AC 2008-2014: CHESS HONING ELECTRONIC SWITCHING SYSTEM (C.H.E.S.S. BOARD): A CASE STUDY OF SUCCESSFUL DESIGN AND IMPLEMENTATION OF A SENIOR DESIGN PROJECT

Ahmed Khan, DeVry University

Jack Manansala, DeVry University, Addison, IL

John Smith, DeVry University, Addison, IL

Gabe Perez, DeVry University, Addison, IL

R Aguayo,, DeVry University, Addison, IL

Rommel Sison, DeVry University, Addison, IL

Chess Honing Electronic Switching System (C.H.E.S.S. Board): A Case Study of Successful Design and Implementation of a Senior Design Project

Abstract

DeVry University's Electronics Engineering Technology/Computer Engineering Technology (EET/CET) program senior project is a two-semester course sequence in which students synthesize knowledge and skills learned in the previous courses. In the first course (EET-400, Project management), students research, plan and develop a project proposal. And in the second course (EET-410L, Senior Project Laboratory) students implement the project plan by building and testing a prototype. A typical project involves a solution to a software/hardware-based engineering problem. The process of developing and implementing a solution to the problem offers a learning opportunity for students to gain new insights and competencies as a result of "*constructivist*" and "*deep learning*" teaching/learning approaches.

According to the *Thesaurus of ERIC Descriptors,* constructivism is a "viewpoint in learning theory which holds that individuals acquire knowledge by building it from innate capabilities interacting with the environment." The constructivist approach is based on recent research about the human brain and what is known about how learning occurs. It is an approach to teaching and learning based on the premise that cognition (learning) is the result of "mental construction." In other words, students learn by fitting new information together with what they already know. Weigel[4] has identified the attributes of *deep learning* as a methodology in which learners (i) relate ideas to previous knowledge and experience, (ii) look for patterns and underlying principles, (iii) check evidence and relate it to conclusions, (iv) examine logic and argument cautiously and critically, (v) are aware of the understanding that develops while learning, and (vi) become actively interested in the course content.

The paper presents a summary of the design and implementation of senior project titled "Chess Honing Electronic Switching System (C.H.E.S.S. Board)." In an effort to fill the world's void of a kinesthetic chess teaching tool, the C.H.E.S.S. Board combines the use of push buttons, LEDs, the 68HC12 microprocessor, and C++ programming to be used by novice and experienced players. The board features an assist mode that displays the legal moves of a piece. Error detection alerts the player that an illegal move was performed. Checkmate check signals that the king is in checkmate and a digital clock counts down for competitive play. Another feature of the C.H.E.S.S. Board is its ability to record the moves of a game. This allows the players to go back and review and learn from their mistakes during the game. Finally the C.H.E.S.S. Board has the capability to save and load unfinished games.

I. Overview

Problem Identification

Chess is a wonderful game of strategy, intellect, and skill. However, many do not have the proper background to play. If one can learn the fundamentals of the game while playing, it is then possible to play with anyone else familiar with the game. Additionally, whether one is a novice or an expert, mistakes do happen. A player can accidentally place a piece on a square illegally and not realize it, which can then affect the outcome of the entire game. For more experienced players, a clock is used to time the games. The clock is usually analog and separate from the board, thus it is difficult to set it to a specific time and therefore it lacks precision of a digital clock. Finally, during tournament play, the players are required to keep a record of the moves performed during a game. This prevents any possible disputes about illegal positions and also allows the game review for future analysis.

What is Available on the Market?

There are many different software programs available in the market that teach the game of chess. Among these are PC-based programs that help with learning moves and strategies. Additionally, there are websites that conduct tutorials on how to play chess. Though these platforms are helpful and informative, many beginning students of the game learn better by performing the real moves and learning from mistakes. There are countless books that also teach chess. These are helpful, but again the best way to learn is to play the game kinesthetically. Finally, there are many electronic versions of the game. They vary from handheld devices with touch screens to full chessboards with integrated LCD monitors. However, none of these electronic versions utilize the idea presented in this project of integrating LEDs underneath each square to show the possible moves.

Idea/Solution

The solution presented in this paper for the stated problems is to implement an electronic chessboard that incorporates the use of LEDs under each playing square. It is designed to be utilized primarily as a teaching tool that shows the possible moves for an individual piece and serve as a form of error detection and avoidance. The user would be allowed to participate actively in the game and learn at the same time.

The solution also has an optional clock for players who wish to play timed games. To eliminate the aforementioned lag time, the player's clock will automatically stop when a move is completed, and identified as a legal move. At this point, the opponent's clock will automatically start. Another feature of C.H.E.S.S. board is the ability to save and load a game from scratch. Additionally, the players' moves are recorded to allow users to review and analyze a previously played game. It is hoped that these features will expand the demographics of chess players viz a viz age and skill level.

Block Diagram of System Components

Figure 1 depicts the block diagram of the C.H.E.S.S. Board system. The brain of this

system is a 68HC12 microcontroller with expanded output ports from a series of 8255 integrated circuits. The 8255s are programmable peripheral interface devices that provide additional input/output capabilities. The outputs from the 8255s are used to light up the LEDs, located under each of the squares, to notify the user of legal moves on the C.H.E.S.S. Board. A keyboard circuit is used to send scan codes to the 68HC12 making it possible to connect the 64 pushbuttons. The pushbuttons are used as the squares on the board where the chess pieces will be placed. Also connected to the 68HC12 is an LCD that communicates with the users. The LCD displays messages to the user notifying them of which players turn it is, check checkmate, and invalid moves. In addition to messages, the LCD also contains the countdown clock for a timed game.

The PC contains the software that processes the information received from the pushbuttons and the keyboard circuit. The software also enables the PC to communicate with the 68HC12 for controlling the LEDs.

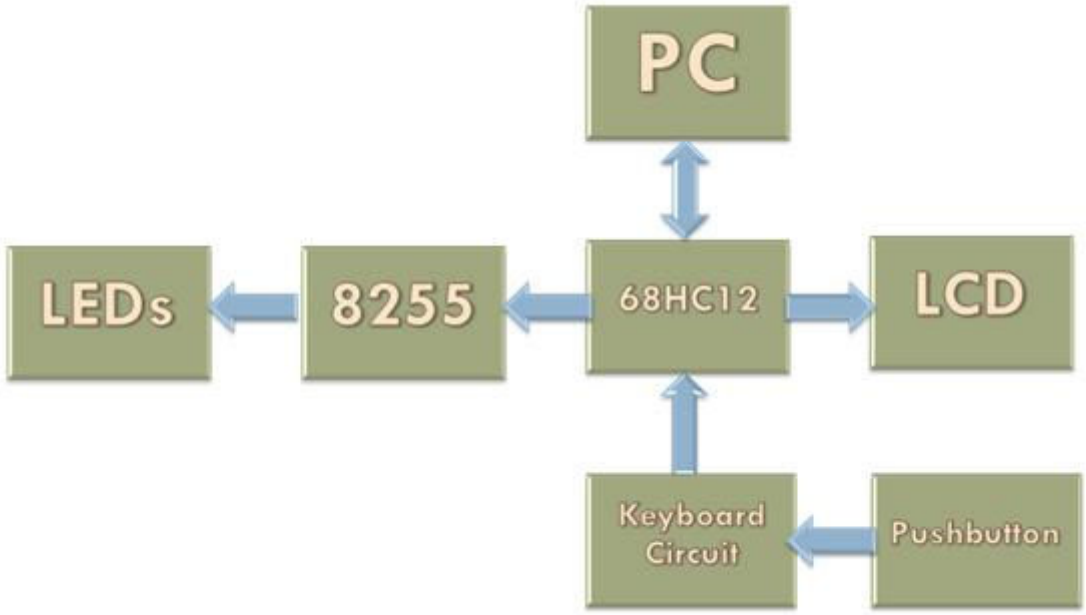

Figure 1: Block Diagram of C.H.E.S.S Board System

II. Project Objectives

The C.H.E.S.S. Board has an initial state as the default mode which allows every game to begin in the same manner, with each piece on its assigned square. A player can select a piece to move by pressing down on the square that piece is positioned on. When the pushbutton is pressed, a scan code is sent to the 68HC12 microprocessor which converts the scan code to position bit code for the software to receive on the PC.

The software then computes the appropriate algorithms and returns to the board and the respective LEDs light up. However, this has some constraints. A square cannot light up if a player's own piece is occupying it. Paths are blocked if a player's own piece or an opponent's piece is in the way. A player cannot expose its own king to check. Since each piece is unique; therefore the code must have several piece-specific classes.

Once the piece in play has been placed back on the board, the pushbutton sends a signal to the microprocessor notifying the change in its state. The program then performs an error check. The error check is to ensure that a legal move was performed. If the move is illegal, the LCD notifies the player by displaying a message. The player then returns the piece to its original position and proceeds with another move. If the move is legal, then there will be another test. The program ensures that the opponent's king is not in check or checkmated. If the king is put in check, the LCD displays a message notifying the player that the king is in check and that game will continue. In case of checkmate, the game is over and all individual squares start to blink simultaneously.

The system also has a timer, which can be set to a desired amount of time before a match. Again, in order to complete a move, a player must place the piece on the square and activate the pushbutton. This initiates the error check algorithm, starts the opponent's clock (as long as the move was legal), and continues the sequence of the game, until a player's clock reaches zero. When this occurs a message will be sent to the LCD informing the players that the game is over and that the player with time running has won the match.

The error check is always activated to reduce any possibility of human error. The assist function and clock function is optional to the players. The players can choose to the turn the functions on or off.

The players also have the option of saving the present status of the game and continuing it at a later time. Additionally, a player can easily modify a file to dictate where each piece should go, thereby creating a game entirely from scratch.

III. Development and Implementation of a Prototype

For the hardware Motorola 68HC12 microcontroller is used with expanded outputs from the 8255 integrated circuits. Figure 2 illustrates the 8255 integrated circuit. The microcontroller sends signals via outputs to control the LEDs. The microcontroller receives input signals from the pushbuttons when the turn is initiated and concluded. The countdown clock – for the more a more competitive user – is synchronized with the internal crystal oscillator of the 68HC12 microcontroller.

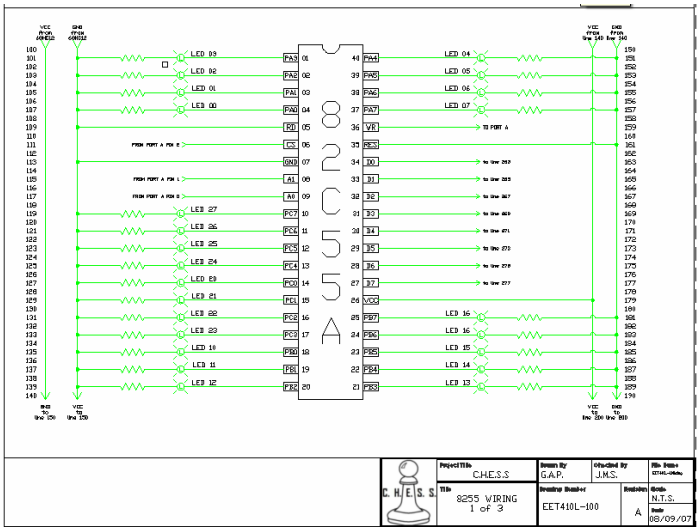

Figure 2: Input/Output Interface of 8255 Circuit

In regards to the software, the chosen programming language is C++ object-oriented programming. Each piece is an object of the program and contains a location on the board and the color of the piece. Functions are designed to light up the legal moves for the piece, check the presence of a piece on a square, perform a check check and checkmate check, and signal the change of turns.

IV. Contractual Aspect

Our project does not have any customers, thus there are no contractual issues.

V. Gantt Chart

Figure 3 illustrates the Gantt chart followed for the design and implementation of the project.

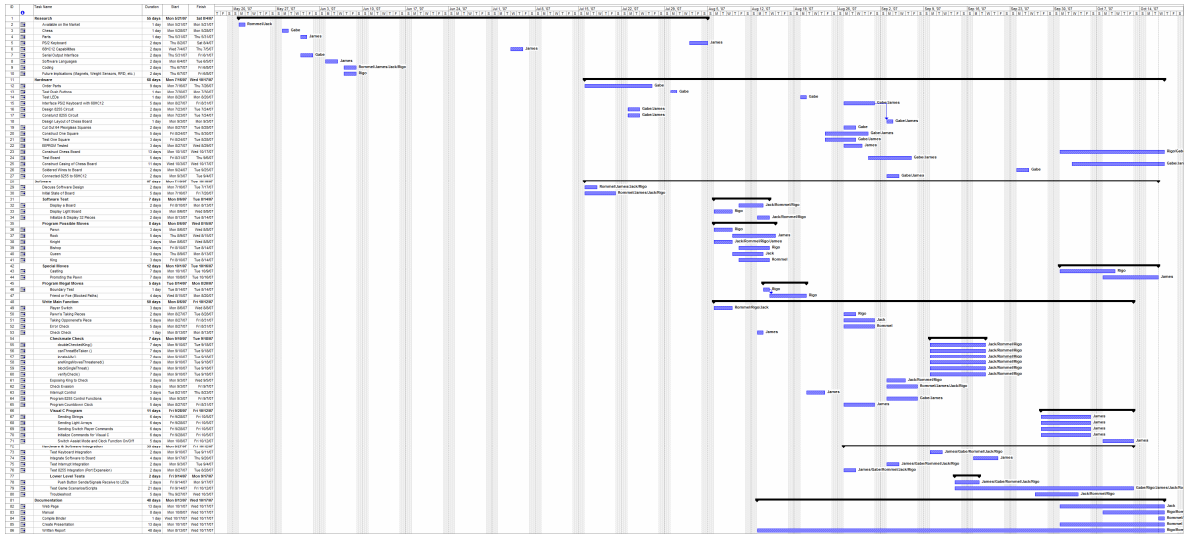

Figure 3: Tasks vs. Time Schedule of C.H.E.S.S. Board (Gantt Chart).

VI. Resources

Table 1 list the resources used and cost breakdown for C.H.E.S.S Board system.

Item Description	Manufacturer	Quantity	Unit Price	Total Price
Small LED - Red	Hobby Engineering	64	$0.20	$12.74
Pushbuttons	e-Bay	64	$0.63	$40.32
68HC12 Dragon Board	Wyntec	1	$98.00	$98.00
8255 Programmable Peripheral Interface	Toshiba	3	$5.00	$15.00
Carbon Film Resistor	DeVry	64	$0.05	$3.20
Wire 22ga Stranded Red 500’	Fry’s Electronics	5	$4.99	$24.95
Wire 22ga Stranded Black 500’	Fry’s Electronics	2	$4.99	$9.98
Plexiglass 24” x 24”	Menards	3	$5.00	$15.00
Plexiglass Cutting Tool	Menards	2	$3.00	$6.00
Shrink Wrap for Wires	RadioShack	1	$2.99	$2.99
Breadboard	Fry’s Electronics	1	$20.00	$20.00
Magnetic Strips	Staples	10	$3.49	$34.90
Illustration Board	Crescent	1	$4.99	$4.99
Plywood		1	0.00	$0.00
			Total	$288.07

Table 1. Resources used for C.H.E.S.S Board system

VII. Personnel

The following is description of team members and assigned tasks.

- Rigo Aguayo
  - Software Design, Software Testing, Documentation, and Final Board Testing

- Jack Manansala
    - Software Design, Assisting in Board Construction, Web Site Construction, and Software Testing
- Gabriel Perez – Project Manager 1
    - Board Construction, Hardware/Software Integration, Hardware Design, and Hardware Testing
- Rommel Sison – Project Manager 2
    - Software Design, Assisting in Hardware/Software Integration, Software Testing, Documentation, and Final Board Testing
- James Smith – MVP
    - Hardware Design, Interfacing, Hardware/Software Integration, Documentation, Visual C Programming, and Software Design

VIII. Technical Details

Hardware Details

The C.H.E.S.S. Board works much the same way a PS/2 keyboard works. This is because the circuit that sends the signals from the chess board to the PC was originally from a PS/2 keyboard. Each square on the C.H.E.S.S. Board is assigned its own scan code.

When a square on the board is pressed, a scan code is sent to the 68HC12. The scan code lets the 68HC12 microcontroller know what square was just pressed. For example, in the beginning of a game, if a player presses down on the pawn on square (1, 1), a scan code of '04' is sent down the serial line to the 68HC12 microcontroller. When the button is released the circuit will send another scan code with a value of 'F0' immediately followed by another '04' to signify what square was last pressed. A look-up table programmed in the 68HC12 microcontroller translates the scan code to a form that is compatible with the game program.

In order to have a sufficient number of outputs ports on 68HC12 microcontroller, it is necessary to use three, 8255 integrated circuits. The 8255 is a programmable peripheral interface which makes it is possible to program each of the 24 outputs that each chip provides. When the game's software on the PC identifies which squares are to be lit up, it sends the states for each LED to the 68HC12 microcontroller. Then the 68HC12 microcontroller sends out the states to the output addresses on the 8255's, thus turning on the appropriate LED's.

Software Details

The main objective for the software code in C++ is to properly light up the moves (squares) of the players chosen piece. The constraint for writing this code was to

recognize that the only information recognized by the C.H.E.S.S. Board is a particular piece's beginning and ending coordinates. With this in mind, an 8 by 8 array named cBoard was created. Each element in this array is meant to contain a piece.

The various pieces are created using an abstract class named Piece. The Piece class is modified by the classes King, Queen, Rook, Bishop, Knight and Pawn. Each of these six classes contain information regarding the piece's color, its type and its most basic moves. A move is determined by its natural algorithms along with the way it is intended to interact with the surrounding pieces.

IX. Evaluation Methods

The test plan for the software involved recreating each in game scenario. For example, after designing and coding the software to determine whether a king is in Check, it needs to be varied if it works. In order to test this, the C.H.E.S.S was initialized with the pieces set in a situation that will result in Check. Once the board is in Check, the outputs were verified to conform to conditions designated for Check. The output message stating 'the board is in check' is displayed. Once this message is received in game, then a Check function was successfully created and tested. Similar testing approaches were employed to test various functions of C.H.E.S.S. Board system.

Hardware Problems

Creating this C.H.E.S.S. Board turned out to be a daunting task. Many hardware problems were encountered before reaching the final system design and implementation. The first and possibly most alarming problem was a flaw in the original concept for the chess board. The original C.H.E.S.S Board was to use phototransistor sensors to identify a piece's position on the board. The problem occurred when after the player selects the piece, where it will be placed? There is no simple solution to that and therefore it was decided to use pushbuttons for each square of the C.H.E.S.S Board. This enabled the system to recognize and to send the x and y coordinate of a piece's starting and ending position.

Software Problems

A major flaw in our plan was that the GNU compiler did not work with C++ code. All of the logic in determining what squares to light up is written in C++. An attempted was made to transform our C++ software into C code for the GNU complier to read. The disadvantage of this approach was that the C programming did not support the creation of classes which was vital to the program. In order to solve this problem PC was linked to the 68HC12 microcontroller board, and let the PC compile and execute the file. In addition a separate code written for the 68HC12 microcontroller handled all the inputs and outputs.

X. Synthesis of Learning

Despite these hardware and software problems which were encountered the project C.H.E.S.S. Board was successfully implemented and demonstrated. The EET/CET program provided a solid educational experience that allowed the development of the C.H.E.S.S. Board system. Each course in the sequence has proven useful in the development process of the project. Since the team consisted of three Electrical Engineering Technology (EET) students and two Computer Engineering Technology (CET) students, the team members had the opportunity to learn outside of their field. Team members thus learned how to integrate electrical engineering theories with the computer engineering concepts. The project also enabled the team to incorporate concepts learned in the introductory classes of electronics and programming. The project has widened the scope of learning and knowledge in the electronics and computer fields for the team members. The project also helped the group to develop higher levels of knowledge by learning totally new items that were not covered during the course work.

XI. Conclusion

The paper presented a discussion on the design and implementation of senior project titled "Chess Honing Electronic Switching System (C.H.E.S.S. Board)." The system fills void of the need for a kinesthetic chess teaching tool. The C.H.E.S.S. Board is a user-friendly chess learning system which allows users to review and analyze the game to enhance their skill level.

Bibliography

"82C55A." Intersil. 16 Nov. 2006. Aug. 2007 <http://www.intersil.com/data/fn/fn2969.pdf

The 8255 data sheet was used to find pin layouts as well as the voltage and current requirements to expand the I/O ports of the 68HC12.

"Chess.Com." Chess.Com. 2007. July 2007 <http://www.chess.com>.

Used to research chess and how to play the game.

"EVBPlus." EVBPlus. Oct. 2007. July 2007 <http://www.evbplus.com>.

This site was used to learn the capabilities of the 68HC12 Dragon board. Documentation for the board was found at this site as well.

Ewer, Gary. "Reading and Writing to Serial Port in C++." FoggyLog. 2007. Sept. 2007

Used to research how to input/output through the serial port in C++.

"Keyboard Scan Codes." BarcodeMan. Jan. 2003. Aug. 2007
<http://www.barcodeman.com/altek/mule/scandoc.php>.

Used to find out the scan codes assigned to each key on a keyboard. It was also used to learn the functionality of the PS/2 keyboard circuit.