ICIT 2015 The 7th International Conference on Information Technology
doi:10.15849/icit.2015.0086 © ICIT 2015 (http://icit.zuj.edu.jo/ICIT15)

Internet-based Troubleshooting and Monitoring System of Industrial Robots

Md Mozasser Rahman[1]
Sulleha Bt Parnin[2]
Department of Mechatronics Engineering
International Islamic University Malaysia (IIUM)
Kuala Lumpur, Malaysia
[1]mozasser@iium.edu.my, [2]sullehaparnin@gmail.com

Abstract—**This paper presents a prototype of an industrial robot monitoring system that is remotely control via internet. The use of industrial robot is increasing in small to medium enterprise (SME's) as low-cost robots are available in market. Productivity can be increase by incorporating robots. The limitations of maintenance and troubleshooting personnel to monitor the robot will always exist because they are not always being in the robot operation area. Thus, a prototype of a monitoring system was developed and tested in the Robotics laboratory of International Islamic University Malaysia, using Denso robots. The remote monitoring system can monitor the robot condition by a real time video streaming using internet. Arduino microcontroller was used to send the signal to the robot from a computer. This open source software fit the requirement of this project which enable the development to be completed with a low cost budget. The user interface program was created using the Guided User Interface which enables the programmer to monitor the robot from a remote area by using TeamViewer application. A USB camera was also used to see the real time video of the robot. All the subsystem was successfully integrated by Define, Data Gathering and Analysis, Develop, Testing and Improve to accomplish the project goals and at the same time becoming a starting point for further development.**

Keywords—Industrial Robots; troubleshooting; monitoring; Internet-based; SME, Lowcost.

I. INTRODUCTION

The productivity of Small and Medium-sized Enterprise (SME) can be increased by using robots. Almost all of the big industrial robot makers have or are working on lightweight and human-friendly arms, but none are offering them at low cost, or with user-friendly training, or the plug and play and safety features. There are other robot startups in the SME marketplace – and on the horizon – but none are as far along in their development and low cost as **Rethink Robotics** and **Universal Robots** [1]. Some of these SME need only one or two robots for the automation process. It is not feasible for them to recruit a programmer or another skilled technician for troubleshooting those robots. There are after sales service provided by the suppliers but down time will be high. Thus, a prototype of a monitoring system was developed and tested in the IIUM Robotics laboratory by using the Denso robots. The remote monitoring system can monitor the robot condition by a real time video streaming using internet with a low cost budget. Arduino microcontroller was used to send the signal to the robot from a computer.

Marin et. al. developed user interface program for remote programming the robot via internet. Programming language is JAVA [2]. Rosado-Muñoz also proposed a web-based remote laboratory for the programming of the industrial robot. A webcam use to provide a video of the robot movement. Both are for educational purposes [3]. 3D modeling of a robot for physical assembly using internet application server for image processing has been developed by Wang et al. [4]. Their experimental result shows that the system is feasible to meet industrial assembly by producing a shorter image processing time.

Interest in the design of Internet-based telelaboratories is increasing enormously, and this technique is still very new. Most of the systems are cost oriented and need a special training [5-6]. Only a low cost and user-friendly system provide the realistic solution for SME's. Therefore an user interface program was created using the GUI which enable the service provider to monitor the robot from a remote area by using TeamViewer application.

II. SYSTEM OVERVIEW

A six-axis articulated robot as shown in Fig. 1 was used as a target robot. The proposed system consists of four subsystems which are:-

Subsystem 1: Robot controller and server PC interfacing.
Subsystem 2: GUI for communication.
Subsystem 3: Camera monitoring.
Subsystem 4: Interfacing client and server computers.

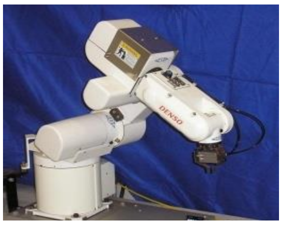


Fig. 1. Denso robot, used for the prototype.

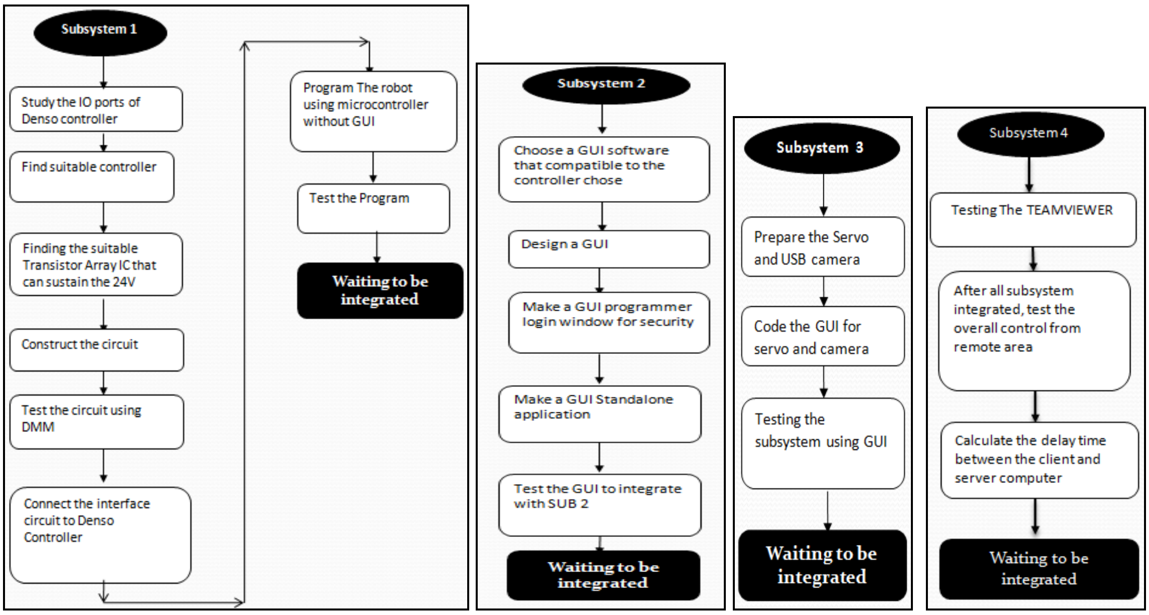

Subsystem 1
Study the IO ports of Denso controller
Find suitable controller
Finding the suitable Transistor Array IC that can sustain the 24V
Construct the circuit
Test the circuit using DMM
Connect the interface circuit to Denso Controller
Program The robot using microcontroller without GUI
Test the Program
Waiting to be integrated
Subsystem 2
Choose a GUI software that compatible to the controller chose
Design a GUI
Make a GUI programmer login window for security
Make a GUI Standalone application
Test the GUI to integrate with SUB 2
Waiting to be integrated
Subsystem 3
Prepare the Servo and USB camera
Code the GUI for servo and camera
Testing the subsystem using GUI
Waiting to be integrated
Subsystem 4
Testing The TEAMVIEWER
After all subsystem integrated, test the overall control from remote area
Calculate the delay time between the client and server computer
Waiting to be integrated

Fig. 2. Subsystem development flowchart

All the systems were developed simultaneusly with priority given to subsystem 1 and 2. The subsystem development flowchart is prensented in Fig. 2. Robot controller and server PC interfacing dominate the system.

Fig. 3 shows the position of the subsystem in the main system. All subsystems are mutually dependent. The arrow in the figure indicates the relationship of signal transferred.

After indetifing the function of the Denso controller i.e to the IO channel of the controller. This input channel is the one that is connected to the digital pin of the Arduino. There are a total of 24 CN5 pins that successfully been connected to the Arduino. By using the GUI, the signal status (high or low) of these pins can be controlled.

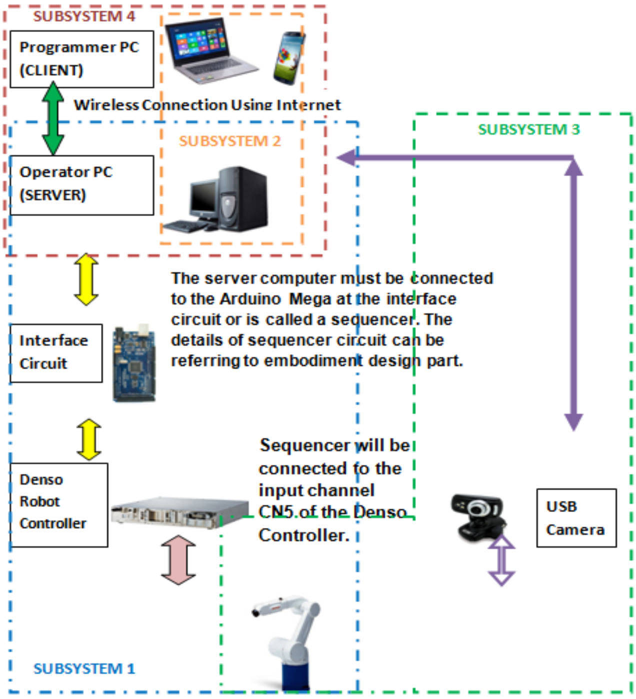

SUBSYSTEM 4
Programmer PC (CLIENT)
Wireless Connection Using Internet
SUBSYSTEM 2
SUBSYSTEM 3
Operator PC (SERVER)
The server computer must be connected to the Arduino Mega at the interface circuit or is called a sequencer. The details of sequencer circuit can be referring to embodiment design part.
Interface Circuit
Denso Robot Controller
Sequencer will be connected to the input channel CN5 of the Denso Controller.
USB Camera
SUBSYSTEM 1

Fig. 3. System Overview.

III. Working Principle

In every system, working principle describe the rule that enable the system to give a desired output. In this case, the input is the command from the programmer that give output in terms of robot motions, sensor data, robot image and etc. This input and output relation is shown by the system of work principle in Fig. 4. It applies that, programmer to sit at the client side and operator/supervisor will be at the server side. This system is powered by Matlab 2013-a, Arduino 1.0.6, and Teamviewer 10.

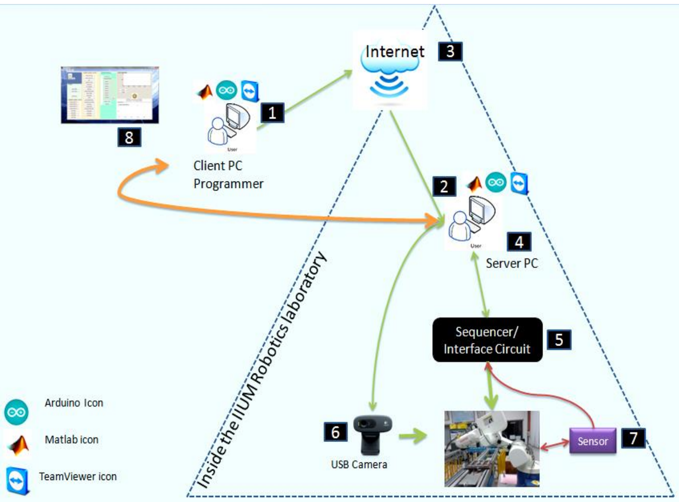


Fig. 4. Working principle of the system.

According to Fig. 4, the triangle in dashed refers to the testing area which is the robotics laboratory. The numbers in the figure can be described by the following:-

1. Programmer sign-in to TeamViewer account. Programmer will then need to connect to the server PC to enable the access to the robot. Programmer will login to the system and change the mode to external mode of operation. It is required that the Client computer (programmer PC) to have Arduino IDE and Matlab software.

2. The server computer must be turned on all the time. TeamViewer applications must be open to enable the access form the client PC. Server computer need to have Arduino IDE and Matlab software as well. The USB of the interface circuit and camera must be connected to the USB ports of the server PC. This is to ensure the direct signal transfer from the GUI to the circuit. An Arduino code to make it as server is uploaded during the subsystem 1 development. Therefore, Arduino code was written inside the Matlab code itself as the Arduino is now become a server that can only received a data from Matlab-GUI through serial port.

3. TeamViewer will take some time to verify the IP address of the server computer. Once the access is successful, programmer can view the server PC as if they were sitting in front of it.

4. Operator or supervisor that is in the robot operational area still can get control over the robot since the mode is INTERNAL. Once the programmer runs the program sequential in the monitoring window, operator/supervisor will not be able to run the robot by teaching pendant. It indicates the mode change from INTERNAL to EXTERNAL. Meanwhile, at the programmer side, they can start to open the Login window of the system and start to take over the robot operation using the user interface monitoring window.

5. Any command from the programmer will be received by interface or sequencer circuit.

6. The motion of the USB camera can be controlled by adjusting the servo position either to the left or to the right. Those two positions will view two different robots. The mechanical part for the camera was design to be able to monitor two robots. A high definition USB camera was used to ensure the quality of the robot video and images. This camera monitoring system covers subsystem 2 and 3.

7. The additional feature of the system is to put a sensor that can detect the distance of the object (for the case that robot was design for a pick and place operation with a conveyer system). In this project, a SHARP IR obstacle detection sensor was used with the capabilities to detect object at most 80cm.

8. Data such as the status of the IO pins, sensors value is accessible through Matlab-GUI with some delays in feedback. Delays can be depending from the speed of the internet connection of both server and programmer side and any other factor such as the computer CPU performances.

The principle of work in this project cannot be accomplish without knowing the interface circuit as it is the only medium to connect the computer to the Denso robot controller. Fig. 5 shows the connection done on the interface circuit. The main components of the circuit are the Arduino Mega and transistor array ULN 2803. According to Fig. 5, three transistor arrays were used to drive the signal form the GUI to the Denso robot controller. It enables the signal to be passing with a minimal input current. The advantage on using transistor array is it can simplify the circuit complexity. It composed of multiple transistors in one package and helps to improve mounting density and reduce the board population costs associated with the use of discrete transistors.

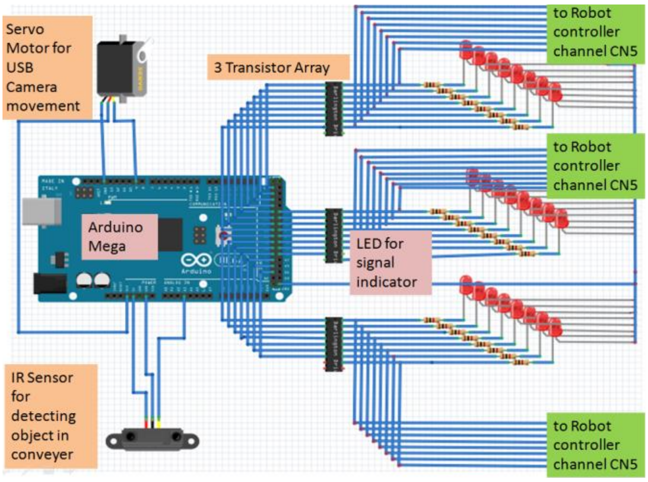

Servo Motor for USB Camera movement
3 Transistor Array
to Robot controller channel CN5
to Robot controller channel CN5
Arduino Mega
LED for signal indicator
IR Sensor for detecting object in conveyer
to Robot controller channel CN5

Fig. 5. Circuit connection for one robot attachment.

IV. Result

The following steps and its respective figure are the result of integrating the whole system.

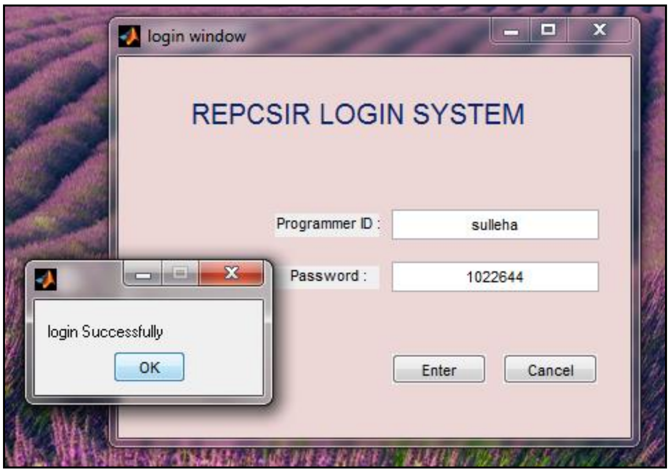
Preparation at the server PC side before the system fully tested:

i. Power on the robot. Robot power on at this time indicates the INTERNAL mode of operation.

ii. Open the Arduino>examples>pde>Adioes. Upload the Adioes code to the Arduino board. This is to be done only once to make the board as the server. All the instructions for the board input and output pins was done in GUI-Matlab editor file.

iii. Open the Matlab. Matlab should not be open directly from the main icon. It must be open through the folder of ‘MatDuino’.

Meanwhile, preparation at the programmer side is for the programmer to sign in to the TeamViewer account and connect to the server PC. After the access successful, programmer can initialize the system by running the Matlab GUI name ‘Login’. Upon running the login, provide the details need in login window such as the programmer ID and password as in Fig. 6. If login is successful, the main monitoring window will appear. If else, user cannot be allowed to enter the main monitoring window. It works like the security for the system from the stranger.


login window

REPCSIR LOGIN SYSTEM

Programmer ID : sulleha

Password : 1022644

login Successfully

Fig. 6. Screenshot of login page

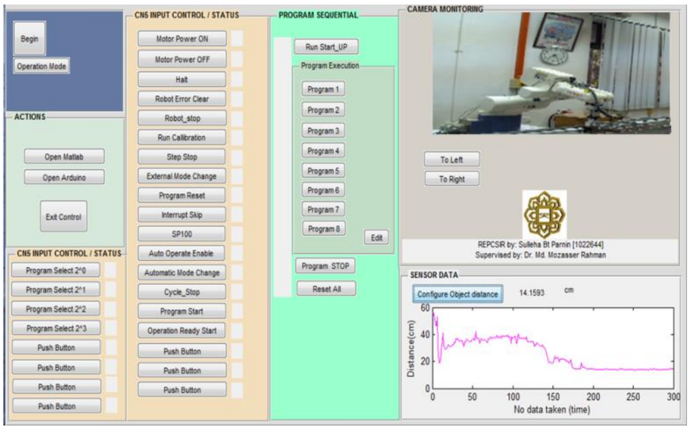


Fig. 7. GUI window for incomplete subsystem integration (only the sensor is tested)

Monitoring window with the name ‘c(at a moment after the login is successful. on the robot can be monitored from the remote location.

To begin, click on the pushbutton name ‘Begin’ to allow control of the robot. The Mode of operation must be in the ‘INTERNAL’ mode. This is because the startup operation is still not executed. Note that when the ‘Startup’ button is pushed, operation mode will automatically change to ‘EXTERNAL’. Fig. 7 shows the feature of the monitoring windows.

Monitoring window was tested and is improved in terms of programming of the speed of the camera and the status of process in the program sequential panel. Fig. 8 shows the ‘EXTERNAL’ mode of operation for the controlled robot while the camera was monitoring another robot operation.

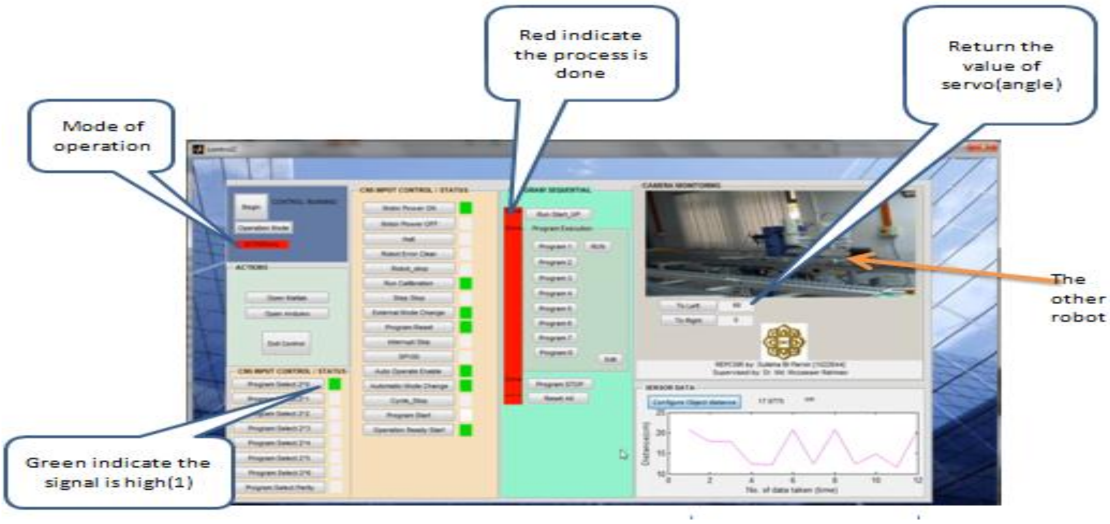

Red indicate the process is done
Return the value of servo(angle)
Mode of operation
The other robot
Green indicate the signal is high(1)

Fig. 8. GUI after imp

Running in the External mode of operation will change the status of the CN5 start-up input pins of the Denso robot controller from low (white) to high (green). Green color on the status panel is also indicates the signal is active and being sent to the robot controller through the interface circuit. Startup operation caused the mode to be change to automatic, motors to turn on, and calibrated the robot position.

The startup step is shown in Fig. 9. Program can be run right after the startup process is finished which indicate by the 'EXTERNAL' in the operation mode. Simultaneously, the camera and sensor can be tested.

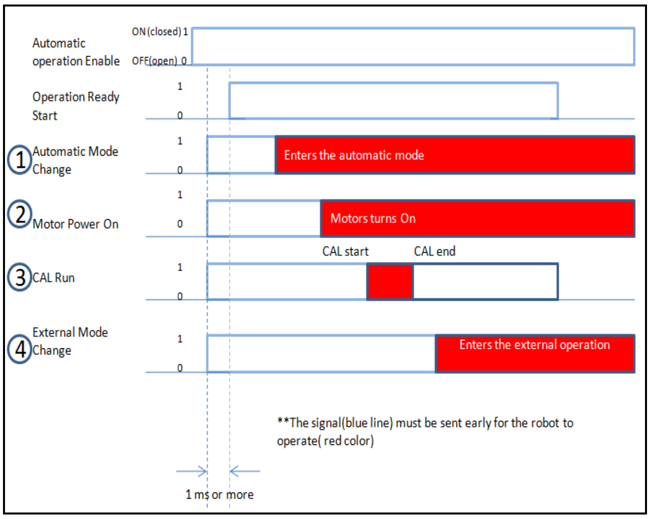

Automatic operation Enable
ON(closed) 1
OFF(open) 0
Operation Ready Start
1
0
1 Automatic Mode Change
Enters the automatic mode
2 Motor Power On
Motors turns On
CAL start
CAL end
3 CAL Run
4 External Mode Change
Enters the external operation
**The signal(blue line) must be sent early for the robot to operate(red color)
1 ms or more

Fig. 9. Robot Startup steps (Signal Operation diagram)

The Digital pins involves are 24(Auto Operate Enable), 48(Automatic Mode Change),43(Operation Ready Start), 44(Motor Power On),46(Run CAL), and 52(External Mode Change). Firstly, digital pin 24 need to be set high . This is followed by pin 48, 44, 46, and 52 and 43. This signal is an input to the Denso controller. It is shown by the blue line of the diagram in Fig. 9.

According to Fig. 9, the output response of the startup is shown by the red colour. Even though the signal is high at the same time, the reponse need to wait for the successfullness of the previous start up steps to initiated. This means that the external mode will not be achieved unless the calibrationof the motor is executed and completed. Internal mode light will be off to indicates that the external mode is turned on. Since the robot is in the external mode, no other instruction from the teach pendant will be executed. Therefore, only the instruction from the Arduino will be allow to operate the robot. In other word, this system will work only when the robot is in EXTERNAL mode of operation.

V. DISCUSSION

It is found that, the duration of startup time for the robot is less than that needed to startup using a teach pendant. The motor is turned on in much faster as the system eliminate the human command delay of pressing teach pendant buttons. If the time taken for a student to program the robot startup operation is compared with the time taken for the system startup operation, the system will approximately faster at a rate of two times the teach pendant. Testing the overall system require about a maximum of 10 minutes just to connect the MatlabIO to the serial port of Arduino. The attempt for Matlab to connect to Arduino is always fail required a special line of code to be executed on the Matlab command window which will delete the currentport and force the release at the serial connection. The following code was used:

i. To delete the current MATLAB serial connection on COM26 : (instrfind({'Port'},{'COM26'}));

ii. To delete all MATLAB serial connections : delete (instrfind('Type', 'serial')); .

While the overall system is tested, the status of each pin of the input pin CN5 can be checked by the red color to indicated the signal is being sent as 'HIGH' or '1' to the particular pin. These features enable the programmer to detect the problem whether any of the pins were caused an error to the robot operation. Programmer also may change the mode of operation from the 'EXTERNAL' operation to the 'INTERNAL' operation by pressing the pushbutton name 'Reset All'. Then all the digital output pins will become low as there were no signals being allowed to pass the sequencer circuit.

Through the CN5 cable of the Denso robot controller, robot fault can be cleared automatically without using the teach pendant in the EXTERNAL mode of operation. The 'High' combination of signal of 'robot error clear (CN5-18) and 'Operation Ready Start (CN5-23) will cause the error of the robot to be cleared. CN5-23 must be set into high at least 1 milliseconds to allow the robot to prepared for the motion and to prevent errors.

VI. CONCLUSION

An Internet based Troubleshooting and monitoring system of industrial robot was successfully developing. It is achieved by design and implementations of the interface circuit. Result shows that signal send by Arduino can be received by the Denso Controller perfectly similar to the teach pendant. The server computer also can be accessible by the client computer by using an internet connection although the quality of the controlled windows is not good enough. Finally, the programmer that is in remote area is able to see the robot status by running some startup operation on the robot. This system will surely can be implemented in a medium or small robotics industry. The controlling of the robot in this way actually mimic the control system in a large organization which to implement a SCADA or DCS.

VII. RECOMMENDATION

For future improvement of the project, additional features of the GUI system can be added such as the error elimination, motor gripper control, and getting the current angle of the robot actuator by using a feedback sensor. Also, it is found that, one Arduino Mega can control up to two Robots and one computer can detect for more than one Arduino COM port. This makes us able to control many robots just in one computer wirelessly. It can be said that the connection made between the server computers to the robot is similar to the Profibus application as we monitor the robot.

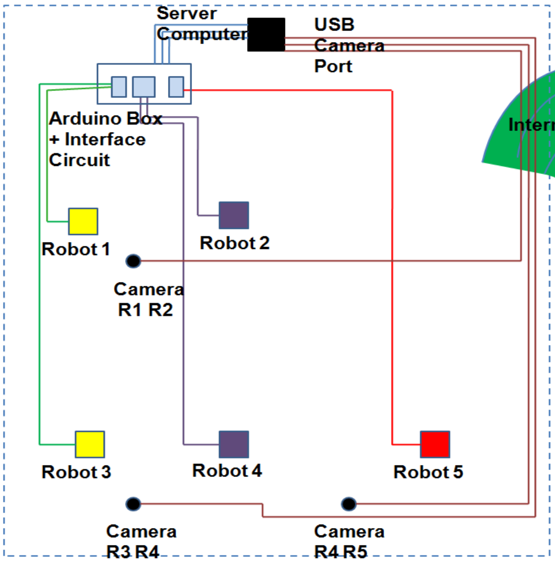


Fig. 10. Floor plan for future development of the system at the IIUM Robotics laboratory.

It is suggested that the system could be expand to multiple robots as shown in Fig. 10. It will reduce cost and help both operator and programmer give a positive impact for the students in terms of demonstrating the online and offline programming in Industrial Robotics. Students will then know that there are many optional for control the robot wirelessly. Furthermore, instead of CN5 input port, there are many other ports such as CN6~7 (for output devices) and CN10 (for controlling the robots motor). By having access on these ports, calibration can be done from the external mode. Other than input channel CN5, which allows us to run the calibration during robot star-up operation, is the CN6-11 (CAL completion). This function enables the programmer to know that the calibration is completed. It is done when the CAL and operation ready start signal is high.

Another suggestion would be to implement the CALSET operation automatically. CALSET refers to the calibration of the positional relation between the robot main unit and controller. It is performed whenever a mechanical end is changed, motor is replaced or the position data of the encoder are lost due to a random encoder backup battery.

CALSET can be done either by the mechanical ends or by entering XY coordinates. Both calibration methods perform almost at the same accuracy but the mechanical end method is easier and faster. Mechanical end can be accomplish by, move the arm by hand until it touches the mechanical end, and then record the position. This requires more operating space to allow the robot to travel to its allowable ends. Therefore, it is suggest that we could put the displacement sensor at every joint of the robots and getting the feedback form it to know the position of the motors.

ACKNOWLEDGEMENT

We acknowledge the financial support of the Research Management Centre, IIUM, under Research Endowment Grant (EDW B14-161-1046).

REFERENCES

[1] F. Tobe, “Low-cost robots like Baxter, UR5 and UR10 successfully entering small and medium enterprises (SMEs),” *Article, Robohub*, 2013. (http://robohub.org/rethink-robotics-baxter-and-universal-robots-ur5-and-ur10-succeeding/)

[2] R. Marin, G. Leon, R. Wirz, J. Sales, J.M. Clever, P.J. Sanz and J. Fernandez, “Remote programming of network robot within the UJI industrial robotics telelaboratory,” *IEEE Transactions On Industrial Electronics*, Vol. 56, No. 12, 2009, pp. 4806-4816.

[3] A. Rosado-Muñoz, J. Muñoz-Marí, J. V. Francés-Villora and M. Bataller-Mompeán, “Remote Laboratory for Industrial Robot,” *Transaction on Control and Mechanical Systems*, Vol. 2, No. 2, 2013, pp. 77-82.

[4] L. Wang, A. Mohammed and M. Onori, “Remote robotic assembly guided by 3D models linking to a real robot,” *CIRP Annals - Manufacturing Technology*, Elsevier. vol. 63, Issue 1, 2014, pp. 1–4,.

[5] A. Turan, S. Bogosyan & M. Gokasan, “Development of a client-server communication method for Matlab/Simulink based remote robotics experiments,” *IEEE ISIE*, Montreal, Quebec, Canada, 2006.

[6] C.S. Tzafestas, M. Alifragis, N. Palaiologou, S.C.A. Thomopoulos, M. Brahman, and A.E. Exarchou, “Development and experimental evaluation of remote laboratory platform for teaching robot manipulator programming” *Proceedings, Int. Conference on Engineering Education*, Florida, USA, 2004.