

Uniwersytet Warszawski
Wydział Fizyki

Wojciech Postek

Nr albumu: 262453

**Stabilizacja interferometru
Fabry-Perot**

**Praca licencjacka
na kierunku FIZYKA**

Praca wykonana pod kierunkiem
dra Wojciecha Wasilewskiego
Wydział Fizyki,
Instytut Fizyki Doświadczalnej,
Zakład optyki

Warszawa, Lipiec 2010

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data

Podpis autora (autorów) pracy

Streszczenie

W pracy przedstawiłem wykonany przeze mnie układ stabilizujący interferometr potrzebny do kontrolowania rozpraszania Ramana w parach atomowych. Opisałem działanie interferometru, a następnie zaprezentowałem sposób funkcjonowania układu stabilizującego opartego o mikrokontroler. Na koniec pokazałem wynik testu, który pozwala stwierdzić, że układ działa zgodnie z zamierzeniem.

Słowa kluczowe

interferometr Fabry–Perot, stabilizacja, mikrokontroler

Dziedzina pracy (kody wg programu Socrates-Erasmus)

13.2 Fizyka

Klasyfikacja tematyczna

Fizyka

Tytuł pracy w języku angielskim

Fabry-Perot's interferometer stabilization

Spis treści

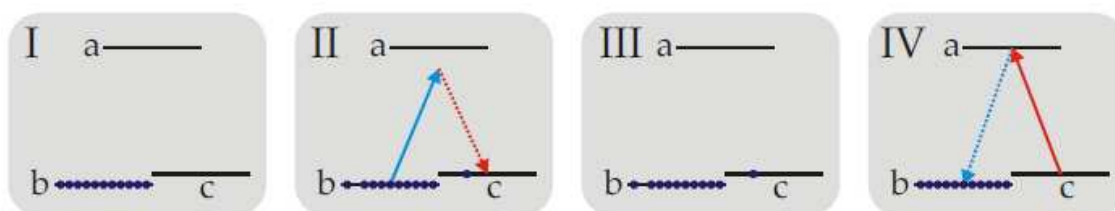
1. Wstęp	5
2. Interferometr	7
3. Stabilizacja	10
4. Testowanie	15
5. Podsumowanie	17
6. Dodatek: kod programu realizowanego przez mikrokontroler	18
6.1. Program „interrupts”	18
6.2. Program „main”	19

Podziękowanie

Dziękuję doktorowi Wojciechowi Wasilewskiemu za umożliwienie mi wykonania tej pracy licencjackiej, a także za cenne uwagi, wyjaśnienia oraz poświęcony czas.

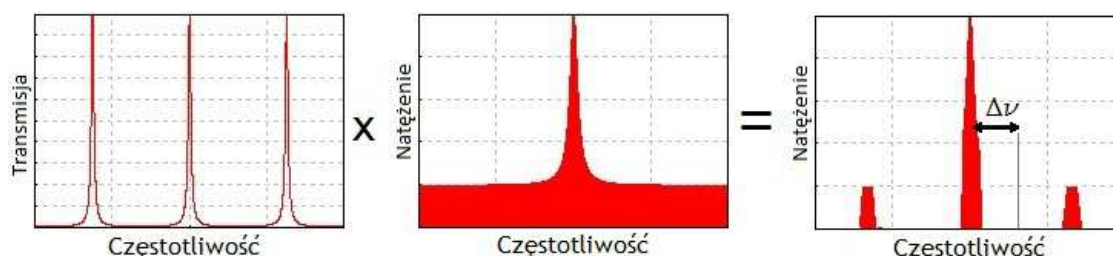
1. Wstęp

Celem pracy była konstrukcja układu stabilizującego interferometr tak aby jego maksimum transmisji stale pokrywało się z linią lasera. Układ ten znajduje zastosowanie w eksperymencie mającym w zamierzeniu generację pojedynczych fotonów dzięki wykorzystaniu rozpraszania Ramana w gorących parach atomów metali alkalicznych. Schemat tych procesów przedstawia rysunek 1. Mając wszystkie atomy w stanie podstawowym (b) I, oświetlamy je impulsem z lasera półprzewodnikowego tzw. „zapisującym”. Może on wywołać przejście na poziom (c) przy jednoczesnej emisji fotonu o częstotliwości nieco mniejszej niż oświetlającego lasera II. O przejściu dowiadujemy się poprzez zarejestrowanie fotonu stokesowskiego. Podobnie można dokonać deekscytacji z użyciem impulsu laserowego tzw. „odczytującego”. O tym, że atom uprzednio znajdujący się na poziomie (c) III znalazł się na poziomie (b) dowiadujemy się dzięki detekcji fotonu antystokesowskiego IV.



Rysunek 1: Generowanie pojedynczych fotonów za pomocą rozpraszania Ramana [1].

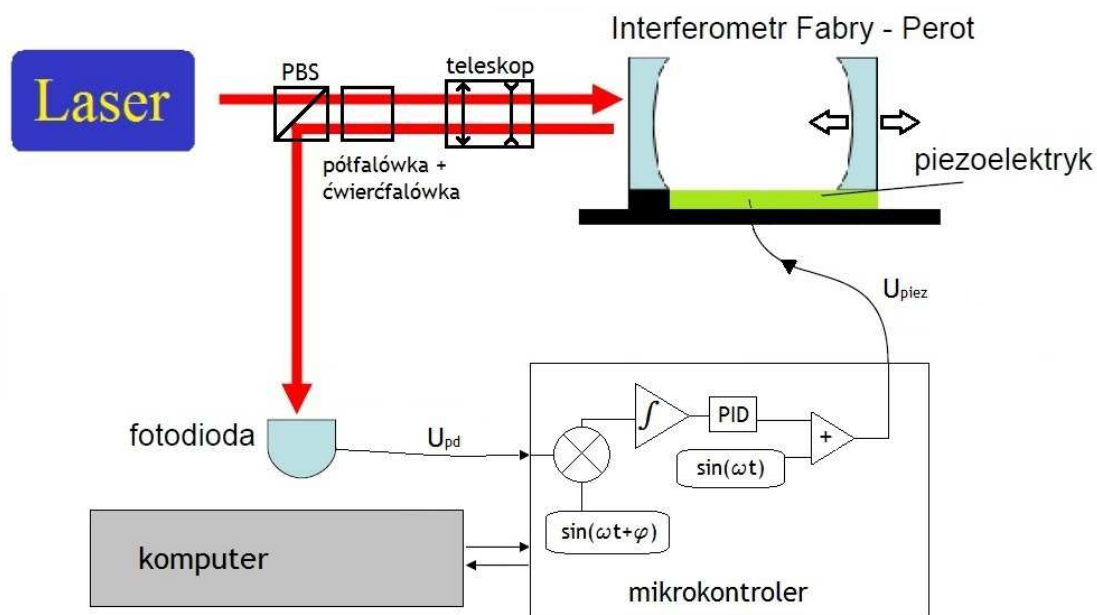
Metalem używanym do fizycznej realizacji tego procesu jest rubid. Dla izotopu ^{87}Rb różnica pomiędzy liniami absorpcyjnymi odpowiadającymi poziomom (b) i (c) w dziedzinie częstotliwości wynosi $\Delta\nu = 6,835 \text{ GHz}$. Z lasera, oprócz światła laserowego o częstotliwości $\nu_0 = 377 \text{ THz} - 795 \text{ nm}$, dostajemy na skutek świecenia diody pewną ilość fotonów o innych częstotliwościach, w tym w obszarze spodziewanej detekcji pojedynczych fotonów ($\nu_0 \pm \Delta\nu$). Odpowiada to różnicy w długościach fali rzędu tysięcznych części nanometra. Potrzebne jest zatem odfiltrowanie niepożądanych fotonów pochodzących z lasera wokół długości fali, na której spodziewamy się rejestracji pojedynczych kwantów światła. Umożliwia to interferometr Fabry-Perot. Ideę postępowania przedstawia rysunek 2.



Rysunek 2: Od lewej transmisja interferometru, widmo światła emitowanego przez laser, widmo światła po przejściu przez interferometr.

Dodatkowo na skutek czynników takich jak np. szum prądu zasilającego diodę laserową, drgania, zmiany temperatury i wilgotności itp. światło lasera nieznacznie zmienia swoją długość fali w czasie. Zmiany temperatury mają również wpływ na interferometr.

Konieczne jest więc manipulowanie interferometrem tak aby maksimum transmisji stale pokrywało się z linią laserową. Do tego celu zwierciadła osadzone są na rurce piezoelektrycznej i poprzez podawanie odpowiedniego napięcia elektrycznego zmienia się długość interferometru. Światło odbite od interferometru pada na fotodiode, z której sygnał przekazywany jest do mikrokontrolera. Mikrokontroler analizuje otrzymany sygnał i podaje napięcie na piezoelektryk tak aby minimalizować sygnał z fotodiody. Dostajemy więc minimum odbicia czyli maksimum transmisji. Schemat opisanego układu widzimy na rysunku 3. Sposób w jaki odbywa się stabilizowanie interferometru omawia część 3 – „Stabilizacja”.



Rysunek 3: Schemat układu stabilizującego interferometr.

2. Interferometr

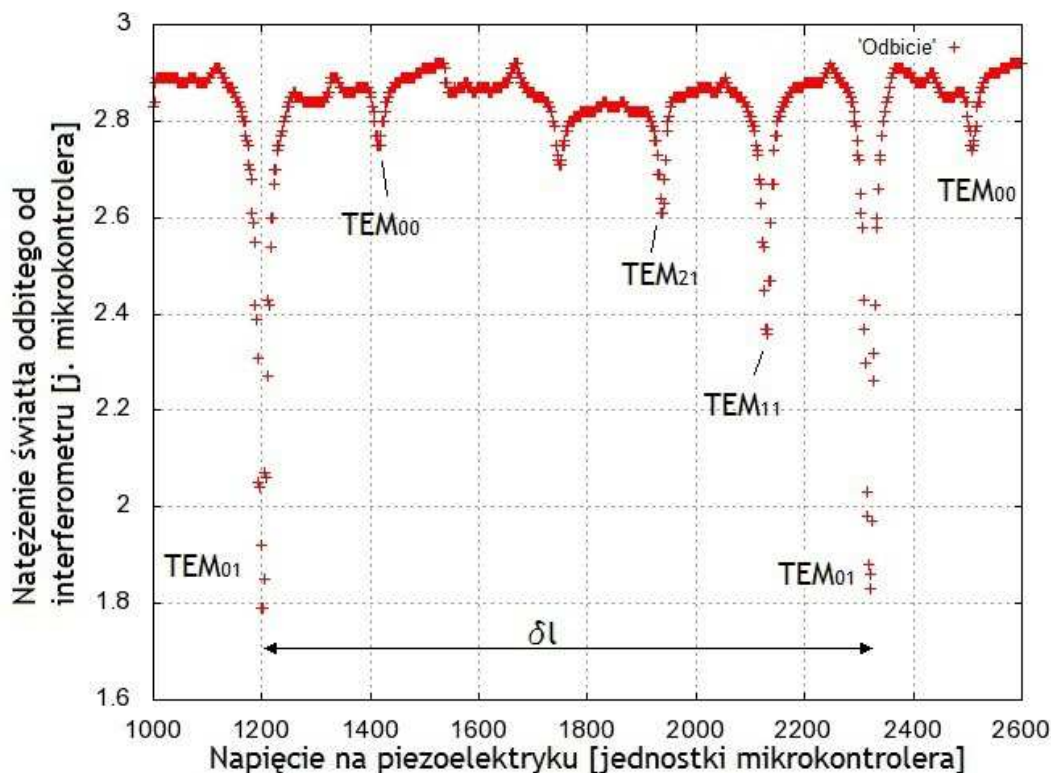
Interferometr Fabry-Perota zbudowany jest z dwóch leżących na przeciw siebie częściowo przepuszczalnych luster. Użyty został interferometr prawie płasko-równoległy (odległość l między zwierciadłami wynosi 11 mm, a ich promień krzywizny 10 cm). Schemat budowy użytego interferometru oraz zdjęcie przedstawiają rysunki 6 i 7. Transmisja interferometru T zależy od częstotliwości ν padającego światła i długości interferometru l [2, s. 147-150; 166-170]:

$$T(\nu, l) = \frac{1}{1 + \left(\frac{2F}{\pi}\right)^2 \sin^2\left(\frac{\varphi}{2}\right)}$$

gdzie $\varphi = 2l\frac{2\pi\nu}{c}$ – faza nabyta przez światło po przejściu podwojonej długości interferometru. F – finesse interferometru zdefiniowana jako stosunek przedziału dyspersji $\delta\nu$ (zakresu częstotliwości pomiędzy dwoma maksimumami transmisji) do szerokości połówkowej $\Delta\nu$ maksimum transmisji przy stałej długości interferometru lub stosunek długości δl (o jaką należy zwiększyć l aby interferometr przepuszczał falę o ustalonej ν) do szerokości połówkowej Δl maksimum transmisji. Okazuje się, że dla interferometru z wklęsłymi zwierciadłami wielkość ta zależy w przybliżeniu głównie od współczynnika odbicia zwierciadeł R .

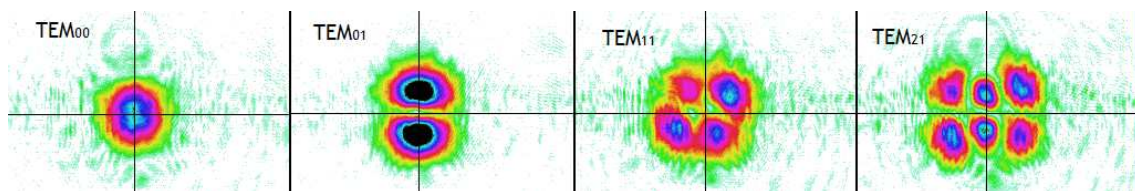
$$F = \frac{\delta\nu}{\Delta\nu} = \frac{\delta l}{\Delta l} = \frac{\pi\sqrt{R}}{1-R}$$

Finesse pozwala stwierdzić jaka jest minimalna różnica częstości między falami pozwalająca na ich rozróżnienie za pomocą interferometru. Zależność natężenia odbitego światła od przyłożonego napięcia (a więc długości interferometru) ukazuje rysunek 4.

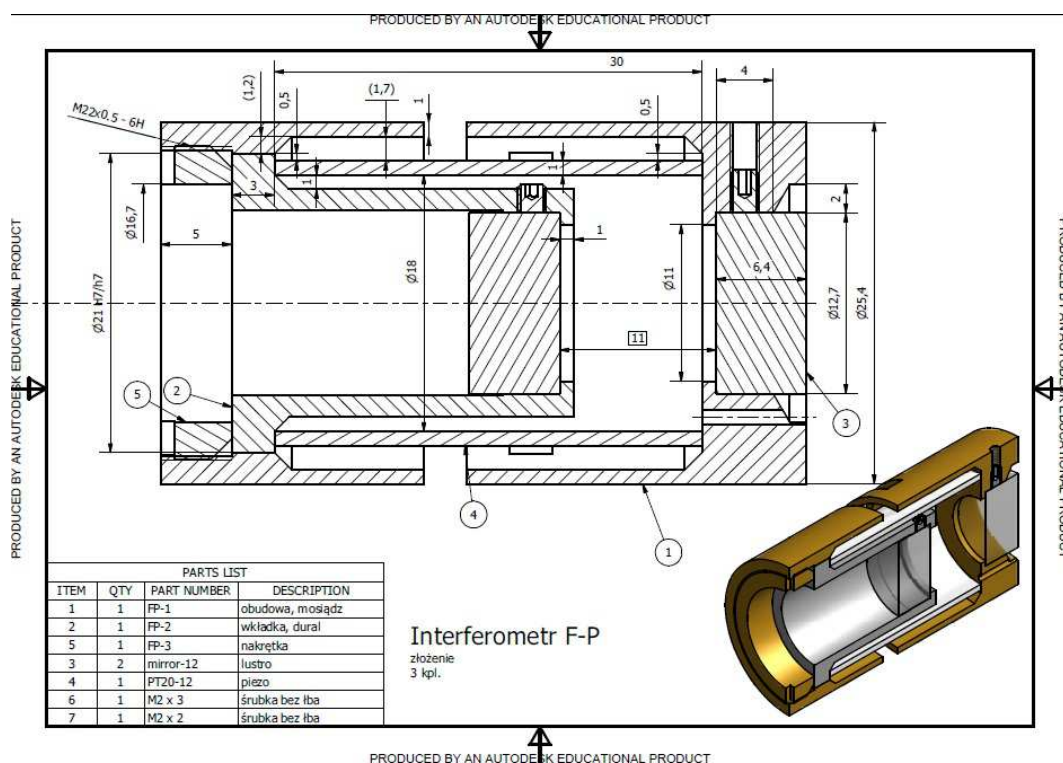


Rysunek 4: Natężenie światła odbitego od interferometru w funkcji przyłożonego do piezoelektryka napięcia, proporcjonalne do $1 - T(\nu = \text{const}, l)$.

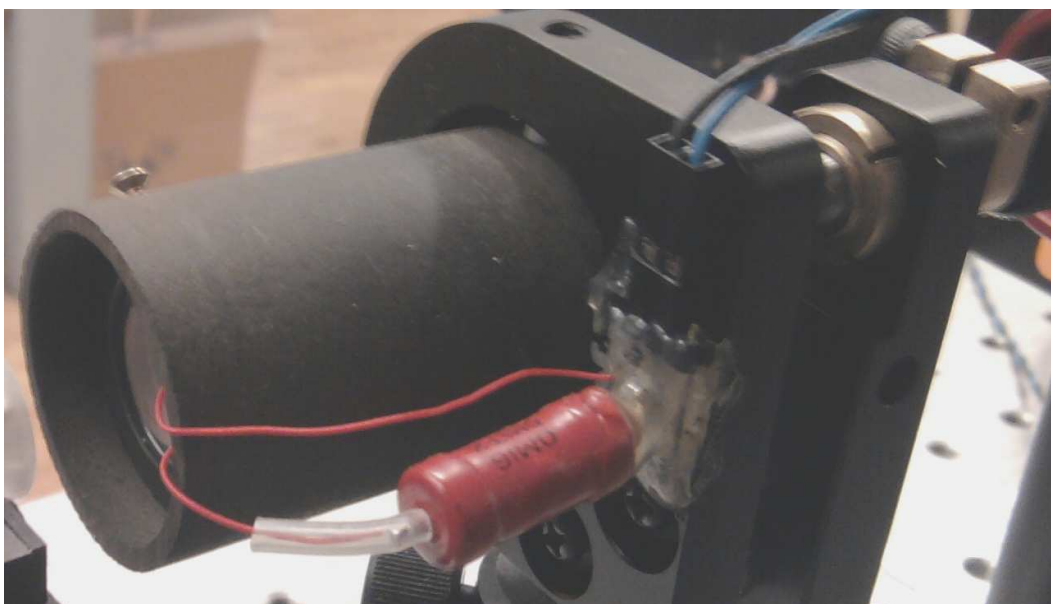
Pomiar ten (tzw. skan) został uzyskany przez program w środowisku LabView. Interferometr oświetlany był laserem. Mikrokontroler podawał na piezoelektryk skokowo rosnące napięcie (od pewnej wartości U_0 , do końcowej U_k z krokiem dU , zadanymi w programie). Jednocześnie światło odbite od interferometru zbierała fotodioda, z której sygnał przekazywany był do mikrokontrolera i wysyłany do komputera. Wyznaczona wartość finesse wynosi 52. Widzimy dwa duże minima odbicia odpowiadające modowi TEM_{01} interferometru oraz minima odpowiadające podstawowemu i wyższym modom interferometru [3, s. 409-411; 559-564]. Wielkości amplitud minimów odpowiadających poszczególnym modom zależą od ustawienia optycznych elementów układu. Przekrój transmitowanej wiązki (rozkłady natężenia modów transmitowanych wraz ze zmieniającą się długością interferometru) pokazuje rysunek 5.



Rysunek 5: Rozkłady natężenia modów powstających wraz ze zmienianiem długości interferometru. Zdjęcia wykonane kamerą BeamStar ustawioną za interferometrem.

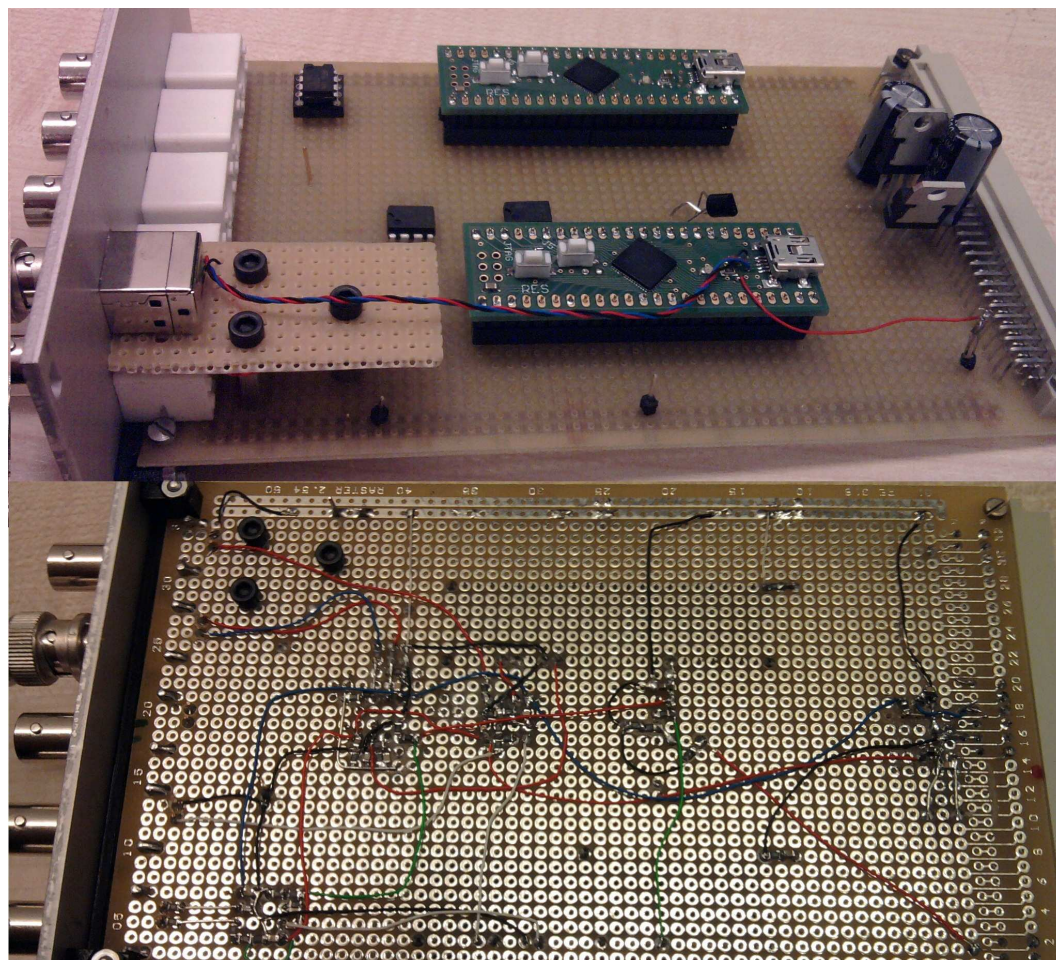


Rysunek 6: Schemat budowy interferometru.



Rysunek 7: Interferometr, widoczne przewody doprowadzające napięcie U_{piez} do rurki piezoelektrycznej.

3. Stabilizacja



Rysunek 8: Płytki z mikrokontrolerami (2 układy stabilizujące).

Opisany we wstępie problem transmisji linii laserowej przy jednoczesnym odfiltrowaniu światła o częstotliwościach spodziewanej detekcji jednofotonowej oraz uwzględnieniu zmian długości interferometru na skutek fluktuacji temperatury rozwiązany został w oparciu o mikrokontroler. Płytki z mikrokontrolerami pokazana jest na rysunku 8. Schemat całego układu pokazuje rysunek 3. Układ półfalówka i ćwierćfalówka użyty jest aby skompensować efekt dwójłomności pochodzący od soczewek w teleskopie, oraz aby polaryzacja fali odbitej od interferometru i padającej na PBS (Polarizing Beam Splitter – polaryzujący dzielnik wiązki) była prostopadła do polaryzacji fali pochodzącej z lasera.

Praca mikrokontrolera polega na realizacji programu w równoodległych w czasie tzw. przerwaniach (interrupts). Oznacza to, że mikroprocesor wykonuje kod zawarty w funkcji „interrupts”, a następnie kod z funkcji „main” do momentu wystąpienia kolejnego przerwania. W momencie wystąpienia przerwania zmienia napięcia na wyjściach. Mikrokontroler podaje na piezoelektryk sinusoidalnie modulowane napięcie $U_{piez}(t)$ tzn. do składowej wolno zmiennej $DCV(t)$ dodawany jest szybko zmienny sygnał sinusoidalny o stosunkowo małej amplitudzie a , częstotliwości ω i fazie ϕ . (Faza modulacji ϕ pozwala kompensować różnice fazowe pomiędzy sygnałem sterującym U_{piez} , a sygnałem z fotodiody U_{pd} powstałe w elementach mechanicznych i elektronicznych układu. Wy-

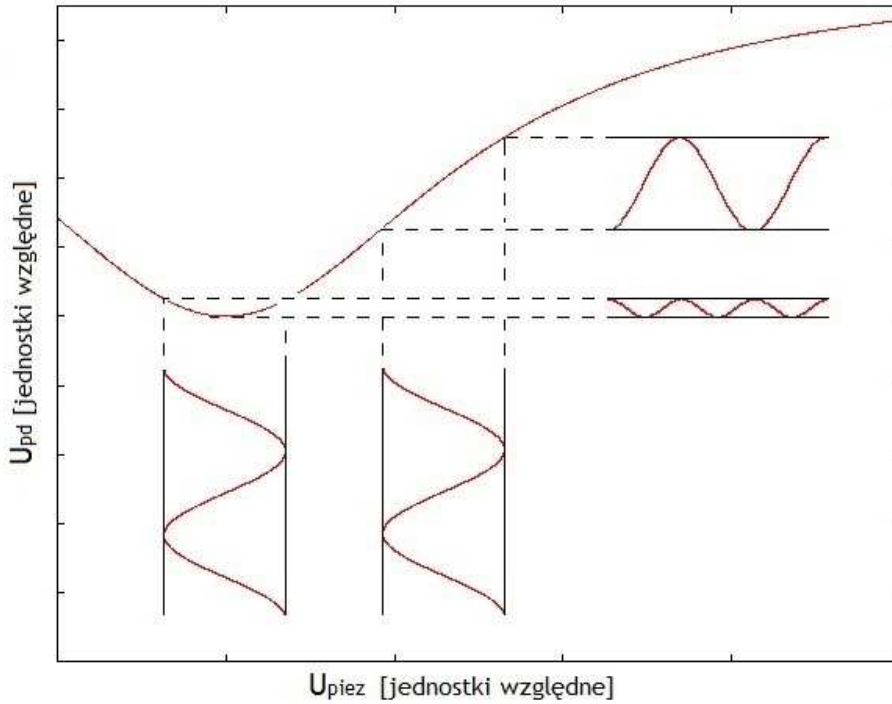
brana częstość modulacji $\omega = 2\pi \cdot 22$ kHz pozwala na szybkie uzyskiwanie potrzebnych danych, a jednocześnie układ nie generuje dźwięków w zakresie słyszalnym.)

$$U_{piez}(t) = DCV(t) + a \cdot \sin(\omega t + \phi)$$

Następnie odbiera sygnał z fotodiody $U_{pd}(t)$ i demoduluje go - oznacza to obliczanie całki (po okresie modulacji $T = \frac{2\pi}{\omega}$):

$$S(t) = \int_t^{t+T} U_{pd}(t') \cdot \sin(\omega t') dt'$$

Modulacja daje możliwość znajdowania minimum funkcji $U_{pd}(U_{piez})$, co schematycznie prezentuje rysunek 9: dla zbocza narastającego napięcie U_{pd} oscyluje z częstotliwością modulacji, a amplituda oscylacji jest duża. Dostajemy pewną większą od zera wartość S (tym większą im większe nachylenie zbocza), natomiast przy zboczu opadającym znak całki S jest ujemny. W minimum – rezonansie – częstotliwość otrzymywanego napięcia U_{pd} jest podwojona w stosunku do częstotliwości modulacji przez co całka S zeruje się.



Rysunek 9: Zachowanie się zmodulowanego napięcia U_{pd} w minimum i przy narastającym zboczu.

Przykładowy przebieg funkcji $U_{pd}(U_{piez})$ oraz całki S i $C(t) = \int_t^{t+T} U_{pd}(t') \cdot \cos(\omega t') dt'$ widać na rysunku 10. Utrzymywanie minimum odbicia dla linii laserowej możliwe jest dzięki zastosowaniu pętli ujemnego sprzężenia zwrotnego w postaci tzw. regulatora PI (proportional-integral controller). Jego działanie polega na minimalizowaniu sygnału błędu err mierzącego odchylenie od rezonansu. Oznacza to, że sygnał wyjściowy $DCV(t)$ zależy od err w następujący sposób:

$$DCV(t) = k_P \cdot err(t) + k_I \int_0^t err(t') dt' \quad (1)$$

k_P , k_I – wzmacnienie proporcjonalne i całkowe. W praktyce wzór (1) realizowany jest z każdym kolejnym okresem modulacji poprzez obliczanie:

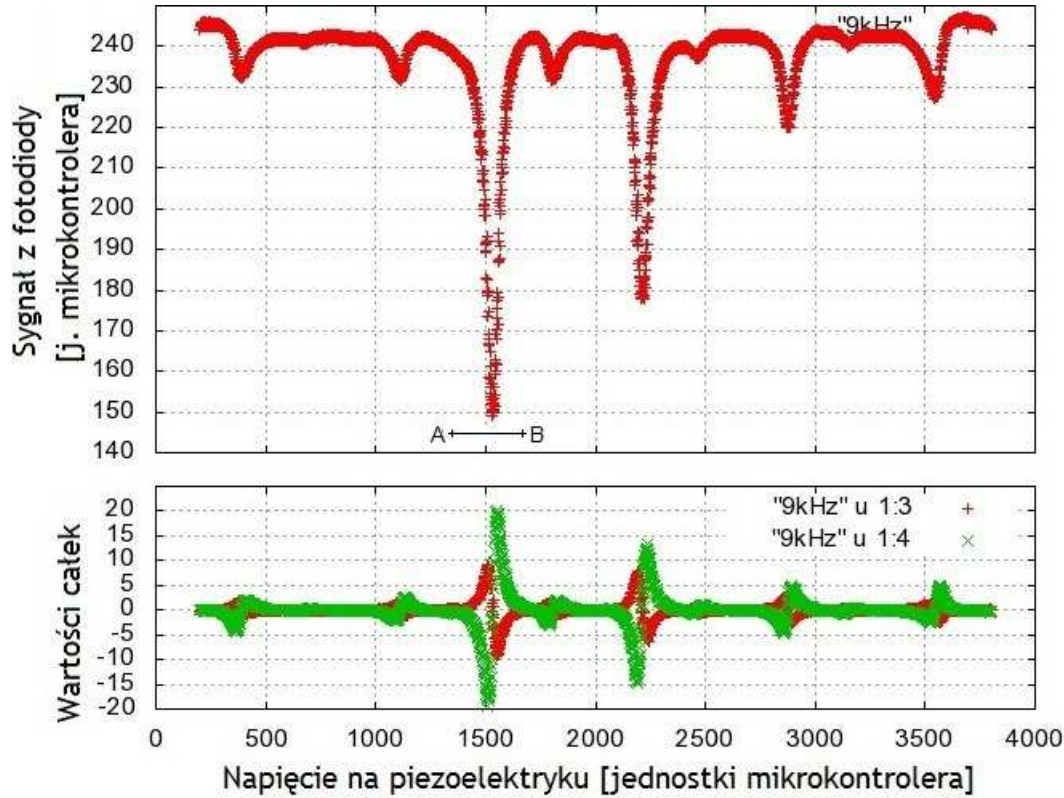
$$DCV_{n+1} = err_{n+1} + I_{n+1} \quad (2)$$

gdzie:

$$err_{n+1} = err_n + \frac{g_p \cdot S_n}{2^6} - \frac{g_{avg} \cdot err_n}{2^{12}} \quad (3)$$

$$I_{n+1} = I_n + \frac{g_i \cdot S_n}{2^8} \quad (4)$$

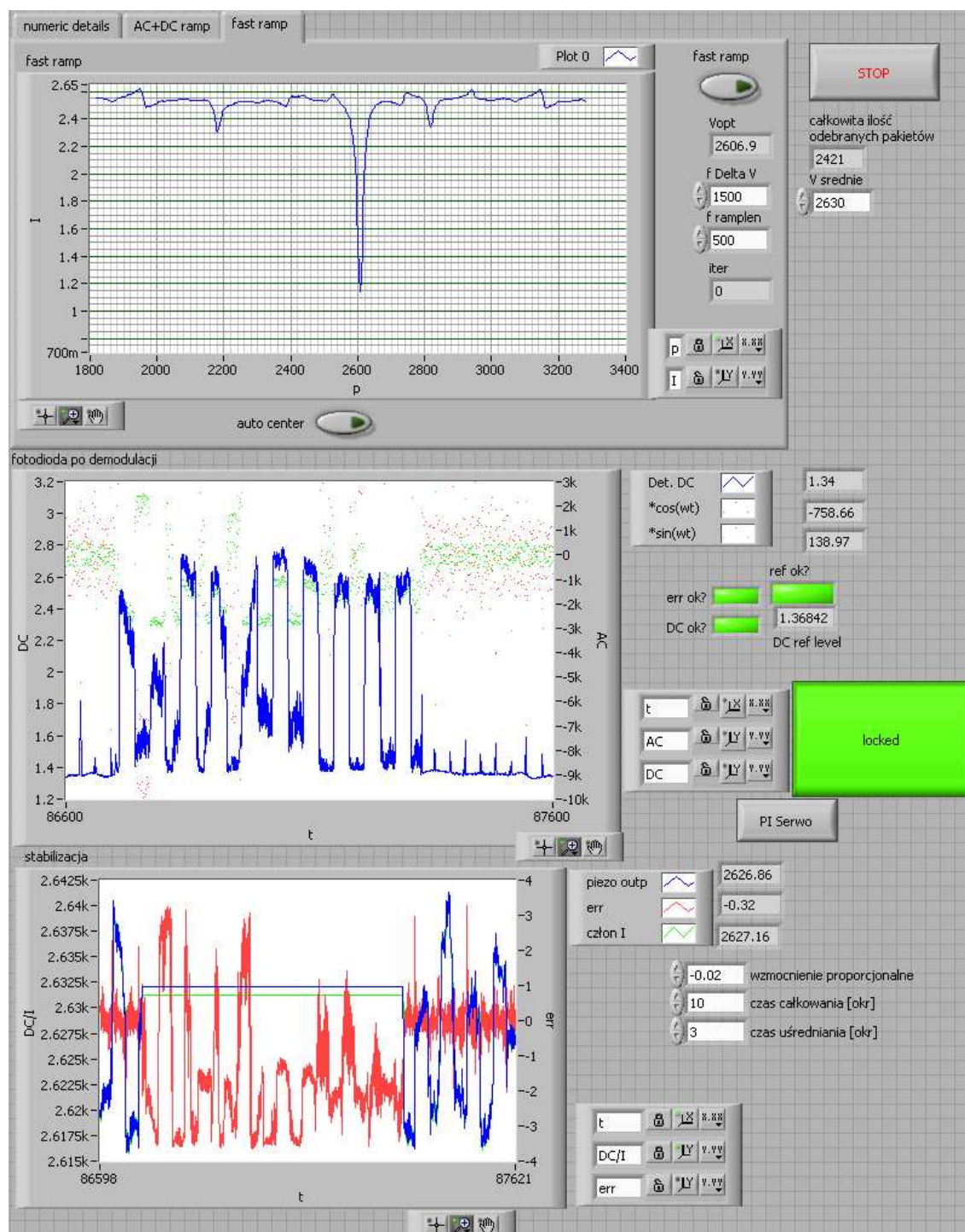
Wraz z oddalaniem się od minimum $U_{pd}(U_{piez})$ sygnał błędu err staje się ujemny dla zbocza narastającego lub dodatni dla zbocza malejącego. Realizacja wzorów (3), (4) i (2) powoduje zbliżanie się sygnału błędu do zera, a w konsekwencji zbliżanie napięcia DCV do wartości odpowiadającej aktualnemu minimum U_{pd} . Parametry g_p , g_{avg} , g_i przesyłane są do mikrokontrolera przez program monitorujący.



Rysunek 10: U góry przykładowy skan $U_{pd}(U_{piez})$ przy częstotliwości sygnału modulującego 9 kHz. Poniżej odpowiadające temu przebiegowi wartości całek S , C .

Program do monitorowania pracy mikrokontrolera został wykonany w środowisku LabView. Płyta czołowa (front panel) pokazana jest na rysunku 11. Komunikowanie się z mikrokontrolerem odbywa się za pośrednictwem portu USB poprzez podawanie i odbieranie ciągów bitów. Interfejs pozwala ustawiać parametry pracy mikrokontrolera takie jak napięcie DCV , amplitudę a , fazę ϕ i częstotliwość ω modulacji, parametry pętli sprzężenia zwrotnego. Możliwe jest śledzenie na bieżąco przebiegu $U_{pd}(U_{piez})$ w zadanym zakresie z żądaną dokładnością poprzez wykonywanie tzw. skanów, przykładowy skan widać na rysunku 4 oraz na rysunku 11 – u góry, pod napisem „fast

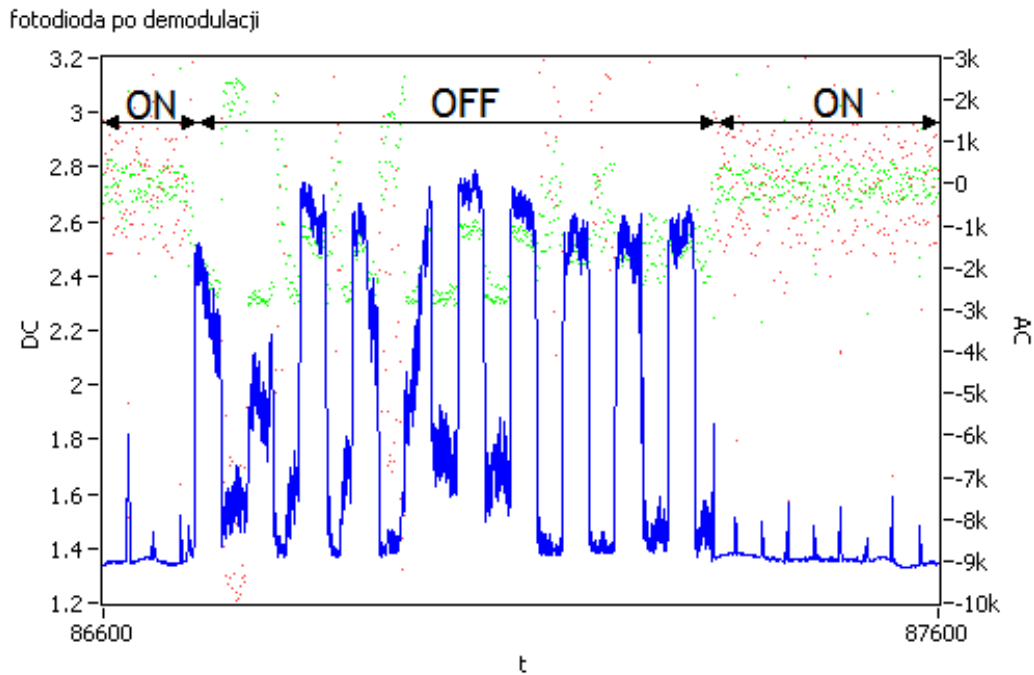
ramp”. Niżej znajdują się pola z czasowymi przebiegami napięcia z fotodiody po demodulacji, funkcji S i C , oraz zmiennych obliczanych podczas stabilizowania interferometru (DCV , I , err). Przycisk „PI Serwo” służy do włączania pętli sprzężenia zwrotnego w mikrokontrolerze, w przypadku powodzenia stabilizacji znajdujący się nad nim prostokąt ma kolor zielony i wyświetla się napis „Locked”, jeśli interferometr nie jest stabilizowany prostokąt ma kolor czerwony, a napis zmienia się na „Out of lock”.



Rysunek 11: Płyta czołowa programu służącego do komunikacji z mikrokontrolerem.

4. Testowanie

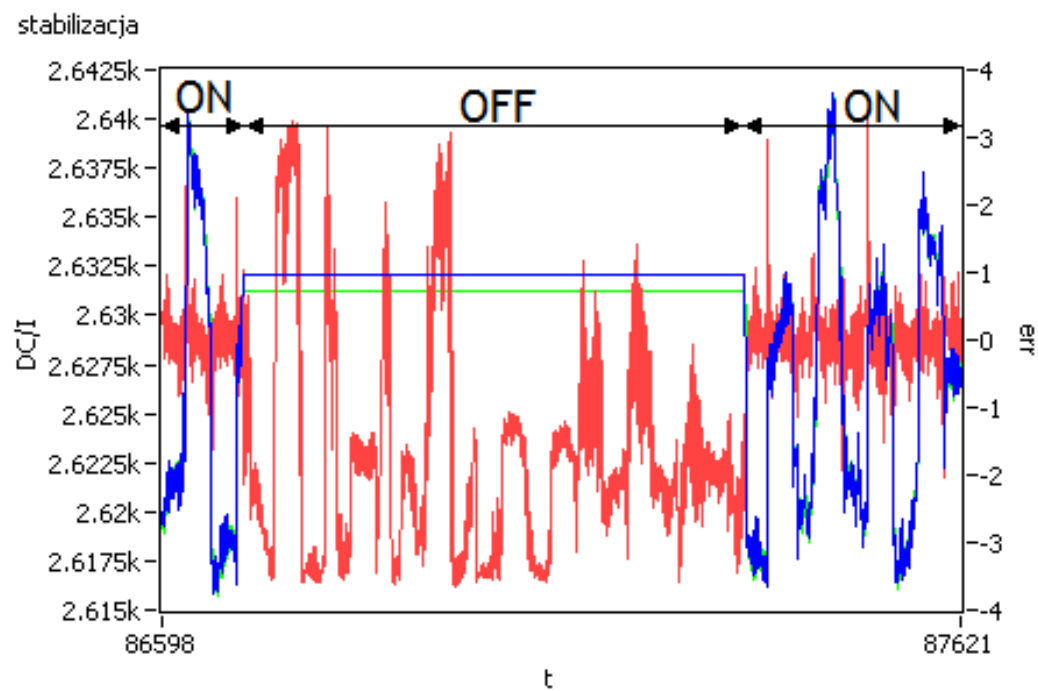
Dla sprawdzenia poprawności działania układu wykonane zostały próby polegające na symulowaniu zmian długości fali poprzez modulowanie lasera prostokątnymi impulsami o amplitudzie około 10 mV. Po wykonaniu skanu $U_{pd}(U_{piez})$ otrzymujemy optymalne napięcie odpowiadające aktualnemu maksimum transmisji interferometru. Następnie podajemy to napięcie na interferometr i włączamy pętlę sprzężenia zwrotnego. Sprzężenie zwrotne PI rzeczywiście pozwala utrzymywać maksymalną transmisję linii lasera. Przebieg czasowy zmian sygnału z fotodiody po demodulacji wraz z funkcjami S, C pokazany jest na rysunku 12. Widać na nim moment wyłączenia, a następnie włączenia pętli regulatora PI.



Rysunek 12: Przebieg sygnału odbicia od interferometru w funkcji czasu (niebieski) oraz funkcji S, C (zielony, czerwony) przy wyłączonej (OFF) i włączonej (ON) stabilizacji.

Gdy regulator jest wyłączony obserwujemy nieregularne zmiany sygnału, wraz z włączeniem sygnał stabilizuje się w minimum, o czym świadczy oscylowanie wartości funkcji S i C wokół zera. Rysunek 13 pokazuje jak zmieniały się parametry pętli sprzężenia zwrotnego. Dla włączonej pętli widzimy pokrywanie się parametru I z podawanym na piezorurkę napięciem DCV , oraz oscylowanie funkcji błędu err wokół zera. Gdy pętla jest wyłączona zarówno DCV jak i I nie zmieniają swoich wartości.

Modulowanie lasera nie może być zbyt duże tzn. amplituda modulacji nie powinna przekraczać odległości pomiędzy sąsiednimi maksimami odbicia (np. odcinek AB na rysunku 10). W takim wypadku interferometr nie stabilizowałby się.



Rysunek 13: Przebieg czasowy napięcia podawanego na piezourukę w interferometrze DCV (niebieski) oraz parametrów pętli sprzężenia zwrotnego I (zielony), err (czerwony) przy wyłączonej (OFF) i włączonej (ON) stabilizacji.

5. Podsumowanie

W ramach mojej pracy licencjackiej wykonałem urządzenie stabilizujące interferometr:

- Napisałem program realizowany przez mikrokontroler, pozwalający na sterowanie interferometrem w czasie rzeczywistym.
- Wykonałem w środowisku LabView interfejs służący do monitorowania pracy mikrokontrolera oraz ustawiania parametrów.
- Zlutowałem płytke elektroniczną, z umieszczonymi na niej mikrokontrolerem oraz wzmacniaczami wejściowym i wyjściowym.
- Wykonałem pomiary charakterystyki interferometru i wyznaczyłem finesse.
- Przetestowałem funkcjonowanie układu.

Zrealizowany układ spełnia swoje zadanie i znajdzie zastosowanie w eksperymencie generacji pojedynczych fotonów.

6. Dodatek: kod programu realizowanego przez mikrokontroler

6.1. Program „interrupts”

```
#include<ADUC7024.H>
#include"uart_lib.h"
#include"config.h"

volatile unsigned char iwfm, nwfm=6, stan=1;
volatile int LA[5],LA2[5],LR[5],ACV=50, owfm[100], swfm[100], cwfm[100];
volatile unsigned short nokresow=0, LRnokr=0, nsend=30;
volatile int DCV=200<<16,err=0,I=200<<16,q_PID=0,gp=0,gi=0,gavg=0;

__irq void FIQ_Handler(void)
{
    short int ADC_data;
    T1CLR1=1; // Clear Timer 1 Interrupt
    DAC1DAT=DCV+owfm[iwfm];

    if(++iwfm>=nwfm)
    {
        iwfm=0;
        err+=((gp*LA[1])>>6)-((gavg*err)>>12);
        if(q_PID)
        {
            I+=(gi*LA[1])>>8;
            if(I<(100<<16)) I=(100<<16);
            else if(I>(4000<<16)) I=(4000<<16);
            DCV=err+I;
        }
        nokresow++;
        ADDI(LA2,LA);
        ZEROI(LA);
        if(stan==1 && nokresow>nsend)
        {
            LRnokr=nokresow;nokresow=0;
            SETI(LR,LA2);ZEROI(LA2);
            stan++;
        }
        GP3DAT ^= 0x00010000;
    }
    //while(!ADCSTA);// czekaj na ADC
    ADC_data = ADCDAT>>16;
    LA[0]+=ADC_data;
    LA[1]+=ADC_data*swfm[iwfm];
    LA[2]+=ADC_data*cwfm[iwfm];
    //ADCSTA=0;
}

__irq void IRQ_Handler (void) {
}
```

6.2. Program „main”

```
#include <ADUC7024.h>
#include "uart_lib.h"
#include "adc_lib.h"
#include "delay.h"
#include "sin_tbl256.h"
#include <stdio.h>

volatile extern unsigned char iwfm, nwfm, stan;
volatile extern int LA[5], LA2[5], LR[5], ACV, owfm[100], swfm[100], cwfm[100];
volatile extern unsigned short nokresow, LRnokr, nsend;
volatile extern int DCV, err, I, q_PID, gp, gi, gavg;

void calc_wfms(int acv, int phi, int nwfms)
{
    int k, pha=0, frq=(1<<24)/nwfms;
    unsigned char sPHA;
    int sswfm=0;
    for(k=0; k<nwfms; k++)
    {
        sPHA=pha>>16;
        owfm[k]=acv*SIN[(sPHA+phi)&0xFF];
        swfm[k]=SIN[(sPHA)&0xFF]/32;
        sswfm+=swfm[k];
        cwfm[k]=SIN[(sPHA+64)&0xFF]/32;
        pha+=frq;
    }
    swfm[0]-=sswfm;
}

short buf[2048];

#define coerce12bit(x) if(x<0) x=0; if(x>=0x10000000) x=0xFFFF0000;

void regular_scan(int n, int V0, int dV)
{
    int k, l;
    FIQCLR|=GP_TIMER_BIT;
    coerce12bit(V0);
    if(dV<2) dV=2;
    if(n<0) n=0; if(n>=2048) n=2048;
    while(DAC1DAT>V0) DAC1DAT-=0x1000;
    while(DAC1DAT<V0) DAC1DAT+=0x1000;
    DAC1DAT=V0;
    while(!ADCSTA);
    for(k=0; k<n; k++)
    {
        buf[k]=0;
        for(l=0; l<16; l++)
        {
            while(!ADCSTA);

```

```

        buf[k]+=ADCDAT>>16;
    }
    DAC1DAT+=dV;
    if(DAC1DAT>=0x10000000) {DAC1DAT=0x10000000;n=k+1;break;}
}
UART_put32int(V0);
UART_put32int(dV);
UART_put32int(n);
for(k=0;k<n;k++)
{
    UART_put16int(buf[k]);
    DAC1DAT-=(dV>>1);
}
}

int main(void){
    int rampstep=10<<16, ramplen=10,q_ramp=1,isend=0;
    unsigned char phi=0xE0;
    //short int adc=0;
    ADC_init();
    UART_init();
    DACCON=DAC1CON=0x33; //UPD@Timer1, RANGE=AVDD
    T1CON=0xC0;          // clk/1,, internal ocs., count down
    GP3CON = 0x00000000; // GPIO
    GP3DAT = 0xFF000000; // out

    UART_putstr("ADuC 7024: RS + AD + DA project v0.5 ");
    //IRQEN|= ADC_BIT;
    T1LD=314;           // wartosc poczatkowa licznika
    calc_wfms(ACV,0,nwfm);
    FIQEN|=GP_TIMER_BIT; // zezwolnie na przerwanie timer0

while(1)
{
    if(!(0x01==(COMSTAO & 0x01)))
    // jesli w buforze UART nie czeka zaden znak
    {

        if (stan ==2)
        { int tabj;

            UART_put16int(LRnokr);
            for(tabj=0;tabj<3 ;tabj++) UART_put32int(LR[tabj]);
            UART_put32int(DCV); // wysyla wartosc DCV
            UART_put32int(err);
            UART_put32int(I);
            if(q_ramp)
                if(q_ramp<=ramplen)
                    { DCV+=rampstep; q_ramp++; }
                else
                    { q_ramp=0;
                        DCV-=(rampstep*ramplen)>>1;
                    }
            }
        }
    }
}

```

```

I=DCV;
UART_putstr(":sync:");isend=0;}

else
    {isend++;
    if(isend>32){UART_putstr(":sync:");isend=0;}
    }
    stan=1;
}

} else {char c; int l;          // jeśli mamy znak w UART
    c=COMRX;
    UART_putchar('?');
    switch(c) {
        case 'f':
            FIQCLR|=GP_TIMER_BIT;
            T1LD = UART_get32int();
            //ustawia dlugosc pojedynczego przerwania
            if(T1LD<250) T1LD=250; if(T1LD>25000) T1LD=25000;
            nwfm = UART_getchar();
            if(nwfm<2) nwfm=2; if(nwfm>100) nwfm=100;
            ACV = UART_get32int();
            if(ACV<0) ACV=0; if(ACV>1024) ACV=1024;
            phi=UART_getchar();
            calc_wfms(ACV,phi,nwfm);
            FIQEN|=GP_TIMER_BIT;
            break;
        case 'l': // loop
            gp=UART_get32int();
            gavg=UART_get32int();
            gi=UART_get32int();
            if(gi!=0) q_PID=1; else q_PID=0;
            break;
        case 'r': //ramp
            DCV=UART_get32int(); coerce12bit(DCV);
            rampstep=UART_get32int();
            ramplen=UART_get32int();
            q_ramp=1;
            break;
        case 's': // ramp w/o mod.
            FIQCLR|=GP_TIMER_BIT;
            DCV=UART_get32int(); coerce12bit(DCV);
            rampstep=UART_get32int();
            ramplen=UART_get32int();
            regular_scan(ramplen,DCV,rampstep);
            I=DCV=DAC1DAT;
            FIQEN|=GP_TIMER_BIT;
            break;
        case 'm': //mointor
            nsend=UART_get32int();
            break;
        case 'Q': // status
            UART_put32int(DCV); // wysyla wartosc DCV
            UART_put32int(err);
    }
}

```

```

        UART_put32int(I);
        UART_put32int(ACV);
        UART_put32int(gp);
        UART_put32int(gavg);
        UART_put32int(gi);
        UART_put32int(rampstep);
        UART_put32int(rampplen);
        UART_put32int(q_ramp);
        UART_put32int(T1LD);
        UART_putchar(q_PID);
        UART_putchar(phi);
        UART_putchar(nwfm);
        for(l=0;l<nwfm;l++) UART_put16int(swfm[l]);
        UART_putstr(":sync:");isend=0;
    }
    UART_putstr(":sync:");isend=0;
} //koniec warunku ze jest znak (else)
}
}

```

Literatura

- [1] Wojciech Wasilewski, „Opis badań - Rozpraszanie Ramana w parach atomowych.”, dostępny w <http://www.fuw.edu.pl/~wwasil/sci/raman.pdf>
- [2] Wolfgang Demtröder, „Spektroskopia Laserowa”, Wydawnictwo Naukowe PWN, Warszawa 1993
- [3] Anthony E. Siegman, „Lasers”, University Science Books, California, Mill Valley 1986