

Analizator widma dźwięku w formie trójwymiarowej matrycy led

Jakub Dembiński, Mateusz Jachniak

Maj 2019

Spis treści

1	Wprowadzenie	2
1.1	Cel projektu	2
1.2	Wymagania	2
1.3	Krótki opis	2
2	Specyfikacja	3
3	Komponenty	4
4	Architektura	5
4.1	Software	5
4.2	Hardware	6
4.2.1	Matryca	6
4.2.2	Tranzystory	7
4.2.3	Przedwzmacniacz	8
4.2.4	Atmega	9
5	Integracja systemu	10
6	Komunikacja systemu	11
6.1	Opis algorytmów wyświetlania	11
6.1.1	Algorytm 1 - Zapalanie pojedynczej diody w każdym słupku odpowiadającym danej częstotliwości oraz 4 kolejnych w głąb	11
6.1.2	Algorytm 2 - Zapalanie krzywej	13
6.1.3	Algorytm 3 - to wzbogacony algorytm pierwszy o zapalenie się diod wszystkich do danego poziomu	13
6.1.4	Algorytm 4 - wzbogacony algorytm 3 o zapalenie się diod wszystkich do danego poziomu	13
7	Podzespoły i ich koszt	14
8	Bibliografia	15

1 Wprowadzenie

1.1 Cel projektu

Głównym celem projektu analizatora widma dźwięku w formie trójwymiarowej matrycy led jest rozwój umiejętności z zakresu systemów wbudowanych, zarówno w dziedzinie hardwerowej, softwerowej jak i planowania architektury systemów. Ponadto będzie to idealny element portfolio, a efekt końcowy na pewno będzie prezentowany przy każdej możliwej okazji [1].

1.2 Wymagania

- Wizualizacja widma dźwięku
- Korekcja szumów (zastosowanie okienkowania)
- Bezpieczeństwo w użyciu
- Stabilna konstrukcja kostki ledowej

1.3 Krótki opis

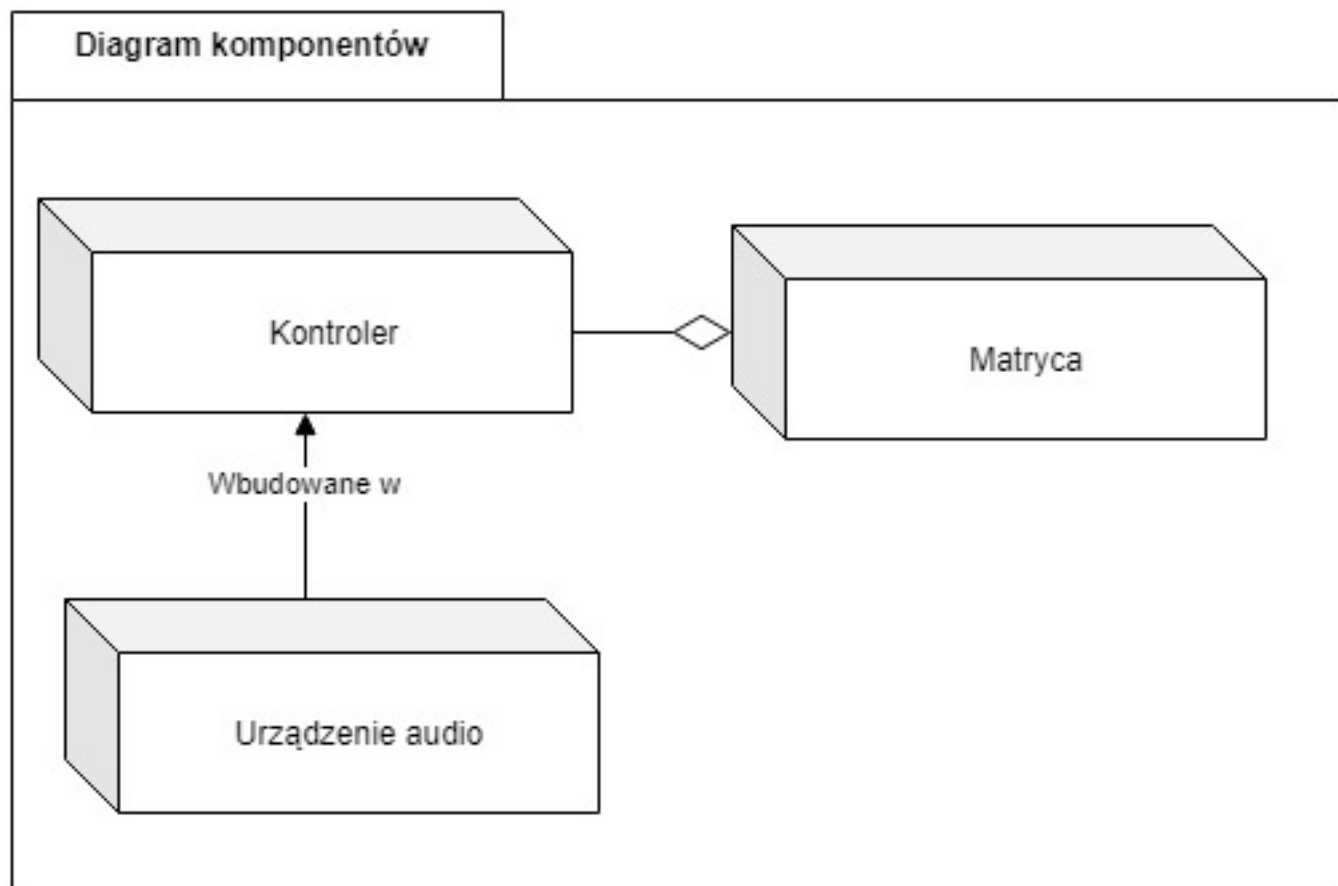
Jak sama nazwa wskazuje, efektem końcowym będzie trójwymiarowa matryca, oprogramowana tak, aby w zależności od otrzymanego fal dźwiękowych była w stanie zapalać konkretne diody, w odpowiedni sposób (opisany poniżej w specyfikacji projektu). Kostka musi być w stanie ustać o własnych siłach, aby można było ją bezproblemowo prezentować oraz ewentualnie przemieszczać. Ostatecznie, nie ma ona służyć jakimś wyższym celom, ma być bardziej jako dodatek do wystroju pokoju. Wszystkie materiały zostały zakupione w sklepach, jednak wszystko zamontowane własnoręcznie, bez pomocy innych działalności. Wszystkie kody źródłowe, są naszego autorstwa, za wyjątkiem algorytmu służącego do obliczenia Dyskretnej Transformaty Fouriera [2].

2 Specyfikacja

- Wymiary (podane w centymetrach): 39x32x22(szer. wys, gł.)
- Tryby wyświetlenia:
 - słupkowo
 - liniowo
 - po krzywej
- Animacja włączania
- Tryb animacji bez analizy dźwięku
- Rozdzielczość matrycy: 2 dpi
- Kolor matrycy : niebieski
- Wejście audio: mikrofon pojemnościowy
- Pasmo przenoszenia: 0 - 20 000 Hz (ludzki słuch)
- Zasilanie: 5V
- Wejście zasilania: złącze dc

3 Komponenty

Poniżej znajduje się prosty diagram komponentów obrazujący sposób połączenia elementów.

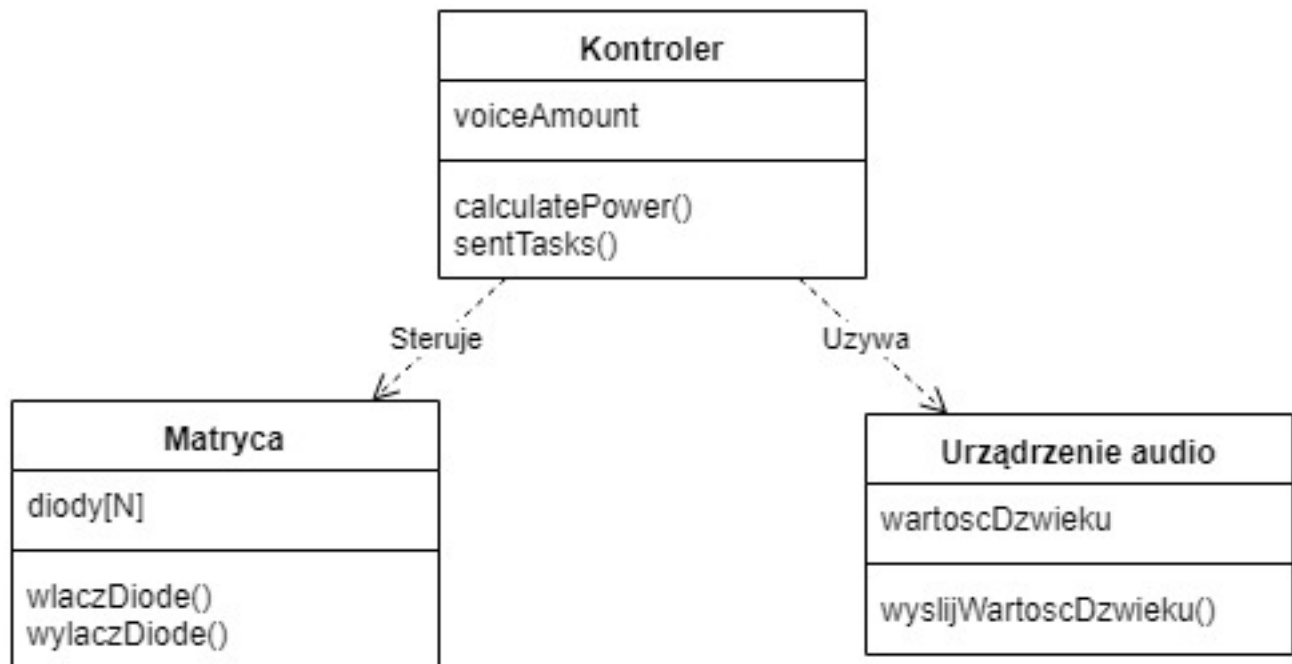


Rysunek 1: Diagram Komponentów [3]

4 Architektura

4.1 Software

Diagram klas przedstawiający zależności w oprogramowaniu.

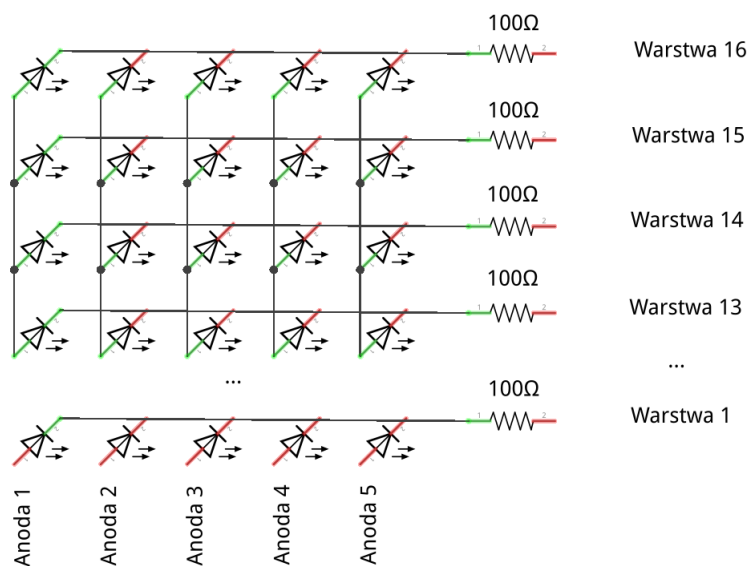


Rysunek 2: Diagram klas [3]

4.2 Hardware

4.2.1 Matryca

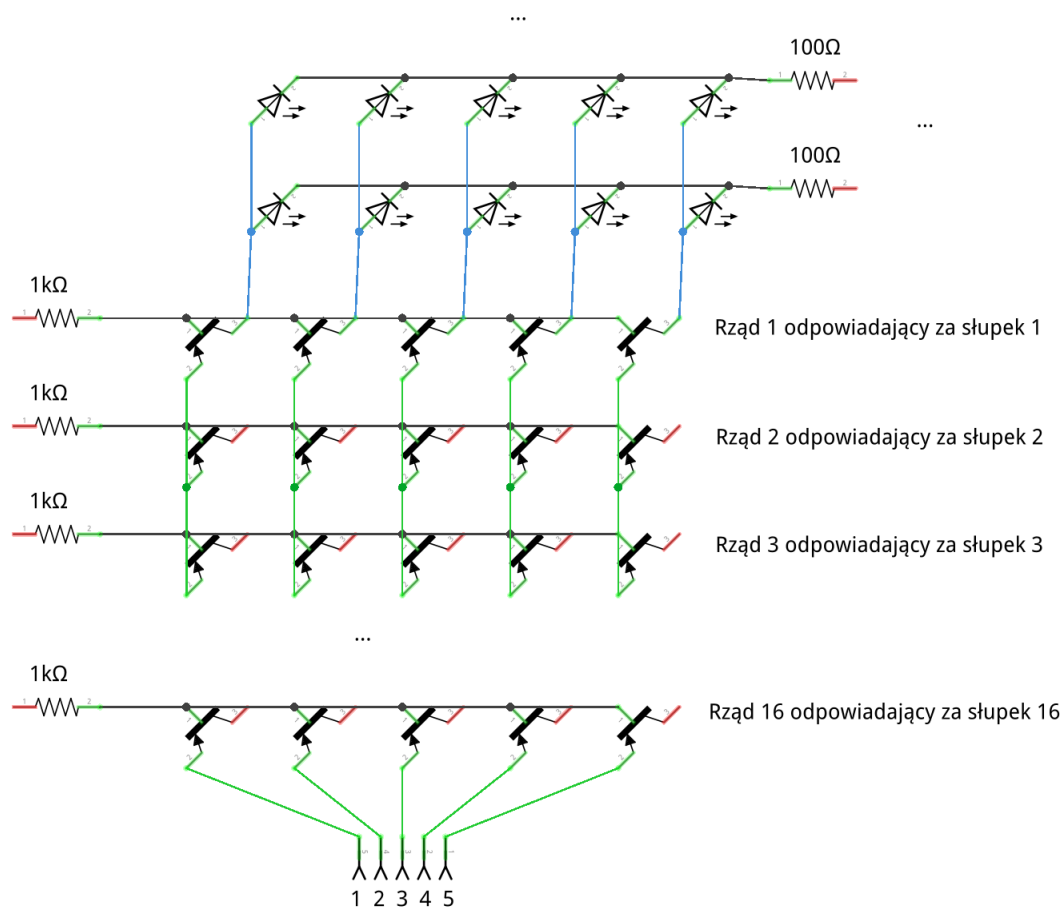
Schemat przedstawia budowę matrycy. Od dołu znajduje się podłączenie do modułów tranzystorów, natomiast po prawej stronie jest podłączenie do mikrokontrolera Amega32 (sterującymi uziemieniami warstw).



Rysunek 3: Połączenie matrycy [4]

4.2.2 Tranzystory

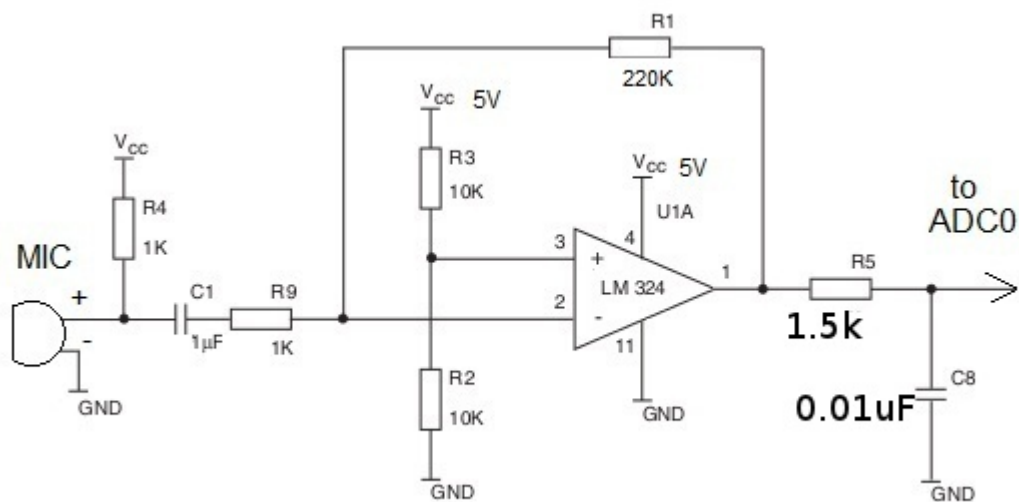
Schemat przedstawia widok na tranzystory. Od dołu znajdują się linie napięcia odpowiadające za "głębokość" matrycy, które są podłączone do głównego kontrolera Amega32. Po lewej stronie są bazy tranzystorów podłączone do głównego mikrokontrolera Atmega32 (służące do wybierania danego słupka). Kolektor każdego tranzystora jest podłączony do anody jednej diody w następujący sposób: pierwszy rząd tranzystorów do pierwszego słupka tak, że pierwszy tranzystor w rzędzie do pierwszej diody w "głębokości" patrząc od przodu.



Rysunek 4: Połączenie tranzystorów [4]

4.2.3 Przedwzmacniacz

Moduł przekształcający fale dźwiękową w sygnał analogowy.



Rysunek 5: Przedwzmacniacz [3]

4.2.4 Atmega

Poniżej znajduje się schemat zasilania i połączenia ze sobą dwóch Atmeg. Aby rysunek był czytelny, zostały pominięte podłączenia z tranzystorami i matrycą.

Porty Atmegi głównej:

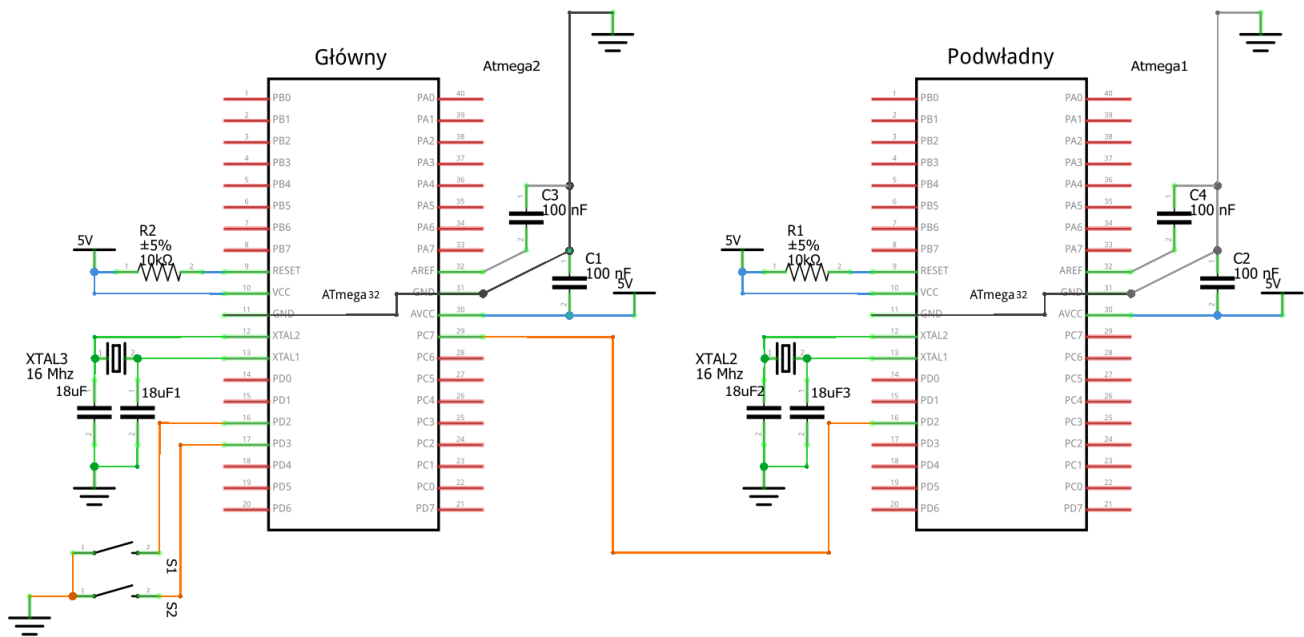
PA0 - Wejście z modułu przedwzmacniacza

PB0, PB1, PB2, PB3 i PB4 - do linii zasilających tranzystory

PB5, PB6, PB7, PD0, PD1, PD4, PD5, PD6, PD7, PC0, PC1, PC2, PC3, PC4, PC5, PC6 - do baz tranzystorów tak, że odpowiadają kolejno za poszczególne słupki matrycy.

Porty Atmegi podwładna:

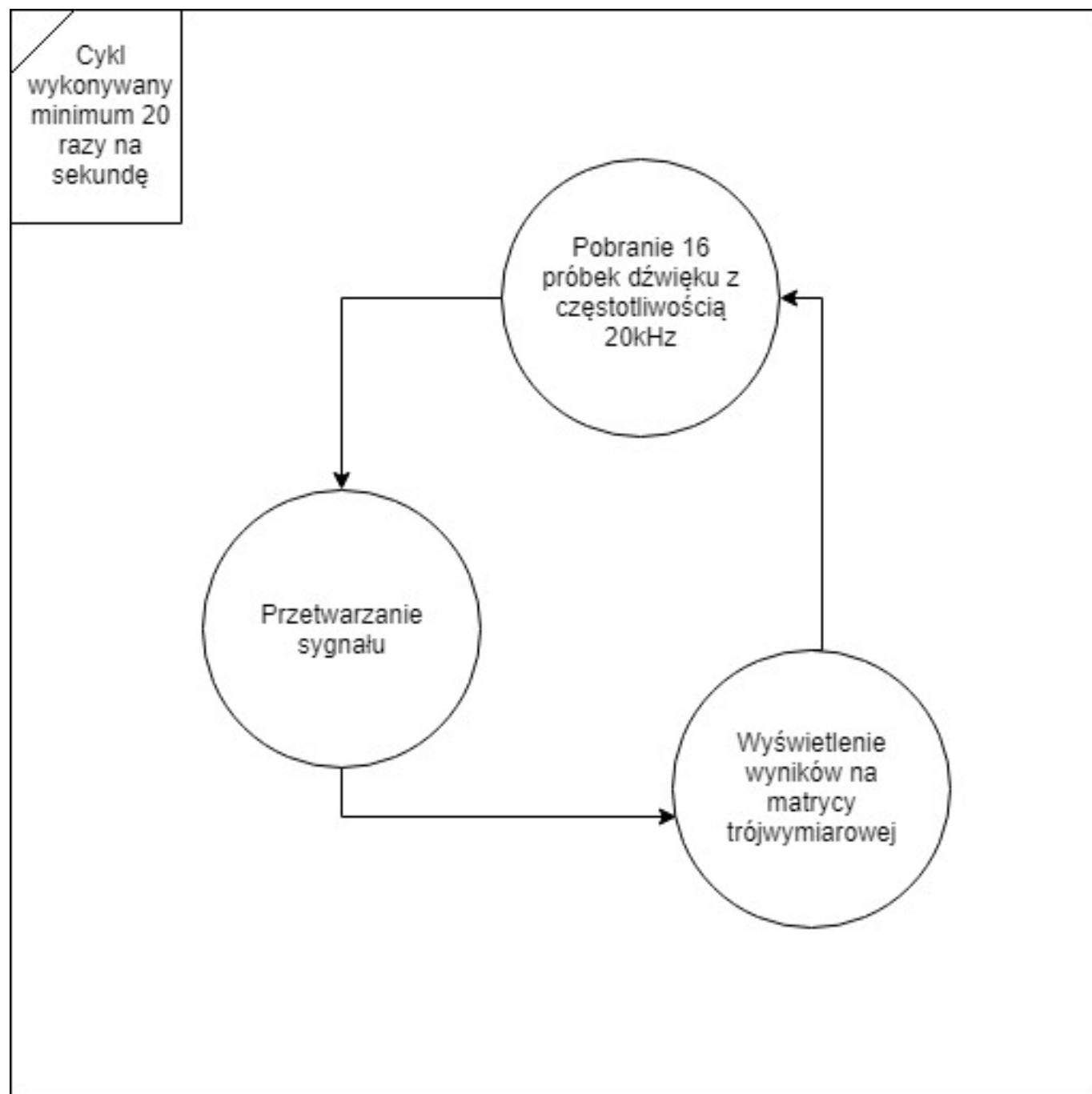
PB0, PB1, PB2, PB3, PB4, PB5, PB6, PB7, PD0, PD1, PD4, PD5, PD6, PD7, PC0, PC1 - do poszczególnych katod matrycy tak, że odpowiadają za uziemianie poziomów od dolnego do górnego



Rysunek 6: Schemat zasilania [4]

5 Integracja systemu

Diagram stanów przedstawiający przepływ informacji między elementami systemu.



Rysunek 7: Diagram stanów [3]

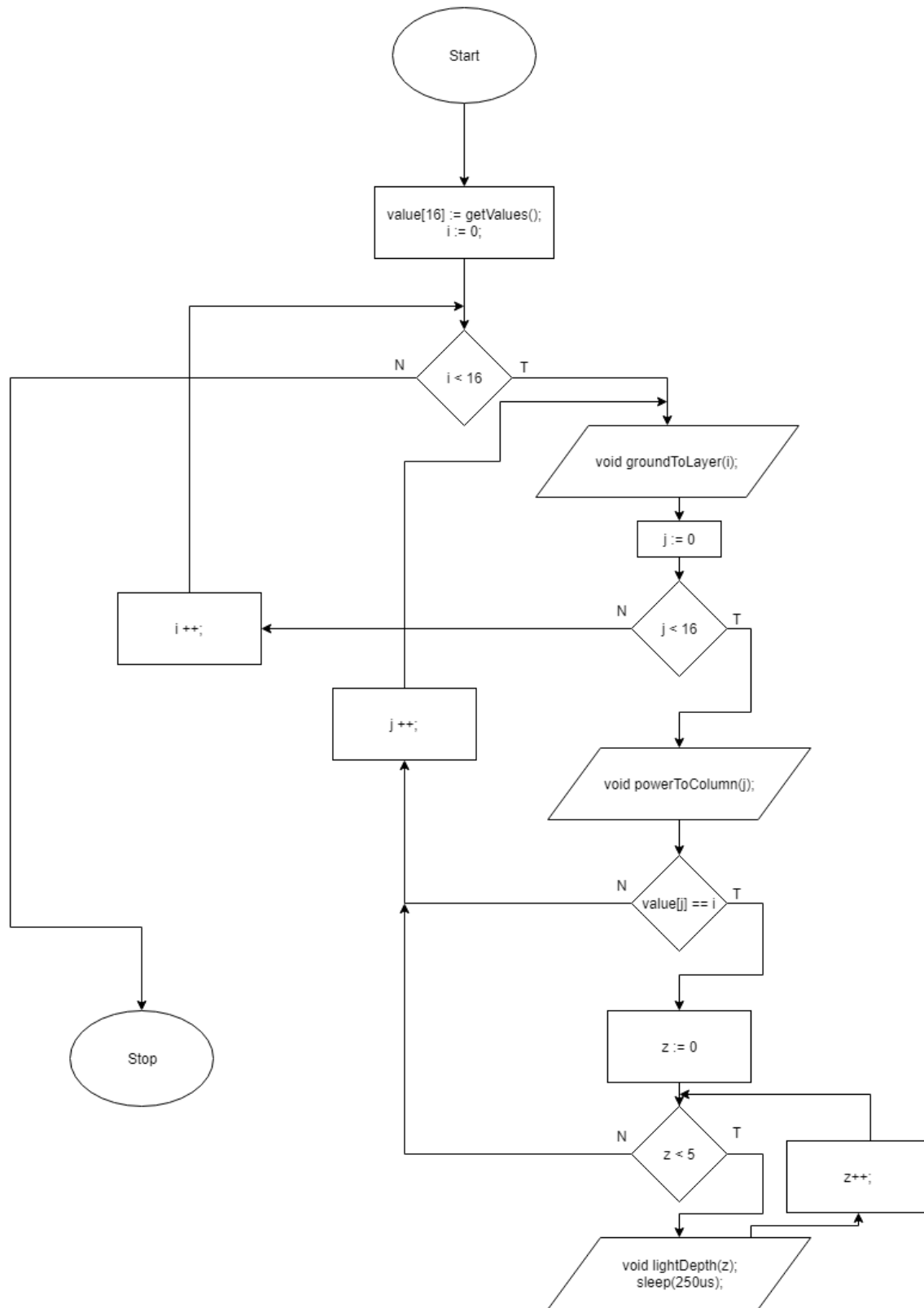
6 Komunikacja systemu

Cały program jest wykonywany w nieskończonej pętli, czyli bez warunku końca. Próbkowanie dźwięków odbywa się z częstotliwością 20kHz po 16 próbek, czyli jedna próbka co 125 micro sekund. Okienkowanie weryfikacji odbywać się będzie w celu ograniczeniu wpływu szumów na inne częstotliwości. Tak przyszykowane dane wykorzystujemy w algorytmie FFT. Na wyjściu dostajemy tablicę 16 elementową – wartość każdego elementu jest określeniem poziomu natężenia danej częstotliwości i jej bliskim. (zakres co 2kHz). Następnie następuje sprawdzanie największych natężeń w celu odniesienia danego poziomu do ogółu. Wyszukiwanie maximum następuje co dziesiąte próbkowanie, aby wyniki się za bardzo nie żoźpłynęły”.

6.1 Opis algorytmów wyświetlania

6.1.1 Algorytm 1 - Zapalanie pojedynczej diody w każdym słupku odpowiadającym danej częstotliwości oraz 4 kolejnych w głąb

Mamy tablicę wyjściową z algorytmu FFT odpowiednio wyskalowaną tak by największe natężenie mieściło się na matrycy. Zaczynamy od wyzerowania licznika wyświetleń. Następnie w pętli po poziomej ścianie (niech ma wymiary "i"=0..15 i "j"=0..15) sprawdzamy czy wartość w "j-tym elemencie jest równa wartości "i". Jeśli tak to inkrementujemy licznik wyświetleń oraz zapalamy w kolejnej pętli z iteratorem żód 0 do 4 (wprowadzamy trzeci wymiar - ż") włącznie o współrzędnych [j-1][i-1][z] na 250 micro sekund po czym ją gasimy. Po każdej iteracji pętli z iteratorem "i"zmieniamy stan pinu, do którego jest podłączone drugi mikrokontroler, który po otrzymaniu przerwania zmienia warstwę diod, których katody uziemia. (Drugi mikrokontroler działa na zasadzie pętli, tj. po otrzymaniu przerwania zmienia uziemienie po kolei od 1 do 16 warstwy a po 16 wraca do 1). Po ostatniej iteracji pętli z iteratorem "i"ponownie dokonujemy próbkowania. Samo próbkowanie zabiera około 125 micro sekund. Wyświetlanie za pomocą Algorytmu 1 zajmuje około 25 ms. Zatem jeden przebieg pełnej iteracji nieskończonej pętli zajmuje około 0,025125 s, co daje odświeżanie matrycy z częstotliwością około 39Hz. Poniżej zostaje przedstawiony schemat blokowy algorytmu pierwszego. Aby zachować czystość dokumentu, nie wstawialiśmy innych schematów blokowych, ponieważ reszta algorytmów działa na bardzo zbliżonej zasadzie.



Rysunek 8: Schemat blokowy algorytmu pierwszego [3]

6.1.2 Algorytm 2 - Zapalanie krzywej

Najpierw tablicę wyjściową z FFT skalujemy tym razem nie na 16 wartości na 21. Wyświetlanie odbywa się podobnie jak w Algorytmie 1 z tą różnicą, że w iteracji po pętli z "j" warunkiem zapalenia diody jest liczba równa "j-5 dla pierwszego i drugiego poziomu głębokości, "j-4 dla trzeciego, "j-3 i "j-2 dla czwartego oraz "j-1 i równe "j" dla 5. Z dodatkowym warunkiem, że jeśli ta liczba jest mniejsza od zera (ostatnie dolne piętro) to traktujemy ją jako zero oraz jeśli jest większa od piętnastki (ostatnie górne piętro) to traktujemy ją jako piętnaście. Wynika z tego że na danym poziomie dla konkretnej częstotliwości zapalimy od 1 do maksymalnie 2 diod. Częstotliwość odświeżania matrycy jest zbliżona do tej dla Algorytmu 1.

6.1.3 Algorytm 3 - to wzbogacony algorytm pierwszy o zapalenie się diod wszystkich do danego poziomu

W celu utrzymania częstotliwości odświeżania na poziomie około 20Hz musimy skrócić czas zapalenia każdej diody do około 50 us co w pewnym stopniu zmniejszy jasność diod, jednak dzięki ich bezwładności nadal będziemy widzieć płynną animację. Druga zmiana w algorytmie jest taka, że warunkiem aktywacji diody jest wartość "j" nie większa niż "i";

6.1.4 Algorytm 4 - wzbogacony algorytm 3 o zapalenie się diod wszystkich do danego poziomu

Zmiany, które musimy dokonać to zmiana czasu zapalenia diody identycznie jak w Algorytmie 3 oraz zmiana znaku równości w warunku zapalenia diody na znak słabej nierówności.

7 Podzespoły i ich koszt

Wszystkie podzespoły kosztowały nas 330,26zł, a zawierało się w nich:

2 x ATMEGA32A - 27,98zł

1280 x LED 3mm niebieski - 140,16zł

80 x tranzystor PNP - 10zł

1 x taśma IDC40 - 8,70zł

1 x złącze IDC40 - 1,27zł

2 x podstawka DIL40 - 3,80zł

1 x pojedyncze złącza goldpin 40 - 1,40zł

1 x podwójne złącza goldpin 40 - 1,05zł

1 x płytki uniwersalna 9x15cm - 11,90zł

1 x wzmacniacz LM321M - 7zł

1 x mikrofon - 2zł

1 x złącze Jack 3,5mm - 1zł

1 x złącze microUSB - 2zł

1 x rezonator kwarcowy 16MHz - 2zł

16 rezystorów 1k, 21 rezystorów 10k + kondensatory 10zł

elementy obudowy - 100zł

8 Bibliografia

- [1] Przykład podobnego projektu, wykonanego w inny sposób. <https://www.youtube.com/watch?v=Vn39txtVIHc>.
- [2] Matematyczny opis algorytmu, wykorzystywanego w projekcie do wyznaczania natężenia konkretnych częstotliwości dźwięku. https://en.wikipedia.org/wiki/Discrete_Fourier_transform.
- [3] Narzędzie do tworzenia diagramów uml. <https://www.draw.io/>.
- [4] Narzędzie do tworzenia schematów. <http://fritzing.org/home/>.