

# **Kalkulator AVR**

**Raport i Dokumentacja**

Piotr Śmieja, 2019

# Spis Treści

<b>Wstęp.....</b>	<b>3</b>
<b>Wykorzystane Elementy.....</b>	<b>3</b>
<b>Schemat.....</b>	<b>3</b>
<b>Rejestry.....</b>	<b>5</b>
Rejestry wykorzystywane do sterowania wyświetlaczem.....	5
Rejestry wykorzystywane do obsługi klawiatury.....	5
<b>Działanie programu i układu.....</b>	<b>6</b>
Sterowanie Wyświetlaczem.....	6
Działanie Klawiatury.....	7
<b>Możliwe poprawki.....</b>	<b>7</b>
<b>Informacje Dodatkowe.....</b>	<b>7</b>

# Wstęp

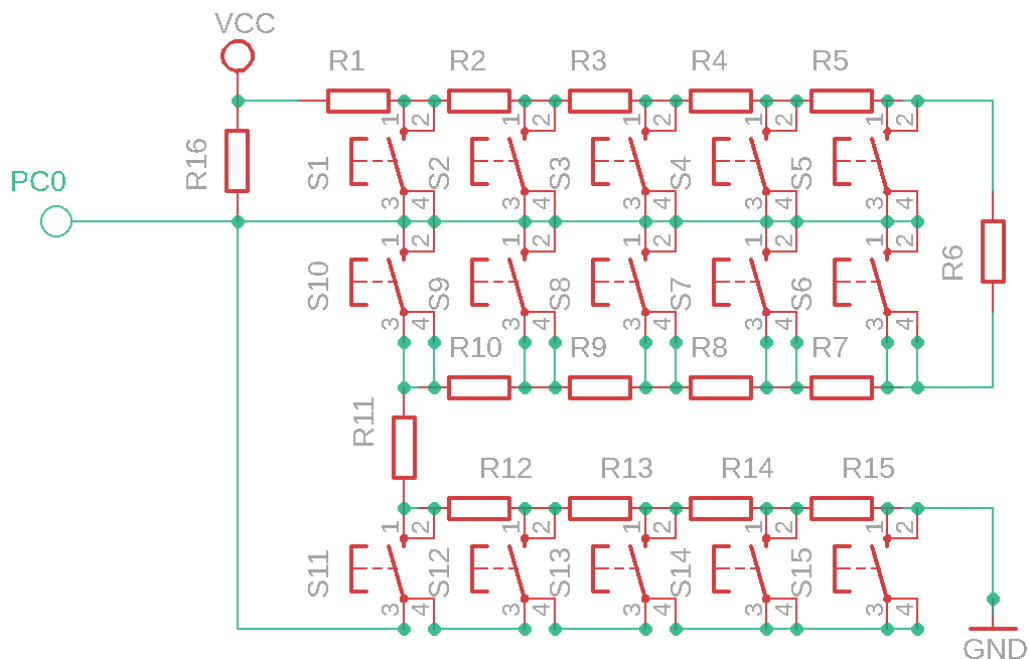
Opisany układ wykonany został na zaliczenie przedmiotu Programowanie Mikrokontrolerów na Wydziale Fizyki Uniwersytetu Warszawskiego. Stanowi on prosty kalkulator wykonujący operacje dodawania i odejmowania, działający dla dowolnych nieujemnych liczb całkowitych mniejszych od 10000.

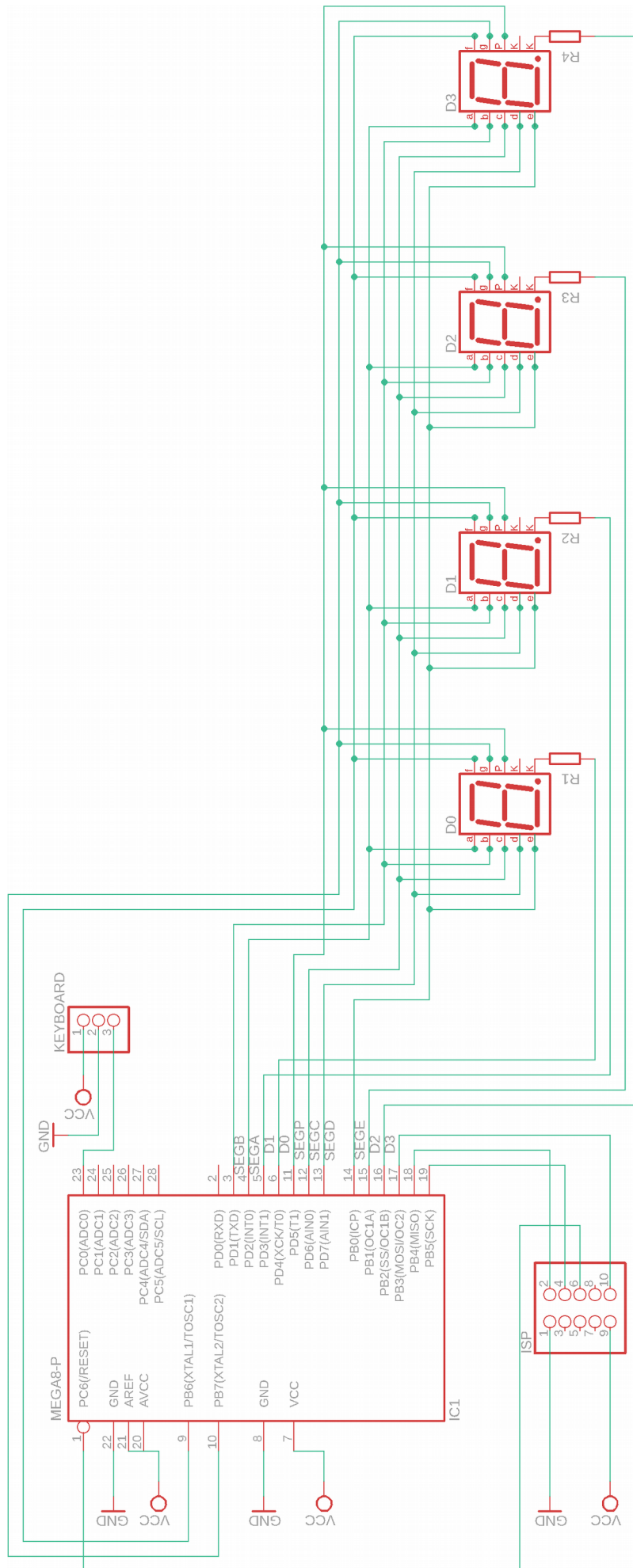
## Wykorzystane Elementy

Układ został zbudowany w oparciu o mikrokontroler ATmega8. Wykorzystano dwie uniwersalne płytki drukowane, przewody, oraz dodatkowe elektryczne elementy przewlekane (rezystory, przełączniki tact switch, etc.).

## Schemat

Poniżej przedstawiono schemat głównego obwodu oraz umieszczonej na osobnej płytce klawiatury:





# Rejestry

## Rejestry wykorzystywane do sterowania wyświetlaczem:

PORTB:

G	F				DISP 3	DISP 2	E
7	6	5	4	3	2	1	0

PORTD:

D	C	DP	DISP 0	DISP 1	A	B	
7	6	5	4	3	2	1	0

Bity zaznaczone jako A, B, C, D, E, F, DP odpowiadają za poszczególne segmenty wyświetlacza (DP to kropka), natomiast bity DISP 0-3 odpowiadają za konkretne wyświetlacze 8-segmentowe (konkretne cyfry wyświetlanej liczby).

## Rejestry wykorzystywane do obsługi klawiatury

PORTC:

							KEY
7	6	5	4	3	2	1	0 (ADC0)

Do obsługi klawiatury wykorzystywany jest wyłącznie jeden pin mikrokontrolera ( PC0 ). Stanowi on wejście przetwornika analogowo-cyfrowego ADC0.

# Działanie programu i układu

## Sterowanie Wyświetlaczem

Cały wyświetlacz urządzenia składa się z czterech multipleksowanych wyświetlaczy 8-segmentowych. Wykorzystane wyświetlacze 8-segmentowe posiadają po 10 wyjść. Osiem z nich (oznaczanych od A do F oraz DP jako kropka) odpowiada ośmiu segmentom wyświetlacza, a dwa połączone są ze wspólną katodą wyświetlacza. Każdemu z segmentów odpowiada określony bit z rejestru PORTB albo PORTD zgodnie z informacją podaną w poprzedniej części.

Żeby zatem zaświecić określony segment konkretnego wyświetlacza należy podać VCC na wyjście przypisane temu segmentowi oraz GND na wyjście common (katoda) charakterystyczne dla ustalonego wyświetlacza. Sytuacja jest inna jedynie jeżeli wyjście common danego wyświetlacza podłączone jest przez tranzystor. Wtedy również to wyjście należy ustawić na stan wysoki. W przypadku naszego urządzenia wyświetlacze podłączone są jedynie przez rezystor.

**UWAGA: W obecnej wersji urządzenia wyświetlacz #0 podłączony został przez tranzystor. W jego przypadku zatem w celu włączenia ustawiany jest stan wysoki, a w celu wyłączenia niski (co sprawdzić można w kodzie programu).**

Maski PORTB\_MASK i PORTD\_MASK mają 0 na bitach odpowiadających za segmenty wyświetlacza i 1 na pozostałych bitach:

```
#define PORTB_MASK 0x3E
#define PORTD_MASK 0x19
```

W ten sposób poprzez operację `PORTi &= PORTi_MASK` zerujemy wszystkie bity segmentów w rejestrze `PORTi`.

Dla każdej cyfry `i` z przedziału od 0 do 9, jak i również każdego rejestru `j` z obu rejestrów PORTB i PORTD ustawiono maskę `ji_MASK` w której jedyne ustawione są na bitach sterujących tymi segmentami, które powinny zostać zaświecone. Po wyzerowaniu więc wszystkich bitów segmentów operacją `PORTj &= PORTj_MASK`, operacją `PORTj |= ji_MASK` ustawiamy już ostatecznie na wyświetlaczu `j` segmenty odpowiadające cyfrze `i`. Oczywiście, jako że bity segmentów występują w obu rejestrach, konieczne jest w rzeczywistości wykonanie czterech operacji; przykładowo, dla cyfry 4:

```
PORTB &= PORTB_MASK;
PORTB |= B4_MASK;
PORTD &= PORTD_MASK;
PORTD |= D4_MASK;
```

Przed ustawieniem określonej cyfry wybierany jest jeszcze wyświetlacz. W tym celu należy włączyć ustalony wyświetlacz (wszystkie wyświetlacze włączane są po kolei w pętli) i wyłączyć pozostałe.

## Działanie Klawiatury

Na głównej linii klawiatury prowadzącej od zasilania do masy jest 15 rezystorów. Za każdym z nich główną linię połączyć można przełącznikiem z linią sygnałową, co daje na linii sygnałowej określone dla danego punktu linii głównej napięcie. Napięcie dla każdego punktu zostało wcześniej obliczone i dodatkowo skalibrowane doświadczalnie. Żeby uniknąć stanów nieustalonych gdy nie jest naciśnięty żaden przycisk, linia jest dodatkowo połączona z zasilaniem opornikiem o wysokiej rezystancji. Wyjście z klawiatury podłączone jest do pinu przetwornika analogowo-cyfrowego w mikrokontrolerze. Różnica wartości napięcia i napięcia odniesienia (VCC) jest odpowiednio przeskalowana i zwracana przez funkcję `adcRead()`. Wartość tej funkcji porównywana jest w pętli do kolejnych wartości granicznych z tablicy. Po dopasowaniu do konkretnej wartości zwracany jest przez funkcję `pressedKey()` kod przycisku na który odpowiednio zareagować może funkcja `main()`.

## Możliwe poprawki

Istnieje możliwość zamontowania przy wszystkich wyświetlaczach tranzystorów (testowo tranzystor został wlutowany jedynie przy wyświetlaczu #0), przez które sterowanoby wyświetlaczami obniżając jednocześnie pobór mocy z mikrokontrolera. W wykonanym w ramach projektu układzie nie umieszczono również odrębnego zasilania i dotychczas kalkulator zasilany był przez piny programatora. Docelowo wskazanym byłoby wykonanie dedykowanej płytki drukowanej na której znalazłby się cały układ wraz z klawiaturą i zasilaniem baterijnym.

## Informacje Dodatkowe

Opis funkcji i zmiennych niewymienionych w dokumentacji zawarty jest w kodzie źródłowym. Dodatkowo na stronie projektu zamieszczono zdjęcia gotowego układu.