

Dział "Projekty Czytelników" zawiera opisy projektów nadesłanych do redakcji EP przez Czytelników. Redakcja nie ponosi odpowiedzialności za poprawność tych projektów, gdyż nie testujemy ich laboratoryjnie. Prosimy o nadsyłanie własnych projektów z modelami (do zwrotu). Do artykułu należy dołączyć podpisane **oświadczenie, że artykuł jest własnym opracowaniem autora i nie był dotychczas nigdzie publikowany.** Honorarium za publikację w tym dziale wynosi 200,- zł (brutto) za 1 stronę w EP. Przesyłanych tekstów nie zwracamy. Redakcja zastrzega sobie prawo do dokonywania skrótów.

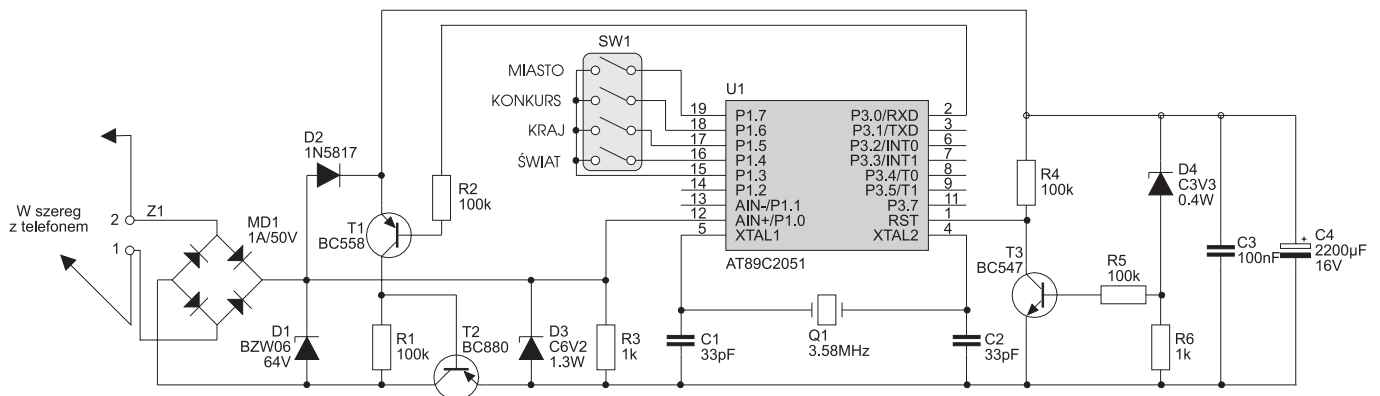
Dyskryminator telefoniczny

Prezentujemy pierwszy z projektów nagrodzonych w konkursie EP, który był poświęcony opracowaniu systemu mikroprocesorowego na dowolnym mikrokontrolerze 8-bitowym. Jest to dyskryminator połączeń telefonicznych, który zapobiega możliwości wykorzystywania zabezpieczonego telefonu do doprowadzenia firmy do stanu upadłości, o co przy dzisiejszych stawkach za impulsy, nie jest trudno.



Projekt 035

- Właściwości dyskryminatora**
- ✓ współpracuje z telefonem z wybieraniem impulsowym;
 - ✓ nie wymaga zewnętrznego źródła zasilania;
 - ✓ umożliwia niezależne ustawienie uprawnień na:
 - 1-miasto, infolinię (0800..)
 - 2-konkursy (0700..),
 - 3-kraj (0..),
 - 4-świat (00..);
 - ✓ zawsze można dzwonić na numery alarmowe (9..).



Rys. 1.

Jeśli nie możemy zdyscyplinować domowników lub pracowników a rachunki telefoniczne ciągle rosną, przyda się urządzenie, które zrobi to za nas - dyskryminator telefoniczny.

Wprawdzie Telekomunikacja Polska S.A. na żądanie abonenta może zablokować automatyczne numery kierunkowe międzynarodowe i międzymiastowe, staje się to jednak kłopotliwe, gdy naprawdę musimy gdzieś zadzwonić, a rozmowę trzeba zamawiać przez telefonistkę.

Opis układu

Schemat elektryczny układu przedstawiono na rys. 1. Dyskryminator włącza się szeregowo z aparatem telefonicznym. Mostek diodowy MD1 umożliwia poprawną pracę licznika niezależnie od aktualnej polaryzacji napięcia z centrali. Dioda lawinowa D1 zabezpiecza tranzystor T2 przed zbyt wysokim napięciem, mogącym pojawić się w linii telefonicznej.

Rolę stabilizatora pełni dioda Zenera D3. Przez diodę Schottky'ego D2 napięcie jest podawane na kondensatory C3 i C4. Elementy te

pełnią rolę zasilacza. Równocześnie napięcie z diody D3 jest podane na wejście P1.0 procesora U1. Z tego wejścia procesor wczytuje informację o tym, czy płynie prąd w pętli abonenckiej. Rezystor R3 ustala próg czułości układu wejściowego. Transzystor T2 z rezystorem R1 i T1 z rezystorem R2 pełnią rolę klucza przerywającego linię telefoniczną sterowaną z wyjścia P3.0 układu U1. Procesor przerywa linię, gdy zostanie wybrany zabroniony numer.

Listing 1.

```

#include <io51.h>

#define L_CYFR 4 /* liczba cyfr nadzorowanych */

#define WEJSCIE P1.0 /* wejście detektora petli 1-płynię prąd */
#define PRZERWA P3.0 /* wyjście na klucz przerywający 1-kłucz zwarty */
#define CZYTAJ P1.3 /* wyjście zezwalające na czytanie DIP-SWITCH-a*/
#define WE_SWIAT P1.4 /* wejście 0-zezwolenie na rozmowy międzynarodowe*/
#define WE_KRAJ P1.5 /* wejście 0-zezwolenie na rozmowy międzymiastowe*/
#define WE_KONKURS P1.6 /* wejście 0-zezwolenie na numery 0700 */
#define WE_MIASTO P1.7 /* wejście 0-zezwolenie na numery miejskie */

#define ZWARCIE 2 /* min. czas zwarcia w cyfrze *10ms */
#define PRZERWA 4 /* min. czas przerwy w cyfrze *10ms */
#define CYFRA 20 /* min. odstęp między cyframi *10ms */
#define ODLOZENIE 10 /* min. czas dla odłożenia *10ms */
#define PODNIESIENIE 20 /* min. czas dla podniesienia *10ms */
#define MAX 255 /* granica zliczania próbek */

#define KWARC 3579545 /* częstotliwość kwarcu */

#define CZAS_10MS (int)(-KWARC/12 * 10/1000)

#define LOW( liczba ) ( (char)( (int)(liczba) ) )
#define HIGH( liczba ) ( (char)( (int)(liczba) ) >> 8 ) )
/*****
 * S T A L E G L O B A L N E
 *****/
const char ALARM[] = "9...";
const char MIASTO[] = "z...0800090.";
const char KONKURS[] = "0700";
const char KRAJ[] = "01..02..03..04..05..06..070z07z007zz08..09z.";
const char SWIAT[] = "00..";

/*****
 * Z M I E N N E G L O B A L N E
 *****/
char numer[L_CYFR]; /* bufor odbieranych cyfr */
char przerwa, zwarcie; /* liczniki czasu przerwy, zwarcia petli */
char faza; /* numer fazy */
char cyfra; /* numer kolejnej odbieranej cyfry */
char i; /* zmienna indeksująca */

/*****
 * F U N K C J E
 *****/

char test(char* t)
{ /* wraca z 1 jeśli numer zawiera się we wzorcu */
  char porow = 0; /* licznik porównań */
  char wynik = 0; /* licznik zgodnych cyfr */

  i = 0;

  while( *t )
  {
    if( numer[i] )
    {
      porow++;
      switch( *t )
      {
        case '1':
        case '2':
        case '3':
        case '4':
        case '5':
        case '6':
        case '7':
        case '8':
        case '9':
          if( numer[i] == ((*t) - '0') )
            wynik++;
          break;
        case '0':
          if( numer[i] == 10 )
            wynik++;
          break;
        case 'z':
          if( numer[i] != 10 )
            wynik++;
          break;
        default :
          wynik++;
      }
    }
    t++;
    i++;
    if( i == L_CYFR )
    {
      i = 0;
      if( wynik == porow ) return( 1 );
      porow = 0;
      wynik = 0;
    }
  }
  return( 0 );
}

interrupt [0x0B] void odb_imp(void) /* obsługa przerwania od T0
*/
{
  CZYTAJ = 0;
  TR0 = 0; /* przeładuj timer */
  TFO = 0;
  TLO = LOW( CZAS_10MS );
  TH0 = HIGH( CZAS_10MS );
  TR0 = 1;

  if( !WEJSCIE ) /* czytaj stan petli */
  {
    zwarcie = 0;
    if( przerwa != MAX )
      przerwa++;
  }
  else
  {
    przerwa = 0;
    if( zwarcie != MAX )
      zwarcie++;
  }

  switch( faza )
  {
    case 0: /* spoczynek */
      if( zwarcie == PODNIESIENIE )
        faza = 1;
      break;
    case 1: /* podniesienie */
      for( i = 0; i < L_CYFR; i++ )
        numer[i] = 0;
      cyfra = 0;
      faza = 2;
      break;
    case 2:
      if( przerwa == PRZERWA ) /* oczekiwanie na przerwę */
        faza = 3;
      if( (zwarcie == CYFRA) && (numer[cyfra] != 0) )
        /* koniec cyfry */
        {
          if( cyfra == L_CYFR )
            faza = 4; /*sprawdzono wszystkie nadzorowane cyfry*/
          else
            {
              cyfra++; /* wczytaj następną cyfrę */
              if( !( test(ALARM) || !WE_MIASTO
                && test(MIASTO) || !WE_KONKURS
                && test(KONKURS) || !WE_KRAJ
                && test(KRAJ) || !WE_SWIAT
                && test(SWIAT) ) )
                faza = 5;
            }
        }
      break;
    case 3:
      if( zwarcie == ZWARCIE ) /* oczekiwanie na zwarcie */
        {
          if( numer[cyfra] < 10 ) /* nie więcej niż 10 impulsów*/
            numer[cyfra]++;
          faza = 2;
        }
      if( przerwa == ODLOZENIE )
        faza = 5;
      break;
    case 4: /* oczekiwanie na rozłączenie */
      if( przerwa == ODLOZENIE )
        faza = 5; /* dodaj własne rozłączenie */
      break;
    case 5: /* rozłącz */
      PRZER = 0;
      faza = 0;
      break;
  }
  CZYTAJ = 1;
}

/*****
 * P R O G R A M G L O W N Y
 *****/
void main(void)
{
  faza = 0;
  zwarcie = 0;
  przerwa = 0;

  TMOD &= 0xf1; /* ustawienie i uruchomienie T0 */
  TMOD |= 0x01;
  TR0 = 0;
  TFO = 0;
  TLO = 0xff;
  TH0 = 0xff;
  PT0 = 0;
  IE |= 0x82;
  TR0 = 1;
  for(;;) /* główna pętla programu */
  ;
}

```

