

# Minimoduł CLIP, część 1

## AVT-955

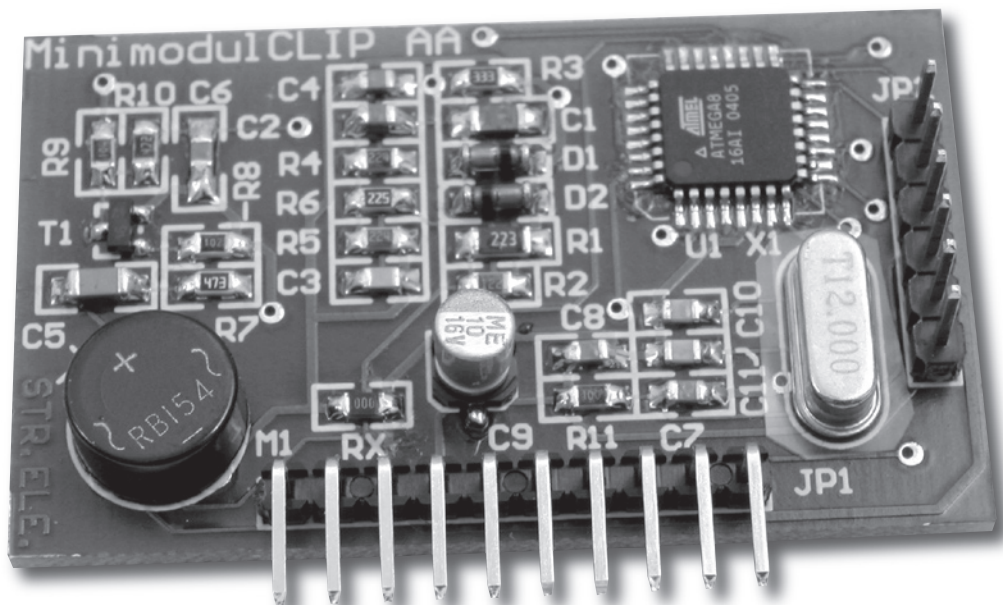
Opublikowany w EP2 i 3/2005 opis „Programowego dekodera CLIP (FSK)” wzbudził bardzo duże zainteresowanie Czytelników Elektroniki Praktycznej. Otrzymałem wiele pytań o możliwość wykorzystania pomysłu programowego dekodowania sygnału CLIP do czegoś więcej niż budowa dekodera pokazującego numer na wyświetlaczu LCD. Wielu Czytelników chciałoby zbudować system, w którym informacja o zdekodowanym numerze przekazywana byłaby do komputera PC za pośrednictwem interfejsu RS232.

**Rekomendacje:** prezentowany CLIP polecamy Czytelnikom zainteresowanym budową układów elektronicznych, w których wykorzystuje się identyfikację numeru abonenta dzwoniącego (CLIP). W artykule opisany jest układ, który dołączony do analogowej linii telefonicznej potrafi zdekodować informację CLIP oraz przesłać ją do dowolnego systemu nadrzędnego za pomocą prostego interfejsu cyfrowego.



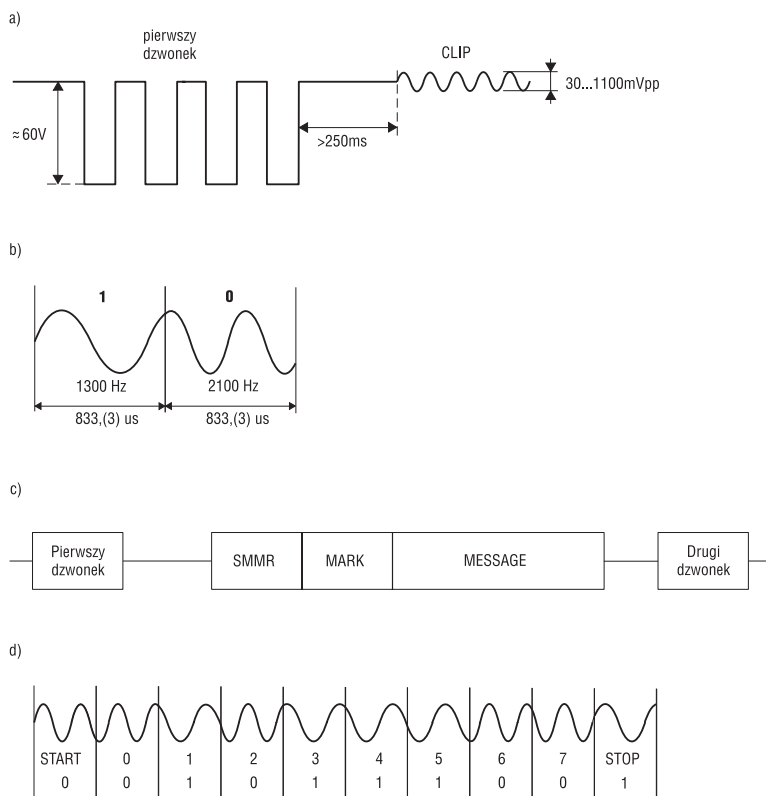
### PODSTAWOWE PARAMETRY

- Płytko o wymiarach 51 x 30 mm
- Zasilanie 5 VDC
- Tryby pracy (wymagają odmiennego oprogramowania):
  - minimoduł z interfejsem cyfrowym (magistrala)
  - minimoduł z interfejsem dla LCD

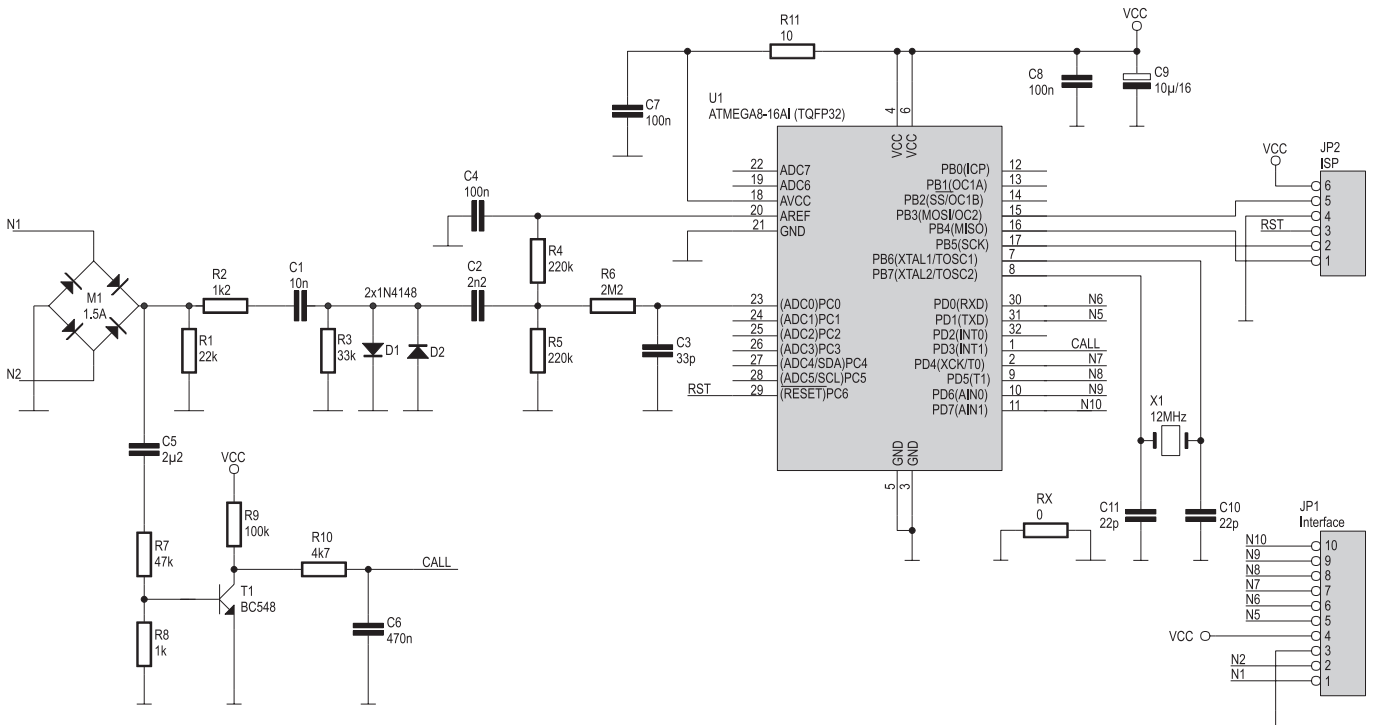


Kierując się prośbami Czytelników postanowiłem opracować remake programowego dekodera CLIP, który pozwoli w dowolny sposób wykorzystać informację o odczytanym numerze abonenta dzwoniącego. Opisany Minimoduł CLIP może być potraktowany przez wykorzystujących go konstruktorów jak hybrydowy układ scalony, umożli-

wiający odczyt numeru przez dowolny budowany przez nich układ elektroniczny. Komunikacja pomiędzy minimodułem, a tym układem odbywa się za pośrednictwem prostego interfejsu mającego charakter szyny (bus). Dla tych, którzy pragną jedynie wykonać prosty dekodery CLIP prezentujący numer na wyświetlaczu LCD przygotowałem



Rys. 1. Zasada przesyłania sygnału CLIP



Rys. 2. Schemat elektryczny Minimodułu CLIP

inną wersję oprogramowania mini-modułu, która bezpośrednio współpracuje z wyświetlaczem ciekłokrystalicznym.

W następnym punkcie tego artykułu przedstawiam opis protokołu V.23, który służy do kodowania informacji CLIP w analogowych liniach telefonicznych. Informacje w nim zawarte były już zamieszczone w EP2/2005, więc Czytelnicy którzy zapoznali się z tamtym opisem mogą od razu przejść do punktu *Opis układu*.

**Opis protokołu CLIP V.23**

Dane o numerze abonenta dzwoniącego wraz z aktualną datą i godziną przesyłane są pomiędzy pierwszym, a drugim sygnałem dzwonienia. Mają one postać sygnału sinusoidalnego o amplitudzie 15 mV...560 mV i składowej stałej równej napięciu panującemu w linii telefonicznej w stanie spoczynku (odłożona słuchawka), które wynosi 40...70 VDC (rys. 1a). Dane cyfrowe kodowane są z wykorzystaniem modulacji FSK (*Frequency Shift Keying*). Polega ona na szeregowym (asynchronicznym) przesyłaniu kolejnych bitów informacji z prędkością 1200 bodów ( $\pm 1\%$ ), przy czym wartości bitów kodowane są za pomocą częstotliwości sygnału. Jedynce logicznej (mark) od-

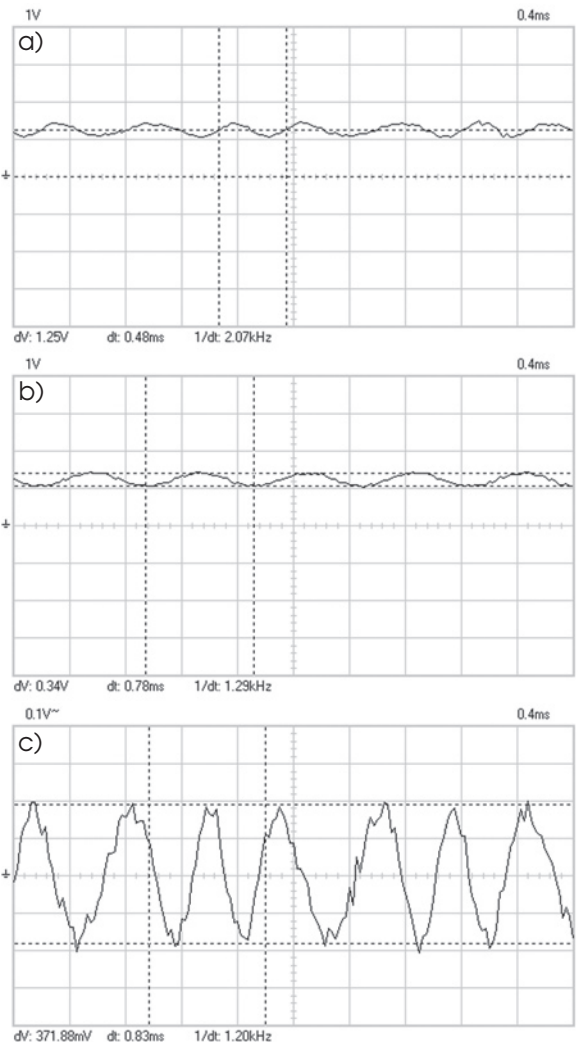
powiada częstotliwość 1300 Hz ( $\pm 1,5\%$ ), zaś logicznemu zeru (space) – 2100 Hz ( $\pm 1,5\%$ ). Jest to pokazane na rys. 1b.

Cały pakiet CLIP składa się z trzech następujących po sobie bloków pokazanych na rys. 1c. Są to:

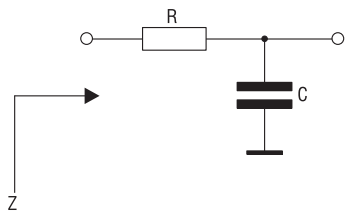
**SMMR.** Sygnał służący do nastrojenia odbiornika. Składa się z ciągu 300 bitów o wartościach zmieniających się naprzemiennie: 0, 1, 0, ...;

**MARK.** Sygnał ten składa się z ciągu 180 bitów – wszystkie o wartości 1;

**MESSAGE.** Jest to główna część pakietu niosąca w sobie informacje m.in. o numerze abonenta dzwoniącego. Składa się z sekwencji bajtów. Bajty przesyłane są, podobnie jak w standardzie RS232, w postaci ramek zawierających 1 bit startu o wartości 0, 8 bitów informacyjnych oraz 1...10 bitów sto-



Rys. 3. Rzeczywisty sygnał CLIP



Rys. 4. Najprostszy filtr dolnoprzepustowy RC

pu o wartości 1. Bity informacyjne przesyłane są w kolejności od najmniej do najbardziej znaczącego. Brak jest bitu kontroli parzystości. **Rys. 1d** przedstawia przykładową ramkę przy przesyłaniu bajtu o wartości 00111010b (58 dec).

Znaczenie kolejnych grup bajtów bloku MESSAGE (wiadomość) jest następujące:

**T1** – rodzaj danych. W przypadku przesyłania sygnału identyfikacji numeru dzwoniącego bajt ten ma wartość 10000000,

**L1** – liczba wszystkich bajtów wiadomości z wyjątkiem T1, L1 i CHECKSUM. Wartość zależy od długości numeru abonenta dzwoniącego,

**T2** – określenie rodzaju danych: czas i data. Bajt ten ma stałą wartość 00000001,

**L2** – liczba bajtów kodujących czas i datę. Bajt ma stałą wartość 00001000,

**V2** – osiem bajtów, które zawierają czas i datę zapisane w kodzie ASCII,

**T3** – określenie rodzaju danych: numer abonenta. Bajt ten ma stałą wartość 00000010,

**L3** – liczba bajtów numeru,

**V3** – numer abonenta w postaci kolejnych cyfr zapisanych w kodzie ASCII,

**CHECKSUM** – suma kontrolna. Bajt ten zawiera dopełnienie do dwóch sumy wszystkich bajtów wiadomości począwszy od **T1**

(z wyłączeniem samej sumy kontrolnej). W przypadku poprawnej wiadomości dodanie do siebie wszystkich jej bajtów łącznie z sumą kontrolną da w wyniku zero. Przy dodawaniu przeniesienie z najbardziej znaczącego bitu jest ignorowane (suma ma być 8-bitowa).

Poniżej przedstawiona jest przykładowa postać bloku MESSAGE jaki zostanie przesłany, gdy abonent o numerze 0600123456 zadzwoni do nas o godzinie 13:42 dnia 12 maja:

```
T1: 10000000 -> CLIP
L1: 00010110 -> 22 bajty od T2 do końca V3
T2: 00000001 -> czas i data
L2: 00001000 -> 8 bajtów na czas i datę
V2:
    00110000 -> '0'
    00110101 -> '5'
    00110001 -> '1'
    00110010 -> '2'
    00110001 -> '1'
    00110011 -> '3'
    00110100 -> '4'
    00110010 -> '2'
```

```
T3: 00000010 -> numer abonenta
L3: 00001010 -> 10 bajtów na numer (10 cyfr)
V3:
```

```
    00110000 -> '0'
    00110110 -> '6'
    00110000 -> '0'
    00110000 -> '0'
    00110001 -> '1'
    00110010 -> '2'
    00110011 -> '3'
    00110100 -> '4'
    00110101 -> '5'
    00110110 -> '6'
```

CHECKSUM: 11001000

**Tab. 1** zawiera zestawienie podstawowych parametrów sygnału CLIP.

## Opis układu

Schemat elektryczny minimodułu CLIP zamieszczono na **rys. 2**. Zamiast mikrokontrolera AT90S2313 (wykorzystanego w poprzedniej wersji dekodera) zastosowałem mikrokontroler ATmega8-16, który programowo analizuje sygnał CLIP pobierany za pomocą wbudowanego przetwornika analogowo-cyfrowego. Rozwiązanie takie pozwoliło uprościć część analogową, a co za tym idzie ułatwiło konstrukcję dekodera i, co bardzo ważne, zwiększyło możliwości jego miniaturyzacji.

Elementy T1, R7...R10, C5 i C6 tworzą detektor dzwonka niemal identyczny z tym, jaki występował w poprzedniej wersji dekodera CLIP. Jediną różnicą jest wartość kondensatora C5, którą zwiększyłem do 2,2  $\mu$ F. Podczas sygnału dzwonienia tranzystor T1 jest naprzemiennie zatykany i nasycany, co skutkuje występowaniem na jego kolektorze przebiegu prostokątnego odzwierciedlającego ten sygnał. Przebieg ten podawany jest na wejście INT1 mikrokontrolera U1 za pośrednictwem prostego filtra R10, C6, który ma za zadanie likwidować ewentualne impulsy szpilkowe, jakie mogą być indukowane w linii telefonicznej.

Załóżmy, że na początku w linii telefonicznej panuje napięcie stałe o wartości 40...70 V. Układ znajduje się w stanie spoczynkowym a mikrokontroler U1 pozostaje w stanie uśpienia (*powerdown*) – wyłączony jest oscylator kwarcowy i wszystkie układy peryferyjne (w tym przetwornik A/C). Prąd pobierany przez cały minimoduł w tym stanie wynosi ok. 20  $\mu$ A, co jest wynikiem wręcz znakomitym. Nadejście sygnału dzwonka powoduje podanie na wejście INT1 mikrokontrolera przebiegu prostokątnego, co z kolei powoduje obudzenie procesora U1 ze stanu *powerdown*. Po ponad 250 ms od zakończenia pierwszego sygnału dzwonienia w linii pojawia się sygnał CLIP, zawierający zakodowany numer abonenta dzwoniącego oraz aktualną datę i godzinę. Kondensator C1 odcina składową stałą i przekazuje ten sygnał na ogranicznik napięciowy zbudowany z diod D1 i D2. Diody te zapobiegają przedostawaniu się na wejście przetwornika A/C impulsów o amplitudzie przekraczającej 700 mV.

**Tab. 1. Parametry protokołu V.23**

| Parametr                      | Wartość   |
|-------------------------------|---|
| Mark (Logiczne 1)             | 1300 Hz $\pm$ 1,5%  |
| Space (Logiczne 0)            | 2100 Hz $\pm$ 1,5%  |
| Poziom sygnału mark           | -40 dBV do -8 dBV (10...398,1 mV RMS)   |
| Poziom sygnału space          | -40 dBV do -8 dBV (10...398,1 mV RMS)   |
| Różnice poziomów              | 6 dB max (2 razy)   |
| Poziom sygnałów niepożądanych | max -20 dB poniżej poziomu sygnału (300...3400 Hz)  |
| Prędkość transmisji           | 1200 baud $\pm$ 1%  |
| Format danych                 | Transmisja szeregowa asynchroniczna. 1 bit START, 8 bitów danych, 1...10 bitów STOP. Bit START 0. Bit STOP 1. |



Ze względu na to, że sygnał CLIP ma amplitudę co najwyżej 560 mV, diody D1 i D2 nie wpływają na jego kształt.

Podczas dekodowania przetwornik A/C mikrokontrolera U1 pracuje z wewnętrznym napięciem odniesienia, które dodatkowo wyprowadzone jest na końcówkę AREF. Wynosi ono typowo 2,56 V i jest filtrowane za pomocą kondensatora C4. Elementy R4 i R5 stanowią prosty dzielnik napięciowy wytwarzający składową stałą sygnału, który analizowany jest przez mikrokontroler. Składowa ta równa jest połowie napięcia referencyjnego, czyli ok. 1,28 V. Na rys. 3 przedstawiłem trzy oscylogramy sygnału CLIP, który występuje w punkcie połączenia elementów R4, R5, R6 i C2. Oscylogramy te wykonałem używając oscyloskopu cyfrowego typu PCS100 dołączanego do komputera PC. Na rys. 3a widzimy markery napięciowe pokazujące składową stałą oraz markery czasowe, które zaznaczają okres sygnału kodującego logiczne 0. Na rys. 3b widzimy fragment bloku MARK, składającego się z następujących po sobie 180 bitów o wartości 1. Markery napięciowe pokazują wartość *peak-to-peak* sygnału CLIP, wynoszącą w tym przypadku ok. 340 mV.

Na rys. 3c pokazano dokładniejszy oscylogram sygnału CLIP wykonany ze sprzężeniem zmiennoprądowym (AC) przy czułości 100 mV/dz. Znaczniki napięciowe pokazują czas, w którym przesyłana jest informacja o jednym bicie, mającym w tym przypadku wartość 0. Widoczny jest czas transmisji bitu równy 833 μs, czyli odpowiadający szybkości 1200 bodów. Jak widać, kształt sygnału daleki jest od idealnej sinusoidy, co jest spowodowane indukowaniem się w linii przeróżnych zakłóceń. Jednak nie przeszkadza to w żaden sposób w dekodowaniu, gdyż polega ono jedynie na pomiarze czasów trwania dodatnich i ujemnych połówek sygnału CLIP.

Elementy R6 i C3 tworzą filtr dolnoprzepustowy o częstotliwości granicznej (-3 dB) równej ok. 2,2 kHz. Filtr ten pozwala na pracę dekodera także w przypadku,

gdy do linii telefonicznej dołączone są inne niż telefon urządzenia, na przykład modem ADSL. Należy mieć jednak świadomość, że jest to prosty filtr pierwszego rzędu i skuteczność jego pracy jest mniejsza niż filtru Sallen-Key'a zastosowanego w pierwszej wersji dekodera. Wartości elementów R6 i C3 zostały dobrane nie tylko pod kątem właściwej częstotliwości granicznej, ale także z uwzględnieniem maksymalizacji (modułu) impedancji wejściowej filtru. Spójrzmy na rys. 4, na którym zamieściłem ogólny schemat najprostszego dolnoprzepustowego filtru RC wraz z zaznaczeniem jego impedancji wejściowej. Aby miał on częstotliwość graniczną 3-decybelowego spadku równą  $f$ , spełniona musi być zależność

$$f = 1/2\pi RC$$

Stąd

$$RC = 1/2\pi f$$

Istnieje nieskończenie wiele wartości  $R$  i  $C$  dla których otrzymamy żadaną częstotliwość graniczną. My wybierzemy takie, aby moduł impedancji wejściowej  $|Z|$  był jak największy. Dzięki temu filtr będzie możliwie słabo obciążał poprzedzający go stopień. Należy w tym momencie zwrócić uwagę na to, że maksymalizujemy **moduł** impedancji, a nie samą impedancję. Impedancja jest liczbą zespoloną i nie można mówić o tym, że jest duża bądź mała. Takie stwierdzenia, które często pojawiają się w literaturze, zawsze są skrótami myślowymi. W naszym filtrze kwadrat modułu impedancji wejściowej wynosi:

$$|Z|^2 = R^2 + 1/\omega^2 C^2,$$

gdzie  $\omega = 2\pi f$

Widzimy, że moduł impedancji jest tym większy, im większa jest rezystancja  $R$  i im mniejsza jest pojemność  $C$ . Nieskończoną wartość  $|Z|$  uzyskamy dla nieskończenie wielkiej wartości  $R$  lub dla  $C=0$ . Oczywiście w praktyce żaden z tych przypadków nie może mieć miejsca. Pojemność  $C$  powinna być co najmniej tego samego rzędu co pojemność wejściowa przetwornika A/C zawartego w strukturze mikrokontrolera U1. Przyjmijmy rozsądną wartość  $C=33$  pF. Otrzymujemy  $R=2,2$  MΩ, co także jest wartością

#### WYKAZ ELEMENTÓW

##### Rezystory

R1: 22 kΩ SMD1206  
R2: 1,2 kΩ SMD0805  
R3: 33 kΩ SMD0805  
R4, R5: 220 kΩ SMD0805  
R6: 2,2 MΩ SMD0805  
R7: 47 kΩ SMD0805  
R8: 1 kΩ SMD0805  
R9: 100 kΩ SMD0805  
R10: 4,7 kΩ SMD0805  
R11: 10 Ω SMD0805  
RX: 0 Ω (zworka) SMD0805

##### Kondensatory

C1: 10 nF SMD1206  
C2: 2,2 nF SMD0805  
C3: 33 pF SMD0805  
C4, C7, C8: 100 nF SMD0805  
C5: 2,2 μF SMD1206  
C6: 470 nF SMD1206  
C9: 10 μF/16 V  
C10, C11: 22 pF SMD0805

##### Półprzewodniki

U1: ATmega8-16AI (TQFP32) zaprogramowany  
D1, D1: 1N4148  
T1: BC548B SOT-23  
M1: mostek 1,5 A  
**Inne**  
X1: kwarc 12 MHz (4 mm)

rozsądną, gdyż rezystancja na poziomie kilku megaomów jest dużo mniejsza od występujących w praktyce pasożytniczych rezystancji równoległych (oporność laminatu, powietrza itp.).

*Minimoduł CLIP* jest zasilany stabilizowanym napięciem 5 V i jak przystało na układ, w którym występują wyraźnie wyodrębnione części analogowa i cyfrowa, ma rozdzieloną masę. Zasilanie części analogowej wytwarzane jest za pomocą filtru RC, który tworzą elementy R11 i C7. Minimoduł łączy się z układem nadrzędnym lub wyświetlaczem LCD za pomocą 10-stykowego złącza *Interface*. Rola pełniona przez poszczególne końcówki tego złącza jest zależna od tego, czy wybrano wersję Minimodułu z interfejsem cyfrowym (magistrala), czy też z wyjściem przystosowanym do wysterowania wyświetlacza LCD.

**Arkadiusz Antoniak, EP**  
[arkadiusz.antoniak@ep.com.pl](mailto:arkadiusz.antoniak@ep.com.pl)  
[www.antoniak.ep.com.pl](http://www.antoniak.ep.com.pl)