Scotty in the Engine Room - A Game to Help Learn Digital System Design and More

Peter Jamieson
Dept. of Electrical and Computer Engineering & Miami University
Oxford, OH

Abstract

Engineering education uses games and competition including events such as robotic competitions, egg drops, pumpkin launches, and paper airplane design to help educate students in design projects. In electrical and computer engineering, robotic competitions tend to dominate the landscape in the form of robot soccer and autonomous vehicle navigation. Though these activities are great for pushing an engineer to greater levels of skill and knowledge they require, for the most part, at least two or three years of education (with the exception of Lego Mindstorm robotics). In this work, I have created a game framework in digital hardware that will allow second year students to experience competitive engineering hardware design and real-time problem solving using a StarTrek model where the engineer solves problems in and for battle. This game includes digital hardware design on a development board, which controls a team's space ship to battle an opposing ship. The goals of this framework are to improve the success rate of student projects in the second year and to provide additional learning benefits.

Introduction

The engineering chief on the Enterprise, Scotty, could always come up with a quote such as, "I'm giving her all she's got, Captain!" that would capture what most engineers feel during design and problem solving while building and testing large systems. This is especially true during the four years of undergraduate education taking the first steps to becoming an engineer. Not many people realize how hard it is to design even simple systems and asking a young adult to build a system that doesn't inspire them is a mild form of torture.

For this reason, engineering education tends to use competition as an element of play that inspires and pushes students to spend significant time designing and working on projects. In electrical and computer engineering, robotic competitions are most popular type of framework where students and university teams compete against each other. For

Comments may be sent to the author at jamiespa@muohio.edu

example, the robot world cup (Robocup, 2010) is world wide competition where more than 3000 competitors meet in a number of divisions. At Miami University, we compete in autonomous vehicle challenges to mow a lawn autonomously (ION, 2010) with more than 20 teams from across the USA.

Though these competitions are great, they tend to be senior capstones or graduate projects that require a student to have at least three years of training. The more junior students can still compete, but do not get the full benefit since much of the complex system is beyond their understanding. Instead, students design simpler systems in programming courses and hardware design that either are toy products such as processors and operating systems, or build games and simple applications that don't necessarily capture their imagination.

In a digital system design course, the goal is to have students understand how transistors can be organized as logic gates, how logic gates can be organized to do calculation and control, and how calculation and control can be used to solve problems. Digital system design courses have a lab component where these concepts are illustrated for two-thirds of the course, and the last third of the course involves a more complex design. These complex designs include a simple computer processor, a vending machine, a traffic light, or an open project of the student's choice. At present, development boards using technology from companies such as Xilinx and Altera and their accompanying university programs (AlteraU, 2010; XilinxU, 2010) allow students to build more and more complex designs. For example, students at Miami University in 2009 built processors, encryption algorithms, hardware based maze games, and music tuners. Unfortunately, these successful projects were offset by approximately 50% project failures where there was no working project. These failures are, likely, due to lack of motivation, too much freedom for some students, and many students lacking experience with large projects.

For these reasons, I have created a game framework that will allow second year students to build their own ship interfaces in digital hardware and experience competition within a constrained system that they will engage with during the last third of their digital system design course. In this paper, I will describe the game framework in some detail and then will discuss which parts of the game focus on allowing students to experience valuable learning opportunities beyond a basic course in digital system design.

Redhawk Duels

The game we have created, "Redhawk Duels" is similar in game mechanics to the first vector graphics game, "Space Wars" (Space Wars, 2010). The basic goal of the game is to win the game by reducing the opposing team's ship to zero power. To do this, your ship can fly around a 2D rectangular map and shoot the opponent using some form of laser cannon. The game rules are similar between the two, but this is where the similarities end.

Redhawk Duels starts with two ships positioned at opposite parts of the map. A timer starts counting to a minute, and during this warm up minute both teams are responsible for configuring their respective power array (more later). Once the minute is up, both ships receive a minute power budget (based on their configuration) that can be used to accelerate the ship, fire the cannon, and use their laser finding sensor. If a team over uses their power budget, then their ship stalls for the remainder of this minute and the following minute. Also, colliding with an opponent or the map border results in both a stall and a teleportation back to the original starting position.

When one ship hits the opponent's ship with the cannon, the damage to the opposing ship is manifested as a degradation to the power array (more on this later in the paper). If at the start of two consecutive minutes a ship has no power then that ship loses the round. If both ships don't have power in two consecutive minutes then the round is declared a draw.

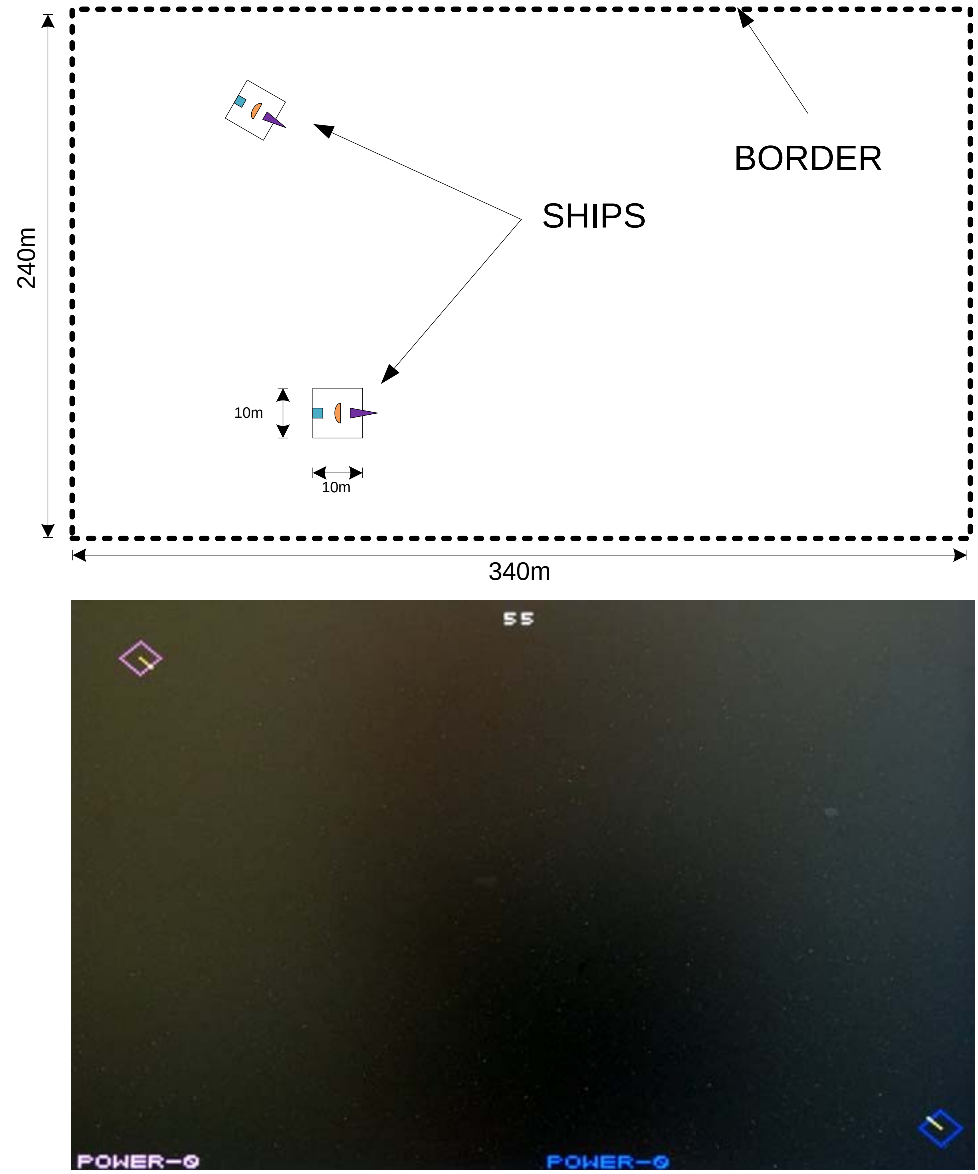

Figure 1. Shows the game start with two ships in a conceptual drawing and an actual screen shot.

Figure 1 shows a conceptual view of the field of play and a screen view of the game from the spectators point of view. Each ship, from the team competitors perspective, only has a sensor laser finder that can be rotated 360 degrees to find what is around them, and therefore, players have no vision of the field of play, much like a submarine, unless they build those capabilities into their hardware design.

From the perspective of the students, the challenge is to take a rudimentary game prototype ship designed in hardware that is implemented on a platform and make that design better for controlling their ship. The students can come up with any solution to improve their ship's design, including better interfaces to control the ship and new computation algorithms to automate aspects of the ship in battle. An additional aspect of the game is real-time problem solving. In particular, the power system for the game is a version of the traveling salesman problem (Papadimitriou & Steiglitz, 1976), and as the game progresses and your ship is shot then new solutions to this problem need to be managed on the fly.

In the following two sections, I will describe many of the educational aspects of both design and real-time problem solving. But before discussing these aspects, it is important that we have a better understanding of the hardware platform and the framework of the game.

The Hardware Platform

Figure 2 shows the hardware platform and the interconnection between them that actually coordinates the game. In particular, we use Altera's DE2 development board for both the games master (GM) and each player's ship (`http://www.altera.com/education/univ/materials/boards/unv-de2-board.html`). Figure 2 only shows a configuration for one player, the GM, and the communication link between the two boards via Ethernet link. In the full game, a hub is used to link all three DE2 boards together. The GM provides a view of the action (as seen in Figure 1) through the VGA cable. Again, this VGA view is only available to the spectators.

These hardware platforms, DE2 boards, are equipped with a number of peripheral interfaces including PS2 for mouse or keyboard, a USB port, an Ethernet port, a VGA port, sound port, LCD display, expansion headers, and switches and LEDs. The DE2 and similar boards are commonly used in digital system design and computer architecture courses (among others) since students can rapidly prototype complex digital hardware systems (Zhu, Weng, & Cheng, 2009; Calazans & Moraes, 2001). During the lab portion of these course, the students learn how to build hardware to interface with the board's peripherals. The digital system design aspect of these DE2 boards is manifested on a programmable hardware microchip called a Field-Programmable Gate Array (FPGA). The FPGA allows students to programmably create digital hardware that can then be used to talk to the peripherals and implement algorithms.

In the game framework, all of the DE2 peripherals, with the exception of the Ethernet port, are usable by the students to create their improved ship. For example, the basic ship prototype that is provided to the student uses 8 switches to control the ship, and a team could instead decide to interface with a traditional keyboard, which would allow the team to enter more complex macros at the touch of a button instead of a clunky and poorly labeled sliding switch.

Figure 2 also shows what is called a breadboard in the lower right hand corner. A breadboard is used to prototype and wire circuits, and in this case, we show the player's DE2 board expansion header connecting to some wires. This is how the power array is

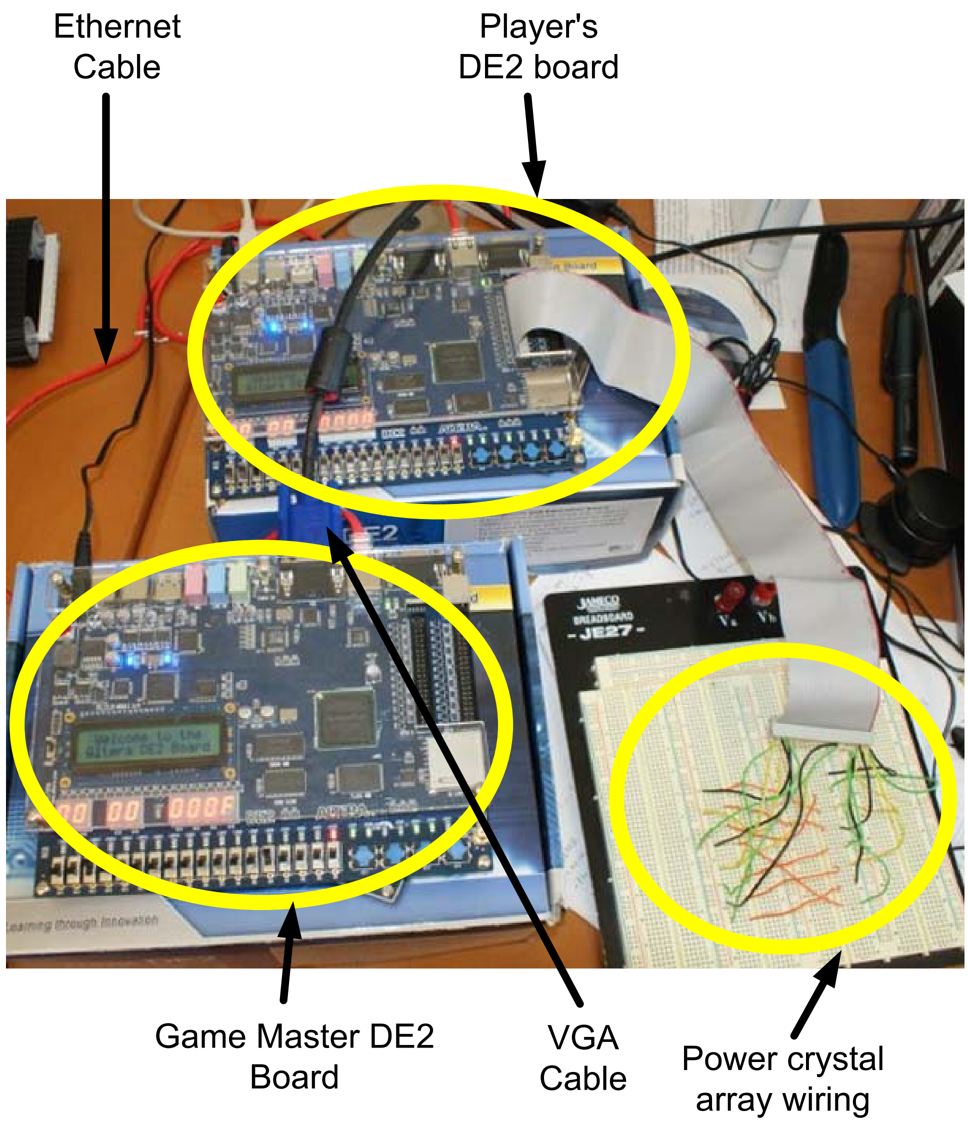

Figure 2. Shows the hardware setup for the game with one games master board and one player board.

implemented, and we will discuss the details of this later in the paper.

Educational Benefits of Designing for the Game

The benefit of educational based games and competition is that they can be an effective motivator and educator for many students within science and engineering (Miglino, Lund, & Cardaci, 1999; Murphy, 2000; Almeida et al., 2000; Gregson & Little, 1999). What tends to happen is the game and competition remove a student's inhibition to explore and dig deeper, and instead, the student is willing to explore far beyond what they normally would. This skill in itself is key for engineers to have since in most cases an engineer will only be involved in one piece of a larger system design, but will have to delve deeper into other aspects of the system when things don't work (which is normal).

This is only one example of the educational possibilities that a game, such as Redhawk Duel, hopes to expose the students to. During the design process, the following skills will be exercised and improved:

- An exposure to how to interface with digital components.
- Exposure to algorithms executed in parallel.
- A practical use of some of the physics and algebra learned in first year.
- The ability to understand an existing system design written and created by someone else.
- The appreciation of how design impacts the user.

The first two aspects in this list are accomplished in both traditional projects and within the game framework. The rubric for marking either digital design projects or a team's ship is the same, and the minimum project requirements would be that the students demonstrate the ability to interface with a peripheral and also demonstrate the ability to create a somewhat complex finite state machine (with more than ten states) that either executes a control algorithm or some other computation. The nature of hardware design is parallel, and this is usually a student's first exposure to parallel computation.

In addition to exposure to these two areas, the game framework has three additional benefits. First, a game such as this requires students to exercise algebra and physics concepts to understand how to control their respective ship. It is possible to ignore these details, but it is unlikely that such a choice will result in a competitive design. Therefore, the design of their ship will force students to begin to understand how science and mathematics provide powerful tools for solving design problems. This is, really, the nature of engineering (or applied science as it is also known).

The second skill that the game framework will expose students to is something that is partially missing in engineering education. The concept is called, code reading, and is related to software programs. It is the skill of taking existing code/design and reading and deciphering the design so that you can understand the system (or part of the system) to make your own modifications. The skill is not just simply reading (as the name may imply), but reading with understanding. For example, to understand the context of a Shakespeare play you need to know some of the social and historical concept as well as just the words, and code reading is similar with the addition to being a non-linear story, which adds even more complexity.

Within Redhawk Duel, there are two aspects that code reading skills help a student with. First, a student learns to understand a system that is written by someone else (the code reading skill), and second, a student is exposed to a design made by a more experienced engineer. This second aspect can provide a student with a good framework to start their own system designs, and this is important since these projects are some of the earliest larger designs that the students will experience.

Finally, the experience of improving the prototype ship is a design experience where they are both the designer and the user. User design is an important skill to have when producing a product, and early in an engineering career, a situation where the developing engineer is both the user and the designer can provide valuable experience and an appreciation for how hard this problem is. Though this experience is by no means the same as designing for an external user, it is an early introduction that includes some measure based on how well does the interface perform in real-time competition.

The design aspect, where a student team improves on an existing ship prototype, provides both the basic skills that a student is expected to develop in digital system design with some additional benefits. It is our hope that the constrained environment of a simple game will help many students by both motivating them to build a system and provides sufficient constraints and design examples so that the majority of the class can be successful.

Real-time Problem Solving and Teamwork

The design of each team's ship has many additional benefits over a traditional project based approach to teaching the course. The other key educational benefit of Redhawk Duel is real-time problem solving that includes teamwork. Engineers in many cases are faced with solving problems in real-time, and Redhawk Duel incorporates this concept in the form of the currency that powers the ship - the power array.

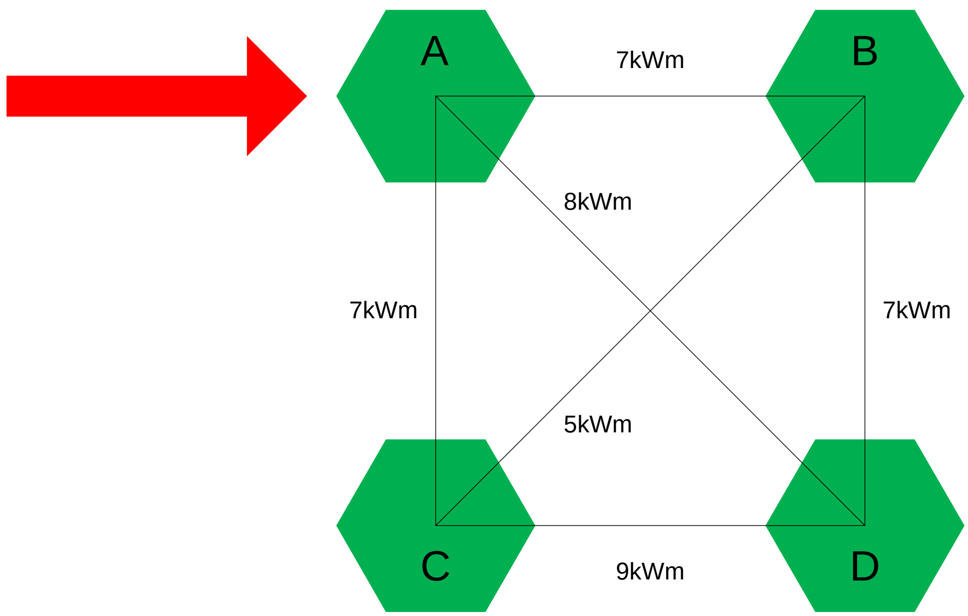

Figure 3. Shows the conceptual crystal array for 4 crystals.

What we've designed into the game is an optimization problem called the traveling salesman problem (Papadimitriou & Steiglitz, 1976). This problem is related by the power system of the ship, which supplies both power for actions and basic survival (win/lose) in the game. Figure 3 shows a simplified example of a 4 crystal array that a ship might contain. The goal is to hook these crystals up in a cycle that passes through each crystal only once such that the amplification of the original signal is maximized. The edges that connect crystals each have random weightings between 0 and 10 representing the amplification for a connection between to crystals, and therefore, the 8 crystal array (the actual number of crystals in the current version) can amplify the power signal by a maximum of 80. During game play the students can reroute their crystal array to improve their per minute power budget.

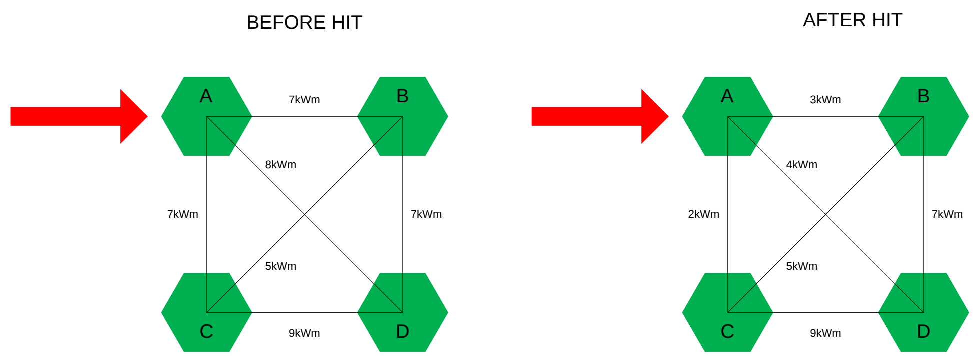

Figure 4. Shows how the crystal amplification degrades after being hit by a shot.

If the cycle routing includes a link (or edge) with 0 amplification, then this results in zero power amplification. Therefore, teams have one minute to create the best power generating routing. As stated before, if an opposing ship hits a ship with their cannon, then the amount of power the opposing team allocated to their shot will be randomly deducted (in both magnitude and which edges) from the edges of the power amplification. Figure 4 shows a simplified example of how the 4 crystal array degrades for a cannon shot with power 18. This new edge weight information is passed to the ship, and a team is responsible for rerouting the power network as needed.

The version of the power crystal problem in the current version consists of 8 crystals and only has 5040 different possible solutions. It is possible that the students will build an algorithm to solve this problem optimally, and if this is the case in future versions we will increase the number of crystals. For example, 12 crystals has over 39 million possible solutions and 14 crystals has over 6 billion (which is not optimally solvable in the one minute).

To perform the physical routing of the crystal we have provided functionality in the student provided framework as illustrated in Figure 2 in the lower right corner. Figure 5 shows a closeup of this wiring array using the expansion port from the DE2 board and some wires.

Conclusion

The creation of Redhawk Duel has the goal of improving student success in digital system design course for the project portion of the course. Not only does the game constrain requirements, provide a sample hardware design, and satisfy the basic requirements of a project within the course, but there are additional benefits associated with competition and the skills that a student will learn.

The students in the coming fall will be the first users of the game framework, and three lab sessions will be encouraged to help each other with the goal of generating 2 or 3 top teams from each section that will compete against the other lab sessions. After this tournament is complete, we will evaluate the situation and consider making this a broader competition for other colleges and universities.

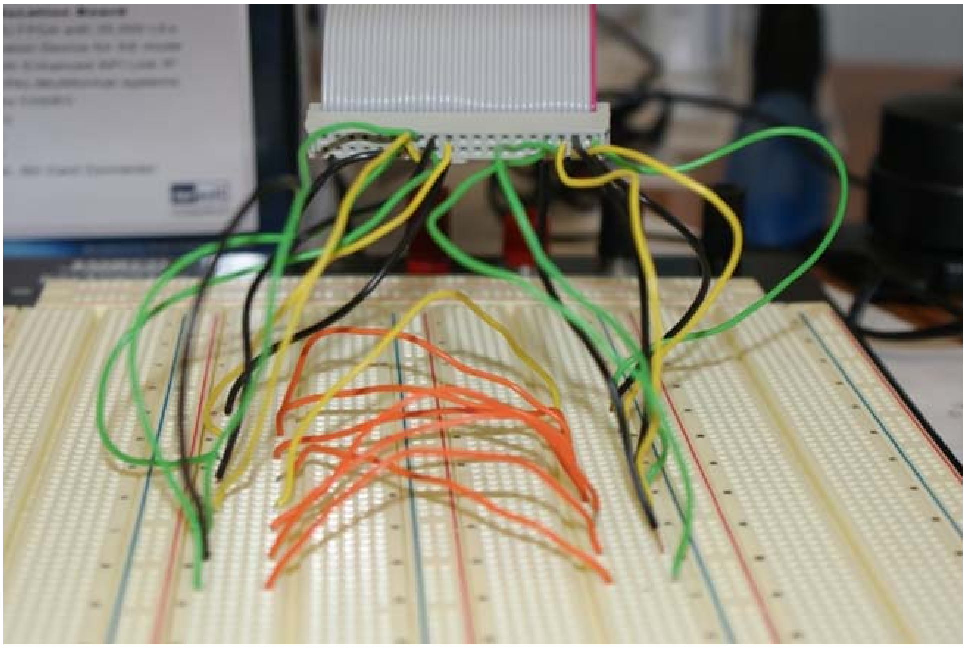

Figure 5. Shows a closeup of the physical wiring system for the crystals.

References

Almeida, L. B., Azevedo, J., Cardeira, C., Costa, P., Fonseca, P., Lima, P., et al. (2000). Mobile robot competitions : fostering advances in research, development and education in robotics.

AlteraU. (2010). *Altera University Program at http://www.altera.com/education/univ/unv-index.html.*

Calazans, N., & Moraes, F. (2001, may). Integrating the teaching of computer organization and architecture with digital hardware design early in undergraduate courses. *Education, IEEE Transactions on*, *44*(2), 109 -119.

Gregson, P., & Little, T. (1999, aug). Using contests to teach design to ee juniors. *Education, IEEE Transactions on*, *42*(3), 229 -232.

ION. (2010). *The 7th Annual ION Robotic Lawn Mower Competition at http://www.ion.org/satdiv/alc/.*

Miglino, O., Lund, H. H., & Cardaci, M. (1999). Robotics as an educational tool. *J. Interact. Learn. Res.*, *10*(1), 25–47.

Murphy, R. R. (2000). Using robot competitions to promote intellectual development. *AI Magazine*, *21*(1), 77-90.

Papadimitriou, C. H., & Steiglitz, K. (1976). Some complexity results for the traveling salesman problem. In *Stoc '76: Proceedings of the eighth annual acm symposium on theory of computing* (pp. 1–9). New York, NY, USA: ACM.

Robocup. (2010). *Robocup at http://www.robocup.org/about-robocup/.*

Space Wars. (2010). *Space Wars at http://en.wikipedia.org/wiki/Space_Wars.*

XilinxU. (2010). *Xilinx University Program at http://www.xilinx.com/univ/.*

Zhu, Y., Weng, T., & Cheng, C.-K. (2009, feb.). Enhancing learning effectiveness in digital design courses through the use of programmable logic boards. *Education, IEEE Transactions on*, *52*(1), 151 -156.