Wiedza i doświadczenie projektowe wizytówką absolwenta kierunku automatyka i robotyka na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej

POKL.04.01.02-00-020/10

Program Operacyjny Kapitał Ludzki współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego

---

Gliwice, 28.II.2012 r.

Międzywydziałowe Koło Naukowe High Flyers
Wydział Automatyki Elektroniki i Informatyki
Kierunek Automatyka i Robotyka

Raport z realizacji projektu:

Projekt stanowiska sterowania komorą podciśnieniową

Zespół projektowy:

Marcin Cierniak (lider projektu)

Piotr Tracichleb

Lucjan Cichoń

Michał Waruga

Dominik Piętka

inż. Mateusz Markowicz (konsultant)

1. Idea stanowiska

Celem było stworzenie projektu stanowiska dającego możliwość stabilizowania podciśnienia wykorzystywanego przy pracach nad wykonywaniem elementów laminatowych w technologii podciśnieniowej.

Idea stanowiska zakłada podział na następujące główne elementy:

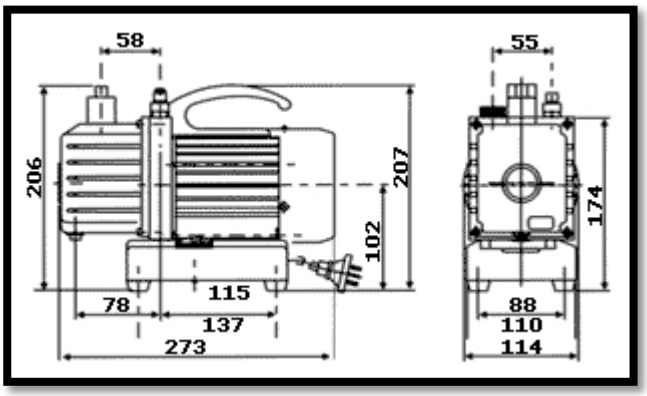
- Sterownik próżni
- Element wykonawczy
- Komora podciśnieniowa

Główne prace realizowane w ramach projektu skupiają się na wykonaniu sterownika próżni. Sterownik ten powinien wykorzystywać jeden z powszechnie dostępnych czujników pomiarowych i przy pomocy mikrokontrolera (np. serii AVR ATmega) odpowiadać za interfejs użytkownika (np. wyświetlacz, przyciski) jak również z procedury regulacji jak i wyzerowywanie wyjść ze sterownika. Jako element wykonawczy można założyć jedną z dostępnych na rynku pomp podciśnieniowych zasilaną napięciem zmiennym 230V przy pomocy przekaźnika którego cewka podłączona będzie do wyjścia napięciowego sterownika. Komora podciśnieniowa powinna umożliwić symulacje pracy układu, może być zrealizowana jako szczelny pojemnik posiadający połączenie zarówno z elementem wykonawczym jak i czujnikiem podciśnienia.

2. Element wykonawczy

Na rynku istnieje szereg pomp które spełniają założenia projektu. Niemniej jednak istotnym czynnikiem, przy wyborze urządzenia była jego cena. Na tej podstawie wytypowano dwa modele pomp dostępnych na rynku TW-1A oraz TW-1AV

Oba modele mają bardzo zbliżone parametry jednak różnią się wyposażeniem w elektromagnetyczny zawór zwrotny, zapobiegający sytuacjom, w których po wyłączeniu zasilania pompy następuje zaciągnięcie do komory podciśnieniowej oleju z pompy. Dane pompy przedstawiono poniżej



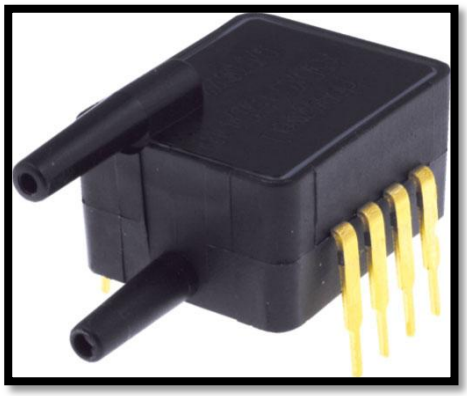
dane techniczne:

zasilanie	230[V] 50Hz
Wydajność	2[CFM]; 57[l/min]; 3,42[m^3/h]
Próżnia końcowa	10[Pa]; 0,1[mBa]
Prędkość obrotowa	1440[r/min]
Moc	120[W]; 1/6[HP]
Ilość oleju	220ml
Przyłącze gazowe ssące	Rozmiar ¼ SAE Male
Wymiary	273x110x207[mm]
Waga	6,9[kg]

3. Dobór elementów pomiarowych

Na podstawie warunków pracy narzuconych przez urządzenie wykonawcze zdecydowano się na wybranie czujników podciśnienia o zakresie pracy 0-15 psi. Dostępne na rynku w ciągłej sprzedaży są w zasadzie dwa modele czujników:

- asdx015d44r



specyfikacja:

Typ czujnika	ciśnienia
Zakres pomiarowy	0-15 psi
Sposób zadziałania	różnicowy
Konfiguracja wyjścia	Analogowe 0,5 do 4,5 VDC
Zakres napięć zasilających	4.75..5.25 VDC
Temperatura pracy	-20-105°C

- mpx2100dp

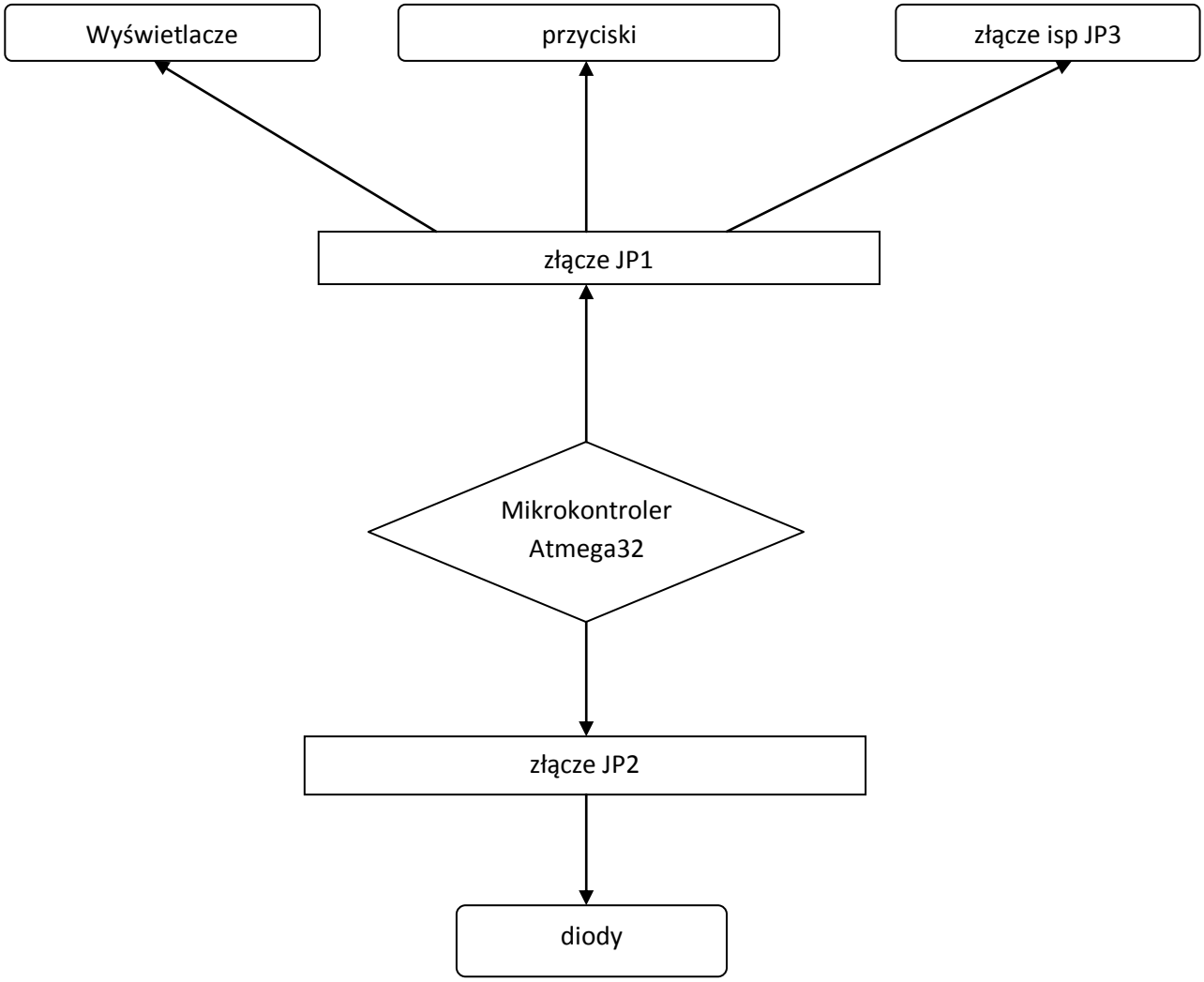
specyfikacja:

Typ czujnika	ciśnienia
Zakres pomiarowy	0-100kPa
Sposób zadziałania	Różnicowy
Konfiguracja wyjścia	Analogowe napięciowe
Zakres napięć zasilających	10-16 VDC

4. Schematy funkcjonalnych części elektronicznej stanowiska, określenie potrzebnych podzespołów elektronicznych

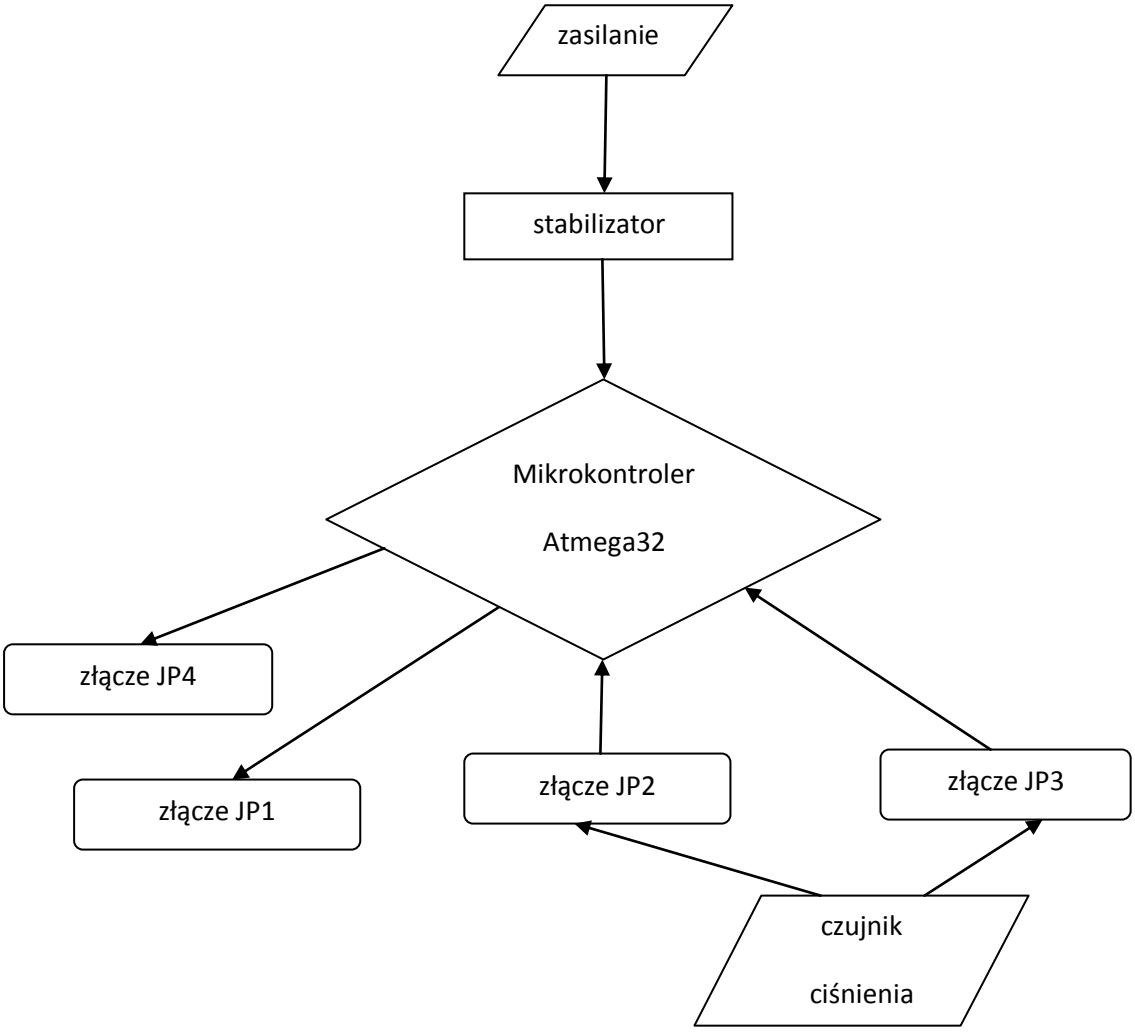
- Panel czołowy

Panel za pomocą którego sterujemy presostatem, składa się z dwóch segmentowych dwukolorowych wyświetlaczy, czterech przycisków oraz sześciu diod. Panel wyposażony jest w złącze szpilkowe goldpin JP1 za pomocą którego ma komunikować się z mikrokontrolerem, oraz w złącze isp JP3 pozwalające programować mikrokontroler bez konieczności wyciągania układu z obudowy. Konieczne było umieszczenie dodatkowego złącza JP3 na podłączenie diod.



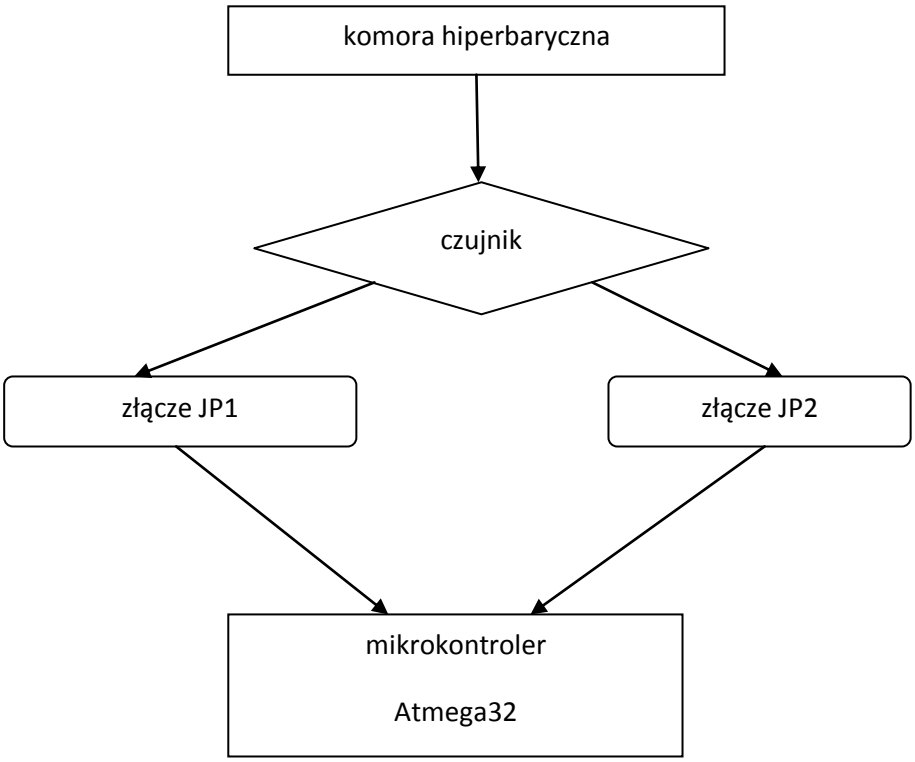
- Mikrokontroler

Podzespołem wykonawczym jest ATMEGA32. Układ jest zasilany napięciem 24V (poprzez złącze śrubowe), które jest wysterowywane przez stabilizator L87L05 na 5V. Za pomocą układu tranzystorów wysterowaliśmy przekaźnik na 24V do drugiego złącza śrubowego. Procesor komunikuje się z panelem czołowym przez gniazdo JP1 i złącze JP4 (diody), zaś z czujnikiem poprzez 2 gniazda JP2 i JP3, które tworzą podstawkę pod płytkę z czujnikiem. Zastosowaliśmy rezonator kwarcowy 16MHz oraz stworzyliśmy układ z przyciskiem pod RESET.



- Układ z czujnikiem

Układ składa się z czujnika i 2 gniazd, którymi łączy się z mikrokontrolerem. Mamy do dyspozycji 10 miejsc na złączach podczas gdy czujnik korzysta tylko z trzech. Będzie to jednak podstawka uniwersalna pod różne podzespoły.



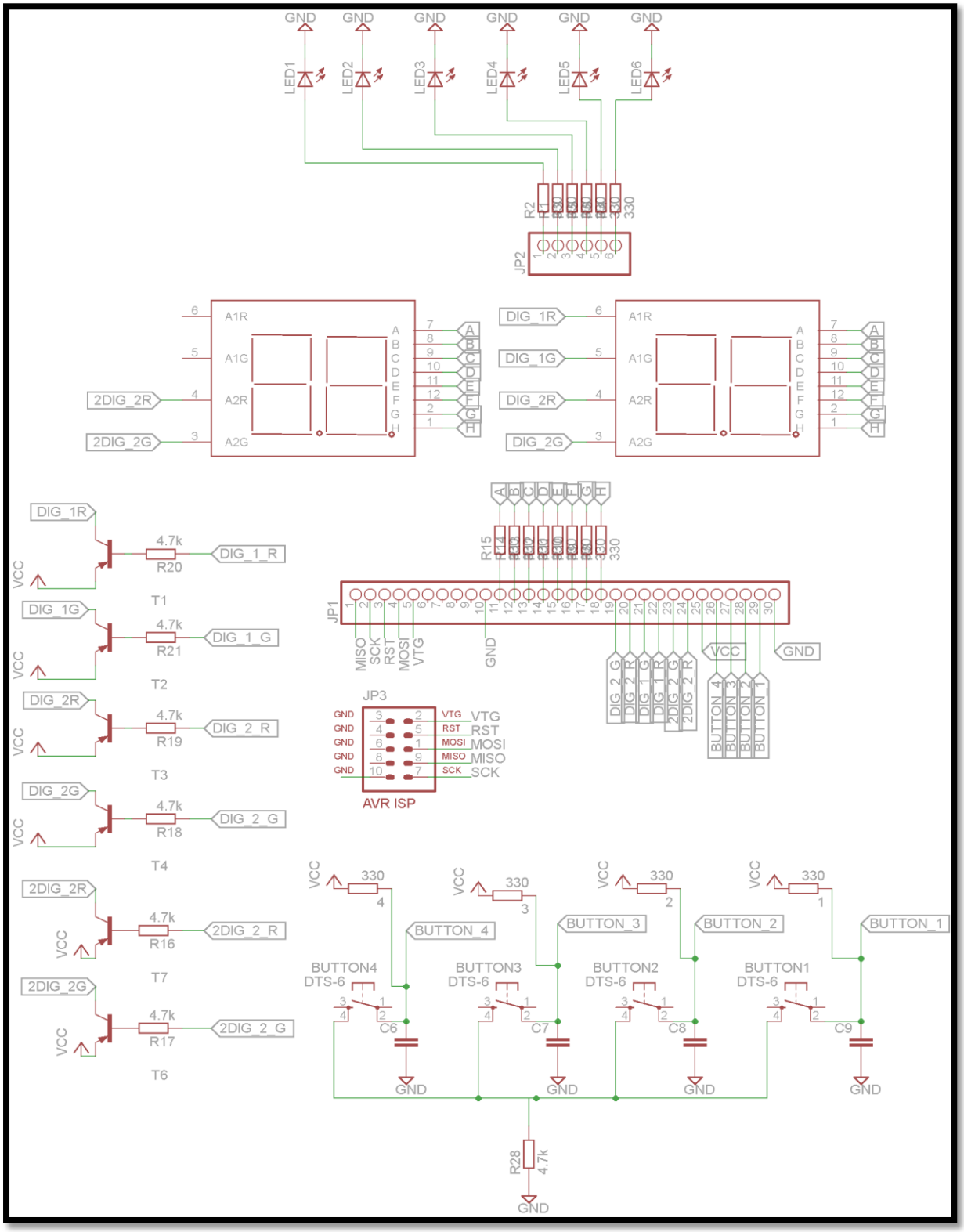
Całość zamykać się będzie w obudowie

Płytka czołowa będzie stabilnie przymocowana za pomocą złącza goldpin, opierać będzie się na szynach pod płytki a od strony zewnętrznej przylegać będzie do czterech podpórek przedniego panelu obudowy. Płytka z mikrokontrolerem połączona złączem goldpin z płytką czołową umieszczona będzie w jednej z szyn. Od tylnej strony obudowy przyłączone zostaną kable z zasilaniem.

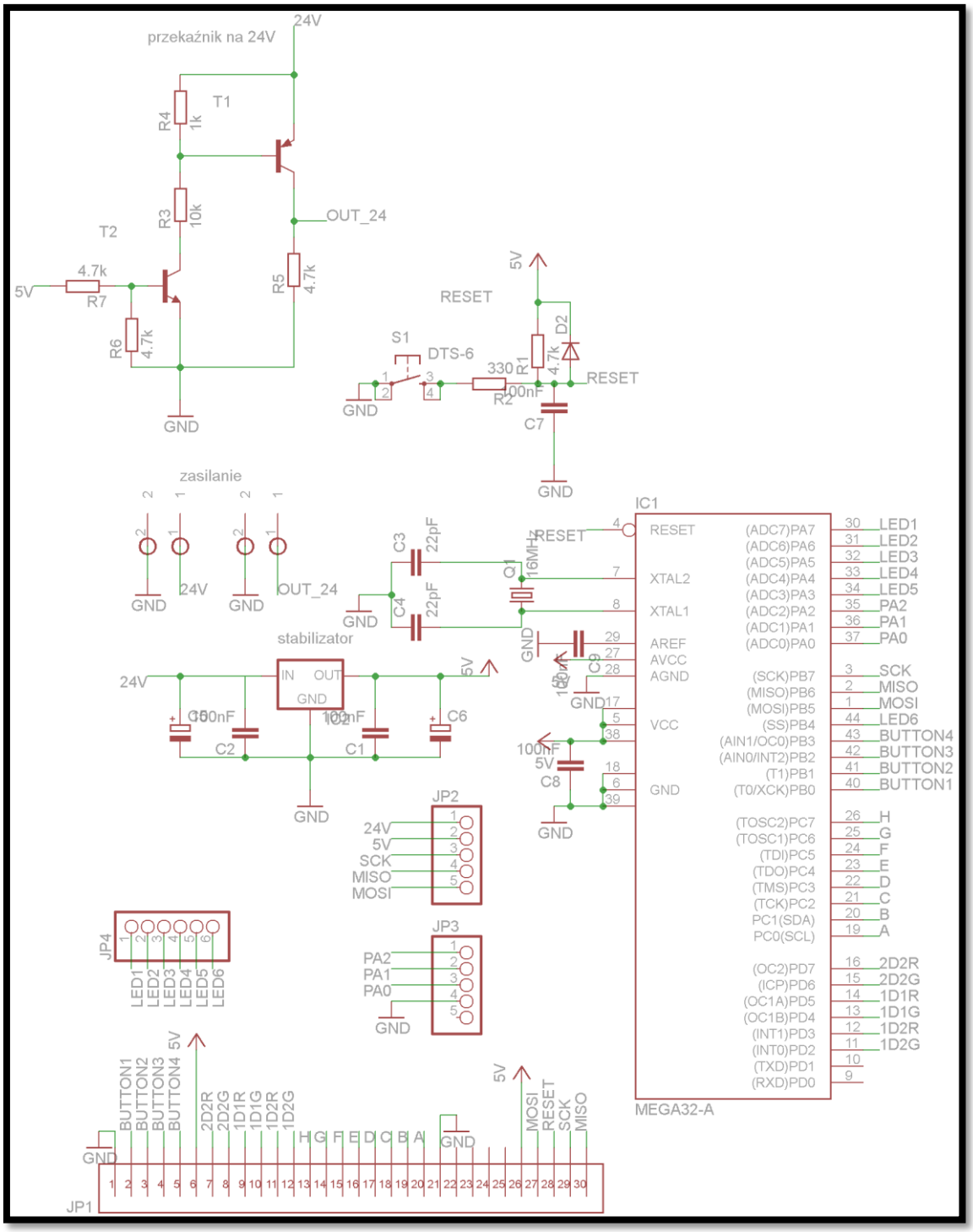
5. Realizacja projektu elektroniki

Schematy poszczególnych układów elektronicznych wykonanych w programie EAGLE

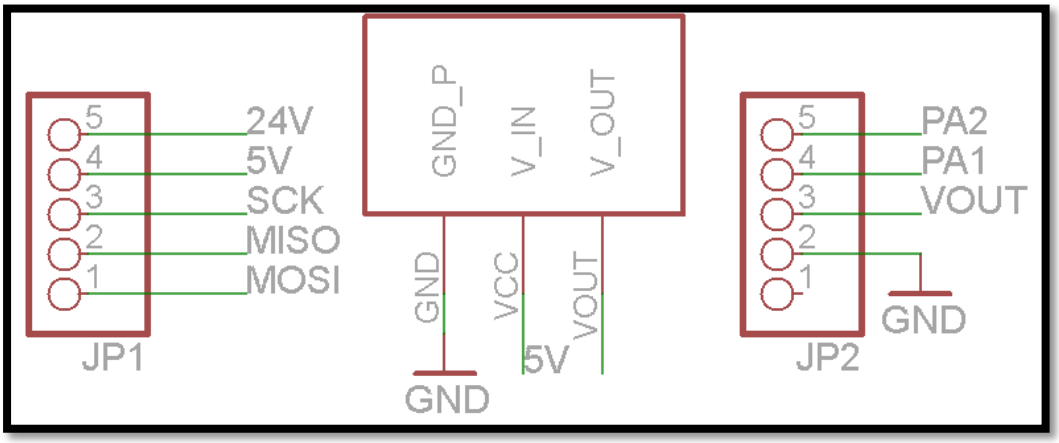
płytka czołowa



mikrokontroler



czujnik



Fizycznie zrealizowany został jeden układ elektroniczny: płytki czołowej.

spis wykorzystanych części:

Element	Wartość/model	ilość
Opornik smd	330Ω	18
Opornik smd	4.7kΩ	7
Kondensator smd	10nF	4
Tranzystor PNP	BC807-40	6
wyświetlacz	JZD080236SRYG-BW	2
Przycisk	3x3/9.5mm	4
Dioda LED	czerwona	6
Złącze szpilkowe „goldpin”	1x36	1
Złącze szpilkowe „goldpin”	2x5	1

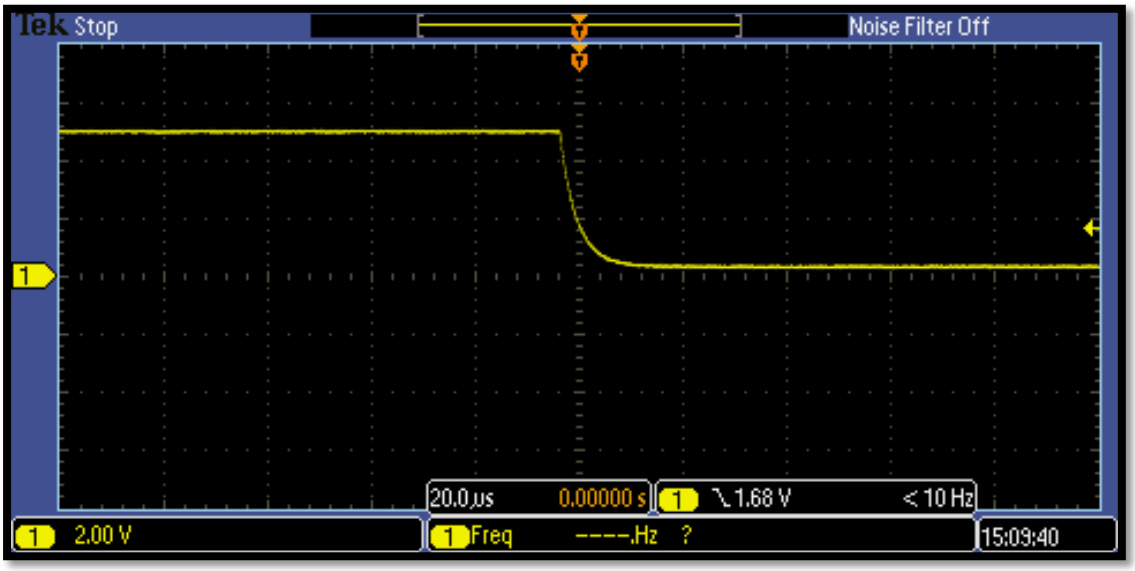
Zastosowane rozwiązania:

Usunięcie efektu drgania styków na przyciskach:
Każdy przycisk lub styki przekaźnika, mają tendencję do drgania w czasie zwierania i/lub rozwierania. Zjawisko to jest mechaniczne i zwane jest drganiem styków (ang. Bouncing). Zjawisko to trwa od kilku do kilkudziesięciu milisekund, zarówno przy zwieraniu jak i rozwieraniu styków
typowe objawy drgania styków:

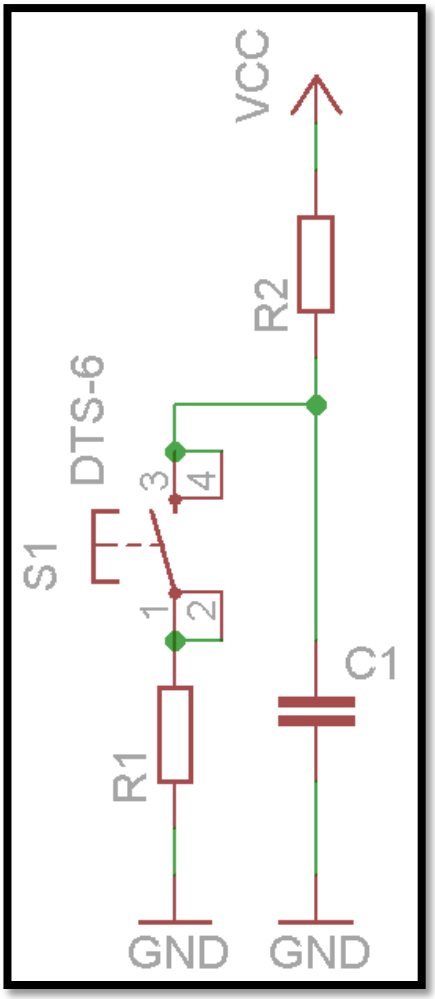
- przewijanie menu o kilka pozycji
- niepoprawna inkrementacja liczb (przeskok o kilka liczb)

Wszystkie konsekwencje związane są z kilkakrotnym naciśnięciem przycisku podczas gdy teoretycznie naciskamy przycisk jeden raz.

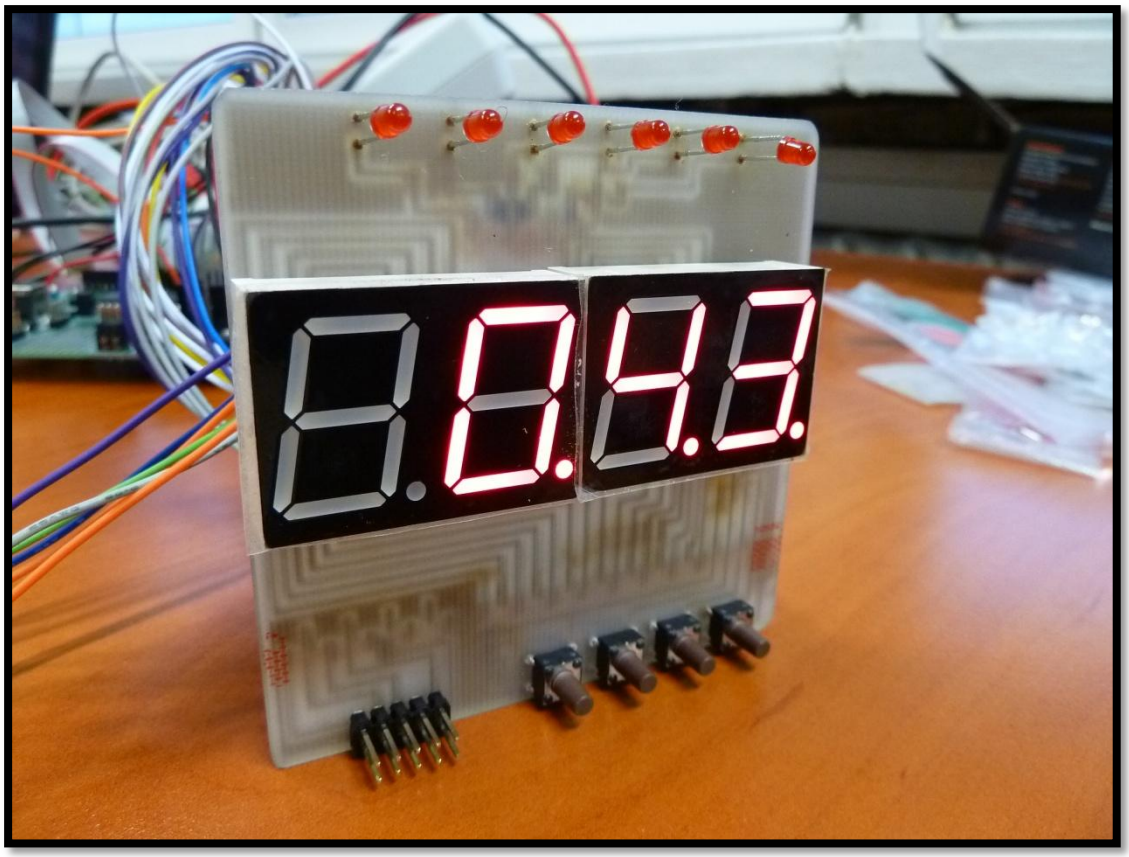
Rozwiązaniem jakie zastosowaliśmy było podłączenie kondensatora równolegle do przycisku. Kondensator wygładza sygnał podawany do mikrokontrolera.
Poniższy wykres przedstawia przebieg sygnału z zastosowaniem kondensatora:

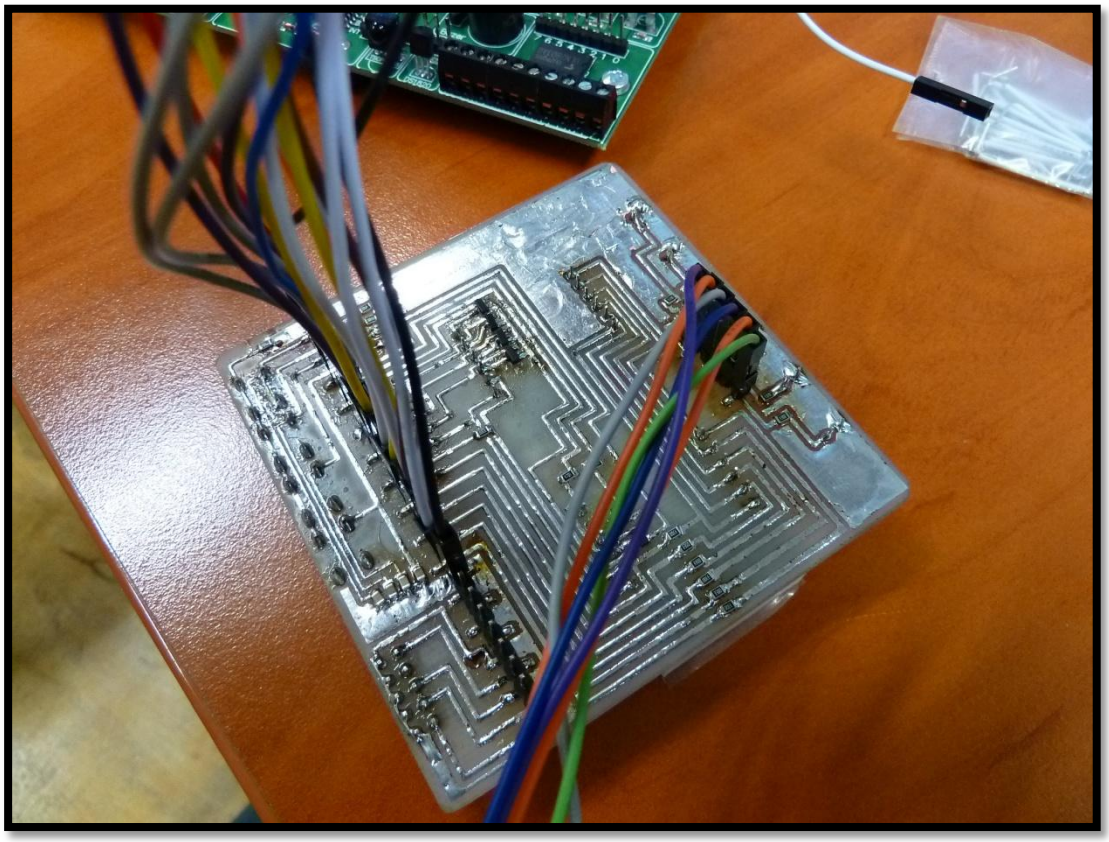


A oto schemat zastosowanego rozwiązania (występuje również na schemacie płytki czołowej oraz płytki mikrokontrolera dla przycisku RESET):



Zdjęcia płytki czołowej





Korzystając z płytki uruchomieniowej z mikrokontrolerem ATMEGA, panelu czołowego, stworzonego oprogramowania, oraz czujnika, przeprowadziliśmy wstępne testy pomiarowe, na podstawie których wnioskujemy, że wyżej przedstawiony projekt elektroniki, może być realizowany ze skutkiem pozytywnym na potrzeby stanowiska pomiarowego.

6. Wymagane funkcjonalności sterownika

Sterownik Atmega32 wykorzystany w tym projekcie został przystosowany do obsługi presostatu.
Jest w stanie zidentyfikować sygnały przesyłane do niego przez czujniki monitorujące ciśnienie w komorze hiperbarycznej. Poszczególne wysokości napięć informują go o aktualnym stanie ciśnienia. Na podstawie tych informacji jest w stanie zwiększyć lub zmniejszyć to ciśnienie.
Zewnętrzny kontakt z mikroprocesorem został rozwiązany przez zainstalowanie 4 guzików. Dzięki nim, użytkownik może ustawić histerezę, zadane ciśnienie, zmieniać je, a obserwuje wszystko na trójsegmentowym wyświetlaczu LED, również obsługiwanym przez procesor. Przyciski pozwalają na ustawienie zadanego ciśnienia oraz dopuszczalnego odchylenia od niego. Na wyświetlaczu wyświetlane będą w zależności od wybranego trybu aktualne ciśnienie lub ciśnienie zadane przez użytkownika pompy.

7. Schemat działania mikrokontrolera w sterowniku

Jeśli ciśnienie w komorze jest zbyt niskie, mikrokontroler wysyła sygnał do pompy aby je zwiększyła. Gdy jest zbyt wysokie, ciśnienie z komory zostaje wypompowywane przez pompę po otrzymaniu odpowiedniego sygnału od mikroprocesora. Celem tego działania jest utrzymanie idealnego poziomu ciśnienia.

Owo działanie, wartości graniczne, wykorzystywane przez mikroprocesor podczas monitoringu stanu w komorze, mogą być edytowane przez użytkownika.

Program:
Poprzez podprogram wykonywany jest pomiar napięcia przy pomocy wbudowanego modułu ADC, a następnie w programie następuje przeliczenie go na odpowiadające mu ciśnienie. Poprzez podprogram w przerwaniu wewnętrznym wartość pomiaru wyświetlana jest na wyświetlaczu. Oprócz tego w programie głównym wykorzystywane są podprogramy do obsługi przycisków służących zwiększaniu i zmniejszaniu ciśnienia zadanego. Regulacja ciśnienia polega na podawaniu sygnału o określonej wartości logicznej, gdy ciśnienie spadnie poniżej pewnej wartości, w celu włączenia pompy, oraz zmianę tego sygnału celem wyłączenia pompy, gdy osiągnięty zostanie górny próg przedziału, co również wykonywane jest w pętli głównej programu.

8. Funkcja obsługująca wyświetlacz segmentowy

Wyświetlacz LED ma wspólną dla wszystkich diod anodę i osobno wyprowadzonej dla każdej diody katodę (wyświetlacz segmentowy składa się z trzech wyświetlaczów LED). Zasada działania opiera się na tym, że ludzkie oko nie jest w stanie zauważyć bardzo szybkich zmian. Najpierw jest wyświetlana pierwsza liczba, następnie druga, a potem ostatnia. Czynność ta jest wykonywana kilka razy w pętli. Wspólna anoda dla wszystkich diod jednego wyświetlacza LED jest stosowana w celu zaoszczędzenia wyjść mikrokontrolera. Do realizacji tego zadania wykorzystana została procedura przerwania wewnętrznego wykonująca się z częstotliwością 150Hz.

Plik nagłówkowy kolorowy.h:

```
#ifndef KOLOR_H_
#define KOLOR_H_

//segmenty

#define SA (1<<0)
#define SB (1<<1)
#define SC (1<<2)
#define SD (1<<3)
#define SH (1<<4)
#define SF (1<<5)
#define SG (1<<6)
#define SDP (1<<7)
//stałe preprocesora
#define LED_DATA            PORTB //port z segmentami
#define LED_DATA_DIR    DDRB
#define ANODY_PORT          PORTA //port z anodami
#define ANODY_DIR           DDRA
//bity na których znajdują się anody zielone (obecnie na porcie A)
#define CA1             (1<<PA0)
#define CA2             (1<<PA1)
#define CA3             (1<<PA2)
//bity z anodami czerwonymi (port A)
#define CA4             (1<<PA3)
#define CA5             (1<<PA4)
#define CA6             (1<<PA5)

//deklaracja zmiennej będącej wyświetlaną liczbą
extern volatile int liczba;
extern volatile int kolor; //zmienna odpowiadająca za kolor (0 - zielony, 1 - czerwony)
//deklaracja funkcji inicjującej działanie z wyświetlacza
void d_led_init(void);
#endif
```

plik źródłowy kolorowy.c:

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include "kolorowy.h"

//volatile uint8_t cyfra1;
//volatile uint8_t cyfra2;
//volatile uint8_t cyfra3;
volatile int liczba;
volatile int kolor;

uint8_t cyfry[15] =		//zostawiam miejsce w tablicy na więcej znaków
{
	~(SA|SB|SC|SD|SH|SF),		//0
	~(SB|SC),
	~(SA|SB|SD|SH|SG),
	~(SA|SB|SC|SD|SG),
	~(SB|SC|SF|SG),
	~(SA|SC|SD|SF|SG),		//5
	~(SA|SC|SD|SF|SG|SH),
	~(SA|SB|SC|SF),
	~(SA|SB|SC|SD|SH|SF|SG),
	~(SA|SB|SC|SD|SF|SG),		//9
	~(SDP),		//kropka
};


void d_led_init(void)
{
	LED_DATA_DIR = 0xFF;	//ustawienie segmentów (wszystkie na stan wysoki)
	LED_DATA = 0xFF;

	ANODY_DIR |= CA1 | CA2 | CA3 | CA4 | CA5 | CA6;		//ustawienie anod
(wszystkie na stan wysoki)
	ANODY_PORT |= CA1 | CA2 | CA3 | CA4 | CA5 | CA6;

	TCCR0 |= (1<<WGM01);		//tryb CTC
	TCCR0 |= (1<<CS02)|(1<<CS00);		//preskaler na 1024
	OCR0 = 52;
	TIMSK |= (1<<OCIE0);
}

ISR(TIMER0_COMP_vect)
{
	static uint8_t licznik=1;
	//wygaszenie anod:
	ANODY_PORT |= 0xFF;
	LED_DATA |= 0xFF;


	if(kolor==0){ANODY_PORT= ( ANODY_PORT & 0b11111000 ) | ( ~licznik &
0b00000111 );}		//wł anody kolor zielony
	else if(kolor==1){ANODY_PORT= ( ANODY_PORT & 0b11000111 ) | ( ~(licznik*8) &
0b00111000 );} //wł anody kolor czerwony
```

```
        int cyfra3=liczba%10;
        int cyfra2=(liczba%100-cyfra3)/10;
        int cyfra1=(liczba%1000-cyfra3-cyfra2)/100;

        if(licznik==1)      LED_DATA = cyfry[cyfra1];
        else if(licznik==2) LED_DATA = cyfry[cyfra2];
        else if(licznik==4) LED_DATA = cyfry[cyfra3];

        licznik <<= 1;
        if(licznik>4) licznik = 1;
}
```

9. Opracowanie funkcji do odczytu danych z czujników

Mikroprocesor Atmega32 otrzymuje od czujników różne poziomy napięć. Interpretuje je prawidłowo na podstawie informacji zawartych w oprogramowaniu wgranym do niego przez użytkownika.
Podprogram realizujący pomiar przy pomocy modułu ADC:

```
void ADC_init()
{
ADMUX |= (1<<REFS0) | (1<<ADLAR);
ADCSRA |= (1<<ADEN) | (1<<ADPS1) | (1<<ADPS0);
}
void zmierz (uint8_t wejscie)
{
        ADMUX |= (ADMUX & 0xF8) | wejscie;
        ADCSRA |= (1<<ADSC);
        while( ADCSRA & (1<<ADSC) );
        pomiar=ADCH;
}
```

Program realizuje pomiar napięcia z ośmiobitową rozdzielczością. Napięcie mierzone jest względem napięcia 5V podłączonego pod pin AREF i podawane w postaci liczby 0-255, co pozwala przeliczyć je na wartości od 0V do 5V.

10. Implementacja algorytmu sterowania

Sterownik działa na następującej zasadzie:

- ciśnienie przeliczone z podanego napięcia porównywane jest z zadanym.
- jeżeli ciśnienie zmierzone znajduje się poniżej zadanego zakresu, sterownik podaje sygnał włączający pompę, a zmienna flagowa ustawiana jest na „1”
- pompa działa, dopóki nie zostanie osiągnięty szczyt zadanego przedziału, a kiedy to się stanie pompa jest wyłączana, a zmienna flagowa ustawiana jest na wartość „0”, co pozwala opaść ciśnieniu na dół przedziału bez ponownego włączania pompy.
- cykl jest powtarzany

11. Podsumowanie

Projekt został ukończony w przewidzianym terminie, aktualny stan pozawala na rozpoczęcie pracy nad realizacją fizycznego stanowiska sterowania komorą.
Dalsze kroki jakie planujemy podjąć jest dopracowanie sterownika, oraz stworzenie podzespołów elektronicznych, na podstawie przygotowanych schematów, które będą wykorzystane przy budowie stanowiska sterowania.
Dzięki realizacji tego projektu poznaliśmy podstawy programowania mikrokontrolerów, zasady budowy płytek elektronicznych, wraz z całkowitym procesem ich tworzenia, zdobyliśmy również cenne doświadczenie mogąc pracować w grupie.