

Centrala alarmowa GSM

Jak wiadomo, wskaźnik przestępczości w naszym kraju nie należy do najniższych. Dlatego obywatele zabezpieczają się przed aktami wandalizmu i złodziejami na różne sposoby. Powszechnie jest stosowanie central alarmowych, które po naruszeniu linii zaczynają głośno wyć. Ma to za zadanie przede wszystkim spłoszyć intruza, ale w praktyce i tak zdąży on wyrządzić znaczące szkody. Alternatywą dla takich rozwiązań są ciche centrale alarmowe, dyskretnie powiadamiające właściciela o włamaniu. Wtedy jest możliwość powiadomienia odpowiednich służb i złapanie przestępców. Urządzenie tego typu jest przedstawione w artykule.

Opis układu

Schemat ideowy urządzenia został przedstawiony na **rysunku 1**. Budowa układu jest w zasadzie bardzo prosta, gdyż cała „inteligencja” zawarta jest w oprogramowaniu. Jako nośnik informacji o zdarzeniu wykorzystano krótkie wiadomości SMS. Cena SMS-ów w zależności od operatora wynosi kilka groszy, więc koszty pracy urządzenia są znikome. Centrala jest przystosowana do pracy i przetestowana z telefonem Siemens C35i. Można też wykorzystać inny model, jednakże użyty aparat musi obsługiwać komendy AT dotyczące wysyłania SMS-ów.

Sercem całego układu jest mikrokontroler AT90S2313, który pracuje z rezonatorem kwarcowym o częstotliwości 4MHz. Urządzenie zawiera kompletny zasilacz stabilizowany o napięciu wyjściowym 5V. Dlatego do złącza J4 można podłączyć bezpośrednio transformator lub na przykład akumulator samochodowy. Ważne, by napięcie źródła było większe od około 7V ze względu na spadki napięć na diodach prostowniczych i stabilizatorze. Kondensatory C9, C10, C11, C12 filtrują napięcie zasilające. Złącze J1 służy do programowania mikrokontrolera bezpośrednio w układzie. Takie rozwiązanie umożliwia zmianę wszelkich

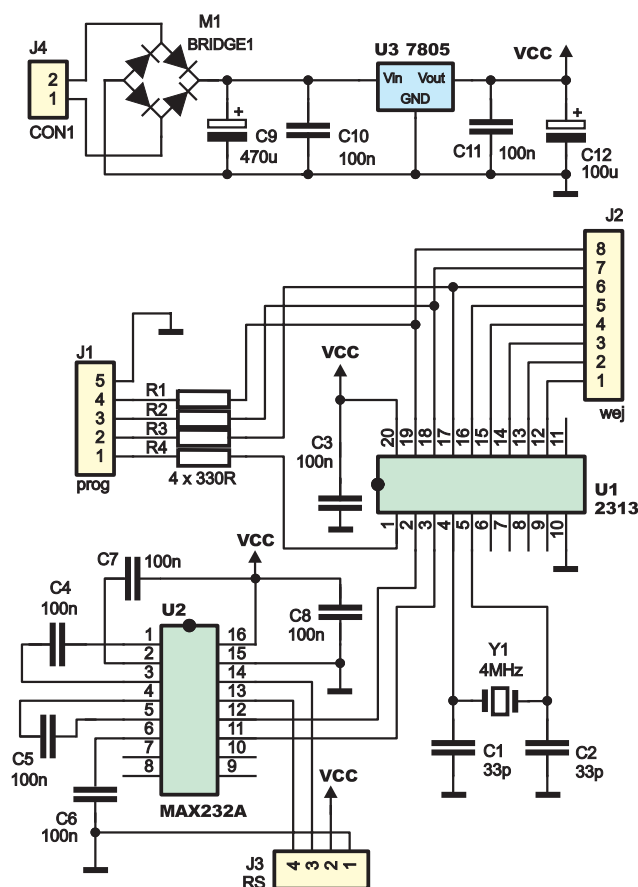
istotnych parametrów pracy centrali. Kondensator C3 ma za zadanie dodatkowo filtrować napięcie zasilające procesor i dlatego jest umieszczony bezpośrednio przy nim. Złącze J2 stanowi wejście centrali, do którego można podłączyć aż osiem linii monitorowanych. W stanie jałowym na monitorowanej linii musi występować potencjał masy. Zerwanie tego połączenia wywołuje alarm. Takie rozwiązanie umożliwia bezpośrednie podłączenie czujki ruchu, które zwykle mają wyjścia typu NC, czyli normalnie zwarte, stąd takie rozwiązanie układowe. Zastosowanie w podczuwanych czujnikach ruchu wyjść NC daje pewność, że nawet przy próbie przecięcia kabla zostanie wywołany alarm.

Do wymiany danych z telefonem został użyty konwerter poziomów MAX232A. Układ ten ma za zadanie dopasowanie napięć w standardzie TTL do standardów RS232. Zastosowanie kostki z oznaczeniem A pozwoliło zastąpić standardowe kondensatory elektrolityczne kondensatorami 100nF MKT. Do złącza J3 podłączamy telefon komórkowy, stosując do tego kabel przeznaczony do połączenia aparatu do komputera. Urządzenie komunikuje się z aparatem za pomocą komend AT.

Budowa SMS

Sposób szyfrowania SMS-a może wydawać się co najmniej dziwny, gdyż można odnieść wrażenie, że ktoś specjalnie utrudnił jego zapis. Jednak zrozumienie tego algorytmu nie jest rzeczą trudną. Cały komunikat jest przesyłany znakowo, w kodzie ASCII. Kompletny, zakodowany SMS ma następującą postać (kolory zostały dodane tylko dla łatwiejszej analizy dalszej części tekstu):

Rys. 1 Schemat ideowy



07918406103287F901000B918405897984F2
000011F7343DAC06D5F579FAFA7D77A7
D775

Jest to tekst: „witaj użytkowniku”, który zostanie wysłany na numer +48600123789, w sieci o numerze centrum wiadomości: +48509897482. Na pierwszy rzut oka jest to całkowicie bez sensu, bo przecież cyfry są dziwnie przestawiane, a tekst nie ma wiele wspólnego ze znakami ASCII. Zaraz postaram się to wyjaśnić: zestaw dwóch znaków np. 84, będziemy nazywać oktetem, ponieważ większość operacji kodowania odbywa się na właśnie takim zestawie.

Numery: centrum wiadomości i odbiorcy, są zakodowane w następujący sposób. Jeżeli przykładowy numer ma postać 48600123789 to najpierw rozdzielamy go na pary cyfr, ale licząc od lewej strony, czyli: 48 60 01 23 78 9F. Literkę F dodajemy na końcu numeru zawsze, gdy ma on nieparzystą liczbę cyfr. Teraz należy dokonać operacji zamiany miejscami w poszczególnych oktetach. Zakodowany numer będzie miał więc ostatecznie postać: 84 06 01 32 87 F9.

Pierwszy oktet komunikatu, czyli 07, to liczba oktetów numeru centrum wiadomości plus pole typu numeracji, czyli 91. Umieszczone dalej wartości 0100 są z reguły stałe i nie ulegają zmianie. Oktet 0B określa długość numeru odbiorcy, ale tym razem podana ona jest jako liczba cyfr tegoż numeru. Jest ich w sumie 11, stąd wartość w kodzie szesnastkowym 0B. Kolejne dwa oktety 00 00 również są stałe. Wartość 11 określa w kodzie szesnast-

kowym (11h czyli 17) liczbę liter komunikatu użytkownika, czyli to, co stanowi właściwą treść SMS-a.

Do zakodowania tekstu wiadomości wykorzystana jest zasada, że wszystkie znaki znajdujące się w tablicy ASCII są zapisane na siedmiu bitach. Wykorzystanie tego pozostałego bitu umożliwia kompresję komunikatu o około 14 – 15%. Schemat kodowania jest przedstawiony na **rysunku 2**. Algorytm postępowania jest następujący. Jeżeli do spakowania mamy na przykład wyraz „witaj”, to najpierw bierzemy dwie pierwsze litery, czyli „w” oraz „i”, a następnie wstawiamy bit z ostatniej pozycji drugiego znaku w miejsce najbardziej znaczącego bitu pierwszego. Drugim etapem będzie teraz przesunięcie w prawo o jedną pozycję kodu litery „i”. Uzyskamy w ten sposób dwie wolne pozycje w utworzonym bajcie. Teraz wstawiamy w nie dwa najmniej znaczące bity z następnego znaku (w naszym przypadku będzie to „t”), a następnie przesuwamy w prawo o dwie pozycje. Identycznie postępujemy dla następnych liter. Należy jedynie pamiętać, że co ósmy znak będzie opuszczany.

Program

Program sterujący został napisany w języku C, przy użyciu darmowego kompilatora AVR – GCC. Zamieszczenie całego listingu zajęłoby zbyt dużo miejsca, dlatego postaram się omówić najważniejsze jego fragmenty. Wszystkie funkcje zostały napisane praktycznie od zera, należą do nich na przykład obsługa sprzętowego UARTa czy też funkcja obli-

czająca długość łańcucha znaków. Po włączeniu zasilania i odczekaniu około 4 sekund, centrala wysłała do uprawnionego użytkownika SMS-a o treści: „centrala w1”, co sygnalizuje włączenie centrali. Następnie urządzenie

Listing 1

```
while(1)
{
while((PINB & _BV(7)) == 0);
wyslijsms("ALARM !!!");
for(j = 0; j < 6; ++j)
delay_ms(5000);
}
```

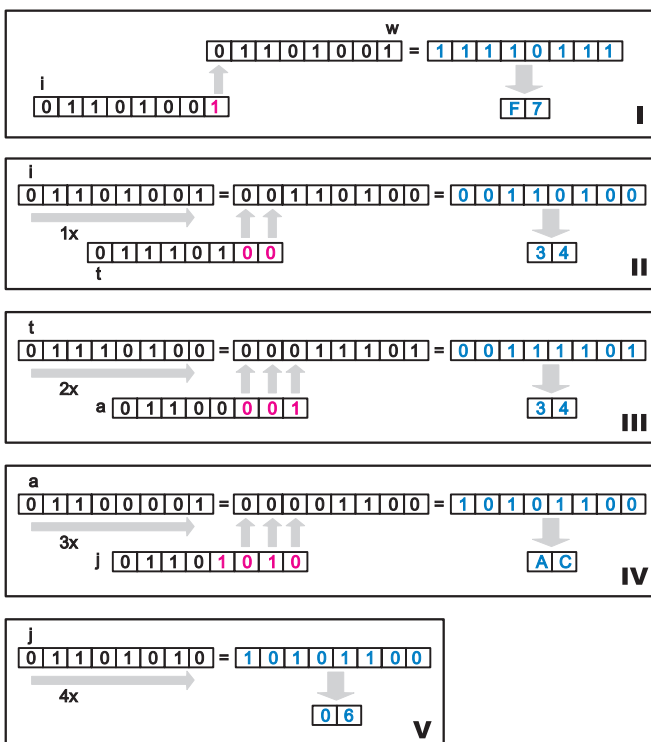
Listing 2

```
uart_putstring(CEN);
uart_putchar(0x0D);
while( (temp = uart_getchar()) != '+' );
for(j = 0; j < 12; ++j)
{
NRCEN[j] = uart_getchar();
}
for(j = 0; j < 12; j += 2)
{
temp = NRCEN[j];
NRCEN[j] = NRCEN[j + 1];
NRCEN[j + 1] = temp;
}
NRCEN[10] = 'F';
delay_ms(200);
```

Listing 3

```
uart_putstring(wyssms);
b = n / 10;
uart_putchar( b + '0' );
b = n % 10;
uart_putchar( b + '0' );
uart_putchar(0x0D);
while((c = uart_getchar()) != '>');
delay_ms(100);
uart_putstring(sms);
uart_putstring(NRCEN);
uart_putstring(smscd);
uart_putstring(NR);
uart_putstring("0000");
b = (dlugosc & 0xF0) >> 4;
uart_putchar( (b < 10) ? (b + '0') : ((b - 10) + 'A') );
b = (dlugosc & 0x0F);
uart_putchar( (b < 10) ? (b + '0') : ((b - 10) + 'A') );
i = 0;
while(i < dlugosc)
{
znak1 = *(text + i);
znak2 = *(text + i + 1);
n = i % 8;
znak1 = znak1 >> n;
n = 7 - n;
znak2 = znak2 << n;
znak1 = znak1 + znak2;
b = (znak1 & 0xF0) >> 4;
uart_putchar( (b < 10) ? (b + '0') : ((b - 10) + 'A') );
b = (znak1 & 0x0F);
uart_putchar( (b < 10) ? (b + '0') : ((b - 10) + 'A') );
i++;
if( n == 1 )
i++;
}
```

Rys. 2



Zakodowany wyraz „witaj” ma więc postać: F7343DAC06

czeka jeszcze nieco ponad minutę, aż czujka ruchu się skalibruje.

Przez większość czasu procesor będzie się kręcił w pętli, sprawdzając nieustannie stan monitorowanych linii. Zostało to zrealizowane za pomocą pętli while() przedstawionej na **listingu 1**.

W pokazanym przykładzie jest sprawdzany stan linii 7. Pętla jest wykonywana dopóty, dopóki linia 7 ma wartość 0, czyli jest na potencjale masy. Jeśli wystąpi stan alarmowy, to program przechodzi do funkcji wysyłającej SMS o treści ALARM !!! Następnie odczeka 30 sekund i znowu powraca do pętli monitorującej stan nadzorowanej linii. Oczywiście nic nie stoi na przeszkodzie, aby dopisać parę linii kodu i tym samym wykorzystać wszystkie możliwości prezentowanego układu.

Procesor po rozpoczęciu pracy najpierw pobiera z telefonu numer centrum wiadomości danej sieci i docelowy numer odbiorcy. Ten ostatni musi być zapisany na pierwszej pozycji karty SIM koniecznym jest prefiksem +48.

Procedury pobierania numerów centrum i odbiorcy są przedstawione na **listingu 2**.

Najpierw jest pobierany numer centrum wiadomości. Procesor wysyła do telefonu żądanie o podanie tego numeru. Jest to komenda AT+CSCA? reprezentowana przez przypis CEN. Następnie urządzenie czeka, aż odebrany znak będzie znakiem „4”, po czym zapisuje do tablicy NRCEN[] jedenaście następujących po sobie cyfr, wykonując przy tym odpowiednie kodowanie, które zostało omówione przy opisie budowy SMS-a. Analogicznie jest pobierany i zapisywany numer odbiorcy.

Najważniejszą funkcją programu jest funkcja kodująca i wysyłająca SMS, fragment został przedstawiony na **listingu 3**. Jej składnia może wydawać się nieco dziwna, a to dlatego, że sposób wystawiania znaków jest dość niecodzienny, ale zacznijmy od początku. Najpierw obliczana jest długość tekstu użytkownika, a później długość całego komunikatu. Aby wysłać SMS-a, należy przesłać do telefonu komendę AT+CMGS=n, gdzie n to wspomniana wyżej druga obliczona zmienna. Wystawienie tej zmiennej wymaga nieco szerszego komentarza. Aby wystawić daną liczbę znakowo, tj. w kodzie ASCII, należy w pierwszej kolejności oddzielić cyfrę jedności od cyfry dziesiątek. Tę pierwszą uzyskujemy poprzez podzielenie bez reszty zmiennej n przez 10, podstawę systemu dziesiętnego. Po tej operacji w pomocniczej zmiennej b mamy już cyfrę dziesiątek, jednakże jest to liczba w formacie binarnym, a my potrzebujemy jej w kodzie

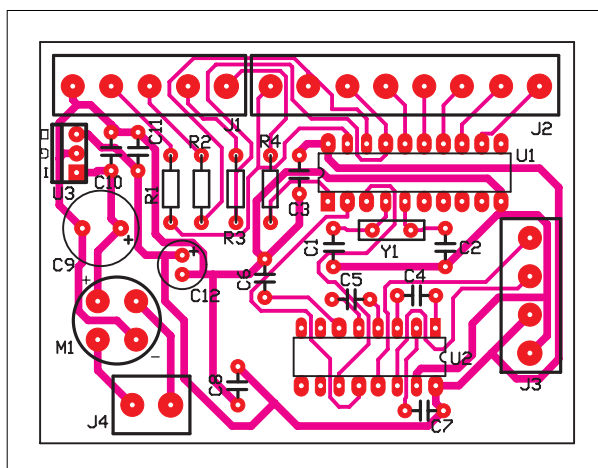
ASCII. Aby to osiągnąć, wykorzystano pewną właściwość tablicy ASCII, mianowicie wszystkie cyfry i litery są w niej umieszczone kolejno. Dlatego wystarczy jedynie do pierwszego znaku (na przykład zera) dodać interesującą nas wartość i już mamy odpowiedni kod danej cyfry... Procedura postępowania z cyfrą jedności różni się tylko tym, że występuje w niej dzielenie moduło.

Po wysłaniu tej komendy AT procesor czeka, aż telefon wyśle znak „zachęty”, którym jest „>” (**listing 3**). Następnie wysyła stałe części komunikatu oraz wymienione już numery centrum i odbiorcy. Po tej czynności mikrokontroler przechodzi do zakodowania długości tekstu użytkownika, tak aby uzyskać jej wartość w formie szesnastkowej. Na koniec następuje kodowanie samego tekstu według algorytmu przedstawionego w opisie budowy SMS-a.

Montaż i uruchomienie

Układ można zmontować na płytce drukowanej przedstawionej na **rysunku 3**. Montaż rozpoczynamy klasycznie, zaczynając od elementów o najmniejszych gabarytach, a kończąc na największych, w tym przypadku będą to kondensatory elektrolityczne. Zastosowanie podstawek pod układy scalone zapewne znacznie ułatwi ewentualny serwis urządzenia. Ja jednak czuję się pewniej, jeżeli wszystkie elementy są przylutowane do płytki bezpośrednio. Wybór pozostawiam Tobie, Czytelniku. Po zmontowaniu urządzenia należy podłączyć zasilanie i przystąpić do wgrania oprogramowania sterującego. Do tego celu można zastosować demonstracyjną wersję znanego pakietu BASCOM. Aby zaprogramować procesor należy w otwartym oknie BASCOMA kliknąć na przycisk programowania. W nowo otwartym oknie wybieramy *Otwórz* i szukamy w katalogu naszego projektu pliku z rozszerzeniem .hex. Po tych czynnościach wystarczy już tylko kliknąć na przycisk programowania układu i po chwili program znajdzie się w procesorze.

Rys. 3 Schemat montażowy



Do połączenia płytki centrali z komputerem można użyć na przykład kabla programującego z zestawu AVT – 3500.

Zmontowane poprawnie urządzenie pracuje od razu po włączeniu zasilania. Należy jedynie indywidualnie dobrać zwłoki czasowe, tak aby były one optymalne do danego zastosowania.

Całość należy jeszcze umieścić w odpowiedniej obudowie i zaopatrzyć we wtyk typu DB9 do podłączenia telefonu.

Możliwości zmian

Płytkę sterownika została tak zaprojektowana, aby możliwa była rozbudowa systemu lub też całkowita zmiana zasady jego działania. Po wgraniu odpowiedniego oprogramowania centrala alarmowa zamienia się w system zdalnego sterowania. Wykorzystać tu można zarówno SMS-y, jak i sygnały dzwonięcia. W tym drugim przypadku sterowanie jest możliwe bez ponoszenia dodatkowych kosztów. Widać więc, że możliwości układu są ogromne, a ograniczeniem może się tu stać tylko rozmiar pamięci FLASH mikrokontrolera, która wynosi niestety jedynie 2kB.

Krzysztof Paprocki
k_p@interia.pl

Od Redakcji. Należy pamiętać, że opisany system nie będzie działał ze wszystkimi telefonami komórkowymi – użyty telefon musi wykorzystywać typowe modemowe komendy AT. W literaturze dostępne są opisy podobnych urządzeń, a większość współpracuje z dość popularnym telefonem Siemens C35i. Na pewno urządzenie nie będzie współpracować z popularnymi telefonami Nokia i innymi, które mają zupełnie inny protokół komunikacyjny. Także nie wszystkie „siemensy” obsługują komendy AT.

Wykaz elementów

Rezystory	
R1-R4	330Ω
Kondensatory	
C1,C2	33pF
C12	100μF
C3-C8,C10,C11	100nF
C9	470μF
Półprzewodniki	
U1	AT90S2313
U2	MAX232A
U3	7805
M1	mostek prostowniczy
Inne	
J1-J4	złącza śrubowe
I3	DB9M
Y1	Q4NHz

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2777