TripleS - Smart Stick System

Team 22

Pranav Nair, Ritvik Manda, Shivam Patel

Dongming Liu

Sep 19, 2024

ECE 445

Introduction

Problem

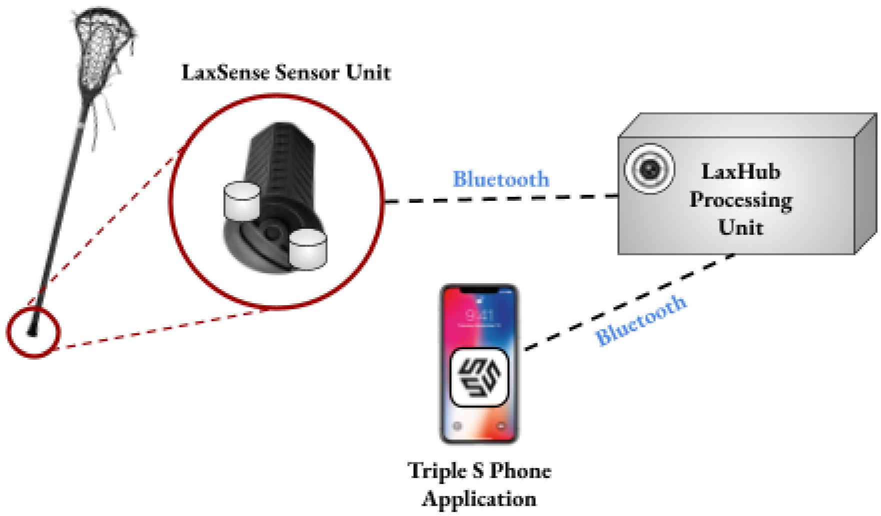
Lacrosse players and coaches currently lack real-time, detailed performance metrics to help improve their gameplay. Traditional training methods rely heavily on subjective observation, which is not very consistent. No tools such as those available for other sports like baseball, golf, soccer, etc are available to monitor and improve lacrosse form and accuracy, especially when a player is training alone. Since lacrosse is not a well known sport, it becomes difficult for beginners and enthusiasts to start learning the mechanics of the stick and being proficient with it.

Solution

This project aims to address the need for a smart, data-driven tool that can measure shot speed, accuracy, and stick form, providing lacrosse players with accurate and instant feedback to enhance their training and technique. We seek to help experienced players to obtain performance data while also aiding beginners in strengthening their form and tactics. Our proposed system consists of a hub unit, a monitoring device fitted to the lacrosse stick, and a mobile app. The base or hub unit (known as the LaxHub) is the main processing unit, and will receive data from the remote unit as well as use a built in camera and lcd screen. We aim to use computer vision based processing to monitor a player's form, suggest changes, and track progress. A secondary subsystem is meant to be securely fit onto the end of a lacrosse stick (known as the LaxSense). This unit contains sensors to transmit information like swing speed and angle back to the LaxHub for processing and estimating metrics like ball speed and trajectory. Finally, the TripleS application provides detailed feedback based on LaxHub processing, and tracks the user's history making use of AWS storage and processing.

The end product is very user friendly and fits a completely exclusive niche in the world of sports performance tracking. It can help new and experienced users alike maintain, keep track of, and improve their lacrosse game. Because the full package is rather portable, with only a small box and minimal stick unit, it can be taken and used almost everywhere. We also plan to have rechargeable batteries to add to this aspect, allowing players to take the system for use outside. Finally, with both an instantly updating built-in LCD screen as well as more detailed metrics on an app, we hope to give essential feedback quickly and clearly to assist in a training session.

Visual Aid

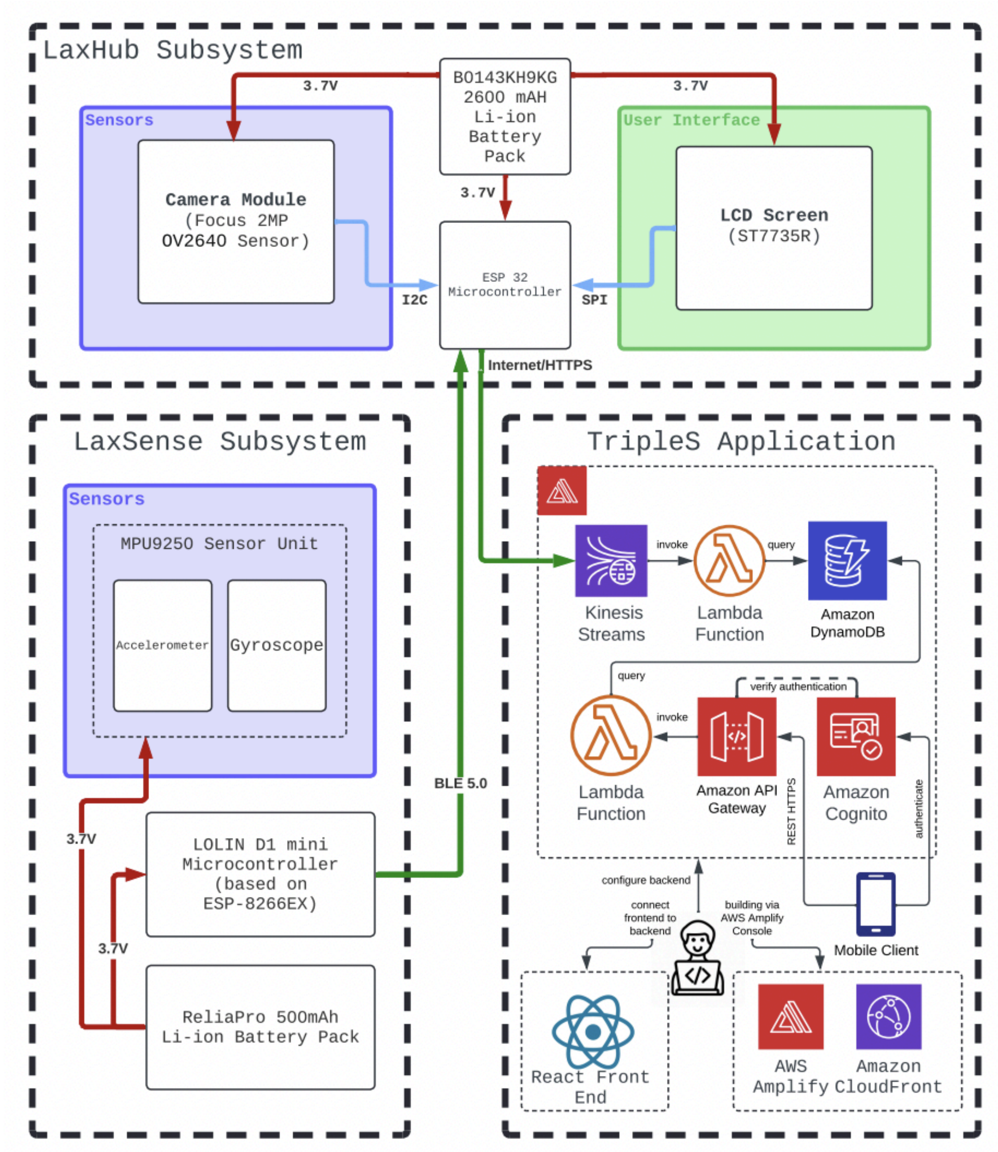


High-level requirements list

1. Our first requirement is **real-time performance tracking and analysis**, more specifically to provide feedback and improvements or failures of a swing within 30 seconds. The system must accurately measure and analyze metrics in near real-time, including shot speed, accuracy, and stick form. This delay also includes sending data to the cloud and outputting a more detailed response in the application.
2. Our second requirement is that the **accuracy of ball speed and trajectory** to be ±10 mph and within 10 feet respectively. It's important to measure this accuracy since players need to clearly see if their speeds and trajectories change over time. A user might want to keep track if the speed of his throws become faster over time, while also maintaining a good trajectory.
3. Our third requirement is that the **LaxSense unit must weigh less than 3 ounces**. It's crucial to keep the unit light to avoid interfering with the mechanics of the lacrosse stick. If the LaxSense unit is too heavy, it could negatively affect shot speed and trajectory, which is not ideal. By keeping the unit under 3 ounces, we can ensure it doesn't disrupt the natural balance of the stick, allowing players to maintain optimal shot speed, accuracy, and overall performance.

Design

Block Diagram



Subsystem Overview

LaxHub is the main processing unit of this system and contains the custom PCB, microcontroller, LCD screen, and camera, as well as a bluetooth 5.0 module (as part of the microcontroller). The LaxHub will need to be powered by a rechargeable battery. The main peripheral connection of this subsystem is the low-power bluetooth connection with the LaxSense subsystem. It also connects directly to the backend cloud component of our application via a Kinesis client.

LaxSense is a subsystem that mounts on the back of the lacrosse stick, which will contain a microcontroller, accelerometer, and a gyroscope. These parts will work in conjunction to keep track of performance metrics such as shot speed, stick angle, and form. Because this is a standalone device, this will need to be powered by a small battery system. LaxSense directly sends data to the LaxHub via the built-in bluetooth module of the microcontroller.

The TripleS application is our final subsystem, and handles data display and analysis. It receives data from the ESP 32 microcontroller through a Kinesis Client and places it in our database, which is DynamoDB in our backend. We will use React as our frontend and use AWS Amplify to help build the infrastructure. A mobile user will have access to this application and the data displayed will be from the very DynamoDB table that the microcontroller sends data to.

Subsystem Requirements

The **LaxHub** consists of four main hardware components all connected through the PCB. Our main processing hub is the ESP 32 microcontroller. We specifically will use a ESP32-S3 which integrates a bluetooth 5.0 and wifi module. The microcontroller will handle communication with both other modules and requires this form of connectivity. More specifically, the microcontroller receives data from the LaxSense through low power bluetooth, and sends data to the backend cloud application by connecting to the internet. Sending data to the application will be done by implementing Kinesis client code in the microcontroller. The next component is the battery unit. We chose a B0143KH9KG, 3.7V-2600mAh-9.62Wh Rechargeable Li-ion Battery Pack in order to have plenty of reserve battery and be able to charge it up when possible. The battery connects to the microcontroller, camera, and LCD screen, and needs to be able to supply a constant 500mA to all. The ESP32 microcontroller cannot function below 500mA. The camera module we will use is a 2MP OV2640 Sensor. It is a small camera option that is still able to capture 1080p at 30 ± 12 fps video for short frame by frame analysis. The camera module is connected as an I2C device to the microcontroller and we will install the appropriate driver and devicetree to read and use its data. We will need a connection between these two components able to send 30 frames per second of data to the microcontroller. Similarly we have an LCD screen that will receive basic data to display from the microcontroller. We chose the ST7735R screen especially to have a SPI interface as opposed to I2C for faster data transfer and being able to display more critical information as quickly as possible. Overall this is the most essential subsystem and as the hub of communication for all processes, it must use its connections with high rates of data movement to satisfy requirements of providing feedback as quickly as possible.

The **LaxSense** subsystem is a critical component mounted on the bottom of the lacrosse stick, responsible for gathering and transmitting performance metrics such as shot speed, stick angle, and form. It utilizes the MPU-9250 sensor unit, which integrates a 3-axis accelerometer, 3-axis gyroscope, and a magnetometer, working together to capture real-time data about the stick's movement. The accelerometer measures the stick's linear acceleration with configurable ranges of ±2g, ±4g, ±8g, and ±16g, ensuring precise tracking of shot velocity and force. Meanwhile, the gyroscope measures angular velocity with full-scale ranges of ±250, ±500, ±1000, and ±2000°/s, allowing for detailed analysis of stick rotation and form. The subsystem is powered by a 500mAh Li-ion battery, which must provide a continuous 3.7 ± 0.75 V output and support up to 5 hours of operation per charge. The MPU-9250 operates with low power consumption, so the connection from the battery to the sensor system must be a stable 3.5 ± 0.75 mA when all axes are enabled. The LOLIN D1 Mini microcontroller, based on the ESP-8266EX, processes the data and transmits it to the LaxHub via

Bluetooth Low Energy (BLE). The microcontroller connects to the sensor system via I2C and so we must properly install the driver and devicetree for these sensors. The BLE connection to the LaxHub must maintain a stable data rate of at least 1 Mbps with a range of up to 10 meters, to allow for real-time performance feedback without noticeable delays. The total weight of the subsystem must remain under 3 ounces to avoid affecting the lacrosse stick's balance. Any failure in sensor accuracy, Bluetooth range, or power efficiency would cause the subsystem to fail, leading to lost data.

The frontend of the **TripleS application** is created using React, and we will use AWS Amplify to develop and deploy the mobile application. The TripleS application contains a singular DynamoDB database. Once the ESP 32 microcontroller has the necessary data, then this data is sent to Amazon Kinesis streams through a Kinesis Client. This is done by first setting up the AWS software development kit on the microcontroller, initializing the client, creating a data stream, and then sending the data in a JSON format. All of this is done via the Internet. The stream then sends the data to DynamoDB using a Lambda function which processes it. Users will go through Amazon Cognito to authenticate, there will be REST HTTPS communication for the API Gateway, which in turn will verify the authentication. This gateway then invokes a Lambda function which queries and reads the data from the DynamoDB table. One of our high-level requirements is that we aim for real-time performance tracking and analysis. To achieve this, we are using Amazon Kinesis streams which are excellent for real-time data streaming and we are using serverless architecture, such as Lambda and DynamoDB. Because of our choice of architecture, we should be able to get analysis and data under 30 seconds. The list of requirements such that if any of them are removed the subsystem would fail, are as follows: Since Kinesis has shard limits, each shard can support up to 1000 messages per second or 1MB per second, so if these thresholds are broken, it may lead to throttling and in turn, failures. Sending the data from Streams to Lambda is 2 MB per second (shared) per shard for all consumers, so if these thresholds are broken, it may lead to throttling and in turn, failures. We also have to ensure that the Lambda functions to send the data to DynamoDB and retrieving the data from the database both follow the same format, or else there will be errors in consistency regarding data.

Tolerance Analysis

One potential risk area was power supply and battery life. Since this is meant to be a portable system that can be used outside, we need it to last sufficiently long as long as it has consistent power. We chose the Li-ion Battery Packs (2600 mAh for LaxHub and 500 mAh for LaxSense) since they are rechargeable, but ultimately low battery life could lead to system failure or frequent recharging, which detracts from the experience we want to give. Here we estimate the battery life of each unit to ensure this is not the case:

LaxHub Subsystem:

1. ESP32 Microcontroller: Typically consumes at least 500 mA during active radio transmission, such as Wifi or BLE
2. Camera Module (OV5647): Consumes approximately 140 mA during operation in compressed mode
3. LCD Screen (ST7735R): Consumes around 50 mA when active, including backlight
4.

Total estimated current draw for LaxHub = 500 mA + 140 mA + 50 mA = 690 mA

LaxSense Subsystem:

1. LOLIN D1 Mini: Consumes at least 500 mA
2. MPU9250 Sensor Unit: Consumes approximately 3 mA

Total estimated current draw for LaxSense = 500 mA + 3 mA = 503 mA

2. Battery Life Calculation

1. LaxHub
    a. Battery capacity = 2600 mAh
    b. Battery life = 2600mAh/690mA ≈ 3.77 hours
2. LaxSense
    a. Battery capacity = 500mAh
    b. Battery life = 500mAh/503mA ≈ 0.99 hours

The LaxHub subsystem can operate for approximately 3.77 hours on a full charge, while the LaxSense subsystem can operate for about 0.99 hours. If continuous operation is needed for longer than this, the design may need larger battery capacities or more better power management strategies. The battery life of the LaxSense subsystem is a greater risk because of its shorter operational time. However overall, this analysis shows that while the current design is feasible for short-term use, some add-ons (or just simply a mobile power source like a power bank) may be needed for longer use.

Ethics and Safety

Our lacrosse performance tracking system raises several important ethical and safety considerations that we must carefully address throughout development and deployment. Our system collects and processes personal performance data of players, and in accordance with the IEEE Code of Ethics principle to "respect the privacy of others", we will implement robust data protection measures. This means allowing users full control over their data (including the ability to delete it), only collecting data necessary for the system's core functionality, and clearly communicating our data practices to users. Transparency is another key ethical principle we will adhere to, following the ACM Code of Ethics principle of honesty. We will clearly communicate the capabilities and limitations of our system, provide accurate information about the accuracy of our measurements, and be transparent about any data sharing practices. Next, we have to keep safety in mind as well for our product design. The LaxSense unit attached to the lacrosse stick must not have a chance of causing any physical danger to players. We will ensure the unit is securely attached and cannot come loose during play, use materials that are safe for skin contact and won't shatter if impacted, and design the unit to minimize risk of injury if a player falls on it. Electrical safety is also a concern, as both the LaxHub and LaxSense contain electrical components. We will comply with IEC 60950-1 for IT equipment safety, ensure all batteries are properly enclosed and protected from impact, and use low-voltage components where possible to minimize electrical hazards. In terms of regulatory compliance, we will ensure our product complies with relevant regulations, including FCC regulations for wireless devices, and Consumer Product Safety Commission guidelines for sports equipment.

References

"Biomechanics of the Lacrosse Shot." *Biomechanics of the Lacrosse Shot*, Center Island Performance Athletics, www.centerislandperformanceathletics.com/2016/03/biomechanics-of-lacrosse-shot.html. Accessed 3 Oct. 2024.

Esp32 Series, Espressif, www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Accessed 3 Oct. 2024.

"Getting Started with ESP32 CAM: Streaming Video Using ESP Cam over WIFI: ESP32 Security Camera Project." Instructables, Instructables, 22 Feb. 2022, www.instructables.com/Getting-Started-With-ESP32-CAM-Streaming-Video-Usi/.

"Lacrosse Passing Guide: How to Throw a Lacrosse Ball." LacrosseMonkey.Com, Lacrosse Monkey, 15 July 2024, www.lacrossemonkey.com/learn/how-to-pass-lacrosse-ball.

Lacrosse Shooting: A Guide to Increased Shot Speed and a Quicker Release, A-Game Sports, 13 Aug. 2018, agamesports.net/2017/09/04/lacrosse-shooting-guide-increased-shot-speed-quicker-release/.

"Lolin D1 Mini." LOLIN D1 Mini - WEMOS Documentation, www.wemos.cc/en/latest/d1/d1_mini.html. Accessed 3 Oct. 2024.

"MPU-9250." TDK InvenSense, 3 Nov. 2023, invensense.tdk.com/products/motion-tracking/9-axis/mpu-9250/.

Rabil, Paul. Lacrosse Passing Guide: How to Throw a Lacrosse Ball, YouTube, 29 Nov. 2017, www.youtube.com/watch?v=TnsMrcQ0H-M.

ST7735R, Adafruit, cdn-shop.adafruit.com/datasheets/ST7735R_V0.2.pdf. Accessed 3 Oct. 2024.