International Journal of Innovative Technology and Interdisciplinary Sciences
www.IJITIS.org
ISSN: 2613-7305
Volume 2, Issue 2, pp. 181-191, 2019
DOI: https://doi.org/10.15157/IJITIS.2019.2.2.181-191
Received August 22, 2018; Accepted November 28, 2018

MATLAB-based Graphic User Interface for Monitoring and Control of Wireless Sensor Networks

Altunin Valeri [*a], **Victor Musalimov** [b]

Department of Mechatronics, Saint Petersburg State University of Information Technologies, Mechanics and Optics, St-Petersburg, Russia
[*a] ghst2192@gmail.com, [b] musVM@yandex.ru

ABSTRACT

Wireless sensor networks are becoming more popular to remote sensing applications due to their high mobility, speed, security, and cost effective. A smart sensing network can be setup with XBee modules, sensors, actuators connected to microcontrollers. However due to the limited memory and speed, embedded microcontrollers cannot perform completed and sophisticated calculations needed in a modern sensing system. Further, the design of the remote human-machine interface, interaction and control for this system is also a challenge. This paper develops a new application for a wireless smart sensor network using XBee modules and microcontrollers LPC2148. The network interfaces to the personal computer's Matlab software for data archiving, processing and exchanging in order to improve the ability of calculation, visualization, and control of remote embedded microcontrollers via this wireless network.

Keywords: Matlab graphical interfaces, XBee modules, Microcontrollers, LPC214x, Autonomous robots.

1. INTRODUCTION

Smart sensors system with embedded microcontrollers and remote wireless networks are becoming more popular in the modern technologies due to the fast development of wireless networks. For example, a new wireless radio XBee network can allow the connections of more than 65,000 modules each other via a mesh network with a data rate of 250kbit/s and low power consumption of only some mill watts for transmission distances of hundred meters. The XBee modules can be functioned in low voltage power supplier of 3.3V through a simple serial port UART. A review of different wireless network including operating frequency range, transmission rate, transmission distance, transmitted power, and maximum devices of WiFi, XBee and Bluetooth are summarized in [1].

In this paper, the remote embedded microcontrollers for smart sensors are designed with LPC2148. The selection of this microcontroller is due to the high-speed, 32-bit code, tiny size, and low power consumption. LPC2148 has inbuilt already a low power Real-Time Clock (RTC) and two serial ports. One port can be used to connect to the XBee module and the other port can be used to connect to the Global System for Mobile communications (GSM). Therefore, this system can be formed a mesh network while allowing the users to receive data and to send the control signal simultaneously from the PCs or from the mobile phones.

LPC2148 microcontrollers based on ARM7 processor have become the most pervasive 32 bit architecture in the world with embedded products in mobile phones, autonomous and automation systems. Update, ARM 32 bit processors have made up 95% of smart phones, 35% of digital televisions. This ARM processor with high speed, low cost, and low power consumption has made them the most popular. 37 billion ARM processors have been produced as of 2013 [2] since they offer high speed, low cost and low power consumption. LPC2148 controller with many inbuilt peripherals makes it more efficient and reliable for embedded application developers.

Applications of wireless sensor networks (WSNs) have been found in a number of projects. A WSN system actually is a computer network of many intercommunicating computers connected with several sensors and peripheral electronic/electrical devices. Each computer makes up a node and the system makes up a mesh network by radio communicating to other nodes. This network allows gathering remote information data of temperature, humidity, pressure, speed, video with high resolution, etc. A comparative performance analysis of wireless communication networks for intelligent sensors and their applications can be read in [3].

XBee modules support multiple wireless protocols and radio frequencies (RF) communication. This allows cloud based access to devices and their data. Applications of XBee wireless projects can be found in a number projects: Wireless network based intelligent home furnishing and smart home in [4]; Wireless network home security alert system with integrated approach in [5]; Zigbee based wireless air pollution monitoring system using low cost and energy efficiency sensors in [6]; Intelligent monitoring and controlling of agricultural field parameters using Zigbee modules in [7]; A multi alert patient health monitoring using Zigbee in [8]; and remote monitoring and control system for DC motor using Zigbee Protocol in [9].

However, as mentioned in above, the embedded microcontrollers always have limitation on memory and software calculation in themselves. Therefore, the idea of this paper is to develop an effective network of remote smart nodes to the PC control center for Matlab interface since Matlab is currently the most powerful computing and popular software for scientists and researchers. Matlab can support to store and process the receiving data from remote devices via serial ports with XBee modules, then, send calculated data, control signals back to the remote devices. Further, Matlab can provide multiple graphic tools via Simulink and Graphical User Interface (GUI). The researchers can design a full Human-Machine Interface, Interaction and Control of all remote nodes with Matlab GUIs.

Applications of Matlab GUIs for wireless sensor networks are still few and most of these applications are for storage and processing data. A new embedded system and Matlab based GUI for online acquisition and analysis of electrocardiogram (ECG) can be read in [10]. Another development for monitoring and controlling GUI based devices using Matlab can be read in [11]. A recent project with Matab calculations for vehicle feasible paths is referred in [12], [13], [14], [15] and [16]. In this research, GUIs, Simulink, computational programing, and data communication from Matlab software will be used for the data-exchanged between the PC control center and remote microcontrollers to store and process data, and then, send back the control commands.

This paper exploits advantages of Matlab GUIs, Simulink, and powerful computational ability of Matlab to supervise and to control the remote microcontrollers via XBee modules wireless network. The outline of this paper is as follows: Section 2 describes the hardware development for this project including mobile robots, smart sensors, microcontrollers and Xbee modules; Section 3 describes the software development and the remote network architecture; Section 3 illustrates experimental results; and finally in Section 4, conclusions and recommendations are withdrawn.

2. HARDWARE DEVELOPMENT

The hardware for this project includes three (3) mobile robots; each mobile robot has four (4) independent hall-effect encoders and four (4) independent DC motors controlled by a 12V DC driver. The mobile robot equips with a LPC2148 microcontroller with onboard sensors and camera. Each mobile robot can communicate with each other robots and with the PC control center via a XBee modules and GSM mobile phones via two inbuilt UART ports already in the LPC2148 microcontroller as shown in figure 1.

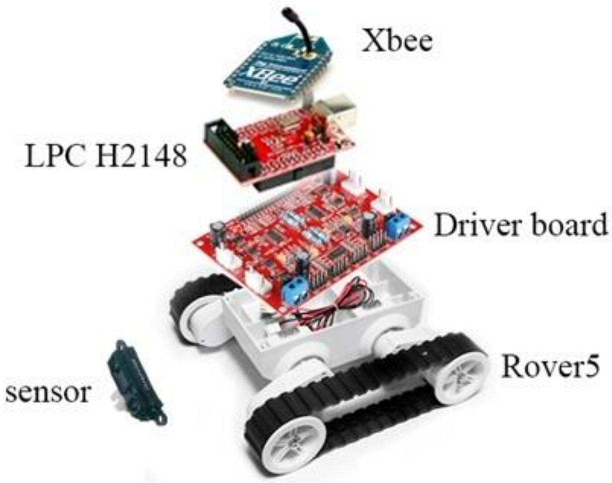

Figure 1: Mobile robot configuration

Therefore, this mobile robot can work fully autonomously based on its sensors and the pre-programming in the LPC2148 microcontroller. This robot can also generate the human-robot interface and interaction to other robots and to the PC control center or from the mobile phones via an wireless network communication.

The control center is designed with a PC, a LPC2148 microcontroller, sensors, cameras and XBee modules for communication to remote microcontrollers. One serial port from LPC2148 is connected to the PC Matlab R2015 software. Sensory data from remote microcontrollers are received, processed and send back to them via this serial port. The control graphic panel and the human robot interface can also be developed from Matlab GUIs and Simulink. Sensory data and video can be stored and displayed on a large PC monitor as shown in figure 2.

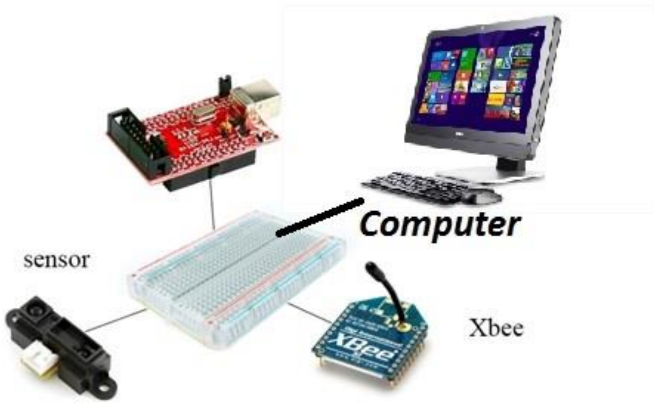

Figure 2: Computer center configuration

The above hardware system is setup mainly to test the ability of the remote data exchanged to the PC Matlab software and the graphics interface by Matlab to supervise and to control the remote microcontrollers via the XBee modules communication. The design of software for this project is described in the next section.

3. SOFTWARE DEVELOPMENT

The software for this project includes the C-code programing for each LPC2148 microcontrollers, the XCTU configuration for XBee modules, and the Matlab m files code for receiving and processing data from remote microcontrollers, the GUIs and Simulink design to supervise and to control the remote devices. The mobile robots are equipped with cameras, smart sensors and wheel DC motors with encoders as mentioned in the above hardware section. Each encoder for each wheel consists of both receiver and transmitter modes. The microcontroller operates sensors, motors and activates the wireless network. Further, the encoder in each wheel can exchanged data to each other and can be controlled from the local microcontroller as well as from the remote PC control center. After processing data, the local microprocessor will make the decision to continue moving or to stop moving for each motor. The moving decision is also transferred to the PC control center where the updated data from smart sensors and the onboard cameras are processed. For example, when the robot sees obstacles, it can determine itself how to overcome or the PC control center can calculate best feasible path for them since data from one robot can also be online exchanged to other robots and to the PC control center in a mesh network. The flowchart for the data communication of this system is presented in figure 3.

The Matlab software at the PC control center can support the control decision for each wheel encoder-motor and allow the mobile robot tracking the best flexible path online avoiding obstacles. The calculation procedure for this feasible path can be read and referred to in our recent project at [12]. The Matlab software help to generate an optimal feasible path, and then, control the robot to track exactly on this path from any given starting points to any given destination points subject to the robot physical contsrtaints and the surrounding updated obstracles. The upper part of figure 4 shows an calculated feasible path for the mobile robot from a starting point of $x_0 = 0$, $y_0 = 0$, and an initial body angle of $\theta_0 = 180^0$, to a final destination point of $x_F = 0$, $y_F = 0$, and to the final the body angle of $\theta_F = -180^0$. The lower part of figure 4 shows the robot controlled velocity. In this example, the robot has to reverse (negative velocity), then, move forward, and then, reverse again to reach the destination point.

For the communication and exchanged data, all XBee modules are configured in a mesh network on XCTU. This allows all nodes communicating each other and to the PC control center. The configuration of each XBee modules on XCTU software allows the two ways communication among all XBee modules. The XCTU window interface is designed as shown in figure 5.

Communication between two XBee modules can be seen in figure 6. The XBee module on the left hand side can establish a two ways communication to the XBee module on the right hand side as in real time. Similarly to any other conventional communication configuration, each sending and receiving message should contain it's specified address, a start and an end mark and with a confirmation feedback from the each senders and receivers.

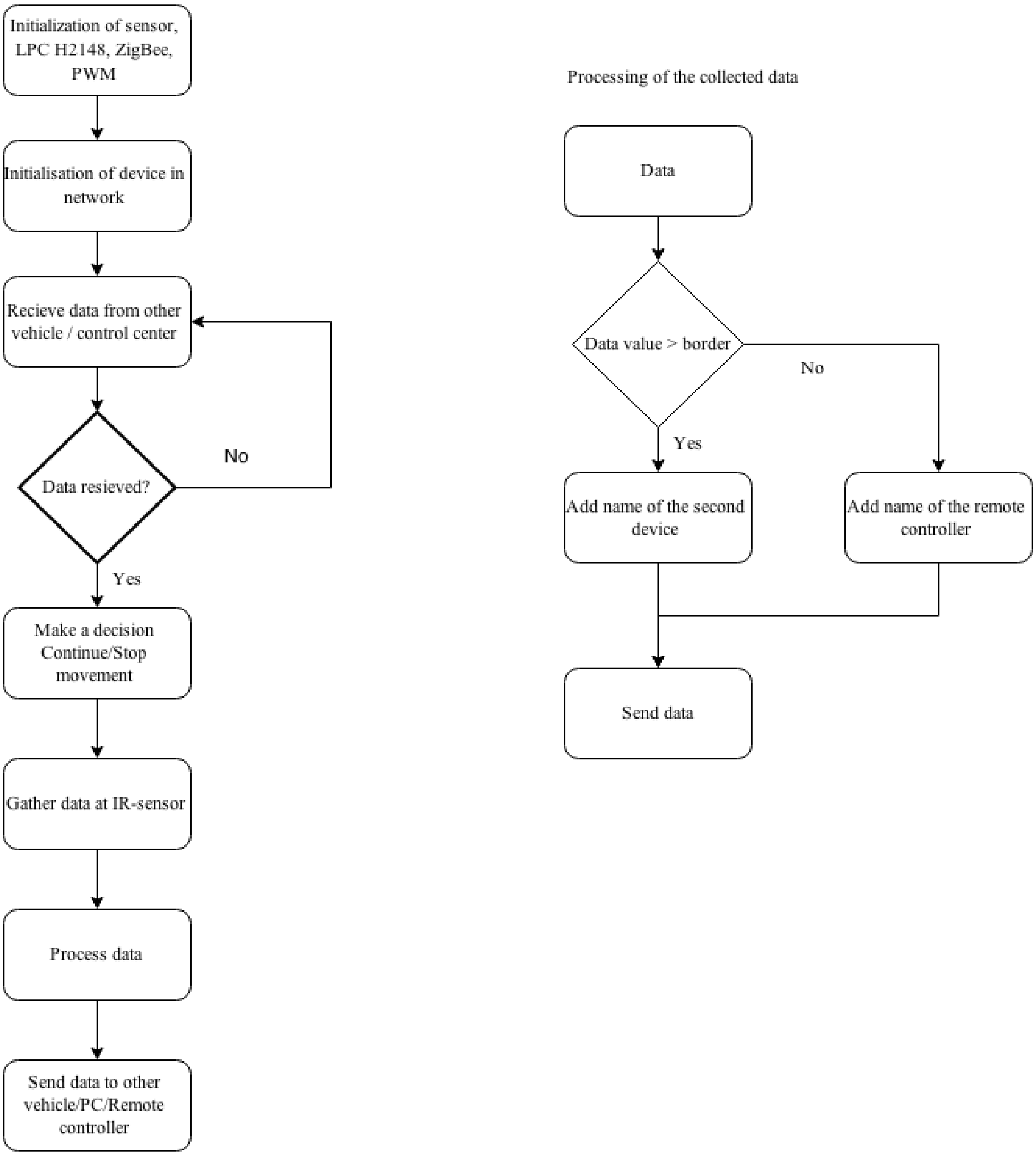

Figure 3. Flowchart of the data communication

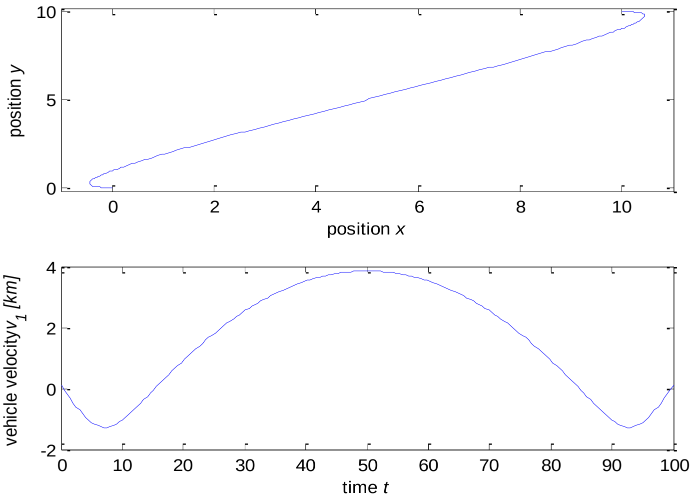

Figure 4. Path trajectory and velocity

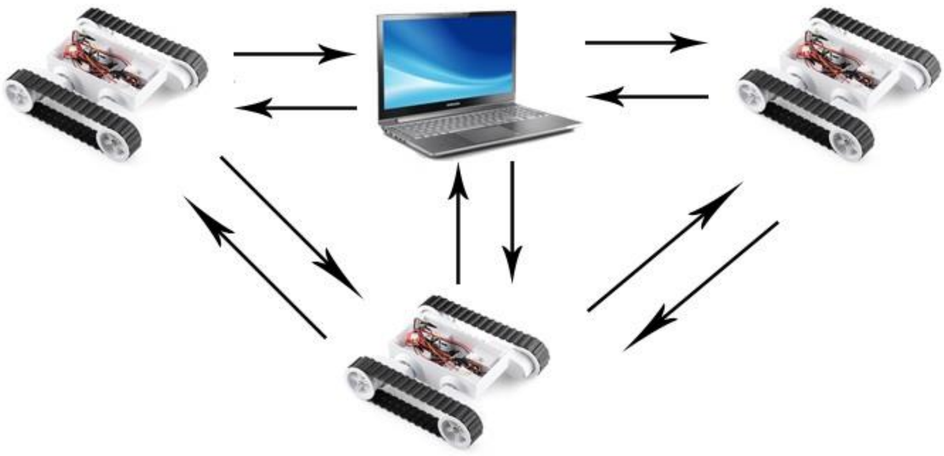

Figure 5: XBee mesh network configuration

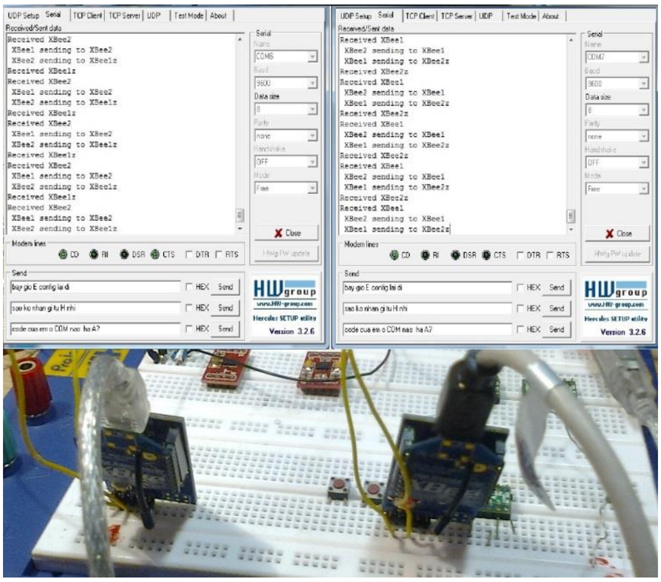

Figure 6: XBee modules on XCTU test

Each LPC2148 microcontroller is programed by C-code and compiled by IAR software. An example C-code for a LPC2148 microcontroller of this project is shown in figure 7. This C-code allows the microcontroller receiving the data from smart sensors (see the upper part of figure 7) and then, exchanging data to other devices (see the lower part of figure 7). Therefore, each robot can work as autonomously itself while receiving the calculation and guidance support from the PC control center with Matlab software.

```
#include<NXP/iolpc2148.h>
#define DESIRED_BAUDRATE 9600
#define CRYSTAL_FREQUENCY_IN_HZ 12000000
#define PCLK CRYSTAL_FREQUENCY_IN_HZ // since VPBDIV=0x01
#define DIVISOR (PCLK/(16*DESIRED_BAUDRATE))

#include <stdio.h>
#include <math.h>
#include <stdint.h>
//**************** Define Register Command *********************

#define CRA          0x00      //00 Config Register A
#define CRB          0x01      //01 Config Register B
#define MR           0x02      //02 Mode Register
#define DXRA         0x03      //03 Data Outpou X MSB   Register
#define DXRB         0x04      //04 Data Outpou X LSB   Register
#define DZRA         0x05      //05 Data Outpou Z MSB   Register
#define DZRB         0x06      //06 Data Outpou Z LSB   Register
#define DYRA         0x07      //07 Data Outpou Y MSB   Register
#define DYRB         0x08      //08 Data Outpou Y LSB   Register
#define SR           0x09      //09 Status Register
#define IRA          0x0A      //10 Identification Register A
#define IRB          0x0B      //11 Identification Register B
#define IRC          0x0C      //12 Identification Register C


//---------------- Define I2C Pin -----------------

z_high = 0061 , z_low = 0061 , z = 15677
y_high = 0061 , y_low = 0061 , y = 15677
x_high = 0061 , x_low = 0061 , x = 15677
z_high = 0061 , z_low = 0061 , z = 15677
y_high = 0061 , y_low = 0061 , y = 15677
x_high = 0061 , x_low = 0061 , x = 15677
z_high = 0061 , z_low = 0061 , z = 15677
y_high = 0061 , y_low = 0061 , y = 15677
x_high = 0061 , x_low = 0061 , x = 15677
z_high = 0061 , z_low = 0061 , z = 15677
y_high = 0061 , y_low = 0061 , y = 15677
x_high = 0061 , x_low = 0061 , x = 15677
```

Figure 7. C-code for LPC2148 and data achieving

Finally, the software is developed at the PC control center with Matlab GUIs and simulink. An example of Matlab m file coding at the PC control center is shown in figure 8. The GUIs and simulink design includes the graphic control panel with touch screen, video displays. Matlab software also support the calculation burden for remote microcontrollers by receiving and then, sending back the calculated data as the real time.

```
function testGUIwebcam1_OK


 hFigure = figure('name', 'MATLAB HUMAN-MACHINE INTERFACE (testGUIwebcam1_OK)',...
                 'Position',[200 200 1000 500],...  %[left bottom width height]
                  'MenuBar','figure',...
                  'KeyPressFcn',@stop_keypress);


ha1 = axes('Units','pixels','Position',[0,300,350,200]); %[left bottom width height]
vid = videoinput('winvideo',1);
src = getselectedsource(vid);
vid.FramesPerTrigger = 1;
hImage = image(zeros(699,1500,1));
preview(vid, hImage);
uicontrol(hFigure,'Style','pushbutton',...  %# Create the button
                    'Position',[20 340 120 20],...
                    'String','Stop Camera','FontSize',11,'ForegroundColor','red',...
                    'HorizontalAlignment','center',...
                    'Callback',@stop_camera);
uicontrol(hFigure,'Style','pushbutton',...  %# Create the button
                    'Position',[20 320 120 20],...
                    'String','Re-Start Camera','FontSize',11,'ForegroundColor','blue',
                    'HorizontalAlignment','center',...
                    'Callback',@start_camera);
```

Figure 8. Matlab m code in PC

The above software development allows setting up a mesh wireless network for exchanging data among remote microcontrollers and the PC control center. Some experimental results from this system are briefly illustrated in the next section.

4. EXPERIMENTAL RESULTS

Experiment for this project is conducted with four (4) XBee wireless modules, three (3) are at remote and one (1) is at the PC control center. The configuration of remote XBee modules and the communication network are configured and tested on XCTU window as shown in figure 9.

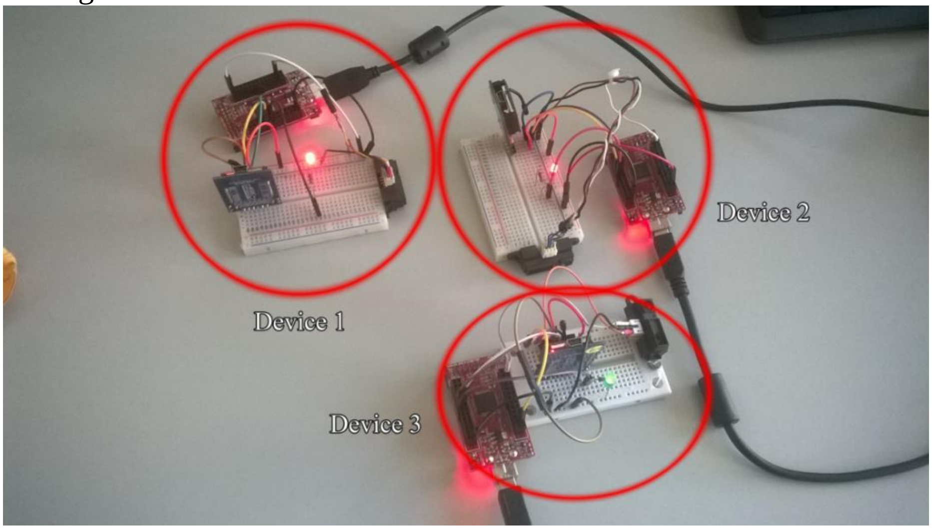

Figure 9. XCTU communication tested for remote three XBee modules

The PWM signals to control the remote DC motors are received by the PC control center and checked on an oscilloscope as shown in figure 10. The online checking PWM signals allow to regulating the accuracy movement for each motors from the feedbacks at their encoders.

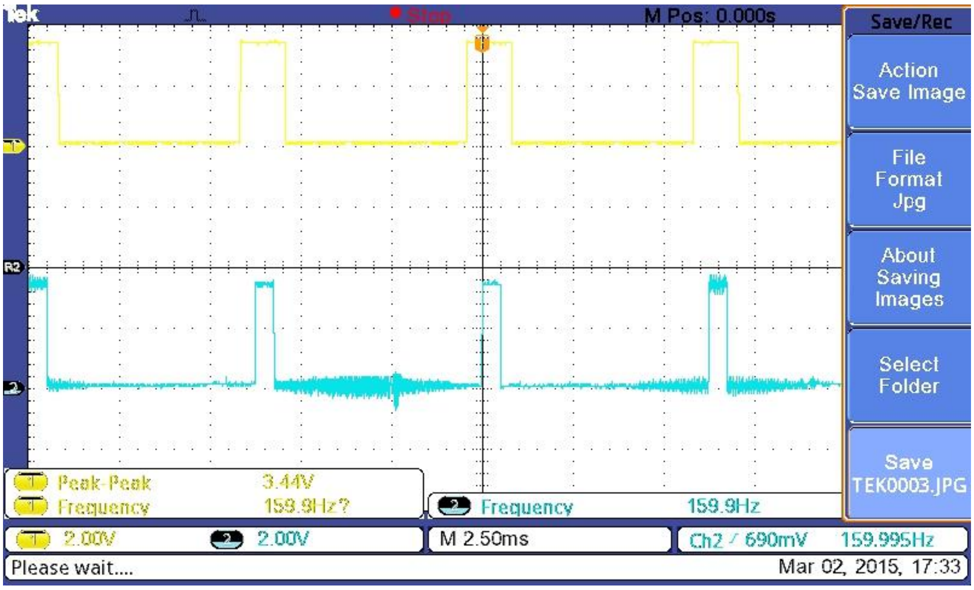

Figure 10: PWM ouputs for each remote DC motor

The PC control center will receive the sensory data from remote microcontrollers and the Matlab will help to process and then send back the calculated data to their remote microcaltrollers as shown in figure 11.

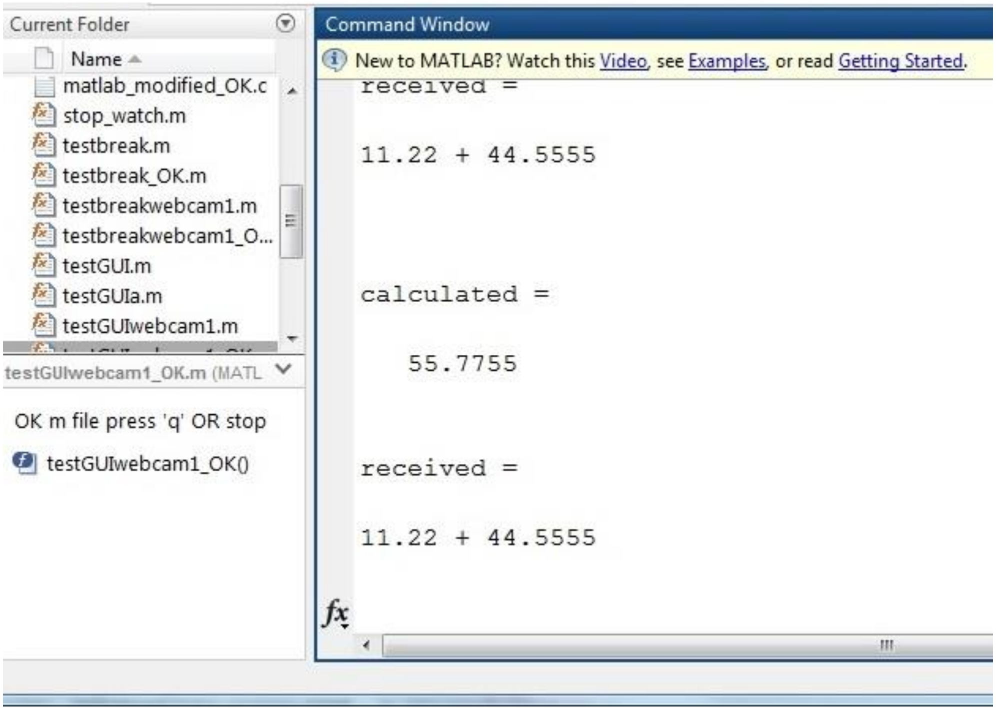

Figure 11: Matlab reduces calculation burden for remote microcontrollers

And the final step is to design the graphics for human-robot interface to supervise and to control the remote microcontrollers. Matlab graphics also connect with remote cameras, four different sensory data diagrams, a counter and several control buttons are designed for this experiment as shown in figure 12. From these Matlab graphics at the PC control center, the operator can supervise and to control the remote microcontrollers.

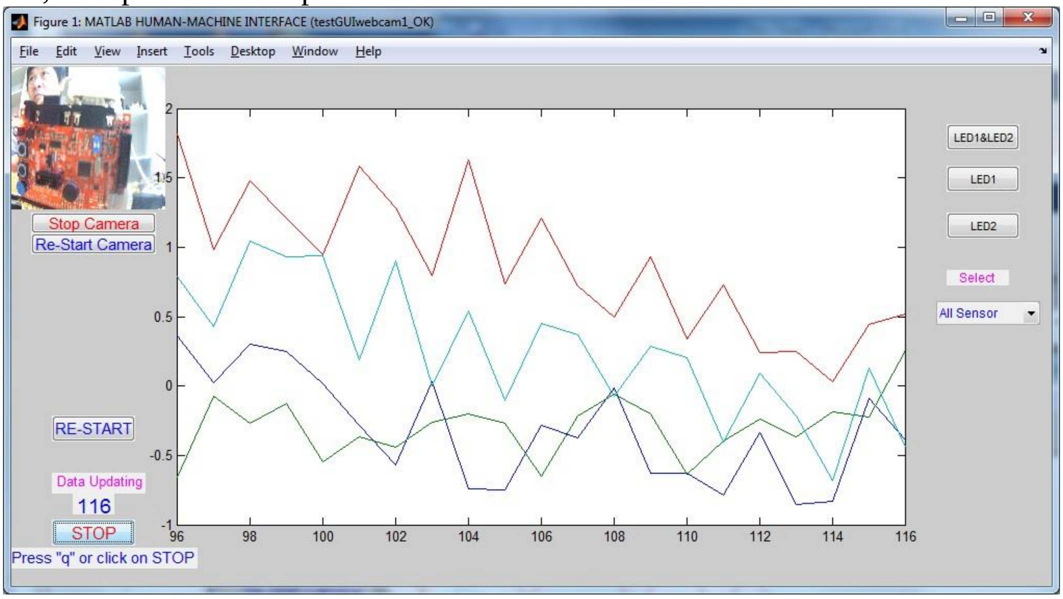

Figure 12: Experiment with Matlab GUIs design

The Matlab GUIs help designing graphic human-robot interfaces, storing data, supervising and controlling the remote microcontrollers. Further, they also allow using touch screen monitors or clicking mouse or typing keyboard. In figure 12, the operator can turn on and off any functions of the remote microcontrollers, to watch and to control them with touch screen or mouse or keyboard. This sensing network allows manipulation and computation of sensory data in a smart interfacing network since this system allows the interaction and cooperation with other devices and operators via PC control center and mobile phones. This smart sensing network can update itself from the changing environment and re-preprograming itself when necessary.

5. CONCLUSION

In this paper, a smart wireless sensor network is presented using XBee modules and Matlab based GUIs for LPC2148 remote microcontrollers. The project is designed with all necessary from Matlab GUIs, simulink, and its serial communication to LPC2148 microcontrollers via XBee modules. Experiment results show that the GUIs in Matlab can be effectively used to reduce the calculation burden for remote microcontrollers, achieve smart sensory data, and perform online data displays and camera visualization via the wireless network. The project allows developing low cost solution and high performance for the XBee modules wireless networks for embedded microcontrollers.

ACKNOWLEDGMENT

The authors would like to thank Saint Petersburg State University of Information Technologies (ITMO) for supporting the preparation of this research article.

CONFLICT OF INTERESTS

The author would like to confirm that there is no conflict of interests associated with this publication and there is no financial fund for this work that can affect the research outcomes.

REFERENCES

[1] Николай Агафонов, *Технологии беспроводной передачи данных ZigBee, BlueTooth, Wi-Fi,* Беспроводные технологии No.1, 2006.

[2] Online: http://www.engineersgarage.com, website for Introduction to ARM7 Based Microcontroller (LPC2148), 2015.

[3] Chakkor S., Baghouri M., Ahmadi C., and Hajraoui A., *Comparative Performance Analysis of Wireless Communication Protocols for Intelligent Sensors and Their Applications*, International Journal of Advanced Computer Science and Application, Vol. 5, No. 4., pp. 76-85, 2014.

[4] Huang Jinhui, Xue ZiYu, Xu Xu and Tian Siyang, *Zigbee-based Intelligent Home Furnishing*, International Journal of Smart Home Vol. 9, No. 1, 2015

1. C.P.Jeba Samuel, S.Prakash, M. Ananda Kumar, D. Manivannan, *ZigBee Based Home Security Alert System: Integrated Approach,* Journal of Artificial Intelligence 6(1), 2013

[5] Vasim K. Ustad, A.S.Mali, Suhas S.Kibile, *Zigbee Based Wireless Air Pollution Monitoring System Using Low Cost and Energy Efficient Sensors,* International Journal of Engineering Trends and Technology (IJETT) – Volume 10 Number 9, 2014

[6] Bhagat Singh, Harender, *Intelligent monitoring & controlling of agricultural field parameters using ZigBee*, IJITE Vol.03 Issue-01, 2015

[7] Malika, Charu Rana, *A Multi-Alert Patient Health Monitoring using Zigbee*, International Journal of Computer Applications, Volume 83 – No.17, December 2013

[8] Mohini Reddy, Vidya Sawant, *Remote Monitoring and Control System for DC Motor using Zigbee Protocol*, International Journal of Application or Innovation in Engineering & Management, Volume 3, Issue 4, 2014

[9] Gupta R., Bera J., and Mitra M., Develoment of an Embedded System and Matlab based GUI for Online Acqusition and Analysis of ECG Signal, Measurement, vol. 43 (2010) pp. 1119-1126, 2010.

[10] Nidhi M., Mayuri S., Shruti V., and Shweta S., Controlling and Monitoring of GUI based Devices Using Matlab, International Journal of Engineering Research and Applications, vol. 4, issue 4, pp. 17-20, 2014.

[11] Minh V.T and Pumwa J "Feasible Path Planning for Autonomous Vehicles", *Mathematical Problem in Engineering*, Vol. 2014, Article ID: 317494, 12 pages, 2014. DOI:10.1155/2014/317494.

[12] Vu Trieu Minh, FB Mohd Hashim, Mokhtar Awang, 'Development of a real-time clutch transition strategy for a parallel hybrid electric vehicle', Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 2012, Vol. 226(2), pp. 188-203.

[13] Vu Trieu Minh, Nitin Afzulpurkar, WM Wan Muhamad, 'Fault detection model-based controller for process systems', Asian Journal of Control, 2011, Vol. 13(3), pp. 382-397.

[14] Minh, Vu Trieu, Tamre Mart, Musalimov Victor, Valeri Altunin, Development of a Wireless Sensor Network Combining MATLAB and Embedded Microcontrollers , Sensor Letters, 2015, 13(12), 1091 – 1096.

[15] D. Katushin, "Overview of Tunnel Boring Machines and Testing Devices for Drag Bits", *International Journal of Innovative Technology and Interdisciplinary Sciences*, vol. 1, no. 1, pp. 34-38, Nov. 2018.

[16] R. Moezzi, V. Minh, and M. Tamre, "Fuzzy Logic Control for a Ball and Beam System", *International Journal of Innovative Technology and Interdisciplinary Sciences*, vol. 1, no. 1, pp. 39-48, Nov. 2018.