IRAQI Academic Scientific Journals
العراقية المجلات الأكاديمية العلمية
Al-Kut University College Journal
ISSN (E) : 2616 - 7808 II ISSN (P) : 2414 - 7419
https://js.alkutcollege.edu.iq/
k.u.c.j.sci@alkutcollege.edu.iq
Special Issue - The 6th Scientific Conference of Al-Kut University College - 2-3 May 2023

# Implementation of a Wireless Sensor/Actuator Distributed Network for a Temperature Control Systems

Ali A. Abed [1] , Abduladhem A. Ali [2] , Nauman Aslam [3] , Ali F. Marhoon [4]

**Abstract**

In the last two decades, Wireless control systems, became an active research topic. This is due to its flexibility and reduction in implementation costs. However, these systems suffer from problems concerning the implementation issues. These problems arise due to the limitation in data rates and the considerable delay in the wireless network, and the ability to manage this network. To overcome this conflict, this paper presents a way to implement a wireless sensor/ actuator network that uses a wireless sensor network for the measurement phase. This network is driven by a TinyOS system; while a CAN field bus is employed for the actuator drive. The CAN field bus software is implemented by using C, while the code for the wireless sensor nodes is written in nesC language. The resulting distributed control system (DCS) is used to control the temperature of a water path. Velocity PID control algorithm was implemented to insure that the control system satisfies the bumpless transfer and also anti-windup concepts such manual/auto changeover does not affect the operation. simple GUI was also designed to monitor, set point changing, PID tuning, and control the switching between manual and auto operation. Real-time experimental results are obtained for the system to prove its operation.

**Keywords:** Wireless control systems, Sensor/actuator network, CAN field bus, Distributed control system, Real-time computer control

**Affiliation of Authors**

[1] Department of Computer Engineering, University of Basrah, Iraq, Basrah, 61004

[2] Department of laser engineering and optoelectronics, Alkut College University, Iraq, Kut, Wasit, 52001

[3] Department of Engineering Mathematics Internetworking, Dalhousie University, Canada, Halifax, NS B3H4R2

[4] Department of Electrical Engineering, University of Basrah, Iraq, Basrah, 61004

[1] aaad_bah@yahoo.com
[2] abduladhem.ali@alkutcollege.edu.iq
[3] naslam@dal.ca
[4] Ali_Marhoon2003@yahoo.com

[2] **Corresponding Author**

**Paper Info.**
**Published:** Aug. 2023

## تنفيذ شبكة موزعة للأجهزة الاستشعار والتشغيل اللاسلكية لأنظمة التحكم في درجة الحرارة

أ. م. د. علي أحمد عبد [1] ، أ. د. عبدالعظيم عبد الكريم علي [2] ، أ. د. نعمان أسلام [3] ، أ. د. علي فاضل مرهون [4]

**المستخلص**

في العقدين الأخيرين، أصبحت نظم التحكم اللاسلكية موضوعًا بحثيًا نشطًا. وذلك بسبب مرونتها وتقليل تكاليف التنفيذ. ومع ذلك، تعاني هذه النظم من مشكلات تتعلق بقضايا التنفيذ. تنشأ هذه المشكلات بسبب الحدود في معدلات البيانات والتأخير الكبير في الشبكة اللاسلكية، وقدرتها على إدارة هذه الشبكة. للتغلب على هذه المشاكل، يقدم هذا البحث طريقة لتنفيذ شبكة حساسات/محركات لاسلكية تستخدم شبكة حساسات لاسلكية لمرحلة القياس. تدفع هذه الشبكة بواسطة نظام TinyOS، بينما يتم استخدام حافلة ميدان CAN لتشغيل المحركات. يتم تنفيذ برنامج حافلة الميدان CAN باستخدام C، بينما يتم كتابة كود العقد اللاسلكية باستخدام لغة nesC. يتم استخدام النظام الموزع للتحكم (DCS) للتحكم في درجة حرارة مسار المياه. تم تنفيذ خوارزمية تحكم PID نوع السرعة لضمان أن النظام يفي بمفهوم نقل الصدمة بسلاسة وكذلك مفاهيم

**انتساب الباحثين**

[1] قسم هندسة الحاسبات، جامعة البصرة، العراق، البصرة، 61004

[2] قسم هندسة الليزر والإلكترونيات الضوئية، كلية الكوت الجامعة، العراق، واسط، 52001

[3] قسم الرياضيات الهندسية والشبكات، كندا، دلهاوس، NS B3H4R2

[4] قسم الهندسة الكهربائية، جامعة البصرة، العراق، البصرة، 61004

[1] aaad_bah@yahoo.com
[2] abduladhem.ali@alkutcollege.edu.iq
[3] naslam@dal.ca
[4] Ali_Marhoon2003@yahoo.com

المؤلف المراسل 2

معلومات البحث
تاريخ النشر: آب 2023

مكافحة التخمة بحيث لا يؤثر التحويل اليدوي/التلقائي على التشغيل. تم تصميم واجهة المستخدم الرسومية البسيطة أيضًا لمراقبة تغيير نقطة التشغيل وضبط PID والتحكم في التحويل بين التشغيل اليدوي والتلقائي. تم الحصول على نتائج تجريبية في الوقت الحقيقي للنظام لإثبات عمله.

**الكلمات المفتاحية :** أنظمة التحكم اللاسلكية، شبكة الاستشعار/التشغيل، حافلة حقل CAN، النظام الموزع للتحكم، التحكم الحاسوبي في الوقت الحقيقي

## Introduction

The use of (Distributed Control System) DCS is the most used hardware in industrial control if the process under control contains many continuous closed-loop controls. Uses benefits from the new development in this field and finds its flexibility is highly useful [1]. The basic principle of such a control system is that it contains a field bus connecting different sensors and actuators and processing elements. The field bus is generally a wired field bus communication operating in certain protocols. Using wireless networks in such a system can significantly reduce the DCS cost and installation time. Such improvements can add many advantages: wireless networking improves operation mobility (e.g. in robotics), and also improves installation speed and node deployment. A typical wireless network for distributed control consists of controllers, wireless sensors, and actuators [2]. Building a DCS system supported by a wireless network requires different design principles for both control algorithms implementation and communication networks. Conventional control techniques use accurate data acquisition with specific time and lossless transmission of these data to the processing elements[2]. Oppositely, wireless networks can suffer from packet loss in addition data can also have a random delay. Therefore, there should be a compromise between the control system requirements and communication performance to develop an optimal wireless DCS system. In industrial control, as the information about the process increases, the accuracy of controlled parameters is improved. Hence to achieve this requirement, more measuring devices have to be deployed. However, in wireless networks, as the number of measurement nodes increases, packet loss and congestion are increased. Hence, a compromise has to be done in a wireless sensor network. Wireless sensor/actuator networks (WSANs) are a new type o wireless sensor network (WSN)[3] WSANs tacks such problems into account. This paper presents a practical design of WSAN that can be used for the implementation of a DCS system to get a WDCS. The developed system is general and not based on a specific existing platform. The developed technique will depend on a Wireless Sensor Network (WSN) for data measurement and a field bus system with CAN (Controller Area Network) communication protocol for the actuation of the plant. A wireless Sensor Network (WSN), has several sensor nodes (also denoted as motes). These motes are used to measure the physical variables of the process under control. The use of such a wireless network will reduce labor and initial construction cost. In addition, it provides flexibility in location distance, improves reliability, and increases calability[4]. The gateway which is also called the base station is located in the control room, and it receives limited signals from remote motes that measure the plant variables. There should be a tradeoff between sampling rate and mote power requirements to improve battery lifetime [5]. Communication between nodes reduces the communication radius and power requirements compared with direct communication. Adding more nodes can be easily adopted either to add more measurements or to reduce the communication path. If a node is reset, it can instantly be connected to the network and hence measures the data after it reboots. This network is used to replace a 4-20mA analog wired system that links b the field devices (sensors and actuators) with the controllers. For the actuators, wired serial transmission can be used to reduce the number of data lines and hence, increases transmission distance. In this case, a field bus communication protocol is required and designed to meet a specific application.

In this paper, we will adopt the CAN protocol for this reason.

The rest of the paper is organized as follows: Section II describes a short review related to this work. Section III describes the CAN protocol and its elements. In section IV the hardware of the designed WSAN is presented. While Section V is specialized to outline the software needed for the overall wireless system. In section VI, the algorithm of the velocity PID controller with its two modes of operation is presented. Section VII concerns the practical implementation of the model. In section VIII, the real-time experimental results are displayed.

## Related Works

Many publications are done for WSN applications in environmental, habitat, greenhouse, and control applications. WSN is used only for monitoring. In 2006, Jose A.B. built a monitoring system that gathers physical quantities from different types of sensors for an environmental data acquisition system [6]. In 2010, Razvan M.E. described a WSN which was used for environmental monitoring. He gave a novel mechanism to efficiently wake up a network. also presented a new data retrieval system. He explored a solution for the interference between WSNs using IEEE802.15.4 radios and WiFi networks in the 2.4 GHz bandwidth [7]. WSANs are a modification on WSN through adding the actuators network to the system. In 2004, Akyildiz I.F. and Kasimoglu I.H. presented the challenges that exist in WSAN for both communication and coordination [8]. In 2006, Li presented a prototyped system for light monitoring and control by using WSAN [9]. Abed A. et al. In 2010 described how WSAN can be used for distributed temperature monitoring [10]. Ali J. and Ali A. implemented hierarchical distributed WSNs for vibration monitoring [1].

Concerning control systems, little work exists. In 2007, Feng X. et al. proposed WSANs for mobile control applications. They discussed the relationship between link quality and packet loss [3]. As a part of a supervisory system, Hazem M.S. 2010 developed a coordination agent control by using a wireless network [12].

## Wireless Network Architecture

Generally, a WSN has several sensor nodes used for the measurements of the physical variables, routing nodes that are responsible for routing the data towards the data sink node. The data sink(s) are used to disseminate queries or commands that control the sensor nodes and receive the measured quantities from the sensor node through routing nodes. There are two types of transmission networks single-hop and multi-hop network architecture [13]. In multi-hop networks, data is transmitted to the sink through one or more of the routing nodes. This principle reduces the consumed power in the nodes. Hence, multi-hop short-distance transmission is more favorable. The architecture of the multi-hop can be implemented with two techniques Hierarchical and Flat Architecture [13,11]. In the hierarchical architecture, the sensor nodes are grouped as hierarchical clusters, The cluster members route their data toward the cluster head which relays the data to the sink. A node with lower energy is used for measuring and sending measured data to the cluster head at a short distance, The higher energy node is selected as a cluster head. Data aggregation is performed at cluster heads to reduce the communication with the sink to improve the energy efficiency of the overall network. In our work, this architecture is used with sensor nodes (MDA100) and a gateway station. The sensor node is placed at the lowest level of the hierarchy. The implemented network architecture is illustrated in Fig. (1).

For network expansion, other sensor nodes can be added to the network designed above as shown in Fig. (2). The router node can be used to increase the coverage distance and avoid obstacles that may be faced in the field. MDA300 is a data acquisition board suitable for analog input, and digital I/O.

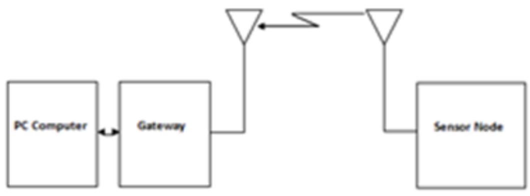

Fig. (1):Network architecture designed for data acquisition

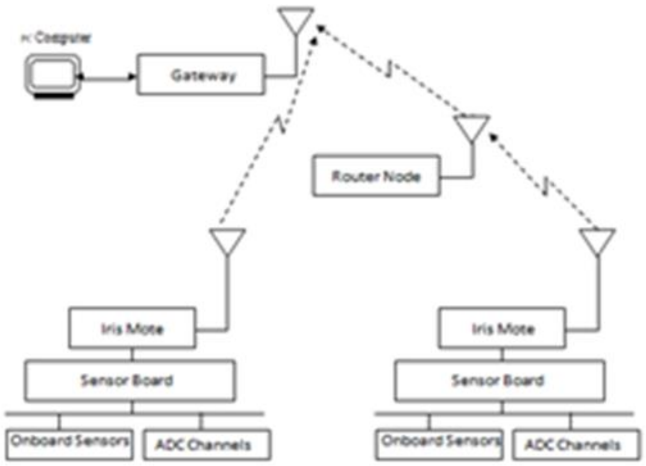

Fig. (2):Distributed controllers communication

**CAN Field Bus System**

The CAN system bus has a 2 wires serial field bus with multiple nodes (slaves) that can communicate with each other. The group delay in the CAN bus is a function of the bus length. For example, the 1 Mb/s is satisfied with a 40 m bus length, while 50Kb/s with up to 1km length. CAN protocol operates with two addressing methods: 11-bit and 29-bits. When 11-bit addressing is used; it is referred to as CAN2.0A, this addressing, is used for small systems with 2048 IDs. While, 29-bit addressing is used for larger systems and is referred to as CAN2.0B different IDs. Practically, the nodes used are also limited by the CAN transceiver used for the implementation. To improve noise immunity, the CAN field bus uses a differential signal. Fig. (3) shows typical master-slave industrial CAN field bus implementation.

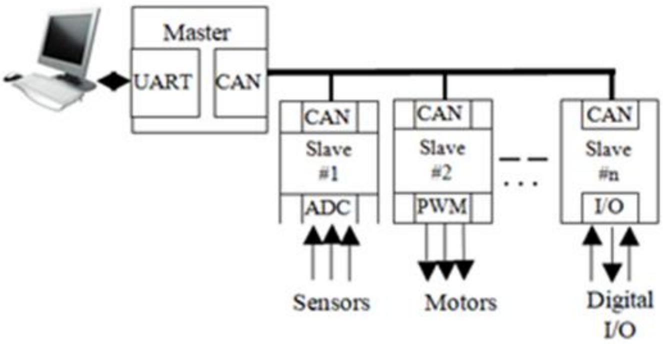


Fig. (3):The overall CAN field bus system

**Hardware Requirements**

In this section, a more detailed description of the major components of the CAN field bus and the wireless sensor nodes and network used in this implementation.

A. The (T89C51CC01) CAN Microcontroller

T89C51CC01 is an 8-bit microcontroller and the first of the CANry family manufactured for the CAN field bus network. It operates up to 20 MHz and uses an external clock. It has 34 CAN special function registers (SFRs) and a total of 99 SFRs. The baud rate can be adjusted by using special internal registers that can be programmed for this purpose [14].

B. The (MCP2551) High-Speed CAN Transceiver

MCP2551 is an interface device that connects the CAN controller with the physical CAN bus. Each node in the bus must be provided with one CAN transceiver. The function of this device is to generate at high speed a differential signal (CANL and CANH) from the TTL level produced by the CAN controllers. This transceiver can be used with up to 112 Node [14].

C. The Sensor Board

The board used in this paper contains four basic elements. They are: The microcontroller which is ATmega 1281with 8-bit ADC output as part of the microcontroller chip. The second part is the sensing device which is connected to the ADC channel. The RF transceiver is the third component that converts the microcontroller TTL output into an RF signal. And the fourth component is the power unit, which is a power supply circuit or batteries.

In this work, the sensor board used for data acquisition is named MDA100. The range of the input analog voltage applied to the holes of the ADC channels is 0~2.5V. The sensor board is attached to an IRIS RF mote such that the sensed signal is transmitted via wireless transmission[15]. The mote presented in this paper has the schematic diagram shown in Fig.(4). It uses 2.4 GHz with IEEE 802.15.4 standards. A tiny wireless measurement system is developed for this embedded sensor network and applied for data acquisition. The ATMega1281 microcontroller of this mote could satisfy a baud rate of 250 Kbit/s. This mote is mounted to the MDA100 sensor board via a 51-pin male connector. For outdoor applications, the mote has line-of-sight ranges that can reach up to 500 m without amplification between two nodes. For indoor applications, it can reach about 50m.

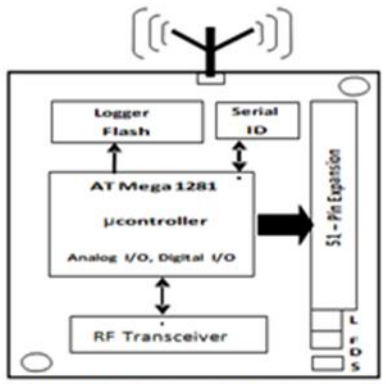

Fig. (4): The schematic diagram of the mote

D. The Wireless Gateway

The MIB520 USB (Universal Serial Bus) Interface Board is available to be connected to the mote which is used for programming the mote by computer. The mote can either be programmed as a sensing mote or as a gateway. The power of this device is supplied by the computer through the USB port. Therefore, this node is considered a high-energy node. It provides a 57.6 Kbit/s baud rate.

**Software Requirements**

A. TinyOSTinyOS

is a lightweight small operating system (OS) designed specifically for small microcontrollers such as ATMega 1281 used in wireless sensor nodes to make the WSN building easier. Its services lie in the domain of sensing, communication, storage, and timers. Network Embedded System C (nesC) is used to write the TinyOS and its applications. are written in nesC language is a C dialect with some extra features. TinyOS contains several APIs (Application Programming Interfaces) used for some functionality such as reading sensors, sending packets, and responding to events. In this work, TinyOS 1.1.10 is used for the embedded software edition, debugging, and motes programming (i.e. downloading the application software into the sensor node.

B. Xserve

Xserve program is an interface user program that is responsible for receiving data from the WSN and then displaying this data in the command window. Terminal Interface is used by the users to interact with Xserve. The Xserve produces pre-configured processing modules which are used for writing data in the file or displaying data on a screen, writing data in a database, etc. In this work, Xserve is used to save the measured data which is the sensor output in a Comma Separated Variable (.csv) Excel file. This makes the Xserve function as a producer for the items (data) and puts the data into a buffer for consumption by a consumer. The consumer is a separate program that functions to read data produced in the buffer and displays this data on the screen as will be seen later. To log data coming from the sensor node into a .csv data file, the following function is written :

$xserve –device=com7 >filename.csv

“filename” is the name of the file used for sensor data storage. All the produced data is automatically saved to a .csv file. COM7 referred to the port number of the computer.

C. MATLAB and Xserve

MATLAB has several commands that can read .csv files such as “csvread”. Each data in the record of the file should contain only numeric values. There are two ways to avoid the non-numeric data and read only the numeric values: By rearranging the .csv file by MATLAB itself to remove all non-numeric data; by using the suitable commands of the Xserve program to remove all the extra non-numeric data and leave only the comma separated numerical values in the .csv file. The second way is chosen in obtaining the suitable format of the .csv file to leave MATLAB running faster. We found out that the suitable commands of the Xserve used to obtain an arranged data file are –xc and –q as written in the following :

$xserve –device=com7 –xc –q >filename.csv

The arranged .csv data Excel file is shown in Fig.(5).

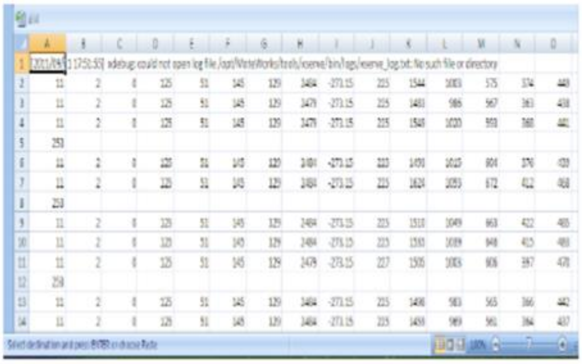

**Fig. (5): The *.csv* data file after rearranging**

The desired data is reached by making MATLAB “csvread” command start reading the two values at (row=1, column=10) and (row=2, column=10) without problems as follows:

x=1;

[v]=csvread

('d:/filename.csv',x,10,[x,10,x+1,10 ]);

After that, the “csvread” command continues in reading consecutive values of the sensor reading available at one of the ADC channels. It clears that the consumer of the sensor data is the m-file that reads the .csv file data and displays it on a display screen. This form of programming is a producer-consumer problem. Such a problem requires a technique of synchronization between producer and consumer of such problems that appear in such a process can be avoided. As an example, a buffer is empty, this problem occurs when the reader (consumer) is much faster than the writer (producer). The presented solution here for satisfying the synchronization is the “busy waiting” as explained in the following codes that will be repeated every sampling time :

```
x=1;
m=csvread('d:/filename.csv',1,0);
j=size(m,1);  %Reading of the csv file
                %size (No. of rows)
while (x<j)  %condition for satisfying
               %synchronization %between
     %producer(Xserve) and consumer (mfile)
 [v]=csvread
('d:/filename.csv',x,10,[x,10,x+1,10]);
         |
         |
  x=x+2; %increment of .csv file' reading
              %index
end
```

By this code, one can be sure that the consumer is prevented from reading the buffer, while empty or still not producing a new item.

**Velocity PID Control System**

Practically, most controllers need to run in two modes: manual and automatic as shown in Fig. (6).. When there are changes in modes, it is important to take care of the transients that occur during switching between these modes. This problem is denoted as the bumpless transfer.

A. Bumpless Transfer

One should be sure that the switching between manual operation mode and automatic operation mode and vice versa has to be smooth. The manual/auto-switching is shown in Fig. (6). Conventional PID controllers suffer from this problem. Special techniques have to be added to solve this problem.

In this paper, a modified PID controller, named velocity algorithm is employed to solve this problem.

The velocity algorithm calculates the differential value of the process-manipulated variable at every sample instead of calculating the absolute value of this variable The differential value of the output (m(k)) can be calculated as [16]:

m(k)= K1 e(k)+K2 e(k-1)+K3 e(k-2) -------(1)

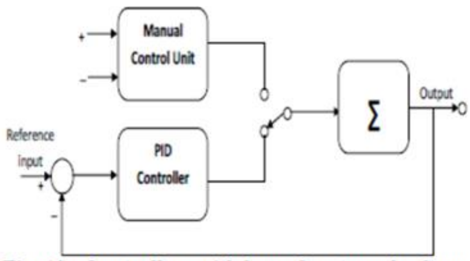


**Fig. (6): Controllers with bumpless transfer from manual to automatic mode**

Where e(k) is the error between the command(setpoint) and measured values, q(r) is the set of the commands, and K1, K2, and K3 are constants to be calculated [16]. This algorithm always satisfies "bumpless transfer" because it only adds the increment to the old value of the controlled variable m(k).

B. Saturation (Integral Windup)& Anti-Windup

When the error is high, the integrator of the PID controller will wind up, especially when the manipulated variable reaches its maximum limit (such as the valve being fully closed or fully open). When the actual error is reduced; the PID controller will continue to produce high output and takes a large time to go back to its normal state This effect is named integrator windup. many methods exist for integral windup avoidance such as the velocity algorithm previously mentioned in which the integral action is calculated though adding the increment to the previous value of the actual manipulated variable value.

C. Timing

An important thing about any real-time software is that they have to run continuously (24/7/365). So, the natural structural element is the infinite loop. Also, the sampling required by the control algorithm should be fixed and synchronized to satisfy the process under control requirements. Hence, we need some form of timing synchronization. This paper used the real-time clock technique for synchronization. The following code shows this method.

```
tic
ttt=toc;
while (ttt~=Ts)
    tt=toc; ttt=fix(tt);
end
```

Where "tic-toc" is the function used in MATLAB. "tic" begins the ticking while "toc" reads the elapsed time. This code segment runs at the beginning of each control cycle and represents a "busy waiting" for a constant time of Ts (sampling time).

## Practical Implementation and Real-Time Results

The process chosen as a case study is the temperature control for a "water bath". Fig.(7) presents the structure of the designed system The master and slave are shown in Figs. (8,9) respectively. The velocity PID is designed and empirically tuned to get Kp=2.49, Ki=.000005, and Kd=330. The real-time response for the velocity PID is shown in Fig. (10).

For manual mode, the adjustment of the controlling value is done with a slider as shown in Fig. (11), which is a simple GUI.

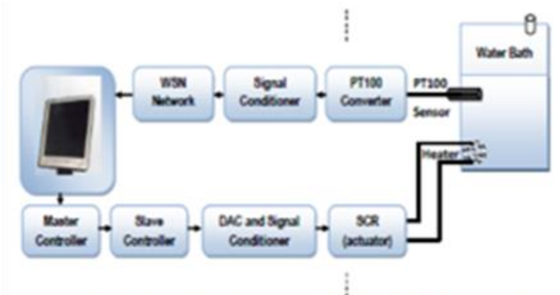


Fig. (7): Schematic diagram of the control rig of the WSAN

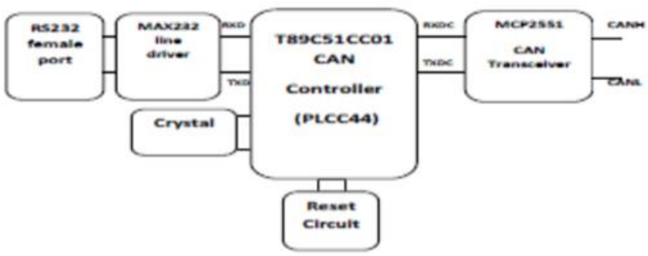


Fig. (8): Master Controller Schematic Diagram

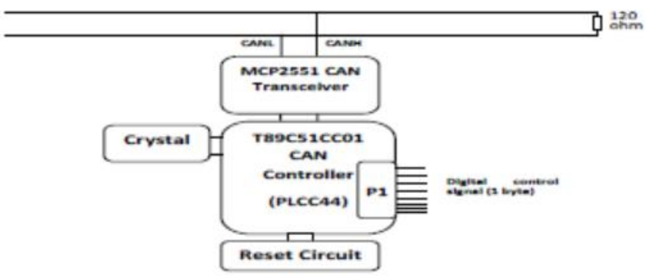


Fig. (9): Slave Controller Schematic Diagram

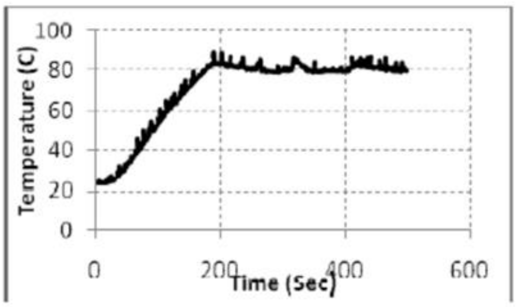


Fig. (10): The response of the wireless real-time velocity PID control system (set point=80 °C)

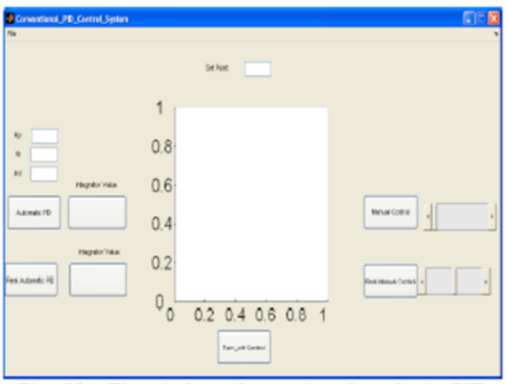

Fig. (11): The window of operating the velocity PID wireless distributed control system

## Conclusions

In this paper, a practical implementation of a WSAN control system suitable for use in the design of a WDCS is presented. All the required hardware is listed with some description. A simple GUI window is designed for this wireless control system to satisfy monitoring, set point changing, PID tuning, and control. The real-time results proved that this system has succeeded in working normally.

## References

[1]Basu, SPlant Intelligent Automation and Digital Transformation: Volume I: Process and Factory Automation. , Elsevier 2023.

[2]Xiangheng L., and Andrea G., "Wireless network design for distributed control", Stanford University, USA, 2004.

[3]Feng X., et al, "Wireless sensor/actuator network design for mobile control applications", Sensors, vol.7, no.10, 2007, pp.2157-2173.

[4]Paul W., et al, "Condition monitoring in end-milling using wireless sensor networks (WSNs)", Transactions of NAMRI/SME, Vol.36, 2008,pp.177-183.

[5]Giri B., "Guidelines for WSN design and deployment", Crossbow Technology, Inc.

[6]Jose A., "Wireless sensor networks: Development of an environment data acquisition prototype system and graphical user interface", M.Sc. Thesis, Faculty of California Polytechnic State University, USA, 2006.

[7]Razvan M., "Low-power wireless sensor networks for environmental monitoring", PhD Thesis, John Hopkins University, Baltimore, Maryland, 2010.

[8]Akyildiz I., and Kasimoglu I., " Wireless sensor and actor networks: research challenges" Ad Hoc Net. 2004.

[9]Li S., "Wireless sensor actuator network for light monitoring and control application" proceeding 3rd IEEE Consumer Communications and Networking Conference, Lass Vegas 2006, pp.974-978.

[10]A. A. Abed, A. A. Ali and N. Aslam, "Building an HMI and demo application of WSN-based industrial control systems," 2010 1st International Conference on Energy, Power and Control (EPC-IQ), Basrah, Iraq, 2010, pp. 302-306.

[11]J. Ali and A. A. Ali, "Design and implementation of low-power hierarchical wireless sensors network," 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), Baghdad, Iraq, 2017, pp. 258-262, doi: 10.1109/NTICT.2017.7976112.

[12] Hazem M., " Wireless networked control system coordination agent", M.Sc. Thesis, University of New Brunswick, Canada, 2010.

[13]Jun Z., Abbas J., "Wireless sensor networks: A networking perspective" Published by John Wiley & Sons, Inc., New Jersey and Published simultaneously in Canada, 2009.

[14]L.M. Pinho and F. Vasques," Reliable Real-time Communication in CAN Networks", IEEE Transaction on Computers, Vol. 52, No.12, 2003, pp.1594-1607.

[15] MTS/MDA sensor board user's manual", Crossbow Technology, Inc., 2006.

[16]Stuart B., "Real-time computer control: An introduction",1st Edition, Prentice Hall International (UK) Ltd., 1988.