

## Mobile Robot Controlled with Android Device

Pásztó Peter · Elektrotechnika

27.05.2011



Modern mobile phones with Android operating system are including many types of sensors and communication interfaces. Most of them have for example G-sensor, compass, proximity sensor, ambient light sensor, camera, Bluetooth, GPS, Wi-Fi, USB and also they usually have a powerful processor. From the large number of possibilities of modern mobile phones comes the idea of using their sensors in robotic applications. Therefore this article is trying to discuss about the ways how to use mobile phone sensors and Android operating system for controlling mobile robots.

### 1. Introduction

Mobile robots usually use information from many types of sensors for navigation and motion control. Modern mobile phones do have also some types of sensors those can provide information for controlling the mobile robot. We will talk about how could be these sensors used for this purpose.

Let's say we want to create an application that can run under Android operating system and control a mobile robot with the Android device fixed on it. The mobile robot should be able to move in internal or external environment without collision with obstacles. We can also consider more mobile robots controlled with Android devices and after mapping the environment they can communicate to meet at a desired place.

### 2. Using Android device sensors for mobile robot controlling

To navigate a mobile robot with an Android device in an unknown environment we need to obtain information about the environment from the available sensors. For establishing communication between mobile robots we need to use some sensors also as communication interface. Therefore we can divide mobile phone sensors into three categories:

- Sensors used for collecting information about environment
- Sensors used for communication between mobile robots
- Sensors used for collecting information and also for communication

#### 2.1 Mobile robot movement controlling

Usually to control the movement of a mobile robot we need to control the speed of

rotation of his engines or the rotation direction. This could be done with one of the Wi-Fi or Bluetooth interfaces. The mobile robot has to have the engines connected to a control circuit with one of these interfaces communicating with the mobile phone. Using an USB interface for controlling the engines is a little bit complicated because Android does not provide a direct access to the USB yet.

### **2.1.1 Controlling with Bluetooth**

The Android Developers internet site shares several examples and descriptions about programming the Android device's Bluetooth interface. One of the documents [1] describes how to use the Android Bluetooth APIs to accomplish the four major tasks necessary to communicate using Bluetooth: setting up Bluetooth, finding devices that are either paired or available in the local area, connecting devices, and transferring data between devices. It is quite important to read the Managing a Connection part because there is said that the methods `read(byte[])` and `write(byte[])` used for reading and writing data to the stream are blocking calls and therefore it is recommended to use a dedicated thread for all stream reading and writing.

While controlling the mobile robot with Bluetooth we also have to consider that our application will not only control the robot but also will do other tasks for gathering information about the environment, computing the mobile robot's moving direction and therefore the Bluetooth connection should not block these tasks. It should wait for these tasks to finish and periodically send information to mobile robot's engine control circuit. This period should be long enough to allow the other tasks to finish or should be variable and could run after finishing signal from the necessary tasks.

### **2.1.2 Controlling with Wi-Fi**

This type of controlling is a bit harder than Bluetooth controlling. The Android Developers site [2] also has a description about working with Wi-Fi, but there is more to do to establish a connection between the phone and the robot. You have to have some knowledge about Wi-Fi networking. For basic connection it is necessary to set the wireless network configuration parameters (SSID, passwords, etc.) and after it establish a connection.

## **2.2 Collecting information about the environment**

There are several problems with navigation of a mobile robot in an unknown environment. He has to be able to detect obstacles and avoid collision with them. Usually robots do create a map of the unknown environment while moving in them. If the robot is situated in external environment he has to take care about turnover.

### **2.2.1 Mobile phone as sonar**

One of the most used methods for obstacle detection in robotics is ultrasonic distance measurement. It is based upon the reflection of sound waves [9]. Sound waves are defined as longitudinal pressure waves in the medium in which they are travelling. Subjects whose dimensions are larger than the wavelength of the impinging sound waves reflect them; the reflected waves are called the echo. If the speed of sound in the medium is known and the time taken for the sound waves to travel the distance

from the source to the subject and back to the source is measured, the distance from the source to the subject can be computed accurately. The medium for waves is air and usually the waves used are ultrasonic, since it is inaudible to humans. The measurement principle is shown at the Fig. 1.

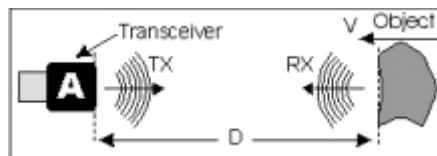


Fig. 1. Principle of ultrasonic distance measurement

The amplitude of the wave reflected is directly proportional to how much surface is available on the object [4] for coherent reflection. Surface size, shape and orientation, are major factors contributing to the strength of the reflected signal; material composition is also a factor. A part of the wave landing on the surface of the material is reflected, while a part of the wave penetrates the material and is eventually reflected of any surface boundaries encountered while travelling within the material. Hence you will receive a signal coming from inside the material as well, but it is minuscular.

If  $T_0$  is the time at which a burst of pulses is transmitted and  $T_1$  is the time at which the transmitted burst is received, then the distance  $D$  from the sensor to the object can be determined as follows:

$$D = \frac{1}{2}C(T_1 - T_0) \quad (1)$$

Where  $D$  is the distance to object,  $C$  is the speed of sound. Speed of sound in air is nominally 344m/s at 25 degrees, this speed drops to 334 at 0 degrees. The temperature dependency is second order given that other parameters are constant. The principle of sonic distance measuring uses an Sonar application available on Android Market [5] . It does not use ultrasound because the phones speaker is not dimensioned for creating an ultrasound. A screenshots of this application are shown at Fig. 2.

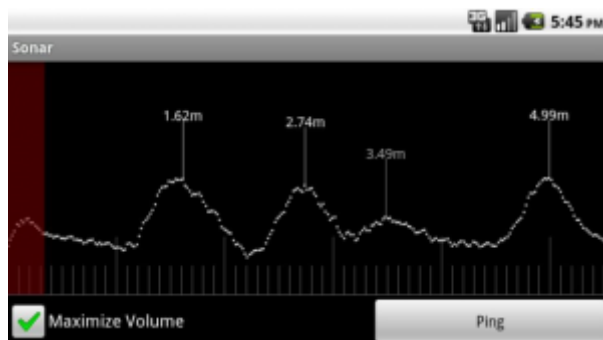


Fig. 2. Screenshot of application Sonar (ad)

This kind of distance measurement using a mobile phone has some inaccuracies because of the different positions of the speaker and the microphone on the phone.

### 2.2.2 Using mobile phone's camera for object detection

The mobile phone's camera is a very useful sensor for scanning the environment as today's computer vision algorithms are often used to detect obstacles or to find the direction for mobile robot to go. Their disadvantage is that they have high requirements for computational complexity and they are time-consuming. Probably we can't expect a real-time image processing but there could be some basic applications that mobile phone can handle in a relatively short time.

One of those basic applications can be edge detection. If working with the mobile device's camera in lower resolution mode the time of edge detection in the input image is a few seconds. At Fig. 3. is shown an application that can detect edges in the input image. Edge detection is not an final algorithm for obstacle detecting but there can follow many other computer vision algorithms that can bring a good result - like finding lines using Hough transform or detecting areas with same color using color filter. These computer vision algorithms can be combined with the previously described sonar to detect the obstacles and to gather information about them.

An example could be the initial object detection with the sonar followed by edge detection using the camera and applying lines finding Hough transform to compute the path for mobile robot he can move along. Some Android devices also have a flash light with the camera so it is possible to use the device's "vision" also in the dark.



*Fig. 3. Application detecting edges*

The devices flashlight can also be used for detecting the nearby obstacles in front of the mobile robot as they may reflect the light with high intensity if there are near. Mobile robot can create a map of the environment after detecting the obstacles by remembering their positions.

## **2.3 Determining the position of the mobile robot**

The type of sensors used for mobile robot position determination depends on the environment the robot is situated in. In external environment is useful the GPS but to determine the position in internal environment is a bit more complicated.

### **2.3.1 GPS and Compass**

If the robot is outdoors the GPS sensor can be used to determine the mobile robot's position. There are several methods for GPS navigation (one of the best known applications in Android using GPS navigation is Google Maps). Using GPS the robot can measure the traveled distance however the GPS's signals are commonly corrupted by various disturbances (like signal bounces, inaccurate receivers etc.) so it is possible that these disturbances should be adequately mitigated at the highest possible rate [8]

GPS navigation often uses an additional sensor like compass. Most of Android mobile devices have implemented compass so it can be used as an addition to determine the mobile robot's direction.

### 2.3.2 Accelerometer

To determine the position of a mobile robot in internal environment usually a GPS sensor cannot be used because of signal unavailability so it is much harder to get the mobile robot location after its movement. There are many approaches how to measure the traveled distance of the mobile robot even using the available sensors of the mobile phone. For example using the camera and comparing the objects in two images taken by mobile robot while moving at the beginning and the end of the movement. This approach should be very time-consuming for a mobile phone.

The other approach often used in robotics is measuring the traveled distance with accelerometers. Mobile phone with 2-axial accelerometer mounted on a mobile robot can provide a simple dataset from which the xy position can be calculated [7] using the integral:

$$x(t) = \int \int a(t) dt dt \quad (2)$$

We have to realise that accelerometers are measuring the acceleration and the acceleration of a mobile robot with constant speed is null. Therefore while using this approach of travelled distance measurement the mobile robot should accelerate or decelerate while moving to make the sensors able to measure it's acceleration. Also there can arise very large variations by double integration of acceleration so the real position of a mobile robot should vary from the calculated.

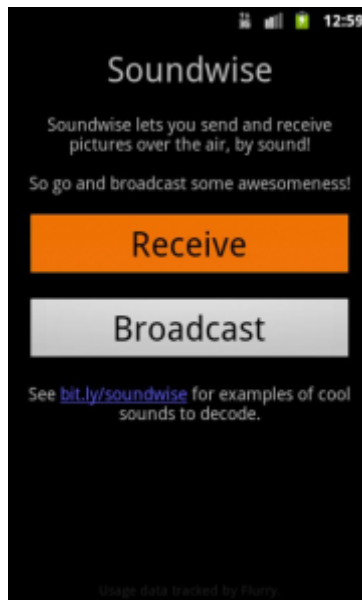
## 3. Communication between more mobile robots

The communication could be done with the same sensors as described above for navigation purposes. It is necessary to consider switching the sensors between navigation or communication modes.

We can talk about wireless communication made by Wi-Fi or Bluetooth where the information (for example about the common position of mobile robots to meet at) can be transferred from one mobile robot to the other. One of the major problems of this kind of communication is that it is often available only between two mobile robots (one is the server and second is client) and therefore they have to manage the communication intervals between each other.

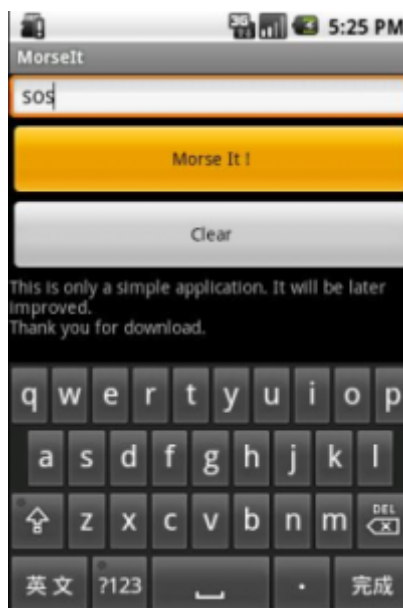
Other probably efficient way of communication is communication by sound or by light.

One of the robots that wants to share information with the others can start to make sound of specified frequency and the others will “hear” it using their microphones. Each frequency of sound can be the holder of some information or every byte of sent information can be coded with specific sound frequency. An example of an application on Android market for sound communication can be the application called Soundwise (and Soundwise Receiver) that turns pictures into sound vice versa.



*Fig. 4. Application Soundwise that turns picture into sound vice versa*

The same approach as sound generating can be the light communication. By blinking the mobile devices flash light can be sent information to other robots while they can “see” the blinking of one of the robots that wants to share information. On Android market there is an example of an application of coding a message into light flashing using Morse code is available. The application name is Morse It! [6].



*Fig. 5. Application Morse It! that transforms the entered text into Morse code*

Light detection using mobile phone’s camera could be done by creating an application that can handle the input image from the camera and compute the light intensity in the image. Because Android operating system uses the YCbCr\_420\_SP (NV21) [2]

color model the computation of the light intensity can be made by summing the value of the Y channel of every pixel in the input image.

#### 4. Conclusion

Modern mobile devices with Android operating system have many sensors those can be used in robotics for mobile robot navigation. In this article there were described the possibilities of using the phone's sensors for that purpose. There was shown that already exists many applications on the Android Market whose combination could lead into mobile robot navigation algorithm.

As mobile robot navigation is not a simple process there are also many difficulties in navigation using mobile phone. The biggest problems are with the mobile device's performance and its sensor's inaccuracies and slow responses. The navigation algorithm must take into account the uncertainties arising from these problems and ensure the navigation of the mobile robot with combination of as many sensors as the mobile device has. It is an interesting experiment to create an algorithm that could navigate the mobile robot using a mobile phone.

#### Literature

1. <http://developer.android.com/guide/topics/wireless/bluetooth.html>
2. <http://developer.android.com/reference/android/hardware/Camera.PreviewCallback.html>
3. <http://developer.android.com/reference/android/net/wifi/package-summary.html>
4. <http://www.hexamite.com/hetheory.htm>
5. <https://market.android.com/details?id=com.dicon.sonar.ad>
6. [https://market.android.com/details?id=pete.android.morseit&feature=search\\_result](https://market.android.com/details?id=pete.android.morseit&feature=search_result)
7. J. D. Jackson, D. W. Callahan - Location Tracking of Test Vehicles Using Accelerometers, Proceedings of the 5th WSEAS Int. Conf. on CIRCUITS, SYSTEMS, ELECTRONICS, CONTROL & SIGNAL PROCESSING, Dallas, USA, November 1-3, 2006, 333 - 336
8. Jurišica, L., Vitko, A., Duchoň, F., Kaštan, D.: Systém GPS-princíp, prednosti a nedostatky (2. časť). In: Automa. - ISSN 1210-9592. - Roč. 17, č. 2 (2011), s. 38-40
9. M. Raju, Ultrasonic Distance Measurement With the MSP430, Texas Instruments, Application Report SLAA136A - October 2001

---

Spoluautorom článku je prof. Ing. Peter Hubinský, PhD.

---