

# Мобильный центр управления

Использование канала GPRS позволяет управлять технологическим процессом производства посредством обычного мобильного телефона с поддержкой Java-приложений

Сергей Бадло,  
gaхр@mail.zp.ua

Полные исходные тексты и ресурсы тестового проекта



(C) Iconics

**И**дея использования мобильных систем для удаленного управления появилась уже достаточно давно. Но в основном дело ограничивалось только лишь отсылкой и приемом SMS-сообщений с кодом доступа (паролем) и минимумом информации. Чаще всего подобным приемом пользуются производители систем умных домов, которые способны самостоятельно информировать хозяина о происшествиях и составлять отчеты тревог, а он в свою очередь может оперативно воздействовать на систему, где бы он ни находился географически.

Слабой стороной данного метода является тот факт, что, в отличие от соединения через Интернет, SMS-сообщения не способны обеспечить необходимую интенсивность и скорость обмена данными. К тому же параллельно с ростом уровня функциональности мобильных аппаратов стоимость услуг доступа в Интернет с них же постоянно снижается. Сам собой напрашивается вывод – вместо SMS для управления удаленными системами автоматизации проще использовать интернет-доступ с мобильного телефона. Однако при этом нужно обеспечить достаточный уровень безопасности подключения.

## Безопасность и работа в реальном времени

Для надежной защиты сетей от несанкционированного доступа обычно применяют такие средства, как цифровая сертификация, шифрование на уровне защищенного сокета (SSL), определение уровней доступа и многое другое. Платформа Java позволяет встроить подобную защиту в каждое отдельное устройство управления, в том числе и в мобильный телефон. Встроенный в устройство сервер Java и Virtual Machine (или защищенный шлюз) ограничивают доступ к источнику информации. А поскольку весь трафик проходит через защищенный шлюз, то отсутствует прямой доступ к отдельному устройству управления, что полностью исключает возможность вывести их из строя. При этом вся информация поступает от безопасных веб-серверов, доступных при соответствующих правах доступа (безопасный HTTP, биометрическая идентификация, отпечатки пальцев и т. д.).

При внедрении системы управления на основе мобильного телефона (обязательно с поддержкой J2ME MIDP) в производство немаловажным аспектом является обеспечение его работы в реальном времени. Самое простое объ-

яснение термина «реальное время» сводится к тому, что время реакции системы управления на любое событие внутри и вне системы должно укладываться в некоторые границы, установленные приложением. Для реализации такого режима необходимо использовать уже не SMS, а GPRS/EDGE-соединение.

## Средства разработки мобильных приложений

Для создания приложений под мобильную платформу (так называемых мидлетов, midlet) применяется язык J2ME, который является урезанной версией языка Java. Непосредственно для работы с приложениями требуется комплект разработчика Java Development Kit (JDK, <http://java.sun.com/j2se/1.4.2/download.html>) и компилятор J2ME Wireless Toolkit от Sun (<http://java.sun.com/products/j2mewtoolkit>).

Особенностью данной среды является наличие подробной докумен-

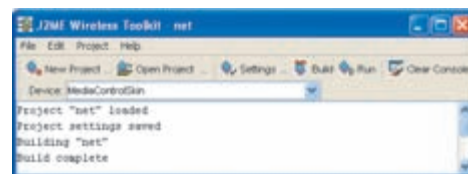


Рис. 1. Среда компиляции KToolbar Wireless Toolkit (WTK)

## ▼ Примеры листинга программы

## Подключаемые библиотеки и формы настроек:

```
package mobicq; //CLASSPATH

import javax.microedition.lcdui.*;
import java.io.*;
import javax.microedition.io.*;
import javax.microedition.midlet.*;
import java.util.*;
import javax.wireless.messaging.*;
import mobicq.tonePlayer;

public GtestCanvas (Mobicq midlet) {
this.midlet = midlet;
dpy = Display.getDisplay(midlet);
prev = dpy.getCurrent();

setFullScreenMode (true); //на полный экран
w = getWidth ();
h = getHeight ();
F = Font.getFont(Font.FACE_PROPORTIONAL,Font.STYLE_
BOLD,Font.SIZE_MEDIUM);
Fi = Font.getFont(Font.FACE_PROPORTIONAL,0,Font.SIZE_SMALL);
fh = F.getHeight();
fhi = Fi.getHeight();

addCommand(cmdAbout); //меню
addCommand(cmdOpt);
addCommand(cmdSend);
addCommand(cmdSMS);
addCommand(cmdL);
addCommand(cmdExit);
setCommandListener(this);
createImg (""+num);

//создание формы настроек-
opt = new Form ("Настройки");
host = new TextField ("Сервер:", "localhost", 20, TextField.ANY);
opt.append (host);
port = new TextField ("Порт:", "8764", 4, TextField.NUMERIC); opt.
append (port);
adr = new TextField ("Адрес MLink:", "3", 4, TextField.NUMERIC);
opt.append (adr);
ch1 = new TextField ("Номер канала:", "0", 4, TextField.NUMERIC);
opt.append (ch1);
ch2 = new TextField ("Номер канала:", "1", 4, TextField.NUMERIC);
opt.append (ch2);
si = new StringItem("Status:", " ");
opt.append(si);
opt.addCommand(backCommand);
opt.addCommand(startCommand);
opt.setCommandListener (this);

//создание формы sms-
sms = new Form ("Отправка SMS");
nsms = new TextField ("Номер:", "80", 13, TextField.NUMERIC);
sms.append (nsms);
tsms = new TextField ("Текст:", "", 140, TextField.ANY); sms.append
(tsms);
sms.addCommand(backCommand);
sms.addCommand(sendCommand);
sms.setCommandListener (this);

btrend(); //инициализируем буфер тренда
}

Декодировка пакета M-Link:

public void run() { //в потоке-
try {
sc = (SocketConnection) Connector.open("socket://
"+host2+"."+port2);
si.setText("connected"); st = "прием";//статус приема-
is = sc.openInputStream();
os = sc.openOutputStream();
sender = new Sender(os);
while (true) {
StringBuffer sb = new StringBuffer();
int c = 0;
while (((c = is.read()) != '\n') && (c != -1)) {sb.append((char) c);}
if (c == -1) {break;}
si.setText(""+ sb.toString()); //tp.play(90);

//декодировка-
packet = sb.toString();
```

```
MLinkDecode (packet);
//log-
repaint();

}
stop();
si.setText("not connected"); st = "нет связи";
} catch (ConnectionNotFoundException cnfe) {
si.setText("not connected"); st = "нет связи";
} catch (IOException ioe) {
if (!stop) {
ioe.printStackTrace();
}
} catch (Exception e) {
e.printStackTrace();
}
}
}
```

```
//===== MLink
public void MLinkDecode (String in) {
//int l = in.length(); //<14
if (in.length() == 14) {
byte[] buf = new byte[13];
for (int i = 0; i<13; i++) {buf[i]= (byte) in.charAt(i);}
madr = buf[2];
mch = buf[4]+buf[5];
mval = buf[8]+buf[9]+buf[10]+buf[11];
}
}
//===== Mlink
```

**Отсылка sms:**

```
MessageConnection sms=null;
String address="sms://"+nsms2; //куда
sms=(MessageConnection) Connector.open(address);
TextMessage text=(TextMessage) sms.newMessage(MessageConne
ction.TEXT_MESSAGE); //сам текст
text.setAddress(address);
text.setPayloadText(tsms2);
sms.send(text);
```

**Графический тренд:**

```
//net-
private Form opt, sms;
String st = "нет связи";
String packet = "";
int madr, mch, mval, mval1, mval2; //параметры MLink
String tadr = "3";
String tch1 = "1";
String tch2 = "2";
int[] trend1, trend2;//тренд
int cnt = 0; //счетчик тренда
int mcnt = 5; //период тренда
```

```
private void btrend () { //буфер тренда
trend1 = new int[mcnt*60];
trend2 = new int[mcnt*60];
for (int i = 0; i<mcnt*60; i++) { trend1[i] = 0; trend2[i] = 0; }
}
```

```
private void gtrend (Graphics g, int[] trend, int osx, int zvet) { //
рисуем тренды-
for (int i = 0; i<mcnt*59; i++) {
g.setColor (zvet);
g.drawLine (i*w/60/mcnt, osx -
2- 30*trend[i]/100, (w/60/mcnt)+(i*w/60/mcnt), osx -2-
30*trend[i+1]/100);
}
}
```

```
protected void paint (Graphics g) { //собственно экран-
g.setColor (0,0,0);
g.fillRect (0,0,w,h); //неразумно все обновлять
g.drawImage (img,w/2,h/2,g.HCENTER|g.VCENTER);
```

```
tr++;
if (tr>360) {tr = 0;};
```

```
if (num==1) { //2- экран
g.setColor (0xffff);
g.drawString ("ТЕСТ ЭКРАНА",w/2,h/2-2-fhi,g.HCENTER|g.TOP);
g.drawString(getclock(false),w/2,h/2,g.HCENTER|g.TOP);
g.drawString(getclock(true),w/2,h/2+2+fhi,g.HCENTER|g.TOP);
```

```

//свойства-
Runtime runtime = Runtime.getRuntime(); runtime.gc();
long free = runtime.freeMemory(); long total = runtime.totalMemory();
g.drawString ("Free = " + free,w/2,0,g.LEFT|g.TOP);
g.drawString ("Total = " + total,w/2,2+fhi,g.LEFT|g.TOP);
//

g.drawString(oNum(tr),0,0,g.LEFT|g.TOP);

// Циферблат
int diameter= 100;
g.setStrokeStyle(Graphics.DOTTED);
g.drawArc(w/2-diameter/2, h/2-diameter/2, diameter, diameter, 0, 360);
g.setStrokeStyle(Graphics.SOLID);

//стрелка
int y=math.sin(tr);
int x=math.cos(tr);
g.drawLine (w/2,h/2,x,y);
g.setColor (255,255,255);
g.drawString ("для выхода #",w/2,h-2-2*fhi,g.HCENTER|g.TOP);
g.drawString (" - далее -",w/2,h-2-fhi,g.HCENTER|g.TOP);
}

if (num==2) { //3- экран
g.setColor (255,255,255);
g.drawString ("- комплексные проекты -",w/2,h-4-2*fhi,g.HCENTER|g.TOP);
g.drawString ("АСУТП",w/2,h-2-fhi,g.HCENTER|g.TOP);
}

if (num==0) { //1- экран (мнемосхема)
// регулятор громкости (на 2 строки, справа, соотношение 5:1)
g.setColor(0,255,0);
for (int i=0;i<vol/20;i++) g.fillRect(25+w-fhi*5+fhi*i/2,fhi*2-fhi*2*i/20-fhi/2,fhi/3,fhi*2*i/20+2);
if (vol<100) {g.setColor(170,150,170); for (int i=vol/20;i<5;i++) g.fillRect(25+w-fhi*5+fhi*i/2,fhi*2-fhi*2*i/20-fhi/2,fhi/3,fhi*2*i/20+2); }

//тест выборка каналов-
if (madr == 3) {
if (mch == 0) {mval1 = mval;}
if (mch == 1) {mval2 = mval;}}
//тренд - 1..2
trend1[cnt] = mval1;
trend2[cnt] = mval2;
cnt++; if (cnt>mcnt*59) {
cnt = 0;
btrend(); //буфер тренда
}

//текстовая информация-
g.setColor (255,255,255);
g.drawString ("тренд "+new String(""+madr)+"-"+new String(""+mch),0,h-fhi-1,g.LEFT|g.TOP);
g.drawString (st,w,h-fhi-1,g.RIGHT|g.TOP);
g.drawString ("Packet: "+packet+" (" +oNum(packet.length())+")",0,0,g.LEFT|g.TOP);
g.drawString ("To = "+oNum(tr),w/2-16,h/2-59,g.LEFT|g.TOP);
g.drawString ("Tk = "+oNum(mval1),w/2-16,h/2-43,g.LEFT|g.TOP);
g.drawString ("Ps = "+oNum(mval2),w/2-30,h/2+10,g.LEFT|g.TOP);

//маркер – визуализация движения
int osx = h-2*fhi;
g.setColor (150,150,150);
g.drawLine (0, osx-32, w, osx-32);
g.drawLine (0, osx, w, osx);
g.drawLine (cnt*w/60/mcnt, osx, (w/60/mcnt)+(cnt*w/60/mcnt), osx + 3);
g.drawLine (cnt*w/60/mcnt, osx-32, (w/60/mcnt)+(cnt*w/60/mcnt), osx -32- 3);
g.drawString (getclock(true),(w/60/mcnt)+(cnt*w/60/mcnt)+1, osx-32-3-fhi,g.HCENTER|g.TOP);

g.setColor (0xffff);
g.drawString (new String(""+mval1),(w/60/mcnt)+(cnt*w/60/mcnt)-fhi, osx+1,g.HCENTER|g.TOP);
g.setColor (255,255,255); //g.setColor (0xff1f);
g.drawString (new String(""+mval2),(w/60/mcnt)+(cnt*w/60/mcnt)+fhi, osx+1,g.HCENTER|g.TOP);

gtrend( g, trend1, osx, 0xffff);
gtrend( g, trend2, osx, 0xfffff);
}
}

```

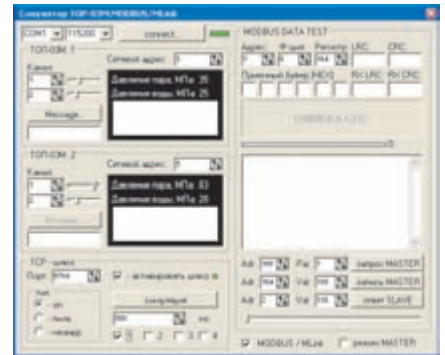


Рис. 2. Наглядность передачи данных и отладки обеспечивает симулятор – шлюз SOCKET

тации, подробных примеров и эмуляторов, что позволяет даже мало знакомому с языком Java человеку разрабатывать собственные приложения под любые мобильные устройства. Специфические классы, предоставляющие доступ к API конкретного телефона (Siemens, Motorola, Nokia, Samsung), рекомендуется по возможности не использовать. Для наглядности передачи данных и отладки можно использовать симулятор (рис. 2), который является параллельно шлюзом SOCKET для клиент-мидлета. Скорость обмена, а значит трафик и, как следствие, расходы на его оплату определяются сервером.

### Реализация программы

Итак, приступим к выполнению тестовой задачи – простой в реализации, но важной в процессе управления: организации доступа к архивам SCADA-системы с мобильного телефона. Для этого нам необходимо добавить экран мнемосхемы (на основе класса Canvas) в мобильный клиент, к примеру Jimm ([www.jimm.org](http://www.jimm.org)). Это делается следующим образом: в среде J2ME Wireless

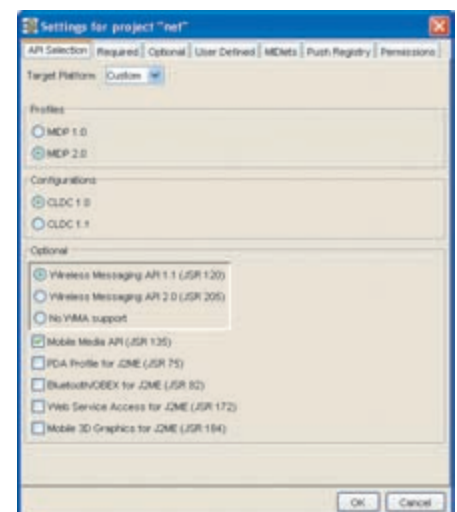


Рис. 3. Важный шаг создания мидлета – правильная настройка параметров проекта

### Требования к мобильному клиенту распределенной АСУ ТП

- ▶ Поддержка нескольких типов соединения для резервирования и повышения надежности: SMS, TCP/IP
- ▶ Возможность обратной связи для передачи команд (реакций) воздействия на систему
- ▶ Собственный протокол обмена пакетами данных
- ▶ Ограничение доступа по паролю
- ▶ Опции настройки протокола и типа соединения
- ▶ Индикация выбранных каналов в графической и/или текстовой формах
- ▶ Возможность архивирования принятых данных (не критично при использовании карт RS MMC)
- ▶ Минимизация использования специфичных API в мидлете.

Реализовать свой протокол поверх TCP/IP нетрудно, поскольку любое приложение в мобильном устройстве, поддерживающем GPRS/EDGE, может иметь доступ к сетевому протоколу. Это обеспечивается интерфейсом прикладных программ (Application Programming

Interface, API) телефона или КПК.

Условиями работоспособности подобной схемы соединения являются наличие у сервера выделенного IP-адреса и открытый соответствующий порт, либо же предлагается реализовать протокол обмена поверх существующих моделей связи, не привязываясь к IP. Очень заманчивыми в этом плане являются сервисы систем передачи сообщений через Интернет в реальном времени — ICQ или IRC, которые уже имеют свои выделенные серверы и позволяют, добавив в шлюз системы сервис ICQ, передавать значения определенному клиенту ICQ или по запросу. В таком случае разработчику достаточно добавить в телефон мобильную версию ICQ — например, Jimm ([www.jimm.org](http://www.jimm.org)) со своим экраном мнемосхемы. Этот вариант позволяет удешевить ИТ-инфраструктуру и упростить ее обслуживание, поскольку отпадает необходимость в собственном сервере для обмена данными.

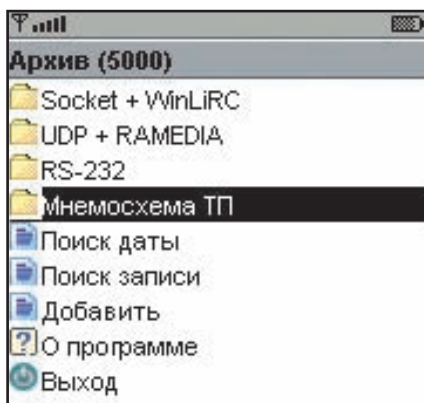


Рис. 4. Выбор нужного пункта меню в скомпилированном тестовом приложении, которое исполняется на мобильном устройстве

Toolkit (WTK) создается новый проект, затем в окне настроек необходимо установить параметры, как указано на рис. 3.

Поскольку за основу мобильного клиента берется свободно распространяемый исходный код Jimm, то остается лишь добавить в него свой класс (назовем его GtestCanvas), осуществ-

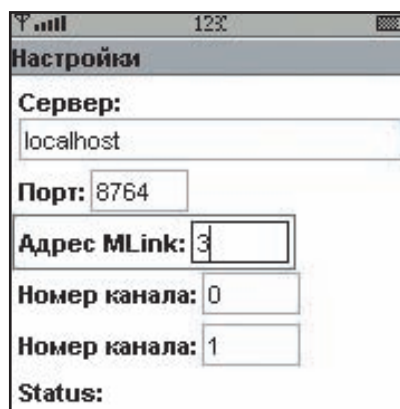


Рис. 5. Для осуществления связи в сети Ethernet или по каналу GPRS TCP/IP необходимо указать соответствующие настройки

ляющий указанные во вставке (см. вверху) требования. Полные листинги файлов смотрите на CD-MA, здесь же мы приводим лишь некоторые ключевые моменты: подключаемые библиотеки и формы настроек, декодировка пакета M-Link (M-Link — протокол, разработанный компанией «Адастра» ([www.adastra.ru](http://www.adastra.ru)), для связи по каналам RS-232/422/485, Dial-Up, GSM/SMS), отсылка SMS и графический тренд (см. во вставке на с. 64–65).

В результате компиляции и упаковки проекта (файлов классов и ресурсов) с помощью WTK или Jbuilder ([www.borland.com](http://www.borland.com), доступен после регистрации) создаются файлы .jar (собственно приложения) и .jad (параметры приложения). Скомпилированный тестовый клиент осуществляет связь в сети Ethernet или по каналу GPRS TCP/IP и отображает данные каналов на подложке мнемосхемы, фоном которой служит соответствующая мнемосхема технологического процесса.

Перечисленные принципы были воплощены в жизнь для получения данных о температуре и давлении отопительного котла в частном доме. Обязательным условием являлось наличие у котла канала связи с ПК. В данном случае настенный газовый котел Super Exclusive 24CAI на 24 кВт имеет интерфейс RS-232, который был соединен нуль-модемным кабелем с COM-портом обычного домашнего ПК. На второй COM-порт ПК

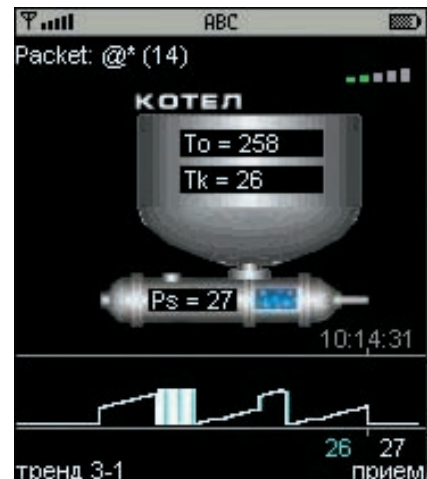


Рис. 6. Визуализация записи архивного тренда на экране мнемосхемы АСУ ТП на экране мобильного телефона

был подключен GSM/GPRS-модем TC-35 с установленной в него карточкой «Джинс». Канал TCP/IP был реализован через PPP (точка-точка), а в качестве мобильного устройства с установленным клиентом был применен телефон Nokia 6230.

Таким образом, используя современные средства коммуникации, можно получить доступ в реальном времени к любым архивным трендам или аварийным ситуациям с возможностью визуализации технологического процесса в системах АСУ, что позволит удаленно получать на АРМ оператора всю необходимую информацию. ■

### Ссылки

Тестовый мидлет клиент-сервер <http://ra-xp.narod.ru/zip/net.jar>  
 Видеоролик записи архивного тренда <http://ra-xp.narod.ru/zip/tr.rar>  
 Симулятор/шлюз IP-RS-232 (MLink) <http://ra-xp.narod.ru/zip/sim.zip>