Сергей Бадло
г. Запорожье
E-mail: **raxp@radioliga.com**

В статье рассмотрена методика создания и отправки кириллических сообщений с мобильных терминалов в формате PDU. В качестве управляющего устройства использован GSM-пейджер с дополнительным радиоканалом, собранный автором для нужд домашней сигнализации.

PDU – это просто. Система оповещения GSM

Введение

Область применения подобных систем поистине безгранична: от охранных функций имущества и автосигнализации, защиты и предупреждения взлома промышленных шкафов, контроля вагон-весов, передачи данных датчиков ПДФ, климатических параметров, аварийных сигналов предупреждения в серверных и многое другое… В то же время наблюдается тенденция роста количества систем с несколькими дублирующими каналами связи, как просто радиоканала, так и GSM.

Использовать для этих целей мобильные аппараты средней цены невыгодно, а вот "бросовые" – самое оно… Как правило в таких аппаратах доступен лишь PDU (Protocol Description Unit) формат отправки сообщений в UNICODE (UCS2) кодировке. Тут все вроде бы ничего: преобразуй в UCS2 и посылай… Но толковых готовых решений не оказалось, даже пресловутый PDUspy [1] не оправдал ожиданий по конвертации SMS.

На местном радиорынке готовые изделия GSM-сигнализаций, так называемых пейджеров, без самого телефона котируются в районе 34 USD. Сам же контроллер, "сердце" устройства, ATTiny2313 – 2 USD, разницу чувствуете? Зачем платить "чужому дяде"…

Краткий экскурс...

Недостатки просто GSM-канала связи очевидны: это и негарантированная доставка сообщений адресату и оплата карточки, т.е. зависимость от оператора. В тех случаях, когда требуется организовать канал связи на небольшие расстояния, но провода нежелательны, скажем, в движущихся системах, системах контроля температурного режима стенки ковша – целесообразно использование радиоканала. Можно, конечно, воспользоваться следующими решениями:

- Wi-Fi точка доступа - Wi-Fi клиенты;

- промышленные радиомодемы типа SST900 (семейства I-7000) - система сбора данных.

Но что делать, если это дорого и избыточно? Ведь "хорошая" точка доступа плюс клиентские карточки стоят недешево, а промышленные радиомодемы тем более. Появление на рынке доступных и простых в использовании интегральных трансмиттеров – по большей части снимает эту проблему.

Таким образом, можем сформировать основные требования к подобному устройству (интеллектуальному модулю):

- максимальное возможное количество дискретных входов - выходов;

- наличие управляющих выходов типа СК (сухой контакт) – релейный выход для внешних нагрузок;

- наличие канала GSM - связи;

Рис. 1. Система оповещения GSM в сборе

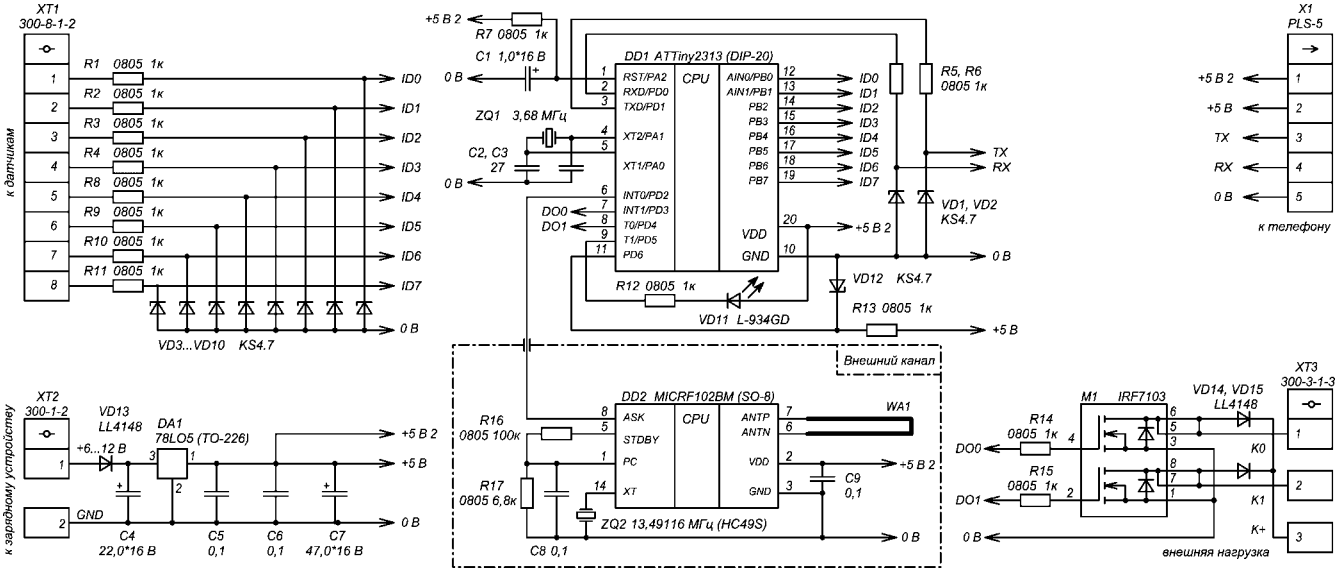

Рис. 2. Схема электрическая принципиальная интеллектуального модуля

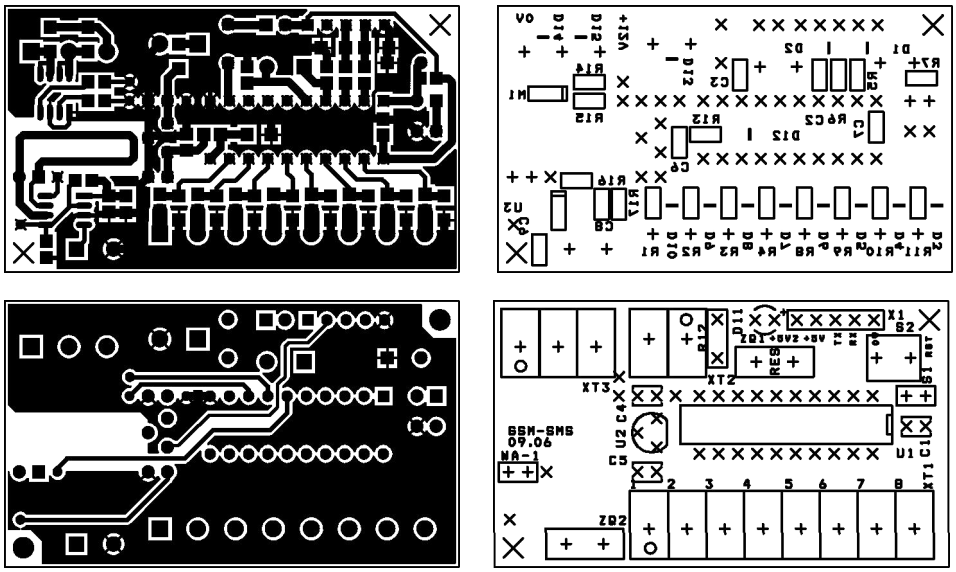

Рис. 3. Рисунки печатной платы

- наличие собственного канала передачи, не подлежащего лицензированию.

Первые два требования обеспечиваются функциональностью самого контроллера [2] и добавлением мощных выходных драйверов (ключей), а последние – любым GSM-модемом или мобильным телефоном и маломощным интегрированным передатчиком диапазона UHF (~433 МГц). Среди прочих трансмиттеров, наиболее приемлемыми по цене оказались радиомодули от фирмы MICREL. Отсутствие моточных навесных изделий особенно порадовало, главное – выдержать контурные данные антенны [3, 4]. Схема устройства представлена на **рис. 2**.

Интеллектуальный модуль имеет 8 входных программируемых каналов, к которым можно подключать:

- датчики затопления;
- ИК датчики движения;
- датчики пожарной сигнализации;
- датчики разбивания стекла;
- любое количество контактов, кнопок, герконов, включаемых последовательно или параллельно между собой для контроля открывания дверей, окон и т.д.

Модуль питается от аккумуляторов подключенного телефона и при пропадании внешнего электропитания передает SMS сообщение "Нет питания контроллера GSM".

Конструктив модуля

Плата [5, 6], габаритами 35х60 мм, разведена в пакете OrCad и изготовлена по "утюжно-лазерной" технологии (**рис. 3**). В качестве защитных стабилитронов подойдут любые малогабаритные в корпусе SOD-80 напряжением от 3,3 до 4,7 В. Для ZQ1 допускается использование любых кварцевых резонаторов типоразмера HC49 при условии коррекции прошивки для подстройки скорости обмена с телефоном и временных параметров работы устройства. При необходимости работы от внешней антенны на плате предусмотрен разъем NA-1 (на схеме не показан). В качестве выходных драйверов были использованы достаточно мощные ключи, но все же следует ограничиться нагрузкой до 1 А и 12 В. Подключение датчиков возможно между каждым из восьми входов XT1 и общим проводом устройства (к.2 – XT2).

Теория. Алгоритм формирования PDU

Прежде всего, для понимания процесса формирования сообщения необходимо изучить структуру PDU. Согласно стандарту ETSI (GSM 03.40), 2-х байтная кодировка используется для сообщений в UCS2*, максимальная длина которых ограничена 70-ю символами. Да, именно, вот откуда ограничение на длину кириллических SMS. Состав полей и алгоритм формирования PDU-SMS [7] виден из текста ниже. При этом, они включают не только текст сообщения, но и необходимую служебную информацию об отправителе и получателе, центре обслуживания SMS, тип сообщения и т.д. (**вставка 1**).

Формирование номера оправки и SMS центра осуществим по простому принципу: переставляем соседние **цифры** местами, если номер нечетный, то добавляем Fh… (**вставка 2**).

вставка 1

```
…
function cpdu(ksz,sz,num,msg: string): string;
begin
 result:= inttohex(length(sz),2) +    // 07=длина номера SMSC
       "91" +                         // 91=интернациональность номера SMSC
       cnv(ksz + sz) +                // 97103701F0=+790173100 с признак конца номера F
       "01" +                         // 01=PDU Type: MTI=01 -> исходящий SMS
       "00" +                         // 00=MR - параметр, устанавливается в SMSC
       inttohex(length(num)-1,2) +    // длина номера получателя-1
       "91" +                         // 91=интернациональность получателя
       cnv(num) +                     // номер получателя
       "00" +                         // 00=PID идентификатор номера протокола
       tcod +                         // 18/08=DCS схема кодирования: кириллическое
       inttohex(length(msg)*2,2)+     // длина сообщения
       ucs2(msg)                      // сообщение
end;
…
```

вставка 2

```
…
function cnv(s: string): string;
var i: integer;
begin
 delete(s,1,1);
 if length(s) mod 2>0 then s:=s + "F";
 i:= 1;   while (i<length(s)+1) do begin
  result:= result + s[i+1]+s[i];  inc(i);inc(i)
 end
end;
…
```

* формат однобайтовой кодировки SMS (максимальная длина сообщений 140 символов) в статье не рассматривается, ввиду отсутствия в ней поддержки кириллицы

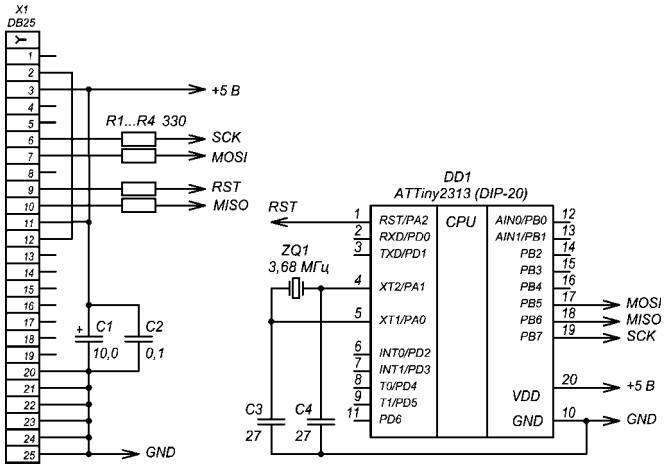

Рис. 4 Схема электрическая принципиальная программатора по LPT

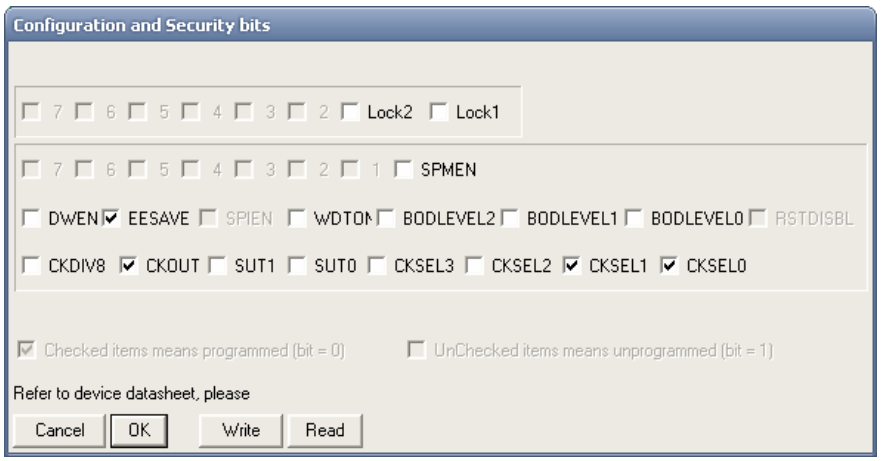

Рис. 6. Конфигурация “фьюзов” контроллера ATTiny

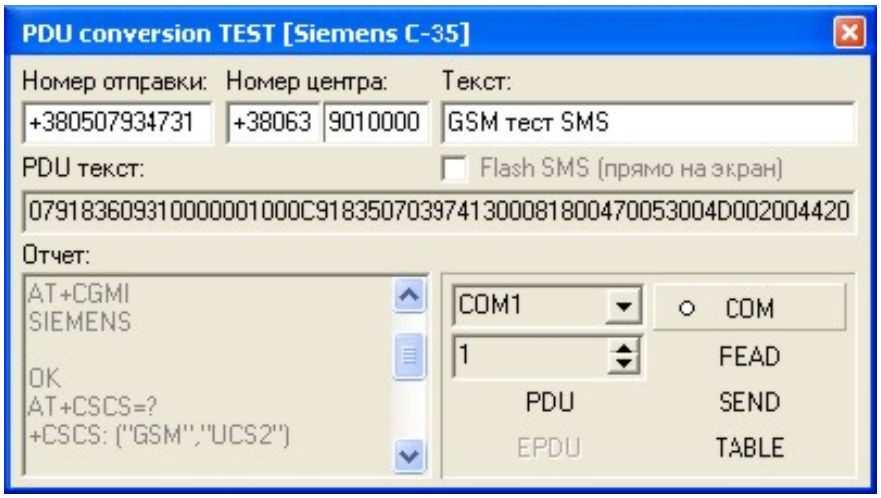

Рис. 7. Тестовый конвертор PDU

Сам текст сообщения в UCS2 в двухбайтном виде: для английских символов кодировки добавляется байт 00h, для русских – из байта вычитается C0h и прибавляется 410h… (**вставка 3**).

Практика. Средства отладки

Итак, приступим к основной задаче. Для работы нам следует запастись следующим:

Рис. 5. Конструктив программатора

Рис. 8. Принятая PDU посылка от ATTiny на экране телефона

- среда разработки aStudio4b460 [8];
- компилятор WinAVR-2005 [9];
- любой мобильный терминал-телефон, соединенный кабелем с COM-портом ПК, например б/у Siemens C-35;
- программа-программатор PonyProg 205 [10];
- простейший программатор по LPT (см. **рис. 4**, **рис. 5**).

Несмотря на свою простоту и навесной монтаж, данная схема прекрасно себя зарекомендовала при программировании большинства Atmel-ских микросхем. В качестве переходного шлейфа между коннектором X1 и панелькой

вставка 3

```
…
function ucs2(s: string): string;
var i,k: integer;
begin
 for i:=1 to length(s) do begin
  k:= ord(s[i]);
  // НЕПОСРЕДСТВЕННО преобразуем + 410h – C0h
  if k>=192 then k:= k + 1040 - 192;
  result:= result + inttohex(k,4)
 end
end;
…
```

Инициализация UART **вставка 4**

```
…
void UART_Init(void) {
        // устанавливаем скорость обмена
        UBRRH = (unsigned char) BAUD_19200 >> 8;
        UBRRL = (unsigned char) BAUD_19200;
        // включаем прерывания RxD/TxD
        UCSRB = (1<<RXCIE)|( 1 << RXEN )|( 1 << TXEN );
        sei();
}
…
```

Реализация временных задержек и "помаргивания" светодиода **вставка 5**

```
...
void wait_1ms(unsigned char ms) {
 while(ms) {
        short int i;
        for(i=0; i<500; i++) inp(PINB);
        ms—;
}}

void wait_1s(unsigned char sec) {
 while(sec) {
        wait_1ms(250);
        wait_1ms(250);
        wait_1ms(250);
        wait_1ms(250);
        sec—;
}
}

void DiagOut(unsigned char n,unsigned int time) {
 while(n>0) {
        sbi(PORTD,5); // установка бита
        wait_1ms(time);
        cbi(PORTD,5); // сброс бита
        wait_1ms(time);
        n—;
        }
 wait_1s(1);
}
…
```

программирования желательно использование экранированного кабеля, а все 6 проводников вести витыми парами. Например, можно воспользоваться куском кабеля UTP (S-FTP) подходящей длины.

Ввиду ограниченности места в журнале, рассмотрим только основные моменты реализации управляющей прошивки в среде WinAVR-2005 под контроллер ATTiny2313 (**вставки 4-6**).

Обратите внимание: чтобы правильно запрограммировать "фьюзы" на ATtiny нужно, согласно спецификации, выставить (для частоты кварца 4 МГц, см. **рис. 6**):

Lock2=1 (галочка снята)
Lock1=1
Spmen=1
Dwen=1
Eesave=0 (галочка стоит)
Spien=1
Wdtom=1
Bodlevel2=1
Bodlevel1=1
Bodlevel0=1
Rstdisbl=1
Ckdiv8=1
Ckout=0
Sut1=1
Sut0=1
Cksel3=1
Cksel2=1
Cksel1=0
Cksel0=0

Опрос вывода шлейфа и антидребезг **вставка 6**

```
…
if(bit_is_clear(PIND, 6)) { //ID6
 char c;
 for(c = 0; c<5; c++) {
                wait_1ms(40); // защита от дребезга
                if(bit_is_set(PIND, 6)) break;
}
 if(c == 10) { // все 10 раз шлейф был замкнут на землю
        DiagOut(5,100); // 5 раз по 100мс
        testSMS(8); // выборка текущего сообщения из массива и отсылка
        continue;
}
}
…
```

вставка 7

```
…
sendcom("AT+CMGS="+inttostr(length(txt.text)*2)+#13+#10);   // 0Dh
sendcom(pdu.Text+#26);                                      // символ конца строки 1Ah
…
```

При корректной заливке прошивки контроллер сразу готов к использованию.

Реализация посылки PDU

Итак, приступим к отправке кириллического сообщения.

Проверим, как работает посылка PDU из тестового приложения (см. **рис. 7**, **рис. 8**), при подключенном терминале-телефоне, используя вышеприведенные функции преобразования (**вставка 7**).

В итоге, имеем сформированное "100% отправляемое" кириллическое PDU сообщение (см. **рис. 6**).

Разберем его подробно (**вставка 8**).

Обратите внимание! При необходимости вывода сообщения непосредственно на экран телефона – используйте параметр режима кодирования данных DCS=18 (режим Flash-SMS).

вставка 8

AT+CMGS=54 - длина в десятичной системе
07 91 836093100000 01 00 0C 91 835050391443 00 08 36
041D043504420020043F043804420430043D0438044F0020043A043E043D04420440043E043B043B043504400430002000470053004D

06	для номера SMSC + 1 байт интернациональности
91	интернациональность SMSC
836093100000	номер sms центра плюс признак окончания номера (перестановка)
01	PDU Type: исходящий SMS
00	MR - параметр, который устанавливается в SMSC
0C	длина номера получателя-1 в hex формате
91	интернациональность получателя
835050391443	номер получателя (перестановка +380505934134)
00	PID идентификатор номера протокола
08	[18/08] = [на экран - Flash-SMS / вну] DCS 2-х байтная схема кодирования данных
36	длина текста сообщения!!! заметьте - это длина в hex (каждый символ по 2-байта)
...	текст сообщения

Заключение

Таким образом, зная алгоритм формирования сообщения, читателю не составит труда получить и обратное преобразование из формата PDU в текст. А значит, при необходимости иметь возможность читать SMS и соответственно реализовать обратную связь, например, выполнять определенные команды управления.

Полные исходные тексты управляющей прошивки (в среде WinAVR-2005), разводку платы (в GERBER-формате RS-274X, так и растровом виде) и ресурсы проекта (файл *pdu-gsm.zip*) вы можете загрузить с сайта нашего журнала:
http://www.radioliga.com (раздел “Программы”)
а также с сайта автора: **http://raxp.radioliga.com**
Вопросы и обсуждение: **http://raxp.radioliga.com/forum**

Ресурсы

1. Сторонний конвертор PDUspy (freeware) - http://raxp.radioliga.com/zip/PDUspy.zip
2. Спецификация контроллера ATTiny2313 - http://raxp.radioliga.com/zip/ATtiny2313.pdf
3. Спецификация трансмиттера mICRF102 - http://raxp.radioliga.com/zip/micrf102.pdf
4. Сайт производителя MICREL - http://www.micrel.com
5. Схема (OrCad) и разводка модуля (GERBER формат RS-274X) - http://raxp.radioliga.com/zip/gsm_pp.zip
6. Исходники тестового проекта системы - http://raxp.radioliga.com/zip/gsm_src.zip
7. Подробное описание формата SMS-PDU - http://raxp.raioliga.com/zip/sms-pdu.zip
8. Среда разработки aStudio4b460 - http://atmel.ru/Software/files/aStudio4b460.exe
9. Компилятор WinAVR-2005 - http://surfnet.dl.sourceforge.net/sourceforge/winavr/WinAVR-20050214-install.exe
10. Программа-программатор PonyProg 205 - http://www.lancos.com/e2p/V2_05/ponyprogV205a.zip