Scientific Bulletin of the „Petru Maior" University of Tîrgu Mureş
Vol. 14 (XXXI) no. 2, 2017
ISSN-L 1841-9267 (Print), ISSN 2285-438X (Online), ISSN 2286-3184 (CD-ROM)

DATA ACQUISITION AND CONTROL SIMULATION

Traian TURC[1], Ioan MOCIAN[2]
"Petru Maior" University of Tîrgu-Mureş, Romania
Nicolae Iorga Street, no.1, 540088 Tîrgu-Mureş, Romania
[1]traian.turc@ing.upm.ro
[2]ioan.mocian@ing.upm.ro

Abstract

The aim of this paper is to build a master application that controls a brake test system. One crucial step is developing a data acquisition and control system, capable of simulating the real physical stand. The simulator must supply data to the master application and signal the receipt of any instructions sent back to it. It is also means to support the development of a monitoring application to be run on mobile devices.

Key words: data acquisition, simulation, master application, slave application, signal, web application, mobile devices, supply data, C#, Visual Studio IDE.

1. Introduction

The purpose of a brake test stand is to measure the quality of a vehicle's brake system and send the measurements to a computer. The data is compiled into a report, showing whether the vehicle meets the law regulations and can therefore operate on public roads.

The communication between the brake test stand and the computer is facilitated by a data acquisition and control system [4] and a master application running on the stand. In the first phase of our project, we build a simulation of the data acquisition system, so that we can develop the master application in isolation, without having to rely on an actual test stand.

2. Data acquisition and control simulation

In order to simulate the brake test stand, we opted for a multi IO acquisition system [3] with 6 analog inputs, 8 digital inputs and 8 digital outputs, based on the ARDUINO MEGA 2560 development board (fig.1) [6].

The protocol used to exchange messages between the development board and the computer is a proprietary protocol based on text messages.

Fig. 1: Development board

3. Data transfer

In this paper is presented a single command, namely M1, cda. M1 is the code of the command, while cda is a number between 0 and 255, coding the decimal value for the 8 digital outputs.

The development board responds with the following string: M1, A0, A1, A2, A3, A4, V1, Val_B. Analog inputs A0...A4 are five values between 0 and 1023. V1 is the value measured by the speedometer. V1 is calculated according logical scheme (fig. 2)

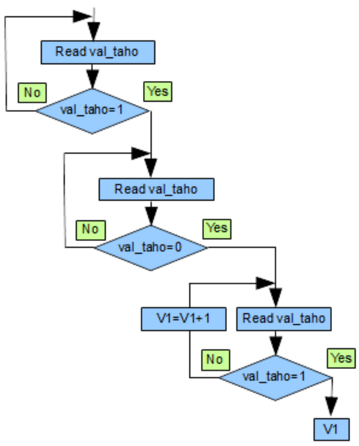

Fig. 2: Logical scheme for speedometer

The application from developer board keeps track of the time elapsed between two consecutive transitions of the signal that comes from an inductive sensor (val_taho), placed in front of a slit disk that rotates together with the cylindrical rollers that simulate the driveway.

Val_B is the decimal value corresponding to digital inputs B1-B4.

After computing the necessary values, is waiting for the data request issued by the master application, and reply with the string M1, A0, A1, A2, A3, A4, V1, Val_B.

4. Instruction execution

The string received from the master application has the following form: “M1, cda”, where “cda” is a number between 0 and 255 representing the decimal value of the instructions to be executed in order to compute the 8 digital outputs C0-C7.

The development board receives the string “M1, cda”, extracts “cda” from this string and converts it to a number. Then, it executes the received instruction by converting “cda” to binary. Next, it sends the right instruction to every output. The software running on the computer is a master application, since it initiates the communication with the development board. After the computer sends the string “M1, cda”, it waits for the slave application on the development board to send back its reply.

5. Master application

The master application running on the computer (fig. 3) is written in C# [1], in the Visual Studio IDE. Its main role is to initiate the communication with the development board, send it the necessary instructions, and read back the values acquired by the development board.

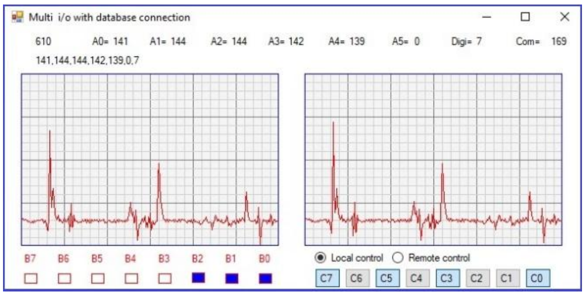

Fig. 3: Master application

As shown fig. 3, the application shows the acquired values (A0-A5), the digital inputs (B0-B7) coded as decimal numbers, Digi and the string received from the development board.

Inputs A0 and A1 are displayed on the graphical interface, allowing their evolution over time to be tracked visually. Thus, the parameters of the braking system can be analyzed in real time, which is one of the primary goals of our project.

Another goal is to make the data accessible in the future to interested parties. To address this goal, we incorporate a database in our system that can store the measurements recorded by the test stand. We built a *cons_el* database with *a sitef* table (fig. 4), with the following structure:

sitef_id	a0	a1	a2	a3	a4	a5	digi	com	c0	c1	c2	c3	c4	c5	c6	c7	remote

Fig. 4: *Sitef* table

The fields A0-A5 are the digital inputs, the *digi* field is a value which represents the decimal number coded by the digital inputs B0-B7, com is an instruction code for the acquisition board for future development, fields C0-C7 are in binary format and correspond to digital instructions C0-C7. The remote field is in binary format and will be used by the computer application whenever it hands over the control to one of the mobile devices.

The master application running on the computer must access this table and store the acquired data in it, allowing slave applications to read it back by connecting to the *cons_el* database and reading the *sitef* table[1]:

Data is refreshed at fixed intervals. After each refresh, the content of the table is similar to the one shown in fig. 5.

sitef_id	a0	a1	a2	a3	a4	a5	digi	com	c0	c1	c2	c3	c4	c5	c6	c7	remote
1	141	144	144	142	149	0	7	1445	1	0	0	1	0	1	0	1	0

Fig. 5: *Sitef* table content

Based on this table, we built a web application (fig. 6) which can run on any device with a web browser. In order to refresh the data, we tested multiple methods, including: reloading the page at regular intervals, Java applets, web services, AJAX technology. The latter proved to be the most efficient.

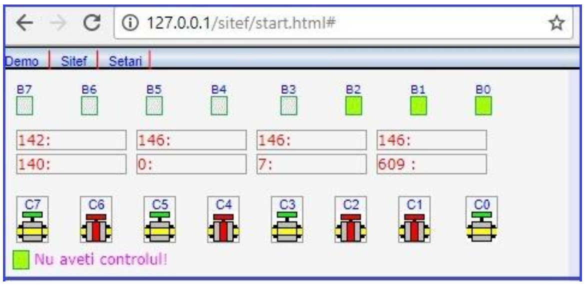

Fig. 6: Web page application as slave

Our web application is based on the AJAX technology to facilitate data refresh at fixed time intervals, without reloading the page.

The request is sent to the server application *answer_d.php*, which accesses the *sitef* table in the *cons_el* database.

The data stored in the table is then shown on the web page in fig. 6. In order to be able to hand control over to mobile devices, users can choose the remote option (fig. 7). The master application sets the remote variable in the database (fig. 8).

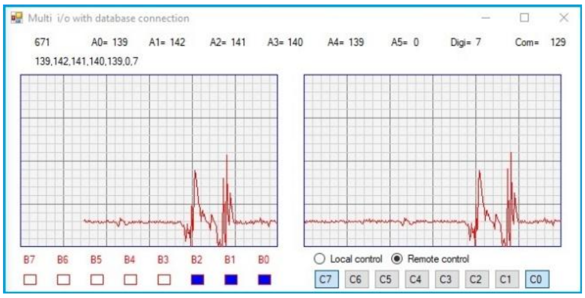

Fig. 7: Master application on remote

In this case, the application running on the computer loses its master authority, handing control over to the mobile device. Instructions C0-C7 are not active anymore; their state is displayed on the graphical interface. The state of buttons B0-B7 agrees with the Digi=7 value; their binary value is 00000111, so the last buttons B2-B0 are active. Figure 7 shows the new contents of the *sitef* table.

sitef_id	a0	a1	a2	a3	a4	a5	digi	com	c0	c1	c2	c3	c4	c5	c6	c7	remote
1	139	142	141	140	139	0	7	129	1	0	0	0	0	0	0	1	1

Fig. 8: *Sitef* table for remote case

In this case, the web page application (fig. 9) can take over the system.

Fig. 9: Web page application as master

The buttons are active. The web application is the one controlling their values, namely 100000001.

6. Signal trends

The data acquisition system collects inputs over time and applies the values of the commands on the digital outputs every time it receives a request from the master application running on the computer. Parameter evolution can be analyzed in depth with precision: we added a series of new instructions that lead to 300 consecutive reads of a single input, placed in a single packet to be sent over. The sampling speed becomes much higher, which allows analyzing high-frequency signals over time. We implemented commands A0,A1,A2,A3,D1 in order to read analog inputs a0,a1,a2,a3 and digital input B0 (which is connected to the speedometer). For instance, the method that reads the analog input a0 is listed below:

These commands are necessary to analyze various inputs over time. We can thus produce a report of the analog inputs that are connected to the electro-resistive sensors and evaluate the level of noise.

We can also analyze the speedometer input, applied to a digital signal that can go up to 700 Hz (fig. 10).

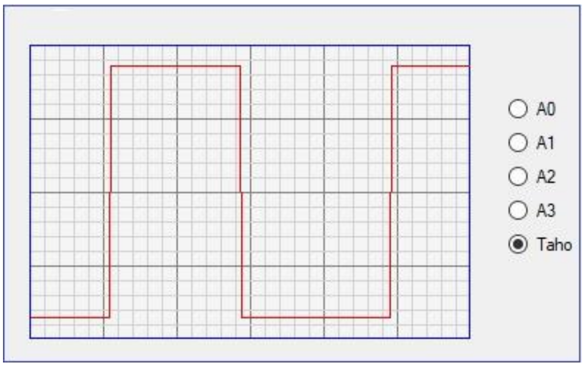

Fig. 10: Speedometer waveform

In order to display the signal on a graphical interface, we implemented the *instrumm* class, instantiated by the *instrumm1* object:

```
instrum1 = new instrum(pozx, pozy, n_maxx,
val_max);
```

where *prozx* an *prozy* are the object coordinates on screen, *n_maxx* is the number of samples, and *val_max* is the maximum value of the signal displayed.

7. Results

Our proprietary protocol is stable and fast due to its simplicity. The data acquisition simulator is able to simulate the real brake test stand and facilitated the development of the master application. When the web application is launched on an intranet, the latency is insignificant; the web application is as fast as a local computer. The AJAX technology used for our web application was the most efficient out of the methods we benchmarked. Using instructions for signal evolution over time, we managed to analyze signals up to 1000 Hz.

8. Conclusions

We developed a data transfer protocol, and an application for a data acquisition simulator. The latter is capable of simulating a real brake test stand, and facilitated the development of the master application on a local computer. We also developed a web application using the AJAX technology, which allows mobile devices to take control over the system. This control handoff from the computer to the mobile devices relies on a novel approach. Due to the fact that our web application uses a database, the information can be partitioned and accessed by multiple interested parties. Furthermore, the data is made available in real time during testing. The results of the tests are also accessible to law enforcement.

Acknowledgement

This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI - UEFISCDI, project number PN-III-P2-2.1-CI-2017-0041, within PNCDI III

References

[1] Microsoft Website (2017). [Online]. Available: https://docs.microsoft.com/en-us/visualstudio/welcome-to-visual-studio

[2] Arduino Website (2017). [Online]. Available: https://cdn.arduino.cc/reference/en/

[3] Zhan, Y., Sun, R., Zhang, X., (2010), *Data Acquisition and Processing System for Vehicle's Braking Performance Based on Visual C plus*, Conference: International Conference on Frontiers of Manufacturing and Design Science (ICFMD2010), Chongqing, Peoples R China.

[4] Wang, Y., Wu, M., Tian, C., (2017), *Development of remote data acquisition system based on OPC for brake test bench*, Conference: 2nd Annual International Conference on Information System and Artificial Intelligence (ISAI).

[5] Agrawal, S.; Kumar, V., Anand, N., (2016), *Development of Data Acquisition System and Data Analysis Technique for Automotive Applications*, Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave), Karpagam Coll Engn, Coimbatore, India.

[6] Stuki, F., Richards, R., (2013) *Sensor and data acquisition requirements for automotive control-systems*, Electronic Engineering, vol. 53, issue 659, pp. 171.

[7] Lita, I., Visan, D., Cioc, I. (2013), *Data Acquisition System for Collision Tests in Automotive Industry*, Conference 36th International Spring Seminar on Electronics Technology, Alba Iulia, Romania.

[8] Gautam, V., Rajaram, V. (2017), Subramanian, *Model-based braking control of a heavy commercial road vehicle equipped with an electropneumatic brake system, Proceedings of the Institution of Mechanical Engineers part d-journal of automobile engineering*, vol. 231, issue 12, pp. 1693-1708.

[9] Song, J., (2017), *Development of Vehicle Integrated Dynamics Control System with Brake System Control* Transactions of the Korean Society of Mechanical Engineers, vol.41, issue 7, pp. 591-597.

[10] Freiwald, A., Hwang, G. (2017), *Safe and Secure Software Updates Over The Air for Electronic Brake Control Systems*, SAE International Journal of Passenger Cars-electronic and Electrical Systems, vol. 10, issue 1, pp. 71-82.

[11] Koeppen, T., Kuepper, T., Makarenkov, O. (2017), *Existence and stability of limit cycles in control of anti-lock braking systems with two boundaries via perturbation theory*, International Journal of Control, vol. 90, issue 5, pp. 974-989.

[12] Midgley, W., Cebon, D. *(2016), Control of a hydraulic regenerative braking system for a heavy goods vehicle*, Proceedings of the Institution of Mechanical Engineers part d-Journal of Automobile Engineering, vol. 230, issue 10, pp. 1338-1350.

[13] Huang, S., Zhou, C., Yang, L. (2016), *Transient fault tolerant control for vehicle brake-by-wire systems*, Engineering & System safety, vol. 149, pp. 148-163.

[14] Fedotov, A.; Mlynczak, M. (2016), *Simulation and Experimental Analysis of Quality Control of Vehicle Brake Systems Using Flat Plate Tester,* Conference: 11th International Conference on Dependability and Complex Systems (DepCoS-RELCOMEX), Brunow, Poland.