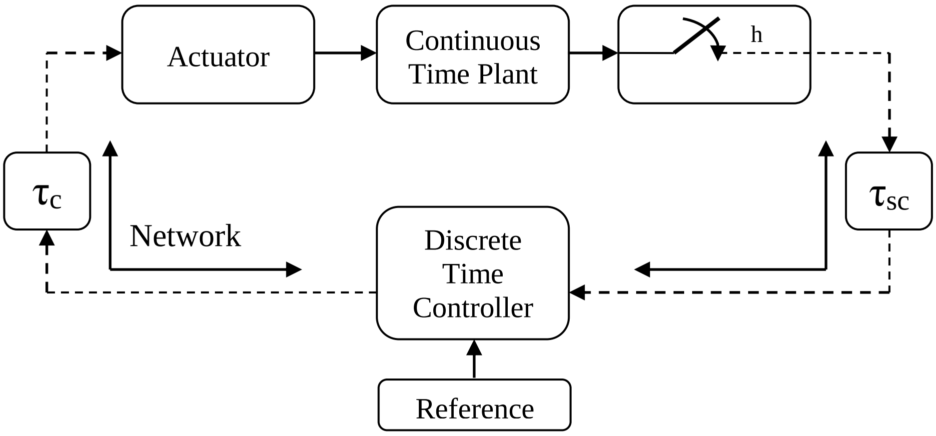


Figure 5: Controller distributed over the communication network: τ_{sc} sensor to controller delay, τ_{ca} controller to actuator delay, τ_c controller execution delay

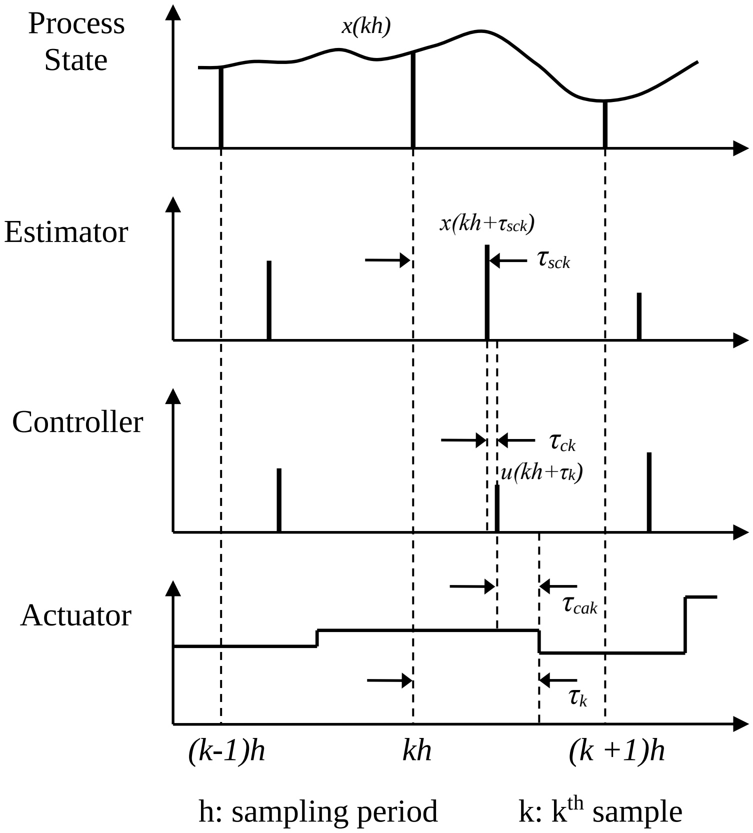


Figure 6: Timing diagram for delays involved in various nodes of the distributed control system [2]

Despite the Switch, state measurements $x(kh)$ of a system distributed over an Ethernet network can get delayed in reaching the controller node as shown in Figure 5. By that time, the actual process/plant state may have changed (in the Figure 6, the process state $x(kh)$ is different from $x(kh+\tau_{sc})$). Therefore an estimator must be used to evaluate the state measurements pretending the states just before the control signal has been released at the actuator node. This is essential as there is another delay τ_{ca} before the control signal reaches the actuator. The control delay τ is unknown prior to the control signal computation and is therefore estimated from the known τ_{sc} and τ_c. The real-time delay compensation scheme based on time stamped state measurements is well described in [2] by the same author.

4. System Hardware

The total system hardware is shown in figure 7 below. The system is consisting of; Ethernet network (three workstations with connecting hub), DC stepper motor with its interfacing and driving circuits connected to workstation-1(ws1) through the parallel port, temperature sensor and its signal conditioning circuit and ADC converter connected to workstation-2(ws2) through the parallel port too and a monitoring and controlling workstation represented by workstation-3 (ws3). In the suggested system, the monitoring and controlling workstation (ws3) collecting the data of the monitored variable (here temperature) from the workstation-2(ws2) and generating control commands for the stepper motor connected to workstation-1(ws1).

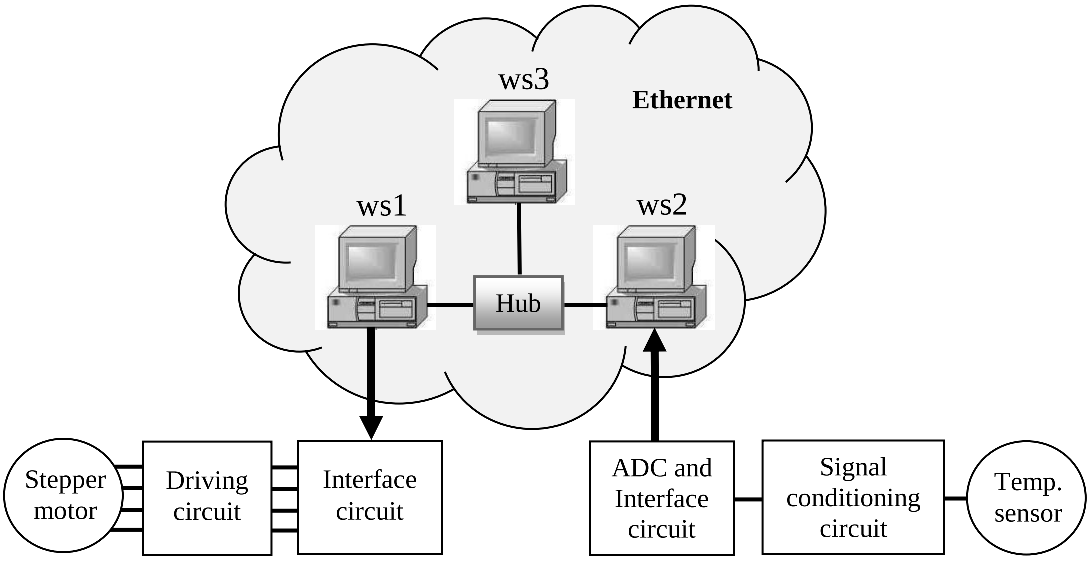


Figure 7: Total system hardware

4.1. Design Approach

The complete design problem is divided into two phases. Namely, Phase(I): Interfacing of stepper motor to the PC including the design of driving and interface circuits and Phase(II): Interfacing of a temperature sensor to the PC including the design of signal conditioning and ADC conversion.

4.1.1. Stepper Motor and Its Driving Circuit

The stepper motor that used in this work is a unipolar (type 42SPM-24DCZA). Unipolar stepping motors, both permanent magnet and hybrid stepping motors with 5 or 6 wires are usually wired as shown in the schematic of figure 8 below. In use, the center taps of the windings are typically wired to the positive supply, and the two ends of each winding are alternately grounded to reverse the direction of the field provided by that winding.

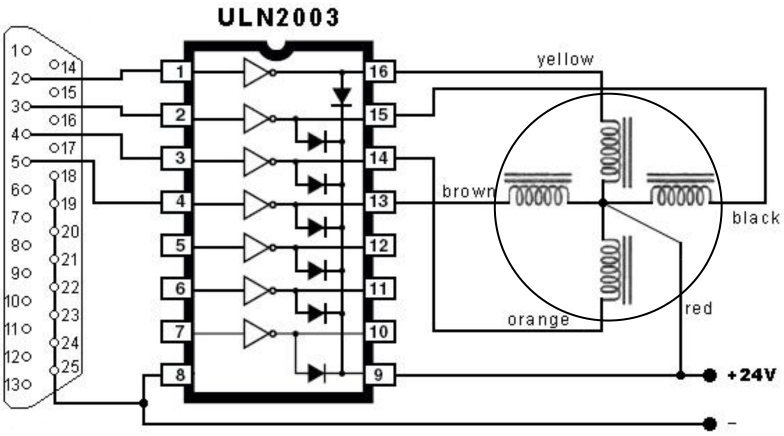


Figure 8: Stepper motor driving circuit

This circuitry is centered on a single issue, switching the current in each motor winding on and off, and controlling its direction. The circuitry introduced here connects the motor windings to the motor power supply directly, and it is controlled by a PC through the parallel port that determines when the switches are turned on or off. In Figure 8 above, ULN2003 IC was used to serve as switches. Control signals, from PC parallel port, are responsible for providing the required control action to open and close the switches at the appropriate times in order to spin the motor. Figure 9 below shows the complete circuit diagram of the stepper motor driver.

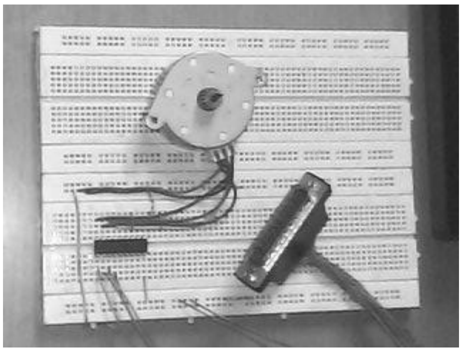


Figure 9: Stepper motor driving hardware

4.1.2. Temperature Sensor and ADC

The temperature sensor used in this work is the LM35D, see figure 10 below.

Figure 10: The temperature sensor LM35D

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature (a 10.0 mV/°C scale factor). The LM35D thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼ °C at room temperature and ±¾ °C over a full -55 to +150 °C temperature range.

The ADC0804 was used to convert the analog signal of the LM35D output into equivalent digital value. The ADC0804 is a CMOS 8-bit successive approximation A/D converter with TRI-STATE output latches directly driving the data bus. This ADC appears like memory location or I/O port to the PC and no interfacing logic is needed. Differential analog voltage inputs allow increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution. Figure 11 below shows the circuit diagram of the temperature sensor and the ADC connected to the PC through the parallel port.

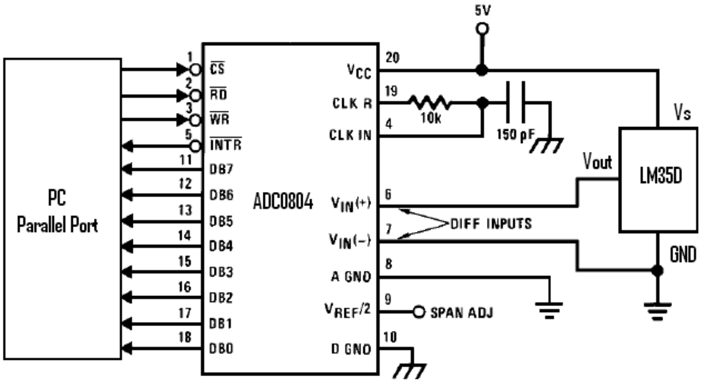
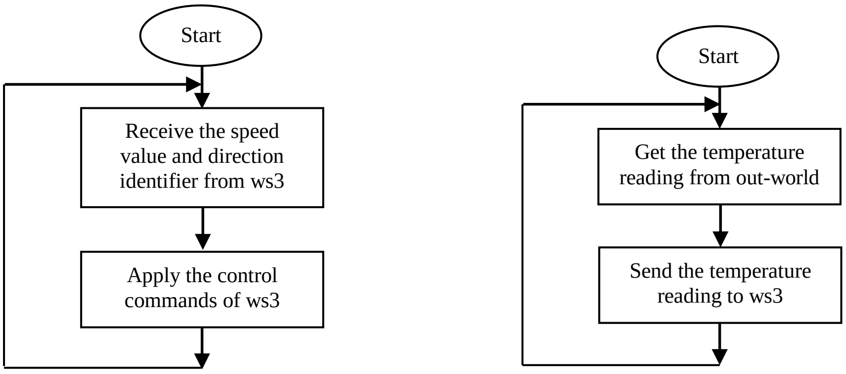


Figure 11: Circuit diagram of the temperature sensor with ADC.

5. System Software

The system software is consists of three parts or programs, each was written in Visual Basic, these are; the stepper motor driving program on (ws1), temperature monitoring program on (ws2) and the monitoring and control program on (ws3), figures 12 and 13 below show the flow chart of each program.



(a) **(b)**

Figure 12: (a) Flowchart of the stepper motor driving program on (ws1)
(b) Flowchart of the temperature monitoring program on (ws2)

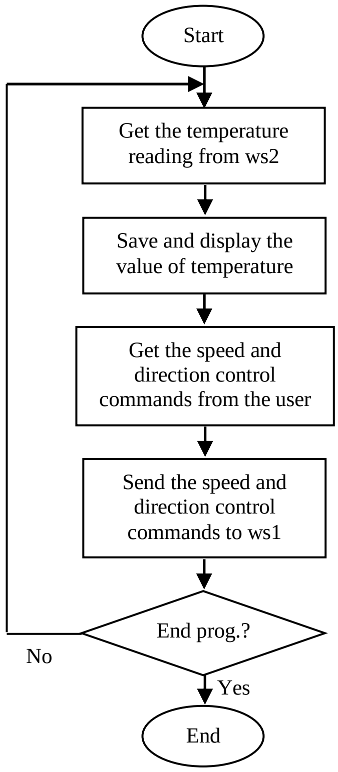


Figure 13: Flowchart of the monitoring and control program on (ws3)

5.1. Stepper Motor Driving Program on (ws1)

This program is written in visual basic and it is responsible to generate the required control signals to the stepper motor driving circuit in order to specify the speed and direction of rotation according to the commands that received from monitoring and controlling PC (ws3).

5.2. Temperature Monitoring Program on (ws2)

This program is also written in visual basic and its duty is to acquire the temperature value generated by the temperature sensor and the ADC and converting the digital inputted value into actual value and sending it to the monitoring and controlling PC (ws3).

5.3. Monitoring and Control Program on (ws3)

This is the main program in the system. It is the manager of the whole monitoring and controlling operations performed by this workstation. The commands required to control the stepper motor rotation on (ws1) and the commands required to acquire the measured value of temperature from (ws2) are generated by this program. The GUI of the program is shown in figure 14. From this Visual Basic GUI menu we can show that the user can control the speed and direction of stepper motor rotation from the (High Speed), (Low Speed), (CW Direction) and (CCW Direction) buttons. The temperature would be displayed in the chart area shown on the left of the GUI form. This program include the communication part that is responsible to manage the communication between this workstation and the other stations through connect/disconnect embedded procedures. The program also accepts user control commands and uses the TCP protocol to transfer these commands through the Ethernet.

All the programs above use Microsoft Winsock control 6.0. The Winsock control operates at the lowest level of all the Ethernet or Internet control, allowing client/server application to communicate using both Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

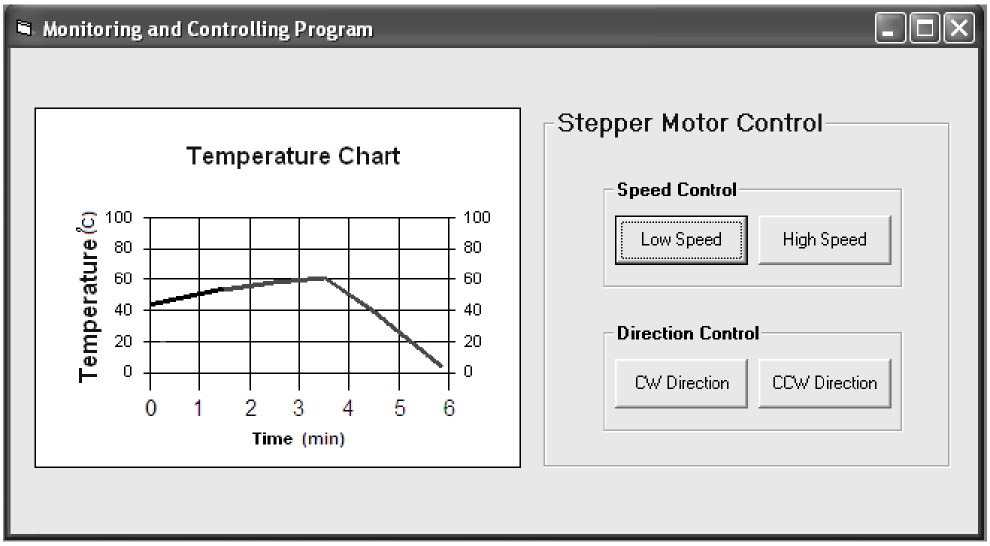


Figure 14: The GUI of the monitoring and controlling program

6. Results and Discussion

The monitoring and control system via Ethernet was designed and implemented and it was effectively used to control the stepper motor rotation on one workstation and monitoring the temperature variation on another station. Figure 14 above shows the temperature variation during 6 minutes (the temperature sensor was heated and then suddenly cooled during this period). Figure 15 below shows the stepper motor during its rotation in low speed and then in high speed mode. This experiment was carried out for Ethernet consisting of 3 workstations connected as shown in figure 7 above and then repeated for 10-workstation Ethernet, for two cases, using the hub and then using the switch.

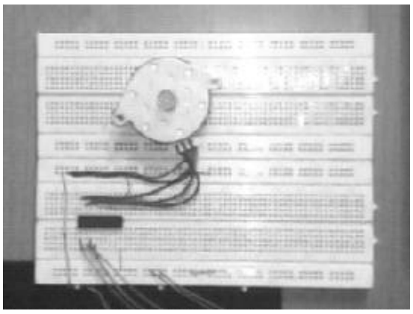


Figure 15: Stepper motor during rotation

From the results obtained after operating the monitoring and control system it is clear that the network traffic has a great influence on the system response (the time required to transfer the control command between the controlling PC (CONTROLLER) (ws3) and the stepper motor driving PC (ACTUATOR) (ws1)) from a side, and transfer the measured temperature between the temperature measuring PC (SENSOR) (ws2) and the controlling PC (CONTROLLER) (ws3) from another side.

Packet transport of connection establishment, data transfer and connection termination phases of Transport Control Protocol (TCP) may vary from frame to frame. Therefore, frame to frame delay profiles were taken for, sensor to controller and controller to actuator. In Figure 16, delay increases with increased network congestion. Due to absence of virtual circuits as in the case of Switch configuration in Figure 16, connection establishment and connection termination phases in Figure 17 takes longer time. Since communication takes place within the same collision domain, delay in the data transfer phase in the hub configuration is shorter than that of the Switch configuration.

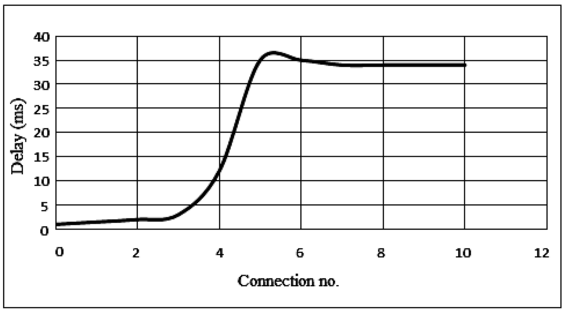


Figure 16: Sensor to controller delay with Switch

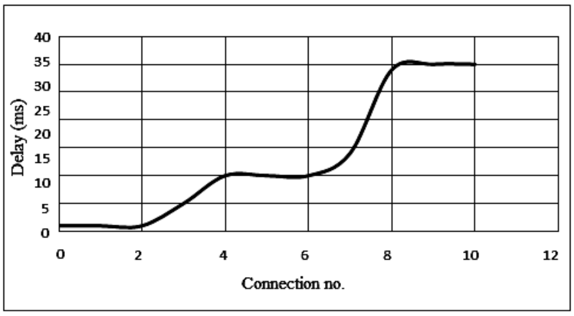


Figure 17: Sensor to controller delay with Hub

In Figures 18 and 19, controller to actuator needs more frames in data transfer phase than in sensor to controller data transfer. In Figure 19, the delay in data transfer phase increases in later TCP sessions degrading the performance due to missing samples.

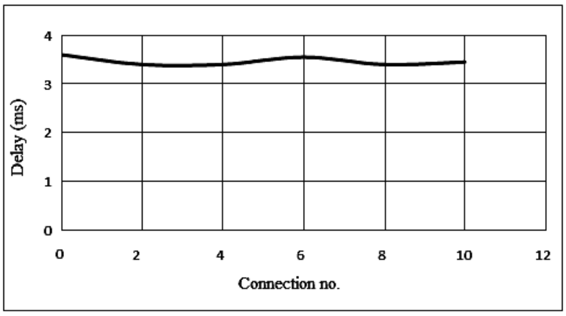


Figure 18: Controller to actuator delay with Switch

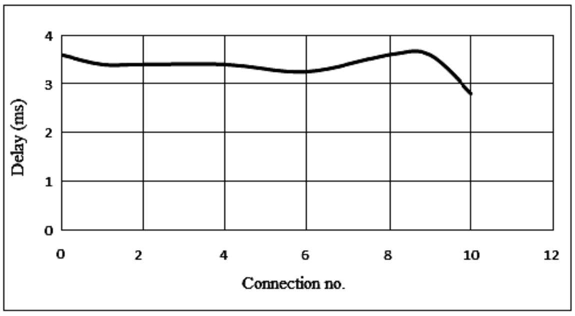


Figure 19: Controller to actuator delay with Hub

7. Conclusion

Ethernet, as defined in IEEE 802.3, is unsuitable for strict real time industrial applications because its communication is non-deterministic. This is due to the definition of its media access control (MAC) protocol, based on Carrier Sense Multiple Access/ Collision Detection (CSMA/CD). The implementation of the distributed real time monitoring and control system via Ethernet requires treatment solutions in both the software and hardware to overcome the drawbacks of using the Ethernet as a fieldbus network. During this paper the delay occurred due to connection establishment and connection termination phases of TCP sessions were measured and discussed and they were constant and their values were found depend on the configuration and the direction of traffic flow. The control delay and controller calculation time was evaluated off line using the system clock readings. Controller calculation time is negligible compared to control delay for low and high network traffic respectively. It was seen from the results that the packets still be delayed or even lost if one of the following scenarios appears:

- The total network load exceeds the switching capability of the switch engine. i.e. the switch is not able to handle full wire speed on each drop link.
- The output buffer capacity is not sufficient. i.e. the amount of packets sent to an output port exceeds the bandwidth of this port for a time period that is longer than the output buffer is able to handle. Thus, packets from several input ports compete for the same output port causing a non-deterministic buffering delay.

These two scenarios can be avoided by using the following Ethernet techniques:

- Back pressure: The switch can send a jam pattern simulating traffic on a port operating in half duplex mode if the amount of packets received on this port is more than the switch can handle.

- Flow control: The switch can send PAUSE packets according to IEEE802.3x on a port operating in full duplex mode if the amount of packets received on this port is more than the switch can handle.
- Priority: Ethernet packets that are identified as high priority packets are put in a high priority queue. Packets from a high priority queue are sent before the low priority packets. The low priority packets may still be lost. This is the most relevant technique with respect to optimal real-time properties for latency sensitive real-time data.

Finally it was seen that with Ethernet switches that offer both half and full duplex connectivity, Ethernet controller never will see any collision if full duplex connectivity is used.

8. References

[1]L. Samaranayake, S. Alahakoon," **Closed – loop Speed Control of a Brushless DC Motor via Ethernet**", Royal Institute of Technology, Stockholm, Sweden, 2006.

[2]L. Samaranayake, "**Condition Monitoring and Distributed Real-Time Control of Industrial Drive Systems via Ethernet**", Royal Institute of Technology, 2004.

[3]"**IEC 61850 Communication Networks and Systems in Substations**", Part 5: Communication Requirements for Functions and Device Models, Part 7-2: Basic Communication Structure for Substations and Feeder Equipment," 1999.

[4]T. Skeie, S. Johannessen, C. Brunner, "**ETHERNET in Substation Automation**", IEEE Control Systems Magazine, pp. 43-51, June 2002.

[5]P. Doyle, "**Introduction to Real-Time Ethernet 1**", University of Limerick, Ireland, 2004.

[6]ISO/IEC 15802-3: 1998, ANSI/IEEE Std 802.1D, 1998 Edition. Information technology – telecommunications and information exchange between systems - local and metropolitan area networks - common specifications. Part 3: Media Access Control (MAC) bridges.

[7]IEEE Std 802.1Q-1998 IEEE standards for local and metropolitan area networks Virtual bridged local area networks.

[8]Georges, J.-P.; Rondeau, E.; Divoux, T “**Evaluation of Switched Ethernet in an Industrial Context by Using the Network Calculus**”, Factory Communication Systems, 2002. 4th IEEE International Workshop on 28-30 Aug. 2002, pages 19-26.