

Odbiornik “DigiReX” - krok w XXI wiek

INSTRUKCJA DLA PROGRAMISTÓW

DigiReX to wyposażony jest w oprogramowanie umożliwiające sterowanie odbiornikiem z komputera PC.

1. Podłączenie do PC

Podłączenie do komputera odbywa się poprzez port równoległy. Należy zachować szczególną ostrożność przy łączeniu urządzeń bo port równoległy nie jest zabezpieczony przed przepięciami. Wykorzystywane są cztery przewody (trzy przewody sygnałowe i masa).

2. Komendy sterujące

Po wejściu do trybu PC odbiornik wyświetla komunikat “PC MODE” i oczekuje na komendy sterujące z komputera. W tym trybie przyciski odbiornika są nieaktywne. Odbiornik jest urządzeniem SLAVE a komputer urządzeniem MASTER tzn. że transmisja jest inicjowana przez komputer. Każda komenda składa się z bajtu startowego (106), bajtu polecenia i ewentualnych bajtów danych. Każda transmisja jest dwukierunkowa a zwracana wartość zależy od trybu pracy.

106 (DEC)	Kod polecenia	Dane (opcjonalnie)
-----------	---------------	--------------------

Listę poleceń przedstawia tabela

Kod	Opis	Parametry	Zwracana wartość
106	Początek polecenia	...	0x44
1	Wyłącz ATT	-	106
2	Włącz ATT	-	106
3	Ustaw VFO	4 bajty B1,B2,B3,B4 częstotliwość w Hz ustawia się wg wzoru $f=B1*1000000+B2*10000+B3*100+B4$	106,0x44,0x44,0x44,0x44
4	Ustaw tryb USB	-	106
5	Ustaw tryb LSB	-	106
6	Ustaw szybkość analogowego ARW	1 bajt, 0-ARW wyłączone, 255 – max szybkość	106,0x44
7	Ustaw poziom ARW	1 bajt, 255-max	106,0x44
8	N/A		
9	N/A		
10	Tryb skanowania	Opis poniżej	
20	Wyświetl znak na wyświetlaczu na podanej pozycji	3 bajty: X – pozycja X, Y – pozycja Y, C – kod znaku	106,0xFF,0xFF,0xFF

Kod	Opis	Parametry	Zwracana wartość
21	Wyczyść wyświetlacz odbiornika	-	106
100	Wyjście z trybu sterowania	-	106

3. Tryb skanowania

Ten specjalny tryb pracy pozwala na skanowanie częstotliwości z zadaniem krokiem i po każdym kroku odczyt poziomu sygnału. Umożliwia to realizację analizatora widma. Procedurę przeprowadzającą skanowanie prezentuje poniższy fragment kodu.

4. Diagnostyka

W opcjach konfiguracyjnych odbiornika dostępny jest tryb "PC MONITOR", który umożliwia diagnostykę komunikacji PC-odbiornik. W trybie tym na wyświetlaczu prezentowane są kody przesyłane do odbiornika.

5. Przykładowy program

Przedstawiony kod programu w języku C może być skompilowany bezpośrednio w systemie linux komendą gcc demo.c

(gdzie "demo.c" to nazwa pliku tekstowego z poniższym programem).

Dla poprawnego działania port równoległy musi mieć przypisany adres 0x378.

Program realizuje klasyczny przykład i wyświetla na ekranie odbiornika tekst "HELLO".

```
#include <stdio.h>
#include <unistd.h>
#include <signal.h>
#include <sys/io.h>
#include <sys/time.h>

void delay(long p)
{
    long i;
    for(i=0;i<p;i++){p=p;};
}

void moutb(char val, unsigned int adr)
{
    outb(val,adr);
}

#define    byte    unsigned char

byte spi_inout(byte vv)
{
    byte i;
    byte p,dane;
    byte PORTD; //mirror

    i=0;
    dane=0;
    PORTD=0xFF; moutb(PORTD,0x378);
    PORTD&=~0x02;moutb(PORTD,0x378);    //CLK down
    while (inb(0x0379)&0x10){};        //wait slave ready
    PORTD|=0x02;moutb(PORTD,0x378);    //CLK UP
    PORTD&=~0x04; moutb(PORTD,0x378);
    while (i<8)
    {
```

```
    if (vv&0x80) //send MSB
    {PORTD|=0x04;moutb(PORTD,0x378);}
    else
    {PORTD&=~0x04;moutb(PORTD,0x378)};
    vv=vv<<1;
    PORTD&=~0x02;moutb(PORTD,0x378); //CLK down
    delay(5);
    p=inb(0x0379)&0x10; //read bit MSB
    dane=dane<<1;
    if (p>0){dane++;};
    PORTD&=~0x04; moutb(PORTD,0x378); //CS DOWN
    PORTD|=0x02;moutb(PORTD,0x378); //CLK UP
    delay(5);
    i++;
}
PORTD|=0x04; //CS UP
moutb(PORTD,0x378);
i=0;
while (((inb(0x0379)&0x10)==0) && (i<200)){i++;};
return dane;
} //spi_inout

void getperm()
{
    if(ioperm(0x378, 3, 1)){};
    return;
}

char trx_txt(char x,char y,char c)
{
    char cc;
    cc=spi_inout(106); //command
    cc=spi_inout(20); //text
    cc=spi_inout(x);
    cc=spi_inout(y);
    cc=spi_inout(c);
    return c;
}

char LCD_CLEAR()
{
    char c;
    c=spi_inout(106); //command
    c=spi_inout(21);
    return c;
}

char exit_spi()
{
    char c;
    c=spi_inout(106); //command
    c=spi_inout(100);
    return c;
}

int main()
{
    byte c;

    getperm();
```

```
printf("DigiReX test\n");

c=spi_inout(0xFF);      //initialize spi
c=spi_inout(0xFF);

c=LCD_CLEAR();

trx_txt(1,1,'H');
trx_txt(1,2,'E');
trx_txt(1,3,'L');
trx_txt(1,4,'L');
trx_txt(1,5,'O');

//c=exit_spi();

}
```

© 2005 SP3FHI