

Appendix A

XML android code

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="gsm.app"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="21" />
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".Main"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
```

```
</manifest>
```

JAVA android code :

```
package gsm.app;
```

```
import android.os.Bundle;
import android.app.Activity;
import android.telephony.SmsManager;
import android.view.Menu;
import android.view.View;
```

```
public class Main extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
```

```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
```

```
    public void buttonOnClick(View view)
```

```
    {
        switch(view.getId())
```

```
        {
```

```
            case R.id.button1:
```

```
                // Code for forward click
```

```
                    String messageToSend = "forward";
```

```
                    String number = "+24994377844";
```

```
        SmsManager.getDefault().sendTextMessage(number, null,
messageToSend, null,null);
        break;

        case R.id.button2:
        // Code for backward 2 click
            String messageToSend1 = "backward";
            String number1 = "+24994377844";

            SmsManager.getDefault().sendTextMessage(number1, null,
messageToSend1, null,null);
            break;

        case R.id.button3:
        // Code for button 3 click
            String messageToSend2 = "left";
            String number2 = "+24994377844";

            SmsManager.getDefault().sendTextMessage(number2, null,
messageToSend2, null,null);
            break;

        case R.id.button4:
        // Code for button 4 click
            String messageToSend3 = "Right";
            String number3 = "+24994377844";

            SmsManager.getDefault().sendTextMessage(number3, null,
messageToSend3, null,null);
            break;
        case R.id.button5:
        // Code for button 5 click
            String messageToSend4 = "STOP";
            String number4 = "+24994377844";

            SmsManager.getDefault().sendTextMessage(number4, null,
messageToSend4, null,null);
```

```
        break;
    }
}

/* AUTO-GENERATED FILE. DO NOT MODIFY.
 *
 * This class was automatically generated by the
 * aapt tool from the resource data it found. It
 * should not be modified by hand.
 */
```

```
package gsm.app;
```

```
public final class R {
    public static final class attr {
    }
    public static final class dimen {
        /** Default screen margins, per the Android Design guidelines.
```

Customize dimensions originally defined in res/values/dimens.xml (such as screen margins) for sw720dp devices (e.g. 10" tablets) in landscape here.

```
    */  
  
    public static final int activity_horizontal_margin=0x7f040000;  
    public static final int activity_vertical_margin=0x7f040001;  
}  
public static final class drawable {  
    public static final int ic_launcher=0x7f020000;  
}  
public static final class id {  
    public static final int action_settings=0x7f080005;  
    public static final int button1=0x7f080003;  
    public static final int button2=0x7f080000;  
    public static final int button3=0x7f080001;  
    public static final int button4=0x7f080002;  
    public static final int button5=0x7f080004;  
}  
public static final class layout {  
    public static final int main=0x7f030000;  
}  
public static final class menu {  
    public static final int main=0x7f070000;  
}  
public static final class string {  
    public static final int action_settings=0x7f050001;
```

```
public static final int app_name=0x7f050000;
public static final int hello_world=0x7f050002;
}
public static final class style {
    /**
    Base application theme, dependent on API level. This theme is replaced
    by AppBaseTheme from res/values-vXX/styles.xml on newer devices.
```

Theme customizations available in newer API levels can go in res/values-vXX/styles.xml, while customizations related to backward-compatibility can go here.

Base application theme for API 11+. This theme completely replaces AppBaseTheme from res/values/styles.xml on API 11+ devices.

API 11 theme customizations can go here.

Base application theme for API 14+. This theme completely replaces AppBaseTheme from BOTH res/values/styles.xml and res/values-v11/styles.xml on API 14+ devices.

API 14 theme customizations can go here.

```
*/  
  
public static final int AppBaseTheme=0x7f060000;  
  
/** Application theme.
```

All customizations that are NOT specific to a particular API-level can go here.

```
*/  
  
public static final int AppTheme=0x7f060001;  
  
}  
  
}
```

Appendix B

Linkit one code

```
#include <LGSM.h>

int i=0;

void setup()
{
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  Serial.println("Stop");
  digitalWrite(13, LOW);
  digitalWrite(12, LOW);
  digitalWrite(11, LOW);
  digitalWrite(10, H
char msg[100]=" ";
char buf[20];
int v;IGH);
  delay(1000);
  Serial.begin(9600);
  while(!LSMS.ready())
```



```
delay(100);
Serial.println("SIM ready for work!");
}
void loop()
{
if(LSMS.available()) // Check if there is new SMS
{
Serial.println("There is new message.");
LSMS.remoteNumber(buf, 20); // display Number part
Serial.print("Number:");
Serial.println(buf);
Serial.print("Content:"); // display Content part
while(true)
{
v = LSMS.read();
if(v < 0)
{

//Serial.println();
Serial.print(msg);
Serial.print(i);
if(strcmp(msg,"Forward")==0 )
{
```

```
Serial.println("Forward");  
digitalWrite(13, HIGH);  
digitalWrite(12, LOW);  
digitalWrite(11, LOW);  
digitalWrite(10, LOW);
```

```
Serial.println("SIM ready for work!");  
LSMS.beginSMS(buf);  
LSMS.print("LOW has been activated");  
LSMS.endSMS();
```

```
}
```

```
else if(strcmp(msg,"Backward")==0 )
```

```
{
```

```
Serial.println("Backward");  
digitalWrite(13, LOW);  
digitalWrite(12, HIGH);  
digitalWrite(11, LOW);  
digitalWrite(10, LOW);
```

```
Serial.println("SIM ready for work!");  
LSMS.beginSMS(buf);  
LSMS.print("backward has been activated");  
LSMS.endSMS();
```

```
}  
else if(strcmp(msg,"Left")==0 )  
{  
  Serial.println("Left");  
  digitalWrite(13, LOW);  
  digitalWrite(12, LOW);  
  digitalWrite(11, HIGH);  
  digitalWrite(10, LOW);  
  Serial.println("SIM ready for work!");  
  LSMS.beginSMS(buf);  
  LSMS.print("High has been activated");  
  LSMS.endSMS();  
  
}
```

```
else if(strcmp(msg,"Stop")==0 )  
{  
  Serial.println("Stop");  
  digitalWrite(13, LOW);  
  digitalWrite(12, LOW);  
  digitalWrite(11, LOW);
```

```
    digitalWrite(10, HIGH);
    Serial.println("SIM ready for work!");
    LSMS.beginSMS(buf);
    LSMS.print("STOP has been activated");
    LSMS.endSMS();

}

else if(strcmp(msg,"Right")==0 )
{
    Serial.println("Right");
    digitalWrite(9, HIGH);
    Serial.println("SIM ready for work!");
    LSMS.beginSMS(buf);
    LSMS.print("right on has been activated");
    LSMS.endSMS();

}

else
{
    Serial.println("SIM ready for work!");
```

```
LSMS.beginSMS(buf);
LSMS.print("No Match any Command");
LSMS.endSMS();
}

i=0;
break;
}
Serial.print((char)v);
if(v > 0)
msg[i]=v;
if(i<=100)
i=i+1;
}
Serial.println();
LSMS.flush(); // delete message
}
//delay(1000);
}
```

Simulation code

\$regfile = "m16def.dat"

\$crystal = 4000000

\$hwstack = 40

\$swstack = 16

\$framesize = 32

Dim A As Byte

Dim B As Byte

Dim C As Integer

Dim L As String * 7

Dim Key As Byte

Dim D As Byte

Config Porta = Output

Config Kbd = Portb , Debounce = 50

Config Lcd = 16 * 2

Config Lcdpin = Pin , Port = Portc , Rs = Portd.2 , E = Portd.3

Do

Q:

A = Getkbd()

If A > 15 Then

 Goto Q

Else

 Key = Lookup(a , Dat)

 If Key = 2 Then

 Locate 1 , 1

 Lcd "forword"

 Porta.0 = 0

 Porta.1 = 0

 Porta.4 = 1

 Porta.5 = 0

 Elseif Key = 8 Then

 Cls

 Locate 1 , 1

 Lcd "backword"

 Porta.0 = 0

 Porta.1 = 0

 Porta.4 = 0

Porta.5 = 1

Elseif Key = 4 Then

Cls

Locate 1 , 1

Lcd "left"

Porta.0 = 1

Porta.1 = 0

Porta.4 = 0

Porta.4 = 0

Elseif Key = 6 Then

Cls

Locate 1 , 1

Lcd "right"

Porta.0 = 0

Porta.1 = 1

Porta.4 = 0

Porta.4 = 0

Elseif Key = 5 Then

Cls

Locate 1 , 1

Lcd "STOP"

Porta.0 = 0

Porta.1 = 0

Porta.4 = 0

Porta.5 = 0

Elseif Key < 10 Then

Cls

Lcd "NOT valid"

Elseif Key = 10 Then

Cls

Lcd "NOT valid"

Elseif Key = 12 Then

Cls

Lcd "NOT valid"

L = Waitkey()

Print L

Lcd L

End If

End If

Loop

Dat:

Data 1 , 4 , 7 , 10 , 2 , 5 , 8 , 0 , 3 , 6 , 9 , 12