Embedded Automatic Scheduling System

Damir Bilić
Faculty of Engineering and Information Technologies
International Burch University
Sarajevo, Bosnia and Herzegovina
damir.bilic@ibu.edu.ba

Tarik Uzunović
Faculty of Electrical Engineering
University of Sarajevo
Sarajevo, Bosnia and Herzegovina
tuzunovic@etf.unsa.ba

***Abstract*—This paper describes an embedded system for the automation of scheduling systems such as school bells, factory shift changes, military drills etc. The system consists of two parts: a remote node used for remote control and setup of the system and the real time actuator node which controls a physical object, for example, a bell system. The hardware and software structure are illustrated in detail for both parts of the system through the implementation as an automatic school bell system.**

***Keywords*— Embedded system, automated bell system, school bell system**

I. Introduction

During the last few decades, advancements in technologies have made it possible to automate various tasks and jobs, ranging from repetitive industrial processes to personal home automation. There is a tendency to free people from execution of tasks that have to be extensively repeated for many times.

However, technology in the educational sector, especially in primary education, has been employed only in the recent years in Bosnia and Herzegovina. There have been 1010 elementary schools in Bosnia before 1946 [1], and around additional 1500 have been built until the 1980's [2]. Most of these schools have not been adapted with new technologies that have become available, mainly due to economic reasons.

An obvious example of the lack of technology in the educational sector in Bosnia are manual ringing bells used in elementary and high schools. The start and end of class or breaks are indicated by ringing bells. These bells currently need to be operated manually in most of these schools built before the 70's and 80's.

The idea of automation of scheduling systems, ranging from school bells to tower bells has been extensively developed by independent companies and developers [3, 4]. These systems are high cost, require changes to be made on the current infrastructure or have a limited amount of bells that can be connected, which requires the purchase of additional components in order to operate a large scale system [5].

Low cost and low power devices for automation of school bells have been already described, however they do not allow easy adjustment of scheduling times by the end user [6]. The issue of end user access simplification is addressed in this paper.

This paper proposes as system that will allow the use of the current infrastructure to automate these bells, and eliminate the need to always have someone stand close-by if it is time to ring.

The major contribution is the introduction of a simple low cost system that is simple to use and maintain. The system is made to be installed into the current infrastructure, making the installation simple and fast.

The following section presents the detailed system structure, describing the two separate parts of the system in detail. After description of the structure, the system behavior for each of the two nodes is presented with details from its practical implementation. Finally, practical and theoretical aspects for the complete systems are discussed and directions for future work are given.

II. System description

The Embedded Automatic Scheduling System proposed in this paper consists of two clearly separable parts:

- Real time actuator node
- Remote control node

The Real time actuator node is a small, compact device intended to be installed in place of the current manually operated switch. The role of this device is to activate the bells at required times. A human machine interface consisting of a small keyboard and display is available to the user allowing for complete control of the device, that is, adding new and modifying current schedules, setting or changing the clock time etc.

Remote control node allows for complete remote control of the actuator node. It is intended to be connected to personal computers over USB.

The PC to which the remote control node is connected, is running an application that offers the same functionality as the human machine interface on the field mounted actuator node. Communication with the Actuator node is established over RF transceivers, allowing the users to adjust the schedule times or ring the bells by a simple interface on their computers.

The Real time actuator node can be functioning as a stand-alone device, it does not require the presence of the remote control node in order to operate correctly. The remote control node is intended as an addition to the actuator node, nevertheless it presents an integral part of the whole system as it eases the access and control and does not increase the cost of the complete system significantly. General block structure of the system is shown in Fig. 1.

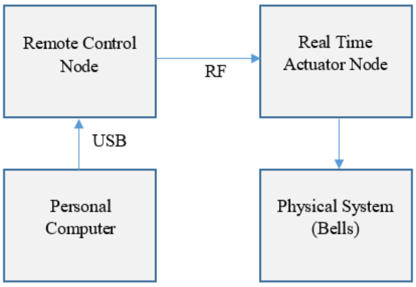


Fig. 1. General system structure

In case that the required application consists of multiple bells that have to be operated independently, it is possible to install multiple actuator nodes on each bell that will be controlled by one control node.

III. Hardware Structure of the System

As mentioned in Section II, the general system consists of two main parts, their detailed hardware structure will be discussed in detail.

A. *Real Time Actuator Node*

The actuator node is intended to be mounted into the wall socket instead of the regular switch used for manual operation. Having minimal space requirements as compared to other solutions for the automation of school bell systems [3, 4]. The hardware structure of the actuator node is shown in Fig. 2 and implemented prototype in Fig. 3.

The prototype device is implemented with help of the Arduino Mini Pro development platform [7]. The Arduino is equipped with an Atmega328P microcontroller that allows use of a large number of features while keeping the power consumption low in case the need for complete battery operation of the device [8].

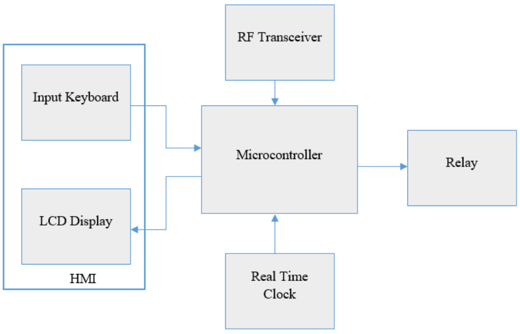


Fig. 2. Real Time Actuator Node Hardware Structure

The Atmega328 is an 8-bit microcontroller with 32KB of flash memory, 2KB SRAM and up to 23 GPIO lines running at 16MHz. Its five software selectable power saving modes allow efficient energy consumption [8].

Accurate time keeping is accomplished by using an external DS1307 real time clock circuit [9]. The real time clock has its own separate battery backup that allows accurate time tracking even when there is a power shortage. In addition to the clock, it features a calendar function, which can be used to disable the school bell system automatically during weekends, specific dates (state or religion holidays for example), etc.

The RF transceiver allows the communication between the Remote control node and the Actuator node, in case there is need for remote control. By using the low power and low cost nRF24L01+ chip the range is around 30m indoors, while this range can be greatly extended when the devices are placed in line of sight [10].

Time setup and schedule control on the device are made possible by using a 48x84 px LCD display to show the menu with instructions and an 4x4 keyboard which allows complete control of the device.

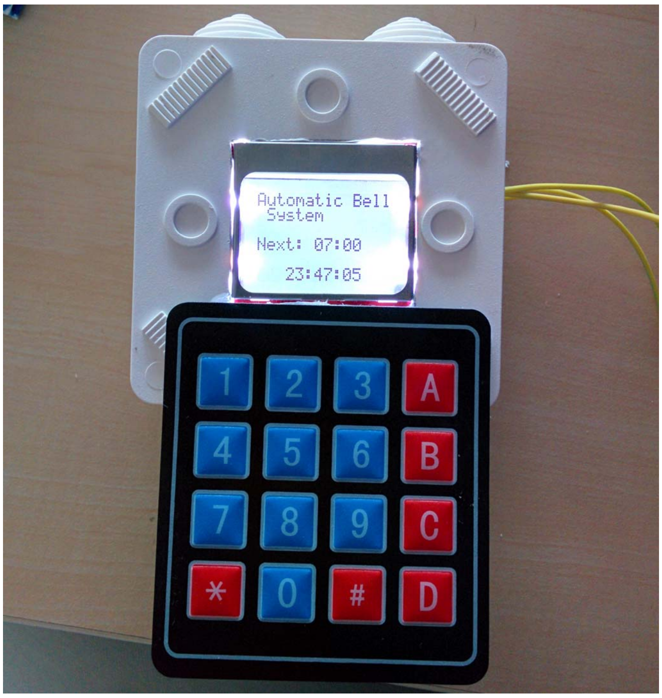


Fig. 3. Real time actuator node prototype

The main idea behind keeping the system adaptable to the current infrastructure is based on using a relay device to replace the manual switch. Relays basically behave like regular switches but they can be simply controlled by a microcontroller.

Combining this fact with the very small dimensions of the Real time actuator device allows for complete implementation of the system virtually without any significant changes to the current infrastructure.

The current prototype can be powered by any DC source with output ranging from 9V to 12V. The power and battery requirements will be further discussed in later sections.

B. *Remote Control Node*

The Remote Control Node can be connected to any personal or office computer in reasonable proximity to the place where the Actuator node is mounted. The hardware structure of the Remote Control Node is shown in Fig. 4. and prototype implementation in Fig. 5.

It is accompanied by a user application that currently works only on Windows operating systems. The user interface of the application is shown in Fig. 6.

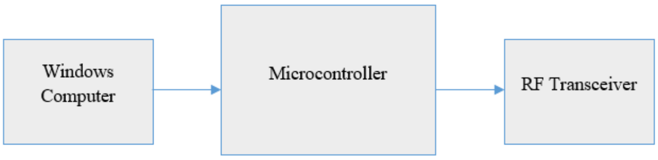


Fig. 4. Remote Control Node Hardware Structure

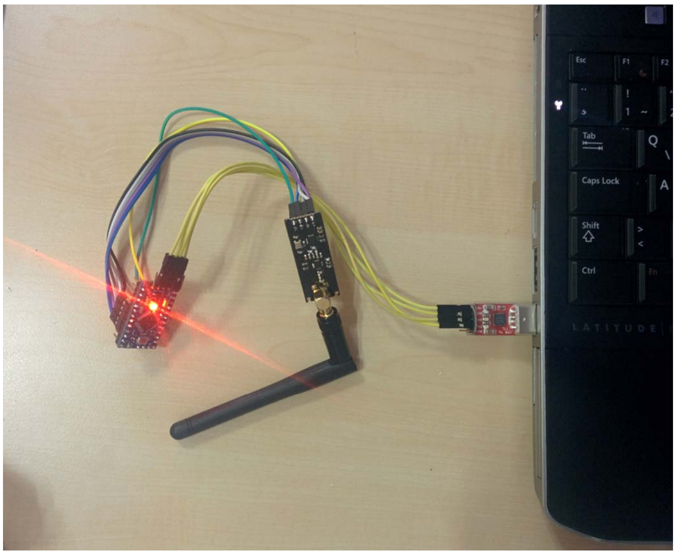


Fig. 5. Remote Control Node Prototype

In order to reduce the cost, again the Arduino Mini Pro development platform is used for controlling the same type of RF transceiver as in the Real Time Actuator Node.

The user application in combination together with the remote control node allows for remote control and setup of any of the parameters of the actuator node: adjusting the correct time, entering custom schedules, activating predefined schedules, etc.

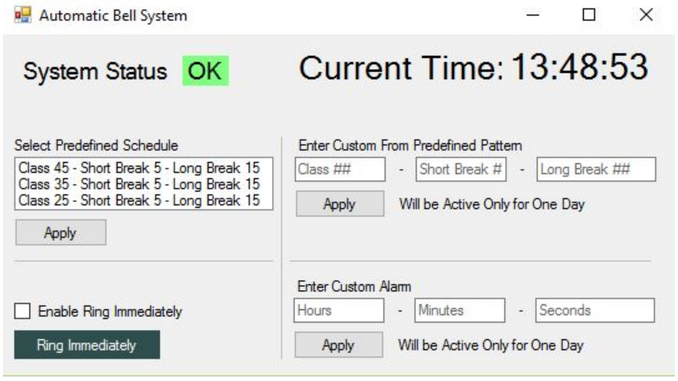
Communication between the microcontroller and the computer is established over the USB, so it is enough to plug in the device to the computer over USB without the need of any additional equipment.

IV. Software Structure

As mentioned, there are two ways of Actuator node control, either directly using the human machine interface (keypad and LCD) or remotely by using the remote control node and the Windows application. The behavior of the system and its functions will be elaborated in this section.

A. *Real Time Actuator Node*

By using the keypad on the Actuator Node it is possible to completely control all functions of the device. Basic functionalities include selecting predefined schedules, entering durations of classes and breaks while using predefined patterns, or entering completely custom schedules that will ring at defined times, including the possibility to ring the bell once immediately, preserving the functionality of the initial mechanical system. All functions of the actuator node are shown in Fig. 7.



Automatic Bell System

System Status OK

Current Time: 13:48:53

Select Predefined Schedule

Class 45 - Short Break 5 - Long Break 15
Class 35 - Short Break 5 - Long Break 15
Class 25 - Short Break 5 - Long Break 15

Enter Custom From Predefined Pattern

Class ## - Short Break # - Long Break ##

Will be Active Only for One Day

□ Enable Ring Immediately

Enter Custom Alarm

Hours - Minutes - Seconds

Will be Active Only for One Day

Fig. 6. Windows Application User Interface

In order to keep the system user friendly and simple to operate, all types of regularly used schedules are already predefined. For example, the basic and most used schedule is with 45min for class time, 5 min for short breaks and 15 min for the long break. The user may adjust the schedule in just a few short steps instead of having to reconfigure the whole device, as it is case for some of the current systems used for school bell automation.

Additionally, it is possible to add individual ring times at any time or completely custom schedules to the system. In case of adding individual ring times, you may disable the regular schedule and create a completely custom one or add individual ring times to the current running schedule. In either case, the custom schedule will be active only for one cycle, meaning it will be active only for one day as these special school events usually require modifications only for one day.

The system offers a possibility to disable the device completely until activated again, this is useful in vacation days, holidays or any other day where it is not necessary to ring the bells.

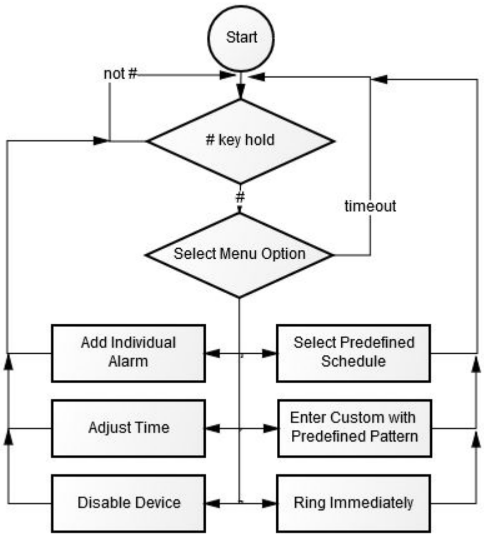


Fig. 7. Actuator Node Function Strucutre

Password protection of access to the system via the human machine interface is optional and can be enabled if required.

It is possible to adjust all settings during runtime without affecting operation of the device. No changes are made to the system during operation over human machine interface until the settings are confirmed at end of input.

The activation of bells is controlled using interrupts, allowing correct operation even when the device is being operated by the user.

Ringing times are stored in non-volatile memory in form of lookup tables, allowing correct operation of the device even when power outages happen. Custom schedules and individual ring times are stored only until beginning of the next day, as these are usually never required to be active longer than one day. These custom schedules are enabled for example during holidays.

When the actuator node receives commands from the remote control node, it sends the current stats of the system back to the actuator node. This mechanism is used to confirm transmission of correct data on the remote side.

B. *Remote Control Node*

The Remote Control Node allows full control of real-time actuator node by using any personal computer.

All features that are accessible from the human machine interface are also accessible from the application, again the user has full control of the device, including custom schedules, predefined schedules, immediate ring and time adjustment.

Even though it is not necessary to use the Remote Control Node, it simplifies the operation of the device.

When commands are sent to the Real-time actuator node, a response is received from the same node. If no response is received, or the received data does not match the sent data, an error has occurred and it is indicated inside the user application as shown in Fig. 6.

V. System implementation

Initial testing was performed inside a controlled environment with constant monitoring of the system. During initial testing, the system was powered over USB cable connected to the computer. Because of the direct connection to the computer it was also possible to debug and correct errors detected in the early stages.

After successful completion of the first initial test the system was tested in the real physical environment. The system was installed in the “Mješovita srednja škola” in Maglaj during a two-day period for testing and demonstration purposes.

Response of school staff has been very good for the prototype system as it eliminates the need for one person to always be available to operate the ring control switch. Additional reason for the good response has been due to the easy operation and control of the system, because all setup steps are followed with instructions on the screen. Although, comments were made about possible changes and improvements which will be considered during future work.

Before permanent installation of the system additional testing and validation must be performed in order to guarantee correct operation.

VI. Power consumption analysis

During testing the system was powered by an 12V adapter. Though, it became obvious that it will not be always possible to power the system from an external source. Battery operation is often a requirement for this kind of devices and power consumption optimization is required.

Current consumption of the unmodified Arduino Nano Pro development board was measured to be around 23 mA with external equipment turned off and 5V applied to the internal voltage regulator. The Arduino contains a LED indicator that is always turned on when the board is on. By removing the LED it is possible to reduce the power consumption. Additionally, the MIC5205 voltage regulator on the Arduino was found to be quite inefficient. To get better efficiency, it is needed to remove the LED and the voltage regulator. It is then possible to reduce the power consumption to less than 15mA when the Arduino is powered directly from a 5V voltage source. By sending the processor to sleep mode between ringing intervals it is possible to reduce the power consumption below 100µA during sleep mode.

In case that battery operation is required, it is possible to use the MPC1700 voltage regulator with 1.6 µA quiescent current during normal operation [11]. Avoiding the inefficient internal voltage regulator.

Power consumption of the DS1307 real time clock can be neglected as it has its own battery that can according to specifications last up to 10 years [9].

The LCD display consumes 10mA when active with backlight on. However, if battery operation is required the light can be turned off and power consumption is reduced to 300µA

[12]. Additionally, the display can be turned off when no operation on the device is required, reducing the power consumption to 1μA when the LCD is off.

The RF transceiver was measured to consume 10mA during transmission at lowest bit rate or idling. As the user can send data from the control node at any time it is needed to have the RF transceivers on at all times. But constant consumption of 10mA is not efficient for battery usage.

One possibility to avoid the always on state of the RF transceiver would be to save the data from the Windows application inside a buffer at the control node. Then, when the actuator node turns on, it can send a request signal to the control node that will reply with the latest data from buffer. The transceiver on the control node then needs to always be turned on, but it is not a problem as the control node is powered from a computer.

Highest power consumption was by the relay. Its consumption was up to 75mA when active. This is very high for battery operated devices. The relay is active at most for 60 seconds during an hour, when using the shortest type of predefined schedule. At long battery operated periods this is still high. One possibility to avoid this high power consumption would be to use digital elements instead of relays.

Power consumption optimization will remain in focus during future development of the project.

VII. Discussion

The proposed embedded system consisting of two parts, actuator and remote control node, has been implemented and tested as an school bell automation system. Its effective and correct functionality have been confirmed through exhaustive testing.

More testing and validation is required to implement the system permanently in the school or any other structure that requires scheduling automation.

During testing in the real physical environment it was shown that the system indeed does not need any significant changes to the current infrastructure. Even more, the current mechanical system at site of testing is powered by 12V, meaning that there is no need for an AC/DC converter to power this prototype system. However, this will not be the case always and future work will include a converter built into the device, depending on the application.

Battery operation at the current level is not recommended as power consumption is not fully optimized, nevertheless there is great potential for optimization which is going to be the main task for future work.

VIII. Conclusion

The combination of the actuator and remote control node provide a flexible and adaptive system which is possible to use in any environment that requires some kind of scheduling automation.

The remote control node allows for complete remote control of the system. It can be controlling an individual actuator node or several nodes at once, for example to control bells at the same time even though they are not physically connected to each other.

Installation of the system without remote control is also possible in case that remote control is not a necessity and cost reduction is required.

Practical implementation of the system in the "Mješovita srednja škola" in Maglaj has been proven to be very effective and greatly improves the current scheduling system. The need for having someone always present to ring bells at defined times has been completely eliminated.

The system described in this paper provides a way of implementing new technologies in the educational sector without the need for any expensive changes in the current infrastructure.

During prototype development, the estimated cost for one system without battery backup except for the real time clock would be around 50$, with possibilities to reduce the cost if larger quantities of equipment would be ordered at once.

IX. References

[1] N. Malcolm, "Povjest Bosne", 1995.

[2] S.Hadžić, S. Selimović, "Kulturna historija Bosne i Hercegovine", Tuzla 2009. p. 351

[3] Innovationwireless, "K-12 SCHOOL Synchronized Wireless and POE Clocks" 2016. [Online] Available: http://www.innovationwireless.com/ [Accessed June 19, 2016.]

[4] Timesystems, "AutoBell Complete School Bell Solution" 2016. [Online] Available: http://www.timesystems.co.nz/ [Accessed June 20, 2016.]

[5] Visplex, "School Bell System" 2016. [Online] Available: http://www.visiplex.com/ [Accessed July 14, 2016.]

[6] R. Bhakthavatchalu, S. Mallia, S., Harikrishnan, R., Krishnan, A., and Sruthi, B., "Low power scheduled alarm system using embedded microcontroller with USB interface", in 2011 International Conference on Emerging Trends in Electrical and Computer Technology, ICETECT 2011, Chunkankadai, 2011, pp. 610-615.

[7] Arduino, "Arduino Pro Mini 328 - 5V/16MHz" 2016. [Online] Available: https://www.arduino.cc/en/Main/ArduinoBoardProMini [Accessed June, 2016.]

[8] Atmel, "Atmel picoPower ATmega328/P Microcontroller" 2016. [Online] Available: http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_datasheet.pdf [Accessed June, 2016]

[9] Dallas Semiconductor, "DS1307 64 x 8 Serial Real-Time Clock" [Online] Available: https://www.sparkfun.com/datasheets/Components/DS1307.pdf [Accessed June, 2016.]

[10] Nordic Semiconductor, "nRF24L01+ Single Chip 2.4GHz Transceiver" 2008. [Online] Available: https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf [Accessed June, 2016]

[11] Microchip "MPC1700 Low Quiescent Current LDO" 2013. [Online] http://ww1.microchip.com/downloads/en/DeviceDoc/20001826C.pdf [Accessed June, 2016]

[12] Philips Semiconductors "PCD8544 48×84 pixels matrix LCD controller" 1999. [Online] https://www.sparkfun.com/datasheets/LCD/Monochrome/Nokia5110.pdf [Accessed June, 2016]