

Intelligent Cornhole Board (ICB)

Charles Belekis, *Senior, ECE The Citadel*, Kieftha Boyd, *Senior, ECE The Citadel*, James A. Bredesen, *Senior, ECE The Citadel*

Abstract - The Intelligent Cornhole Board was engineered to be an electronically scored game that incorporates Image Processing, Radio Frequency Identification (RFID), Bluetooth Communication, and programming to appropriately identify the color and placement of bean bags on a playing field. Then, the ICB calculates, transmits, and displays the score on a smart phone.

The programming was done using LabVIEW and consists of three main components: 1) Camera Image, 2) RFID and 3) Bluetooth Communication. The Image Processing captures images on the board's surface for four bean bags. The RFID identifies each bag as it enters the hole, and the Bluetooth communicates and displays the correct score on the mobile device. It is intended for the entertainment industry.

The project concept has proven successful; however, it could benefit from redesign to make it competitive in the commercial market. The next stage of development would be redesign for a microchip processor, and reprogramming for eight bean bags.

Index Terms — Android Application, Android Device, Bluetooth Communication, Computer, Game Board, Image Processing, and Radio Frequency Identification Processing.

I. NOMENCLATURE

Bluetooth – Is an open wireless technology standard for exchanging data over short distances using radio transmissions.

Cornhole – A bean bag game where the bags are commonly filled with 1 pound of corn.

LabVIEW – Laboratory Virtual Instrumentation Engineering Workbench is a visual programming language from National Instruments.

UID – Unique Identification.

UUID – Universally Unique Identification.

VI – Virtual Instrument.

VISA – Virtual Instrument Software Architecture.

II. INTRODUCTION

The ICB was a senior design project that fulfilled a 2 semester capstone requirement. The project utilized practical

application of multiple and varied academic subjects taught in an Electrical Engineering degree, as well as demonstrated the design process from conception to production (in this case a prototype). The design team wanted to have fun with the project but also delve into different aspects of electrical engineering topics that interested different members of the group, such as communications, systems integration, and electromagnetic energy. Additionally, the team wanted something that had the potential to be competitive in a commercial market. After some research it was determined that there is a hole in the market for electronically scored, outdoor games. With further discussion, the team chose the ICB as the design project.

The game is commonly called cornhole, and is played by two to four people, and involves throwing one pound bean bags at a game board approximately 27 feet away. Scoring occurs as a result of landing bags on the game board surface or through a six inch diameter hole on the game board. The player or team scores one point for each bag landed on the board and three points for each bag going through the hole. The game is won by reaching a score of 21 points before the other player or team. Additionally, points are canceled out by another team or opposing player scoring. For instance, if the red team threw a bag that landed on the board they would score a point. However, if during their turn the blue team threw two bags that landed on the board this would cancel out the red team's score. Instead of having a 1-2, it would be 0-1.

Early in the design and development process the team was using two microprocessors. This was done because of serial communications requirements for the peripheral devices, like the camera, the RFID reader, and the original microprocessor (Lego NXT). The RFID reader could only communicate using RS-232 protocol, and the Lego NXT could only communicate in I²C protocol. Since the PIC18 microprocessor could communicate with both, it was used to close the gap between the RFID reader and the Lego NXT. However, this proved to be inefficient. The course correction was to use LabVIEW, which supported all three main components of the design.

Fig. 1. is a picture of the ICB.

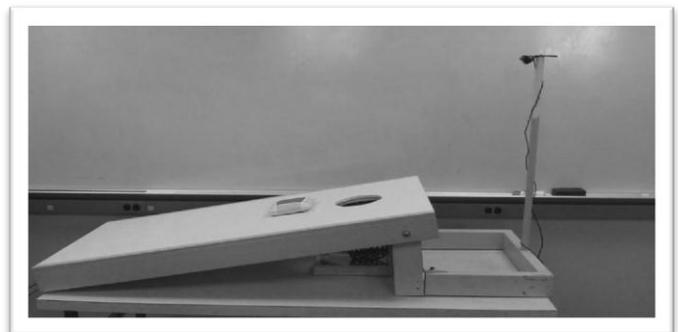


Fig. 1. Intelligent Cornhole Board

This work was supported by the Department of Electrical and Computer Engineering at the Citadel.

Affiliation footnotes:

C. Belekis is a student at the Department of Electrical and Computer Engineering, The Citadel, Charleston, SC 29409 USA (email: charles.belekis@gmail.com).

K. Boyd is a student at the Department of Electrical and Computer Engineering, The Citadel, Charleston, SC 29409 USA (email: kieth45@hotmail.com).

J. Bredesen is a student at the Department of Electrical and Computer Engineering, The Citadel, Charleston, SC 29409 USA (email: jamesbredesen@att.net).

III. TECHNICAL OVERVIEW

ICB is designed so the players would play a regular game of cornhole as they always have, but the board would keep the score. This is achieved by dividing the game's hardware into three parts: the board's surface, the center hole, and the bean bags. Each component plays a vital role in tracking the score of the game. The surface of the board is continuously monitored by a Microsoft LifeCAM HD 3000 webcam while the center hole is monitored by an APSX RFID reader. The bean bags are either red or blue and correspond to the red and blue team respectively. The webcam uses this information to assign the corresponding team with the right surface score of one point. Each bag is also equipped with a Texas Instrument RFID tag that the reader uses to identify which team's bag has entered the hole. Both the RFID reader and the webcam communicate to a personal computer via USB where the data is processed using LabVIEW software. The game begins by pressing the start button on LabVIEW's front panel and then the players will play their round normally. A real-time round score is displayed for the players as the four bags are thrown. When the round is over, the players will hit an "end of turn" button and the total score will be calculated and displayed on both the computer monitor and the player's mobile device. Fig. 2. displays the Top Down Flowchart of the ICB.

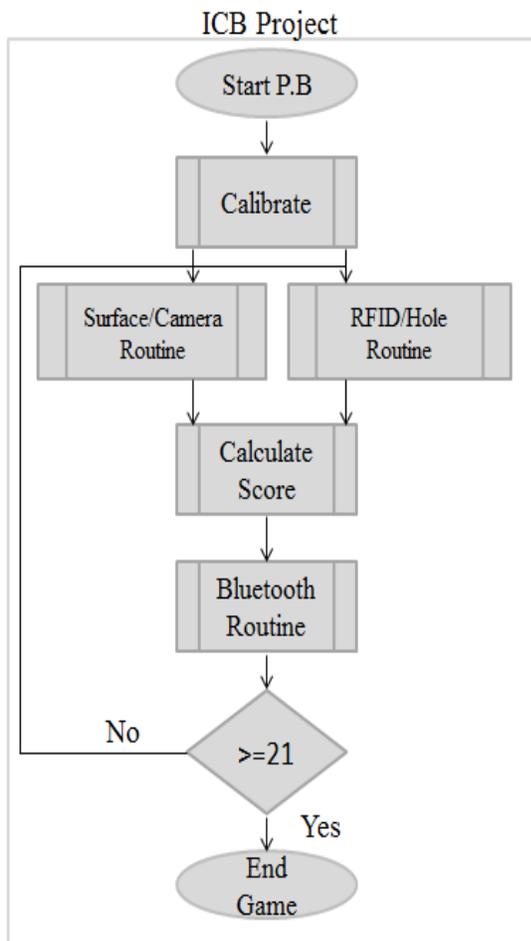


Fig. 2. Intelligent Cornhole Board Top Down Flowchart

IV. HARDWARE

A. Image Processing

A Microsoft LifeCAM HD 3000 webcam is mounted 15 inches behind and 32 inches above the surface and is used to continuously monitor the board's surface during gameplay. The camera provides the LabVIEW program with the number of bags on the surface and each bag's corresponding color.

B. RFID Processing

1) RFID Reader:

The RW-310 reader was manufactured by APSX, and it was chosen for its low voltage power supply (6 volts) and its low power consumption during fast read operation. Additionally, it has a large reading distance depending on the type of passive tags used that other readers in the same price range could not match. This reader also had the capability of placing a tag in quiet mode after it entered the electric field. This feature was necessary to ensure the tag was only counted once per round. The tags reset once the bags are removed from the hole (electric field). The reader is powered by four 1.5 volt batteries wired in series to produce the required 6 volts.

2) APSX C-100:

The APSX C-100 TTL to RS232 level converter allows communications using RS-232 protocol.

3) Passive Tags:

The passive tags are the Texas Instrument Tag-it HF-1 Pro Transponder inlays. They use High Frequency (13.56 MHz), and are compliant with ISO-15693. The aluminum antenna size is 1.7 inches by 2.99 inches. This tag has a read distance of 9.5 inches.

C. Bluetooth Communication

1) Bluetooth Adapter:

The Rocketfish RF-MRBTAD Micro Bluetooth 2.1 EDR Adapter was used for wireless communication between the computer and the mobile device. This adapter had to be installed in the computer that was running the software; because it did not have Bluetooth capability built in. The adapter allows communications to the mobile device up to 10 m or 32.9 ft away from the computer, and allows the use of three different mobile devices at the same time.

2) Android Device:

The android device used in this project was the HTC EVO 4G PH44100 smart phone. Bluetooth capability allowed the use of wireless communication to add and display the score on the mobile device. Having this capability added flexibility to the way the user interfaces with the software.

D. Game Board

ICB's game board is a slightly modified version of a standard cornhole board. The playing surface is a 2x4ft rectangle made of 0.5in plywood with a six inch hole in the surface. The hole has center point located nine inches down from the long side and 12 inches in from the short side. The board rests on two 12in legs that serve to elevate one side of the playing surface to approximately 10° of tilt. A 32in camera mounting pole is mounted 15in behind the elevated side of the board. A basket is mounted underneath the board surface and serves to catch and suspend the bean bags 1in above the RFID reader. Fig. 3. displays the dimensions and layout of the gameboard.

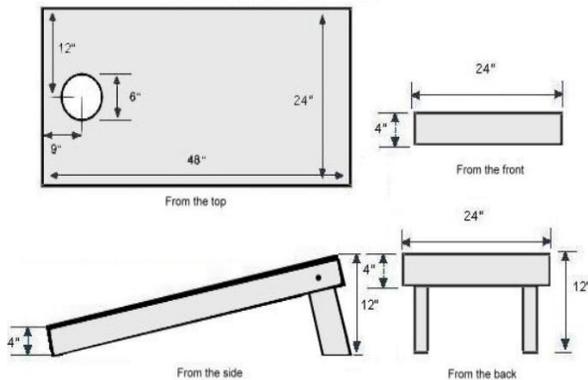


Fig. 3. Board Dimensions

E. Computer

The LabVIEW program responsible for scoring the game is run on an HP Pavilion dv7 personal computer. This computer is equipped with a 17in monitor used for one of the two scoreboards and an Intel I7 processor necessary for running the inefficient LabVIEW code at high speeds.

V. SOFTWARE

A. Image Processing

The image processing portion of the LabVIEW code is responsible for counting how many bean bags are on the surface of the board and the color of each bean bag. Before the game is played, the program must be taught the colors of the bags that the players will use. A separate calibration VI was written to handle this task. The user places a red bag and blue bag in the center of the board. The camera snaps a picture of each bag individually and the program identifies their color spectrums. Calibration must be performed periodically because the bag colors look different to the camera in various lighting conditions. Once the spectrums are identified they are defined as a global parameter for use by the main program. Fig. 4. is a grayscale image captured by the camera.

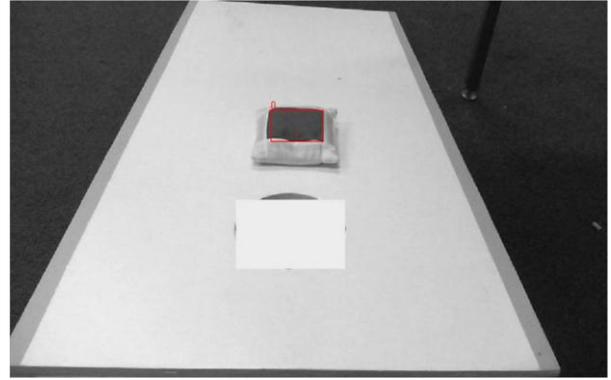


Fig 4. Grayscale Image from Camera

The main image processing program works sequentially in three segments. First the program snaps a picture of the board in grayscale and in color. In grayscale, the red and blue beanbags look like dark objects. Next the grayscale image is used as an input to a single "count objects" function block that frames, counts, locates, and determines the area of each dark object. The frame location of the dark objects serves as each objects' region of interest (ROI). The spectrum of each ROI is compared to the red and blue bag's calibration spectrum. Finally, the program counts the number of color matches identified and the team's surface score is calculated. Fig. 5. is LabVIEW code for calibration of the blue objects.

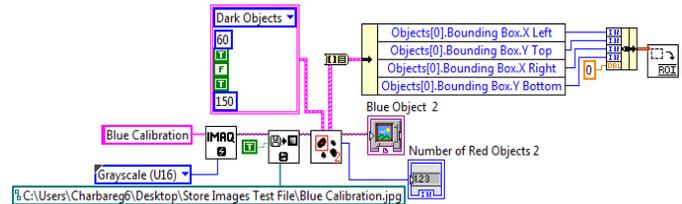


Fig. 5. Subset of Blue Calibration Code

B. RFID Processing

This component of the game is responsible for detecting if a bean bag enters the hole, determining which team the bag belongs to, counting the bag one time per round, and then correctly awarding points. Fig. 6. is a software flowchart that displays a top level overview for the RFID processing.



Fig. 6. Top Level view of the RFID process.

1) Initialization (sub-VI):

The initialization establishes the communications between the software and the hardware. It sets the serial configuration (RS-232), and opens the instrument and sets the VISA attributes. Fig. 7. displays the initialization software.

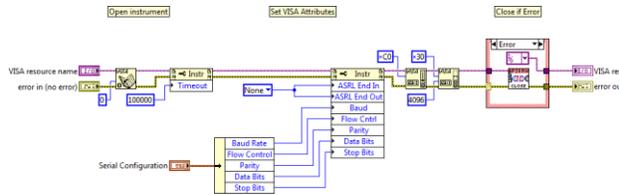


Fig. 7. LabVIEW code to initialize the RW-310 Reader

2) Read Tag (sub-VI):

The read tag sends the serial command to inventory the tag. If a tag is present in the Electric Field it will send its UID to be scored. Fig. 8. displays the code for reading the tag.

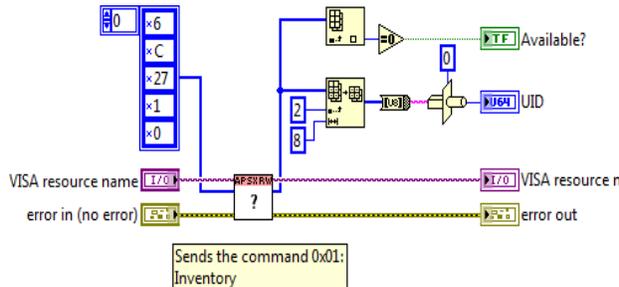


Fig. 8. LabVIEW code to read the tags.

3) Quite Tag (sub-VI):

This sub routine sends the serial command for the tag to stay quiet as long as it is in the Electric Field. The tag will only send its UID one time. Once the tags are removed from the hole they will reset so that they send their UID the next time they enter the hole. Fig. 9. displays the code for quieting the tag.

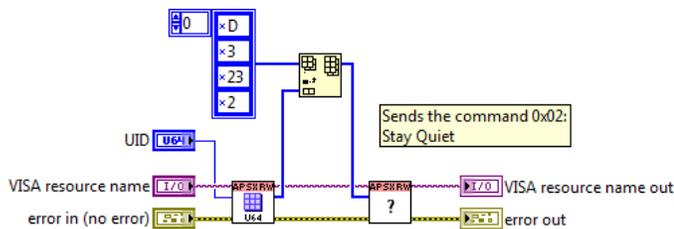


Fig. 9. LabVIEW code to quite the tags.

4) Hole Calculation (VI):

It uses the above mentioned sub-VI's to read the UID from each bag that enters the hole. It compares these values against the known values of the UIDs assigned to the blue team and the red team. If there is a match it adds three points to the appropriate team.

C. Bluetooth Communication

1) LabVIEW Bluetooth:

In order for the Bluetooth to work, it must be determined which device will act as the server and which will act as the client. The main purpose of the server is to listen for incoming connection requests. If a request is detected then a Bluetooth Socket is provided by the server. LabVIEW was chosen to be the server, and it was assigned a UUID. This UUID was then programmed into the Android Device to allow the server and client to conduct a handshake for wireless communication. The server listens for a UUID (using RFCOMM) from the client and after it detects one, it assigns a communications channel number to the client.

The program is set up to wait for 30 seconds to find a device with the same UUID. Once found it displays the MAC address on the front panel in the indicator box. If the program doesn't find a matching UUID after the 30 seconds it causes an error and the program stops listening. The code is wired into a while loop and goes into a write command. The block that is wired into "data out" port is a block of command converting a number to a string. The number that is inputted to that block is a four digit number that represents the score. This score is later added to the current score on the score board. The next block of code is a read command which reads the information from Bluetooth client devices. Finally when the game is over, the program closes all connections to the client.

2) Android Application:

An emulator has to be used in order to build the application (app). For this project the Eclipse emulator was used due to the amount of information on the web. Eclipse uses Java as the program language of choice. The program starts off by importing all the packages/libraries that will be used throughout the entire code. Once the app is open, the program waits until the "Start Game" button has been pushed. Once pushed it verifies that the program is supported and ensures that the mobile device is enabled for Bluetooth. Since this app is called to be paired before opening the app, which mean that each device is aware of each other's existence. This information is then used to initialize the connection.

The program then checks to see if the phone is connected to the right computer, this is a hard coded description of the computer. The program then obtains a Bluetooth socket, by sending the UUID, which means the phones is acting as the client. The Bluetooth Socket is obtained by creating RFCOMM channel when the device compares the UUIDs, in which they should match on server and the client side. Once the devices compare the UUIDs and they match, they then can send data back and forth on Data Streams. The program begins reading the information off the InputStream. The program then reads a String off the InputStream and that is then converted to a four digit number. The number is then spilt into two digital numbers. The first two digits of the four digital numbers represent the red team score and the last two digits represent the blue team score. Each team score gets added to the current score and stored in a register that displays on the screen when the

“End of Round” button is pressed. The program then sets in a loop until the “End of Game” or the “Exit” button is pressed. The “End of Game” button resets the current scores back to zero and the “Exit” button exits out the game completely. Fig. 10. is a picture of the display that is shown by Android Application.



Fig. 10. Picture of the Android Application Display.

VI. CONCLUSION

As technology advances, we find that more and more of life's everyday tasks are becoming automated. This holds true in the field of entertainment as well. ICB was an idea inspired by a game that is seen at tailgates and barbeques all across the United States. Team Bs set out to determine the feasibility of automating the game of cornhole and the potential market value of such a product. Ultimately it was determined that the game of cornhole can be automated but, it must be done in a more sophisticated fashion than described in this report for the concept to appeal to the general public.

VII. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the contributions of the following people for their support in completing this project:

Dr. J. Peebles for his advice, experience and insight into the design process and the technology industry. It was worth its weight in gold.

Dr. J. Skinner for championing our cause at the board of senior projects, and for also providing support as our project advisor.

Dr. Moore for his expert instruction on JAVA and Android Applications.

Dr. M. McKinney for his expert instruction on LabVIEW.

Dr. R. Hayne for his expert instruction on the PIC 18. Although, we eventually changed direction from the PIC 18 the knowledge he provided is something that we can carry with us into our future careers.

Dr. T. Jerse for his tutelage on the mysterious happenings of electromagnetic waves.

Mr. D. Crawford for expediting parts.

VIII. REFERENCES

- [1] J. Essick, *Hands-on Introduction to LabVIEW for Scientists and Engineers*, Vol. I. New York: Oxford University Press, 2009.
- [2] C. S. Horstmann, "Java Concepts," 5th ed., New Jersey: Wiley, 2008.
- [3] D. Lauren, "Android Wireless Applications Development: Android Essentials," 5th ed., New Jersey: Addison-Wesley, 2012.
- [4] "Android Developers." *Android Developers* Available: <http://developer.android.com/>

IX. BIOGRAPHIES



Charles Belekis was born in Charleston, SC, on April 15, 1986. He graduates from the Citadel on May 5, 2012 with a B.S. in Electrical Engineering. He currently has a B.S. in Mathematics from the University of South Carolina which he obtained in 2009. His employment experience includes 1 year as a statistical reporter for Companion Property and Casualty and 2 years experience in HVAC. He recently landed a job at Nucor Steel Berkeley as an electrician following his year long internship there.

Charlie passed the Fundamentals of Engineering Exam in Fall of 2011. His special fields of interest are power, systems, PLCs, and automation.



Kieftha "Kieth" Boyd was born in Darlington SC, on April 29, 1983. He graduates from The Citadel on August 9, 2012 with a B.S. in Electrical Engineering. He also received an A.S. in Electrical Engineer Technology from Florence-Darlington Technical College on May 6, 2003. After receiving his degree, he went to work for Nan Ya Plastic as an E&I Technician where he received experience in troubleshooting and install PLCs, three phase motors, and Industrial instrumentation. Kieth is current

employed at Santee Cooper as contractor drafter for 5 years where he uses CAD software to create and modifying drawings.



James Bredesen (Jim) was born in Newfoundland, Canada on a U.S. Navy Base, on March 15, 1968. He graduates from the Citadel on May 5, 2012 with a B.S. in Electrical Engineering. His employment experience includes 23 ½ years in the United States Navy where he completed tours on 5 different nuclear powered submarines. He also completed tours of duty at Naval Mobile Construction Battalion-5, Submarine Squadron 1, Naval Nuclear Power Training Unit, and the Pentagon. He retired

as a Naval Officer (mustang). Jim passed the Fundamentals of Engineering Exam in Fall of 2011. His special fields of interest are power, communications, and electromagnetic compatibility.