ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 4, Issue 5 , May 2017

Object Tracking based on GPS Technology

Sahar Z. Razzaq, Maysem N.Abd, Enteha G. Muhssan

Computer Department, University of Kufa, Najaf, Iraq

ABSTRACT: There are many objects in various fields such as transportation, industry and health that have to be tracked. In this paper, the current location of anobject is monitored by the monitoring server. System consists of two devices work on environment of android. Object device receives the data of current location from the satellite through the sensor which is included inside the device while the monitoring server device queries the current location of presence of the object by sending a request through the GSM network to the object’s device and after confirming authenticity of the sender, object transmit the current location of a device after encrypting it. The monitoring server program relies on Google map tool to show the current object location on the map.

KEYWORDS: GPS, Tracking system, GSM, Android.

I. INTRODUCTION

GPS is a system which is used worldwide to track and monitor the mobile objects using set of satellites. It has allowed civil authorities since early 80's the civilian location determination using GPS. But it is not universally the only one of its kind, there are several similar systems, such as the Russian Glonass system or the regulations under development and research such as Galileo in Europe and some similar system in china, India and japan. Since development of GPS, a lot of applications have been developed that provide location services using this technique. One of the important fields that use GPS is human tracking. The latitude and longitude data is received by GPS receiver of tracked device. This data is then transmitted to the monitoring device in order to keep localization the device. In this paper the monitoring servers can track their ld using build in GPS receiver signals which will be sent using SMS over GSM.

The Global Positioning System (GPS) is a satellite-based navigation system developed and operated by the US Department of Defense. GPS is a position, velocity and time determination system which is truly global, is able to operate 24 hours a day under all weather conditions, and charges no user fees. GPS offers comparatively high accuracy when operational conditions are favourable [1]. GPS uses a system of satellites orbiting the planet to help a receiver (an Android device in this paper) determine its current location. The term GPS refers to the entire GPS system, which consists of satellites, receivers, and the control stations that monitor and adjust it. The receiver that is located in the phone is useless without the rest of the system [2].

The Open Handset Alliance (OHA) is a business alliance comprised of many of the largest and most successful mobile companies on the planet. Its members include chip makers, handset manufacturers, software developers, and service providers. The entire mobile supply chain is well represented. Andy Rubin has been credited as the father of the Android platform. His company, Android Inc., was acquired by Google in 2005. Working together, OHA members, including Google, began developing a non-proprietary open standard platform based upon technology developed at Android Inc. that would aim to alleviate the aforementioned problems hindering the mobile community. The result is the Android project. Most Android platform development is completed by Rubin’s team at Google, where he acts as VP of Engineering and manages the Android platform roadmap [3].

This paper aims to develop a system to track anobject by aset of monitoring servers. The monitoring servers at every time can query an object location by sending a request through GSM and receiving its current location as a marker on the map. The monitoring server device uses Google maps to draw current object location on the map. The object and monitoring server programs have been programmed using java programming language on android studio software development kit (SDK).

Paper is organized as follows. Section II describes the literature review of related works. Section III includes the proposed system and represents the steps of the algorithm.

II. RELATED WORK

Many researchers have considered the tracking system as a headline in their work. Brief reviews on some of these researches are introduced in following:

Alassady and et al present system architecture for Human security monitoring, which can be used in the personal locators for children, elderly people or those suffering from Alzheimer’s or memory loss, and monitoring the

movement for law execution. This architecture consists of GPS part for collecting information about a movable object, spatial database part for storing this information bylistener server [4].
Kamel and George checks patient health status using wireless medical device that is connected to an Android based mobile device, also track patient's position using internal GPS sensor and sends these data to a server using GPRS[5].
Chandil and et al proposed a GPS tracking system called Goo-Tracking that is composed of commodity hardware, open source software and an easy-to-manage user interface via a web server with Google Map or via Google Earth software [6].
Dabbiere has designed a tracking system. The tracking capabilities of the system uses a global positioning system and includes a complete reconstruction of a series movements over a period of time. The system gathers the exact time and position information of an employee over a period of each work day and the task being worked on at any given moment of time [7].

Kamel proposed a vehicle tracking system based on GPS and GPRS. The location of the vehicle is retrieved using embedded GPS sensor. A modified coding method is used to encode and compress location data before it is sent to offer cost effective usage of network traffic. The privacy of the transmitted data is guaranteed using a simple security mechanism. The encoded and encrypted location data then send to tracking server using GPRS technology. The authorized user can track a vehicle using a secure web interface [8].
Bahl and et al present RADAR, a radio-frequency (RF)-based system for locating and tracking users inside buildings. RADAR operates by recording and processing signal strength information at multiple base stations positioned to provide overlapping coverage in the area of interest [9].

All the previous mentioned researches depend on GPRS technology to transmit location data. In our paper we have used a new method that is not based on GPRS and use SMS over GSM to transmit data.

III.PROPOSED SYSTEM

The proposed objecttradingsystem works on Android platform for mobile devices, using built in GPS receiver to determine the current location of the device, and using GSM network to send location data. Figure 1 illustrates the proposed system work flow.

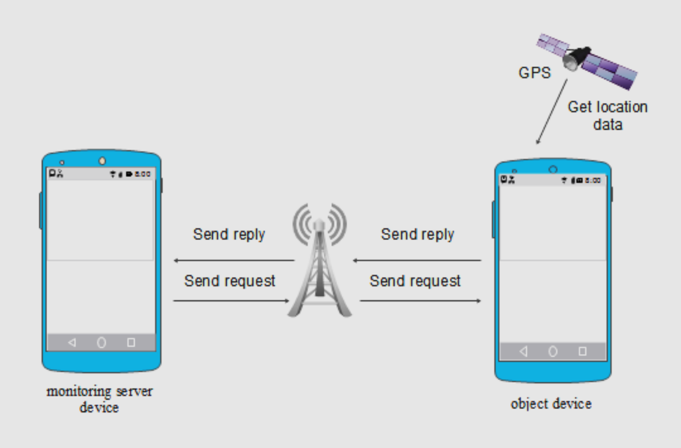


Fig. 1 Object Tracking System Work Flow

This paper aim to develop tracking system with two components: monitoring server and object as shown in figure2. First the phone number of monitoring server stores at object device. Then by starting monitoring the object will be listening continually to any incoming request from monitoring server device. By receiving the request, the object device checks its current latitude and longitude and send back this information to the monitoring server device. The monitoring server's device by receiving the object's location data using Google maps shows the current location of object on screen. All the transmitting data is based on method described in [10] to guarantee the functionality of the system that is based on third party agent who can keep the address secrecy of both communicators besides keeping confidentiality, integrity and availability.

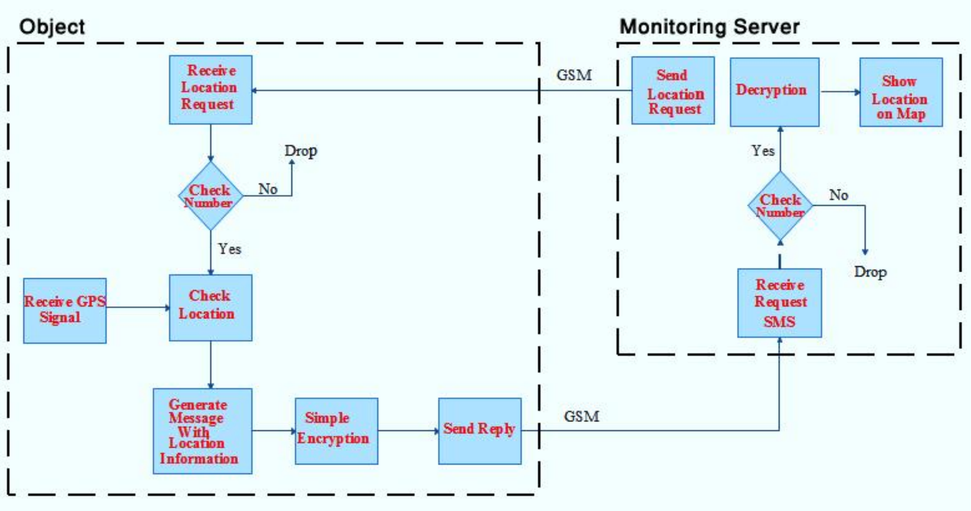


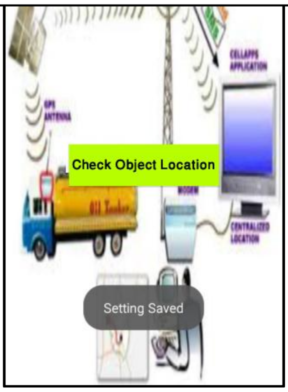
Fig. 2 Object Tracking System Flowchart

The first step is to read and write the system related information including members' phone numbers on SD card as shown in figure 3 This step is done through the use of two functions: setting writer and setting reader.

In writing procedure file output stream is used which is responsible for entering the data and then storing them to a predefined location in SD card and through the output stream procedure writer will flow data from the application to the internal local file.

In reading procedure file input stream is used that reads the content of file and importing it into the program. The data is taken out after its flow through the input stream function which input data character by character from file. The manifest file needs the following permissions:

```
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"></uses-
permission>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-
permission>
```


Check Object Location

Setting Saved

Fig. 3Setting interface

For sending Request and reply the SMS manager class has been used. The sending process consists of two steps: First, initializing the object and get default ( ) function. Second, sending the SMS using predefined phone number using send text message ( ) function. Note that this function takes five parameters, three of them will be null which are as the following:Calling center number, sending delivery and receiving deliver, respectively

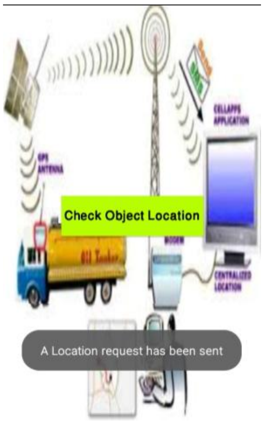
```
void sendSMS(String phoneNo, String msg) {

SmsManager smmanager = SmsManager.getDefault();
   smmanager.sendTextMessage(phoneNo, null, msg, null, null);
}
```

The manifest file needs the following permissions:

```
<uses-permission android:name="android.permission.SEND_SMS"></uses-permission>
```

Figure 4 shows the sending location request interface which is part of the monitoring server application.



A Location request has been sent

Fig. 4 Send SMSInterface

To receiving incoming SMS a Broadcast Receiver has to be defined. In the broadcast receiver the onReceive function has to be overloaded to get bundle variable by calling intent.getExtras. The incoming message is originally a pdus type which is an object variable for saving the original data. Every pdus can be converted to an SMS message by calling createfrompdu function. Every incoming contains a set of data which can be retrieved using following functions:
1.The content of message by calling getMessageBody.
2.The time of receiving the message by calling getTimestampMillis.
3.The sender's phone number by calling getOriginatingAddress.

```
SmsMessage message;
message = SmsMessage.createFromPdu((byte[]) pdus[i]);
String msg = message.getMessageBody();
long when = message.getTimestampMillis();
String sender = message.getOriginatingAddress();
```

The manifest file needs the following permission:

```
<uses-permission
android:name="android.permission.RECEIVE_SMS"></uses-permission>
```

The Broadcast Receiver first receives the incoming SMS which contains transmitted data. This data, then, has to be send to other part of the application which is response for checking the incoming SMS which may consist of location request (incoming from monitoring server to object) or location data (incoming from object to monitoring server). The following part of code shows broadcasting of incoming message from broadcast receiver to other part of the application as explained in figure 5

```
Intent in=new Intent("SmsMessage.intent.MAIN").putExtra("get_msg", sender + ":" + msg);
context.sendBroadcast(in);
```

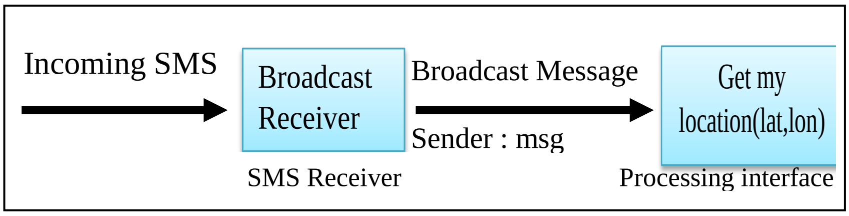
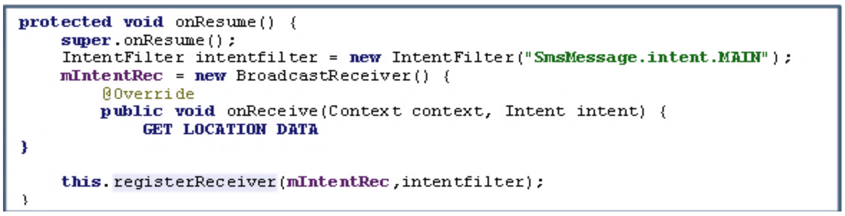


Fig.5 Location Listener Interface

After broadcasting the incoming SMS internally (inside the application itself), the broadcasted message has to be received by calling Broadcast Receive function inside the processing class as illustrated in the below part of code. After receiving the message and decrypting it, the current object's location data (latitude and longitude) will be extracted from the message.



```
protected void onResume() {
    super.onResume();
    IntentFilter intentfilter = new IntentFilter("SmsMessage.intent.MAIN");
    mIntentRec = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            GET LOCATION DATA
}

    this.registerReceiver(mIntentRec,intentfilter);
}
```

The monitoring server to query the current location of the object has to send the request through GSM network. The object by receiving the incoming location request and checking the sender phone number will retrieve the current location data and send them back to the monitoring server.

Object application to be able to know the current location will query location data by using *Location Listener* that determines longitude and latitude data of it. Then the *Location Manager* instance that manages the object's current location and then retrieve the last known location to the application by calling *getLastKnownLocation*function and passing the *Location Provider* parameter which is the built in GPS receiver.

```
location = lm.getLastKnownLocation(locationProvider);
```

In the paper the system update location data every 2000 ms and with accuracy of five meters which is enough for monitoring server to determine her current object location. The paper uses a simple model to coding latitude and longitude [11]. This simple model deals with both redundancy and desired accuracy of the location data. After determining the desired accuracy for the latitude/longitude coordinate parameters, the system starts to code this set of location data using simple predictive coding scheme.The system uses Google map to show the current object location in the monitoring server interface. To using Google maps we have to register the application in the Google maps. To add marker on Google map the addmarker function has been used. This function takes three parameters: Markeroption, position and title marker. This function will be called after receiving the object location SMS, decrypting the message and getting current location. Figure 6 shows final object's location in monitoring server device.

```
mMap.addMarker(new
MarkerOptions().position(childLocation).title("Your Child
Location"));
```

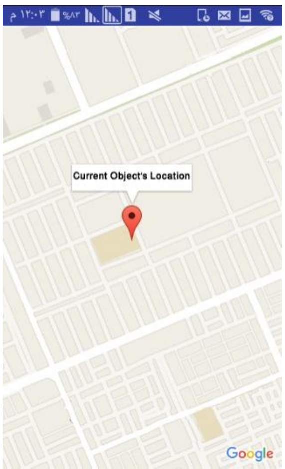


Fig. 6 Current object’s location

To encrypt the transmitted messages between monitoring server and object XOR encryption [12] is used. To prevent sending special characters which may result in some errors, the encrypted message will be padded by value 96 which represents small ‘a’ character in ASCII. The following part of code shows the main ciphering and deciphering function:

```
for (int i = 0; i < input.length(); i++)
     output+= ((char) ((input.charAt(i) ^ key[i % key.length])^ 'a'));
```

Figure.7 shows the XOR encrypting and decrypting process.

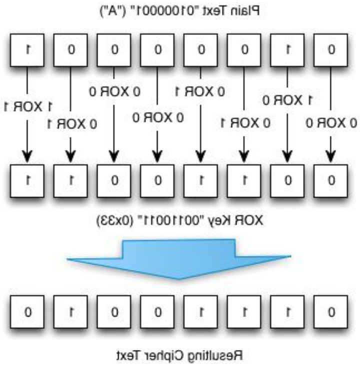


Fig.7 XOR encrypting and decrypting

The monitoring server and object applications have been developed using java programming language on Android studio 1.5 SDK. After running monitoring server and object on a Samsung S2 android based device, and testing resource consumption of these two applications, the CPU load and memory consumption information on figures 8 and 9 have been registered.

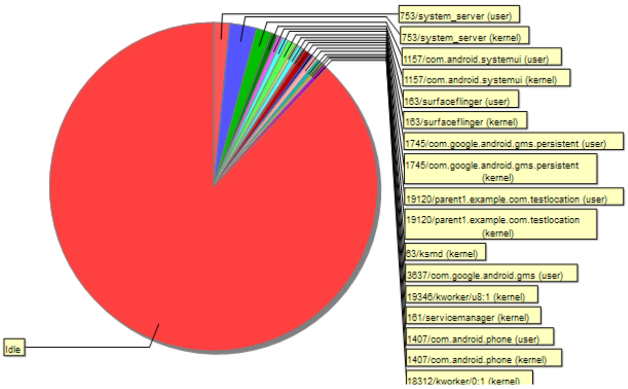


Fig.8CPU Load

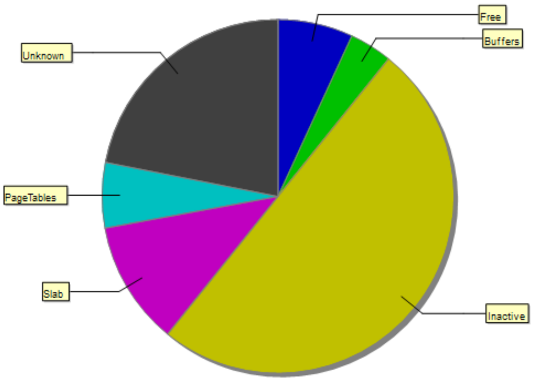


Fig.9Memory Usage

VII. CONCLUSION

Many conclusions have been derived during the development and test results, among these conclusions. The object program using GPS positioning to determine the current location of the device. During the tracking the built in GPS receiver has to be enabled. The system does not need an Internet connection in order to determine and track the object's location. The location information will be send via GSM network. To show the current location of object, monitoring server needs Internet connection to connect to Google maps and retrieve the map. Note that without Internet connection the marker will be draw on and empty background. The monitoring server and object application needs very small amount of CPU and memory, so these applications can be used along with other applications of the devices without reducing its speed.

The system is working on GSM to transmit messages between monitoring server and object. Internet connection can be used to transmit these messages. XOR encryption is used to encrypt and decrypt messages. Other encryption algorithms such as DES and AES can be used. The application works on Android platform. Other platforms' versions such as Windows mobile or iOS can be developed. Monitoring server can only know current location of her object. A more complicated algorithm can be developed to track the current and previous locations of her object.

REFERENCES

[1] D. Dao, C. Rizos, J. Wang," Location-Based Services: Technical And Business Issues ",GPS Solutions, volume 2002 pp. 169–178,2002

[2] G. Milette, A. Stroud,"Professional Android™ Sensor Programming ", John Wiley & Sons, Inc, 2012

[3] S. Conder,L. Darcey, "Android™ Wireless Application Development ", Pearson Education, (2012).

[4] Al-Assady, A J., Bahaa T. Sh., Hazem M. E. "A Proposed Model for Human Securing using GPS", International Journal of Electronics Communication and Computer Engineering, (2015): 689-697.

[5] M. B. M. Kamel, L. E. George. "Remote Patient Tracking and Monitoring System." International Journal of Computer Science and Mobile Computing 2, no. 12 (2013): 88-94.

[6] N.Chadil, R. Apirak, and K. Phongsak " Real-Time Tracking Management System Using GPS, GPRS And Google Earth" , University, Technology North Bangkok, (2008).

[7] A.J.Dabbiere, "Methods and devices utilizing a GPS tracking system", U.S. Patent, (2001).

[8] M. B. M. Kamel. "Real-Time GPS/GPRS Based Vehicle Tracking System." International Journal of Engineering And Computer Science 4, no. 8 (2015): 13648-13652.

[9] P. Bahl, V.N. Padmanabhan, " Enhancements To The Radar User Location And Tracking System", University California at San Diego, (2000).

[10] M. B. M. Kamel, L. E. George. "Secure Model for SMS Exchange over GSM." International Journal of Computer Network and Information Security 8, no. 1 (2016): 1-8.

[11] M. B. M. Kamel, L. E. George. "Simple Model for Coding Geographic Location Coordinates." International Journal of Scientific & Engineering Research 4, no. 7 (2013): 2468-2470.

[12] M. Al-qdah, Y. H. Lin. "Simple Encryption/Decryption Application." International Journal of Computer Science and Security 1, no. 1 (2007): 33.