E-ISSN 2656-8039

INTERNATIONAL JOURNAL OF ENVIRONMENT, ENGINEERING & EDUCATION

www.ijeedu.com

Vol. 1, No. 1, 2019

Research Article

Microcontroller Implementation for Automatic Smart Bell

Muhammad Sabri Annas[1*], Haripuddin[1], Satria Gunawan Zain[2]

[1]Department of Information Engineering Education, Universitas Negeri Makassar, Makassar 90222, Indonesia
[2]Department of Electrical Engineering Education, Universitas Negeri Makassar, Makassar 90222, Indonesia
*Corresponding author, contact email: muhammadsabriannas@gmail.com

Received: *January 14, 2019;* ***Accepted:*** *March 29, 2019;* ***Published:*** *April 6, 2019.*

Abstract: This study aims to design automatic school bell hardware based on the ATmega32 microcontroller, design automatic school bell software based on the ATmega32 microcontroller using Basic programming language, and test the performance of the automatic school bell based on the ATmega32 microcontroller. This automatic school bell uses a minimum system of ATmega32 microcontrollers which equipped with data input from the push button, LCD to display character output, ISD2560 to output sound, and a timer taken from DS1307 RTC. This research is a Research or Development (R & D) research. This research is carried out through two stages of design. The first stage is the stage of hardware design. The second stage is the stage of software design. Programs are made using the Basic programming language using the Bascom-AVR application. The results of the product were tested using the black box method with two stages. In the first stage, software testing uses a series of simulations on the Proteus 7 Professional application. The second stage, testing hardware.

Keywords: ATmega32, Proteus 7, Real Time Clock, Sound System, Time Display.

1. Introduction

The education process will not succeed if there is no application of discipline for students. Discipline deals with order and order [1]. The order can be interpreted as someone's adherence to following the rules because they are driven by something that comes from outside themselves. Discipline is obedience that arises because of the awareness and encouragement of that person whereas orderly means the regulatory device that applies to create orderly and orderly conditions [2]. To instill discipline in schools, adequate facilities and infrastructure are needed. One means to instill discipline in school is the bell. Every school generally has a bell. The bell is used as a marker for the time of entering school hours, the time of changing hours of study, rest periods and the end of class hours. By setting these times, it expected that students could be disciplined with time [3].

There are various types of school bells, but generally, still, use human power to ring them. But often picket teachers or officers on duty to ring the bell are negligent. As a result, time management is disrupted and can interfere with the teaching and learning process. For this reason, now many are offered software-based school bells. However, software-based school bell operational costs are expensive. Also, the loudspeakers used must be turned on manually before the teaching and learning process begins.

The development of computer technology is currently very rapid, almost every aspect of our lives has experienced computerization so that everything can be completely automated [4]. With a combination of computers and microcontrollers, we can create a tool that is used to control electronic equipment. The microcontroller is an electronic component that can work by a program that loaded into memory like a very simple computer. It makes it very possible to use a microcontroller as a control center. Progress in technology will facilitate human work.The development carried out is to integrate hardware and software. With a microcontroller, school bell and loudspeakers used can control. So, the school bell can turn on and turn off automatically according to the time

Citation this article: M. S. Annas, Haripuddin, S. G. Zain, “Microcontroller Implementation for Automatic Smart Bell,” *Int. J. Environ. Eng. Educ.*, vol. 1, no. 1, pp. 34-39, 2019.
DOI: https://doi.org/10.5281/zenodo.2633673

specified. Operational costs for using microcontroller-based school bells are cheaper [5].

Research conducted by [6]–[9], shows that the use of a microcontroller as a component that can work well by the program entered. The use of ATmega Microcontrollers is very suitable to be applied in terms of automation, apart from the quite economical price, the ability is considered enough to store orders that are not too much and complicated, besides the ATmega Microcontroller is very easy to obtain on the market.

The bell is a device that can emit sound and has a function as a code, a reminder and a communication tool — examples of simple bells such as bells and bells. Following its function, the bell is one of the essential facilities in the school. The bell is used as a marker for the time of entering school hours, the time of changing hours of study, rest periods and the end of class hours. By setting these times, it expected that students could be disciplined with time [10].

Along with the development of technology, many electronic bells now developed. This automatic bell serves to give a change of time for each lesson, and there is no longer a picket who forgets not to ring the bell. This automatic bell will ring at the right time, so that the change of lesson no longer becomes chaotic, even though the duration between one lesson and another lesson varies. The purpose of the research is to make bell control automatically with several other additional hardware.

2. AVR Microcontroller

The microcontroller is a particular purpose computer in an IC that contains the CPU, memory, timer, serial and parallel communication channels, input/output ports, ADC (Analog Digital Converter). The microcontroller is used for a task and runs a program [11].

2.1. ATmega32

ATmega32 AVR Microcontroller is an 8-bit microcontroller from Atmel from the AVR family. This microcontroller is designed based on the AVR RISC architecture (Reduced Instruction Set Computer) which executes one instruction in one clock cycle so that it can achieve instruction execution of 1 MIPS (Million Instruction Per Second) every 1 MHZ clock frequency used by the microcontroller. The clock frequency used can be set via the fuse bits and crystals used. If the crystal used at 16 MHZ, so the clock frequency is 16 MHZ, the execution of the instructions reaches 16 MIPS [12].

ATmega32 has the main features including 16K x 16-byte In-System Programmable Flash Program memory from addresses 0000H to 3FFFH. This flash memory divided into two parts, namely the application flash section and boot flash section. Memory data is 2144 bytes divided into 32 general purpose registers, 64 I/O registers, and 2KB internal SRAM (Static Random Access Memory), 1 KB EEPROM (Electrically Erasable Read Only Memory), 32 I/O pins, three timer units/counter, internal and external interrupt, USART (Universal Synchronous and Asynchronous Receiver Transceiver), TWI (Two-wire Serial Interface), eight-bit ADC (Analog to Digital Converter), SPI (Serial Programmable Interface), watchdog timer, and the internal clock generator. As mentioned above ATmega32 has 32 general purpose registers, and this register is directly connected with the ALU (Arithmetic Logic Unit) so that two registers can simultaneously be accessed in one instruction that executed each clock. So this kind of architecture is more efficient in program code execution and can achieve execution ten times faster than the CISC microcontroller (Complete Instruction Set Computer) [12].

2.2. Block Diagram ATmega32

ATmega32 AVR microcontroller uses RISC technology where instruction sets are reduced in width so that all instructions have 16-bit lengths and most directions executed in a single clock, as well as a reduction in the complexity of addressing [13]. The AVR microcontroller uses the Harvard architecture by separating memory and bus lines for programs and data to improve capabilities because it can access memory programs and memory data simultaneously.

The AVR microcontroller has fast access register files with 32 register x 8 bits. With 32 registers the AVR can execute several single cycles. 6 of the 32 registers available can use as indirect address registers for 16 bits for addressing data space that allows efficient address calculation.

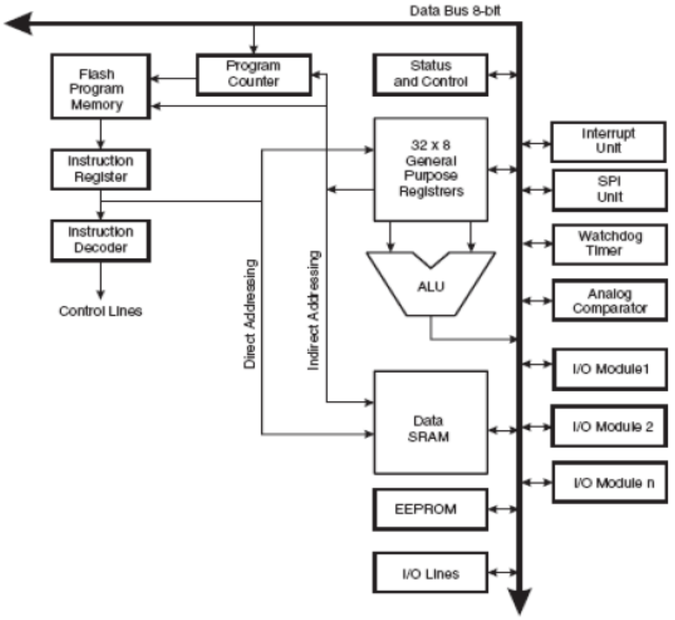


Figure 1. ATmega32 AVR Microcontroller Architecture

3. Research Methods

3.1. Product Design

The design of electronic bell products based on ATmega32 microcontrollers includes designing and testing tools. The implementation of the design was carried out at the Laboratory of Informatics and Computer Education Study Program at Makassar State University (Universitas Negeri Makassar).

The design of this automatic bell program uses a minimum ATmega32 microcontroller system that equipped with input data from the push button, LCD to display character output, IC ISD2560 to output sound, and a timer taken from DS1307 RTC IC. The bell product design can be seen in Figure 2 as follows:

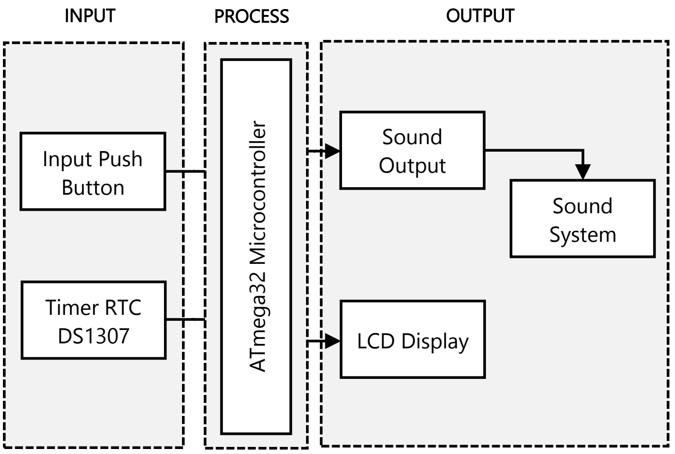


Figure 2. Design of automatic bell products with ATmega32 Microcontroller.

3.2. Design Procedure

This research is carried out through two stages of design. The first stage is the design phase of hardware equipped with ATmega32, push button, LCD, RTC 1307, and ISD2560 sound IC. The stages of designing hardware from the school bell can be seen in Figure 3 as follows:

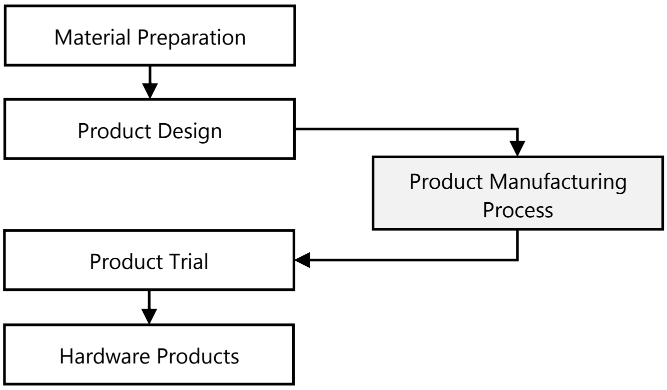


Figure 3. Hardware Design Stages

The second stage is the stage of software design. The author designed the automatic school bell circuit based on the ATmega32 microcontroller using the Proteus 7 Professional application [14]. This circuit is used to simulate the ATmega32-based automatic school bell program. Programs are made using the Basic programming language using the Bascom-AVR application. The results of the program created will be saved in a file with the extension .bas. The .bas file will be compiled into a hex extension file using the Bascom-AVR application [15].

The hex extension file is used to run the program in the Proteus application by uploading the file in the simulation circuit. With the file. Also, the hex file also used when testing the automatic school bell based on the ATmega32 microcontroller using a series of previously designed hardware. The stages of designing software from the school bell can be seen in Figure 4 as follows:

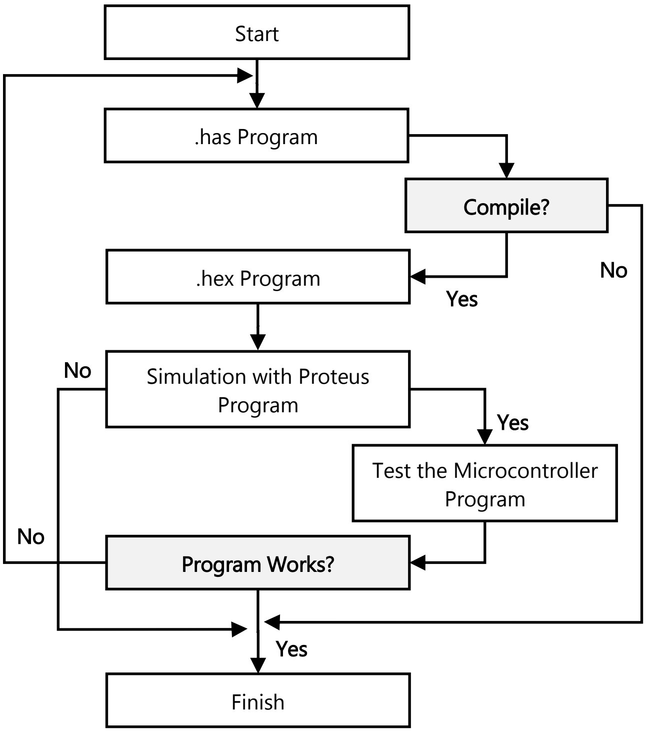


Figure 4. Software Design Stages

3.3. Product Trial

The product results of this study will test in two stages of testing. The first stage is the testing phase using a simulator. The simulator will be created using the Proteus application. At this stage, the software will be tested. The product will be tested at the hardware level. In this test, software that has tested previously will be recorded/downloaded to the ATmega32 microcontroller using the ProgISP application.

4. Results and Discussions

4.1. Hardware Design

The design phase of the design planned for this product series using the Altium Designer Winter 09 application. Figure 5 is a schematic circuit, and Figure 6 is a PCB design using the Altium Designer Winter 09 application [16]–[18]. The results of this design are printed using Kingstrik paper. The printout transferred to the PCB board by ironing the printout on the PCB board. After the circuit design moves to the PCB board, the board soaked with water. After that, dissolve the PCB solution with hot water then insert the PCB board while shaking the container. This process is to remove the copper layer on an unused PCB board. After that, rinse the PCB board with water until clean.

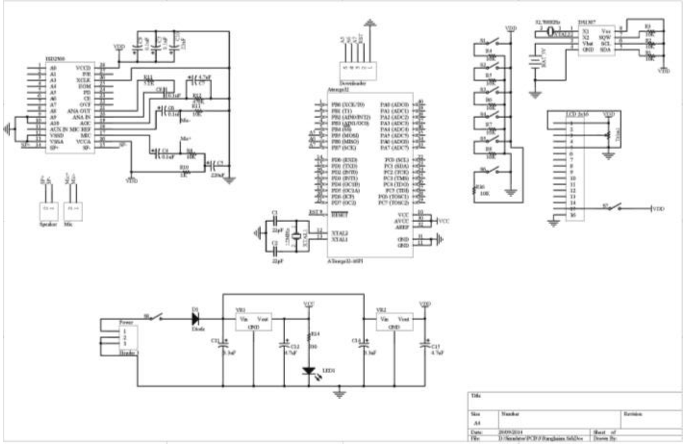


Figure 5. Schematic design using the Altium Designer Winter 09 application.

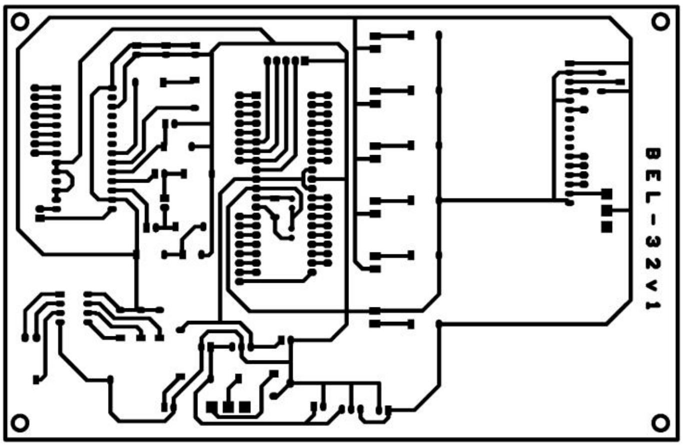


Figure 6. PCB design using the Altium Designer Winter 09 application.

The next step is to puncture the PCB board using the drill according to the design; then the PCB board is rubbed using fine sandpaper until the copper layer is visible. It is so that the tin is easily attached when soldering. Next, installing the circuit components according to the design then connects the input/output pins on the ATmega32 to the DS1307 IC, the ISD2560 IC, the LCD circuit, and the push button circuit.

4.2. Product Testing

The previously made circuit will be tested for the connection of each pin at the product trial stage. At this stage, the author discovers design errors, namely the VCC pin and ground pin on the LCD circuit exchanged. The writer then removes the path of each pin and connects the pins using a cable. Also, several lines are cut off and then connect these lines using tin.

4.3. Hardware Product

The results of the design of automatic school bell hardware based on the Atmega32 microcontroller are a series of these products. Figure 7 is a display of a circuit on a PCB board.

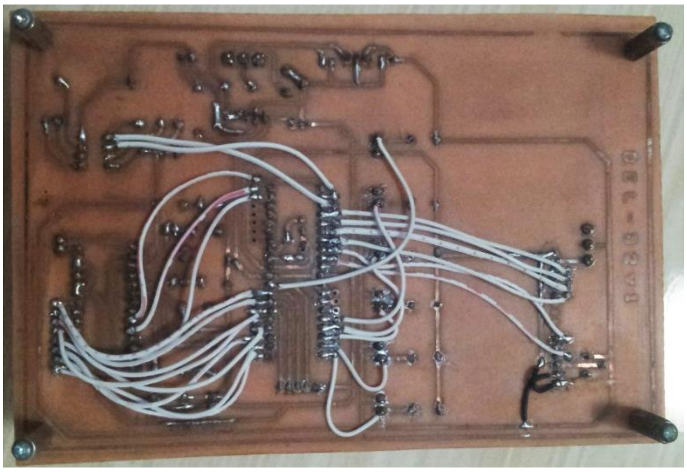


Figure 7. Automatic school bell circuit based on the Atmega32 microcontroller.

Figure 8 is a display of circuit components installed on a PCB board visible from above.

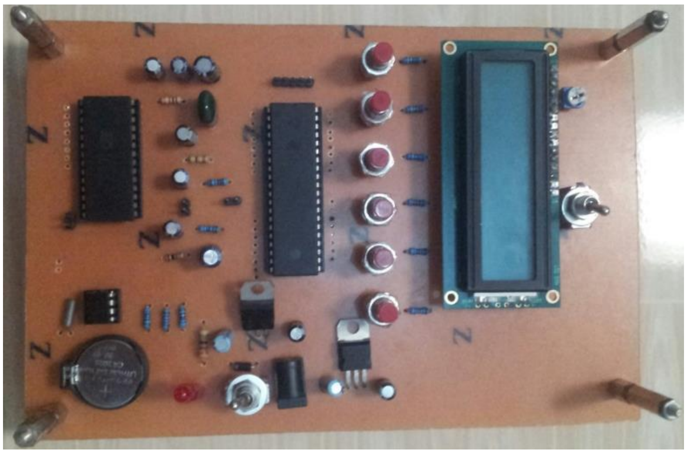
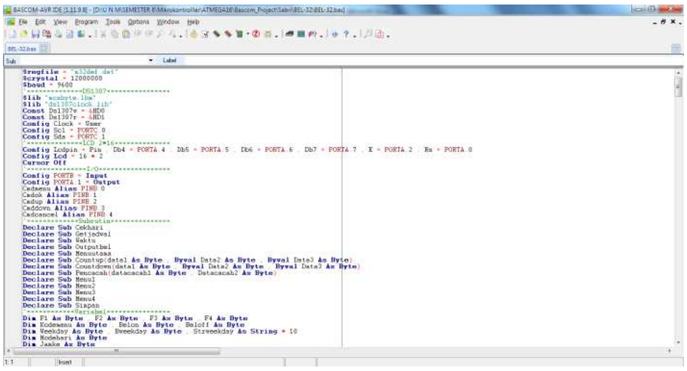


Figure 8. The automatic school bell component is based on the Atmega32 microcontroller.

4.4. Software Design

The stages of software design produce programs from electronic bell products based on the ATmega32 microcontroller. The author designed the program using Basic with the Bascom-AVR application.

Microcontroller based automatic bell program. The results of the program created will be saved in a file with the extension .bas. The .bas file will be compiled into a hex extension file using the Bascom-AVR application. Figure 9 is the display of the automatic school bell program based on the ATmega32 microcontroller in the Bascom-AVR application.


BASCOM-AVR IDE [1.11.9.8] - [D:\U N M\SEMESTER 6\Mikrokontroller\ATMEGA16\Bascom_Project\Sabi\BEL-32\BEL-32.bas]
File
Edit
View
Program
Tools
Options
Window
Help
BEL-32.bas
Sub
Label
1:1
Insert

```
$regfile = "m32def.dat"
$crystal = 12000000
$baud = 9600
'**************DS1307***************
$lib "mcsbyte.lbx"
$lib "ds1307clock.lib"
Const Ds1307w = &HD0
Const Ds1307r = &HD1
Config Clock = User
Config Scl = PORTC.0
Config Sda = PORTC.1
'**************LCD 2*16**************
Config Lcdpin = Pin , Db4 = PORTA.4 , Db5 = PORTA.5 , Db6 = PORTA.6 , Db7 = PORTA.7 , E = PORTA.2 , Rs = PORTA.0
Config Lcd = 16 * 2
Cursor Off
'***************I/O*****************
Config PORTB = Input
Config PORTA.1 = Output
Cadmenu Alias PINB.0
Cadok Alias PINB.1
Cadup Alias PINB.2
Caddown Alias PINB.3
Cadcancel Alias PINB.4
'*************Subrutin**************
Declare Sub Cekhari
Declare Sub Getjadwal
Declare Sub Waktu
Declare Sub Outputbel
Declare Sub Menuutama
Declare Sub Countup(data1 As Byte , Byval Data2 As Byte , Byval Data3 As Byte)
Declare Sub Countdown(data1 As Byte , Byval Data2 As Byte , Byval Data3 As Byte)
Declare Sub Pencacah(datacacah1 As Byte , Datacacah2 As Byte)
Declare Sub Menu1
Declare Sub Menu2
Declare Sub Menu3
Declare Sub Menu4
Declare Sub Simpan
'************Variabel***************
Dim F1 As Byte , F2 As Byte , F3 As Byte , F4 As Byte
Dim Kodemenu As Byte , Belon As Byte , Beloff As Byte
Dim Weekday As Byte , Bweekday As Byte , Strweekday As String * 10
Dim Modehari As Byte
Dim Jamke As Byte
```

Figure 9. Display of automatic bell program based on the ATmega32 microcontroller.

The automatic bell program based on the ATmega32 microcontroller is then compiled into hex extension files using the Bascom-AVR application. Next, the author designed a series using the Proteus 7 Professional application to simulate an automatic school bell program based on the ATmega32 microcontroller that had made previously. Figure 10 is an automatic school bell circuit based on the ATmega32 microcontroller on the Proteus 7 Professional application.

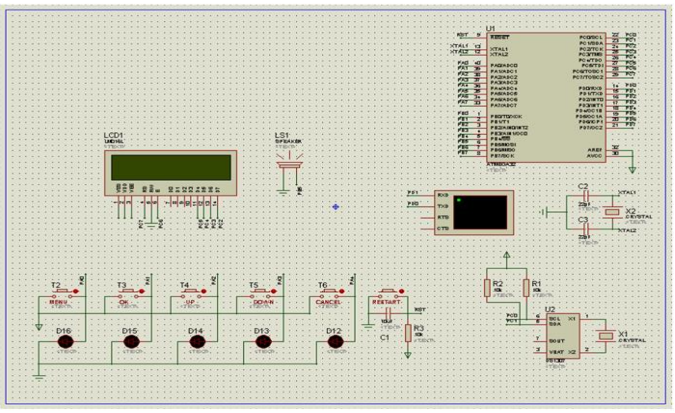


Figure 10. Automatic bell circuit based on the ATmega32 Microcontroller on the Proteus 7 Professional application.

4.5. Product Testing Results

Product testing in the first stage is testing the automatic school bell software based on the Atmega32 microcontroller using the Black Box testing method. In this test, the program will be tested using a series designed in the Proteus 7 Professional program. Based on this stage, test results will be produced according to the schedule that has set at the automatic school bell based on the Atmega32 microcontroller with a bell ring schedule in the simulation circuit. Table 1 is a schedule conformity test.

Product testing in the second stage is testing the automatic school bell hardware based on the Atmega32 microcontroller. In this test, the microcontroller based automatic school bell circuit will be tested using the Black Box testing method. Based on this stage, the schedule of conformity testing data will be generated on the automatic school bell based on the Atmega32 microcontroller with an automatic school bell sound schedule based on the Atmega32 microcontroller on the automatic school bell found on the Atmega32 microcontroller. Table 2 is a schedule conformity test.

Table 1. The first stage of testing is schedule conformity.

No.	Period	Sound Time	Result	
			Exactly	Not Exactly
1	Start On	07.05	✓	
2	First hour	07.15	✓	
3	Second hour	08.00	✓	
4	Third hour	08.45	✓	
5	Fourth hour	09.30	✓	
6	First break	10.15	✓	
7	Fifth hour	10.30	✓	
8	Sixth hour	11.15	✓	
9	Seventh hour	12.00	✓	
10	Second break	12.45	✓	
11	Eighth hour	13.00	✓	
12	Ninth hour	13.45	✓	
13	Go home	14.30	✓	
14	Finish Off	14.40	✓	

Table 2. The second stage of testing is schedule conformity.

No.	Period	Sound Time	Result	
			Exactly	Not Exactly
1	Start On	07.05	✓	
2	First hour	07.15	✓	
3	Second hour	08.00	✓	
4	Third hour	08.45	✓	
5	Fourth hour	09.30	✓	
6	First break	10.15	✓	
7	Fifth hour	10.30	✓	
8	Sixth hour	11.15	✓	
9	Seventh hour	12.00	✓	
10	Second break	12.45	✓	

No.	Period	Sound Time	Result	
			Exactly	Not Exactly
11	Eighth hour	13.00	✓	
12	Ninth hour	13.45	✓	
13	Go home	14.30	✓	
14	Finish Off	14.40	✓	

It is rare for schools to use school bells automatically which can work alone ring the bell according to the schedule of lessons available. The work process of this tool is to set the current time clock, then enter the bell list according to the schedule. After that, the microcontroller will read the bell data list today that has been stored repeatedly and compares it with the current time. If there is a match of the data hour and minute of the current time with the bell list today, the microcontroller will send a command to the relay to connect the relay switch so that the round bell rings. The same study was conducted by Ohize [19], that with this design, it obtained excellent use applied at the level of elementary and secondary schools where the teaching section could be on time during the desired period including rest. The extraordinary advantage is that the bell is lined up at the beginning of each period without human intervention, so it is very appropriate.

5. Conclusions

The design of automatic school bell hardware based on the ATmega32 microcontroller through the preparation phase of tools and materials, PCB design using the Altium Designer Winter 09 application which then printed on the PCB board. The results of the automatic school bell hardware design based on the ATmega32 microcontroller are a series of these products.

The design of automatic school bell software based on the ATmega32 microcontroller through the program design stage using the Basic programming language in the Bascom-AVR application and designing simulators using the Proteus 7 Professional application. The results of the automatic school bell software design based on the ATmega32 microcontroller are programs of the product.

The ATmega32 microcontroller automatic school bell product was tested to determine the suitability of the schedule on the program with the schedule on the product. The test method uses the black box method with two stages. The first stage is the testing phase using a simulator. The test results in the first stage are the sound of the bell, the bell lights automatically, and the automatic bell dead according to schedule. In the second stage, the product will be tested at the hardware level. The test results in the second stage are the bell; the bell lights automatically, and the dead bell automatically on a schedule.

Acknowledgments

This work is supported Laboratory of Informatics and Computer Education Study Program at Makassar State University (Universitas Negeri Makassar).

References

[1] D. D. Gehring, "The objectives of student discipline and the process that's due: Are they compatible?," *NASPA J.*, vol. 38, no. 4, pp. 466–481, 2001.

[2] C. M. Golde and G. Walker, "Preparing stewards of the discipline," *Envisioning Futur. Dr. Educ. Prep. Steward. Discip.*, pp. 3–20, 2006.

[3] B. Grant, "Disciplining students: The construction of student subjectivities," *Br. J. Sociol. Educ.*, vol. 18, no. 1, pp. 101–114, 1997.

[4] K. Cooper, *Rapid prototyping technology: selection and application*. CRC press, 2001.

[5] J. Habiyaremye, "Labview Based Remote Controlled Automatic & Manual Bell For Boarding and Day Schools," in *The International Journal Of Engineering And Science (IJES)*, vol. 2, no. 7, Department of Electronics and Instrumentation, Bharathiar University, 2013, pp. 96–102.

[6] A. Djajadi, R. Rusyadi, T. Handoko, M. Sinaga, and J. Grueneberg, "Analysis, Design and Implementation of an Embedded Realtime Sound Source Localization System Based on Beamforming Theory," *Telkomnika*, vol. 7, no. 3, p. 151, 2009.

[7] M. F. Ruzaij and S. Poonguzhali, "Design and implementation of low cost intelligent wheelchair," in *2012 International Conference on Recent Trends in Information Technology*, 2012, pp. 468–471.

[8] M. M. Haque, "AVR microcontroller Implementation for customized Sound generation," *Int. J. Electr. Comput. Eng.*, vol. 2, no. 1, pp. 112–119, 2012.

[9] M. Stager, P. Lukowicz, and G. Troster, "Implementation and evaluation of a low-power sound-based user activity recognition system," in *Eighth International Symposium on Wearable Computers*, 2004, vol. 1, pp. 138–141.

[10] S. Wang and Z. Chen, "Design of Automatic Bell System Based on Campus Network [J]," *Sci. Technol. Eng.*, vol. 31, 2011.

[11] F. Eady, *Networking and internetworking with microcontrollers*. Elsevier, 2004.

[12] Atmel, "Microcontroller with 32 KBytes In-System Programmable Flash," *Microchip.com*, 2011. [Online]. Available: https://www.microchip.com/wwwproducts/en/ATmega32U4. [Accessed: 28-Mar-2019].

[13] W. Vanhaverbeke and N. G. Noorderhaven, "Competition between alliance blocks: The case of the RISC microprocessor technology," *Organ. Stud.*, vol. 22, no. 1, pp. 1–30, 2001.

[14] W. F. X. C. X. Bin, "Application of Proteus and Keil on opened experiment of microcontroller [J]," *Electron. Meas. Technol.*, vol. 6, 2008.

[15] C. Kuhnel, *BASCOM Programming of microcontrollers with ease: An introduction by program examples*. Universal-Publishers, 2001.

[16] H. Zhang and P. Zhang, "Integrated Design of Electronic Product Based on Altium Designer [J]," *Radio Commun.*

Technol., vol. 6, p. 19, 2008.

[17] A. A. Beltran Jr, C. D. Hiwatig, N. J. R. Laguna–Agustin, and M. B. Villanueva, “Teaching Electronic Circuits Using Altium Designer,” *Int. J. Sci. Eng. Technol.*, vol. 3, no. 10, pp. 1239–1243, 2014.

[18] Y. Niu, C. Zhu, and X. Rui, “Application of Altium Designer Software in Print Circuit Design [J],” *Electron. Sci. Technol.*, vol. 8, 2011.

[19] H. O. Ohize, E. N. Onwuka, and A. Ibrahim, “Design of Microcontroller-Based Automatic School Bell,” *AU J. Technol.*, vol. 15, no. 2, 2011.