مجلة العلوم البحثة والتطبيقية
Journal of Pure & Applied Sciences
www.Suj.sebhau.edu.ly ISSN 2521-9200
Received 05/01/2018 Revised 02/02/2018 Published online 14/02/2018

Low-cost Arabic SMS Text Messages

*Amna Elhawil , Youssef Gdura , Hajer Elhamrouni
Computer Engineering, Engineering Faculty/University of Tripoli, Libya
Corresponding Author: A.elhawil@uot.edu.ly

Abstract Short Message Service (SMS) technology is a wide field and an easy communication tool. However, many ideas can be implemented to get more benefits from this technology. The purpose of this paper is to implement a mobile application for transferring low-cost Arabic messages using Short Message Service (SMS) technology. The results are very motivating. Users have very simple interface with considerable code to perform data transfer via mobile. The developed application can be installed on smartphones, downloaded as various mobile software distribution platform.
Keywords: (Android, SMS, Unicode, Arabic SMS and English SMS)

ارسال الرسائل العربية القصيرة عن طريق الهاتف النقال بتكلفة أقل

*أمنة الحويل و يوسف قدورة و هاجر الحمروني
قسم هندسة الحاسب الآلي– كلية الهندسة – جامعة طرابلس، ليبيا
*للمراسلة: A.elhawil@uot.edu.ly

الملخص يعتبر مجال تقنية الرسائل القصيرة مجال واسع وأداة سهلة الاستخدام. لذلك العديد من الأفكار يمكن برمجتها لتمكين استفادة أكبر من هذه التقنية. الهدف من هذه الورقة البحثية هو تطوير تطبيق لارسال النص العربي بتقنية الرسائل القصيرة بتكلفة أقل. النتائج التي تم التوصل اليها مشجعة. لقد أصبح لدى المستخدم تطبيق سهل بواجهة بسيطة يمكن تحميله في الهواتف الذكية كأي تطبيق رسمي.
الكلمات المفتاحية:(الرسائل القصيرة، أندرويد، التشفير، تكلفة أقل، الهاتف النقال**)**

Introduction

Mobile phone users use (Short Message Service) SMS to communicate with each other routinely. SMS is more convenient than email to communicate amongst distributed and mobile groups of people. The SMS concept was developed in the Franco-German GSM cooperation in 1984 by Friedhelm Hillebrand and Bernard Ghillebaert [1]. SMS first appeared in Europe in 1992. It was included in the Global System for Mobile Communications (GSM). Later it was ported to wireless technologies like CDMA and TDMA. The GSM and SMS standards were originally developed by ETSI. ETSI is the abbreviation for European Telecommunications Standards Institute. Now the Third Generation Partnership Project (3GPP) is responsible for the development and maintenance of the GSM and SMS standards [2]. In 2010, 6.1 trillion (6.1×10^{12}) SMS text messages were sent [3]. This translates into an average of 193,000 SMS per second. SMS has become a huge commercial industry, earning $114.6 billion globally in 2010 [4]. However, implementing sophisticated applications based on SMS technology has become very important topic. The main purpose of this paper is to investigate a new method for transferring Arabic text between mobiles using SMS with low cost. The application should be implemented and uploaded as any professional application. It should also be easy to use and does not conflict with the standard SMS applications.

SMS Technology

As mentioned before, SMS stands for Short Message Service. It is a technology that enables sending and receiving messages between mobile phones. As suggested by the name "Short Message Service", the data can be held by an SMS message is very limited. One SMS can contain at most 140 bytes (1120 bits) of data, so one SMS message can contain up to:

- 160 character if 7-bit character encoding is used. 7-bit character encoding is suitable for encoding Latin characters like English alphabets.
- 70 characters if 16-bit Unicode UCS2 character encoding is used. This includes non-Latin characters like Arabic characters [5].

However, SMS text messaging supports languages internationally. It works fine with all languages supported by Unicode, including Arabic, Chinese, Japanese and Korean**.**
Furthermore, one drawback of the SMS technology is that one SMS message can only carry a very limited amount of data. To overcome this problem, an extension called concatenated SMS text message (also known as long SMS) was developed. A concatenated SMS text message works as following: The sender's mobile phone breaks down a long message into smaller parts and sends each of them as a single SMS message. When these SMS messages reach the destination, the recipient mobile phone will combine them back to one long message [6].

Coding Arabic Text

The Unicode of the Arabic letters is shown in Table 1. For example, the Arabic letter "أ" has the Unicode of $(0623)_{16}$. The main 36 Arabic letters have the Unicode of $(0621)_{16}$ to $(0640)_{16}$ and from

$(0641)_{16}$ to $(064A)_{16}$. Note that the codes from $(063B)_{16}$ to $(0640)_{16}$ are reserved. On the other hand, Table 2 lists the Unicode of the English characters [7].

Table 1: Unicode of Arabic characters.

	U+062x	U+063x	U+064x
0	□	ذ	-
1	ء	ر	ف
2	آ	ز	ق
3	أ	س	ك
4	ؤ	ش	ل
5	إ	ص	م
6	ئ	ض	ن
7	ا	ط	ه
8	ب	ظ	و
9	ة	ع	ى
A	ت	غ	ي
B	ث	□	ً
C	ج	□	ٌ
D	ح	□	ٍ
E	خ	□	َ
F	د	□	ُ

Table 2: Unicode of English characters.

	U+004x	U+005x	U+006x	U+007x
0	@	P	`	p
1	A	Q	a	q
2	B	R	b	r
3	C	S	c	s
4	D	T	d	t
5	E	U	e	u
6	F	V	f	v
7	G	W	g	w
8	H	X	h	x
9	I	Y	i	y
A	J	Z	j	z
B	K	[	k	{
C	L	\	l	|
D	M	]	m	}
E	N	^	n	~
F	O	_	o	

The main idea of this work is to convert the Arabic Unicode to English. From Tables 1 and 2, it is notable that the similarity between the both tables regardless of the letters. So if Arabic Unicode has to be converted to English one can subtracting the number $(5e0)_{16}$,that is $(0621)_{16} - (0041)_{16}$), from Arabic Unicode. For example, the Arabic letter 'ب' which has the Arabic Unicode of $(0628)_{16}$ can be converted to English Unicode to become $(0048)_{16}$ that represent the English letter 'H' as in Table 2. Moreover, the Arabic message

"السلام عليكم ورحمة الله وبركاته..."

has the equivalent English letters of

"*GdSdGe Ydjce hQMeI Gddg hHQcGJg..*"

We use this simple technique to encode any Arabic text to English one. In order to distinguish the coded messages in the receiving side, a header or *Start of Scan* code (SOS) is added at the beginning of the message. In this work, the SOS is chosen to be "*h1r*". However, for our example the complete sent message will look like:

"*h1rGdSdGe Ydjce hQMeI Gddg hHQcGJg..*" .

Fig. 1 shows the bock diagram of the procedure that is performed at the sender side.

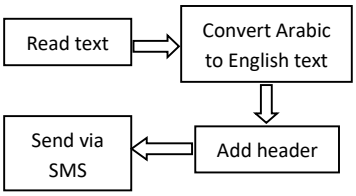

Fig. 1: Block diagram of sender side

Implementation

The most common program environments that deal with SMS technology are Android, iOS, BlackBerry, HP WebOS, Symbian OS, Bada from Samsung and Windows Mobile. In this paper, we have chosen Android software. That is because Android is free, open source and runs on most major OS platforms. The Android developer tools (ADT) Bundle includes Eclipse + ADT plug-in, Android SDK tools, Android Platform tools and latest Android platform.

The programming has been implemented using both Java and XML languages. The minimum requirement needed in order to run the application is Android-powered phone that includes minimum Android 2.3.3 version.

Sending the message

There are two possible ways to send a text message from an android application:

1. The first way is to send the message by invoking the built-in SMS application.
2. The second way is to send the message programmatically from the implemented application.

In this work we have used the first method. It should be noted that the following fixed permission

<users permission.android:name ="android. permission.SEND_SMS" />

has to be included in application's *Android Manifest.xml* file to allow the application to send SMS messages.

Receiving the message

There are three main steps to extract the original Arabic text at the receiving side. Fig. 2 shows the block diagram. It starts with *first*, catching and recognizing the SMS. *Second*, decoding the

message and convert it back to Arabic letters and finally the system will display the message.

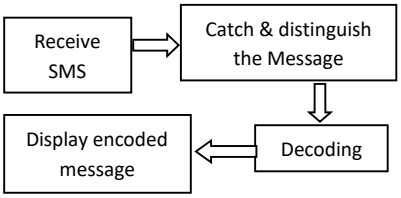

Fig. 2: Block diagram of receiver side

1. Catching and recognizing SMS

In Android, SMS is first interpreted using *Broadcast Receiver* component. This component used also to retrieve the contents of the message which will be handled by our project. Our application recognizes the message by its header or the *Start of Scan code* (SOS) which is "*h1r*".

2. Decoding

The decoding step is performed by converting the English letters back to Arabic letters. That is done by adding the code $(5e0)_{16}$ to each letter. The result is the original text that has been entered by the sender.

Results

The Fig. 3 shows the icon of our application. In order to explain the advantages and main features of our application, we will compare between the performance of sending a message using the traditional method and sending it using our application. We use the following Arabic text as an example:

السلام عليكم ورحمة الله وبركاته ...

هذا النص عينة لاختبار برنامج ارسال رسالة باللغة العربية

على الجهاز المحمول بتكلفة أقل للمشاركة في مؤتمر 2017 ICST

Fig. 4 depicts this text in the standard Android's messages application. This message consists of 149 letters (with spaces).

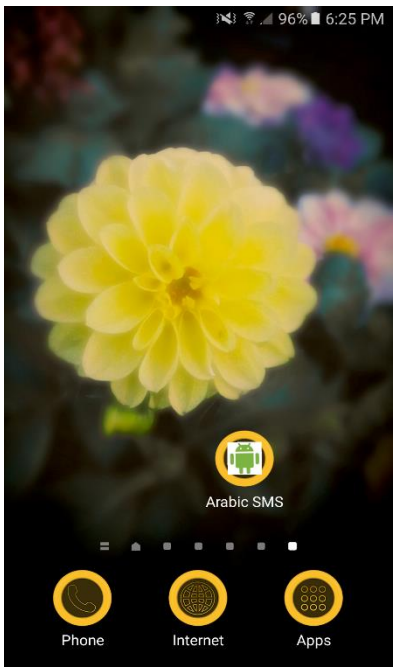

Fig. 4: Application icon

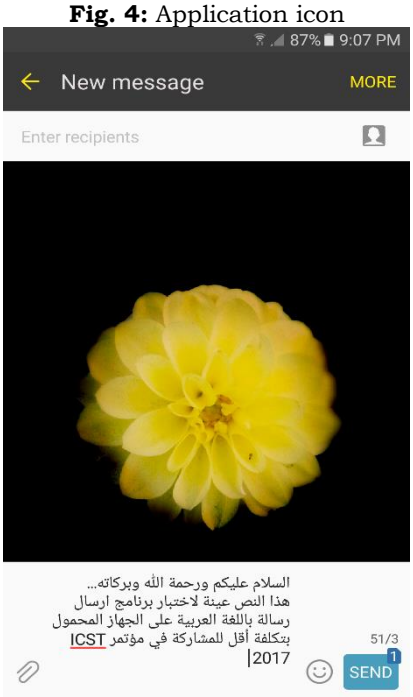

Fig. 4: Sending SMS by Standard Android's messages application

Thus, it requires three SMS messages to be sent successfully. That is because of the limit of Arabic letters per SMS which is 70 Arabic letter/ SMS.

On the other hand, the same text is encoded to English letters using our developed application as shown in Fig. 5. It requires only one SMS to be sent. That is because the English SMS limit is 160 English letter/ SMS. So the cost is also reduced. However, the conversion of Arabic text into English one and sending it as a coded English message gives higher SMS limitation to Arabic text regardless of the small SOS at the beginning of the message.

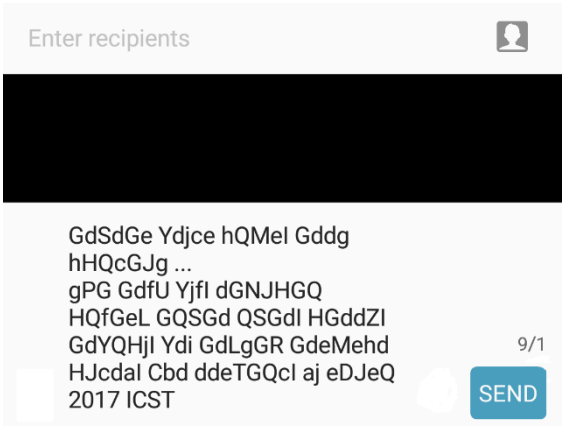

Fig. 5: Sending SMS by our application

Conclusion

In summary, the standard SMS uses 16-bit Unicode UCS2 character encoding for Arabic letters, while it uses 7-bit Unicode letters to send Latin letters. In this paper the Arabic letters are encoded to Latin letters. This increases the limit of Arabic SMS messages from 70 to 160 characters/SMS and therefore reduces the cost of Arabic SMS by triple of its original cost.

References

[1]- F. Hillebrand, F. Trosby, K. Holley and I. Harris, SMS the creation of Personal Global Text Messaging, 2010, John Wiley & Sons, Ltd.

[2]- N. Legis, Mobile Technology Integration to the Moodle System; Institutional Repository of Kaunas University of Technology, 2009.

[3]- The World In 2012 ICT facts and figures- The rise of 3G, International Telecommunication Union (ITU), 2010.

[4]- S. Katie, OMG: Text messaging turns 19 this week … and this is the Brit we have to thank for our sore thumbs, Mailonline, December 7, 2011.

[5]- IP NGN Mobility; Lesson Refreshments, https://www.scribd.com/document/324508020/Day1b-pdf, 2006.

[6]- http://www.developershome.com/sms/smsIntro.asp", 2013.

[7]- Arabic script in Unicode, https://en.wikipedia.org/wiki/Arabic_script_in_Unicode, 2016