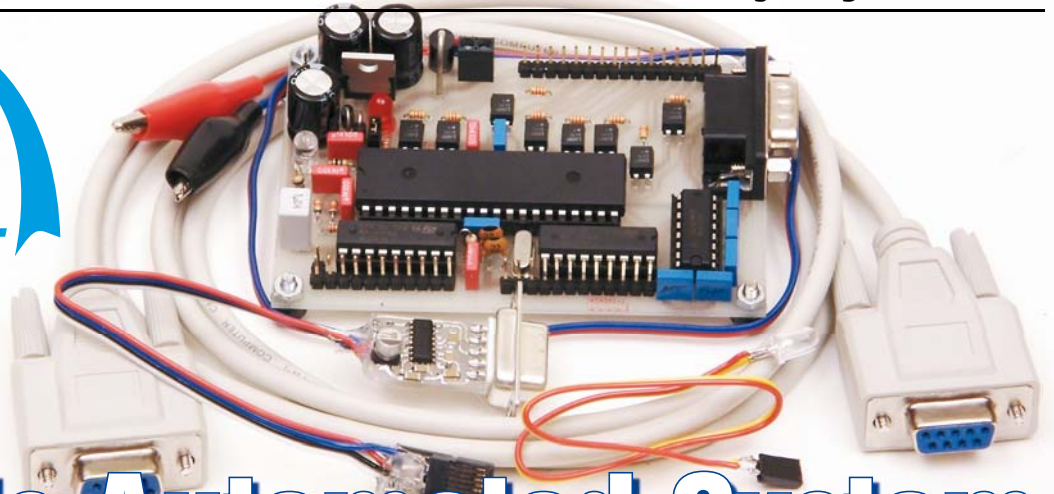




HAS –

House's Automated System



Czasami chciałoby się częścią nudnych obowiązków obarczyć kogoś innego, a samemu zająć sprawami bardziej twórczymi. Do takich właśnie celów jest przeznaczony domowy system automatyczny – HAS. Dzięki niemu, nie musimy samodzielnie zaświecać oświetlenia na noc ani włączać pewnych urządzeń na określony czas. To wszystko, i wiele więcej, może wykonać za nas HAS. Jednym poleceniem możesz określić, kiedy dane urządzenia mają się włączać. Na razie to tylko garstka z możliwości systemu tajemniczo nazwanego HAS.

A może chcielibyśmy mieć całkowitą kontrolę nad domem i wiedzieć, jaki jest stan czujników, np. czy drzwi frontowe są zamknięte. No dobrze, a jeśli chciałbym być powiadomiony o fakcie otwarcenia okna i przy okazji rabowania mojego domu, czy HAS to umożliwia – tak, może nawet powiadamiać o fakcie zalania piwnicy, a liczba zastosowań jest nieograniczona. Skoro już mogę sterować pewnymi urządzeniami i sprawdzać stan różnych czujników, czy też otrzymywać powiadomienie o zmianie ich stanu, to czy mój sąsiad też może? Będzie mógł, jeśli mu na to pozwoli: HAS pozwala na administrowanie użytkownikami, co więcej pozwala na zarządzanie ich uprawnieniami. Co to znaczy? Znaczy to tyle, że wyznaczeni użytkownicy mogą zmieniać/odczytywać stan pewnych wejść/wyjść, do których są dołączone czujniki, urządzenia sterowane, mogą też otrzymywać powiadomienie o zmianie stanu wejść. Nie oznacza to, że jeśli jeden użytkownik ma przypisane urządzenie, to inny nie może mieć tego samego, bo może.

Dzięki wspomnianym cechom HAS pomoże też oszczędzić energię.

Funkcje oprogramowania

Sterowanie wyjściami:

- Włączanie lub wyłączenie na stałe.
- Włączanie lub wyłączenie na noc.
- Włączanie lub wyłączenie na określony czas, możliwość powiązania tej funkcji ze zmrokiem, jednorazowo lub wielokrotnie. Co oznacza, że urządzenie może być włączone po zmroku na dany czas, lub wyłączone, tylko raz lub każdorazowo przy zdarzeniu zmroku.
- Cykliczne włączanie i wyłączenie. Możliwość określenia czasu włączenia i wyłączenia, jednorazowo lub w zapętleniu, istnieje możliwość powiązania tej funkcji z zejściem zmroku, jednokrotnie lub każdorazowo.
- Tygodniowy harmonogram. Możliwość określenia godziny włączenia i wyłączenia i wybrania dni, w których to obowiązuje.
- Możliwość sprawdzenia aktualnie ustawionych funkcji na poszczególnych wyjściach.

Sterowanie wejściami:

- Sprawdzanie stanu poszczególnych wejść.
- Możliwość ustawienia powiadomienia o zejściu zdarzenia na poszczególnych wejściach. Możliwe warianty to: zbocze narastające, opadające lub oba.
- Przy zarządzaniu stanem wyjść czy wejść istnieje możliwość podania ich listy w jednorazowym poleceniu.

Zarządzanie użytkownikami: dodawanie nowych i nadawanie im początkowych uprawnień, modyfikowanie uprawnień istniejących, usuwanie, wyświetlenie istniejących użytkowników i ich uprawnień.

Administracja systemem: logowanie za pomocą hasła, wylogowanie, zmiana hasła, ustawianie godziny i dnia tygodnia, wyświetlanie godziny i dnia tygodnia, wyświet-

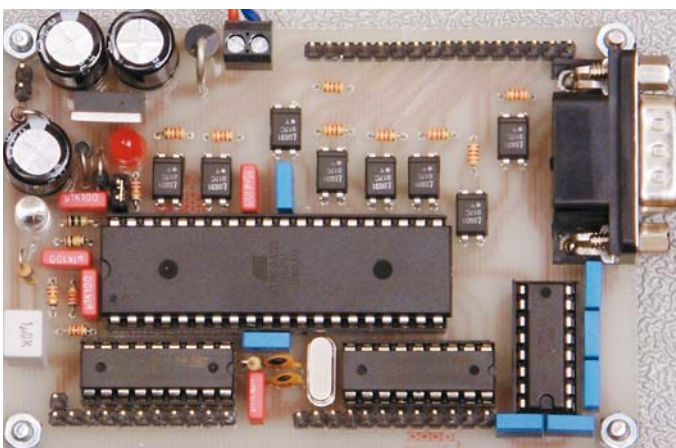
lanie pomocy, przełączenie do trybu automatycznego (praca z telefonem), przełączenie do trybu manualnego (praca w konsoli).

Cechy sprzętu: 16 wyjść, 8 wejść, zasilanie z zasilacza o napięciu stałym 6–15V, złącze umożliwiające dołączenie baterii awaryjnego zasilania, możliwość wyłączenia wyjść w przypadku zasilania awaryjnego, wejścia odizolowane galwanicznie, wyjścia umożliwiające bezpośrednio sterowanie przekaźnikami lub diodami transoptorów.

HAS może pracować zarówno z konsolą, jak i z telefonem komórkowym. Własny wiersz poleceń i, co ważne, pisane pełnymi zdaniami polecenia, umożliwiają współpracę HAS z dowolną platformą sprzętową i programową, jeśli tylko ma klienta umożliwiającego połączenie się przez port szeregowy. Możliwość pracy z telefonem – ułatwia to zdalne wykonywanie poleceń, co jest też możliwe i w przypadku pracy z konsolą, gdyż z komputerem (np. domowym) można połączyć się zdalnie. Ponieważ HAS przechowuje wszystkie instrukcje w pamięci i wykonuje je samodzielnie, nie jest potrzebny zewnętrzny program, HAS może zostać odłączony od hosta, a host wyłączony celem oszczędzania energii.

Opis układu

Schemat urządzenia jest widoczny na **rysunku 1**. Cała złożoność tego urządzenia kryje się w pamięci mikrokontrolera U3 ATmega32. Ostatecznie procesor ten ledwo udźwignął wymagania stawiane przez oprogramowanie, szczególnie w kwestii pamięciowej. Nie będe tu przytaczał kodu programu, bo zajęłoby on ok. 90 stron. Wspomniany procesor współpracuje z zewnętrznym rezonatorem kwarcowym X1 o częstotliwości 8MHz, obciążonym przez kondensatory C13, C14. Zastosowanie rezonatora o częstotliwości 8MHz było konieczne z uwagi na możliwość pracy na bateriach, w przypadku zaniku napięcia zasilania. Przy tej częstotliwości napięcie na trzech bateriach AA (R6) może spaść nawet do 3V. Sam mikrokontroler zresetuje się przy 2,7V, ponieważ tak został skonfigurowany



jego moduł BOD. Pozostałe 0,2V–0,3V jest tracone na diodzie Schottky'ego D5, przełączającej między pracą na bateriach a zasilaniem ze stabilizatora. Dodatkowa dioda została dodana tylko po to, by zapobiec przepływowi prądu z baterii do wyjścia stabilizatora. Jeśli już mowa o stabilizatorze, to jest nim stabilizator LDO (Low Dropout Voltage) U5 LM2940CT-5.0 – zapewnia on niewielki spadek napięcia mniejszy niż 0,5V, zwłaszcza przy małym obciążeniu. To powoduje, że nasz HAS może być zasilany już napięciem 6V, a nawet 5,5V jeśli jest to stabilne źródło. Kondensatory C16–C18 wygładzają tętnienia i stanowią magazyn energii. Dodatkowa dioda D6 chroni nasz układ przed odwrotnym podłączeniem napięcia zasilania, a D7 przed odwrotnym lub zbyt wysokim napięciem baterii. Dodatkowy obwód R15, C12, zapewni właściwe zresetowanie mikrokontrolera po włączeniu napięcia zasilania.

W celu zapewnienia odpowiedniej obciążalności prądowej wyjść sterujących peryferiami zastosowano tzw. matryce tranzystorów Darlingtona – U1, U2. Tranzystory w niej zawarte połączone są w układ OE i umożliwiają sterowanie dość wysokimi prądami (do 500mA na wszystkie wyjścia) i przy znacznych napięciach (do 50V) odbiorników. Matryca zawiera również diody, które mogą tłumić indukujące się napięcie np. na przełącznikach, a wewnętrzne rezystancje ograniczające prąd baz tranzystorów umożliwiają dołączenie jej wprost do mikrokontrolera. Prąd ten może znacząco obciążać baterie w przypadku zasilania awaryjnego. Dlatego istnieje możliwość wyłączenia wyjść na czas zasilania bateryjnego (nie powoduje to zmiany zaprogramowanej funkcji). Można tego dokonać przez rozwarcie zworki J1 nawet w trakcie pracy urządzenia, ale uwaga – wyjęcie zworki celem włączenia wyjść już na pracy bateryjnej nie spowoduje ich wyłączenia, a włożenie włączenia. Sama realizacja tej funkcji polega na pomiarze napięcia zasilającego mikrokontroler. Jest ono doprowadzone przez obwód R16, R4, R6, C5 do wejścia ADC3 przetwornika analogowo/cyfrowego (ADC). R16 i R4 stanowi dzielnik napięcia na dwa, ogranicza to zakres napięć na wejściu multipleksera ADC. R4, C5 to dodatkowy filtr. Zasilanie samego przetwornika A/D doprowadza się do nóżki AVCC, jest ono odfiltrowane zalecanym filtrem dolnoprzepustowym, zbudowanym z elementów C10, L2. Nóżka AREF jest tak skonfigurowana, że pracuje jako wyjście, w którym występuje

napięcie aktualnie używanego źródła referencyjnego. Samo źródło referencyjne, którym może być napięcie 2,56V lub zasilające, jest w takiej konfiguracji dołączone do nóżki AREF przez wewnętrzny klucz o stosunkowo dużej rezystancji, więc konieczne jest zastosowanie obciążającego kondensatora C15. Napięcie referencyjne nie jest stałe i wynosi 2,56V w przypadku pomiaru napięcia zasilania lub ok. 5V w przypadku pomiaru natężenia światła. Co ważne, zastosowanie wymienionego dzielnika napięcia i źródła napięcia referencyjnego 2,56V jest niezbędne do pomiaru napięcia zasilającego mikrokontroler, gdyż jeśli byśmy mierzyli napięcie bezpośrednio zasilające mikroprocesor, to zmieniłoby się ono razem ze zmianą napięcia odniesienia.

Pomiar natężenia światła jest realizowany za pomocą fotodiody D2, zasilającej ją rezystora R1. Dodatkowy obwód R2, C6 to szeregowy bocznik, tłumiący ewentualne skoki napięcia przemiennego, mogącego się indukować w przewodach do fotodiody. R3, C7 to filtr, może on się przydać w przypadku światła z lamp wyładowczych. Wspomniane pomiary są wykonywane naprzemiennie, uśredniane, dodatkowo wprowadzono czas na ustabilizowanie się napięć w obwodach, po których wyniki pomiarów będą akceptowane.

Do wyrowadzeń U3 realizujących funkcję wejść dołączono transoptory, zapewniające izolację galwaniczną. Pozwala to na dołączenie do wejść czujników połączonych dość długimi przewodami, oczywiście w takim przypadku czujnik powinien mieć własne źródło zasilania. Napięcie wystarczające do wywołania stanu niskiego, między E–C tranzystora, to 2V, maksymalnie 40V. Dodatkowe rezystory podciągające kolektor tranzystorów w transoptorach nie są wymagane, bo tę funkcję realizuje sam procesor.

Dioda LED (D3) znajdująca się na płycie sygnalizuje pracę urządzenia. Świeci ona światłem ciągłym, zawsze w przypadku pracy z konsolą, a w trybie automatycznym może migać, tylko gdy zachodzi komunikacja z telefonem lub jest jej brak.

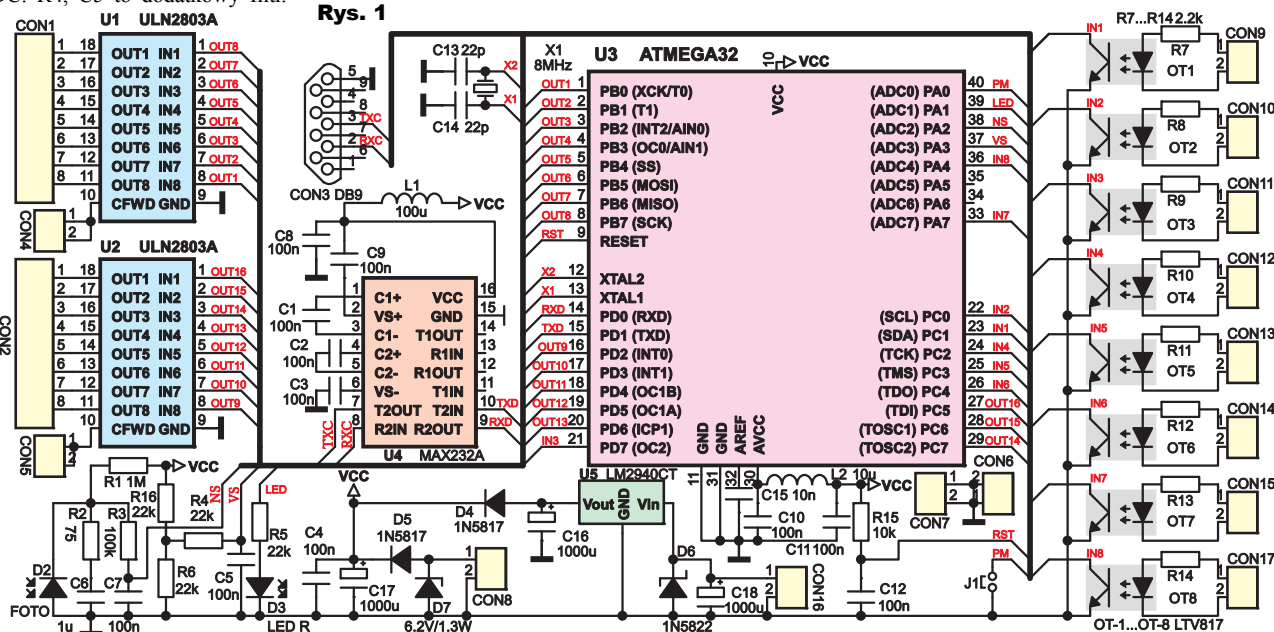
Ostatni obwód zapewniający komunikację standardem EIA/TIA-232 zbudowano w oparciu o MAX232A (U4) i C1–C3, C8, C9 i L1. Zastosowanie wersji układu z literą A na końcu umożliwia użycie kondensatorów (C1–C3, C9) wchodzących w skład generatora podnoszącego napięcie. Pozwala to na większe szybkości transmisji przez zapewnienie bardziej stromych zboczy przebiegu. Kondensatory C9, C11 to tzw. pętle prądowe, które zapewniają utrzymanie się odpowiedniego napięcia przy impulsowym poborze prądu. L1 ze wspomnianymi kondensatorami tworzy filtr eliminujący zakłócenia pochodzące od MAX232A.

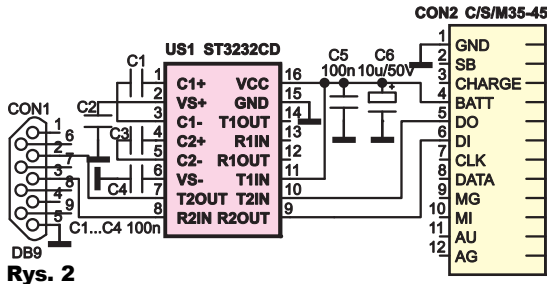
Przewód umożliwiający połączenie telefonu ze złączem RS-232 HAS został zbudowany w oparciu o ST3232 i przedstawiony jest na rysunku 2. Jest to także sterownik RS-232, lecz pracujący przy napięciu 3V. Napięcie zasilające wspomniany układ nie jest pobierane ze złącza portu RS, tylko z telefonu, który w swoim złączu ma wyprowadzenie umożliwiające zasilanie akcesoriów. Fakt ten uwalnia nas od konieczności ustawiania odpowiednich linii w porcie RS-232 w celu zasilenia drive-ra, dodatkową zaletą jest brak konieczności dostosowania poziomów napięcia.

Montaż i uruchomienie

Na rysunku 3 przedstawiony jest schemat montażowy płytki głównej, a na rysunku 4 – „kabela”. Montaż należy rozpocząć od wlutowania elementów najniższych. Ponieważ pod U3 znajdują się dwa dławiki, w niektórych podstawkach może zająć potrzeba wycięcia wewnętrznych wsporników, należy tego dokonać ostrożnie, tak aby nie połamać samej podstawki. Na U5 możemy zamontować niewielki radiator. Fotodiody możemy zamontować na niewielkim odcinku przewodu, konieczne musi to być przewód ekranowany, o długości nieprzekraczającej 2m. Osoby chcące samodzielnie przygotować procesor do pracy powinny: wgrać zawartość pliku HAS_FlashROM do pamięci Flash, następnie

Rys. 1





Rys. 2

wgrać zawartość pliku HAS_EEPROM do pamięci EEPROM i ustawić bity zabezpieczające, zgodnie z opisem w pliku FuseBits. Wykonanie wyżej wymienionych czynności w podanej kolejności jest niezbędne do prawidłowego działania HAS.

Uruchomienie. Urządzenie należy zasilać napięciem stałym 6-15V, niekoniecznie stabilizowanym o wydajności ok. 100mA. Sam układ nie powinien pobierać więcej jak 25mA w stanie wyłączenia wszystkich wyjść i 50mA w przypadku włączenia wszystkich wyjść. Możemy dołączyć dodatkowe baterie o napięciu 4,5V (3xR6). Do wyjść możemy dołączyć odbiorniki, np. przekaźniki, powinny one być zasilane z oddzielnego źródła zasilania i włączone między kolektor tranzystora (czyli konkretne wyjście) a dodatni potencjał ich zasilania, dodatkowo masa tego zasilania powinna być połączona z układem w miejscu GND. Diody tłumiące ewentualne przepięcia są wbudowane w matryce tranzystorów i możemy z nich skorzystać dzięki wyjściu FWCD.

Do wejść możemy dołączać różnego rodzaju czujniki w postaci przełączników czy oddzielnych układów. Na złącze J1 możemy założyć zworkę lub pozostawić rozwarte. Służy ono do wyboru zachowania się wyjść przy pracy na bateriach – w przypadku rozwarcia wyjścia zostaną wyłączone na czas pracy bateryjnej. Jeśli czułość fotodiody okazałaby się za duża, możemy zmniejszyć wartość R1, to samo zaleca się w przypadku połączenia jej z płytką długimi przewodami. Jeśli wszystko wydaje się poprawnie zmontowane i przygotowane, możemy włączyć HAS, dioda LED powinna świecić.

HAS należy połączyć z portem szeregowym (zgodnym ze standardem EIA/TIA-232-C) komputera lub serwera, można też łączyć go z terminalami np. VT-100. Klient umożliwiający komunikację po stronie hosta powinien obsługiwać ANSI ESCAPE CODE lub emulować VT-100. Zalecane programy to PuTTY dla Windows, minicom dla systemów UNIX-owych. W starszych wersjach Windowsa był dodawany program HyperTerminal, który też może być wykorzystany lub doinstalowany oddzielnie. HAS z hostem należy połączyć kablem odwróconym, z adapterem DB-9. Po włączeniu i połączeniu HAS-a z hostem, uruchamiamy klienta i przystępujemy do jego konfiguracji, ustawiamy: liczbę bitów na sekundę – 19 200, bity danych – 8, parzystość – brak,

bity stopu – 1, sterowanie przepływem – brak. **Rysunek 5** pokazuje główny ekran programu PuTTY z konfiguracją, wszystkie pozostałe ustawienia możemy pozostawić domyślne. Możemy zapisać nasze ustawienia, wpisując nazwę np. HAS w polu tekstowym Saved Session, i naciskając Save lub załadować, wybierając z listy i wciskając Load. Ostatecznie wciskamy Open, powinniśmy się połączyć z HAS-em by sprawdzić, czy odpowiada. Wciskamy Enter, w konsoli powinien pojawić się znak zachęty.

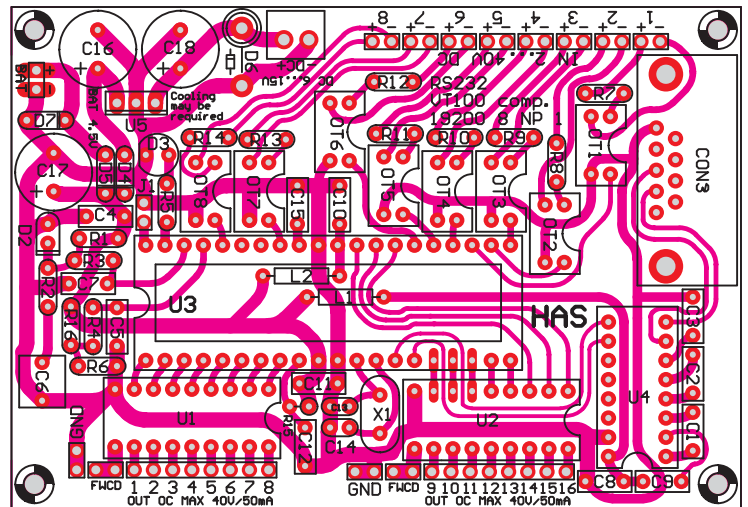
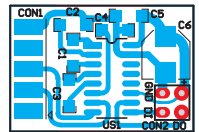
Obsługa urządzenia

Jeśli jesteśmy już połączony z HAS-em, to spróbujemy się załogować. Listę dostępnych poleceń i ich składnię możemy zobaczyć w diagramach syntaktycznych, które można ściągnąć z Elportalu. Można ją też wyświetlić poleceniem *help*, po załogowaniu. Logowania dokonujemy za pomocą polecenia *logon <hasło>*. Od razu zdradzę, że hasło domyślne to 1234, **rysunek 6** przedstawia wycinek konsoli, możemy zobaczyć ten proces, hasło nie jest ukrywane podczas wpisywania, jednak po wciśnięciu Enter konsola zostanie wyczyszczona (przewinięta) i hasło zniknie. Jeśli wszystko pójdzie zgodnie z planem, to zobaczymy napis OK lub ERROR w przypadku błędu. Jeśli pomylimy się i wpisujemy polecenie, którego nie ma lub nie mamy do niego uprawnień, to zostaniemy o tym poinformowani (*ERROR: Application permission denied/ERROR: Whoa! Application don't find*). Tak samo w przypadku podania zbyt dużej lub małej liczby parametrów (*ERROR: Too more/few params*) – powyższa zasada dotyczy też innych poleceń. Do wylogowania służy polecenie *logoff*. Mamy też możliwość zmiany hasła poleceniem *passwdmod*. Ja przesunąłem konsolę tak, żeby polecenia były widoczne, normalnie po ich wpisaniu konsola jest czyszczona. Po pierwszym załogowaniu wypadaloby ustawić zegar, możemy tego dokonać poleceniem *set*, przy okazji możemy też ustawić dzień tygodnia. Tym samym poleceniem można zmieniać te ustawienia. Jeśli się pomylimy, możemy usunąć błędne znaki klawiszem *BackSpace*. Usuwanie ze środka nie spowoduje przesunięcia

końca napisu, tylko „dziurę”, w którą możemy wstawić nowy znak, możemy także przesunąć kursor strzałkami. Wyświetlenia czasu dokonujemy poleceniem *time*.

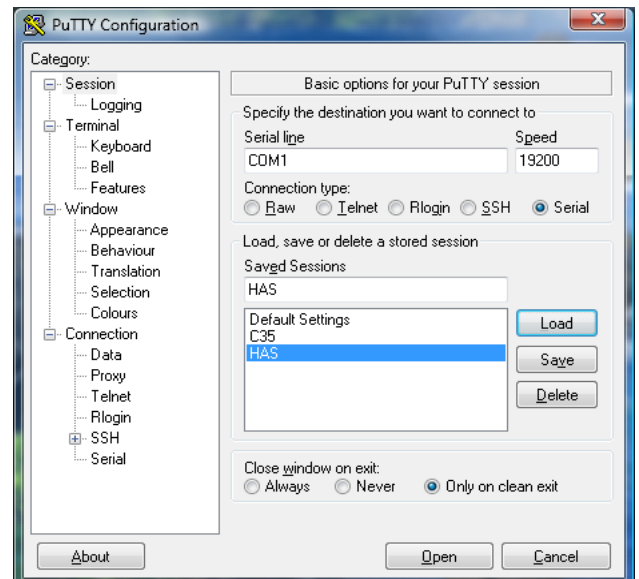
Przejdźmy teraz do poleceń sterujących wyjściami, poczynając od tych najprostszych. Polecenie *on* włączy dane wyjście na stałe, a *off* wyłącza na stałe. Dodatkowo możemy włączać/wyłączać wyjście na stałe w powiązaniu ze zdarzeniem nocy, przykłady pokazano na **rysunku 7**. Kolejnymi poleceniami przydatnymi przy czasowym sterowaniu wyjściami jest *timer* i *countdown*. Różnica między nimi polega na tym, że *timer* wyłącza, a *countdown* włącza wyjście na zadany czas. Dodatkowo możemy powiązać odliczanie czasu z zajęciem nocy, działanie tego przełączenia jest identyczne z działaniem wymienionych funkcji, lecz powoduje, że niejako zostały one napisane dopiero przy nastaniu pory nocnej. Dodatkowo możemy powtarzać tę komendę każdorazowo przy zajęciu nocy, przełączeniem *repeat*. Należy zwrócić szczególną uwagę na fakt, że *timer* powoduje tak jakby wyłączenie wyjścia na zadany czas i po tym czasie pozostawienie go włączonym. Dodatkowo ustawiając *timer*

Rys. 4



Rys. 3

Rys. 5



na noc, należy pamiętać o tym, że w czasie gdy on nie liczy, ustawia wyjście w stan włączenia, w przeciwieństwie do polecenia *countdown*, które wyłącza wyjście gdy nie zlicza. Jeśli zadany czas jest tak długi, a my ustawiliśmy wyzwalanie faktem nocy, to *timer/countdown* zacznie liczyć od pierwszej napotkanej nocy i skończy po danym czasie, nie zważając przy tym na to, czy jest dzień, czy noc. Przykłady użycia polecenia można zobaczyć na **rysunku 8**. Chyba najdłuższym poleceniem w sensie składni jest polecenie *cycle*.

Umożliwia ono ustawienie na zadanych wyjściach cyklu, czyli w sumie przebiegu o odpowiednim wypełnieniu. Domyślnie, bez żadnych przełączeń zostanie odliczony tylko jeden cykl. Cykl rozpoczynany jest od czasu wyłączenia, czyli wyjście początkowo zostanie wyłączone, po czym włączone na zadany czas i, co ważne, wyłączone po tym czasie. Możliwe jest zadanie powtarzania cyklu, wtedy uzyskamy coś w rodzaju generatora na danym wyjściu. Przełączenie *reverse* spowoduje zanegowanie wyjścia, którym steruje cykl. I tak pozostanie ono włączone przed rozpoczęciem lub po zakończeniu cyklu, tj. w czasie wyłączenia będzie włączone, w czasie włączenia – wyłączone. Dodatkowo, wcześniejsze operacje można powiązać ze stanem zapadnięcia zmroku i powtarzać każdorazowo, gdy on zachodzi. Tak samo jak w przypadku timerów i tu jeśli zadany czas jest tak długi, że kolejny raz zajdzie zdarzenie zmroku, to nie wpłynie to na aktualny czas odliczania cyklu, cykl musi się zakończyć. Jeśli ustawiono cykl nieskończony, to musi zostać odliczona całkowita liczba cykli, zanim możliwe będzie przejście wyjścia w stan statyczny – w dzień. Cykl może okazać się niezmiernie przydatną funkcją, np. jeśli chodzi o sterowanie zewnętrznymi lampkami choinkowymi nocą. **Rysunek 9** przedstawia przykłady tego polecenia, przy okazji w jednym z przykładowych poleceń „zjadłem” słówko *on*. I tu uwaga, maksymalna liczba znaków, jaka może zostać wpisana wynosi 97, więc w przypadku długich poleceń lepiej rozszerzyć konsolę do rozmiaru 100 znaków. Ostatnie polecenie z tej serii, *daily*, umożliwia ustawienie tygodniowego harmonogramu. Pozwala ono na określenie godziny, minuty, sekundy o jakiej wyjście ma być włączone i tak samo godziny, o której ma być wyłączone. Dodatkowo można określić, w jakie dni tygodnia akcja ta ma być wykonywana. Polecenie to jest ograniczone tylko do jednego przedziału czasowego dla całego dnia i wszystkich dni tygodnia. Warto wspomnieć, że ustawienie godziny wyłączenia mniejszej od włączenia też jest możliwe. Zmiana ustawień czasu poleceniem *set time*, podczas gdy wyjście już zostało włączone, nie spowodu-

```
> cycle 1 on 1s off 5s
OK
> cycle 4,5 on 1m 30s off 1h 15m 30s repeat
OK
> cycle 3 on 1d off 2h reverse
OK
> cycle 4-6 on 2m 30s off 15s repeat reverse
OK
> cycle 7 on 1d 6h off 1d repeat on night
OK
> cycle 8-9,16 30s off 20s repeat on night repeat
ERROR
> cycle 8-9,16 on 30s off 20s repeat on night repeat
OK
> cycle 1,5,7-11,2-4,15 on 999d 23h 59m 59s off 999d 23h 59m 59s repeat reverse on night repeat
OK
```

Rys. 9

Rys. 6

```
> logon 1234
OK
> logoff
OK
> logon 1235
ERROR
> logon 1234
OK
> set time 00:15:30 day 3
OK
> time
0:15:37 day 3
OK
```

Rys. 7

```
> on 2
OK
> on 3,5
OK
> on 8-11
OK
> off 2-3,5,8-11
OK
> on 2 if night
OK
> off 3,4 if night
OK
```

je przesunięcia pory wyłączenia do nowej

godziny – odliczenie odbędzie się wg czasu, jaki był w trakcie rozpoczynania odliczania. Dodatkowo, gdy nastawimy godzinę włączenia mniejszą niż aktualna tego samego dnia, kolejny cykl zacznie się dopiero za tydzień. Przykład można zobaczyć na **rysunku 10**. Ostatnim poleceniem związanym z obsługą wyjść jest polecenie *describe*. Umożliwia ono wyświetlenie aktualnie ustawionych funkcji na poszczególnych wyjściach – **rysunki 11 i 12**.

Inną, mniej obszerną częścią poleceń, są komendy związane z obsługą wejść. Możliwe jest wyświetlenie ich stanu, czyli czy wejście jest w stanie wysokim czy niskim, możemy to sprawdzić, wywołując polecenie *state*. Stan *up* oznacza, że podano wystarczające napięcie na dany transoptor, *down* – że nie podano. Dodatkowo istnieje możliwość powiadomienia o fackie zmiany stanu danego wejścia, można to ustawić za pomocą polecenia *event*. Sygnalizacja zmiany stanu może następować przy odpowiednim zboczach, dostępne opcje to *rise*, *fall*, *both*, można także ją wyłączyć, podając opcję *off*, przykład tych poleceń można zobaczyć na **rysunku 13**, dodatkowo wymusiłem zmianę stanu, by zobaczyć sposób wyświetlania zdarzenia. Zdarzenia, raz ustawione, pojawiają się w konsoli

Rys. 8

```
> timer 1,4,5-9 on 1d 44m
OK
> timer 3 on 1h if night
OK
> timer 3 on 999d 23h 59m 59s if night repeat
OK
> timer 1 on 20s if night repeat
OK
> countdown 2 on 30s if night repeat
OK
> countdown 3-7 on 1m 30s
OK
```

same, nie trzeba za każdym razem ich konfigurować poleceniem *event*.

Przejdźmy teraz do poleceń, które umożliwiają administrację użytkownikami. To za ich pomocą można będzie ich dodawać i zmieniać ich uprawnienia. Jako użytkownik w tym systemie jest rozumiany unikalny numer telefonu z nadanymi uprawnieniami. Praca z telefonem zostanie omówiona w dalszej kolejności. Aby dodać użytkownika, używamy polecenia *useradd*, po którym wpisujemy numer telefonu z prefiksem (dla polski 48), następnie możemy od razu nadać mu uprawnienia do ustawiania wyjść (+w), do odczytu stanu wejść (+r) lub otrzymywania informacji o zdarzeniach (+e), po tym przełączeniu występują numery urządzeń, tak jak w przypadku poznanych już poleceń *on*, *off*, *timer*; itd. W celu dodania, usunięcia, zmiany uprawnień danego użytkownika, używamy polecenia *usermod*, numeru telefonu wcześniej dodanego użytkownika, a także jednego z przełączeń (+|-#)(w|r|e). Znak plus oznacza, że dodajemy uprawnienia dla kolejnych wejść/wyjść/zdarzeń, znak minus – że usuwamy

```
> daily 1 on 5:00:00 off 6:30:00 day 1-7
OK
> daily 3-5,7 on 9:23:56 off 2:00:00 day 2,5,7
OK
> daily 2 on 23:00:00 off 22:00:00 day 4-7,3
OK
```

Rys. 10

Rys. 11

```
> describe 1-16
1: on - up
2: on if night - down
3: off if night - up
4: timer has been set for 30s, time has been deducted - up
5: countdown has been set for 1m, waiting for the night, on night - up
6: timer has been set for 20s, waiting for the night, on night, repeat - down
7: cycle on time: 1s, off time: 4s, time has been deducted - down
8: cycle on time: 30m, off time: 1h, now count 55m 49s remain, repeat on - down
9: cycle reverse on time: 10m, off time: 5m, now count 1m 24s remain, repeat on - up
10: cycle reverse on time: 5m, off time: 10m, now count 6m 39s remain - up
11: cycle on time: 7m, off time: 15m, waiting for the night, on night - down
12: cycle on time: 1m, off time: 2m, waiting for the night, on night, repeat - down
13: cycle on time: 10m, off time: 5m, waiting for the night, on night, repeat, repeat on - down
14: cycle reverse on time: 20m, off time: 10m, waiting for the night, on night, repeat on - up
15: off - down
16: daily on:9:0:0 off:17:0:0 days:3-7 - down
OK
```


uprawnienia z wyszczególnionych w ostatnim parametrze polecenia urządzeń, a znak kratki (#) – że zamieniamy uprawnienia na nowe. W szczególnych przypadkach, aby nadać jednemu użytkownikowi uprawnienia do ustawiania wyjść, odczytu wejść i powiadomień o zdarzeniach, polecenie *usermod* będzie trzeba wywołać przynajmniej trzy razy z różnymi przełączeniami tj. *r/w/e/*. Usunięcia użytkownika i jego uprawnień dokonujemy poleceniem *userrem*, a wypisania listy użytkowników i ich uprawnień poleceniem *users*. Szczegóły zawarte są na diagramach syntaktycznych, przykłady widoczne są na **rysunku 14**. Maksymalna liczba użytkowników została ograniczona do 10.

Praca z telefonem

Jak łatwo się domyślić, telefon dołączamy do złącza DB-9 HAS, tego samego, do którego zazwyczaj jest dołączona konsola. Liczba współpracujących modeli telefonów (z modemem AT) jest ograniczona, a jedyne sprawdzone modele to Siemens S45 i C35. Prawdopodobnie praca z modelami Siemens oscylującymi wokół tych będzie prawdopodobna, ale jeśli ktoś ma zamiar kupić taki telefon do celów współpracy z HAS-em, to niech lepiej wybierze jeden z modeli testowanych.

Jeśli już chcemy to zrobić na początku, wydajemy magiczne polecenie *auto*, odłączamy konsolę, a następnie podłączamy włączony i z wpisanym ewentualnym PIN-em telefon. Znajdująca się na płytce HAS dioda LED zacznie migać,

```
> state 1-3
1: up
2: up
3: down
OK
> event 2,5-7 on both
OK
> Event occur on device 7 has change state on fail edge.
> Event occur on device 6 has change state on fail edge.
> Event occur on device 5 has change state on fail edge.
> Event occur on device 2 has change state on fail edge.
> state 2,5-7
2: down
5: down
6: down
7: down
OK
> Event occur on device 7 has change state on rise edge.
> Event occur on device 6 has change state on rise edge.
> Event occur on device 5 has change state on rise edge.
> Event occur on device 2 has change state on rise edge.
> state 2,5-7
2: up
5: up
6: up
7: up
OK
```

Rys. 13

Rys. 12

```
> describe 1-16
1: on - up
2: on if night - up
3: off if night - down
4: timer has been set for 30s, time has been deducted - up
5: countdown has been set for 1m, now count 49s remain, on night - down
6: timer has been set for 20s, now count 9s remain, on night - up
7: cycle on time: 1s, off time: 4s, time has been deducted - down
8: cycle on time: 30m, off time: 1h, now count 53m 21s remain, repeat on - down
9: cycle reverse on time: 10m, off time: 5m, now count 8m 56s remain, repeat on - down
10: cycle reverse on time: 5m, off time: 10m, now count 4m 11s remain - up
11: cycle on time: 7m, off time: 15m, now count 14m 48s remain, on night - down
12: cycle on time: 1m, off time: 2m, now count 1m 48s remain, on night - down
13: cycle on time: 10m, off time: 5m, now count 4m 48s remain, on night, repeat on - down
14: cycle reverse on time: 20m, off time: 10m, now count 9m 48s remain, on night, repeat on - up
15: off - down
16: daily on:9:0:0 off:17:0:0 days:3-7 - down
OK
```

a po połączeniu telefonu po chwili powinna przestać. Sygnalizuje ona, że zachodzi komunikacja z telefonem lub jest jej brak. Przy dołączaniu telefonu należy uważać, w przypadku zapelnienia więcej niż 85% pamięci wiadomości SMS zostanie uruchomiony mechanizm Garbage Collector, który wykasuje wiadomości SMS. Mechanizm ten może się także uruchomić po dołączeniu, jeśli pamięć SMS się zapelni, zapobiega on „zapchaniu” się pamięci wiadomości, i tym samym niemożności odbioru poleceń. Przejęcia między pracą w konsoli a trybem pracy z telefonem i polecenia temu towarzyszące widoczne są na uproszczonym diagramie stanów – **rysunek 15**. W trybie automatycznym liczba poleceń jest ograniczona, możliwe polecenia to: *on*,

off, *timer*, *countdown*, *cycle*, *daily*, *state*. Oprócz tego dodatkowym ograniczeniem w wydawaniu poleceń jest lista użytkowników i ich uprawnień. Polecenia wydane jeszcze w trybie konsoli nie tracą mocy w trybie automatycznym, np. polecenie *event* trzeba wydać w trybie konsoli, by uprawnieni użytkownicy mogli otrzymać SMS o zdarzeniu. Należy pamiętać, że przypisując uprawnienia tym 10 użytkownikom do jakiegoś zdarzenia, które nastąpi, zostanie wysłane jedno-

```
> useradd 48123456789 +w 2,5,9-11
OK
> useradd 48111222333 +r 1-4,7
OK
> users
48123456789 w:2,5,9-11 r: e:
48111222333 w: r:1-4,7 e:
OK
> usermod 48123456789 +e 4,5
OK
> usermod 48123456789 -e 5
OK
> usermod 48123456789 #r 1-3
OK
> users
48123456789 w:2,5,9-11 r:1-3 e:4
48111222333 w: r:1-4,7 e:
OK
> userrem 48123456789
OK
> users
48111222333 w: r:1-4,7 e:
OK
```

Rys. 14

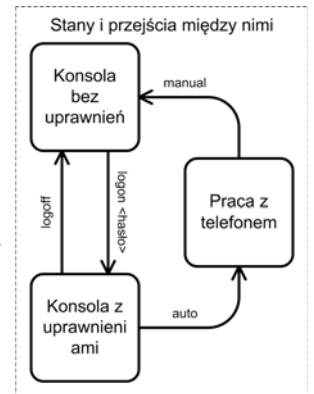
Wykaz elementów

| | | |
|-----------------------|---|--|
| Płyta główna | U3 | ATmega32-16PU |
| Rezystory | U4 | MAX232A |
| R1 | U5 | LM2940CT-5.0 |
| R2 | | Pozostałe |
| R3 | OT1-OT8 | LTV817 |
| R4-R6,R16 | X1 | .8MHz 4mm |
| R7-R14 | CON1,CON2,CON4-CON15, CON17,J1 | |
| R15 | złącze szpilkowe proste goldpin 1*40pin + 1*4pin do lamania | |
| Kondensatory | CON3 | wtyk D-SUB katowy do druku L=7,2mm (z bolcami) |
| C1-C5,C7-C12 | CON16 | ARK2/SM |
| C6 | | Podstawka standardowa 18pin 0,3" 2szt. |
| C13,C14 | | Podstawka standardowa 16pin 0,3" 1szt. |
| C15 | | Podstawka standardowa 40pin 0,6" 1szt. |
| C16-C18 | | Półprzewodniki |
| Półprzewodniki | D2 | fotodioda „Kabel” |
| D3 | D3 | LED R 5mm |
| D4,D5 | D4,D5 | 1N5817 |
| D6 | D6 | 1N5822 |
| D7 | D7 | Zenera 6,2V/1,3W |
| U1,U2 | U1,U2 | ULN2803A |
| | CON1 | złącze D-sub, gniazdo do lutowania przewodów |

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2946.

razowo 10 SMS-ów. Maksymalna liczba SMS-ów, jakie system może wysłać w ciągu minuty została ograniczona do 120 i zależy od szybkości z jaką telefon może je wysyłać. SMS-y

przychodzące od użytkowników niedodanych, niemających uprawnień, z błędnymi poleceniami będą odrzucane bez powiadomienia. System nie wysyła powiadomień także o wykonaniu danego polecenia. Jeśli na liście urządzeń polecenia jest choć jedno urządzenie, do którego użytkownik nie ma uprawnień, całe polecenie zostanie odrzucone. W przypadku polecenia *state* (uprawnienie +r), otrzymamy SMS zwrotny ze statusem wejść, jeśli nadano nam takie uprawnienia. Jeśli zająd problemy z komunikacją z telefonem, dioda LED miganiem zacznie to wskazywać po czasie najdłużej 20 minut. Przy wpisywaniu poleceń wielkość znaków nie ma znaczenia, a edytorzy telefonów lubią ją zmieniać. Wyjście z trybu automatycznego i praca w konsoli będzie możliwa po ponownym jej dołączeniu i wpisaniu komendy *manual*. Komendę tę należy poprzedzić paroma Enterami, jeśli nadal nie zobaczymy znaku zachęty, trzeba powtórzyć tę czynność.



Rys. 15

Szymon Janek
sx13@o2.pl