

Project Proposal

Player Tracking Camera

Team 21

TA: Feiyu Zhang

Shivang Charan

Aksh Gupta

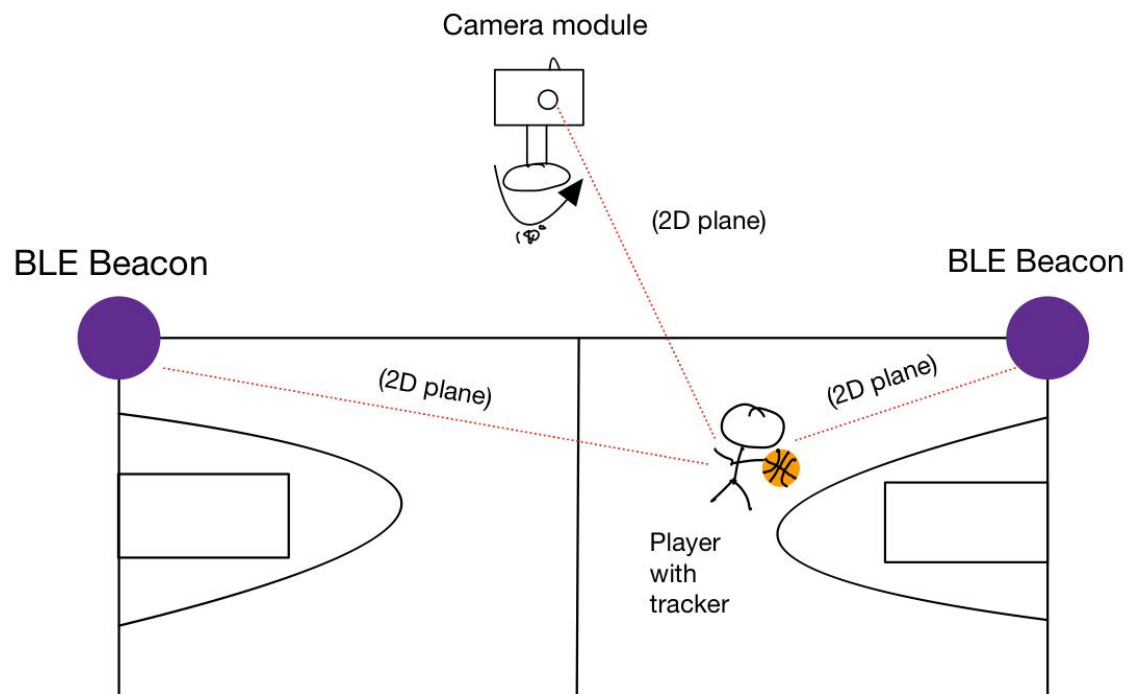
Oreoluwa Sunmola

Introduction

Problem - Watching and rewatching amazing highlights by athletes has become quite popular across the internet. These viral highlights have primarily been reserved for organized sports teams with dedicated film teams - as regular people generally do not have a cameraman to film the pickup games they play.

Solution Overview - In order to solve this problem, we intend on creating a moving stand that will keep the intended target in the view of their camera. This stand will be adjustable to fit most existing cameras. By setting up small beacons around the playing area we can calculate the position of the target person running - which would then be used to rotate the camera to capture the subjects' movements in real-time. Functionally, this would eliminate the need to have a cameraman follow the game's movements and would automate the filming process.

Visual Aid:



High-level requirements list:

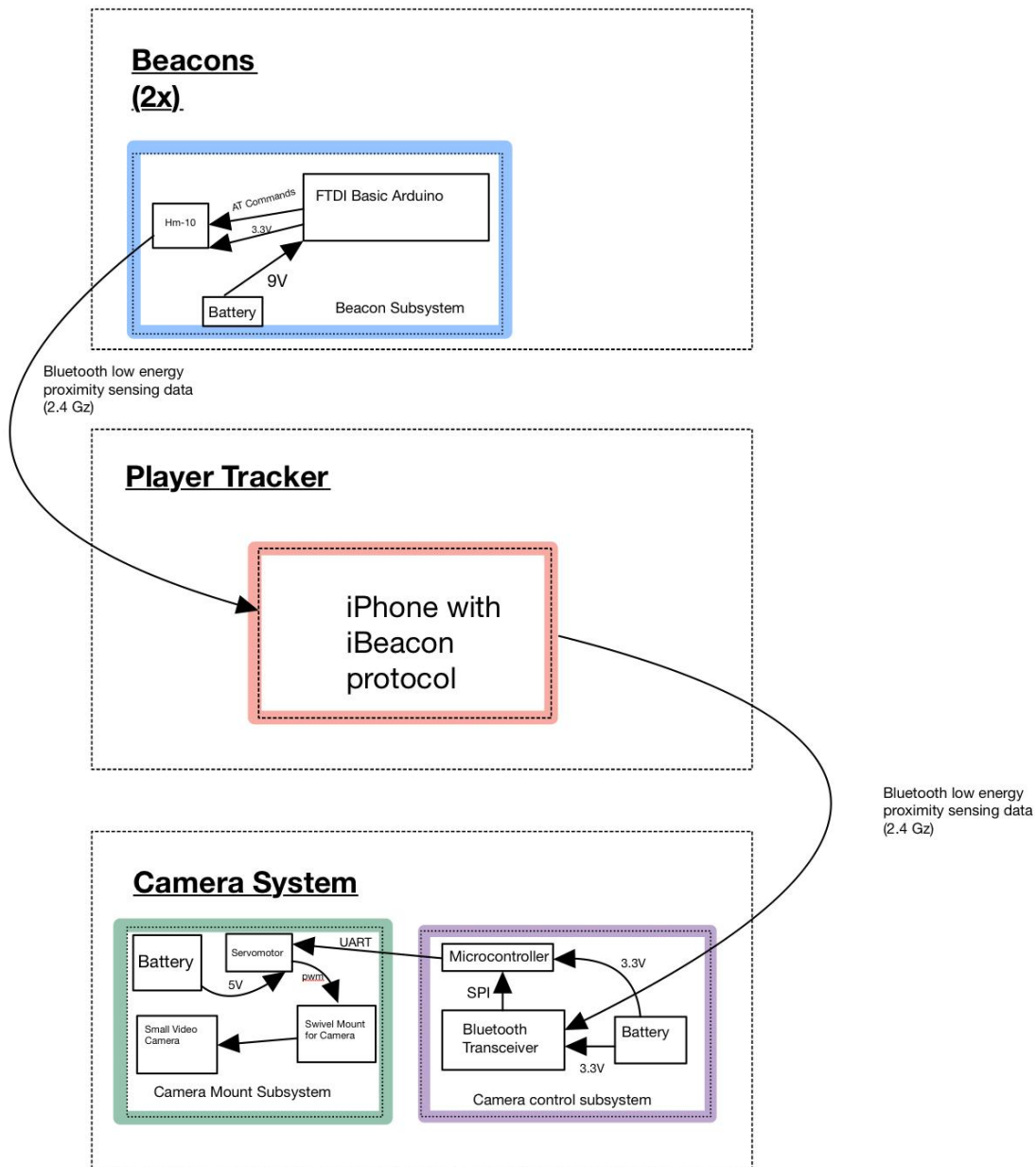
BLE beacons should be able to create a unique 31-byte data packet and have it constantly transmitting, so that the player tracker is able to pick up the data packet.

The location is determined using the iBeacon protocol between the BLE Beacons and the target player's iPhone to determine an approximate position within a 10 meters radius.

The camera system should be able to keep the user within the camera's view for 90 percent of the recording duration and have the person within the frame but not necessarily centered.

Design

Block Diagram



Subsystem Overview:

Beacons: There will be 2 BLE beacons that will be responsible for sending a packet of 31 bytes of data that can be used in conjunction with the player tracker to determine the relative 2D position of the user. Our BLE beacons are cheaper, faster, and more power consumption efficient than traditional Bluetooth beacons. Beacons send a small amount of data that can be translated in order to calculate position from the receiver. This gives a strong way to get an approximate location of the user so that we can see if that position has left the camera's view or not.

Player Tracker: In order to use the iBeacon protocol we are going to use an iPhone so that we can actively track the approximate current location of the client. Since this device is actively being worn by the user it has to be safe and not cause the user danger so we decided to use an iPhone. Our product would be mostly used for physical activity; therefore, it was important to use an iPhone which is already used as a device while doing physical activity which proves it is safe for the wear. This iPhone will be responsible for transmitting coordinates of the person to the camera subsystem, so that the servo can keep the person in the view of the camera.

Camera Mount Subsystem: This subsystem is responsible for securely holding the camera and then based on the feedback received the Camera Control Subsystem will have the servos turn the camera smoothly to the position that will keep the user in the view of the mounted camera.

Camera Control Subsystem: The Bluetooth receiver gets coordinates from the player tracker. By using logical computation in the microcontroller it is able to determine the turn angle of the servo. The servo then adjusts the position of the camera to keep the user within its line of sight.

Subsystem Requirements:

Beacons: The beacon will consist of a FTDI Basic Arduino, HM-10m and 3.3V lithium coin battery that supplies 225mAh. BLE is better than regular Bluetooth because of its low energy usage; therefore, these beacons can be active for ~3 years on a single 3.3V coin battery. It is important to mention the difference between BLE and iBeacon. The iBeacon is just a protocol that is used to standardize the data sent from the BLE beacons. This protocol allows us to use iPhones in order to collect the data and keep track of the user's position. BLE chips are also 60 - 80 percent cheaper than regular Bluetooth chips which can make the cost to produce lower and more desirable to consumers. The range of a BLE is 70 meters so there will be full coverage of the player's movements over the course of a game.

Data transmitted: These beacons transmit a packet of 31 bytes of a unique code sent. Specifically for iBeacon format is ("iBeacon prefix (9 bytes), Proximity UUID (16 bytes), Major (2 bytes), Minor (2 bytes), TX Power (1 byte)"). The most important part from all of this is the TX power because this is used to calculate the distance from the receiver to the iBeacon. The typical speed for BLE is ~3ms which is much faster than normal Bluetooth which is ~100ms. To send the 31 byte data the FTDI Basic Arduino will be working in union with the HM-10 by programming it with AT commands. The AT commands are most commonly used in modems. For this subsystem however, they are important in order to tell the HM-10 chip when to send data and what specific data to send.

Requirement 1: BLE Beacon should be able to create a 31 byte unique code and have to constantly send so that the player tracker can pick it up.

Requirement 2: The Arduino and HM-10 should be able to properly communicate with AT commands in order to send the data packet.

Power Consumption requirements: There are two major components that will be drawing power inside the beacon:

HM-10 Bluetooth 4.0 BLE UART module. This module transmits Bluetooth data at 2.4GHz ISM band. This will be an arduino plug-in, it will be tapping into the 3.3V, 50mA voltage supply on the arduino board . It has two working modes (sleeping state and active state). During active state, it has a power rating of 8.5mA while at sleep state, it ramps down to 50~200uA.

Arduino UNO: The board has a recommended input voltage of 7-12V. We will be using commercial 9V batteries as the power supply.

Player Tracker: In order to monitor the players' constantly changing position we will be using an iPhone that can be easily worn on anyone's arm or kept in their pocket, for example. An iPhone is already safe to be used while running around and produces the least amount of risk to the user when they use our product.

Input Data: BLE Beacons send out data about every 10s or so. The iPhone constantly has an app running that will be picking up the 31 byte data packet that is sent from the BLE Beacons using Apple's iBeacon protocol. The requirement is for the iPhone to be able to actively pick up the data packets from our BLE Beacons and convert them into 2d coordinates and then send them to the Bluetooth receiver connected to the microcontroller on the camera control subsystem.

Requirement 1: The app on the phone should be able to gather data from multiple BLE Beacons and calculate the cartesian coordinates and transmit them to the Camera Control Subsystem.

Requirement 2: Phone must be constantly listening to the BLE Beacons and constantly sending the changing coordinates to the Camera Control Subsystem.

Camera Mount Subsystem: This will be a pivoted support that permits rotation in the x-axis (from left to right). This camera mount system will consist of the following components:

Servo Motor: This will be rotating the phone and its holder. The motor will be located at the neck (middle) of the whole camera system. The motor is connected to the microcontroller to determine when to activate the servo, hereby turning the camera phone and in which direction the motor should spin. It will have its own dedicated power supply through a battery, whose size will depend on the servo motor we chose later on in the project.

Requirement 1: Motor should be able to support weight of phone and its holder

Requirement 2: It should have the capability to spin forward, turning the phone to the right and backward, turning the phone to the left within 180 degrees

Requirement 3: It should be able to make a full 180 degree rotation within a second

Requirement 4: During operation, temperature must remain within safety limits of below 120 degree celsius.

Mount/Stand: This will be built using a selfie stick phone holder in order to keep the phone held securely and have the ability to fit most small cameras. In order to attach this to our servo motor we will have a middle level component that would be 3d printed in order to properly connect the motor to the phone holder, which would be secured to an adjustable tripod base to reach the height of play the camera will record.

Requirement 1: Have a mount which can fit a camera properly and securely.

Camera Control Subsystem: This subsystem is responsible for receiving coordinates of the user from the player tracker and then needs to calculate if the camera mount

should spin or stay stationary. If its stationary means that the person is in the view of the camera mounted.

Microcontroller: will gather all the data from the BLE beacon, and the player tracker and then calculate the degree of turn needed for the servo motor to position the phone camera to capture the person.

Requirement 1: It must be able to perform computation required to determine the x and y coordinates of the user wearing the player tracker within 1~3ms.

Requirement 2: It must be programmable using popular programming languages.

Bluetooth Receiver: This Bluetooth receiver device will be different from the HM-10 chips because HM-10 can only send 31 bytes of data whereas from the player tracker we want more bytes of data to be transferred. This is also important because in the future if we want to send more specific data this will let us do that. Bluetooth is slower than BLE; however, the transmit rate for Bluetooth is ~100ms which is still faster than a person running outside of the window view.

Requirement1: Receive coordinates from the player tracker within 100ms and then be able to transfer data to the microcontroller.

Power Supply: All of the components in the camera control subsystem will be embedded on a PCB and will be powered by a 3.3V lithium coin battery.

Requirement 1: The battery must be small enough in order to downsize the module at the midsection to keep the overall structure of the whole stand structurally stable.

Requirement 2: It must be easily replaceable

Tolerance Analysis:

An aspect of our design that poses a significant risk to the successful implementation of our project is the speed of the camera mount. More specifically, the issue of making sure that the camera is oriented towards the player over the course of a fast paced game is one which presents us with a unique challenge. If the camera does not move fast enough it will make the endeavor ultimately fruitless as the player will not be in frame, as desired. There are two different sub challenges that we are presented with when trying to address this. First there is the issue of input delay. As you may imagine, the player location data takes time to be sent and processed by the microcontroller. The second problem associated with this is that the camera will have to move fast enough to ensure that given the current location data, it is able to swivel at an adequate rate such that the player is actually captured, and hasn't already moved to a completely different part of the playing area.

Through our, quite literal, field work we have developed a strategy to mitigate this issue. First, we intend on operating the camera with a wide angle display. This allows us to optimize the playing surface coverage per frame. Furthermore, BLE technology operates even faster than the already fast Bluetooth technology. By taking roughly 103ms to send and receive data (3ms for the BLE beacons and 100ms for the Bluetooth receiver) we functionally eliminate the issue of input lag as the system will operate at a much faster rate than the player. What this means is that no matter how fast the athlete is moving, our design will operate faster and have a large enough buffer such that they are always within the recording.

Ethics and Safety:

The most relevant ethical concern is related to ACM code of ethics standard 1.6 - about respecting data privacy. This issue is pertinent to people's personal data being secure and being obtained and used responsibly. In our project we do collect tracking data but we avoid any issues related to data privacy. We do this in two ways. First, we do not store the data for long periods of time. The data input is used instantly and cleared as new data comes in. There is not any sort of memory of location that can be accessed by those with bad intentions. Additionally, if the instantaneous data were to be captured, the data being collected does not disclose any new information as the person turns on the camera and is standing right in front of it so there is no issue of location data being revealed.

As a result of physical devices being used, there is a possibility of them interfering with human movement. According to the IEEE Code of Ethics section I subsection 1. we must "hold paramount the safety, health, and welfare of the public". We can adapt the usage of our project to not interfere with the game or the players in a way that causes harm. We can suggest to the user to place the BLE beacons behind the hoop/net/playing surface so that they don't come in contact with the players. We also have the camera on a mount so that it can be placed a safe and reasonable distance away from the players such that it does not interfere with the game and cause anyone any bodily harm. Given the extensive range of BLE and Bluetooth users are able to operate the technology without risk of interference and potential injury through collision.

Citations:

“Bluetooth Low Energy (BLE) Beacon Technology Made Simple: A Complete Guide to Bluetooth Beacons.” *Beaconstac RSS*,
blog.beaconstac.com/2018/08/ble-made-simple-a-complete-guide-to-ble-bluetooth-beacons/.

“Determining the Proximity to an iBeacon Device.” *Apple Developer Documentation*,
developer.apple.com/documentation/corelocation/determining_the_proximity_to_an_ibeacon_device.

“How to USE HM-10 Ble Module with Arduino to Control an LED Using Android App.” *Circuit Digest*,
4 July 2019,
circuitdigest.com/microcontroller-projects/how-to-use-arduino-and-hm-10-ble-module-to-control-led-with-android-app.

“IEEE Code of Ethics.” *IEEE*, www.ieee.org/about/corporate/governance/p7-8.html.

“The Code Affirms an Obligation of Computing Professionals to Use Their Skills for the Benefit of Society.” *Code of Ethics*, www.acm.org/code-of-ethics.

“What Is iBeacon? All You Need to Know.” *Bleesk*, bleesk.com/ibeacon.html.

“What Is iBeacon? A Guide to iBeacons.” *iBeacon.com Insider*, 13 June 2014,
www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/.