

IoT Platform for Distance Contest During COVID-19 Pandemic

Tsu-En Hsueh
Department of Computer Science and
Information Engineering,
National Dong Hwa University
Hualien, Taiwan
410821305@gms.ndhu.edu.tw

Min-Xiou Chen
Department of Computer Science and
Information Engineering,
National Dong Hwa University
Hualien, Taiwan
mxchen@gms.ndhu.edu.tw

Wei-Che Chien
Department of Computer Science and
Information Engineering,
National Dong Hwa University
Hualien, Taiwan
wcc@gms.ndhu.edu.tw

Abstract—Under the influence of the epidemic, many meetings or works have been changed to remote, and lots of competitions have been postponed or even suspended. Nowadays, few platforms replace physical competition in distance. Therefore, we provide and implement a platform for tug-of-war using strandbeest, enabling contestants from two places apart to compete at the same or different times. We create the strandbeest tug-of-war platform via the development board, database system, and backend logic. The uniqueness of the mac address of the development board and the current time of the round filter the strandbeests' tension data and take the arithmetic average. After both contestants finish the round, the system compares the score of each contestant. The arithmetic average of the filtered tension data from the strandbeest is made to decide who the winner is. The maximum number of tension data is shown to be read and sent to the database at different times and the change before and after the round. Even if the pandemic slows down in the future, people can compete with this platform, especially contestants from different cities or countries, since it beats the constraint of distance.

Keywords—Strandbeest, tug-of-war, Vertically Integrated Project (VIP), Internet of Things (IoT), Arduino

I. INTRODUCTION

There are currently few platforms that replace physical competitions in distance. The postponement or even suspension of the competition discourages contestants. Thus, we provide and implement a strandbeest tug-of-war platform so that contestants can compete in different places at different times or at the same time. If the competition cannot be held due to the pandemic or other reasons, this strandbeest tug-of-war platform is the best choice for everyone.

With a development board, Arduino, it is possible to upload data to the database on the local side using a third-party kit. By reforming the pressure sensor into a tension sensor, the tension value of the strandbeest can be read as fast as possible. By having the techniques above, the system allows the contestants to read the tension value for the competition in sequence and in the same place, but not at the same time. About the backend, we use an application programming interface (API) to receive the data to calculate the final score of the contestant's strandbeest.

Therefore, we focus on how to make the development board smoothly connect to the Wi-Fi to achieve remote data transmission and database design so that contestants in different places can compete simultaneously.

II. SYSTEM ARCHITECTURE

We separate the system architecture into four parts: network, devices, storage, and web service (Fig. 1). The development board contains the Wi-Fi information, including

the Wi-Fi account and the Wi-Fi password by the smartphone or any other devices. The development board is connected to the Wi-Fi turned on by the smartphone.

With the Wi-Fi connection, the development board receives the tension value from the tension sensor, packages the value and the mac address of the development board into a JavaScript Object Notation (JSON) file, and transmits the data to the database. By manipulating the website by contestants, the system recognizes which development board is using. The current time and the mac address of the development board is used to filter the data after the round ends.

For the devices, we need a tension sensor and a Hx711 module to read the tension value of the strandbeest. We also need a development board to connect to the Wi-Fi for sending the data to the database. For the network, it is needed to establish a Wi-Fi connection between the development board and the access point. The Wi-Fi account and the password can be fixed. After the connection, the data is sent to the database. For web development, an Application Programming Interface (API) is created to fetch the data from the development board. A login system and an admin page are required to create contest rooms in batches and see the status of each contest room. After the round ends, the system filters the data and calculates the score of the contestant.

The number of tension values per round plays an important role in the competition. The higher the number of tension values, the more accurate the contestant's performance. Therefore, since the tension data are discrete, not continuous, we need to read the tension value as much as possible to obtain the data that is closest to the actual situation. Table I shows the number of tension values in different seconds. The frequency of uploading data to the database by the development board per second has been adjusted to the maximum value.

Figure 2 shows the room management page after the admin logs in. Here, the admin can add contest rooms and delete all contest rooms. When the competition is over, the page refreshes to watch the contest status in each contest room.

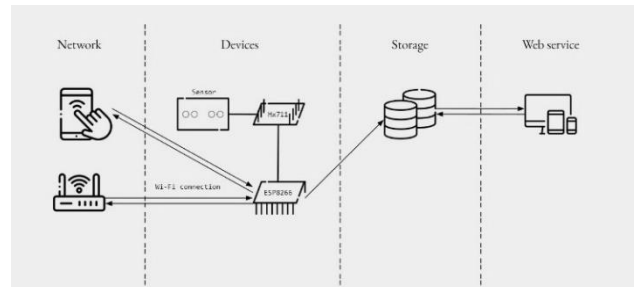


Fig. 1. System architecture of the platform.

ID	Room ID	Room Password	p1	p2	p1Trigger	p2Trigger
1	417805002	056KAYH061	Tyler2023	JacksPCA	Finished	Finished
2	433124856	7upRuc276			Empty	Empty
3	408713524	K3R75ouduH4			Empty	Empty
4	9680372154	K5Gou2u56			Empty	Empty
5	101140096	08p26u717C1			Empty	Empty
6	2543802876	7H4L0M5W6CQ			Empty	Empty
7	3046819271	uEVP4u41Q6			Empty	Empty
8	3285127545	1R0CY4U7U06			Empty	Empty
9	365340827	4R0p4R02u5			Empty	Empty
10	0103575246	PrvYH6G2u5			Empty	Empty

Fig. 2. Admin page of the platform.

TABLE I. TENSION VALUE READ IN DIFFERENT SECONDS

Time (sec)	Number of Tension Values
3	6
10	20
30	61
60	116

III. SYSTEM IMPLEMENTATION

Distance contests are for competition under epidemic. Before building the strandbeest tug-of-war distance contest system, it must be figured out what the competition requires and what devices and technologies. In the planned schedule, cooperators from the Department of Education and Human Potentials Development teach the students who participate in the competition to make their strandbeest. After the workshop, the students in this competition compete with their new strandbeest and race for the win using the platform.

A. Hardware Equipment

We need a tension sensor to detect the tension value from the strandbeest made by students. We reform the pressure sensor with an Hx711 module for reading the tension value of the strandbeest. The pressure sensor reads the force from top to bottom. Thus, to reform the pressure sensor into a tension sensor, we turn it 90° counterclockwise and place it in a horizontal direction. By applying a force from left to right, the pressure sensor detects a value to reform the pressure sensor into a tension sensor. Moreover, a development board is also needed. We choose NodeMCU ESP8266 v3 as the development board since it is cheap and includes a Wi-Fi module that allows sending data to the database. The contestants are given a device box with the tension sensor, Hx711 module, and a development board in it. A string also is extended from the device box to tie up the strandbeest.

B. Backend Framework

Contestants log in through the website and compete on hardware devices. Therefore, we choose a backend framework to set up the website and handle operations including login, user authentication, and competition. We choose Flask as the backend framework of this study since its strong extensibility. Being a micro framework, the loading of the website is fast to fetch more data from the development board.

C. Wi-Fi Connection

The development board connects to the Wi-Fi before sending data to the database. The development board completes the auto connection to the Wi-Fi. The way to make the development board connect to the Wi-Fi is opening the Access Point (AP) of the development board. By doing this, the admin of the competition or the contestants connects to the development board via 192.168.4.1 and provides the Wi-Fi account and password to it. After resetting the development board by pressing the button on the development board, it

opens the Station (STA) mode and automatically connects to the Wi-Fi so that the admin of the competition or the contestant opens and closes the AP mode. If the Wi-Fi cannot be connected, after 50 tries, the development board will close the STA mode and open the AP mode again to wait for the next person to type in the correct Wi-Fi account and password.

D. Tension Value Sending

After the development board is connected to the Wi-Fi, the AP mode of the development board closes and the STA mode of the development board opens. The tension value and the mac address of the development board are packed into a JavaScript Object Notation (JSON) file and sent to the database using the HTTP POST method (Fig. 3).

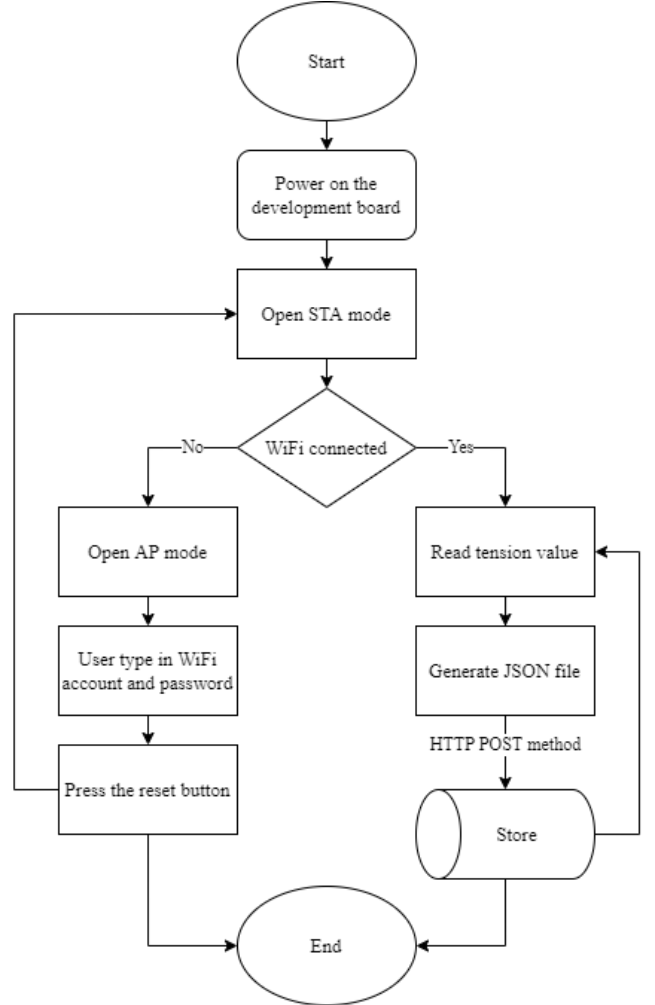


Fig. 3. Wi-Fi connection and value sending.

E. Database Design

Contestants log in, view personal information including username, email, and the number of wins and losses, and enter the room number, room password, and the board ID of the development board before entering the competition room. On the other hand, in addition to the above operations, the admin also enters the competition room operation page. On this page, the admin can add or delete the room in the competition and watch the status of all room venues (contestants have not entered, contestants have entered, contestants in competition and contestants finished the current round). There are several tables in the database, including “boardvalue”, “competition”, “macaddress”, “users”, and the tables that the name is the character “t” followed by the string of the competition room.

The column names in the table “boardvalue” are “value”, “time”, and “macAddress”. Values in the column “value” are the tension value from the tension sensor sent by the development board, the “time” column stores when the data is stored in the database, and the column “macAddress” stores the mac address of the development board.

The column names in the table “competition” are “competitionName”, “competitionPass”, “player1”, “player2”, “p1trigger”, and “p2trigger”. The column “competitionName” and “competitionPass” stores the room number and the password of the room separately. The column “player1” and “player2” stores the username if the contestant entered the room, empty if the contestant is not in the room. The column “p1trigger” and “p2trigger” stores the states of the contestants, 0 means that there is no player, 1 means the contestant is in the room, 2 means that the contestant is playing, and 3 means that the contestant finished his or her round.

The column names in the table “macaddress” are “boardID” and “macaddress”. The column “boardID” is the index of the development board. The reason is to number the development board, since this system is a distance contest system. There might be several development boards connecting to the Wi-Fi and send the data to the database simultaneously; the column “macaddress” is the macaddress of the development board.

The column names in the table “users” are “username”, “password”, “email”, “win”, and “loss”, which store the username, encrypted password, email, number of wins, and losses respectively.

The last table is the character “t” followed by the string of the room number. When the admin creates a room, the backend generates the room number and room password for the room, this information is given to the contestants privately for them to enter. The room number is randomly generated with a length of 10 and 12 for the password. However, the room password contains uppercase letters, lowercase letters, and numbers. When the room number is generated, the system checks if there exists a room that has the same room number. Moreover, the room number regenerates if there exists a room number that is the same as the generated room number. The column names in the table are “player” and “score”. The column “player” stores the username of the contestant, and the column “score” stores the arithmetic average of the tension value.

F. Backend Logic

Before the competition starts, the admin logs in and creates the rooms. When the admin clicks the button on the website, the backend generates room numbers and passwords, checks if there are the same room numbers, regenerates if there already exist, and then writes all the information into the table “competition” in the database.

During the competition, the admin or the contestants set up the development board and connect to the Wi-Fi. On top of that, the string extended from the device box is tied up correctly. Moreover, the contestants log in and type in the room number and password that the admin gives them privately and the board ID on the device box to enter the room. Since the table “macaddress” in the database stores the board ID on the device box and the corresponding development board mac address, it is known which development board is in use. After the round starts, the start time of the current round is remembered by the session in Flask.

After the current round ends, the system finds the filtered data in the table “boardvalue”, and the data are filtered according to the mac address of the development board corresponding to the board ID. The contestants log in before the current round starts and the time between the round starts and ends. The filtered data is added and divided by the number of the data, which is the arithmetic average of the filtered data. The arithmetic average and the contestant’s username are stored in the table of the database. After that, all of the old data in the table “boardvalue” are deleted. When there are two data in the table, the round is over. The system compares the value and determines the winner. The contestant who has the bigger value wins. Therefore, the column “win” in the row of the contestant’s username corresponding to the table “users” is added 1. On the other hand, the contestant who has the smaller value losses. Thus, the column “loss” in the row of the contestant’s username corresponding to the table “users” adds 1. By playing rounds, the competition is over until deciding the final winner (Fig. 4).

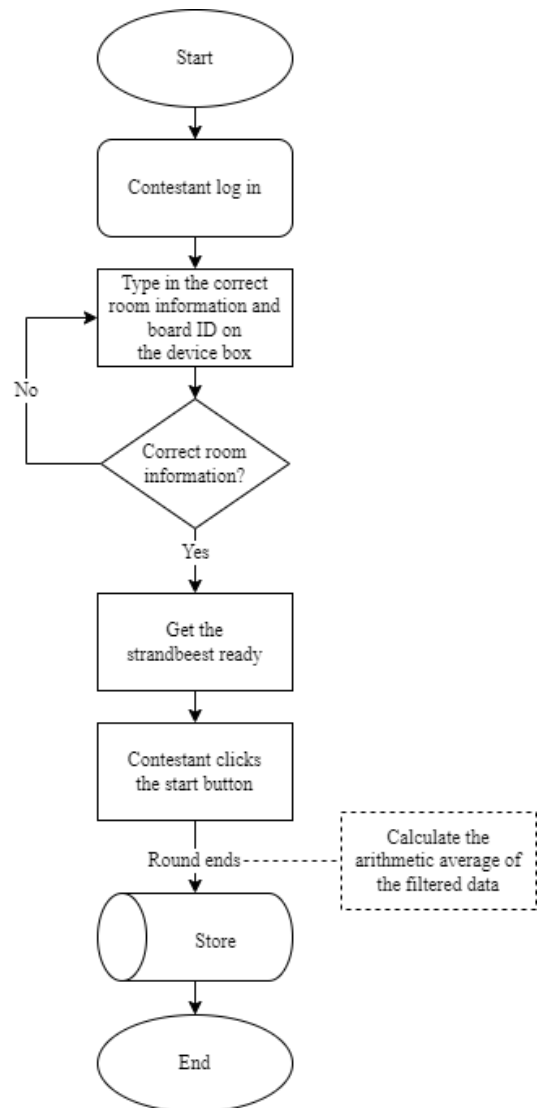


Fig. 4. Backend logic per round.

IV. CONCLUSION

A method of a tug-of-war distance contest is proposed by using strandbeests. Due to the uniqueness of the mac address of the development board and the use of time as an indicator, the system locks the value of the tension force generated by

the development board, or the strandbeest of the contestant, to implement this distance contest of the strandbeest successfully.

The contestants need to start the competition by tying their strandbeest on the device box and then start the whole process mentioned above. There are three devices in the device box, a tension sensor, a Hx711 module, and a ESP8266 v3 development board (Fig. 5). This is a Vertically Integrated Project (VIP) that combines two specialties. With the cooperation of the Department of Education and Human Potentials Development, the students are taught in the elementary school in Hualien to make their strandbeest and build a competition platform for those who participated in the competition to race for the championship.

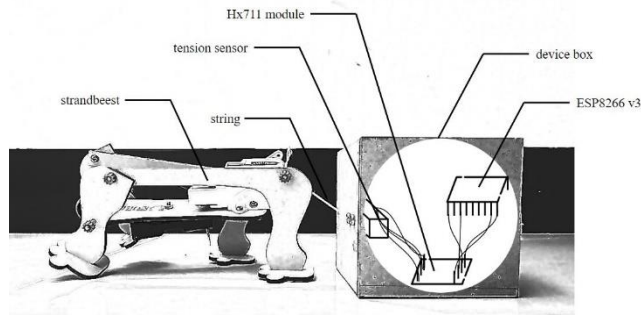


Fig. 5. Hardware illustration.

About the operation, it is only necessary for the admin to create a competition room in batches and set up the Wi-Fi connection of all development boards before the competition starts. On the other hand, the contestants only need to tie their strandbeest with the string extending from the device box properly, and then log in, enter the room number, the room password, and the board ID written on the device box. Although these operations are a bit more than physical competition, we ensure the smooth completion of the competition without the influence of other unexpected situations.

Except for all the features, we add more new features such as a leaderboard. Since we obtain the average tension value of the contestants in a round, the value is stored as the arithmetic average of the tension value. The leaderboard records the results of all previous rounds, which is the tension value and the contestant's username as the basis for ranking and displaying on the website. In addition to determining the results, it also enables healthy competition with other contestants.

Since it is possible to create rooms in batches, the feature is added to create accounts in batches with multiple accounts to create accounts quickly in the progress of the competition. Furthermore, the contestants log in by scanning the QR Code to save time to link to the website. Contestants can spend more time on the competition rather than the processing before the competition. This strandbeest tug-of-war platform is currently a subject of a Vertically Integrated Project in collaboration with the Department of Education and Human Potentials Development. During the whole event, the collaborators from the Department of Education and Human Potentials Development taught students in elementary schools in Hualien and made the students' strandbeest. Students used this strandbeest tug-of-war contest platform for competition in the second half of the event. They concentrate on making their strandbeest and listen to the teaching content of the

collaborators to achieve healthy competition and entertainment in the second half of the event. At the current stage, distance competition becomes the main way of competition due to the pandemic. Compared to holding a physical event, distance contests are safer and more health-friendly. This platform overcomes the distance of space. After the epidemic slows down in the future, people can play tug-of-war with strandbeest in different cities and even different countries. This platform brings people solutions due to the epidemic and a competitive medium that can be widely used in the future.

ACKNOWLEDGMENT

This work was supported in part by the Ministry of Science and Technology of Taiwan, R.O.C., under Contracts MOST 11-2221-E-259 -008 -.

REFERENCES

- [1] Mejias, Sam, et al. "The trouble with STEAM and why we use it anyway." *Science Education* 105.2 (2021): 209-231.
- [2] Salmi, Hannu, et al. "Individual creativity and career choices of pre-teens in the context of a Math-Art learning event." *Open Education Studies* 3.1 (2021): 147-156.
- [3] Itikala, Varshitha. "Arduino Weighing Machine Using Load Cell and HX711 Module." Available at SSRN 3918720 (2021).
- [4] Al Dahoud, Ali, and Mohamed Fezari. "NodeMCU V3 for fast IoT application Development." *Notes* 5 (2018).
- [5] Liang, Mei-Hui, et al. "Greenhouse Environment dynamic Monitoring system based on WIFI." *Ifac-Papersonline* 51.17 (2018): 736-740.