SAFE AND SOUND

By

Quentin Klingler

Ryan Plyman

Tim Press

Final Report for ECE 445, Senior Design, Spring 2019

TA: Thomas Furlong

1 May 2019

Project #20

Abstract

Our group's goal was to design an automated first base umpire system for use in baseball in order to minimize the risk of human error present from close calls made by first base umpires. The final design consisted of a glove subsystem used for monitoring catches and a base subsystem for distinguishing player foot contact. The base also utilizes a green and red LED to indicate if the runner is either safe or out, respectively. We were satisfied with the reliability and accuracy we achieved as we constructed and tested this product. There was room for improvement in some minor aspects of our project. However, these had no effect on the overall performance we achieved by the end of the semester.

Contents

1. Introduction

For well over a century, baseball has been a cornerstone of American sports. Some of America's most well-known baseball players in history have become significant cultural icons and left behind an enormous legacy. Today, the sport is still going as strong as ever. The MLB World Series regularly draws in millions of viewers every year [1]. Everyone will agree that it takes a lot of training and expertise to become a reliable player on the field. The same can also be said for umpires.

Whether we are talking about little league or the major league, becoming an umpire requires extensive training and can be a very selective process [2]. Despite this, umpires are still prone to making mistakes. In 2017 alone, there were 660 overturned umpire calls in the MLB [3]. With all of this in mind, our goal in designing this project was to mitigate the issue of human error present in very close umpire calls; specifically, at first base. The end result of our work was an automated first base umpire system that can reliably call a runner as safe or out as seen in Figure 1.

In the following sections, we will describe the design process and testing procedures we followed that led to the final product. This report will conclude with the major successes of the project, shortcomings of the project, and the lessons learned from the development process.

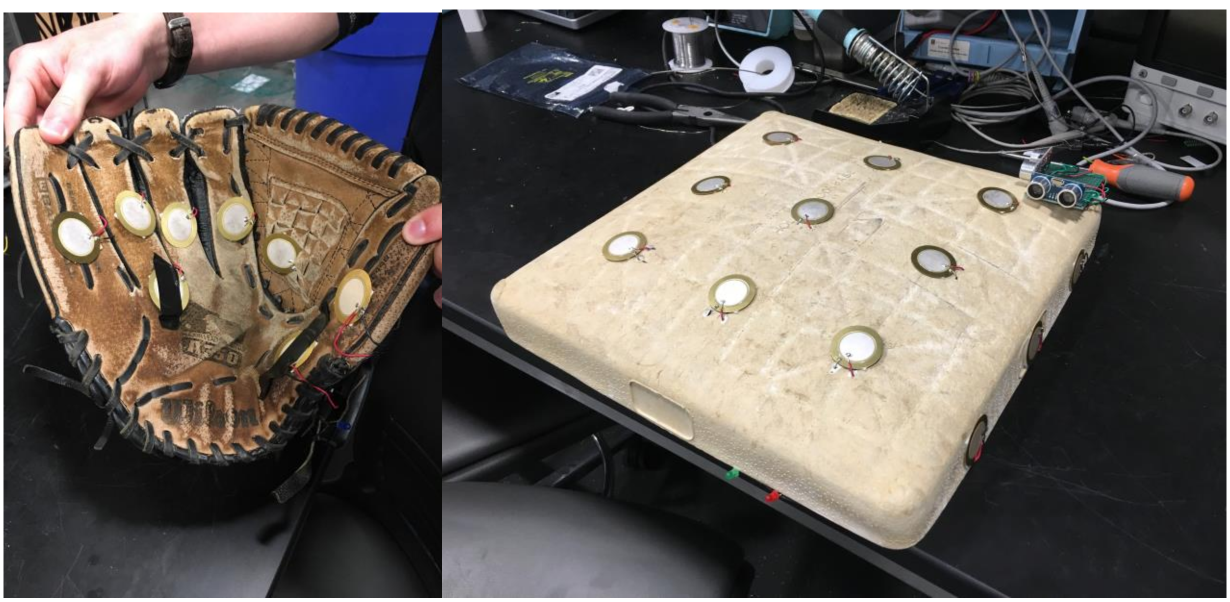

Figure 1. Pictures of the final Safe And Sound design

2. Design

Safe And Sound consists of two subsystems as seen by Figure 2: A glove subsystem that can detect catches, and a base subsystem that can distinguish between runner and baseman foot contact. The base subsystem is also responsible for making the correct call (safe or out) for the runner by lighting up a green LED for safe or a red LED for out. Once the glove detects a catch, it will wirelessly communicate with the base subsystem to inform it of the catch. The base subsystem uses the order of player feet arrival and catch events to make the proper safe or out call decision.

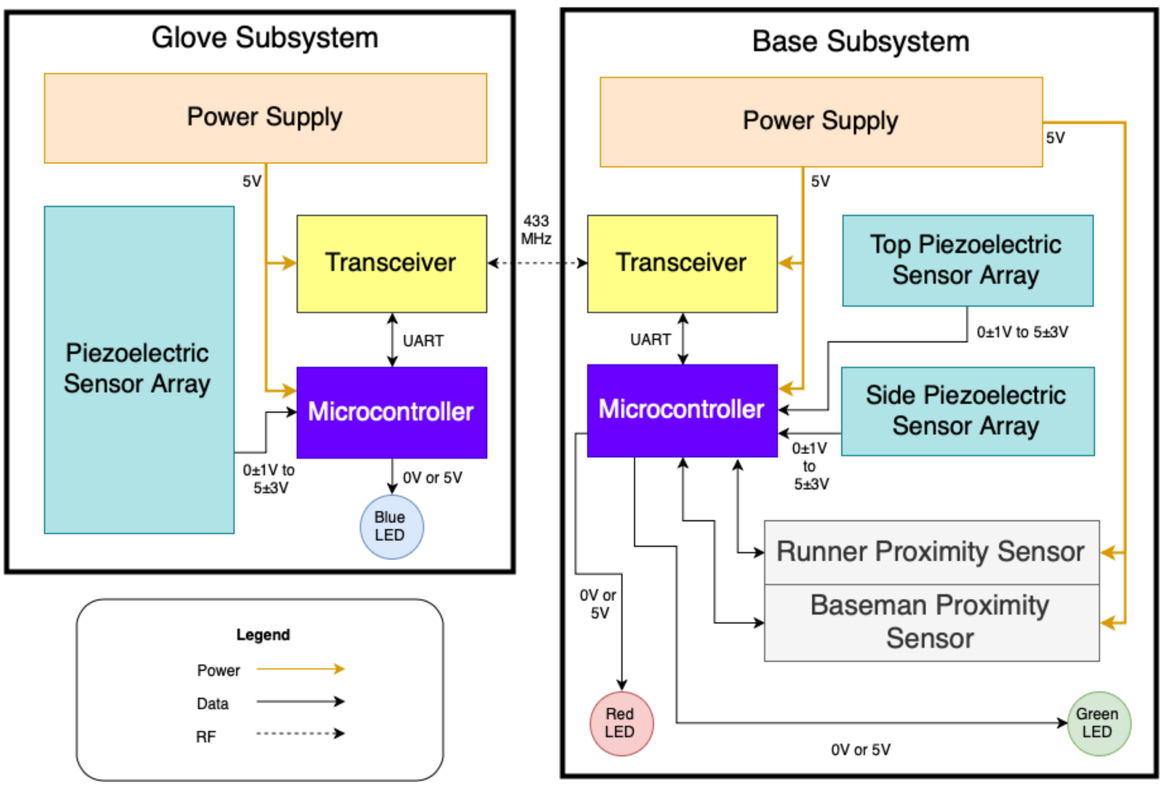

Figure 2. Block diagram of the final design

2.1 Glove Subsystem

2.1.1 Piezoelectric Sensors

Piezoelectric sensors are ceramic diaphragms that produce a voltage drop when subjected to mechanical stress. They work best for instantaneous impact events, so it seemed like an obvious choice to implement catch detection. The glove subsystem has eight piezoelectric sensors which are wired in a series array and mounted to the inner surface of the glove. When a ball is tossed into the glove, they produce an instantaneous voltage drop of at least 5 V ± 2 V, which we refer to as the glove impact voltage threshold (GIVT). When the impact voltage reaches this threshold, the glove will use its HC-12 transceiver module to inform the base subsystem of the catch.

2.1.2 Microcontroller

The glove subsystem uses an ATMega328P microcontroller as seen in Figures 4 and 5 to monitor readings from the piezo array and control its HC-12 transceiver module. The microcontroller also lights up a blue LED as seen in Figure 3 whenever the glove subsystem receives the catch acknowledgement message from the base.

2.1.3 Transceiver

The HC-12 is a pre-assembled long-range wireless communication module often used with Arduinos. The glove subsystem uses the HC-12 to establish wireless communication between itself and the base subsystem. Whenever the impact voltage reaches the GIVT, the microcontroller tells the HC-12 to send a small message ("c" for catch) to the base's HC-12. Its physical dimensions are 27.8 mm × 14.4 mm × 4 mm, so it fits snugly into the raspberry pi case on the back of the glove next to the PCB.

2.1.4 Power Supply

Based on the calculations listed within Table 4, a rechargeable 9 V, 800 mAh, Li-ion battery was ideal to power the glove subsystem. The microcontroller and the HC-12 draw power from the output of a 5 V voltage regulator on the PCB.

Figure 3. Assembled glove PCB on the back of the glove. The HC-12 antenna is seen sticking out underneath the power switch, and the acknowledgement LED is on the left.

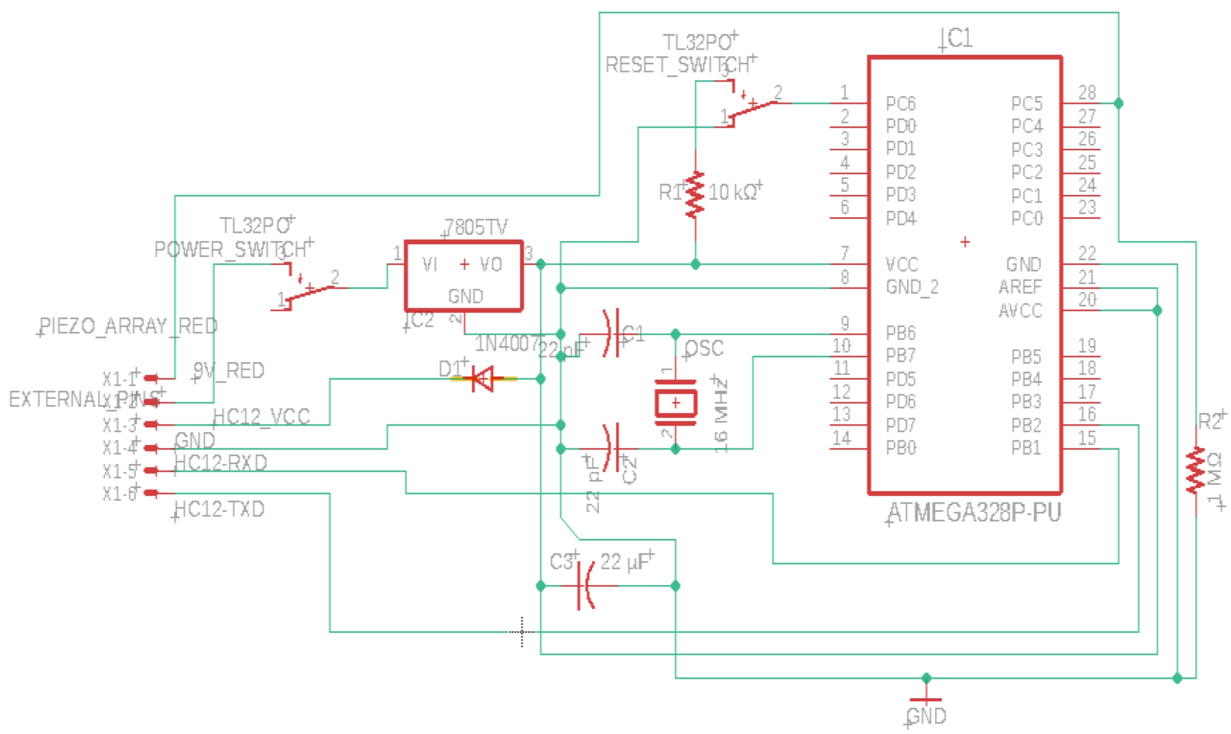

Figure 4. Circuit schematic for the glove subsystem

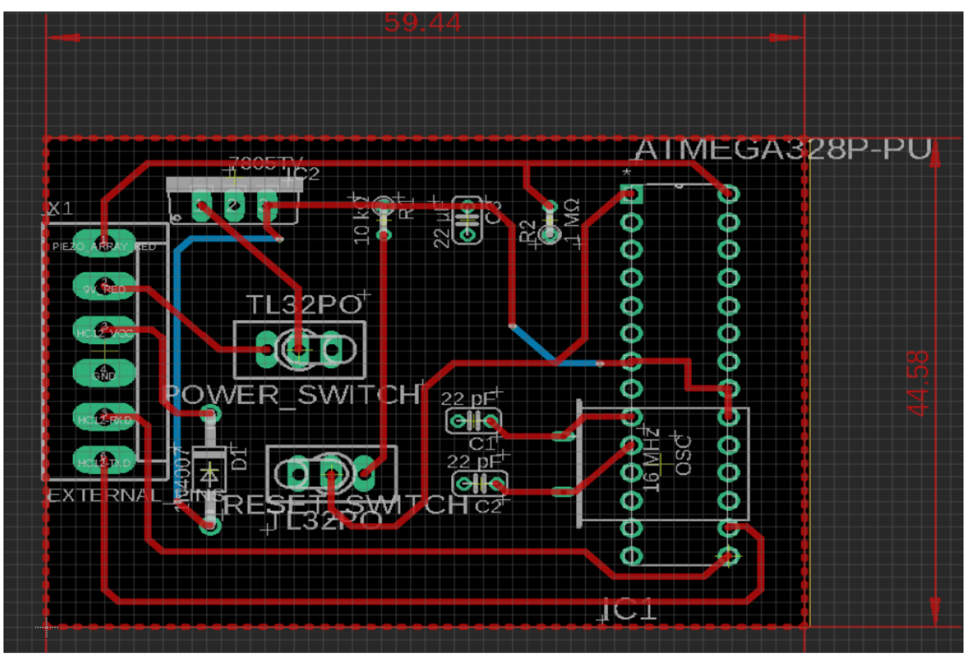

Figure 5. PCB layout for the glove subsystem. Circuit dimensions are in millimeters.

2.2 Base Subsystem

2.2.1 Piezoelectric Sensors

The piezoelectric sensors used in the base subsystem are the same type of piezoelectric sensors used in the glove subsystem. However, unlike the glove subsystem, the base subsystem uses two distinct series arrays of sensors instead of just one as seen by Figure 9. The nine-sensor array on top of the base detects runner foot impact, while the three-sensor array on the side of the base detects baseman foot impact. This design decision was made to give the baseman more flexibility in how their feet can approach the base. Additionally, both sensor arrays utilize a base impact voltage threshold (BIVT) of 5 V ± 2 V.

2.2.2 Proximity Sensors

The HC-SR04 is an ultrasonic proximity sensor used to measure the distance of objects within its viewable range. It works by sending out an ultrasonic pulse and using the travel time of reflected sound waves to determine the distance in centimeters of nearby objects. These sensors play a similar role to the piezoelectric sensors, such that they monitor the base surface for approaching feet. Two proximity sensors were mounted on the corner of the base perpendicular to each other in order to prevent interference. One sensor monitors the base surface for the runner's foot across the path between home plate and first base, while the other sensor monitors the base surface for the baseman's foot across the path between first and second base.

2.2.3 Microcontroller

Like the glove subsystem, the base subsystem also utilizes an ATMega328P as seen by Figures 6, 7, and 8 to monitor sensor feedback and control its HC-12 transceiver module. It's also responsible for analyzing the order of foot and catch detection events in order to make the correct call decision for the runner. If the runner is determined to be safe, it will light up a green LED. Otherwise, it will light up a red LED. Lastly, the microcontroller will also tell the base's HC-12 to send a short acknowledgement message whenever it receives a catch message from the glove subsystem.

2.2.4 Transceiver

Most importantly, the base subsystem also contains an HC-12 transceiver module so it can receive a catch detection message from the glove subsystem. When a catch detection message ("c" for catch) is received by the HC-12 module, it will send a short acknowledgement message ("a" for acknowledge) back to the glove subsystem's HC-12 module. Furthermore, the base subsystem gives every registered catch an expiration time of 10 seconds to give the first baseman enough time to run towards first base.

2.2.5 Power Supply

The power supply for the base subsystem is exactly the same setup as the glove subsystem. It utilizes a rechargeable 9 V, 800 mAh, Li-ion battery in tandem with a 5 V voltage regulator to power the components soldered to the PCB.

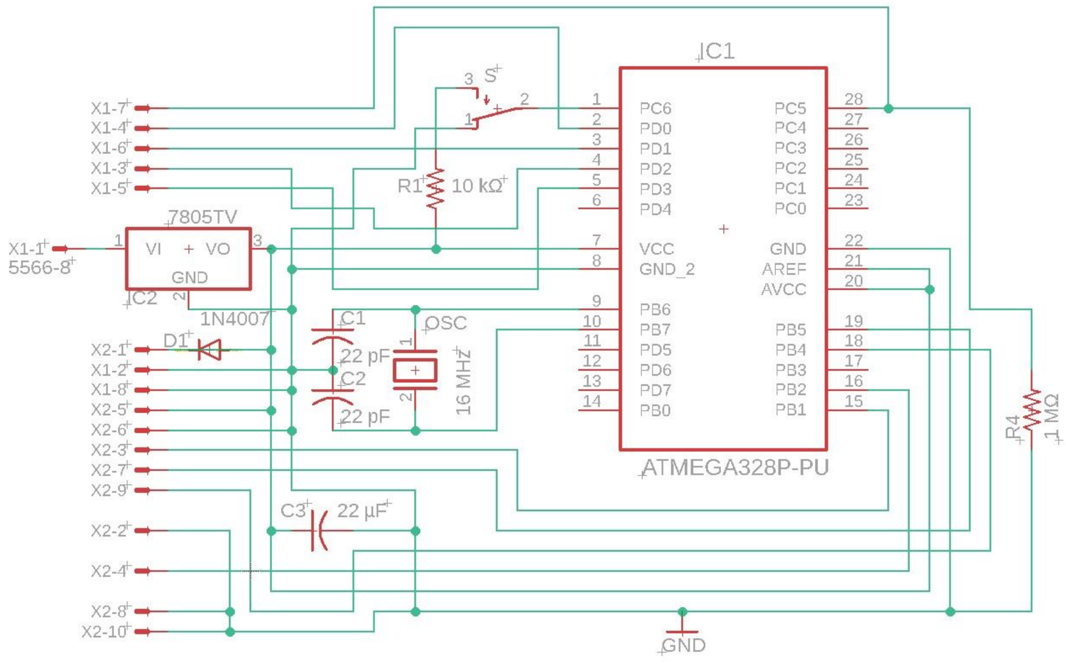

Figure 6. Circuit schematic for the base subsystem

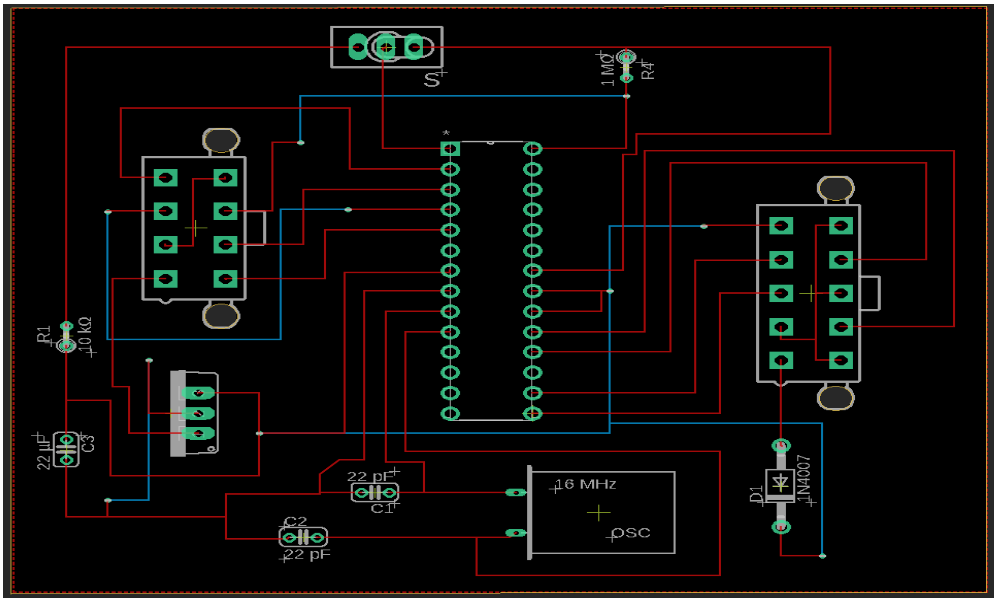

Figure 7. PCB layout for the base subsystem

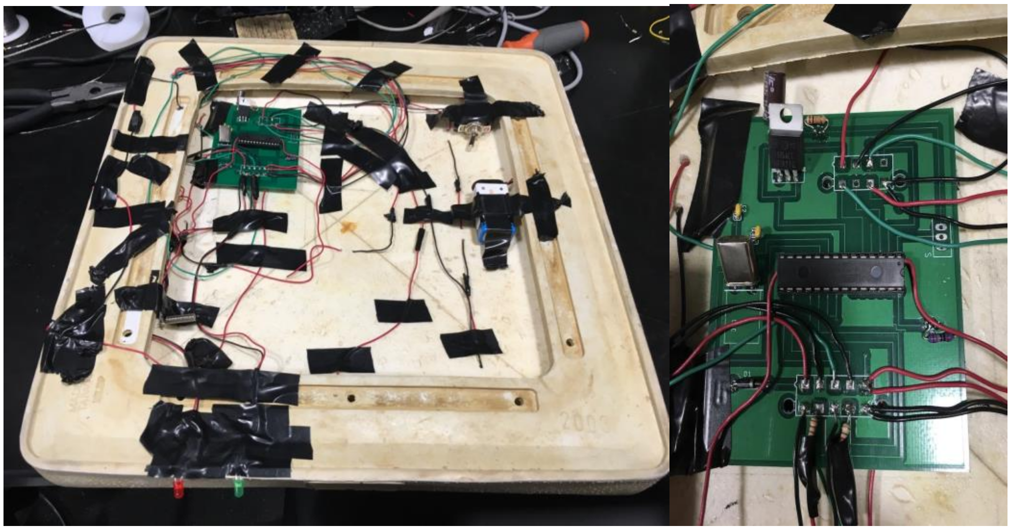

Figure 8: Pictures of the assembled base subsystem. In the left image, the PCB, HC-12, power switch, LEDs, and battery are all visible.

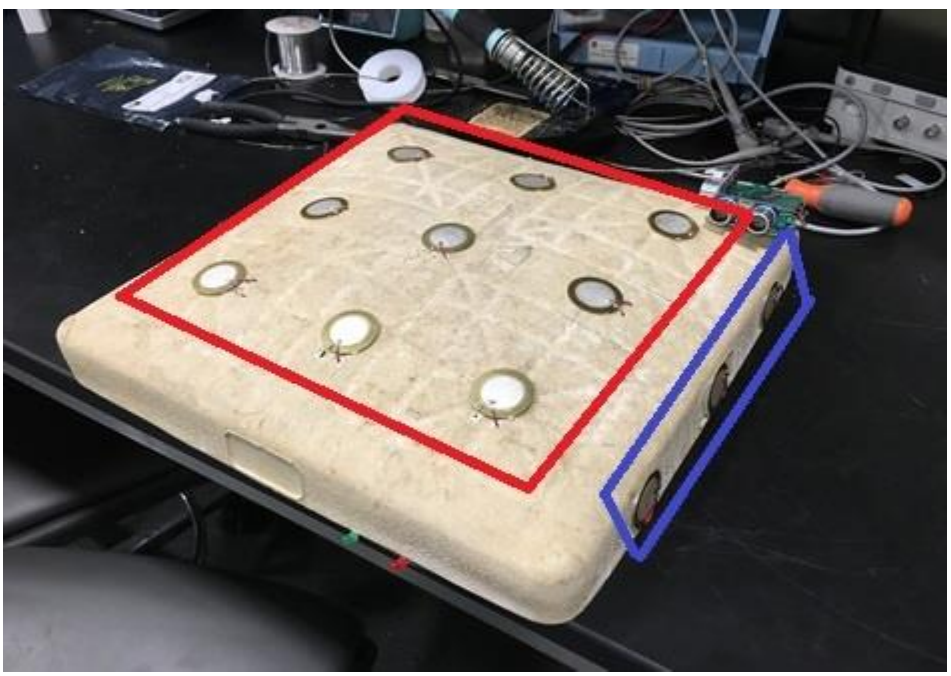

Figure 9. Piezoelectric sensor array layout for the base subsystem. The red piezo array looks for runner foot contact, while the blue piezo array looks for baseman foot contact.

3. Design Verification

3.1 Catches

To verify proper communication between the glove and base subsystems, we added a blue LED to the glove subsystem's PCB. The glove's microcontroller will light up this LED for 500 ms if it receives the expected acknowledgement message ("a") from the base subsystem. Furthermore, the piezoelectric sensors in the glove required occasional repositioning to detect ball impacts properly. Eventually, the glove's catch detection reliability vastly improved once an optimal positioning of the sensors was determined. The placement of the piezoelectric sensors was on each of the four fingers of the glove in addition to two sensors placed on the thumb, one sensor placed in the webbing, and one sensor placed near the palm of the glove as seen by Figure 1.

3.2 Proper Safe Call

Runner foot arrival occurs when *both* the runner's proximity sensor and piezo array on top of the base detect either foot. In order for the base subsystem to make a proper safe decision call, runner foot arrival must occur before the baseman triggers an out-decision call.

3.3 Proper Out Call

Unlike the runner, baseman foot arrival occurs when *either* the baseman's proximity sensor or piezo array detects either foot. In other words, the baseman can either place their foot on the side of the base or on top of it. A proper out decision call occurs when the base subsystem receives a catch message and there is detection of the baseman's foot. If the out-decision call occurs before the runner's foot arrives, the red LED lights up to indicate an out as seen by Figure 10. Otherwise, the green LED lights up to indicate the runner is safe.

Figure 10. Snapshot of a slow-motion video where the runner was determined to be out (red LED) due to the detection of a catch and the first baseman's foot before the detection of the runner's foot

4. Costs

4.1 Labor

Our labor cost is estimated using the average annual salary of an electrical engineer from Illinois, $71,166 [4]. Using this annual salary, we are able to calculate an hourly salary based on a 40-hour work week across all 52 weeks in a year. Thus, the hourly wage was determined to be $34.21 per hour. Additionally, if we estimate that we worked 10 hours for each of the 16 weeks of the semester that we had to complete this project, we are able to calculate the total labor cost for three electrical engineers as seen by Equation (1).

$$3*\frac{\$71,166}{year}*\frac{1\,year}{52\ weeks}*\frac{1\ week}{40\ hours}*\frac{10\ hours}{1\ week}*16\ weeks*2.5=\$41,057.31 \tag{1}$$

4.2 Parts

As seen in Table 1, the total cost for designing Safe And Sound was determined by estimating the total labor and total parts cost for the entire project.

Table 1. Parts Costs

Parts	Manufacturer	Product #		Price/part ($)	Quantity
PCB+Shipping	PCBWay			22	1
Microcontroller	Microchip			1.38	2
Piezo sensors	Murata Electronics North America	7BB-35-3L0		1.47	15
4 Rechargeable 9V Batteries + Charger	EBL	FBA_LN-8161*4+LN-6415		21.99	1
9V Battery Connector	Pacific Science Supply	P56045		3.1	2
5V Voltage Regulator	ECE Supply Shop	LM7805ACT-ND		0.57	2
Crystal Osc	ECE Supply Shop	520-HCA1600-SX		0.57	2
HC-12 Transceiver	Seeed	113990039		12.9	2
Base	Schutt (Jack Corbett MLB Hollywood Base)	12901040		99.99	1
Glove	Rawlings	SL125BF		89.99	1
Cleats	New Balance	L4040		89.99	1
Baseball	Rawlings	ROML		12.99	1

Assorted resistors	ECE Supply Shop	20J10K	10k Ohms	2.03	2
	Yageo	CFR-25JB-52-220R	220 Ohms	0.1	2
	Yageo	CFR-25JB-52-1M	1M Ohms	0.1	1
Assorted capacitors	ECE Supply Shop	1C25Z5U223M050B	22 pF	0.11	4
	ECE Supply Shop	1C20Z5U103M050B	10 pF	0.23	1
LED	ECE Supply Shop	HLMP3507	Green	0.18	1
	ECE Supply Shop	HLMP3301	Red	0.16	1
Reset button	Sparkfun	9276		1.95	2
5 Proximity Sensors	SunFounder	B00E0NXTJW		8.78	1
Pack of Adhesive Squares	3M	B07PBSBGMB		3.99	1
Electrical Tape	3M	6132-BA-10		4.1	1
Testing					
Arduino	Sparkfun	13975		19.95	1
			Total Cost	$ 438.03	
Labor			Annual EE Salary	Hourly EE Salary	
			$ 71,166.00	$ 34.21	
			Total Labor	$ 41,057.31	
			Total Cost+Labor	$ 41,495.34	

5. Conclusion

5.1 Accomplishments

Overall, Safe And Sound satisfied all of the high-level requirements it needed to be successful. The base and glove subsystems reliably communicated with each other whenever a catch occurred. Additionally, the glove subsystem was able to detect that a catch happened a majority of the time using its piezoelectric sensors. The base subsystem was able to distinguish between runner and first baseman foot contact using two piezoelectric sensor arrays and ultrasonic proximity sensors. Furthermore, the base subsystem was also able to make the correct safe or out decision call based on when either player triggered their respective sets of sensors. Thus, we believe that our project is a success since it's able to act correctly as an automated first base umpire system that's able to make correct calls even during very close plays that would happen too quickly for a human to judge.

5.2 Ethical considerations

Our design introduced multiple ethical and safety concerns for baseball players and umpires. Player safety was the number one concern for this project, and it was our primary goal that players felt completely safe when using our system. Throughout the development and testing process, any technical decisions regarding specific components or design layouts tried to ensure player safety. Thus, there was no intent to use our project in the rain in order to protect players and the system itself from harm. Furthermore, we tried to mitigate risks by following the IEEE Code of Ethics in order "to hold paramount the safety, health, and welfare of the public" [5]. It was our job to develop a safe product for the use of baseball leagues of all age levels.

It takes a lot of hard work and dedication to become an MLB umpire [2]. With the possibility of complete automation of all umpire-related tasks in future iterations of this technology, it would seem like Safe And Sound is doing a disservice for professional umpires. However, this technology does not have to replace umpires entirely. It can aid umpires for when they have to make close calls. In addition, there are still many other important tasks that have not been delegated to automated technology like calling strikes, foul balls, and ejecting unruly players or managers. Furthermore, code number five in the IEEE code of ethics suggests, "the understanding by individuals and society" of how the emergence of intelligent systems within sports will affect the game. [5]. This is an important goal to achieve for projects like ours so they can find acceptance as tools that can only enhance the accuracy of decision making to allow for fair and unbiased outcomes.

5.3 Uncertainties

In the original Safe And Sound design, the plan was to employ two RFID readers in order to distinguish between runner and baseman foot contact. This also would've required players to have baseball cleats equipped with RFID tags. However, since the ATMega328P microcontroller can only listen to one serial input port at a time, the RFID reader and tag system could give false readings depending on which reader was active or idle. Thus, this system could not reliably determine which player arrived at the base first. Furthermore, after testing each RFID reader independently connected to their own respective microcontroller, it was discovered that frequency interference would occur preventing the readers from operating properly when the two were less than a foot away from each other. This would limit the position of them when placed under the base. Additionally, delays would occur between the slave-master relationship, each RFID readers' microcontroller, and the central microcontroller causing further time of event reliability issues that led to wrong safe or out decision calls. Hence, implementation of ultrasonic proximity sensors was a better alternative in order to provide the needed player contact identification and to make reliable and correct decision calls.

5.4 Future work

In order to improve the design of this project for future implementation, a few areas would need more observation. First, one of the largest problems we faced during the construction and testing phases of our project was the strength of the solder joints of the piezoelectric sensor leads. In order to help prevent the leads from disconnecting from the piezoelectric sensors when either a catch or a foot impact occurs, the solder joint strength for all of these sensors would need some fortification. Another problem that will need addressing for future implementation is the physical footprint of the circuit for the glove subsystem. Since the circuit attached to the glove is currently bulky, the creation of a more compact circuit could be a fix to this problem. An alternative location for the placement of this circuit could also be an interesting idea to explore, such as creating a wrist-mounted circuit for the glove subsystem. Furthermore, another problem that would need more observation for future iterations of this project is the physical footprint of the proximity sensors on the base corner. In order to prevent damage to the sensors or from interfering with players during a game, the installation of a more rigid structure to house these sensors would be a good direction to take in order to protect them from dirt and/or collision. Additionally, we could explore other alternatives to these sensors that would be less intrusive to players such as integrating an RFID reader and tag system into our currently designed base subsystem. Ultimately, if we were able to address all of the problems listed above, our project's reliability and game-readiness would unquestionably improve.

References

[1] T. Fitzgerald, “World Series Ratings: The Highs And Lows From The Past 50 Years,” *Forbes*, 23-Oct-2018. [Online]. Available: https://www.forbes.com/sites/tonifitzgerald/2018/10/23/world-series-ratings-the-highs-and-lows-from-the-past-50-years/#2b175f372047. [Accessed: 30-Apr-2019].

[2] "How Much Are MLB Umpires Paid?", *Bleacher Report,* 2018. [Online]. Available: https://bleacherreport.com/articles/2773321-how-much-are-mlb-umpires-paid. [Accessed: 30-Apr-2019].

[3] "Number of MLB reviews, overturned calls drop in 2017", *USA Today*, 2017. [Online]. Available:https://www.usatoday.com/story/sports/mlb/2017/10/01/number-of-replays-overturned-calls-drop-in-mlb/106223230/. [Accessed: 30- Apr- 2019].

[4] “Illini Success”, Engineering Report 2016-2017, 2017 [Online]. Available: https://ecs.engineering.illinois.edu/files/2018/03/Engineering_Report_2016-2017_FINAL.pdf. [Accessed 30- Apr- 2019]

[5] Ieee.org, "IEEE Code of Ethics", 2019. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 30- Apr- 2019].

Appendix A Requirement and Verification Tables

Table 2. Glove Subsystem Requirements and Verifications for Piezoelectric Sensors

Requirements	Verifications
Piezo array must produce a voltage reading above 5V±0.5V when a ball impacts the glove	1. Throw ball into glove 2. Check if the blue LED is high for verification

Table 3. Glove Subsystem Requirements and Verifications for Microcontroller

Requirements	Verifications
Distinguish between piezo readings below and above 5V±0.5V	1. Throw ball into glove 2. Check if the blue LED is high for verification
Be able to send event text to HC-12 transmitter pin	1. Throw ball into glove 2. Check if the blue LED is high for verification

Table 4. Glove Subsystem Requirements and Verifications for Power Supply

Requirements	Verifications
Power source must last for 3 hours to last for an entire game	1. Draw 200mA (HC-12 current draw) + 0.3mA (microcontroller current draw) from the battery 2. Monitor output voltage of battery for 3 hours, ensuring it doesn't deplete below 5V earlier Battery Life = Battery Capacity/Current Draw = (800 mAh / 200.3 mA) = 4 hours Assumptions: 9V battery contains 800 mAh of capacity

Table 5. Glove Subsystem Requirements and Verifications for HC-12 Transceiver

Requirements	Verifications
Reliably send event text (e.g. “c”) to the base subsystem’s transceiver module in less than 10ms after the piezo array spikes above the glove impact voltage threshold	1. Throw ball into glove 2. Check if the blue LED is high for verification

Table 6. Base Subsystem Requirements and Verifications for Piezoelectric Sensors

Requirements	Verifications
Piezo array on top of the base must produce a voltage reading of 5V±2V when a foot impacts the base	1. The runner’s shoe is less than 30 cm from front of proximity sensor 2. Step on base piezoelectric sensors 3. Green LED turns high
Piezo array on the first baseman’s side of the base must produce a voltage reading of 5V±2V when a foot impacts the base	1. The baseman’s foot touching the side piezo array 2. Throw ball into glove 3. Red LED turns high

Table 7. Base Subsystem Requirements and Verifications for Microcontroller

Requirements	Verifications
Distinguish between piezo readings for top array below and above 5V±2V	1. The runner's shoe is less than 30 cm from front of proximity sensor 2. Step on base piezoelectric sensors 3. Green LED turns high
Distinguish between piezo readings for side array below and above 5V±2V	1. The baseman's foot touching the side piezo array 2. Throw ball into glove 3. Red LED turns high
Be able to receive runner's distance from proximity sensor 1 when either of the runner's shoes approach the base	1. The runner's shoe is less than 30 cm from front of proximity sensor 2. Step on base piezoelectric sensors 3. Green LED turns high
Be able to receive baseman's distance from proximity sensor 2 when either of the baseman's shoes approach the base	1. The baseman's shoe is less than 30 cm from front of proximity sensor 2. Throw ball into glove 3. Red LED turns high
Receive event text on HC-12 receiver pin	1. Throw ball into glove 2. Check if blue LED is high on glove for verification

Table 8. Base Subsystem Requirements and Verifications for Power Supply

Requirements	Verifications
The power source must last for at least 3 hours to last for an entire game	1. Draw 200mA (HC-12 current draw) + 0.3mA (microcontroller current) from the 9V and voltage regulator combination. 2. Monitor output voltage of the 9V battery for 3 hours, ensuring it doesn't deplete below 5 V earlier.

Table 9. Base Subsystem Requirements and Verifications for HC-12 Transceiver

Requirements	Verifications
Reliably receive event text (e.g. “a”) from the glove subsystem’s transceiver module in less than 10 ms.	1. Throw ball into glove 2. Check if blue LED is high on glove for verification

Table 10. Software Requirements and Verifications

Requirements	Verifications
Glove subsystem’s software should initiate message transmission if PAV > IV	1. Throw ball into glove 2. Verify the blue LED on the glove is set high
Base subsystem’s software should set the green LED high if (1) runner’s proximity sensor has detected the runner’s foot (2) PAV > IV for the top piezo array and (3) both these events happen before baseman foot contact and ball-glove contact is confirmed	1. Have runner approach and step on base 2. Verify green LED is set if baseman’s foot is on base without ball 3. Repeat, but baseman should have foot off base and catch ball
Base subsystem’s software should set the red LED high if (1) it received the “c” message from the glove subsystem, (2) the baseman’s proximity sensor has detected the baseman’s foot, or PAV > IV for the side piezo array and (3) all these events happen before runner foot contact is confirmed	1. Have baseman approach base and catch ball before runner approaches 2. Verify the red LED has been set

Appendix B Software Code

```
#include <SoftwareSerial.h>
const int HC12TxPin = 10;
const int HC12RxPin = 9;
const int piezoArrayPin = A5;
const int piezoArray2Pin = A3;
const int redLedPin = 13;
const int greenLedPin = 12;
const int runnerTrigPin = 3;
const int runnerEchoPin = 2;
const int basemanTrigPin = 8;
const int basemanEchoPin = 5;
const int footDistanceThreshold = 30; //30cm from proximity sensor
unsigned long catchtime;
unsigned long current_time;
unsigned long catch_expiration = 10000;
long duration;
int distance;
int currSensorVal;
int voltageDifference;
int prevSensorVal = 0;
int prevSensorVal2 = 0;
int impactThreshold = 1023;
int differenceThreshold = 700;
int catchvar = 0;

SoftwareSerial HC12(HC12TxPin, HC12RxPin);

void setup()
{
  pinMode(redLedPin, OUTPUT);
  pinMode(greenLedPin, OUTPUT);
  digitalWrite(redLedPin, LOW);
  digitalWrite(greenLedPin, LOW);
  pinMode(runnerTrigPin, OUTPUT); // Sets the runnerTrigPin as an Output
  pinMode(runnerEchoPin, INPUT); // Sets the runnerEchoPin as an Input
  pinMode(basemanTrigPin, OUTPUT); // Sets the runnerTrigPin as an Output
  pinMode(basemanEchoPin, INPUT); // Sets the runnerEchoPin as an Input
  Serial.begin(9600);
  HC12.begin(9600);
}

```

Figure 11. First Part of Base Subsystem Software Code Written in Arduino IDE

```
void loop()
{
  if(HC12Loop() == 1)
  {
    catchvar = 1;
    catchtime = millis();
  }
  if(catchvar == 1)
  {
    current_time = millis();
  }
  if(runnerLoop() == 1 && basemanLoop() == 0 && piezoarray2Loop() == 0)
  {
    if(piezoarrayLoop() == 1)
    {
      digitalWrite(greenLedPin, HIGH);
      delay(2000);
      digitalWrite(greenLedPin, LOW);
      catchvar = 0;
    }
  }
  else if((basemanLoop() == 1 || piezoarray2Loop() == 1) && runnerLoop() == 0 )
  {
      if(catchvar == 1)
      {
        digitalWrite(redLedPin, HIGH);
        delay(2000);
        digitalWrite(redLedPin, LOW);
        catchvar = 0;
      }
  }
  else if((basemanLoop() == 1 || piezoarray2Loop() == 1) && runnerLoop() == 1)
  {
    if(piezoarrayLoop() == 1)
    {
      digitalWrite(greenLedPin, HIGH);
      delay(2000);
      digitalWrite(greenLedPin, LOW);
      catchvar = 0;
    }
    else
    {
      if(catchvar == 1)
      {
        digitalWrite(redLedPin, HIGH);
        delay(2000);
        digitalWrite(redLedPin, LOW);
        catchvar = 0;
      }
    }
  }
  if(current_time - catchtime == catch_expiration)
  {
    catchvar = 0;
  }
}
```

Figure 12. Second Part of Base Subsystem Software Code Written in Arduino IDE

```
//runner's sensor!
int runnerLoop()
{
  // Clears the runnerTrigPin
  digitalWrite(runnerTrigPin, LOW);
  delayMicroseconds(2);

  // Sets the runnerTrigPin on HIGH state for 10 micro seconds
  digitalWrite(runnerTrigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(runnerTrigPin, LOW);

  // Reads the runnerEchoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(runnerEchoPin, HIGH);

  // Calculating the distance
  distance= duration*0.034/2;

  if(distance < footDistanceThreshold)
  {
    return 1;
  }
  else
  {
    digitalWrite(redLedPin, LOW);    // turn the LED off by making the voltage LOW
    return 0;
  }
}

//baseman's sensor!
int basemanLoop()
{
  // Clears the runnerTrigPin
  digitalWrite(basemanTrigPin, LOW);
  delayMicroseconds(2);

  // Sets the runnerTrigPin on HIGH state for 10 micro seconds
  digitalWrite(basemanTrigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(basemanTrigPin, LOW);

  // Reads the runnerEchoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(basemanEchoPin, HIGH);

  // Calculating the distance
  distance= duration*0.034/2;

  if(distance < footDistanceThreshold)
  {
    return 1;
  }
  else
  {
    digitalWrite(greenLedPin, LOW);   // turn the LED off (LOW is the voltage level)
    return 0;
  }
}
```

Figure 13. Third Part of Base Subsystem Software Code Written in Arduino IDE

```
int HC12Loop()
{
  //check for new available data
  if((char)HC12.read() == (char)'c')
  {
    //send acknowledgement to the glove
    HC12.write("a");
    return 1;
  }
  else
  {
    return 0;
  }
}

int piezoarrayLoop()
{
  // read the value from the sensor:
  currSensorVal = analogRead(piezoArrayPin);
  voltageDifference = currSensorVal - prevSensorVal;

  //send catch message if criteria is met
  if(currSensorVal == impactThreshold && voltageDifference > differenceThreshold)
  {
    return 1;
  }
  else
  {
    return 0;
  }

  //update previous sensor value
  prevSensorVal = currSensorVal;
  delay(200);
}

int piezoarray2Loop()
{
  // read the value from the sensor:
  currSensorVal = analogRead(piezoArray2Pin);
  voltageDifference = currSensorVal - prevSensorVal2;

  //send catch message if criteria is met
  if(currSensorVal == impactThreshold && voltageDifference > differenceThreshold)
  {
    return 1;
  }
  else
  {
    return 0;
  }

  //update previous sensor value
  prevSensorVal2 = currSensorVal;
  delay(200);
}
```

Figure 14. Fourth Part of Base Subsystem Software Code Written in Arduino IDE

```
#include <SoftwareSerial.h>

SoftwareSerial HC12(10,9); //hc-12 tx pin, hc-12, tx pin

int sensorPin = A5; //piezo lead
int blueLedPin = 8; //confirms base acknowledgement of catch
int currSensorVal;
int voltageDifference;
int prevSensorVal = 0;
int impactThreshold = 1023;
int differenceThreshold = 700;

void setup()
{
  Serial.begin(9600);
  HC12.begin(9600);
  digitalWrite(blueLedPin, LOW);
}

void loop()
{
  //set blue led low
  digitalWrite(blueLedPin, LOW);

  // read the value from the sensor:
  currSensorVal = analogRead(sensorPin);
  voltageDifference = currSensorVal - prevSensorVal;

  //send catch message if criteria is met
  if(currSensorVal == impactThreshold && voltageDifference > differenceThreshold)
  {
    //send the message to the base here
    HC12.write("c");
  }

  //ack should be received almost immediately

  if(HC12.available())
  {
    if((char)HC12.read() == (char)'a')
    {
      digitalWrite(blueLedPin, HIGH);
      delay(500);
    }
  }

  //update previous sensor value
  prevSensorVal = currSensorVal;

}
```

Figure 15. Final Iteration of Glove Software Code Written in Arduino IDE