



## PTM 2

|  |  |
|--|--|
| <b>Kierunek</b><br><i>Automatyka i Robotyka</i>        | <b>Termin</b><br><i>Czwartek 8:00 TP</i> |
| <b>Temat</b><br><i>Skaner 3D</i>                       | <b>Problem</b>                           |
| <b>Skład grupy</b><br><i>Adam Jankowiak 252919</i>     | <b>Nr grupy</b>                          |
| <b>Prowadzący</b><br><i>Mgr inż. Paweł Dobrowolski</i> | <b>data</b><br><i>2 lutego 2022</i>      |

## 1 Wstęp

Główym celem projektu było stworzenie urządzenia mogącego skanować różnego rodzaju przedmioty w trzech wymiarach. Dzięki dokonanym pomiarom można odtworzyć zewnętrzną siatkę przedmiotu w różnego rodzaju programach komputerowych. Daje to bardzo dużą ilość możliwości do analizowania, przetwarzania bądź obróbki elementów. Skanery 3D są dzisiaj stosowane w przeróżnych gałęziach przemysłu takich jak motoryzacja, budownictwo oraz przemysł. Głównym założeniem projektu było stworzenie prostego skanera 3D potrafiącego zebrać odpowiednie dane, przekształcić je na współrzędne kartezjańskie i zapisać na karcie pamięci.

## 2 Opis kodu

**Silniki krokowe** - program w odpowiedni sposób ustawia stan wysoki na dany pinach. Istnieje możliwość wyboru 2 trybów pracy - pełno krokowy i pół krokowy. Różnią się one sekwencją włączania i wyłączania odpowiednich pinów. W celu ustawienia odpowiednich przerw między sygnałami został wykorzystany timer 0. Przerwania są wywoływanie co 1,6ms co dla silnika krokowego w trybie pół krokowym daje maksymalną szybkość. Zmniejszenie tego czasu powoduje gubienie kroków.

**Wyświetlacz LCD** - w celu komunikacji wyświetlacza mikrokontrolera z wyświetlaczem wykorzystano tryb 4 bitowy. Polega on na przesyłaniu danych za pomocą 4 linii sygnałowych. Dodatkowo wykorzystywany jest RS - register select, wybór pomiędzy komendami a rejestrami. Enable - sygnał 0 lub 1 dla potwierdzenia danych.

**Dalmierz laserowy VL53l0X** komunikuje się z mikrokontrolerem za pomocą protokołu  $I^2C$ . Magistrala  $I^2C$  opiera się na transmisji szeregowej prowadzoną za pomocą dwóch linii sygnałowych SDA (linia przesyłu danych) oraz SCL (sygnał zegarowy). W celu ustawienia linii SCL został wykorzystany Timer 1. Istnieje możliwość wybrania jednego z 3 trybów pracy: pojedynczy pomiar - wykorzystywany w programie, automatyczny pomiar - dalmierz kilkukrotnie mierzy odległość, pomiar w danym czasie - dalmierz ciągle wykonuje aż upłynie odpowiednia ilość czasu. Producent przewidział także wybór różnych trybów skanowania takich jak: default mode, high speed, high accuracy, long range jednak nigdzie w nocy katalogowej nie ma podanych rejestrów przez co program w głównej mierze został oparty na gotowym programie udostępnionym przez producenta.

**Karta SD** - Komunikacja między mikrokontrolerem została wykonana za pomocą protokołu SPI. Wykorzystano trzy linie sygnałowe: MISO - przesył danych z mikrokontrolera do karty SD, MOSI - przesył danych z karty SD do mikrokontrolera oraz SCLK - sygnał zegarowy. Do ustawienia odpowiedniej częstotliwości taktowania zegara wykorzystano Timer 2. Wykorzystane zostały funkcje takie jak `fmount()` - sprawdza czy poprawnie wykryto kartę SD, `fopen` - umożliwia otwieranie, zapisywanie bądź tworzenie nowych plików, `fwrite` - zapisuje w otwartym pliku odpowiednie dane, `fsync` - usuwa pamięć podręczną pliku co przyczynia się do zabezpieczenia pliku przed uszkodzeniem, `fclose` - zamyka otwarty plik.

## 2.1 Kod programu

```
1 void kalibruj(){
2
3     uint16_t pomiar = 0; // Zmienne pomocnicze
4     statInfo_t xTraStats;
5     char text[20];
6     uint8_t flaga = 1;
7
8     LCD_Clear();
9     sprintf(text, " Kalibracja");
10    LCD_GoTo(0, 0); // Wyswietla komunikat
11    LCD_WriteText(text);
12    sprintf(text, " skanera"); // Wyswietla komunikat
13    LCD_GoTo(0, 1);
14    LCD_WriteText(text);
15
16
17    while(flagaj{
18        half_step_motor_left(256,2); // uruchamia silnik krokowy
19        pomiar = readRangeSingleMillimeters( &xTraStats ); // Dokonuje pomiaru odleglosci
20
21        if(pomiar < 200){ // Jezeli pomiar jest mniejszy od 20cm
22            flaga = 0;
23            LCD_Clear();
24            sprintf(text, " Skalibrowano"); // Wyswietla komunikat
25            LCD_GoTo(0, 0);
26            LCD_WriteText(text);
27            sprintf(text, " skaner"); // Wyswietla komunikat
28            LCD_GoTo(0, 1);
29            LCD_WriteText(text);
30        }
31    }
32
33    half_step_motor_right(512,2); // Uruchamia 2 silnik krokowy
34
35}
36
37}
38
39
40
41 int main(){
42
43     statInfo_t xTraStats;
44     init(); // Inicjalizacja timera dla czujnika
45     oleg_o_ci
46     initVL53L0X(1); // Inicjalizacja czujnika odlegosci
47
48     ioinit(); // Inicjalizacja po czenia z kart
49     disk_initialize(0); // inicjalizacja karty SD
50
51     setMeasurementTimingBudget( 500 * 1000UL ); // Ustawienie prescalera na 500ms
52
53     LCD_Initialize(); // Inicjalizacja wywietlacza
54     LCD_Home();
55
56     cbi(DDRA,7); // Ustawienie PD0 jako PULL-UP
57     sbi(PORTA,7);
58
59     char text[20]; // Tablica pomocnicza
```

```

61     uint16_t pomiar = 0;                                // Zmienne pomocnicze
62     uint16_t licznik = 0;
63     uint16_t Ppros = 0;
64
65     uint16_t x = 0;                                     // Dlugosc przeciwpromienowej
66     uint16_t y = 0;
67     uint16_t z = 0;
68
69     uint16_t r = 130;                                    // Wspolrzedne punktu
70     float kat = 0;
71
72     FRESULT fr;                                       // Odleglosc czujnika od srodka
73     char name[12];
74     char buf[50];
75     int b = 0;
76
77     sprintf(text, " Skaner 3D");                      // Wy wietlanie tekstu
78     LCD_Clear();
79     LCD_GoTo(0, 0);
80     LCD_WriteText(text);
81
82     _delay_ms(5000);
83
84     fr = f_mount(&FatFs, "", 1);                      // Po czenie si z kart SD
85     _delay_ms(50);
86
87     while(fr != 0){                                     // Wy wietlenie komunikatu o bladzie
88         LCD_Clear();
89         LCD_GoTo(0,0);
90         LCD_WriteText("f_mount_FAILL");
91         sprintf(text, "Blad nr - %d",fr);
92         LCD_GoTo(0,1);
93         LCD_WriteText(text);
94         _delay_ms(4000);
95         disk_initialize(0);                            // ponowna inicjalizacja dysku
96         fr = f_mount(&FatFs, "", 1);
97     }
98     do{
99         sprintf(name, "skaner%d.txt",b);              // Tworzenie nowego pliku o danej nazwie
100        fr = f_open(&Fil, name, FA_WRITE | FA_CREATE_NEW);
101        f_sync(&Fil);                                // Tworzenie pliku tekstowego
102        b++;
103        if(b>9){                                     // Mo liwe jest stworzenie 10 plikow
104            LCD_Clear();
105            LCD_GoTo(0,0);
106            LCD_WriteText("f_open_ERR");
107            LCD_GoTo(0,1);
108            LCD_WriteText("New_file");
109            return 0;
110        }
111    }while(fr!=0);
112
113    kalibruj();                                       // Kalibracja skanera
114
115    while(!bit_is_clear(PINA,7)){}                   // Czekanie na reakcje uzytkownika
116
117    sprintf(text, " Rozpoczeto" );                  // Wy wietlenie komunikatu
118    LCD_Clear();
119    LCD_GoTo(0,0);
120    LCD_WriteText(text);
121    LCD_GoTo(0,1);
122    LCD_WriteText(" skanowanie");                  // Wy wietlenie komunikatu

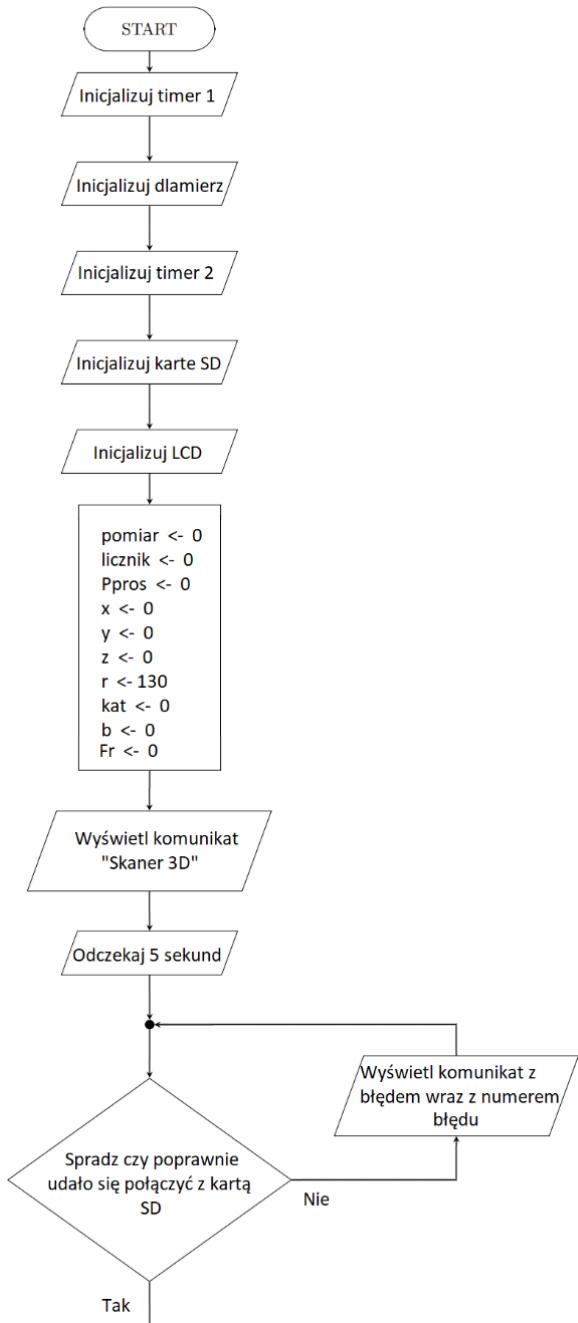
```

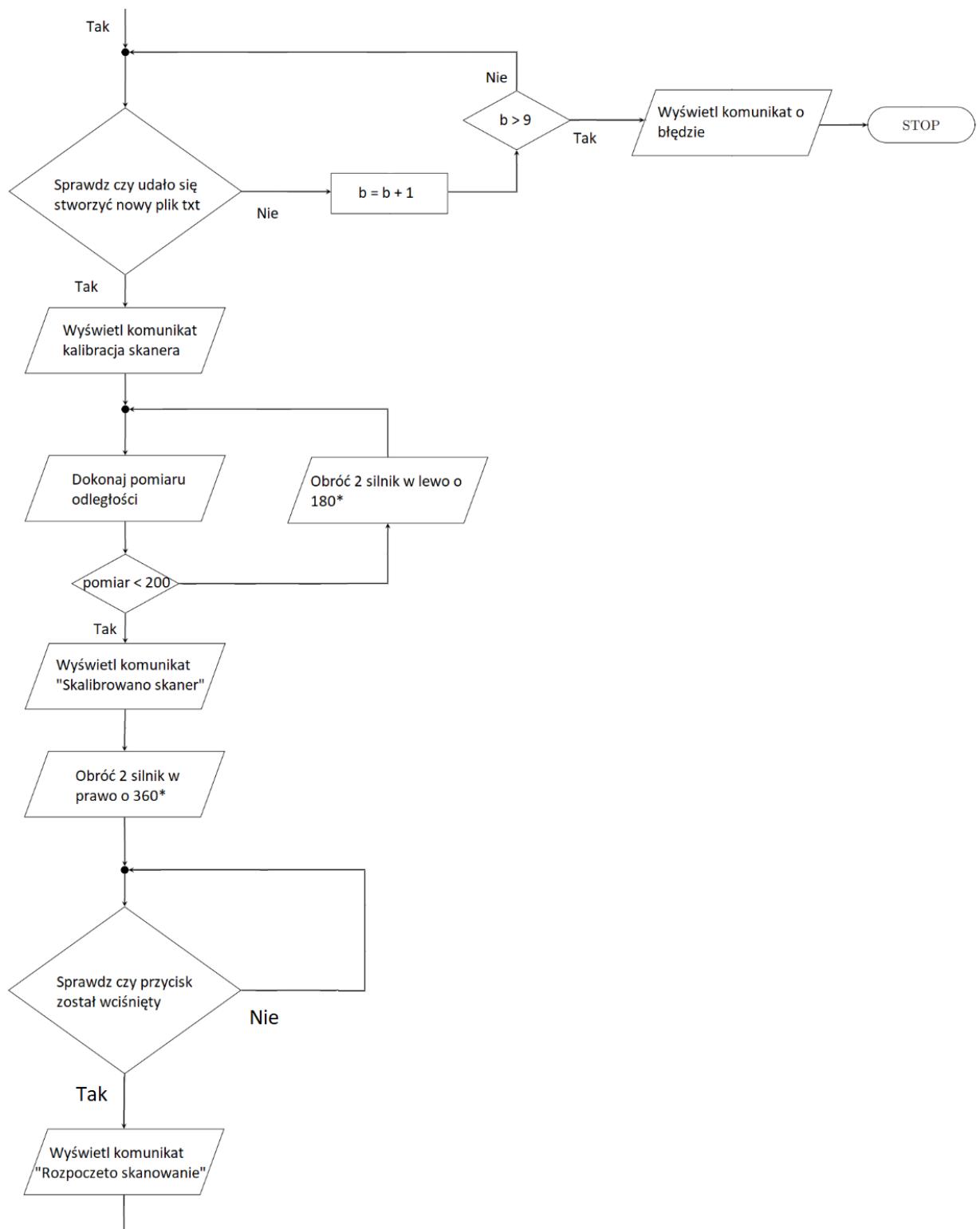
```

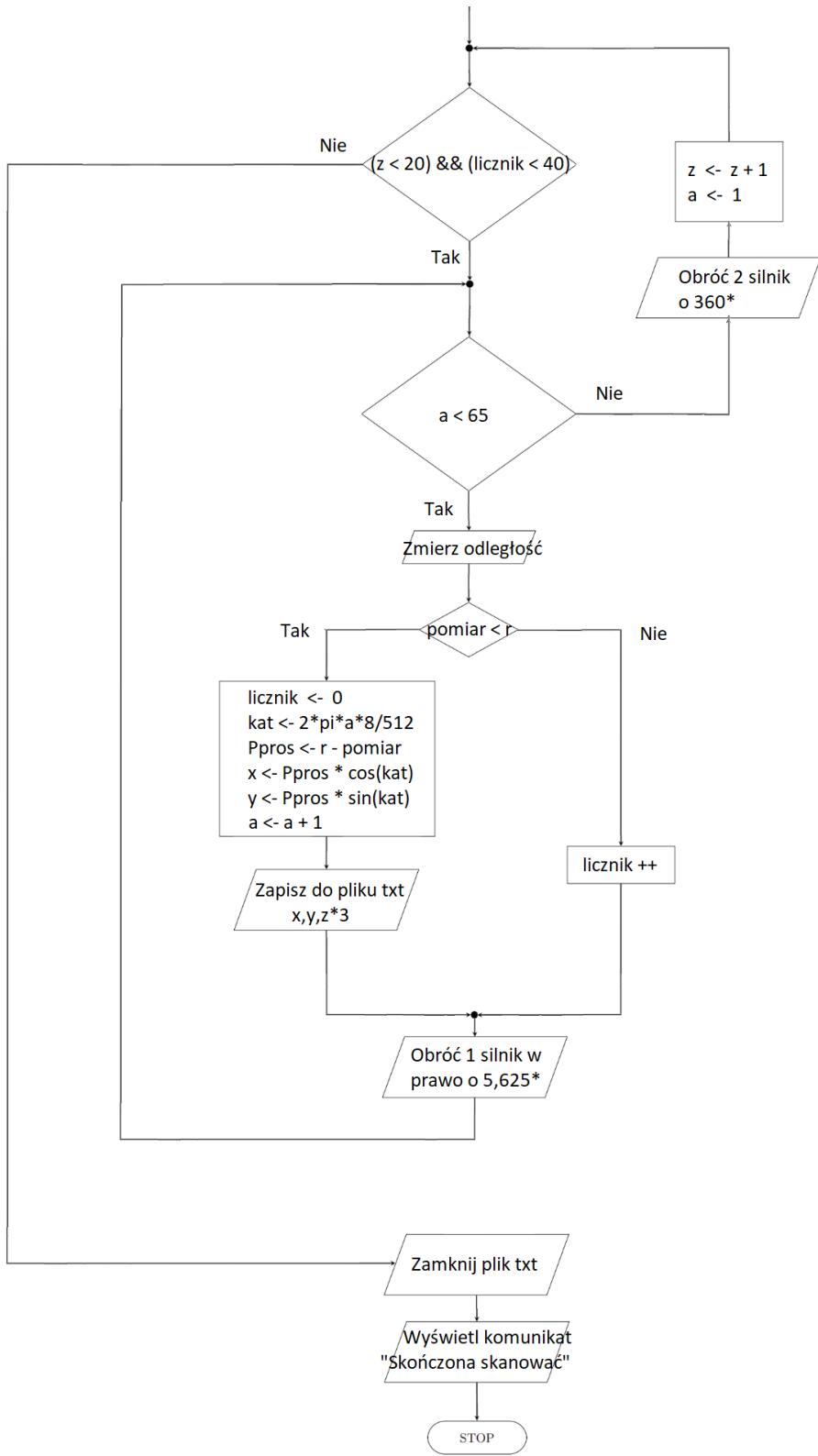
123     _delay_ms(4000);
124
125     while((z<20)&&(licznik<40)){ // P tla powtarzaj ca si 20 razu
126         for(int a=1; a<65 ;a++){
127
128             pomiar = readRangeSingleMillimeters( &xTraStats ) - 10; // Dokonanie pomiaru
129             odlegoci;
130
131             if(pomiar<r){ // Odrzucanie b dnych pomiar w
132                 licznik = 0; // Oblicza kat w radianach
133                 kat = (2*M_PI*a*8)/512; // Oblicza odlegosc od srodka
134                 Ppros = r - pomiar; // silnika krokowego
135                 x = Ppros*cosf(kat); // Wspolrzedna X
136                 y = Ppros*sinf(kat); // Wspolrzedna Y
137                 sprintf(buf, "%d,%d,%d\n",x,y,3*z ); // Zapisanie wspolrzednych do bufora
138                 f_write(&Fil , buf, sizeof(char)*strlen(buf) , &bw); // Zapisanie bufora na
139                 karte SD
140
141                 f_sync(&Fil); // Funkcja zabezpieczajaca otwerty
142                 plik txt w razie naglego unieruchomienia // Zapisane dane nie zostaja stracone
143             } else{ // Zwiekszenie licznika
144                 licznik++;
145                 half_step_motor_right(8,1); // Obrot silnikiem krokiem
146             }
147             z++; // Zwiekszenie zakresu
148
149             sprintf(text, " Z %d z 20 ",z ); // Wyswietla komunikat
150             LCD_Clear(); // LCD_Clear();
151             LCD_GoTo(0,1); // LCD_GoTo(0,1);
152             LCD_WriteText(text); // LCD_WriteText(text);
153
154             half_step_motor_right(1024,2); // uruchamia 2 silnik krokiem
155         }
156
157         f_close(&Fil); // Zamyka plik txt
158         LCD_Clear(); // LCD_Clear();
159         LCD_GoTo(0,0); // LCD_GoTo(0,0);
160         LCD_WriteText(" Skonczono"); // Wyswietla komunikat
161         LCD_GoTo(0,1); // LCD_GoTo(0,1);
162         LCD_WriteText(" skanowanie!"); // Wyswietla komunikat
163
164         return 0;
165     }

```

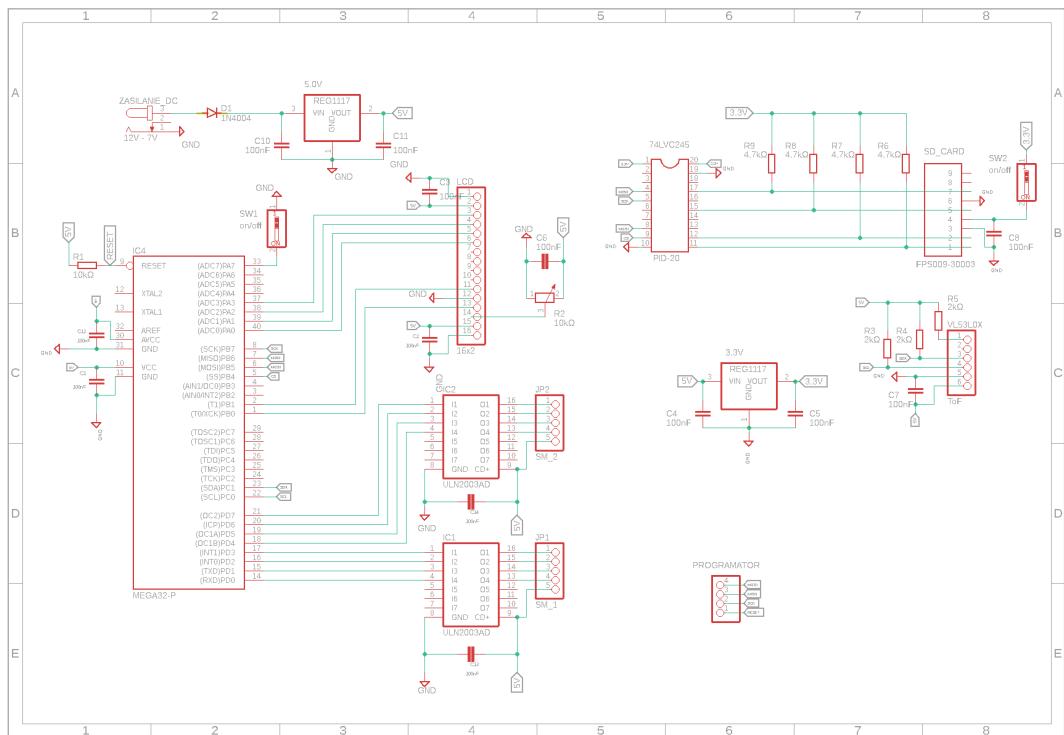
### 3 Schemat blokowy algorytmu sterowania







## 4 Połączenia z procesorem



Schemat 1: Przedstawia schemat elektryczny skanera 3D.

## 5 Tabela sygnałów

| Wyświetlacz LCD 16x2 |        |                        |                   |
|----------------------|--------|------------------------|-------------------|
| Numer pinu           | Sygnal | Opis sygnału           | Pin na procesorze |
| 1.                   | gnd    | Masa podświetlacza     | -                 |
| 2.                   | + 5V   | Zasilanie wyświetlacza | -                 |
| 3.                   | D7     | Komunikacja            | PA3               |
| 4.                   | D6     | Komunikacja            | PA2               |
| 5.                   | D5     | Komunikacja            | PA1               |
| 6.                   | D4     | Komunikacja            | PA0               |
| 7.                   | -      | -                      | -                 |
| 8.                   | -      | -                      | -                 |
| 9.                   | -      | -                      | -                 |
| 10.                  | -      | -                      | -                 |
| 11.                  | E      | Gotowość do odczytu    | PB1               |
| 12.                  | gnd    | Odczyt                 | -                 |
| 13.                  | RS     | Wybór rejestru         | PB0               |
| 14.                  | V0     | Kontrast               | -                 |
| 15.                  | +5V    | Zasilanie logiki       | -                 |
| 16.                  | gnd    | Masa logiki            | -                 |

Tabela 1: Przedstawia tabelę sygnałów dla wyświetlacza LCD.

| Slot na kartę SD |        |                                 |                   |
|------------------|--------|---------------------------------|-------------------|
| Numer pinu       | Sygnal | Opis sygnału                    | Pin na procesorze |
| 1.               | CS     | Sygnal wyboru struktury pamięci | PB4               |
| 2.               | MOSI   | Dane dla układu peryferyjnego   | PB5               |
| 3.               | GND    | Masa 1 układu                   | -                 |
| 4.               | 3.3V   | Zasilanie 3.3V                  | -                 |
| 5.               | SCK    | Sygnal zegarowy                 | PB7               |
| 6.               | GND    | Masa 2 układu                   | -                 |
| 7.               | MISO   | Dane z układu peryferyjnego     | PB6               |
| 8.               | -      | -                               | -                 |
| 9.               | -      | -                               | -                 |

Tabela 2: Przedstawia tabelę sygnałów dla modułu karty SD.

| Złącze JP1 – silnik krokowy 1 |        |                   |                   |
|-------------------------------|--------|-------------------|-------------------|
| Numer pinu                    | Sygnal | Opis sygnału      | Pin na procesorze |
| 1.                            | O1     | Sterowalna masa   | PD0               |
| 2.                            | O2     | Sterowalna masa   | PD1               |
| 3.                            | O3     | Sterowalna masa   | PD2               |
| 4.                            | O4     | Sterowalna masa   | PD3               |
| 5.                            | 5V     | Zasilanie silnika | -                 |

Tabela 3: Przedstawia tabelę sygnałów dla pierwszego silnika krokowego.

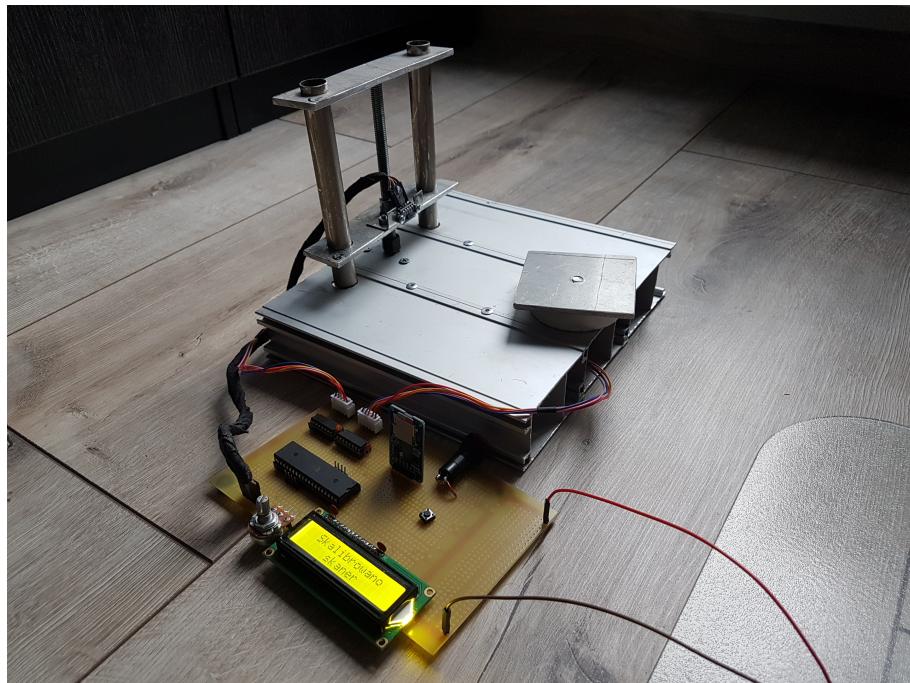
| Złącze JP2 – silnik krokowy 2 |        |                   |                   |
|-------------------------------|--------|-------------------|-------------------|
| Numer pinu                    | Sygnal | Opis sygnału      | Pin na procesorze |
| 1.                            | O1     | Sterowalna masa   | PC4               |
| 2.                            | O2     | Sterowalna masa   | PC5               |
| 3.                            | O3     | Sterowalna masa   | PC6               |
| 4.                            | O4     | Sterowalna masa   | PC7               |
| 5.                            | 5V     | Zasilanie silnika | -                 |

Tabela 4: Przedstawia tabelę sygnałów dla drugiego silnika krokowego.

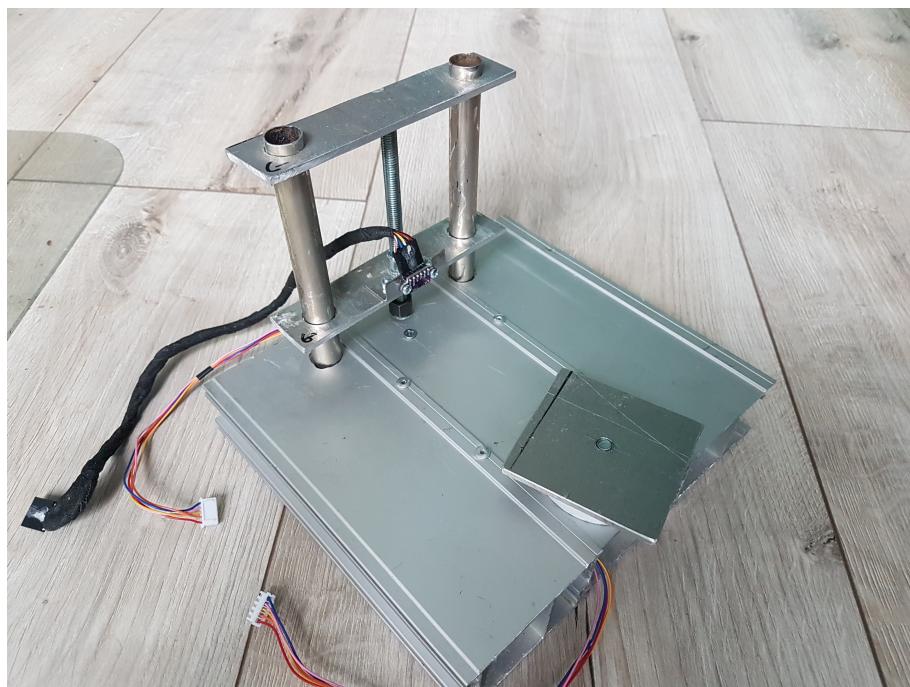
| Programator |        |                               |
|-------------|--------|-------------------------------|
| Numer pinu  | Sygnal | Opis sygnału                  |
| 1.          | MOSI   | Dane dla układu peryferyjnego |
| 2.          | MISO   | Dane z układu peryferyjnego   |
| 3.          | SCK    | Sygnal zegarowy               |
| 4.          | RESET  | Reset urządzenia              |

Tabela 5: Przedstawia tabelę sygnałów dla programatora.

## 6 Zdjęcia projektu



Zdjęcie 1: Przedstawia płytę wraz z stelażem.



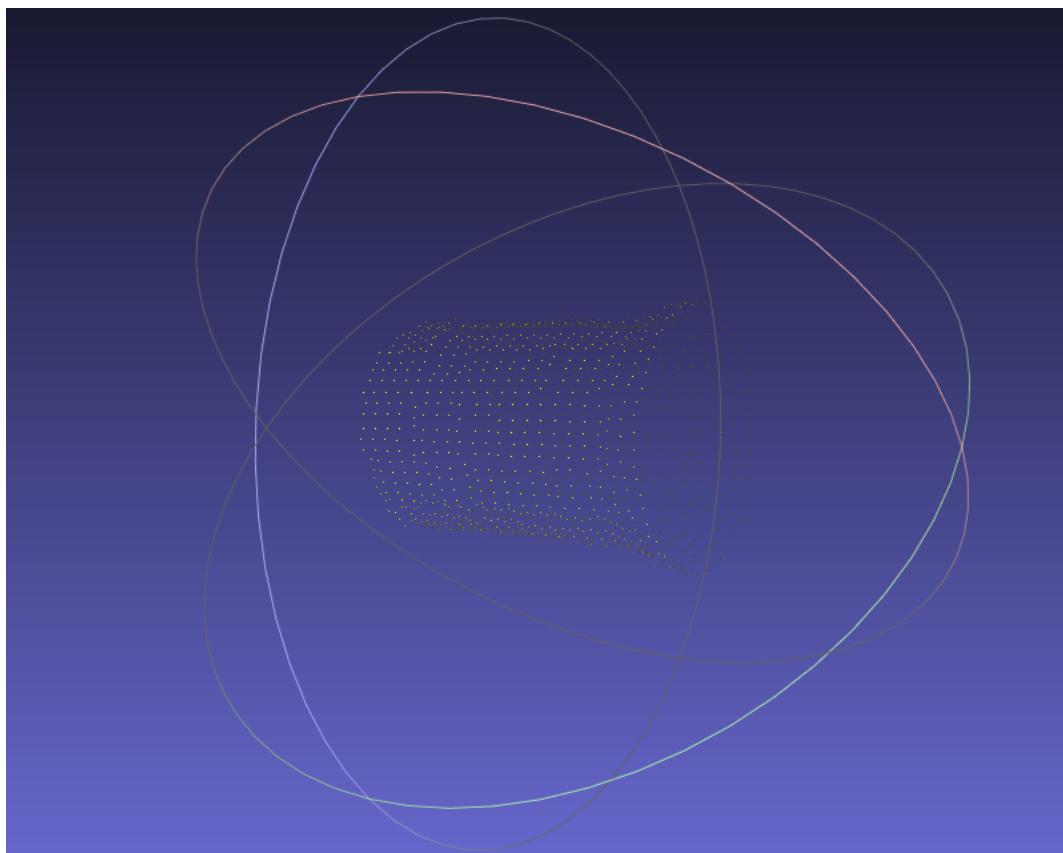
Zdjęcie 2: Przedstawia sam stelaż z dalmierzem.



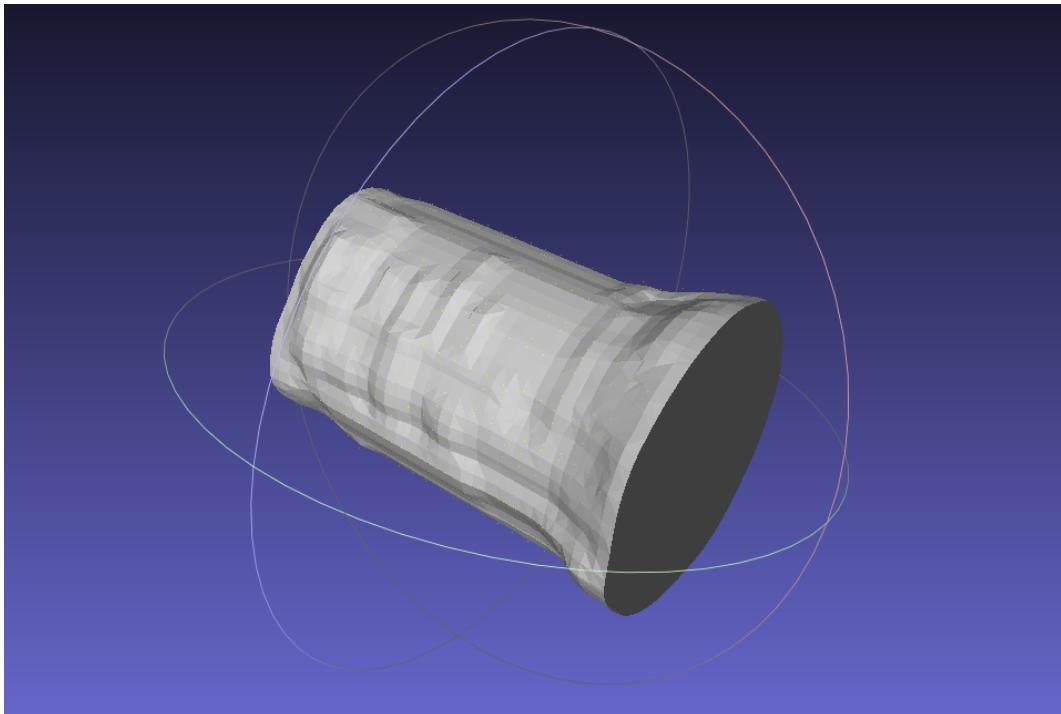
Zdjęcie 3: Przedstawia skanowany obiekt.

## 7 Finalny efekt

Poniżej znajduje się zdjęcie gotowego skanu drewnianego walca. Jak można zauważyc finalny efekt jest dosyć zadowalający. Różnego rodzaju odstępstwa, zniekształcenia są spowodowane dokładnością czujnika odległości oraz luzami, które występują na stelażu.



Zdjęcie 7: Przedstawia siatkę z zaznaczonymi punktami.



Zdjęcie 8: Przedstawia siatkę po odpowiedniej filtracji.

## 8 Podsumowanie projektu

Wszystkie założenia projektu zostały spełnione. Skonstruowanie i zaprogramowanie całego układu zajęło znacznie więcej czasu niż początkowo się spodziewałem. Projekt może jeszcze zostać rozwinięty poprzez:

1. Zmianę skanera VL53L0X na skaner laserowy o większej dokładności.
2. Wykorzystanie lepszej jakości silniki krokowe.
3. Zniwelowanie luzów występujących na stelażu
4. Stworzenie bardziej interaktywnego menu, dzięki czemu użytkownik będzie w stanie wybierać odpowiednie tryby skanowania.
5. Dołożenie klawiatury 3x4 (Jak w starszych telefonach) umożliwiającej wpisywanie wartości oraz nazw plików.