

Development of Hardware Interfacing System for Visual C++

Doyce T. Martinez, Tomas U. Ganiron Jr, Catherine S. Lacsamana

Emilio Aguinaldo College/Qassim University, San Marcelino Street, Manila/Buraidah City

Article Info

Article history:

Received Jun 25, 2013

Revised Aug 12, 2013

Accepted Aug 27, 2013

Keyword:

Accelerometer

Algorithm

Hardware interfacing

Visual C++

ABSTRACT

This paper discussed on the development of automated data acquisition circuits using the parallel port of a computer through Visual C++. The aim of this study is to produce low cost laboratory apparatus that are both accurate and easy to use. Two revisions of an accelerometer device are described in this paper. Each prototype determined sources of errors (initial moment of inertia, device delay, code optimization) that were used in improving the design for the next revision. For the first revision errors of 13.63% and 16.38% between the experimental and theoretical values were recorded while the second revision showed errors of 12.58 % and 7.20 % on two different angles. The next revision of the prototype would include the use of embedded assembly code in the Visual C++ code together with advanced thread synchronization techniques to improve the accuracy of the measured data.

*Copyright © 2013 Institute of Advanced Engineering and Science.
All rights reserved.*

Corresponding Author:

Tomas U. Ganiron Jr,
Emilio Aguinaldo College/Qassim University,
San Marcelino Street, Manila/Buraidah City.
Email: tomas@qec.edu.sa

1. INTRODUCTION

The Emilio Aguinaldo College (commonly abbreviated as EAC) is a private nonsectarian institution of learning located in San Marcelino Street, Manila.

The Physics laboratory of the Emilio Aguinaldo College uses different types of apparatus for data acquisition to perform experiments that demonstrate the different topics in the Physics subjects. The authors intend to fully automate these data acquisition apparatus using electronic circuits that convert measurable data through sensors to a readable bit format for the parallel port of the computer. A survey was done to determine which of the apparatus can be readily automated and can be replaced by readily available low cost materials. One of the experiments that showed possible automation was the calculation of the acceleration of the different moving bodies. It was observed that the error from this experiment was mainly due to the reaction time of the students. By substituting the apparatus with electronic circuits that are inexpensive, rugged and reliable, students can obtain data with ease and can focus more on the concept rather than its procedural and computational parts. Another advantage of automating the experiment through the computer is that a customized Graphical User Interface (GUI) can be used through the Microsoft Foundation Classes (MFC) of Visual C++. This also helps in the automatic data logging of the results.

2. RESEARCH METHOD

The implementation of the acceleration measurement device has gone through two revisions. The first revision consisted of four sensors that heavily relied on the algorithmic flow of the Visual C++ API [1]. The second implementation uses only two sensors and relies on the internal clock of the computer accurate up to the millisecond. Another revision is being designed that includes the use of advanced thread synchronization and embedding assembly language into the Visual C++ code to allow for more versatility and low-level accuracy.

2.1. Revision 1

Figure 1 is the schematic diagram of the revision 1 prototype of the accelerometer. A laser pointer was used for the light source and LDRs for light detectors. When the body interrupts the light, the potential (voltage) in the LDR will change. This signal (either high or low) will then be fed up to parallel port that will compute for the time elapsed.

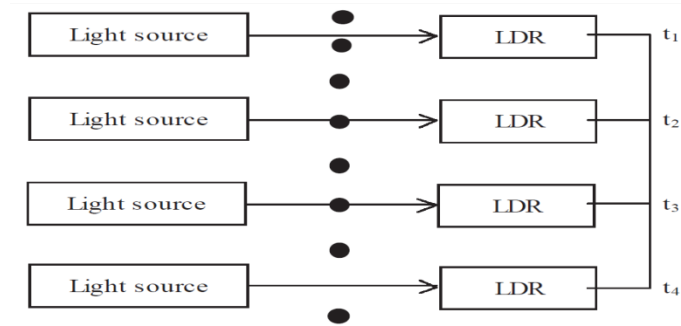


Figure 1. Effects of selecting different switching under dynamic condition

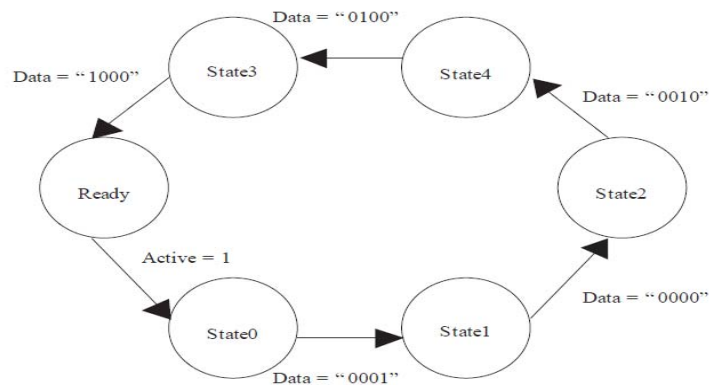


Figure 2. Effects of selecting different switching under dynamic condition

Figure 2 is the state diagram of the application. Upon running the program, it enters the READY state. This state waits for a flag that is activated by the “Active” button. When the start flag is set, the program reads the parallel port where the four light sensors are connected. The state moves to STATE1 when the ball is placed on the first sensor and moves to STATE2 upon leaving the same sensor. It moves to the different states (STATE3 and STATE4) upon interrupting the next set of light sources and detectors. When the last light is interrupted, the program calculates the time elapsed between those points and outputs the velocity v and acceleration of the body [2]-[3].

2.2. Revision 2

Figure 3 shows the schematic diagram of the acceleration measurement device. Similar to Revision 1, a laser pointer was used for the light source and phototransistor (WPT440F) for the light detector [4],[5]. The values of the sensors are constantly polled every millisecond and depending on the data taken through the parallel port, the time between the activation and deactivation of the two sensors are logged [6],[7].

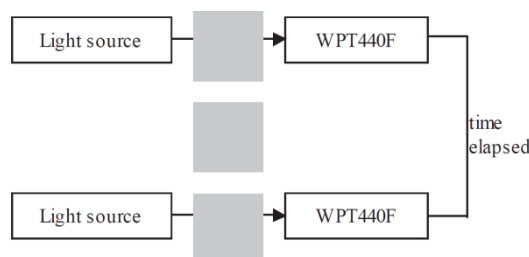


Figure 3. Schematic diagram of the accelerometer

3. DATA GATHERING

In performing the experiment for revision 1, the acceleration of a solid spherical ball was used. It was assumed that the frictional force between the ball and the inclined plane is negligible and therefore does not contribute to the torque brought about by its rotation [8],[9]. The angles 11° and 13.5° inclination for the plane were used and a 2cm diameter spherical ball. Ten trials were performed for each angle. The following formulas were used [10]:

$$a_{accepted} = \frac{5}{7} g \sin \theta$$

and

$$a_{experimental} = \frac{\sqrt{\frac{10gS \sin \theta}{7}}}{t}$$

where: g = gravitational acceleration (980 cm/s²)

θ = angle of inclination (degrees)

S = displacement of the body (cm)

The above equations were obtained by considering the rotational and linear velocity of the body as it rolls down the plane [11]-[13]. The $a_{experimental}$ is the average acceleration of the body taken from its acceleration as it passes the three other sensors.

In performing the experiment for revision 2, the acceleration of a sliding body (collision cart) was computed. It was assumed that the frictional force between the body and the inclined plane is negligible. The angles 15° and 20° were used at a traveling distance of 62 cm. Ten trials were performed for each angle. The formulas used were

$$a_{accepted} = g \sin \theta \quad \text{and} \quad a_{experimental} = 2S/t^2$$

4. RESULTS AND ANALYSIS

For revision 1, the acceleration computed by the program was compared to that of its theoretical value. An error of 13.63% was recorded when the angle is 13.5° and 16.38% when the angle is 11°. These errors were believed to be caused by the following: (1) during the release of the ball, the body is trying to overcome its moment of inertia causing its acceleration to deviate from its theoretical value; (2) just before the body reaches the last sensor, the effect of its rotational velocity becomes negligible, causing its acceleration to approach its linear acceleration only and (3) the delay (time response) of the LDR is suspected to be longer compared to that photodiode that are usually used as switch causing an error in the computed acceleration. The sources of errors in these data were used to form the second revision conditions such as the change from an LDR to a phototransistor and the change in the device being measured [14],[15].

For revision 2, the acceleration computed from the experiments was also compared to the theoretical value. A 12.58 % error was found on the 15° angle and an error of 7.20 % was found on the 20° angle. The errors were attributed to the following: (1) data from the parallel port is not directly sent to the computing side of the processor due to the inherent delays between chipsets; (2) reliability of software on low level operations are limited to the millisecond due to the nature of the programming language used (Visual C++) and (3) the capturing of the internal clock may be limited to the delays caused by the multiple cycle commands of the different Visual C++ APIs [16].

5. CONCLUSION

The prototypes in the conducted experiments were used to continually improve the methodology in creating a device that is low cost, easy to use and accurate for the experiments where it is going to be used. Revision 1 has shown the different sources of errors that are inherent in the first prototype. The sources of errors came from the object whose acceleration is being measured and the response time of the sensor (LDR). To eliminate the factor of the object trying to conquer its moment of inertia, a sliding cart was used, which is assumed to be frictionless with the inclined plane in Revision 2. A phototransistor was also used in the second revision so as to increase response time of the sensors.

The results of revision 2 have shown other sources of errors. The delay due to the different commands in the Visual C++ API is speculated to be one of the sources of errors. The lowest sampling time that Visual C++ can handle is 1 millisecond but due to the execution of other commands, the sampling time is not achieved consistently every 1 millisecond. This can be remedied by optimizing the Visual C++ code to perform the least number of necessary commands during operation and to include the use of using embedded assembly language code in the Visual C++ code to allow for more low-level control together with advanced thread synchronization. This is the planned design changes for the next revision.

Another recommendation would be to use a separate circuit in between the parallel port and the sensors as show in figure 4. The data logging circuit consists of a microcontroller whose sole function is to store the data sent by the sensors to a memory element that can later be retrieved by the computer through the parallel interface.

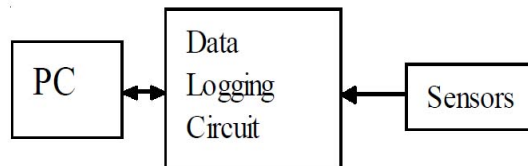


Figure 4. Block diagram of suggested configuration

REFERENCES

- [1] Sheard, Nicolas C., Larry J. Fischer, Richard W. Matthews, Himabindu Gurla, Qilin Hu, Wendy J. Zheng, and Boyle Y. Mow. "Visual data integration system and method", *U.S. Patent*, Vol. 6. Pp. 208-345, 2001.
- [2] Bateman, Andrew, and Iain Paterson-Stephens. *The DSP handbook: algorithms, applications and design techniques*, Vol. 1. Prentice Hall, 2002.
- [3] Vuletic, Miljan, Laura Pozzi, and Paolo Jenne. "Programming transparency and portable hardware interfacing: Towards general-purpose reconfigurable computing", In *Application-Specific Systems, Architectures and Processors, Proceedings. 15th IEEE International Conference on*. Pp. 339-351, 2004.
- [4] Hollabaugh, Craig. *Embedded Linux: hardware, software, and interfacing*, Addison-Wesley Professional, 2002.
- [5] Godsey, Randall D., Charles T. Walneck, James Santee, Ronald J. Gaulke, Luis R. Ortez Perez, and Alexis T. Ramos. "Interfacing system for an international-type pay-telephone", *U.S. Patent*, Vol. 5. Pp. 046-085, 1991.
- [6] Brebner, Gordon. 'A virtual hardware operating system for the Xilinx XC6200', In *Field-Programmable Logic Smart Applications, New Paradigms and Compilers*. Pp. 327-336. Springer Berlin Heidelberg, 1996.
- [7] Paulish, Daniel J., Albert J. Meyerhoff, and George R. Nickett. "Method and apparatus for interfacing stations in a multiloop communications system", *U.S. Patent*, Vol. 4. Pp. 287-592, 1981.
- [8] Clements, Alan. *Microprocessor systems design: 68000 hardware, software, and interfacing*, PWS Publishing Co., 1992.
- [9] Gajski, Daniel D., Frank Vahid, Sanjiv Narayan, and Jie Gong. "SpecSyn: An environment supporting the specify-explore-refine paradigm for hardware/software system design", *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, Vol/Issue: 6(1). Pp. 84-100, 1998.
- [10] Garcia, Philip, and Katherine Compton. "A reconfigurable hardware interface for a modern computing system", In *Field-Programmable Custom Computing Machines, FCCM 2007. 15th Annual IEEE Symposium on*, Pp. 73-84, 2007.
- [11] Millenson, John Rand Julian C. Leslie. *Principles of behavioral analysis*, New York: Macmillan, 1967.
- [12] Beach, Richard D., Robert W. Conlan, Markham C. Godwin, and Francis Moussy. "Towards a miniature implantable in vivo telemetry monitoring system dynamically configurable as a potentiostat or galvanostat for two- and three-electrode biosensors", *Instrumentation and Measurement, IEEE Transactions on*, Vol/Issue: 54(1). Pp. 61-72, 2005.
- [13] Shiakolas, Panayiotis S., and Damrongrit Piyabongkarn. "Development of a real-time digital control system with a hardware-in-the-loop magnetic levitation device for reinforcement of controls education", *Education, IEEE Transactions on*, Vol/Issue: 46(1). Pp. 79-87, 2003.
- [14] Huang, Han-Way. *PIC microcontroller: an introduction to software and hardware interfacing*, Cengage Learning, 2005.
- [15] Martin, Peter. "A hardware implementation of a genetic programming system using FPGAs and Handel-C", *Genetic Programming and Evolvable Machines*, Vol/Issue: 2(4). Pp. 317-343, 2001.
- [16] Chou, Pai, Ross B. Ortega, and Gaetano Borriello. "Interface co-synthesis techniques for embedded systems", In *Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design, IEEE Computer Society*. Pp. 280-287, 1995.