The International Journal Of Engineering And Science (IJES)
||Volume||2 ||Issue|| 7 ||Pages|| 96-102||2013||
ISSN(e): 2319 – 1813 ISSN(p): 2319 – 1805

Labview Based Remote Controlled Automatic & Manual Bell For Boarding and Day Schools

Joseph Habiyaremye
Department of Electronics and Instrumentation, Bharathiar University

----------ABSTRACT----------

A LabVIEW based automatic or manual school bell is a project which shall help head- teachers to remotely control the ringing of a bell either automatically or manually in their schools.In this system, all bell timings and durations are predefined and depend on: School type (day or boarding)or School organization. These different timings are defined in LabVIEW's block diagram part. The front panel will be user interface, from the user interface:

i. The user(head-teacher) will start by entering the current time, from his/her watch for initialization purpose ;
ii. The user can manually switch ON the bell, for example in case of urgent meeting;
iii. The user can know the current teaching hour, the current teaching period;
iv. The user can know if the bell is ringing or not; as the bell is remotely controlled

At the end or starting of each time slot, LabVIEW will generate a signal through Arduino microcontroller which is interfaced to a Zigbee transceiver and this Zigbee will transmit this information to a second remote Zigbee which is interfaced to an AC bell through a MOSFET and Relay. The time duration for the bell to ring is defined in LabVIEW itself. The connection of Arduino to LabVIEW requires a plug in called LIFA (LabVIEW interface for Arduino) .The Zigbee transceiver are configured by help of X-CTU,TMFT or any other Terminal software by using AT commands (Zigbee series 2 are used).The Zigbee modules are configured in IO scheme; network scheme also can be used if some other controls are added to the system. This project, an installer can be done for easy distribution. This project can even be used in colleges and universities as well as in some companies

KEYWORDS: *Arduino Uno, AT commands, LabVIEW, LIFA, ZigBee Network.*

Date of Submission: 16 July 2013  Date of Publication: 7, July 2013

I. INTRODUCTION

All over the world, high schools use timetable for subjects, classrooms, teachers and time slots allocation so that they can have an organized education. In a timetable each subject is assigned a given time slot which may change according to given school. In timetable teachers are assigned different subject in different classrooms, one teacher may teach one or more than one subject in different classrooms, just in different time. A typical timetable has: all school teachers' names, all subjects' names, teaching time, break time, lunch time and it shows also the starting and the ending time for a period of one week and it is in form of table. In each and every school, there is either electrical or mechanical bell which is used to give signal for each beginning and end of each time slot and this bell is manually controlled and there must be someone called timekeeper who have to ring that bell. Sometimes, the timekeeper forgets to ring that bell which will totally disturb the teaching structure. This project comes as solution to this problem in form of " LabVIEW based remote controlled automatic & manual bell for boarding and day schools"

II. OBJECTIVE & APPROACH

To develop a LabVIEW [1] based remote controlled automatic & manual bell for day and boarding schools to be used in education field for timetable monitoring purpose, this system may also be used in some companies where the time slots distribution is required.

III. SYSTEM DESIGN

3.1 case study

This system has been designed and implemented according to ADB (Association des Anciens de DON BOSCO)[2] Nyarutarama one high school from Rwanda. In this day school:

- There are around 50m from head teacher's office to the location of the bell;
- Classes start at 07h00' am and end at 02h10'pm;
- Each time slot is 50mins wide;
- The break time is equal to 30mins, this time is even used for students to take some snacks;
- After 02h10'pm classes and offices are closed; this is an off class time;
- A day is divided into two periods , first period and second period;

The time slots allocation is shown in table below:

Starting time	Ending time	Duration [min]	Event	Designation
7h00	7h50	50	First hour	First period
7h50	8h40	50	Second hour	
8h40	9h30	50	Third hour	
9h30	10h20	50	Fourth hour	
10h20	10h50	30	Break time	Break time
10h50	11h40	50	Fifth hour	Second period
11h40	12h30	50	Sixth hour	
12h30	13h20	50	Seventh hour	
13h20	14h10	50	Last hour	
14h10	7h00	1010	Off class	Off class

Table 1: ADB Nyarutarama Teaching time slots allocation

3.1.1 Design of the transmitting end part

The transmitting end part is composed by an HMI (Human Machine Interface), a PC with LabVIEW installed with LabVIEW interface for Arduino (LIFA) [3] it has also an Arduino [4] microcontroller a transmitting Zigbee[5].

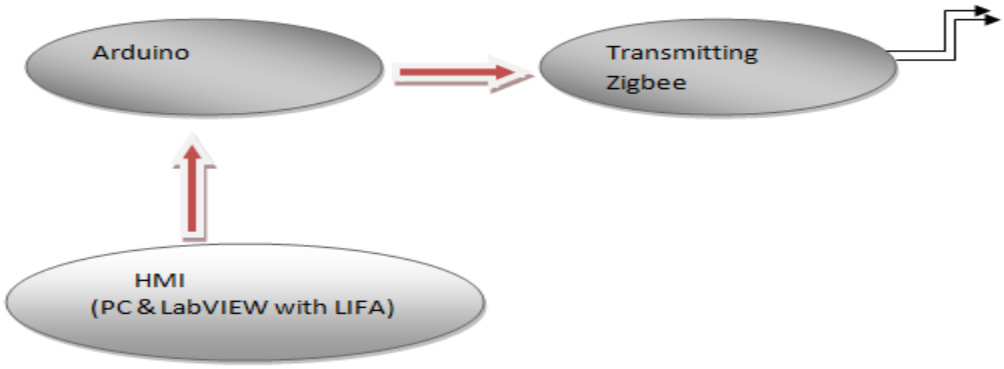

Figure 1: Transmitting end block diagram

3.1.1.1 HMI

This is the main part of the system, it is nothing but a PC with LabVIEW program installed and that LabVIEW program has LIFA as a plug -in so that the Arduino microcontroller can be accessed from LabVIEW.

3.1.1.1.1 System user interface

As we can see on the figure below, from the user interface, the user:

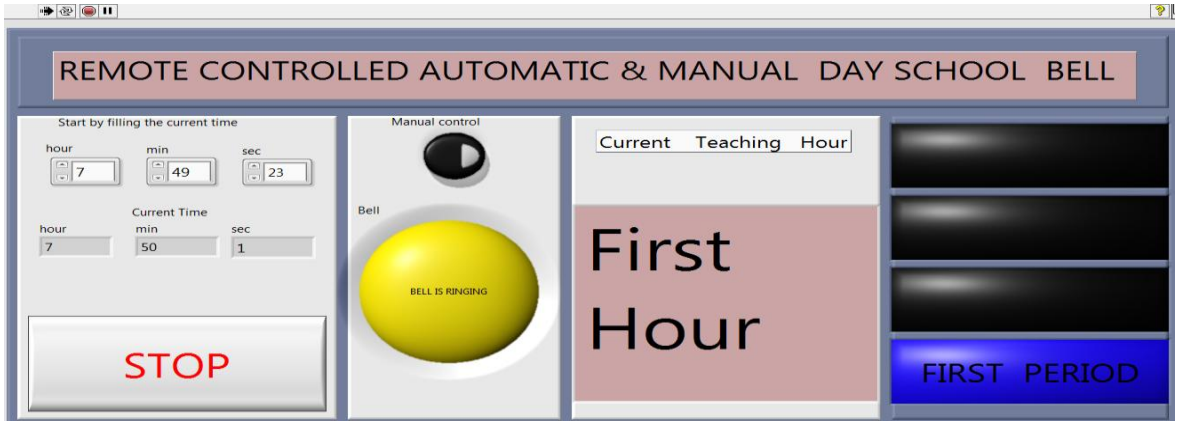

Figure 2: System user interface

- Starts by imputing the current time, taken from his/her watch
- Can know if the bell is ringing or not;
- Can start or stop this application;
- Can manually control the bell;
- Can know the current teaching period.

3.1.1.1.2 LabVIEW Block diagram

As LabVIEW is a graphical programming language, its coding is in form of block diagrams as it is shown below:

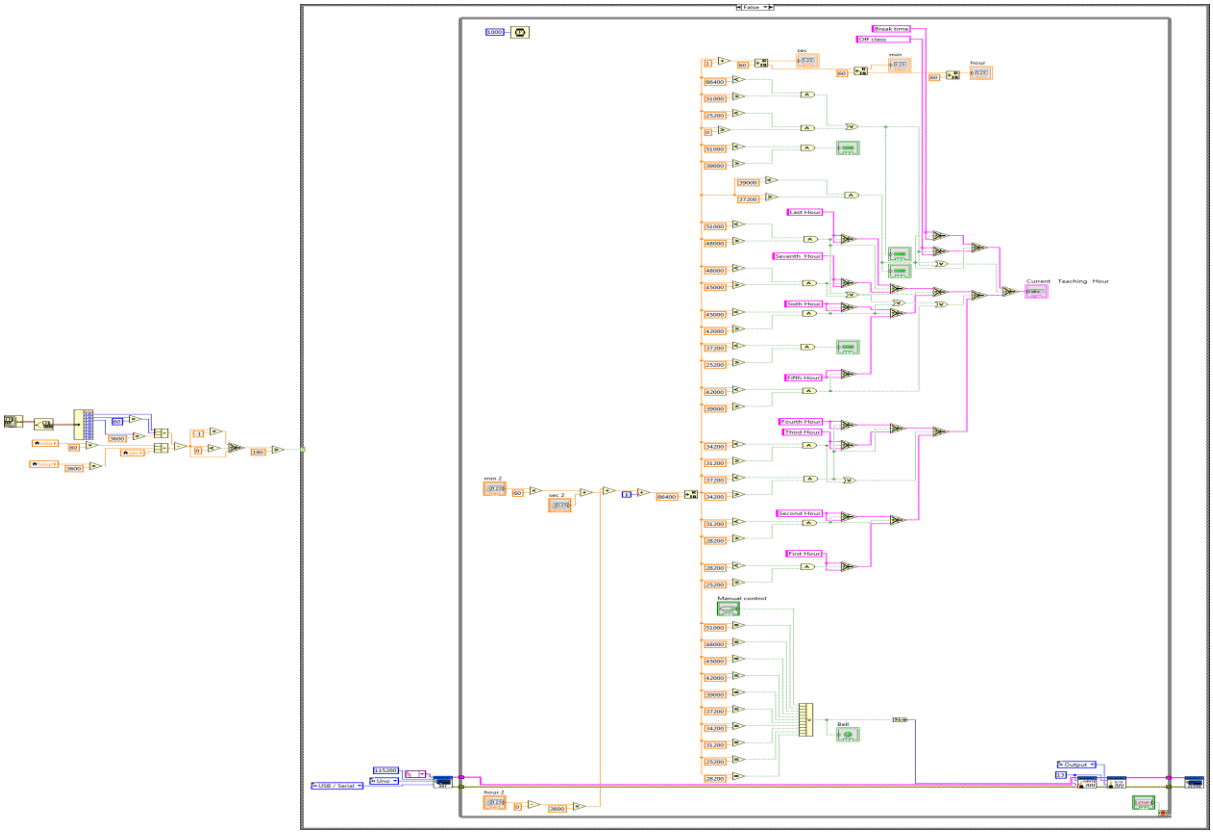

Figure 3: LabVIEW block diagram

In LabVIEW programming side, all time slots are defined and some formulae are used for different conversion and comparisions , all those blocks are placed inside a while loop as the system runs continously.

3.1.1.2 Arduino Uno Microcontroller

Arduino is a microcontroller board based on ATmega 328. It has 14 digital input/output pins (of which 6 of them can be used as PWM output) 6 analog inputs, a 16 MHz crystal oscillator, USB connection, a power jack, an ICSP header and reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC to DC adapter or battery to get started.

3.1.1.2.1 LIFA

LabVIEW Interface For Arduino is a kind of plug in which must be installed in LabVIEW so that it can recognize and work together with Arduino Microcontroller. For LabVIEW to work together with Arduino, a software called VI package[6] Manager is required which is the recommended method for downloading and managing the LabVIEW add-ons and give an instant access to the add-ons on the LabVIEW tool network, this VI package Manager works with LabVIEW 2009 and later . The following steps are used:

- From VI package Manager browse LabVIEW interface for Arduino;
- Install LabVIEW interface for Arduino;
- Connect your Arduino to your computer;
- Open Arduino IDE;
- User this path(for LabVIEW2011 installed on driver C) : C:\Program Files\National Instruments\LabVIEW 2011\vi.lib\LabVIEW Interface for Arduino\Firmware\LIFA_Base;
- Open LIFA_Base file and upload it to your Arduino ;
- Close the Arduino IDE.

3.1.1.3 Transmitting Zigbee

Zigbee is wireless communication protocol which is working on 802.15.4 IEEE standard. In this system, Trang [7] module has been used

3.1.1.3.1 Zigbee configuration

Zigbee module configuration or reading is achieved by using a terminal application such as HyperTerminal; in this system TMFT 2.6 software has been used.

3.1.1.3.2 Command mode

Command mode is used for reading or modifying the parameter of a module. Once the command mode is initiated, incoming character through serial port are considered as commands.

3.1.1.3.2.1 Entering AT Command Mode

The AT command mode is entered by taping "+++" symbols within a short time which can be estimated to one second, the following is the AT command format:

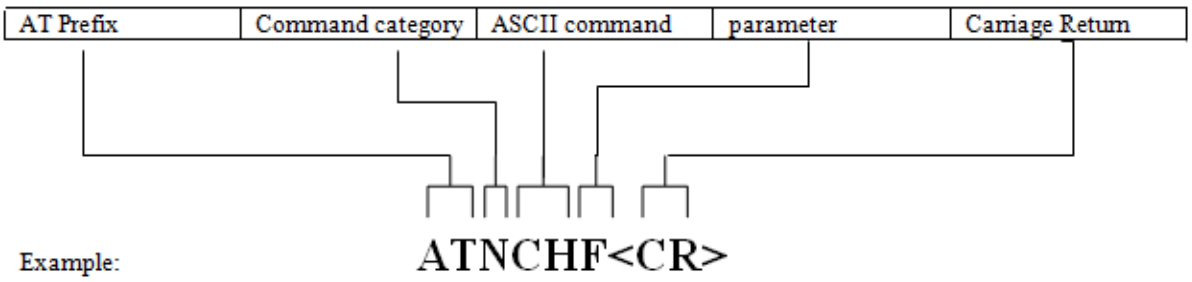

Figure 4: AT command format

The given example would change the Zigbee module channel to "0X0F" (15). To store that value to memory, we have to send write command else the value will be lost once the module is re-booted. When the command is sent to module, the module will execute the command and if the execution is successful the module return an OK message otherwise an error will be shown. To exit the AT command mode, send ATGEX command.

3.1.1.3.2.2 Zigbee module programming

Series 2 Zigbee modules can be used in Unicast network, Broadcast network or Peer to Peer (point to point) network. In this system , they have been used as in Unicast network as shown below:

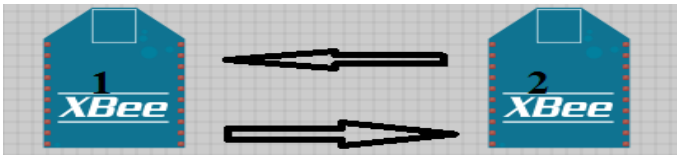

Figure 5: Zigbee Pee-to-peer communication

In any networking scheme, address is given to each and every network device. Let Zigbee 1 be source and Zigbee 2 be destination. Then addresses are given as shown:

Parameter	Zigbee 1	Zigbee 2
Source address	0x1000	0x2000
Destination address	0x2000	0x1000

Table 2: Zigbee addressing mode

Programming of each module follows these steps:[7]
1 :Open window HyperTerminal (TMFT 2.6) ;
2 :Connect the Zigbee module to the serial/USB port;
3 :Choose the appropriate port and serial parameters in terminal software(TMFT2.6);
4 :Configure the module using AT command
By using the above steps, transmitting and receiving Zigbee are configured as follow:

- These two Zigbee must have the same settings i.e. baud rate, channel, data bit, etc.
- One Zigbee should be configured with I/O pins as input and other with I/O pins as an output.
- For setting I/O pins as input and output the following steps are followed:
  - Enter the command mode with '+++'
  - Enable the desired I/O pin as input with command ATIDXX where XX indicate which pin to be enabled. In this system first I/O line ID0 is used. For configuring it to digital I/O input, the ATID02 is used, and wait for OK response.
  - Write the parameter to the memory with ATGWR command.
  - Exit command mode with ATGEX command.
  - Follow the same steps for configuring the receiving Zigbee
  - After receiving the OK response from module, write it to its memory and exit the command mode.

3.1.2 Receiving end side

The receiving end part is composed by a receiving Zigbee interfaced to an electrical bell, the interfacing circuit is nothing but a MOSFET and relay as is shown below:

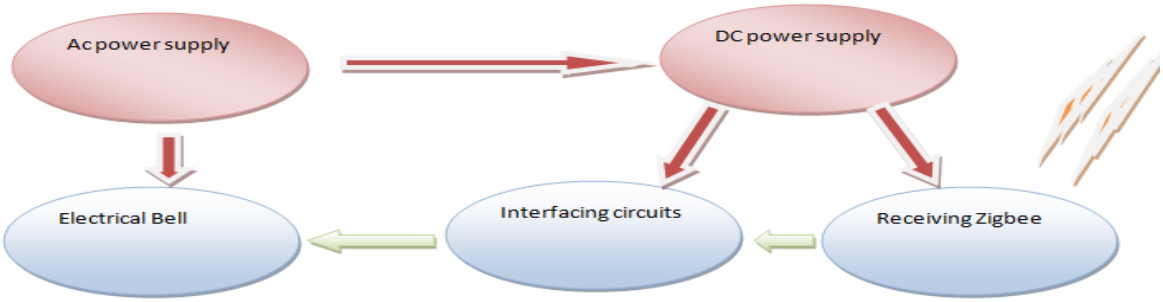

Figure 6: receiving end Block Diagram

As you see from the above circuit, the receiving end side has two different power supplies one Ac and other DC. The DC power supply supplies Zigbee through a zigbee shielder as Zigbee must be supplied by 3.3V

3.1.2.1 Schematic diagram for receiving end side

The receiving end ciruit is detailled below:

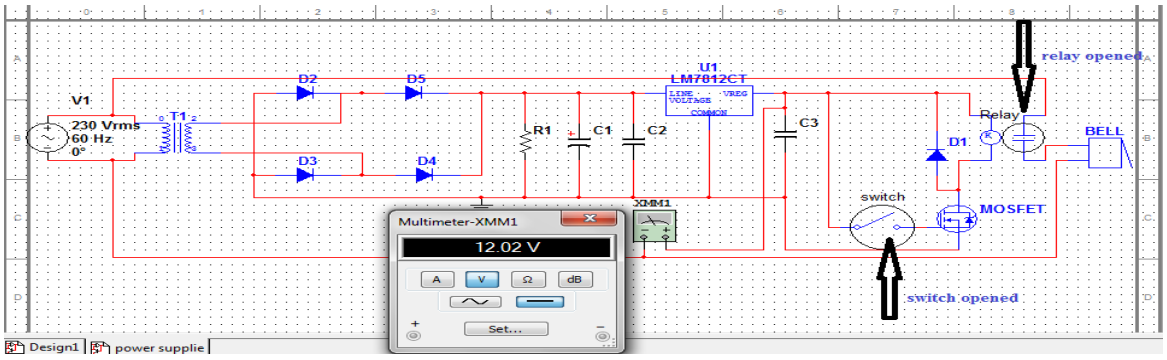

Figure 7: The receiving circuitry showing the state of the bell when there is no signal received by Zigbee

The above circuit is a 12V DC regulated power supply along with an electrical bell and its interfacing circuit. For simultaion purpose, a swich has been used as an output from receiving zigbee, as it is shown when the switch is open (when no signal availble at the output of zigbee) the relay is open and the bell is in OFF condition.

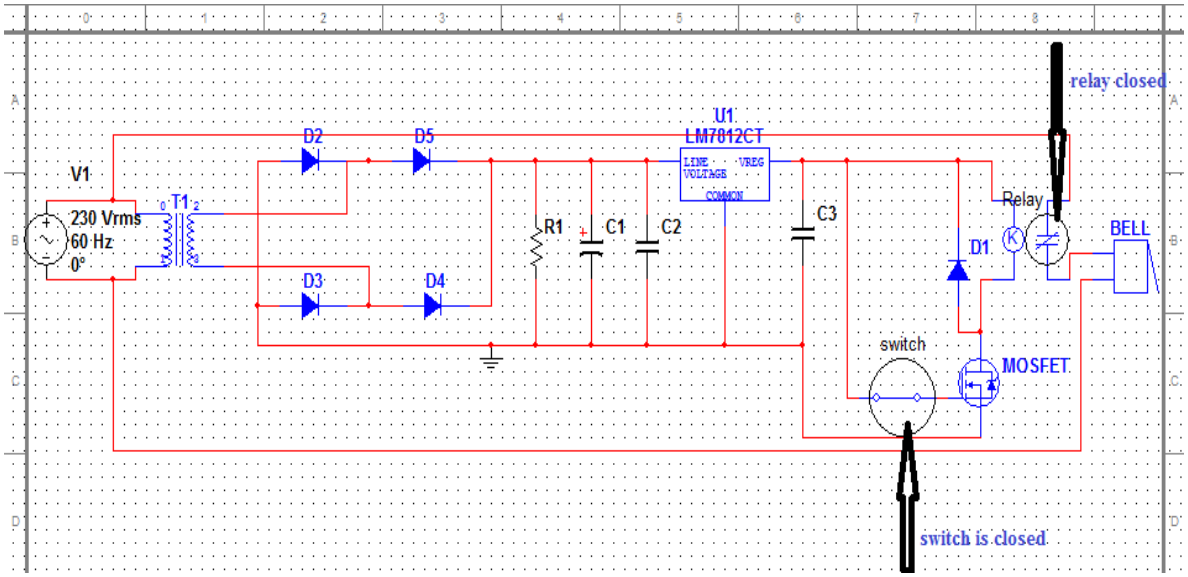

Figure 8: The receiving circuitry showing the state of the bell when a signal is received by Zigbee

From the above circuit, we can see that when the switch is closed (the output of zigbee Pin is high) the relay closes and the bell rings.(Fernandez-Lopez, Afonso et al. 2012)

3.2 System working principale

The user (headtecher) will start by entering the current time from his/her watch for unitialization purpose, LabVIEW program will compare the entered time and the computer's time and if there is a differnce of two minutes, it will ask the user the match both times (computer's time and watch's time). If both times are matched, LabVIEW program will start comparing the current time to predifined time slots. If the current time is equal to the end or the starting of any time slot, LabVIEW will generate a signal which will be transmitted to transmitting zigbee through arduino and arduino shielder. The same event will happen also if the user manually press the control button, for example in case of urgent meeting. The imformation from the transmitting end will transmitted to the receiving end through wireless communication. Reaching the receiving Zigbee, one of its I/O pin configured as output pin will be high and as is connected to the gate of MOSFET, the MOSFET will close the parth from VCC to ground through relay's coil. The coil will be energized and relay's normally open(NO) will close and the bell will ring.

IV .CONCLUSIN AND RECOMMENDATION

4.1 Conclusion

The primary goal was to design and implement a system which can be able to monitor the school timetable's time slots by controlling the school bell. the implimentation was successifull.The system will replace the current electrical bell, the same system is not only used in schools but also in some other componies where their activities depend on time slots allocation. This system will save time and money, there is no need of going to ring the bell. For distribution purpose, an installer or an executable file can be made from LabVIEW program so that it could even run on a computer without LabVIEW program installed.

4.2 Recommendation for futher work

I recommend that if the work is continued, to make this system a closed syestem by putting a second controller at the receiving end side so that we can know the information about the bell. I also recommend to make a full control system for the whole school which may have an application form for student registration, a system for remotely requesting marksheet, system for library monitoring, etc…

REFERENCES

[1] http://www.ni.com/labview/
[2] http://www.gsadb.com/index.html
[3] http://sine.ni.com/nips/cds/view/p/lang/en/nid/209835
[4] D'Ausilio, A. (2012). "Arduino: a low-cost multipurpose lab equipment." Behav Res Methods 44(2): 305-313.
[5] Fernandez-Lopez, H., et al. (2012). "ZigBee-based remote patient monitoring." Stud Health Technol Inform 177: 229-234.
[6] http://sine.ni.com/nips/cds/view/p/lang/en/nid/209002
[7] http://www.melangesystems.com/tarangF4.html