



Termometr SMS

Termometr to jedno z urządzeń, najczęściej konstruowanych przez hobbystów. Zbudowanie termometru przy użyciu mikrokontrolera i scalonego przetwornika temperatury jest bardzo proste, dlatego często takie konstrukcje stanowią pierwsze próby praktycznego zastosowania mikrokontrolera. Prezentowane urządzenie odróżnia się od innych termometrów sposobem komunikacji z użytkownikiem. Mianowicie **pozwała mierzyć i monitorować temperaturę na odległość, a jedynym ograniczeniem jest zasięg telefonii komórkowej**. Komunikacja z urządzeniem odbywa się za pomocą telefonu komórkowego z interfejsem modemowym AT (w praktyce starsze modele Siemens) – poprzez dzwonienie lub wysyłanie komend w wiadomościach SMS.

Aby odczytać temperaturę, należy zadzwonić na numer urządzenia. Połączenie zostaje przerwane i za chwilę dzwoniący otrzymuje wiadomość SMS z odczytem temperatury. Jednak po obniżeniu temperatury poniżej zadanej przez użytkownika, urządzenie automatycznie wysyła na wybraną liczbę numerów telefonów komórkowych wiadomości SMS z odpowiednią informacją. Urządzenie wysyła również powiadomienia o zaniku napięcia w sieci elektrycznej i wyzwoleniu alarmu (pojawieniu się napięcia na wejściu „Alarm”). Także zmiana konfiguracji urządzenia może odbywać się za pomocą wiadomości SMS.

Opis układu

Urządzenie składa się z dwóch części: telefonu komórkowego z modemem AT (Siemens C75) oraz sterownika opartego na mikrokontrolerze ATmega 8. Telefon ze sterownikiem połączone są za pomocą interfejsu UART. Czujnik temperatury to popularny układ scalony DS18B20.

Schemat ideowy urządzenia pokazany jest na **rysunku 1**. Jest to w zasadzie „goły” mikrokontroler z bezpośrednio wyprowadzonymi sygnałami. Jedynie wejście „Alarm” wpięte jest do mikrokontrolera nie bezpo-

średnio, ale poprzez mostek prostowniczy oraz transoptor. Uzyskuje się w ten sposób izolację optyczną, a dzięki zastosowaniu mostka prostowniczego nie trzeba zwracać uwagi na polaryzację przewodów wyzwalających, co znacznie ułatwia podłączenie urządzenia do np. centralki alarmowej. Wartość rezystora R2 jest tak dobrana, żeby przy zastosowaniu transoptora PC817 napięcie sygnalizujące alarm mogło zawierać się w przedziale 5V–24V. Przy zastosowaniu innego transoptora należy odpowiednio zmienić wartość rezystora R2.

Interfejs urządzenia jest tak prosty dlatego, że wszystko, co ważne, dzieje się w mikrokontrolerze, do którego program napisałem w języku BASCOM. Program zajmuje całą pamięć mikrokontrolera, jest w nim jednak kilka elementów, które można usunąć (na przykład programowa obsługa dodatkowego portu UART, którego używałem do debugowania), by uzyskać miejsce na dodatkową funkcjonalność.

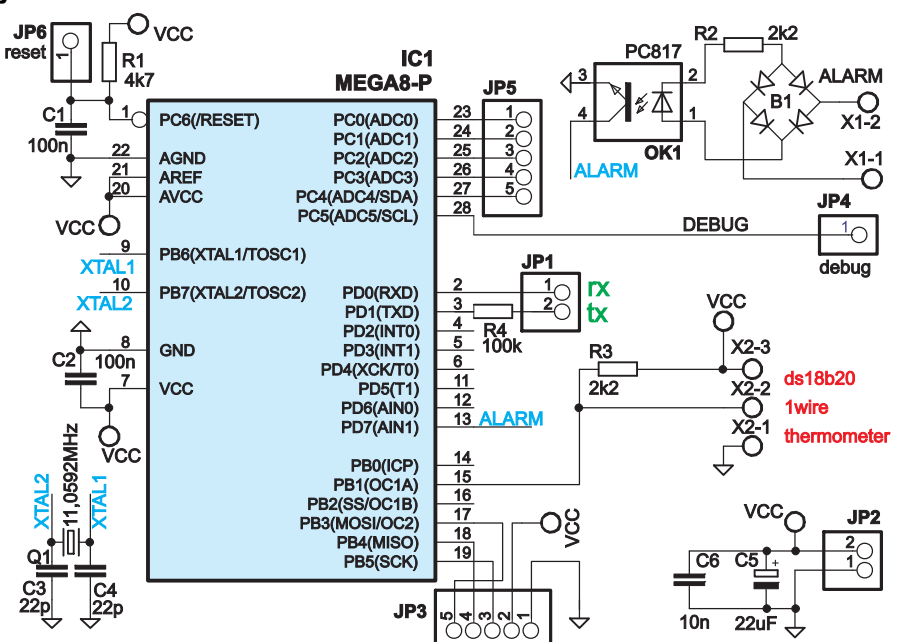
Opis programu

Na początku program wysyła do telefonu szereg komend konfiguracyjnych: wyłączenie echa (**ATE0**), włączenie identyfikacji osoby dzwoniącej (**AT+CLIP=1**), wybranie pamięci telefonu dla wiadomości SMS (**AT+CPMS=ME**), włączenie powiadomienia o przychodzącej wiadomości (**AT+CNMI=1,1,0,2**). Dokładny opis komend AT stosowanych w telefonach komórkowych można bez problemu znaleźć w Internecie, wpisując w wyszukiwarkę na przykład *siemens AT command reference*.

Następnie w programie czytane są dane z pamięci EEPROM mikrokontrolera do odpowiednich zmiennych (zmienne te dla przejrzystości zaczynają się w programie od znaków Eep_). Pamięć zorganizowana jest następująco:

- pierwszy bajt nic nie znaczy,
- kolejne sto bajtów to dziesięć numerów telefonów, które mogą korzystać z urządzenia. Numery telefonów wpisywane są do

Rys. 1



pamięci jako zmienne string – każdy numer zajmuje 10 bajtów (1 bajt na każdy znak ASCII dziewięciocyfrowego numeru telefonu + 1 bajt końca zmiennej string).

- 101 bajt (zmienna **Eep_alertrt**) to temperatura w stopniach Celsjusza, poniżej której urządzenie wyśle informację SMS.

- Kolejny bajt (**Eep_hysteresis**) to histereza temperatury alarmowej w stopniach Celsjusza x 0,1. Po obniżeniu się temperatury poniżej zadanej wysłany zostanie SMS; kolejny SMS z informacją o przekroczeniu temperatury zostanie wysłany dopiero wtedy, kiedy temperatura podwyższy się o wartość temperatury alarmu + histereza.

- Bajt 103 w pamięci EEPROM (**Eep_alarminput**) determinuje sposób zachowania się urządzenia po wyzwoleniu alarmu (napiecie na wejściu „alarm”). Wartość 0 oznacza, że zdarzenie zostanie zignorowane. Wartość 127 oznacza wysłanie wiadomości SMS o treści „Alarm!”. Każda inna wartość oznacza „puszczenie dzwonka”.

- Kolejny bajt (**Eep_power**) określa czas, jaki musi upłynąć po zaniku napięcia, żeby urządzenie wysłało odpowiednią informację SMS. Aby otrzymać wartość w sekundach, należy pomnożyć wartość zmiennej *eep_power* przez 8 (dla mikrokontrolera taktowanego zegarem 8MHz: $1/(8000000/1024/256/256) \cdot 8$).

- Kolejny bajt w pamięci EEPROM (zmienna **Eep_batterytosleep**) to procent naładowania baterii w telefonie, przy którym urządzenie się wyłączy (i wyłączy też telefon).

- Kolejny bajt (**Eep_prefix**) to prefiks kraju, w którym urządzenie się znajduje (dla Polski jest to 48).

- Kolejny bajt (**Eep_tempererror**) to liczba błędnych odczytów temperatury, po których zostanie rozesłana informacja o błędzie.

- Ostatni (108) bajt konfiguracyjny w pamięci EEPROM (zmienna **Eep_span**) to liczba numerów telefonów, do których będą wysyłane wiadomości alarmowe (o przekroczeniu temperatury, zaniku napięcia, błędach czujnika temperatury). Na przykład dla wartości 3 wymienione informacje będą wysyłane do 3 pierwszych numerów wpisanych do pamięci EEPROM.

W głównej pętli programu wykonywanych jest kilka czynności: po pierwsze sprawdzane jest, czy jest coś w buforze przy-

chodzącym UART. Zmienna *Mamycos* typu bit – flaga jest ustawiana przez przerwanie **Serial0charmatch**, wywoływane po pojawieniu się znaku ASCII 13 (CR) w buforze. Jeśli jest coś w buforze, wtedy wykonywane są odpowiednie działania:

- Jeśli tekst w buforze zaczyna się od „+CBC:”, wtedy telefon wysłał informację o stanie baterii i ładowaniu. Informacja ta zapisywana jest do zmiennej „Cbc” (typu string) w formacie *y,xxx*, gdzie *y* to 1 (ładowanie) albo 0 (brak ładowania – brak napięcia w gniazdku), a *xxx* to procent naładowania baterii. Zmienna „Cbc” używana jest później przy wysyłaniu informacji SMS. Przy definiowaniu zmiennej „Cbc” pośrednio definiowana jest zmienna „Pow”. Zmienna ta zadeklarowana jest w następujący sposób: **Dim Pow As Byte At Cbc + 0 Overlay**. Znaczy to tyle, że zmienna, a dokładnie jej pierwszy bajt, współdzieli pamięć ze zmienną „Cbc”. Dzięki temu bez dodatkowej rozbiórki zmiennej „Cbc” mamy wyłuskaną informację o ładowaniu lub braku ładowania baterii telefonu: dziesiętne 48 (kod ASCII dla 0) – brak ładowania, 49 (kod ASCII dla 1) – ładowanie.

- Jeśli tekst w buforze zaczyna się od „+CLIP:”, wtedy telefon wysłał informację o tym, że ktoś dzwoni. Do tymczasowej zmiennej *stmp* ładowany jest numer telefonu. Połączenie jest odbierane i zaraz potem odrzucane. Można połączenie od razu odrzucić, ale wtedy (przynajmniej w sieci Plus) dzwoniący usłyszy informację, że połączenie nie może być zrealizowane. Odebranie połączenia i szybkie rozłączenie się rozwiązuje ten problem – dzwoniący usłyszy trzy krótkie dźwięki oznaczające koniec (trwającej 300 milisekund) rozmowy. Dalej wywoływany jest podprogram „Checknumber”, który sprawdza, czy numer telefonu dzwoniącego jest wpisany do pamięci EEPROM. Jeśli numer jest zapisany w pamięci, odsyłany jest SMS z informacją o temperaturze i ładowaniu. W przeciwnym razie (gdy dzwoni „obcy” numer) – nic się nie dzieje.

- Jeśli tekst w buforze zaczyna się od „+CMTI:” to znaczy, że przyszedł SMS – wywoływany jest podprogram „Readsms”.

Dalej w pętli głównej programu sprawdzana jest wartość zmiennej *B8sec* (jest to zmienna, do której dodawana jest wartość 1 przy każdym wywołaniu przerwaniami od prze-

pełnienia licznika *timer0*). W ten sposób z częstotliwością około 0,12 herca – co około 9 sekund ($8000000/1024/256/256$, gdzie odpowiednio 8000000 to zegar mikrokontrolera, 1024 to wartość preskalera, 256 to 8-bitowy licznik *timer0*, 256 to 8-bitowa zmienna *b8sec*) w pętli głównej programu wykonywane są następujące czynności:

- odczytywana jest temperatura z układu DS18B20 (z poprzedniej konwersji),

- zlecana jest kolejna konwersja temperatury (do następnego odczytu),

- sprawdzane zostaje, czy jest napięcie w sieci oraz czy bateria nie jest rozładowana poniżej stanu, przy którym należy urządzenie i telefon wyłączyć i ewentualnie wysłane są odpowiednie wiadomości,

- wysyłana jest do telefonu komenda **AT+CBC** – w odpowiedzi telefon wysłał informację o naładowaniu baterii i czy jest ona ładowana (czy jest napięcie w gniazdku elektrycznym),

- sprawdzane jest, czy zmierzona temperatura nie jest niższa niż ustalona minimalna i ewentualnie wysyłana jest odpowiednia wiadomość,

- sprawdzany jest stan wejścia „Alarm” i ewentualnie wysyłana jest odpowiednia wiadomość,

- sprawdzane jest, czy przy pomiarze temperatury wystąpił błąd i ewentualnie wysyłana jest odpowiednia wiadomość.

Za odczytywanie przychodzących wiadomości SMS odpowiedzialny jest podprogram „Readsms”. Ponieważ telefon Siemens C75 używa formatu PDU do kodowania treści wiadomości SMS, do dekodowania numeru telefonu i treści wiadomości używane są podprogramy, odpowiednio **Pdu2nbr** i **Pdu2txt**. Dokładniejsze informacje na temat kodowania i dekodowania tekstu do formatu PDU można znaleźć w Internecie, wpisując w wyszukiwarce *pdu coding/decoding*. Podprogram „Readsms” w pierwszej kolejności sprawdza numer telefonu, z którego przyszedł SMS (za pomocą wywołania wspomnianego już podprogramu „Checknumber”). Dalej odczytywana jest treść wiadomości i wykonywane są odpowiednie do niej czynności:

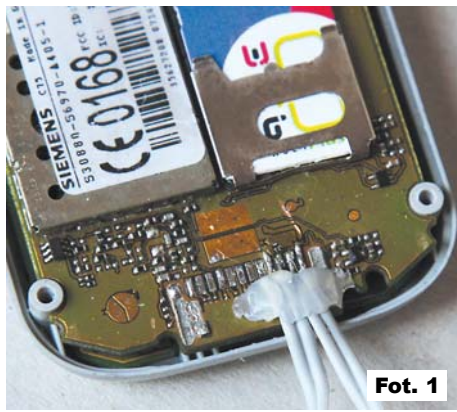
- wiadomość zawiera dane konfiguracyjne dla „termometru sms” (słowo kluczowe „set” – dane są wpisywane do pamięci eeprom. Dane konfiguracyjne powinny mieć następujący format: **set xx, yy**, gdzie „set” to komen-

R E K L A M A

da, **xx** to kolejny numer bajtu w pamięci EEPROM, **yy** to wartość, np. jeśli chcemy ustawić temperaturę alarmu na 10 stopni, należy wysłać komendę: `set 101 10`, a jeśli chcemy wpisać do pamięci numer telefonu, należy wysłać komendę: `set 51 555666777`.

- wiadomość zawiera słowo „konto” – wtedy wywoływany jest podprogram „Money”, który dzwoni na specjalny numer *100# i odczytuje odpowiedź sieci komórkowej dotyczącą stanu konta, a następnie rozbiera ją i zwraca zmienną string w formacie „Konto **xx do yy**”, gdzie **xx** to pozostałe na koncie środki, a **yy** to data, do kiedy konto jest ważne. W urządzeniu użyłem telefonu z kartą sieci Plus (marka 36,6), w innej sieci należy podprogram „Money” odpowiednio zmodyfikować,

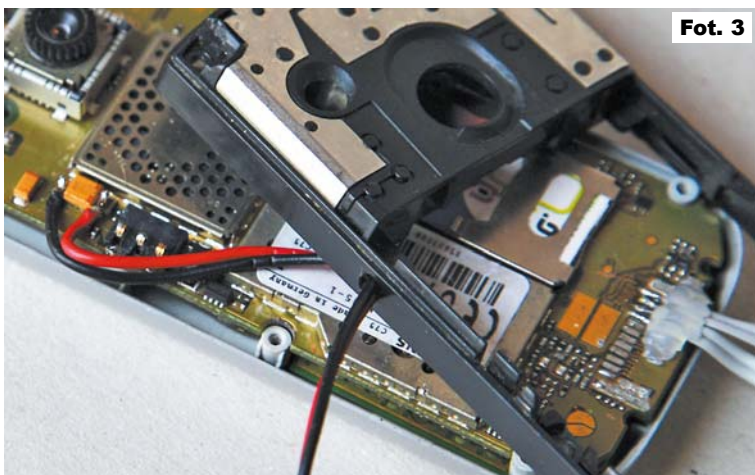
- wiadomość zawiera słowo „test” – odsyłane są bajty konfiguracyjne od 101 do 108,
- wiadomość zawiera słowo „wył” – wyłączony jest telefon i samo urządzenie,



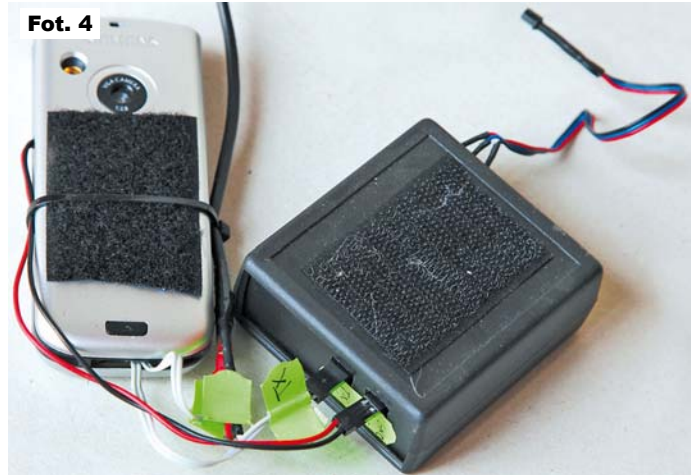
Fot. 1



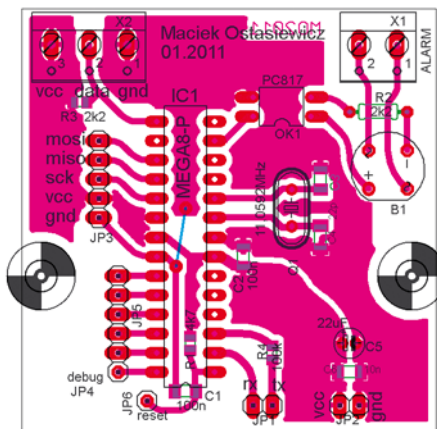
Fot. 2



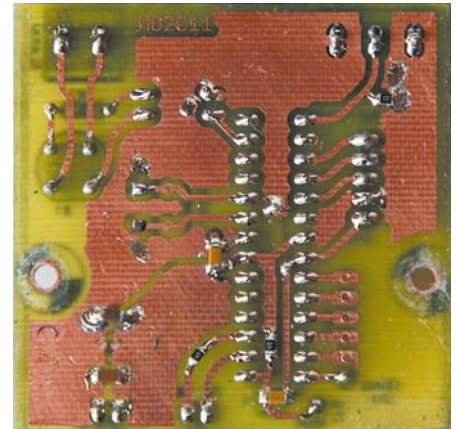
Fot. 3



Fot. 4



Rys. 2



Rys. 3



- wiadomość zawiera słowo „status” – odsyłana jest informacja o temperaturze, ładowaniu i stanie wejścia alarmowego.

Podprogram „Sendsms” używany jest do wysyłania wiadomości SMS. Przy kodowaniu numeru telefonu i treści do formatu PDU wywoływane są odpowiednio podprogramy „Nmb2pdu” i „Txt2pdu”.

Montaż i uruchomienie

Konstrukcja sterownika jest na tyle prosta, że bez problemów można zbudować go, wykorzystując płytkę uniwersalną lub na tak zwanego „pajaka”. Jednak najlepiej jest układ zmontować na płytce, pokazanej na **rysunku 2**. Płytkę zaprojektowałem, używając zarówno elementów przewlekanych, jak i montowanych powierzchniowo – w zależności jak mi było wygodniej. Płytkę nie jest jednak mocno „upakowana” i wymiana wszystkich elemen-

tów na przewlekane nie powinna stanowić problemu i jest możliwa nawet bez przeprojektowywania całości. Płytkę jest tak zaprojektowana, żeby pasowała do uniwersalnej obudowy Z-67. Płytkę mocowana

jest tymi samymi wkrętami, które skręcają obudowę. Przy grubości laminatu 1,5mm płytka akurat mieści się pomiędzy tulejkami otworów do skręcenia obudowy.

Przy montowaniu układu na płytce według rysunku 2, w pierwszej kolejności należy przylutować elementy SMD. Następnie należy wlutować zworzkę znajdującą się pod gniazdem mikrokontrolera. Dalszy montaż prowadzimy standardowo, od elementów najmniejszych do największych. Chociaż na płytce uwzględniłem możliwość zastosowania zewnętrznego rezonatora kwarcowego, w prezentowanym urządzeniu skorzystałem z wewnętrznego oscylatora RC ustawionego na 8MHz – jego dokładność okazała się wystarczająca do bezproblemowej transmisji UART.

Aby urządzenie działało prawidłowo, należy je oczywiście połączyć z telefonem. Na **rysunku 3** przedstawione jest złącze telefonu Siemens C75. Interesują nas piny od 1 do 4. Pin 1 i 2 to odpowiednio: zasilanie i masa zasilania. Do tych pinów należy podłączyć ładowarkę telefonu. Piny 3 i 4 to odpowiednio nadawanie i odbiór interfejsu UART telefonu,

który należy połączyć ze złączem JP1 na płytce sterownika (pin 3 z pinem sterownika opisanym **rx**, pin 4 z **tx**). Połączenie można wykonać, wykorzystując wtyczkę z kabla USB/RS232 telefonu lub wylutować gniazdo i przylutować przewody bezpośrednio do płytki telefonu (**fotografia 1**). Zasilanie sterownika (złącze JP2) należy poprowadzić z akumulatora telefonu. W przypadku modelu Siemens C75, najłatwiej chyba podpiąć się pod kondensator tantalowy filtrujący zasilanie, jak na **fotografii 2**, a przewód wyprowadzić przez otwór wywiercony w obudowie (**fotografia 3**).

Podobno nie wszystkie telefony komórkowe dobrze radzą sobie ze stałe podłączoną ładowarką, jednak z Siemensem C75 nie miałem żadnych problemów. Przy zasilaniu układu z akumulatora telefonu stale podłączonego do oryginalnej ładowarki, uzyskujemy zasilacz buforowy za darmo. Dzięki takiemu zasilaniu, po zaniku napięcia urządzenie nadal funkcjonuje. Zasilanie układu takim samym napięciem jak telefon pozwala także połączyć interfejs UART mikrokontrolera bezpośrednio z telefonem bez skomplikowanego dostosowywania poziomów napięć.

Opisywany termometr SMS powinien działać z każdym telefonem, który

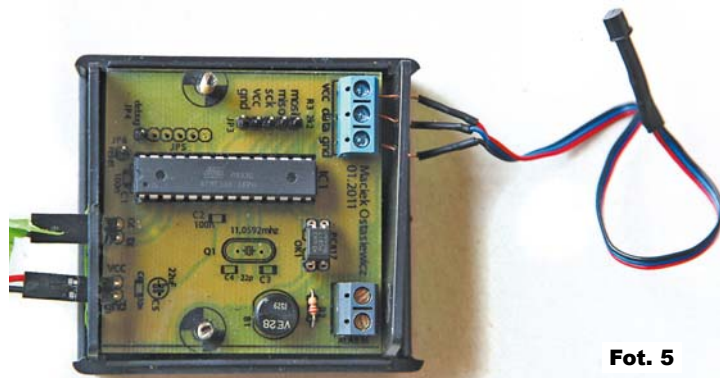
Wykaz elementów

B1	mostek 1A
C1	100n SMD 1206
C2	100n SMD 1206
C3	22p SMD 1206
C4	22p SMD 1206
C5	22uF
C6	10n SMD 1206
IC1	MEGA8-P DIP28
JP1-JP6	złącza szpilkowe
OK1	PC817 DIP4
Q1	11,0592MHz
R1	4k7 SMD 0805
R2	2k2
R3	2k2 SMD 0805
R4	100k SMD 0805
X1	ARK2
X2	ARK3

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2993.

wa tylko dla niektórych modeli telefonów Siemens.

Po zmontowaniu układu należy zaprogramować pamięć FLASH i pamięć EEPROM mikrokontrolera. Przy zastosowaniu wewnętrznego rezonatora RC mikrokontrolera, fusebity powinny być ustawione: CKSEL=0100, SUT=10. Odpowiednie piny do podłączenia



Fot. 5

ma interfejs UART i komunikuje się poprzez standardowe komendy AT. Jedynie do wyłączenia telefonu stosowana jest niestandardowa komenda „**at^smso**” – jest to komenda właści-

wa dla niektórych modeli telefonów Siemens. Po zaprogramowaniu numerów telefonów w pamięci EEPROM, układ powinien od razu działać. Jeśli byłoby inaczej, pomocne przy uruchamianiu może być złącze JP4 (*debug*) na płytce, do którego wyprowadzony jest pin procesora odpowiedzialny za nadawanie dodatkowego, programowego portu UART. W kodzie programu instrukcje odpowiedzialne za informacje wysyłane przez ten port zaczynają się od „**print #1**”. Ze względu na zajmowane miejsce w pamięci FLASH, część tych poleceń jest „zakomentowana”. Przy problemach w uruchomieniu, pomocne może być uaktywnienie niektórych z tych poleceń.

Maciej Ostasiewicz
maoss@gazeta.pl