

# Rapid Prototyping Using App Inventor in Information and Computer Sciences Course

Fujio Yamamoto  
Department of Information and Computer Sciences  
Kanagawa Institute of Technology  
1030 Shimo-Ogino, Atsugi, Kanagawa  
Japan

[yamamoto@ic.kanagawa-it.ac.jp](mailto:yamamoto@ic.kanagawa-it.ac.jp)

## Abstract

For approximately four years, we have taught development of mobile applications using App Inventor in small information and computer sciences classes. Our intention is to raise student's ability concerning idea creation, software design and implementation, through rapid prototyping of real world applications. Students created applications by using a large number of basic functions of the official MIT App Inventor and also by using several related extensions. We had several practical applications that would take much time to implement if traditional programming language like JAVA were used. But students were able to build them using App Inventor efficiently without commonly possible bugs. We analyzed the process how they tried to accomplish their work. Applications that we are much interested in are related to getting indoor vertical positional information. Such three applications that we developed are presented here. The first application uses NFC tags, the second one judge the current position by observing the atmospheric pressure, and the third one detects acceleration pattern in an elevator movement, and guesses the current elevator car position.

## 1. Using Embedded Functions and Connectivity

We began to make simple test programs, and gradually made applications of intermediate complexity with using each function provided by App Inventor. Figure 1 illustrates which function was used to each application. Among many embedded functions, students soon became familiar with the acceleration sensors and TinyDB.

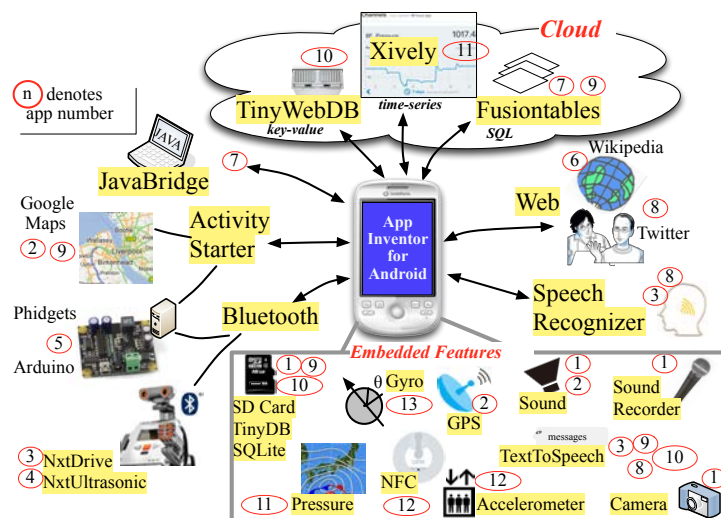


Figure 1 Functions and Connectivity Used in Our Applications

The TextToSpeech was considered very useful to show results of execution by sound. The NFC function adopted in recent version stimulated them to think about appropriate usage. Although pressure sensor was not treated directly in the App Inventor, we could use it by connecting to a small JAVA program. Additionally, the Bluetooth components, which facilitate control of LEGO robots, surprised general students. Several web services and external devices were also available with Activity Starter and Web component.

Among many connectivity functions, connections to cloud databases are most important for us. We take advantage of several types of database including TinyWebDB (key-value), Fusion Tables (SQL), and the xively (time-series). Connections to the xively database can be made by the Web component.

## 2. Acquisition of Vertical Positional Information Indoors

### (1) Using NFC tags

We pasted an NFC tag that indicates the floor number, to the wall near stair entrance of each floor in our university buildings. When touched on it with an Android, a pair of time stamp and the floor number is stored in its TinyDB. Using this application, we could collect many data indicating when each student passed through each floor during an evacuation drill in the buildings. These data are expected to enhance the evacuation plan for future disaster.

### (2) Calculating the Difference of Atmospheric Pressure between Floors

This system, which consists of three Android, guesses the current floor where the specific Android is placed. Outline of the system is illustrated in Figure 2. One Android is placed on the base floor, and it transmits the atmospheric pressure to the xively database periodically. The second Android (the client) is assumed to be in an elevator car. The client judges its current position (namely its current floor) from the difference between its own pressure and the base pressure that is obtained from the xively. This judged floor information is stored to TinyWebDB.

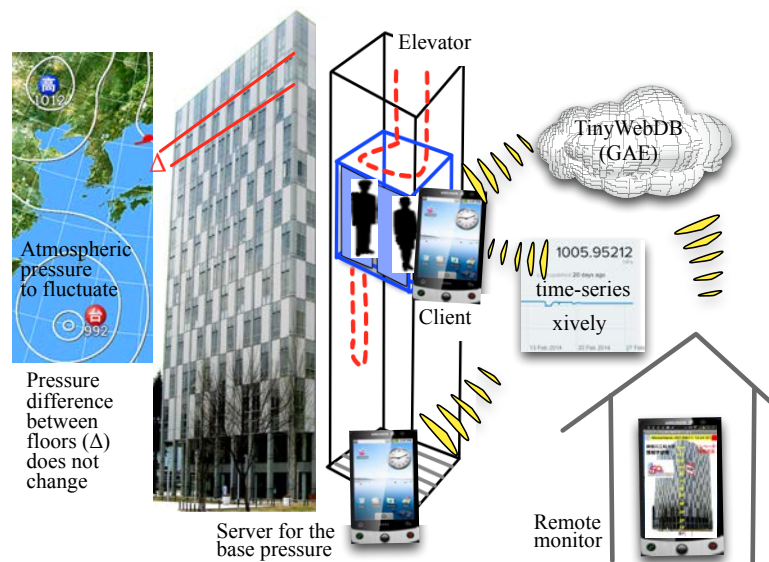


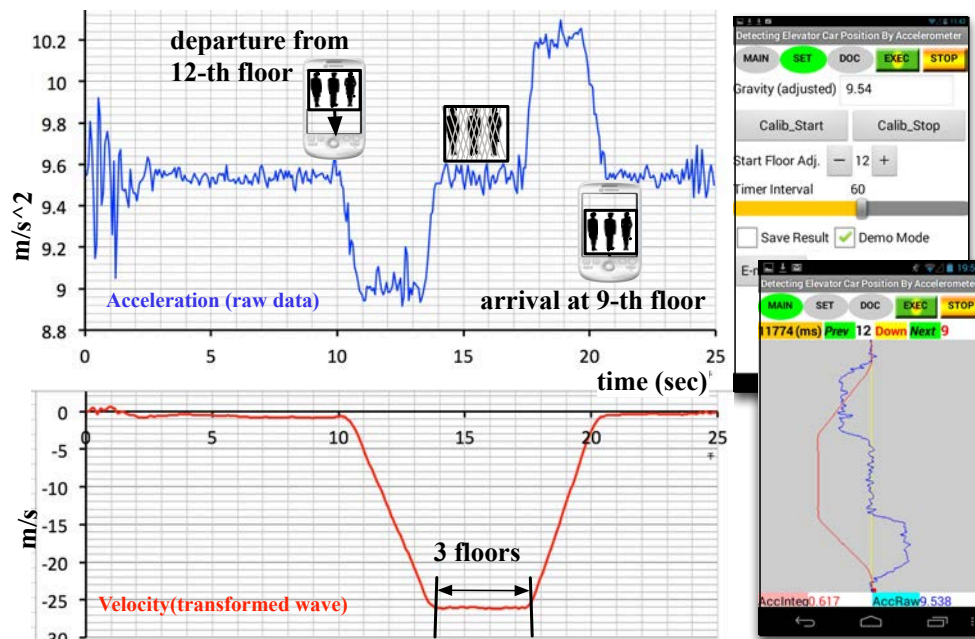
Figure 2 Remotely Monitoring of an Elevator

The third Android, which might be remotely placed somewhere, displays the current position of that elevator car graphically according to the content of TinyWebDB. In such a way, we could create a remote monitoring system for an elevator in very low cost. This

method is also applicable to the case where human takes the Android and goes around indoors.

### (3) Detecting Acceleration Pattern in Elevator Movement

This method evaluates the current position of an elevator car by analyzing the acceleration waveform using accelerometer of an Android. Some waveform examples and screen shots from an execution of our application are illustrated in Figure 3. In this example, the car departed from the 12-th floor and arrived at 9-th floor. The blue line in the upper part denotes the raw acceleration waveform. Immediately after departure, downward acceleration was increased, and soon the acceleration became zero (with only the earth gravity) until approaching the destination floor. After approaching the destination, the acceleration increased again toward the reverse direction. As seen in this example, acceleration wave is generally noisy and inconvenient to our analysis. Therefore, we performed integral calculus (with setting the reference level to the earth gravity) to this acceleration and got velocity waveform, which is indicated with red line in the lower part.



**Figure 3 Detecting and Transforming Acceleration Wave**

Important point here is to prepare in advance the length of the constant velocity section for each traveling length, namely, the number of floors between the departure and the arrival. Under this condition, we can guess the travel length from the observed length of the constant speed section by pattern matching. Arrival floor can be set as the new departure floor, and we can guess the next arrival floor in succession. Using TextToSpeech, this application can automatically issue a voice guide like "you will soon arrive at 9-th floor". This method may be applicable to track a thing moving among different floors by using an elevator. If this method cooperates with the previous one, which uses pressure, accuracy of floor estimation will be improved.