

Dokumentacja techniczna Niduc 2			
Numer grupy	7	Termin zajęć	Czw. parz. 11:15
Skład grupy:		Prowadzący	Ocena
Aleksander Dziągwa	281055	Prof. hab. Stanisław Piestrak	
Filip Kwiek	280947		
Kacper Mikosiński	280985		

Spis treści

1	Wprowadzenie	2
2	Instalacja	2
2.1	Wymagania wstępne	2
2.2	Instrukcje instalacji	2
3	Użycie	2
3.1	Podstawowe użycie	2
3.2	Przykłady kodu	2
4	Architektura	3
4.1	Struktura projektu	3

1 Wprowadzenie

Nazwa projektu: System tolerancji błędów

Opis: Projekt oparty na platformie Arduino, który wykorzystuje czujniki soniczne do pomiaru odległości. Wyniki są przetwarzane za pomocą różnych algorytmów, takich jak głosowanie większościowe, medianowe, Largest Difference Rejection oraz iteracyjne.

Cel: Celem projektu jest dokładne mierzenie odległości przy użyciu trzech czujników i przetwarzanie wyników za pomocą różnych algorytmów w celu uzyskania najbardziej wiarygodnego wyniku.

2 Instalacja

2.1 Wymagania wstępne

- Arduino IDE lub PlatformIO
- Płytką Arduino
- Trzy czujniki ultradźwiękowe
- Kable połączeniowe

2.2 Instrukcje instalacji

1. Sklonuj repozytorium projektu na swój komputer.
2. Otwórz projekt w Arduino IDE lub PlatformIO.
3. Podłącz płytkę Arduino do komputera.
4. Wgraj kod na płytkę Arduino.

3 Użycie

3.1 Podstawowe użycie

1. Po wgraniu kodu, uruchom płytkę Arduino.
2. Wyniki pomiarów będą wyświetlane w konsoli seryjnej co 1,5 sekundy.

3.2 Przykłady kodu

Kod główny znajduje się w pliku `src/main.cpp`:

```

src > main.cpp > loop()
1  #include <Arduino.h>
2  #include "lib.h"
3  #include "majority.h"
4  #include "Median.h"
5  #include "Iterative.h"
6
7
8  double *data;
9  void setup() {
10
11     initSensors(); //inicjalizacja czujników
12     Serial.begin(9600); //inicjalizacja portu seryjnego
13
14 }
15
16 void loop() {
17     data = allDistances(); //szczytanie pomiaru z czujników
18
19     Serial.print("\nWyniki pomiarow:\n1. "); //wypisanie wyników pomiarów do konsoli
20     Serial.print(data[0]);
21     Serial.print("cm\n2. ");
22     Serial.print(data[1]);
23     Serial.print("cm\n3. ");
24     Serial.print(data[2]);
25     Serial.print("cm\n");
26
27     Serial.print("Algorytm głosowania większościowego: "); //wypisanie wyniku algorytmu głosowania większościowego
28     Serial.println(majority_vote(data[0], data[1], data[2]));
29
30     Serial.print("Algorytm medianowy: "); //wypisanie wyniku algorytmu medianowego
31     Serial.println(find_median_sorted(data[0], data[1], data[2]));
32
33     Serial.print("Algorytm odrzucenia największej różnicy: "); //wypisanie wyniku algorytmu odrzucenia największej różnicy
34     Serial.println(large_difference_rejection(data[0], data[1], data[2]));
35
36     Serial.print("Algorytm iteracyjny: "); //wypisanie wyniku algorytmu iteracyjnego
37     Serial.println(iterative_vote(data[0], data[1], data[2]));
38
39     free(data); //zwolnienie pamięci
40
41     delay(1500);
42 }
43

```

Obraz 1: Przykładowa funkcja main

4 Architektura

4.1 Struktura projektu

- `src/`: Zawiera główny plik `main.cpp`, który inicjalizuje czujniki i przetwarza dane.
- `include/`: Zawiera pliki nagłówkowe z definicjami funkcji i algorytmów.
- `lib/`: Przeznaczone na biblioteki specyficzne dla projektu.