



POLITECHNIKA ŚLĄSKA
WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI

Praca dyplomowa magisterska

ZAPROJEKTOWAĆ SYSTEM AUTONOMICZNEGO WYZNACZANIA TOPOLOGII POMIESZCZEŃ

Autor:
WOJCIECH GAJDA

Promotor:
DR INŻ. TOMASZ PRZYBYŁA

Gliwice, 10 listopada 2009

Rodzinie, dzięki której jestem szczęśliwy;

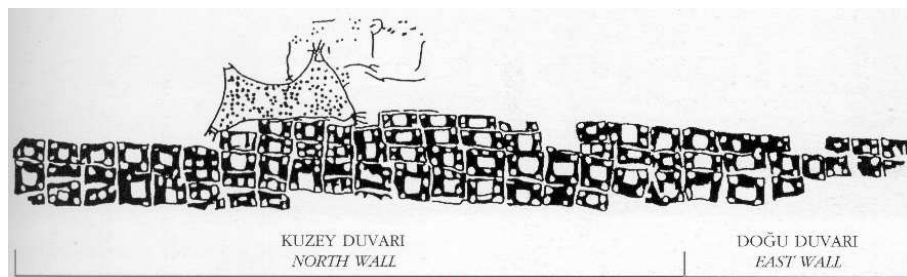
Spis treści

1	Wstęp	1
2	Cel pracy	7
3	Konstrukcja systemu	9
3.1	Wybór głównego układu sterującego	9
3.2	Przegląd metod pomiaru odległości	19
3.3	Wybór metody pomiaru odległości	27
3.4	Komunikacja z dalmierzem laserowym	29
3.5	Konstrukcja mechaniczna	36
3.6	Konstrukcja elektroniczna	39
3.7	Pomiar położenia	42
4	Oprogramowanie	47
4.1	Oprogramowanie urządzenia	47
4.2	Oprogramowanie PC	62
5	Przykładowe wyniki pomiarów	67
5.1	Pomiary przestrzeni referencyjnych	67
5.2	Pomiary pomieszczeń	71
6	Podsumowanie i wnioski końcowe	77
	Bibliografia	79

Rozdział 1

Wstęp

Od zarania dziejów ludzi cechowała chęć poznania otaczającego świata. Podczas odkrywania kolejnych zjawisk i miejsc otaczających człowieka zaistniała potrzeba utrwalenia zdobytych informacji. Już siedem tysięcy lat przed naszą erą człowiek sporządził pierwszą znaną nam mapę [1].



Rysunek 1.1: Rekonstrukcja najstarszej odnalezionej mapy

Dzięki swoim zmysłom człowiek potrafił ustalić gdzie jest i dokąd ma się udać porównując zawartość mapy z otaczającym go krajobrazem.

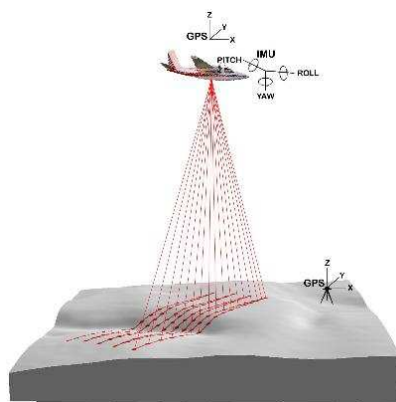
Szybki rozwój komputeryzacji sprawił, że mapy szybko zaczęły przyjmować formę elektroniczną. Dzięki temu łatwiej było je modyfikować i łatwiejsza była ich dystrybucja. Po wystrzeleniu pierwszych satelit, które stworzyły obraz całej kuli ziemskiej wydawało się, że nic więcej już nie będzie w tej dziedzinie potrzebne. Rozwój m.in. serwisów internetowych takich jak Google Earth czy NASA World Wind pozwolił każdemu na zwiedzanie świata bez opuszczania domu. Poszerzające

się możliwości zastosowań map elektronicznych, m.in w wojsku, przyczyniły się do powstania zaawansowanych urządzeń pomagających w zbieraniu informacji niezbędnych do tworzenia dokładniejszych map.

Kluczowym elementem, który zmienił oblicze tworzenia map jest system nawigacji satelitarnej (GPS) stworzony przez Departament Obrony USA. GPS pozwala na ustalenie swojej pozycji na kuli ziemskiej. Od momentu rozpoczęcia działania systemu do zlokalizowania pozycji na mapie nie było konieczności porównywania zawartości mapy z otaczającym krajobrazem - wystarczyła znajomość pozycji dostarczanej dzięki orbitującym wokół Ziemi satelitom. Lepsza znajomość pozycji umożliwia szybsze tworzenie dokładniejszych map. Proces tworzenia mapy może polegać na zapisywaniu rodzaju i pozycji GPS punktu szczególnego mapy.

Wynalezienie laserowych technik pomiaru odległości pozwoliło skonstruować urządzenia, które potrafią szybko i dokładnie mierzyć odległość do wskazywanego punktu. Konsekwencją było skonstruowanie tzw. lidaru - urządzenia mierzącego odległość w dwóch osiach obrotu. Lidar służy do określenia względnej pozycji wskazywanego punktu. Nowoczesne lidary pozwalają na pomiar pozycji milionów punktów w ciągu kilku sekund.

Lotniczy skanowanie laserowe (ang. ALS – Airborne Laser Scanning) jest metodą pomiaru pozycji, która łączy laserową technikę pomiaru relatywnej pozycji z systemem GPS ustalającym pozycję absolutną [2]. Tworzone w ten sposób chmury



Rysunek 1.2: Idea lotniczego skaningu laserowego

punktów (ang. point clouds) wykorzystywane są do generacji trójwymiarowych map i wizualizacji terenu.



Rysunek 1.3: Obraz Manhattanu stworzony przez Armię Stanów Zjednoczonych

Lotniczy skanowanie laserowe często służy do pozyskiwania informacji przestrzennych dotyczących lasów [3].

Popularyzacja robotyki sprawiła, że roboty stają się elementem codziennego życia. Roboty ratunkowe i te wykorzystywane przez policję, np. do rozbrajania bomb, najczęściej sterowane są bezprzewodowo przez operatora. W instytutach badawczych powstają roboty działające w pełni autonomicznie. Dzięki temu możliwe jest ich wykorzystanie tam gdzie ważny jest czas wykonania zadania. Poruszające się roboty nie mogą funkcjonować autonomicznie bez oceny możliwości ruchu. Każdy poruszający się system powinien znać otoczenie w którym się przemieszcza. System nawigacji satelitarnej oraz dokładne mapy geograficzne świata umożliwiają łatwe poruszanie się robotów w otwartych przestrzeniach. Systemy pracujące tam, gdzie nie docierają sygnały z satelitów lub dla których dokładność ich wskazań jest zbyt niska, potrafią dokonać oceny swojego położenia w inny sposób.

Pojazdy serii Pioneer są przykładem kompletnego systemu potrafiącego poruszać się i samodzielnie ocenić swoje położenie w pomieszczeniach zamkniętych. Produ-

kowane są przez firmę Mobile Robots i służą głównie naukowcom zajmującym się tworzeniem innowacyjnych systemów autonomicznych.

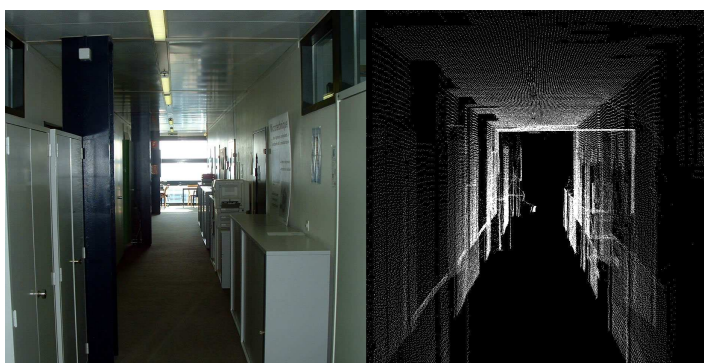


Rysunek 1.4: Pojazd Pioneer wyposażony w lidar firmy SICK połączony z komputerem osobistym

Podobne systemy, wykorzystywane m.in. do tworzenia map pomieszczeń, tworzone są na wielu technicznych uczelniach świata. Stworzony w Szwajcarskim Instytucie Technologicznym robot Pygmalion jest wykorzystywany do generowania trójwymiarowych map pomieszczeń [7]. Efektem jego pracy są chmury punktów dzięki którym można dokonać wizualizacji pomieszczenia. Dane te mogą także posłużyć innym systemom przemieszczającym się w już zbadanych pomieszczeniach.



Rysunek 1.5: Robot “Pygmalion”



Rysunek 1.6: Przykładowy wynik pomiaru pomieszczenia (po prawej) wykonany przez “Pygmalion”

Większość systemów tworzących mapy pomieszczeń dokonuje pomiarów za pomocą lidar. Jednym z najciekawszych przykładów użycia lidar w systemach autonomicznych jest wykorzystanie go w pojazdach startujących w zawodach Grand

Challenge organizowanych przez Departament Obrony Stanów Zjednoczonych (ang. DARPA). Zawody te mają na celu stworzenie pojazdu potrafiącego poruszać się po zatłoczonych drogach zgodnie z zasadami ruchu drogowego. Jednym z sześciu pojazdów, które ukończyły zawody w 2007 roku jest “Talos” stworzony przez drużynę z Massachusetts Institute of Technology [5]. Pojazd ten wyposażony jest w:

- 5 kamer
- 15 radarów
- 12 lidarów o ograniczonym polu widzenia
- 1 lidar dookólny



Rysunek 1.7: “Talos” z MIT

Zastosowane lidary pozwoliły pojazdowi skutecznie omijać przeszkody, a także wykrywać inne pojazdy znajdujące się w pobliżu [4]. W trakcie pomiarów pojazdy znajdowały się w ciągłym ruchu i dane z wielu sensorów musiały być przetwarzane w sposób ciągły. Zastosowanie tak dużej liczby sensorów nie wystarczyło jednak do zwycięstwa w zawodach. Do sukcesu zwycięskiej drużyny z Uniwersytetu w Stanford w głównej mierze przyczynił się autorski algorytm generowania mapy otoczenia na

podstawie danych z sensorów laserowych [6]. Pokazuje to jak ważnym elementem autonomicznych systemów jest proces tworzenia mapy otoczenia.

Rozdział 2

Cel pracy

Celem niniejszej pracy jest zaprojektowanie i konstrukcja systemu tworzącego elektroniczne mapy pomieszczeń.

System ten powinien działać samodzielnie - bez obsługi człowieka. W razie potrzeby użytkownik będzie mógł połączyć się bezprzewodowo z urządzeniem i zmienić parametry działania bądź dokonać pomiarów manualnie.

Głównym elementem wykonawczym będzie autonomiczne urządzenie pomiarowe - robot mogący dokonać pomiaru odległości do obiektów znajdujących się wokół niego. Dodatkowo system będzie w stanie określić własną pozycję w otaczającej go przestrzeni.

W ramach projektu zostanie utworzone oprogramowanie niezbędne do późniejszej wizualizacji wyników na ekranie komputera, a także do manualnego sterowania robotem.

Rozdział 3

Konstrukcja systemu

3.1 Wybór głównego układu sterującego

Na etapie projektowania, do sterowania urządzeniem został wybrany 32bitowy mikrokontroler z rdzeniem ARM. Mikrokontrolery te wydają się być standardem w urządzeniach przemysłowych i domowych gdzie wymagana jest duża moc obliczeniowa, niski pobór prądu, stabilność i szeroka gama peryferiów.

Największą siłą rdzenia ARM jest dobrze przemyślana struktura wewnętrzna i lista rozkazów. Wszystkie instrukcje wykonywane przez rdzeń posiadają regularną budowę. Rozkazy mają ortogonalne argumenty, tzn. dowolny rejestr może być zarówno rejestrem źródłowym jak i docelowym. Każdy rozkaz może być wykonywany warunkowo, dzięki temu zbędne stają się instrukcje skoków warunkowych, które to w procesorach RISC stanowią utrudnienie (po każdym skoku potok instrukcji musi zostać ponownie napełniony). Dostępnych jest 15 mnemoników instrukcji warunkowych. Rozkazy arytmetyczne, logiczne i przesłania mogą opcjonalnie uaktualniać stan flag rejestru statusu. W każdej operacji dodatkowo możliwe jest użycie sprzętowego przesuwnika bitowego umożliwiającego odpowiednią modyfikację danych przed wykonaniem operacji. [12]

Z pośród szerokiej gamy producentów mikrokontrolerów z rdzeniami ARM i wie-

lu modeli różniących się peryferiami, został wybrany układ AT91SAM7S256 firmy Atmel, posiadający 256kb pamięci flash oraz 64kb pamięci SRAM.

Historia firmy ARM

ARM (Advanced RISC Machines) jest firmą, która powstała w 1990 roku w celu promowania i rozwoju procesorów 6502 uprzednio produkowanych przez firmę Acorn Computers. Początkowo firma zajmowała się poza rozwojem także produkcją procesorów dzięki współpracy z firmą VLSI (aktualnie należąca do firmy Philips). Firmę ARM szybko jednak zaczęło wyróżniać to, że przestała sprzedawać same procesory, a zaczęła sprzedawać jedynie licencję na ich produkcję i umieszczanie we własnych rozwiązaniach. Od tamtego czasu pojęcie IPC (Intellectual Property Cores) zaczęło na dobre funkcjonować w branży elektronicznej. Takie rozwiązanie było dobre dla każdej ze stron. Duże koncerny produkujące części elektroniczne kupowały sprawdzony, znany inżynierom, szybki procesor i bez dużych wydatków przeznaczonych na projektowania i testowanie mogły wprowadzić go do swoich układów jako element sterujący. Z drugiej strony, dzięki temu, że jest on używany przez tak wielu producentów, powstało dużo kompilatorów, w tym i darmowe wspierających tę architekturę. Obecnie mikrokontrolery z rdzeniem ARM można spotkać niemal wszędzie. Największą branżą zdominowaną przez mikrokontrolery z rdzeniem ARM są telefony komórkowe, smartphony, palmtopy.

Przegląd kompilatorów i środowisk programistycznych dla mikrokontrolerów z rdzeniem ARM

Duża popularność mikrokontrolerów z rdzeniem ARM sprawiła, że na rynku powstało sporo kompilatorów, a także kompletnych zintegrowanych środowisk (IDE) przeznaczonych do tworzenia oprogramowania dla tych kontrolerów.

Keil

Wśród największych producentów tego typu oprogramowania czołowym jest Keil (marka należąca do firmy ARM). W ofercie Keil można znaleźć:

- Środowisko μ Vision będące zaawansowanym edytorem kodu źródłowego, a także środowiskiem do debugowania i symulacji procesora.
- Kompilator C/C++, Assembler oraz konsolidator (linker).
- Biblioteka MicroLib będąca wysoce zoptymalizowaną pod kątem rozmiaru, napisaną w języku C, wykorzystywaną w systemach wbudowanych.
- System RTX będący systemem czasu rzeczywistego przeznaczonym do zastosowań w systemach wbudowanych.
- Szereg sprzętowych programatorów i debuggerów łączących edytor kodu źródłowego z docelowym, programowanym urządzeniem.

Firma Keil w swojej ofercie posiada więc wszystko co potrzebne jest do szybkiego rozpoczęcia pracy z mikrokontrolerami z rdzeniem ARM. Środowisko to jest płatne, a wysoka cena świadczy o tym, że docelowo głównymi odbiorcami Keil powinny być duże koncerny profesjonalnie zajmujące się pisaniem oprogramowania do systemów wbudowanych.

Rowley Associates

Firma Rowley Associates oferuje środowisko CrossWorks for ARM będące połączeniem rozbudowanego edytora, wraz ze środowiskiem do symulacji i debugowania. CrossWorks for ARM wydaje się być dobrym kompromisem pomiędzy drogim Keil a darmowymi środowiskami, które wymagają czasochłonnej konfiguracji.

IAR

Firma IAR Systems znana z tworzenia środowisk programistycznych dla różnych procesorów w swojej ofercie posiada pakiet: IAR Embedded Workbench for ARM, składający się z edytora oraz kompilatora. Najnowsza wersja oprogramowania współpracuje ze wszystkimi aktualnie dostępnymi na rynku wersjami rdzeni ARM. Istnieje także możliwość połączenia kompilatora IAR z edytorem Eclipse.

CodeSourcery

Firma CodeSourcery przy współpracy z firmą ARM zajmuje się rozwojem darmowego kompilatora, konsolidatora i symulatora dla procesorów z rdzeniem ARM. Dodatkowo firma ta oferuje płatną wersję, w skład której wchodzi dodatkowo odpowiednio skonfigurowane środowisko Eclipse wraz ze specjalną wtyczką umożliwiającą szybko tworzyć projekty dla systemów wbudowanych. Płatna wersja posiada także bibliotekę CS3 unifikującą inicjalizację i obsługę przerwań różnych rdzeni mikroprocesorów (m.in ARM, MIPS, ColdFire). Firma CodeSourcery przez cały czas rozwija swój projekt i regularnie wypuszcza nowe wersje darmowych kompilatorów.

WinARM

WinARM jest to pakiet narzędzi dla mikrokontrolerów z rdzeniem ARM mający być odpowiednikiem WinAVR - powszechnie wykorzystywanego otwartego pakietu narzędzi i bibliotek stosowanych do tworzenia oprogramowania dla mikrokontrolerów

AVR pod systemem Windows. W skład otwartego pakietu WinARM wchodzi:

- Kompilator C/C++, Assembler oraz konsolidator (linker).
- Biblioteka newlib pozwalająca na wykorzystywanie niektórych funkcji systemowych UNIX.
- Program Programmers Notepad - będący prostym edytorem kodu źródłowego.
- OpenOCD (Open On-Chip debugger) - darmowe narzędzie służące programowaniu i debugowaniu mikrokontrolerów z interfejsem JTAG.
- H-JTAG - program czytający i zapisujący zawartość pamięci flash do mikrokontrolera.

Pakiet WinARM pozwala rozpocząć w pełni pracę z mikrokontrolerami z rdzeniem ARM. Wszystkie składniki pakietu są darmowe, a większość z nich udostępniania jest na zasadach OpenSource. Pełna konfiguracja edytora, kompilatora i debuggera może być czasochłonna. Pakiet ten w chwili pisania tej pracy od kilku lat nie jest już rozwijany.

YAGARTO

YAGARTO (Yet another GNU ARM toolchain) - podobnie jak WinARM jest otwartym pakietem narzędzi służącym pracy z mikrokontrolerami z rdzeniem ARM dla systemu windows oraz Max OS X. W skład YAGARTO wchodzi trzy pakiety:

- Kompilator C/C++, Assembler, konsolidator oraz biblioteka newlib
- Debugger OpenOCD (Open On-Chip debugger)
- Biblioteka newlib - pozwalająca na wykorzystywanie niektórych funkcji systemowych UNIX.
- Edytor Eclipse z odpowiednimi wtyczkami

O ile kompilator C/C++, Asembler i konsolidator są wciąż rozwijane i ze strony producenta można pobrać gotowy instalator o tyle pozostałe dwa pakiety (debugger i edytor Eclipse) z powodu różnic licencyjnych należy pobrać ze strony odpowiednich producentów, zainstalować i skonfigurować samemu.

Konfiguracja środowiska Eclipse

Po ocenie możliwości środowisk do tworzenia oprogramowania mikrokontrolera autor niniejszej pracy postanowił zastosować darmowe środowisko współpracujące z edytorem Eclipse.

Eclipse - jest to platforma udostępniania na zasadach otwartego oprogramowania i rozwijana przez Fundację Eclipse. Została napisana w języku Java i dzięki temu jest łatwo przenoszona na różne platformy sprzętowe. Powszechnie dostępne setki dodatkowych wtyczek do Eclipse, rozszerzają jego funkcjonalność i umożliwiają edycję kodów źródłowych wszystkich znanych języków programowania. Dodatkowo edytor ten oferuje szerokie możliwości debugowania dzięki dobrej integracji z GDB (GNU Debugger). Eclipse ze względu na duże możliwości rozbudowy i konfiguracji jest często wykorzystywany w profesjonalnych zastosowaniach.

Do kompilacji źródeł programu w niniejszej pracy został użyty pakiet narzędziowy CodeSourcery dla mikrokontrolerów z rdzeniem ARM. Autor nie skorzystał z płatnej wersji, która zawiera odpowiednio skonfigurowanym edytorem Eclipse, a jedynie z darmowego (również do celów komercyjnych) kompilatora. Następnie została przeprowadzona konfiguracja edytora Eclipse, a także całego środowiska do pracy z układem AT91SAM7S256 firmy Atmel:

Pierwszym krokiem jest pobranie niezbędnych narzędzi:

- Darmowy pakiet CodeSourcery - Sourcery G++ ARM Lite Edition
- Eclipse - wersja CDT - dla developerów C/C++

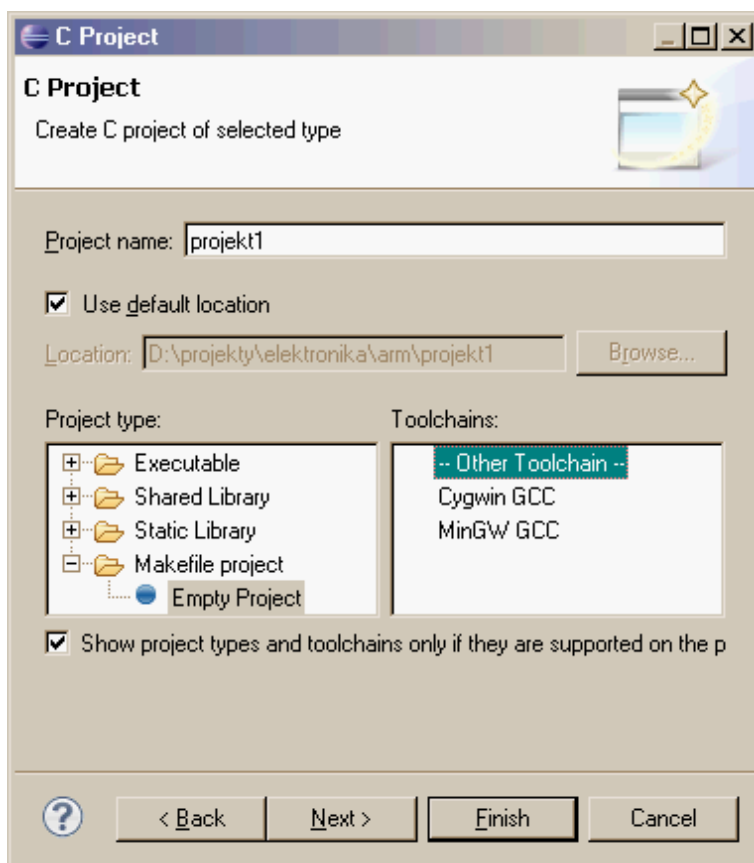
- Biblioteka o otwartym źródle dostarczana przez firmę Atmel dla swoich mikrokontrolerów z rdzeniem ARM - AT91LIB.
- Program H-JTAG służący do programowania pamięci flash mikrokontrolera

Praca w środowisku Eclipse opiera się na przestrzeniach pracy (workspace), w których znajdują się projekty. Przestrzenią pracy najczęściej będzie katalog, w którym trzymane są projekty powiązane z jednym rodzajem programowanego układu. Na przykład:

- Dla układów Atmel z rdzeniem ARM stworzymy specjalną przestrzeń pracy, dzięki czemu wszystkie projekty jakie w niej się znajdą będą mogły korzystać z wspólnych ustawień przestrzeni - jak np. dołączane pliki nagłówkowe, biblioteki lub używany kompilator.
- Inną przestrzeń pracy można mieć skonfigurowaną do pisania dokumentacji w systemie LaTeX, gdzie niewidoczne będą niepotrzebne opcje (np. debugger).

W celu rozpoczęcia pracy ze środowiskiem Eclipse, należy stworzyć na lokalnym dysku katalog, w którym znajdować się będzie przestrzeń pracy (workspace). W katalogu tym najlepiej jest umieścić bibliotekę AT91LIB w celu nie powielania jej w każdym osobnym projekcie. Po uruchomieniu edytora Eclipse i wybraniu odpowiedniej przestrzeni pracy (workspace) można utworzyć nowy projekt. Dla mikrokontrolerów z rdzeniem ARM i kompilatora CodeSourcery może to być *C Project* lub *C++ Project*. Domyślna instalacja Eclipse-CDT nie zawiera odpowiednich konfiguracji, aby móc pracować z kompilatorem CodeSourcery. Niemniej jednak zawsze istnieje możliwość zdefiniowania dowolnego pakietu narzędziowego (toolchain) dzięki obsłudze plików Makefile. Aby móc poprawnie pracować z kompilatorem CodeSourcery należy wybrać: – *Other Toolchain* – z typu: *Makefile Project* (Rysunek 3.1).

Do poprawnej obsługi plików Makefile dostarczanych z biblioteką AT91LIB należy zmienić w projekcie domyślną komendę wykonującą plik Makefile (Rysunek 3.2).

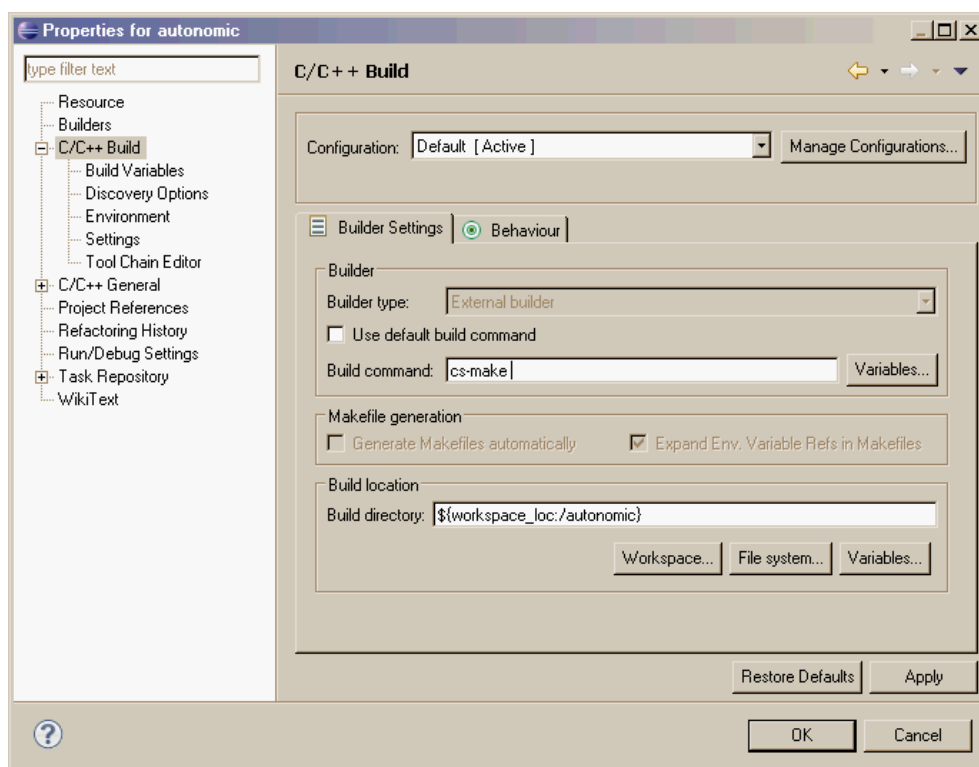


Rysunek 3.1: Wybór odpowiedniego pakietu narzędziowego w środowisku Eclipse

W celu zmiany komendy należy we właściwościach danego projektu w ustawieniach *C/C++ Build* odznaczyć opcję używania przez Eclipse domyślnej komendy do budowania (*Use default build command*), a używać tej, dostarczonej z pakietem CodeSourcery: *cs-make*.

Pozostała część konfiguracji powinna znaleźć się w pliku Makefile projektu. Tworzenie pliku Makefile może być bardzo czasochłonne. Zazwyczaj jednak raz zrobiony plik Makefile można wykorzystać w wielu projektach z minimalnymi poprawkami. Firma Atmel opracowała własny plik Makefile, udostępniany na zasadach otwartego oprogramowania wraz z biblioteką AT91LIB, który z powodzeniem może zostać skopiowany i wykorzystany w kolejnym projekcie. W tworzonym projekcie można użyć dowolnego Makefile znajdującego się w jednym z przykładowych projektów biblioteki AT91LIB.

Jedyną modyfikacją o jakiej należy pamiętać przy rozpoczynaniu nowego pro-



Rysunek 3.2: Zmiana komendy przetwarzającej plik Makefile we właściwościach projektu Eclipse

jektu jest zmiana kilku istotnych parametrów w pliku Makefile:

- dyrektywa *AT91LIB* powinna wskazywać na katalog w którym umieszczona jest biblioteka AT91LIB, najlepiej, aby wskazanie to było relatywne. np: `../at91lib`
- w dyrektywnie *OUTPUT* ustalamy nazwę pliku wynikowego - bez rozszerzenia. Najczęściej będzie to nazwa projektu.
- parametrem *CROSS_COMPILE* można łatwo zmienić kompilator stosowany w projekcie. Dla CodeSourcery parametr ten powinien być równy: *arm-none-eabi-*

Jeżeli planowane jest użycie dodatkowej biblioteki peryferii z AT91LIB należy dodać wybrany katalog z daną biblioteką do *VPATH* za pomocą dopisania na końcu definicji *VPATH* linijki *VPATH+=\$(PERIPH)/dbgu* (dla przykładu użycia

układu DBGU znajdującego się w układzie AT91SAM7S256). Dodatkowo należy dodać wskazaną bibliotekę do kompilacji dopisując dodatkowy obiekt w dyrektywie *C_OBJECTS* np. *C_OBJECTS += dbg_u.o.*

Utworzony projekt powinien zawierać plik *main.c* z główną pętlą programu. Projekt można skompilować za pomocą opcji Build w środowisku Eclipse (ikona). Wynik kompilacji znajduje się w katalogu projektu w folderze *bin*.

Plik wynikowy z rozszerzeniem *.bin* nadaje się do zaprogramowania pamięci flash docelowego układu. W celu załadowania pamięci flash układu z rdzeniem ARM najłatwiej jest użyć programu H-JTAG.

Należy pamiętać, że po zaprogramowaniu układ nie dokona resetu automatycznie, gdy opcja resetu nie została programowo włączona w trakcie trwania programu.

3.2 Przegląd metod pomiaru odległości

Pomiar odległości metodą taktylną

Pomiar taktylny (dotykowy) najczęściej informuje o tym czy dana odległość jest mniejsza od założonej. Istnieją jednak inne rozwiązania dzięki którym można dokonać dokładnego pomiaru odległości i siły nacisku. Metoda ta jest szeroko stosowana w czujnikach krańcowych, jednakże jest rzadko używana do pomiaru odległości z powodu wielu wad, z których główne to:

- przy badaniu sondą - długość sondy
- przy badaniu czujnikiem krańcowym konieczność przemieszczania się systemu

Idea pomiaru odległości metodą taktylną

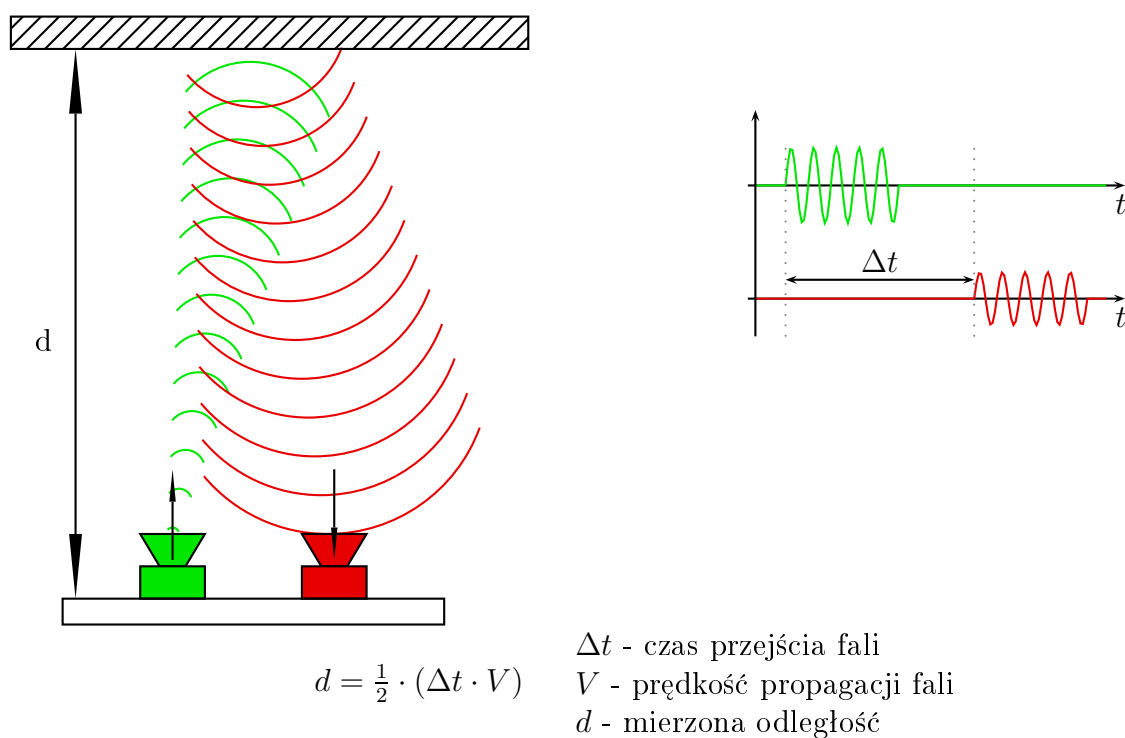
Najprostszym przykładem czujnika stykowego jest przełącznik elektromechaniczny. Gdy odległość do mierzonego obiektu będzie wystarczająco mała, przełącznik zostanie zwarty. Inną metodą jest użycie czujników tensometrycznych, dzięki którym znana jest siła nacisku. Pomiaru odległości do obiektu dokonuje się poprzez badanie nacisku wraz ze zbliżaniem czujnika do obiektu. Innym sposobem może być pomiar prądu płynącego przez uzwojenie silnika sterującego długością sondy badawczej. Kiedy sonda dotknie obiektu prąd płynący przez uzwojenie silnika wzrośnie. Odpowiednia kalibracja lub sprzężenie zwrotne pozwoli na dokładny pomiar odległości do obiektu [8].

Pomiar odległości przy użyciu fal ultradźwiękowych

Pomiar odległości falami ultradźwiękowymi jest jedną z prostszych i tańszych metod bezstykowych, która wykorzystywana jest głównie w robotyce i w samochodowych systemach parkowania.

Idea pomiaru odległości przy użyciu fal ultradźwiękowych

Pomiaru odległości za pomocą fal ultradźwiękowych dokonuje się przy użyciu nadajnika i odbiornika fali dźwiękowej. Odległość jest wprost proporcjonalna do zmierzonego czasu, który upłynął od momentu wysłania fali do jej powrotu. Znając prędkość rozchodzenia się fali w powietrzu można dokładnie obliczyć odległość jaką pokonała fala dźwiękowa.



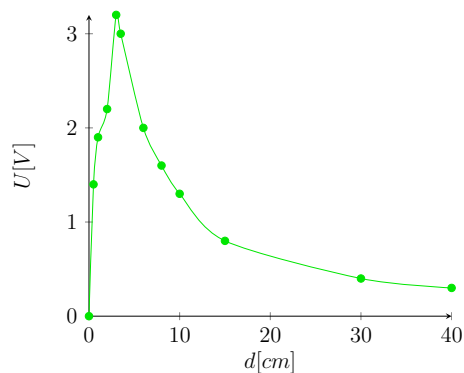
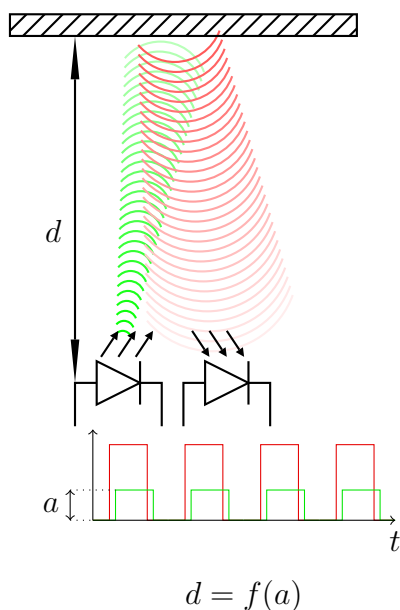
Rysunek 3.3: Idea pomiaru odległości falami ultradźwiękowymi

Pomiar odległości za pomocą badania natężenia odbitego światła

Jest to metoda optyczna, najczęściej nazywana metodą IR z uwagi na stosowane fotodetektory reagujące na światło podczerwieni. Na rynku dostępnych jest kilka gotowych rozwiązań. Wynik pomiaru jest w dużej mierze zależny od zewnętrznego oświetlenia, a także materiału na jaki pada wiązka światła. Wadą metody jest to, że pomiar nie może być wiarygodny w nieznanym otoczeniu. Poprawne pomiary są możliwe do osiągnięcia jedynie w środowisku odpowiednio do tego celu przygotowanym lub odpowiednio zbadanym. Pomiaru odległości można dokonać w bardzo wąskim zakresie. Dodatkowo, sygnał z czujnika jest nieliniowo zależny od odległości co dodatkowo obniża dokładność pomiaru. Metoda ta doskonale nadaje się do wykrywania jasności elementu. Stosowana jest głównie w robotyce w celu rozpoznawania poziomu jasności powierzchni po której porusza się robot, a także do wykrywania przeszkód znajdujących się tuż przed czujnikiem.

Idea pomiaru odległości za pomocą badania natężenia odbitego światła

W metodzie tej wykorzystywany jest efekt słabnięcia fali świetlnej wraz ze wzrostem odległości lub kąta padania. Światło emitowane po odbiciu od obiektu odbierane jest przez fotodetektor. Po kalibracji sygnał na wyjściu fotodetektora może posłużyć jako informacja o odległości do obiektu.



Rysunek 3.5: Przykładowa zależność odległości od napięcia wyjściowego z czujnika firmy SHARP [9]

Rysunek 3.4: Idea pomiaru odległości za pomocą badania natężenia odbitego światła

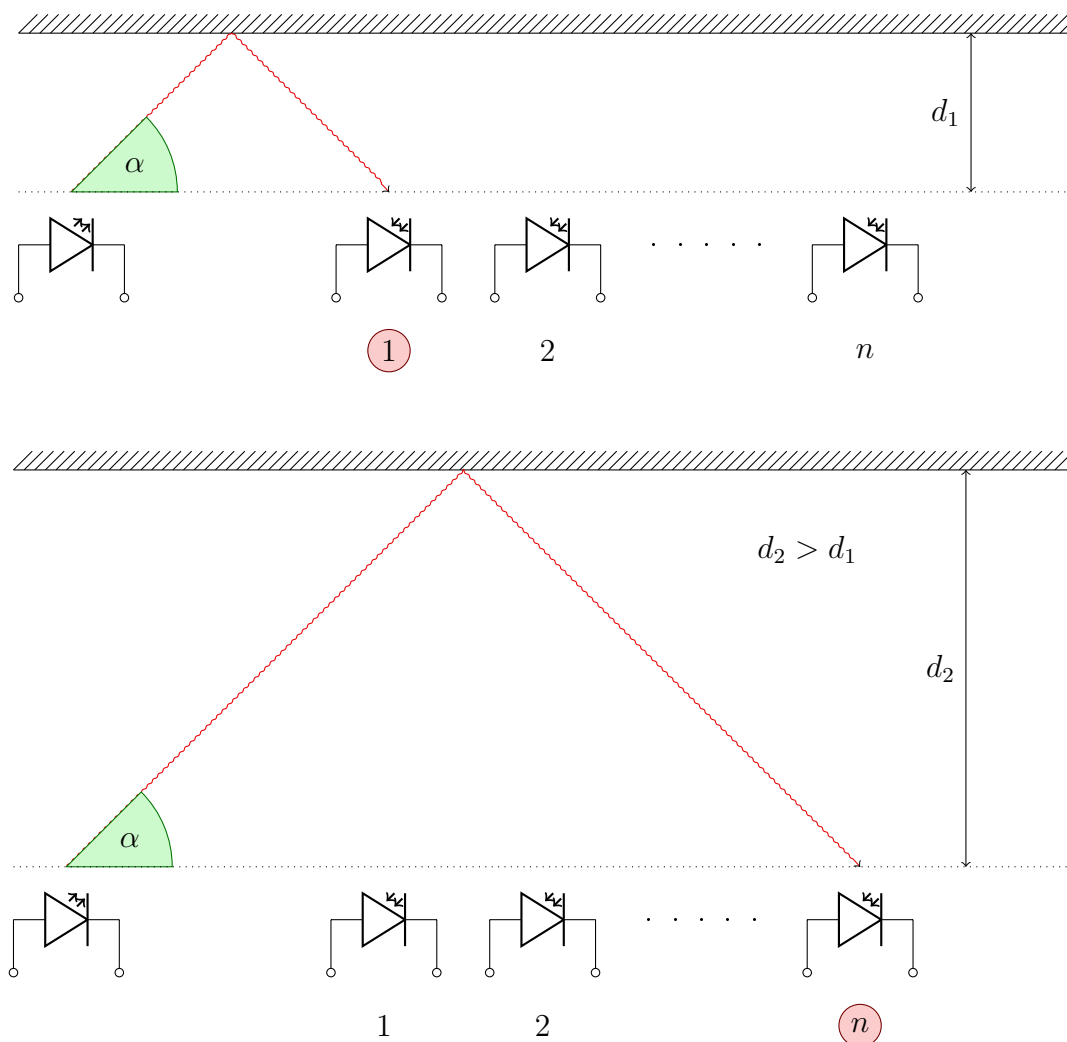
Pomiar odległości za pomocą szeregu fotodetektorów

Pomiar odległości przy wykorzystaniu szeregu fotodetektorów jest metodą optyczną działającą na niedużych odległościach, mogąca dawać bardzo dokładny pomiar. Dokładność pomiaru odległości w powszechnie stosowanych czujnikach tego typu może dochodzić do μm . Metoda ta jest także najszybszą metodą pomiaru odległości. Największą jej wadą jest wąski zakres pomiarowy (najczęściej kilka cm). Najlepsze rezultaty daje przy badaniu obiektów dobrze odbijających światło. Stosowana jest głównie przy taśmach montażowych, gdzie badana jest dokładność wykonania danego wyrobu. Najczęściej przy użyciu czujników tego typu badana jest jakość spawów i wykonania profili metalowych w zautomatyzowanych liniach produkcyjnych.

Idea pomiaru odległości za pomocą szeregu fotodetektorów

W tej metodzie zazwyczaj mamy do czynienia ze źródłem światła laserowego, które ustawione jest pod odpowiednio dobranym kątem. Światło lasera, po odbiciu

od obiektu trafia na linię lub matrycę fotodetektorów (np. linia fotodiod lub matryca CMOS). Jeżeli obiekt, do którego badana jest odległość znajduje się w zasięgu pracy czujnika, to odległość jest wprost proporcjonalna do pozycji fotodetektora wskazującego największe natężenie odbitego światła.



Rysunek 3.6: Pomiar odległości za pomocą szeregu fotodetektorów - idea działania

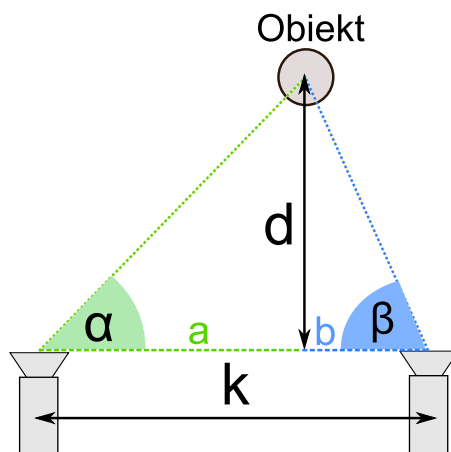
Pomiar odległości za pomocą pary kamer

Para kamer pomimo, że często wykorzystywana jest w innych celach może posłużyć do pomiaru odległości. Jest to metoda optyczna, głównie stosowana jako dodatkowa metoda pomiaru odległości w robotach posiadających system wizyjny.

Idea pomiaru odległości za pomocą pary kamer

Proces pomiaru odległości za pomocą dwóch kamer jest podobny do ludzkiej, wzrokowej oceny odległości. Polega on na triangulacji, czyli pomiarze różnicy położenia obiektu w zależności od kąta obserwacji. Najczęściej w tego typu pomiarach stosowana jest para kamer cyfrowych oddalonych od siebie na pewną odległość i ustawiona w odpowiednim kierunku. Pobierając obraz z obu kamer i porównując na obu położenie tego samego obiektu można za pomocą odpowiednich obliczeń wskazać odległość do tego obiektu. Dodatkową zaletą jest możliwość jednoczesnego określenia rozmiarów obiektu jak i rozpoznania kształtu.

Pomimo dużego nakładu sprzętowego potrzebnego do użycia tej metody nie daje ona zadowalających rezultatów. Dokładność pomiarów jest w dużej mierze zależna od rozdzielczości kamer, a wraz z rozdzielczością rośnie zapotrzebowanie na moc obliczeniową urządzenia pomiarowego. Najlepsze rezultaty osiągnąć można przy pomiarze odległości do niewielkich, wyraźnych elementów. Przy pomiarze odległości do dużych, płaskich powierzchni (np. ściany) wymagane jest znalezienie miejsca załamania światła np. róg pomieszczenia. Metodę tę można łatwo ekstrapolować do więcej niż dwóch kamer. Na rynku dostępne są gotowe systemy wizyjne, które posiadają nawet 25 kamer ułożonych matrycowo (PointFUSION 25 firmy Point Grey).



$$k = a + b$$

$$k = \frac{d}{\operatorname{tg}(\alpha)} + \frac{d}{\operatorname{tg}(\beta)}$$

$$d = \frac{k}{\frac{1}{\operatorname{tg}(\alpha)} + \frac{1}{\operatorname{tg}(\beta)}} \quad (3.1)$$

k - znana odległość między kamerami

d - mierzona odległość

Rysunek 3.7: Idea pomiaru odległości przy użyciu dwóch kamer

Optyczny pomiar odległości - badanie przesunięcia fazowego

Metoda szeroko stosowana w wielu dziedzinach, głównie w wojsku, geodezji i budownictwie. Ze względu na dużą dokładność stosowana jest również w trójwymiarowych skanerach. Na rynku istnieje szeroka gama gotowych urządzeń mierzących odległość tą metodą z dużą dokładnością. Najczęściej stosowane w budownictwie dalmierze laserowe potrafią mierzyć odległość nawet do 100m z dokładnością $1 \div 5\text{mm}$ (w zależności od wersji urządzenia). Metoda ta wykorzystywana jest także w lotnictwie, gdzie używa się jej do tworzenia map środowiskowych za pomocą lidarów (obrotowych dalmierzy laserowych - radarów optycznych). Dalmierze laserowe oraz tańsze wersje lidarów stosowane są w robotyce i przy liniach produkcyjnych. Metoda

badania przesunięcia fazowego fali świetlnej wykorzystywana jest w pomiarach wolno zmiennych ruchów tektonicznych, a także do badania stanu zbiorników z cieczą nieprzeźroczystą lub przeźroczystą (z dodatkowym pływakiem).

Idea pomiaru odległości za pomocą pomiaru przesunięcia fazowego fali świetlnej

Metodologia pomiaru odległości metodą optyczną mogłaby być zbliżona do pomiaru odległości przy użyciu fal ultradźwiękowych. Prędkość rozchodzenia się fali świetlnej jest jednak dużo większa od prędkości rozchodzenia się fali dźwiękowej przez co wymogi stawiane układom elektronicznym zliczającym czas, który upłynął od czasu wysłania wiązki światła do jej powrotu, nie są jeszcze możliwe do spełnienia w praktycznych zastosowaniach.

W praktyce stosuje się pomiar przesunięcia fazowego pomiędzy falą wysłaną a odbieraną. W najprostszych rozwiązaniach wysyłana jest fala prostokątna o odpowiednio dobranym wypełnieniu i częstotliwości. Fala odebrana i wysyłana trafiają do detektora fazy, na wyjściu którego można zaobserwować krótkie impulsy których szerokość jest wprost proporcjonalna do odległości. Różnice wynikające ze zmiany odległości są jednak na tyle małe w porównaniu do opóźnień wynikających z czasu propagacji układów i zakłóceń, że aby otrzymać wiarygodny wynik, sygnał wyjściowy z detektora fazy poddawany jest całkowaniu. Odpowiednio scałkowany sygnał jest wprost proporcjonalny i silnie zależny od odległości.

Największą trudnością w tworzeniu urządzeń mierzących odległość tą metodą jest dobór odpowiedniego fotodetektora, a także czasochłonna kalibracja układów optycznych.

3.3 Wybór metody pomiaru odległości

Projektowane urządzenie najczęściej będzie używane przy pomiarze pomieszczeń zamkniętych. Mierzone odległości w takich pomieszczeniach mogą sięgać od 1m (w pomieszczeniach mieszkalnych) do 50m (w halach przemysłowych). Wymagana dokładność w tego typu pomiarach wynosi $1 \div 10\text{mm}$. Przy tak dużych odległościach metoda stykowa nie dałaby dobrych rezultatów. Metoda ultradźwiękowa mogła by się nadawać do pomiaru niewielkich odległości. Przy większych dystansach pojawiają się duże problemy z odbiciem fali dźwiękowej od wielu obiektów znajdujących się w pomieszczeniach. Metoda laserowa nie nadaje się do pomiaru odległości do obiektów przezroczystych (np. szyby), a także odbijających dobrze światło (np. lustra czy chromowane poręcze). Niemniej jednak elementy refleksyjne stanowią niedużą część pomieszczenia.

W projekcie wybrana została metoda laserowa ze względu na bardzo dobrze dobrane parametry pomiaru do wymagań. Wykonanie dalmierza laserowego jest rzeczą trudną z powodu skomplikowanej kalibracji układów optycznych. Autor postanowił skorzystać z gotowych rozwiązań. Pierwszym urządzeniem do pomiaru odległości, które było brane pod uwagę był lidar laserowy firmy SICK, model LMS200.

Lidary są często stosowane w podobnych projektach. Stosowane były m.in. w zawodach DARPA Grand Challenge [10]. Projekt ten ma na celu skonstruowanie samochodu samodzielnie poruszającego się po mieście. Lidary zamontowane na samochodach pozwalały na skuteczne omijanie przeszkód, a także odnajdywanie się w przestrzeni w czasie rzeczywistym. Niestety jednak zbyt wysoka cena lidarów uniemożliwiła wykorzystanie tej metody w projekcie. Autor postanowił wykorzystać tańsze rozwiązanie - dalmierz laserowy do zabudowy w układach elektronicznych, który mierzyłby odległość tylko do jednego punktu, a pozostała część układu zapewniłaby możliwość obrotu dalmierza i wykonywania pomiarów całych pomieszczeń. Podobnie jak w przypadku lidarów, gotowe układy dalmierzy laserowych z wyjściami cyfrowymi bądź analogowymi okazały się zbyt kosztowne. Najtańsze dalmierze laserowe obecne



Wymiary	15 cm 15 cm 21 cm
Waga	4,5kg
Napięcie zasilania	24V
Moc	20W
Interfejs	RS-232 RS-422
Zakres pomiaru	0 ÷ 80 m
Rozdzielczość	1 mm
Kąt widzenia	180°
Rozdzielczość kątowa	1 ÷ 0.25°
Częstotliwość skanowania	75 Hz
Czas odpowiedzi	13 ÷ 53 ms
Błąd systematyczny	±15 mm
Błąd statystyczny	5 mm

Rysunek 3.8 i Tabela 3.1: Lidar SICK LMS200

aktualnie na rynku to dalmierze ręczne, wykorzystywane w budownictwie do wymiarowania pomieszczeń. Do największych producentów ręcznych dalmierzy laserowych należą m.in. firma BOSCH i Leica. Standardowe dalmierze produkowane przez obie



Wymiary	14 cm 6 cm 3 cm
Waga	200 g
Napięcie zasilania	4x1,5V
Zakres pomiaru	0,05 ÷ 200 m
Rozdzielczość	1 mm
Błąd	±1 mm
Dokładność czujnika pochylenia:	±0,2°

Rysunek 3.9 i Tabela 3.2: Dalmierz laserowy Leica DISTO D8

firmy mierzą odległość do 50m z dokładnością 1 ÷ 3mm. Najnowsze modele dalmierzy Leica potrafią mierzyć odległość nawet do 200m. Posiadają też wbudowany inklinometr - układ służący do wykonywania pośrednich pomiarów odległości metodą triangulacji, a także interfejs bluetooth wykorzystywany do przesyłu danych

do komputera. Wadą ręcznego dalmierza laserowego jest długi czas pomiaru dochodzący do 5 sekund. W pomiarach ręcznych jest to pomijalnie mały czas, jednak w systemie elektronicznym, który będzie takich pomiarów robił znaczną ilość może to stanowić duże utrudnienie. Z założenia projekt jednak ma służyć do pomiarów obiektów statycznych, więc długi czas pomiarów jest akceptowalny. Żaden dalmierz do pomiaru ręcznego nie dysponuje zewnętrznym interfejsem, dzięki któremu można łatwo kontrolować pracę urządzenia. W celu wykorzystania możliwości ręcznych urządzeń w systemie elektronicznym można jednak spróbować użyć wewnętrznych sygnałów dalmierza.

3.4 Komunikacja z dalmierzem laserowym

Dalmierz BOSCH DLE-50 i BOSCH PLR-30

Dalmierz laserowy PLR-30 firmy BOSCH, jest to jednym z najmniejszych i najtańszych dalmierzy ręcznych na rynku. Model DLE-50 jest ulepszoną wersją dalmierza PLR-30 różniącą się dwoma znaczącymi parametrami - większym maksymalnym zasięgiem oraz większą dokładnością pomiaru.



Wymiary	11 cm 7 cm 3 cm
Waga	180 g
Napięcie zasilania	4x1,5V
Zakres pomiaru	0,02 ÷ 30 m
Rozdzielczość	1 mm
Czas odpowiedzi	0,1 ÷ 4 s
Błąd	±2 mm

Rysunek 3.10 i Tabela 3.3: Dalmierz laserowy BOSCH PLR-30



Wymiary	10 cm 6 cm 3 cm
Waga	180g
Napięcie zasilania	4x1,5V
Zakres pomiaru	0,05 ÷ 50 m
Rozdzielczość	1 mm
Czas odpowiedzi	0,1 ÷ 4 s
Błąd	±1,5 mm

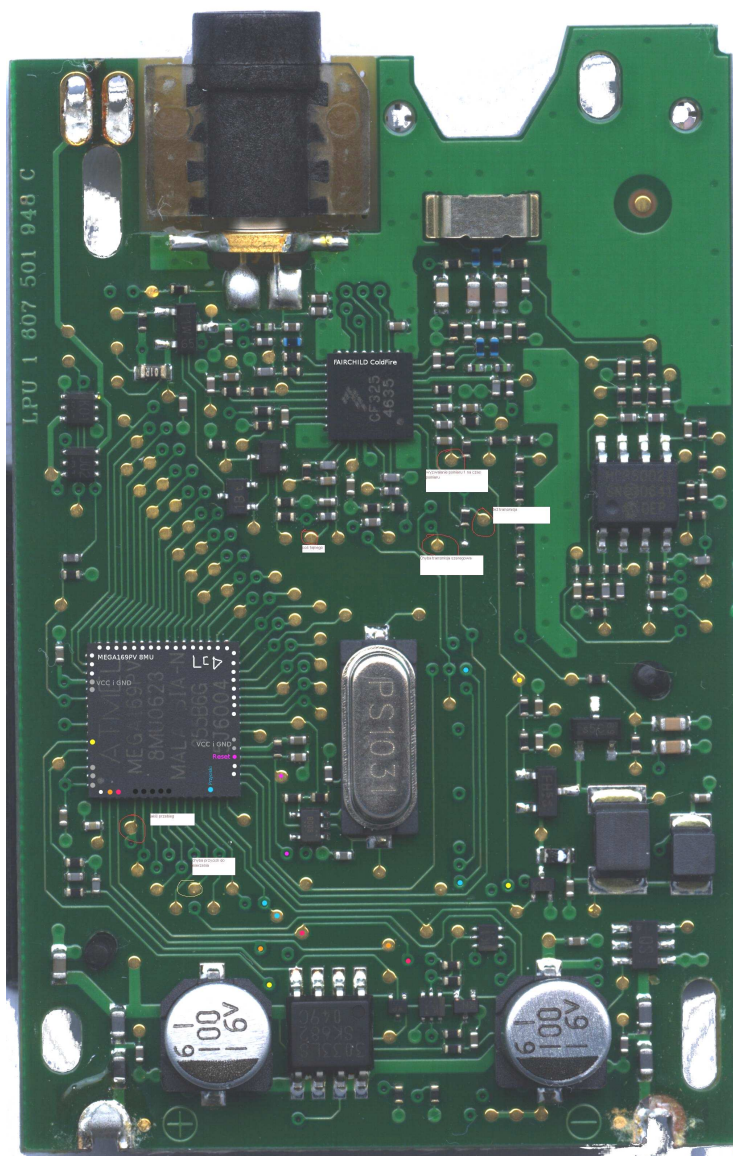
Rysunek 3.11 i Tabela 3.4: Dalmierz laserowy BOSCH DLE-50

Budowa wewnętrzna dalmierza laserowego BOSCH DLE-50

Po dokładnych badaniach obu modeli dalmierzy można stwierdzić, że ich fizyczna budowa elektroniczna jest bardzo podobna. Do najbardziej istotnych elementów elektronicznych należą:

- układ scalony Atmel MEGA169PV
- układ scalony CF325
- nadawcza dioda laserowa
- odbiorcza fotodioda lawinowa

Cztery porty *JTAG* oraz porty *RXD* i *TXD* układu Atmel wyprowadzone są na wierzchnią warstwę obwodu drukowanego na specjalne pola przeznaczone do testowania i programowania urządzenia w czasie produkcji. Trzydzieści portów układu Atmel przeznaczonych jest do sprzętowego sterowania wyświetlaczem segmentowym. Niestety w chwili pisania pracy niedostępna była dokumentacja układu CF325. Analiza symbolu nasuwa jedynie przypuszczenia, że jest to mikroprocesor CrossFire firmy Fairchild.

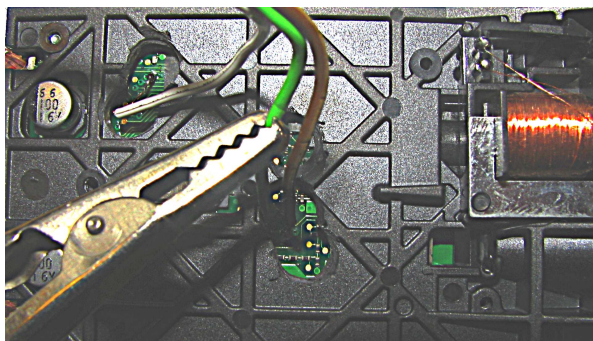


Rysunek 3.12: Budowa wewnętrzna dalmierza BOSCH DLE-50

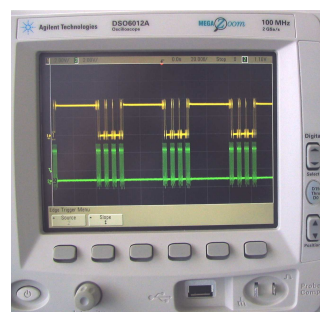
Wewnętrzne sygnały dalmierza laserowego BOSCH DLE-50

Oba modele dalmierzy zostały poddane wnikliwym badaniom podczas prowadzenia pomiarów. Włączanie i wyłączanie urządzenia było możliwe dzięki zwarceniu dwóch odpowiedzialnych za to styków na wierzchniej warstwie płytki z obwodami. Podobnie z wszystkimi innymi funkcjami urządzenia do których użytkownik końcowy ma dostęp dzięki klawiaturze umieszczonej na obudowie urządzenia. Analiza wszystkich sygnałów transmisji szeregowych dochodzących do układu MEGA169PV pozwoliła rozpoznać sposób transmisji oraz wyodrębnić ramki sterujące startem i końcem po-

miaru. Niestety jednak analiza danych przesyłanych szeregowo do układu Atmel nie wykazała zależności od mierzonej odległości. Dla różnych odległości w obu modelach dalmierzy dane docierające do układu MEGA169PV były takie same. Wykluczyło to więc możliwość łatwego odczytu zmierzonej odległości. Odczyt danych wysyłanych z mikrokontrolera na wyświetlacz segmentowy zakończył się niepowodzeniem z uwagi na dynamiczny charakter wyświetlania danych i 4 różne poziomy napięcie wysyłane bezpośrednio na dużą ilość segmentów diodowych przez wbudowany w układ kontroler LCD.



Rysunek 3.13: Przygotowanie dalmierza do badań



Rysunek 3.14: Sygnały dalmierza na ekranie oscyloskopu

Dalmierz Leica DISTO Lite 5

Dalmierz Leica DISTO Lite5 jest to jeden z pierwszych ręcznych dalmierzy laserowych dostępnych na rynku.



Wymiary	14 cm 7 cm 5 cm
Waga	315 g
Napięcie zasilania	2x1,5V
Zakres pomiaru	0,2 ÷ 200 m
Rozdzielczość	1 mm
Czas odpowiedzi	0,1 ÷ 4 s
Błąd	±5 mm

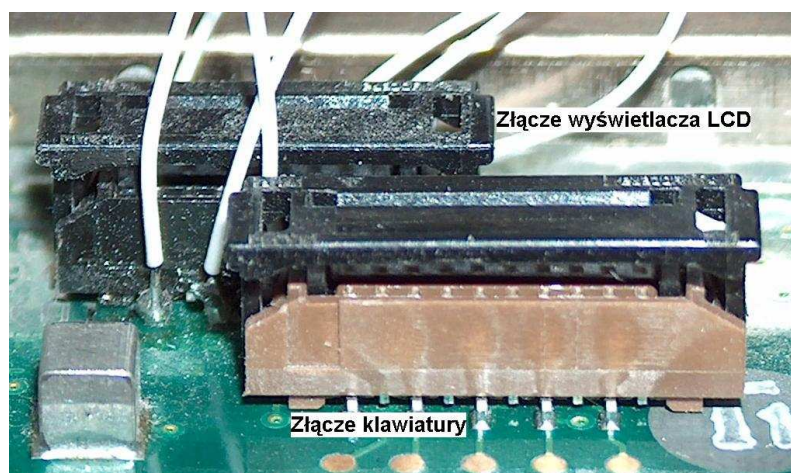
Rysunek 3.15 i Tabela 3.5: Dalmierz laserowy Leica DISTO Lite5

Budowa wewnętrzna dalmierza laserowego Leica DISTO Lite 5

Dalmierz firmy Leica posiada bardziej modułową budowę niż dalmierze firmy BOSCH. Pod metalowym ekranem znajduje się kilka układów elektronicznych firmy Leica. Klawiatura jak i wyświetlacz urządzenia dołączony jest do głównego obwodu drukowanego za pomocą giętkiego laminatu. Taśma klawiatury zawiera 10 przewodów, a taśma wyświetlacza 8 przewodów. Obok gniazd przyłączeniowych dla modułu klawiatury i wyświetlacza, w plastikowej obudowie znajduje się głośnik informujący krótkimi sekwencjami dźwiękowymi o wciśnięciu klawisza lub zakończeniu pomiaru.

Wewnętrzne sygnały dalmierza laserowego Leica DISTO Lite 5

Dzięki wcześniejszym opracowaniom na Politechnice Śląskiej [11], a także badaniom autora poznane zostały funkcje najważniejszych sygnałów modułów klawiatury oraz wyświetlacza dalmierza Leica DISTO Lite5. Emulacja naciśnięcia klawisza dalmierza odbywa się poprzez zwarcie na odpowiedni okres czasu odpowiedniej pary



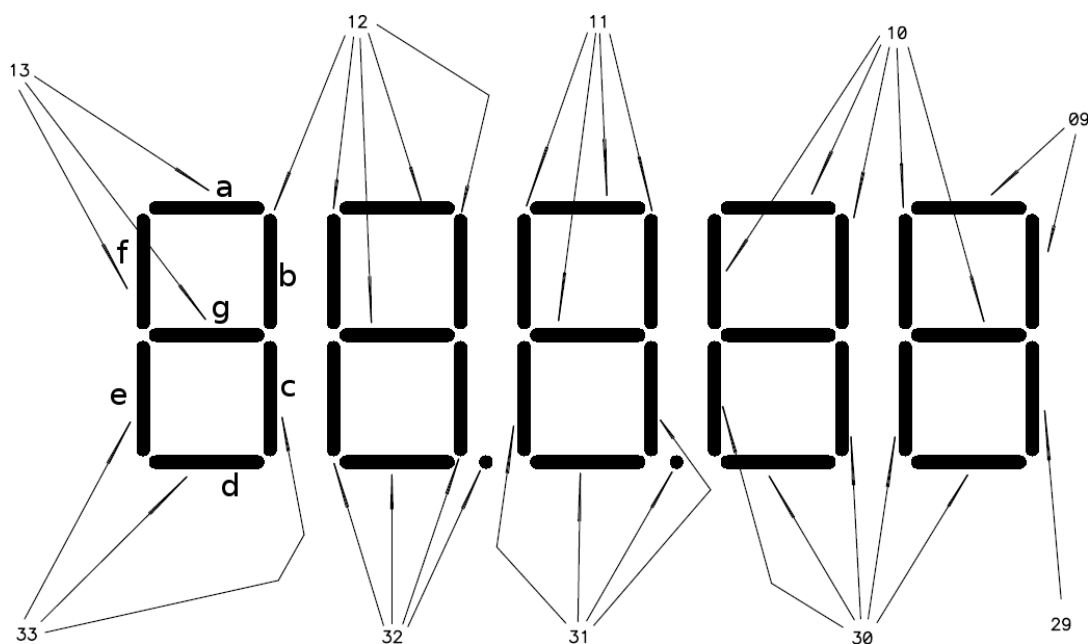
Rysunek 3.16: Złącza dalmierza Leica DISTO Lite5 [11]

styków gniazda klawiatury. Odczyt zmierzonej odległości odbywa się za pomocą analizy danych wysyłanych na moduł wyświetlacza. Transmisja danych przypomina transmisję *SPI*. Analiza wyników polega na monitorowaniu stanów trzech linii sygnałowych wyświetlacza:

- sygnału zegarowego (*SCLK*)
- sygnału danych (*MOSI*)
- sygnału końca ramki (*FRAME*)

Dane do wyświetlacza LCD przesyłane są w postaci kolejno po sobie występujących ramek, oddalonych od siebie o 3,1ms. Każda ramka zbudowana jest z czterdziestu bajtów, oddalonych od siebie o 156μs. Każdy bajt taktowany jest rozdzielonym sygnałem zegarowym o okresie 1μs, występującym jedynie na czas jego emisji. Sygnalizacja stanu ustalonego bitów danych odbywa się za pomocą narastającego zbocza zegarowego. Ramka trwa 6,4ms. Sygnały aktywne są w stanie niskim. Po każdej ramce przesyłany jest impuls na linii *FRAME* trwający 1080μs. Sygnał ten pojawia się 120μs od powrotu stanu zegarowego do stanu wysokiego [11].

W ramach projektu zostało rozpoznane znaczenie konkretnych bitów ramki, wymaganych do poprawnego odczytu pomiarów. Pomiar należy odczytywać jako war-



Rysunek 3.17: Bajty ramki transmisyjnej kodujące segmenty wyświetlacza dalmierza Leica DISTO Lite5 [11]

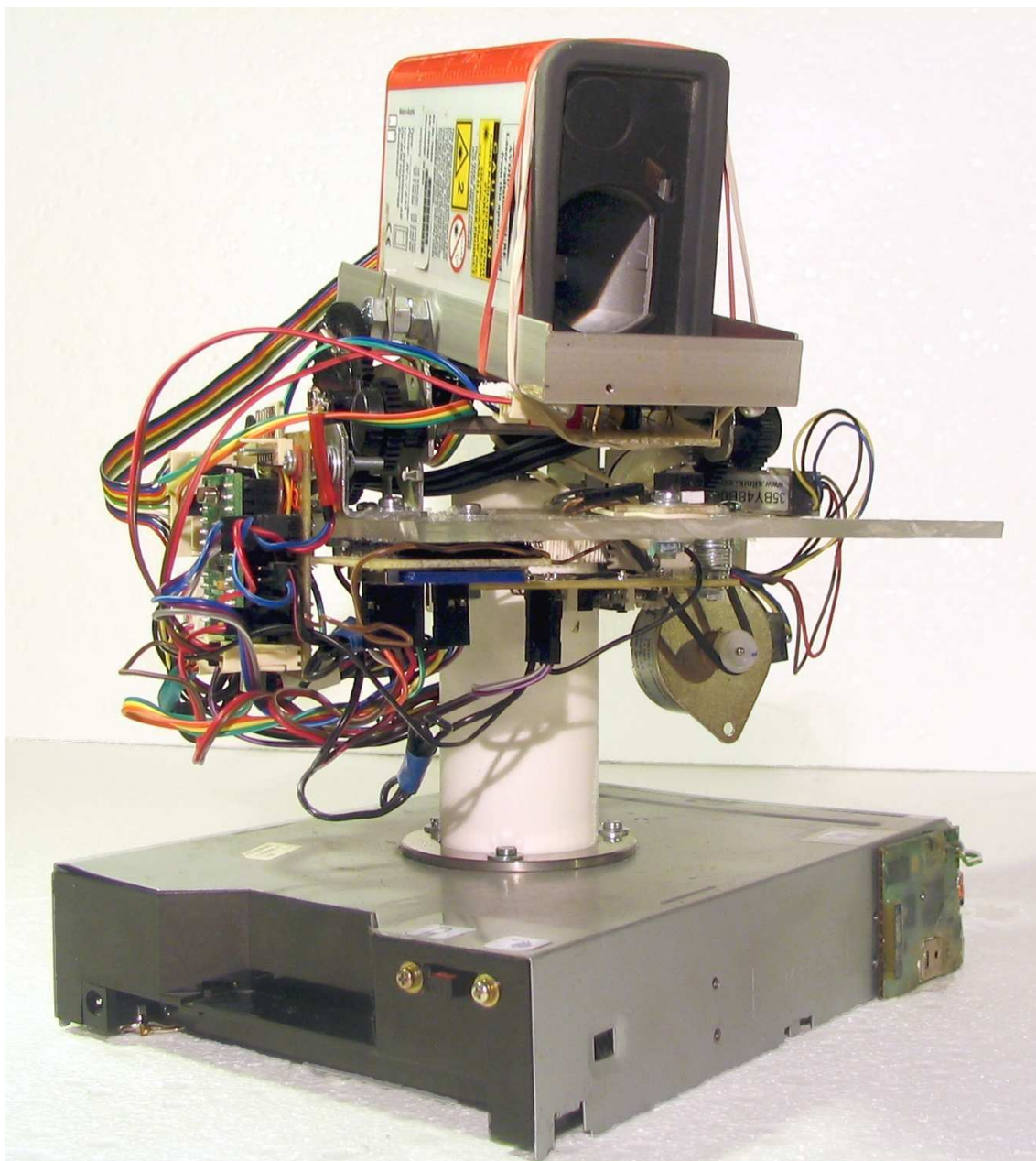
tość podaną w *mm* na podstawie odczytu wartości wyświetlanych przez 5 cyfr składających się z 7 segmentów. Przyjęto numerację bajtów ramki: od 1 do 40 i bitów danego bajtu: od 0 do 7.

Segment:	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
Pozycja	bajt,bit	bajt,bit	bajt,bit	bajt,bit	bajt,bit	bajt,bit	bajt,bit
10^0	9,0	9,1	29,0	30,7	30,5	10,5	10,7
10^1	10,3	10,4	30,3	30,2	30,1	10,1	10,2
10^2	11,4	11,6	31,4	31,3	31,2	11,2	11,3
10^3	12,6	12,7	32,6	32,5	32,4	12,4	12,5
10^4	13,6	12,3	33,6	33,5	33,4	13,4	13,5

Tabela 3.6: Znaczenie bitów ramki wyświetlacza Leica DISTO Lite5.

Do sprzętowej obsługi odczytu pomiarów można wykorzystać kontroler *SPI* (Serial Peripheral Interface) w trybie slave lub odbiorczą część kontrolera *SSC* (Synchronous Serial Controller). Moduły te można znaleźć w wielu popularnych mikrokontrolerach.

3.5 Konstrukcja mechaniczna

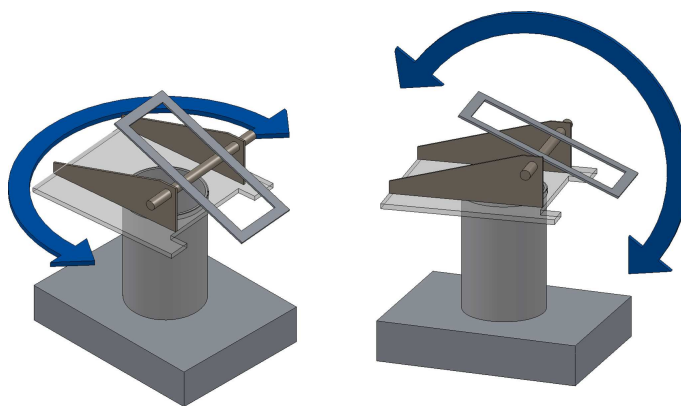


Rysunek 3.18: Kompletna część fizyczna systemu

Zasadnicza konstrukcja robota składa się z dwóch części:

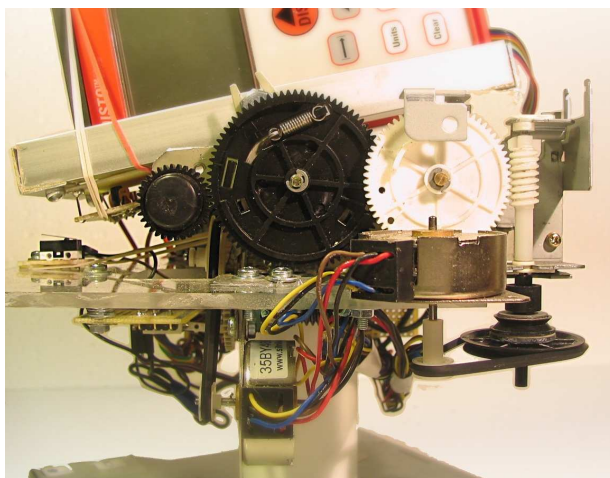
- Podstawy
- Platformy obrotowej

Podstawą jest metalowa obudowa, w której znajdują się przestrzeń umożliwiającą montaż dodatkowego napędu robota, układów elektronicznych i akumulatora z którego robot mógłby być zasilany. Podstawa połączona jest z platformą obrotową za pomocą plastikowej rurki stanowiącej osłonę dla kabli zasilających i sygnałowych.



Rysunek 3.19: Koncepcja budowy mechanicznej urządzenia

Na platformie obrotowej znajdują się dwa silniki krokowe. Pierwszy silnik steruje obrotami całej platformy w płaszczyźnie poziomej. Drugi silnik steruje wychyleniem w płaszczyźnie pionowej dodatkowego ramienia. W celu zwiększenia ilości kroków na obrót a tym samym dokładności pomiarów oraz momentu obrotowego silników krokowych zastosowano trzy rodzaje przekładni mechanicznych: ślimakowe, zębate, pasowe.



Rysunek 3.20: Silniki krokowe i przekładnie zamontowane na platformie obrotowej

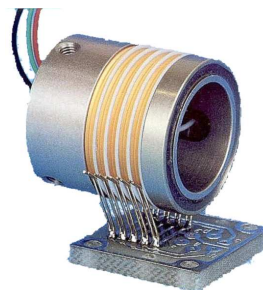
Aby doprowadzić sygnały z podstawy do części obrotowej, należało rozwiązać problem zawijających się kabli. Zastosowano obrotowe złącze sygnałowe. Umożliwia to ciągły obrót platformy w płaszczyźnie poziomej. W początkowej fazie projektu zastosowano obrotowe złącze RJ9. Zastosowanie tego złącza nie dało jednak dobrych rezultatów - często dochodziło do rozłączania zasilania układu. W projekcie zdecydowano się na zastosowanie czteropinowego złącza typu *Jack*. Jest to rozwiązanie tanie, możliwe do zastosowania w urządzeniach gdzie prędkość obrotowa jest bardzo niska, a ilość wymaganych sygnałów jest nie większa niż 4. W profesjonalnych urządzeniach stosuje się tzw. pierścienie ślizgowe (*ang. slip rings*), mogące przenieść nawet setki sygnałów, a jednocześnie sprostać wymaganiom dużych prędkości obrotowych. Wadą pierścieni ślizgowych jest ich duży rozmiar, a także wyższa cena.



Rysunek 3.21: Obrotowe złącze RJ9



Rysunek 3.22: Złącze obrotowe firmy Mercotac



Rysunek 3.23: Pierścienie ślizgowe

3.6 Konstrukcja elektroniczna

Część elektroniczna zaprojektowanego systemu rozmieszczona jest w dwóch miejscach.

W podstawie urządzenia znajdują się:

1. złącza przyłączeniowe zasilania i dodatkowych sygnałów,
2. przełącznik umożliwiający wybór zasilania:
 - z akumulatora
 - z zewnętrznego zasilacza wpinanego do gniazda znajdującego się na zewnątrz obudowy
3. 5V stabilizator napięcia LM7805, z którego rozprowadzone jest zasilanie całego systemu
4. moduł Bluetooth OEMSPA 33i firmy ConnectBlue

OEMSPA 33i jest modulem działającym jako bezprzewodowy mostek UART. Moduł ten posiada zaimplementowany stos Bluetooth. Zasilany jest napięciem 5V i pracuje w trybie transparentnego mostka. W zaprojektowanym systemie wykorzystywane są jego dwa sygnały UART - RXD i TXD działające na poziomach logicznych zgodnych z TTL.

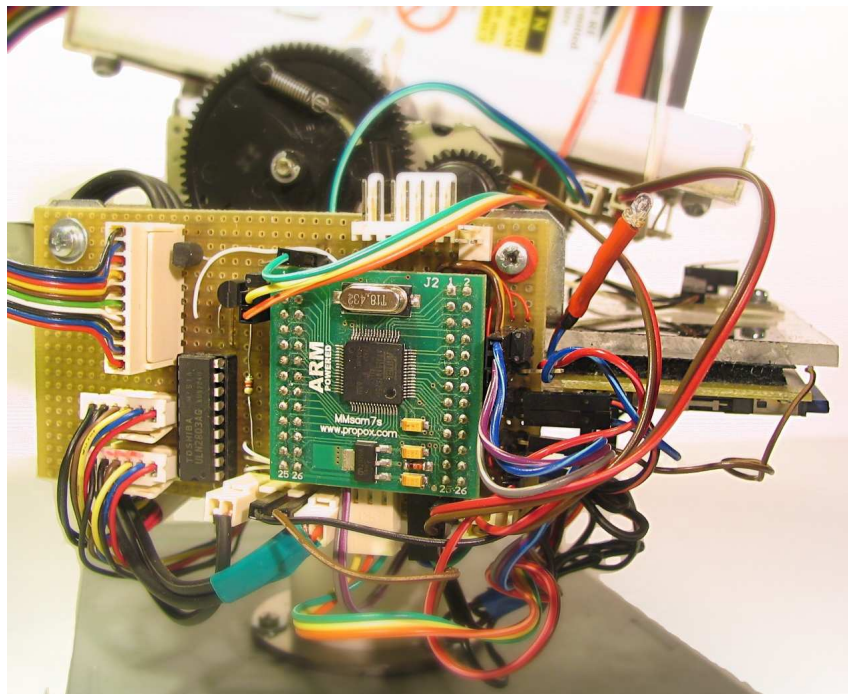


Wymiary	20 mm x 4 mm x 3 mm
Napięcie zasilania	3 ÷ 6V
Prędkość transmisji	300 ÷ 115200 kBits/s
Moc wyjściowa	Class 2, typ 1.5dBm
Częstotliwość wyjściowa	2.402 ÷ 2.480 GHz
Antena	Wewnętrzna
Wzmocnienie anteny	+2.7dBi
Interfejs	RS232 UART

Rysunek 3.24 i Tabela 3.7: Moduł Bluetooth OEMSPA 33i firmy ConnectBlue

Masa i 5V zasilanie oraz sygnały RXD i TXD modułu bluetooth doprowadzone są do złącza znajdującego się wewnątrz rurki łączeniowej. Dzięki temu możliwe jest przekazanie najważniejszych sygnałów do platformy obrotowej.

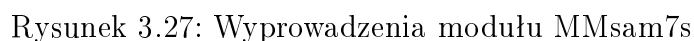
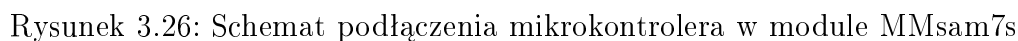
Na platformie obrotowej znajdują się główne obwody systemu. Jednostką sterującą jest moduł MMsam7s firmy Propox zawierający mikrokontroler z rdzeniem ARM - AT91SAM7S256.



Rysunek 3.25: Główne obwody elektroniczne systemu

Porty sygnałowe mikroprocesora odpowiedzialne za komunikację *UART*, *SPI*, *I²C*, *USB* zostały wyprowadzone na złącza szpilkowe umożliwiając rozbudowę urządzenia o dodatkowe moduły.

Wyprowadzono dziesięciostykowe złącze szpilkowe umożliwiające podłączenie dalmierza laserowego, a także dwa sześciostykowe złącza szpilkowe przeznaczone do podłączenia unipolarnych silników krokowych. Silniki krokowe zasilane są napięciem 5V i sterowane są przy pomocy tranzystorów w układzie Darlingtona znajdujących się wewnątrz układu ULN2803.



Rysunek 3.28: Opis przeznaczenia złącza J1 i J2 modułu MMsam7s

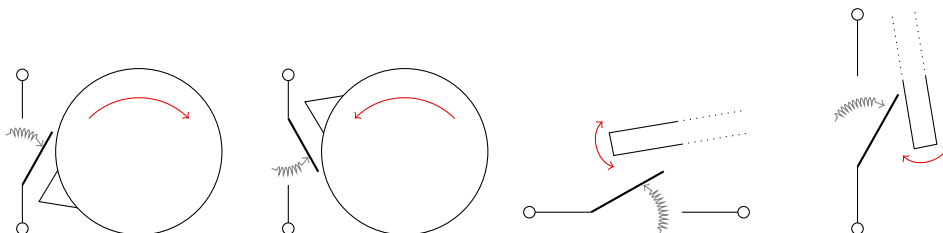
3.7 Pomiar położenia

Proces tworzenie elektronicznej mapy pomieszczeń polega na zbieraniu informacji o odległości do punktów w okół urządzenia. Do określenia położenia punktu w przestrzeni, który wskazywany jest przez dalmierz laserowy wymagana jest znajomość położenia dalmierza tj. dwóch kątów obrotu.

Czujniki krańcowe

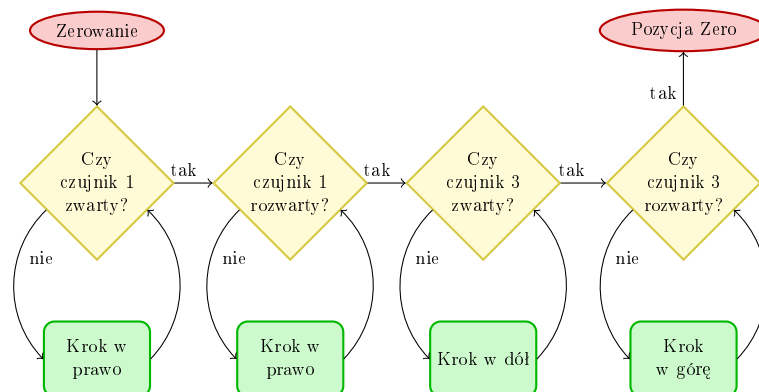
W urządzeniu zostały zamontowane 4 mechaniczne czujniki krańcowe informujące układ sterujący o dotarciu części obrotowej robota w skrajne położenia:

Czujnik 1: Prawo Czujnik 2: Lewo Czujnik 3: Dół Czujnik 4: Góra



Czujniki te w trakcie zwarcia zwierane są do masy układu, a w stanie rozwarcia są wewnętrznie podciągane do napięcia zasilania przez mikrokontroler. Układ sterowania po każdym ruchu sprawdza stan czujników. Jeżeli na *czujniku 1* wystąpiło zbocze narastające (nastąpiło zwolnienie czujnika) jest to uznawane za poziomą pozycję zero. Jeżeli na *czujniku 3* wystąpiło zbocze narastające jest to uznawane za pionową pozycję zero. W celu umieszczenia części obrotowej robota w zerowej pozycji poziomej i pionowej stosuje się prosty algorytm.

Informacja o zmianie stanu na którymkolwiek z czujników jest odnotowywana w wynikach pomiarów. Gdy zostanie zwarty *czujnik 4* oznacza, że układ znalazł się w położeniu końcowym.



Rysunek 3.29: Schemat blokowy algorytmu zerowania położenia ramienia robota

Pomiar ilości kroków silników

Dzięki zastosowaniu silników krokowych oraz ich bezpośredniemu sterowaniu za pośrednictwem mikrokontrolera, możliwy stał się również pomiar ilości kroków. Przy każdym pomiarze odległości wraz z pomiarem zapisywana jest także informacja o ilości kroków jaką wykonały silniki od położenia zerowego. Po każdym przejściu części obrotowej przez *czujnik 1* ilość kroków jest zapisywana, a następnie zerowana.

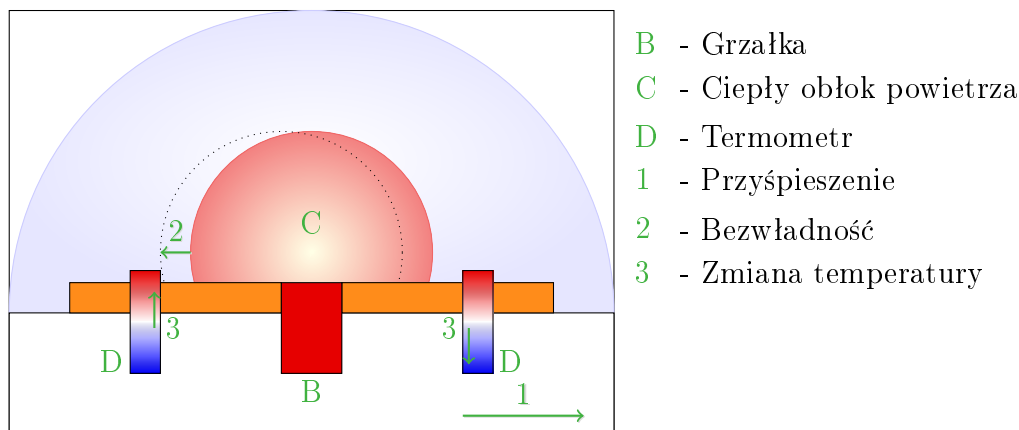
Pomiar przyśpieszenia ziemskiego

W urządzeniu jako dodatkowe źródło informacji na temat położenia zastosowano trójosiowy akcelerometr elektroniczny. Akcelerometr zamontowany jest na spodzie dalmierza dzięki czemu możliwy jest pomiar wektora przyśpieszenia w obu osiach obrotu.

Zasada działania akcelerometru

Zastosowany w projekcie akcelerometr MXR9500G/M firmy MEMSIC wykorzystuje do pomiaru wektora przyśpieszenia zjawisko konwekcji cieplnej. Źródło ciepła umieszczone wewnątrz układu zawieszono jest w środku przestrzeni wypełnionej powietrzem. Równo rozmieszczone stopy termoelektryczne (*ang. thermopile*) ułoko-

wane są w równych odległościach dookoła źródła ciepła. Przy zerowym przyspieszeniu gradient temperatury jest symetrycznie rozłożony wokół źródła ciepła przez co temperatura jest taka sama we wszystkich stosach termoelektrycznych i powoduje to powstanie tego samego napięcia na każdym z wyjść układu. Po wystąpieniu przyspieszenia ciepły obłok przemieści dzięki sile bezwładności, przez co zmieni się rozkład temperatury wokół źródła ciepła [13].

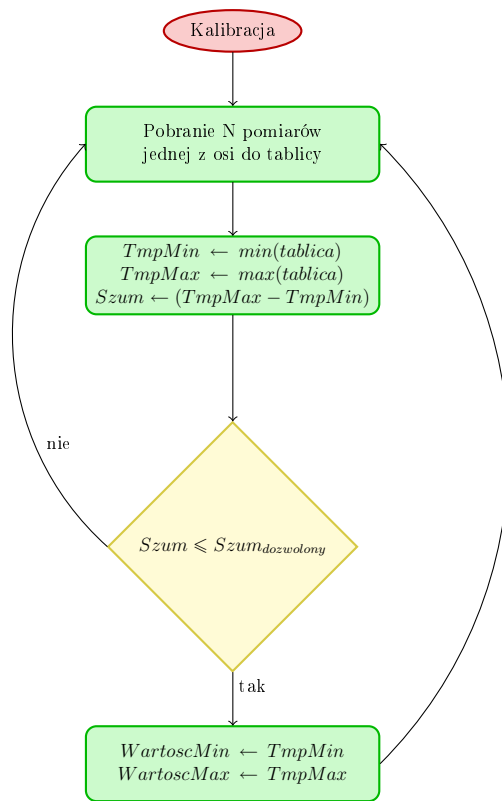


Rysunek 3.30: Kawitacyjny pomiar przyspieszenia

Kalibracja akcelometru

W systemie został stworzony specjalny tryb służący kalibracji trójosiowych akcelometrów z wyjściami analogowymi. Do kalibracji akcelometru jako przyspieszenie wzorcowe używane jest przyspieszenie ziemskie. Proces kalibracji polega na wyszukiwaniu skrajnych (minimalnych i maksymalnych) wartości wskazań każdej z osi akcelometru. Gdy dana oś akcelometru będzie skierowana równolegle do wektora przyspieszenia ziemskiego to wskazywana wartość będzie maksymalna. Wartość ta dla każdej osi jest zapamiętywana dzięki czemu przyszłe wyniki mogą być normalizowane. Pomiary z akcelometru są normalizowane do wartości z przedziału od -1 do +1. Podczas kalibracji należy ręcznie dokonywać obrotu akcelometrem w celu znalezienia wartości skrajnych. Każdy ruch wprowadza niepożądane przyspieszenia. W celu zapamiętywania tylko tych pomiarów, które wykonane były w stabilnym

położeniu akcelometru stosowany jest następujący algorytm:



Rysunek 3.31: Schemat blokowy procesu kalibracji akcelometru

Pomiar natężenia pola magnetycznego

Pomiar ilości kroków silników krokowych daje jedynie informację o relatywnym kącie obrotu względem podstawy robota. W celu pomiaru absolutnego kąta obrotu w płaszczyźnie poziomej w urządzeniu został dodatkowo umieszczony trójosiowy kompas elektroniczny MMC312xMQ firmy MEMSIC sterowany za pomocą magistrali I^2C . Umożliwia on trójosiowy pomiar wektora natężenia pola magnetycznego.

Zasada działania kompasu elektronicznego

Wykorzystywany w projekcie kompas elektroniczny MMC312xMQ firmy MEMSIC dokonuje pomiaru natężenia pola magnetycznego za pomocą anizotropowych

sensorów magnetorezystywnych (AMR) [14]. W trakcie produkcji układu cienka powierzchnia poddawana jest działaniu silnego pola magnetycznego w celu ustalenia domen magnetycznych w jednakowym kierunku, tworząc tym samym wektor magnetyzacji. Po tym zabiegu, zewnętrzne pole magnetyczne, przyłożone pionowo do boków warstwy magnetorezystywnej powoduje obrót magnetyzacji. Zmiana kąta, która wynika z obrotu magnetyzacji powoduje wahanie się rezystancji warstwy AMR. Sensory AMR ułożone są w mostek Wheatstonea dzięki czemu zmiana rezystancji jest wykrywana jako zmiana napięcia różnicowego i siła przyłożonego pola magnetycznego może być łatwo zmierzona [14].

Kalibracja kompasu elektronicznego

Kalibracja kompasu przebiega w sposób analogiczny do kalibracji akcelerometru. Obie rodzaje kalibracji różnią się interfejsem pomiarowym pomiarowy, a także występującymi błędami. Jako referencyjne źródło natężenia pola magnetycznego wykorzystywany jest ziemski biegun magnetyczny. W trakcie pomiaru natężenia pola magnetycznego elementy metalowe otaczające układ kompasu wprowadzają duże zakłócenia. Najlepiej więc jeżeli kalibracja przeprowadzana jest na kompasie umieszczonym w docelowym systemie. Zalecane jest także, aby w trakcie pracy kompas nie zmieniał swojego położenia względem metalowych części. Kalibracji kompasu można dokonać w zaprojektowanym systemie za pomocą specjalnie przeznaczonego do tego trybu.

Rozdział 4

Oprogramowanie

4.1 Oprogramowanie urządzenia

Podczas pisania oprogramowania dla mikrokontrolera z rdzeniem ARM autor starał się stworzyć wiele funkcji mogących dodatkowo rozszerzyć funkcjonalność urządzenia. Konfiguracja niektórych trybów pracy odbywa się za pomocą odpowiednich plików znajdujących się na karcie SD. Dzięki temu użytkownik może zmienić niektóre aspekty działania układu bez potrzeby programowania pamięci flash mikrokontrolera.

Opis programu

Program układu AT91SAM7S ma budowę modułową. W programie wykorzystano bibliotekę AT91LIB. Biblioteka AT91LIB została uzupełniona o dwa moduły napisane specjalnie na potrzeby projektu:

- Biblioteka obsługi dalmierza laserowego Leica DISTO Lite5,
- Biblioteka sterująca pracą silników krokowych.

Działanie programu oparte jest o pętlę główną oraz trzy funkcje przerwań:

- Przerwanie występujące podczas odebrania znaku przez kontroler UART,
- Okresowe przerwanie z kontrolera PIT (Programmable Interrupt Timer) służące obsłudze systemu plików FAT,
- Przerwanie IRQ wywoływane sygnałem końca ramki pochodzącym z dalmierza laserowego.

Działanie pętli głównej programu oparte jest na trybach i komendach.

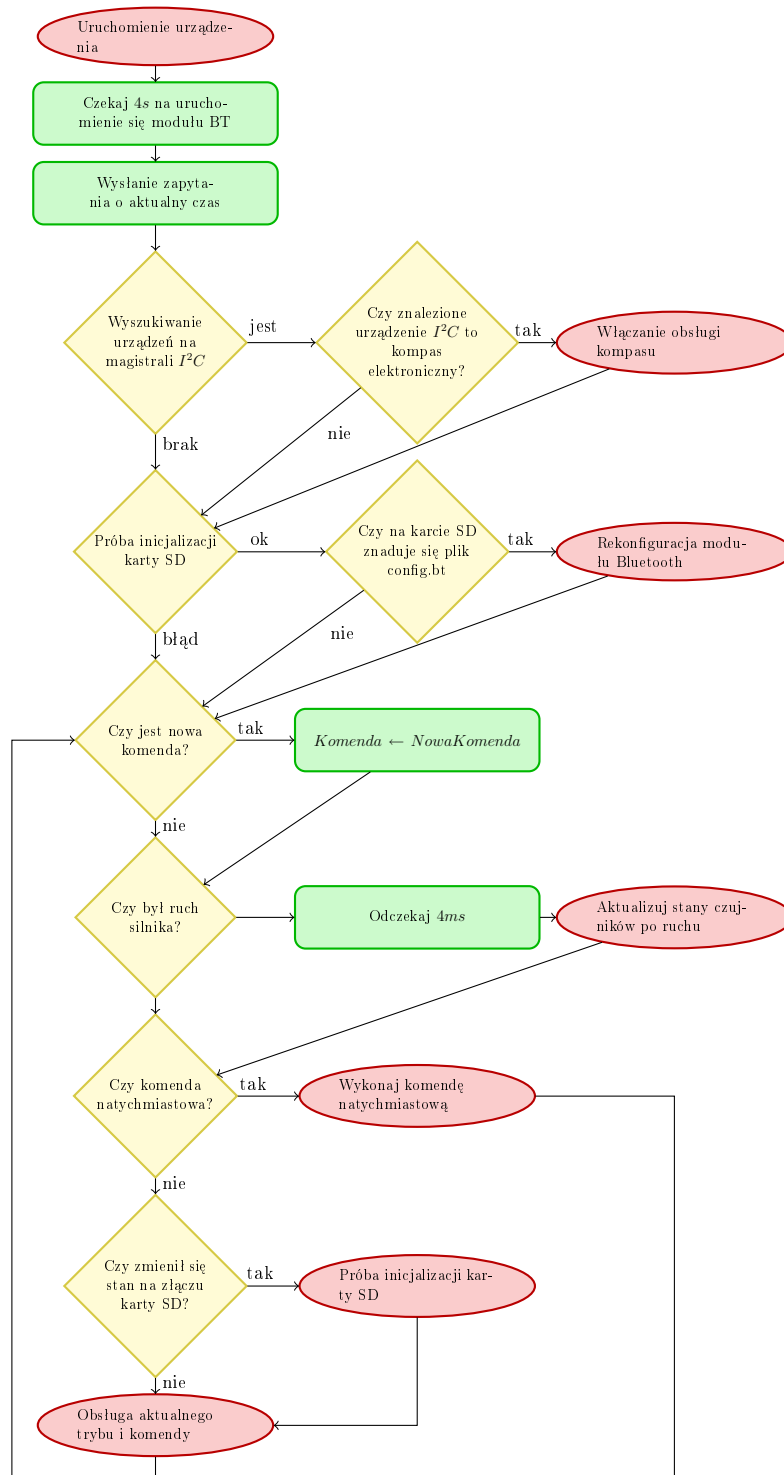
Komendy są to odpowiednio zinterpretowane polecenia użytkownika, odebrane przez interfejs *DBGU (UART)* mikroprocesora. Polecenia te powinny mieć odpowiedni format:

Słowo zawierające co najmniej jeden znak alfanumeryczny, opcjonalna numeryczna wartość komendy poprzedzona znakiem równości, znak końca linii (`\n`).

W komendach przyjmowane są znaki ASCII od a do z, od 0 do 9, znak równości (=), znak zapytania (?), znak kropki (.), znak przecinka (,) oraz znak nowej linii (`\n`). Wielkość liter nie ma znaczenia - małe litery są zamieniane na duże, a wszystkie inne znaki są ignorowane. Znak równości oddziela część rozkazu od jego argumentu, a znak kropki lub przecinka może być zastosowany jako separator dziesiętny wartości argumentu. Odbiór każdej komendy potwierdzany jest odpowiednim komunikatem.

Układ przesyła przez interfejs UART informacje o swoim stanie za pomocą odpowiednio sformatowanych komunikatów. Każdy komunikat zakończony jest znakiem nowej linii (`\n`). Komunikaty te powinny być odpowiednio interpretowane: znak spacji na początku komunikatu świadczy o tym, że jest to komentarz i nie powinien być on interpretowany, a jedynie służyć użytkownikowi za dodatkową, tekstową informację. Pozostałe komunikaty przyjmują format składający się z:

słowa lub znaku alfanumerycznego, znaku średnika (;), opcjonalnych wartości parametrów komunikatu oddzielonych znakiem średnika (;), znaku końca linii (\n).



Rysunek 4.1: Schemat blokowy funkcji main() układu

Przykładowo komunikat: *D;150;10* zakończony znakiem końca linii, informuje, że dalmierz wykonał pomiar odległości. Zmierzona odległość to 150mm, a pomiar został odebrany w 10 ramce wyświetlacza dalmierza od momentu rozpoczęcia pomiaru.

W wartościach parametrów jako separator dziesiętny używany jest znak kropki (.). Dodatkowo w celu transferu bloku danych (np. zawartości plików) użyto znaku mniejszości (<) po którym, do czasu pojawienia się znaku większości (>), każdy komunikat powinien być traktowany jako linia pliku z danymi.

Pętla główna na początku każdego obiegu sprawdza czy została wprowadzona nowa komenda, jeżeli nie to ustawiana jest komenda domyślna - tzw. brak komendy.

Dostępne tryby pracy

Układ opiera swoje działanie o tryby pracy. Domyślnym trybem pracy jest *tryb stop*, w którym układ oczekuje na komendy od użytkownika. W trybie tym także odmierzany jest czas zwłoki. Jeżeli przez dłuższy okres brak jest ingerencji ze strony użytkownika układ rozpoczyna prace autonomiczną - przechodzi w tryb, który ustawiony jest jako *tryb automatyczny*. Trybem automatycznym powinien być jeden z trybów zerowania. Po zakończeniu trybu automatycznego układ przechodzi w tryb, który ustawiony jest jako *tryb następny*. Tryb następny powinien być jednym z trybów pomiarowych. Dzięki zastosowaniu tej metody układ może działać autonomicznie, bez ingerencji użytkownika. Użytkownik w dowolnym momencie może zmienić aktualny tryb pracy, *tryb automatyczny* i *tryb następny*.

Tryby zerujące

W trybach zerujących system ustawia określone położenie dalmierza laserowego. Po ustaleniu pozycji układ wykonuje dokładny pomiar wektora przyspieszenia i natężenia pola magnetycznego i tworzy plik wynikowy pomiarów o nazwie w formacie: <data>_<pozycja x_y_z>.hist. Data jest liczbą określającą ilość sekund, która minęła od 1 stycznia 2000 roku. Pozycja robota jest określana względem pierwszego miejsca pomiarowego i podawana w mm.

Dla przykładu nazwa pliku wynikowego: 0000019987_0000000000_0000000000_0000000000.hist

świadczy o tym, że pomiar został rozpoczęty 8 października 2009 roku o godzinie 21:45. Pozycja 0, 0, 0, świadczy o tym, że system nie wykonywał do tej pory ruchu w żadnym z kierunków.

Tryb nr 8 - zerowanie w obu płaszczyznach

System w tym trybie dokonuje obrotu w prawo do czasu wystąpienia narastającego zbocza na *czujniku nr 1*. Dokonywany jest także obrót w płaszczyźnie pionowej do czasu pojawienia się narastającego zbocza na *czujniku nr 3*.

Tryb nr 11 - zerowanie w płaszczyźnie pionowej

W trybie tym dokonywany jest jedynie obrót w płaszczyźnie pionowej do czasu pojawienia się narastającego zbocza na *czujniku nr 3*.

Tryby pomiarowe

Tryb nr 2 - pomiar dokładny

W trybie pomiaru dokładnego układ wykonuje obrót w prawo o ustaloną liczbę kroków. Po wykonaniu obrotu dokonywany jest pomiar odległości, wraz z pomiarem wektora przyspieszenia i natężenia pola magnetycznego. Wyniki wraz z pozycją i aktualnym czasem zapisywane są do pliku wynikowego. Gdy dalmierz wykona pełen obrót w płaszczyźnie poziomej, dokonywany jest obrót o wskazaną ilość kroków w górę. Pomiary trwają do czasu dojścia układu do pozycji końcowej, w której układ zakańcza plik wynikowy komentarzem *#KONIEC* i przechodzi w *tryb stop*.

Tryb nr 10 - pomiar w płaszczyźnie pionowej

Tryb ten wykorzystywany jest do zbierania informacji o odległości tylko w płaszczyźnie pionowej. Działa on podobnie do trybu nr 2 (dokładnego) z wyjątkiem tego, że nie jest wykonywany żaden ruch w płaszczyźnie poziomej.

Tryb nr 12 - wykonywanie pliku z programem pomiaru

W trybie tym wykonywane są pomiary w ściśle określonych punktach, zapisanych w pliku konfiguracyjnym na karcie SD. Plik posiada rozszerzenie *prog*. W pliku tym

każda linia zawiera informację o pozycji dalmierza, w jakiej powinien zostać wykonany pomiar. Pozycja dalmierza określona jest przez ilość kroków jaką system powinien wykonać od punktu zerowego. Ilość kroków w płaszczyźnie poziomej oddzielona jest znakiem średnika (;) od ilości kroków w płaszczyźnie pionowej. Układ dokonuje obrotu tylko w jednym kierunku. Do czasu, kiedy zaprogramowany punkt nie ma innej wartości kroku w płaszczyźnie pionowej niż aktualna, wykonywany jest obrót w prawo. Gdy układ znajdzie się w położeniu aktualnie wskazywanym przez plik z programem pomiaru dokonywany jest pomiar odległości wraz z pomiarem wektora przyspieszenia i natężenia pola magnetycznego, a wyniki zapisywane są do pliku. Następnie pobierany jest kolejna pozycja dalmierza, w której system powinien dokonać pomiaru odległości. Jeżeli zaprogramowana wartość kroków w płaszczyźnie pionowej jest różna od aktualnej wykonywany jest obrót w górę. *Tryb nr 12* jest zakończony wtedy, gdy wszystkie punkty z pliku zostaną odczytane lub jeżeli układ osiągnie pozycje końcową.

Tryby kalibracji

Tryb nr 4 - kalibracja silników

W trybie tym układ dokonuje pełnego obrotu w każdym z możliwych kierunków w celu ustalenia ilości kroków potrzebnych na obrót.

Tryb nr 5 - kalibracja kompasu

W trybie tym dokonuje się kalibracji kompasu elektronicznego.

Tryb nr 6 - kalibracja akcelerometru

W trybie tym dokonuje się kalibracji akcelerometru elektronicznego.

Tryby przeznaczone do komunikacji z użytkownikiem

Tryb nr 3 - sterowanie manualne

W trybie manualnym układ oczekuje na komendy użytkownika. Możliwe jest stero-

wanie silnikami krokowymi i dalmierzem laserowym. Tryb ten obowiązuje do czasu zmiany trybu wywołanej przez użytkownika.

Tryb nr 7 - transfer plików

W trybie transferu możliwe jest pobranie listy plików znajdujących się na karcie SD, ich transmisję, a także usunięcie. W trybie tym możliwe jest także ustawienie parametrów tworzących plik z programem pomiaru (wykorzystywany w trybie programowym). Tryb ten trwa do czasu wymuszenia zmiany trybu przez użytkownika.

Komendy dostępne w systemie

Komenda *TRYB*=<numer trybu>

Komenda *tryb* służy do zmiany aktualnego trybu pracy układu.

Komenda *TRYB?*

Po odebraniu komendy układ podaje numer trybu w jakim aktualnie się znajduje.

Komenda *HELP*

Po odebraniu komendy układ przesyła przez interfejs UART listę dostępnych komend i trybów pracy.

Komenda *T*=<wartość>

Komenda ustawia aktualny czas. Wartością powinna być liczba sekund jaka minęła od 1 stycznia 2000 roku.

Przykładowo komenda: *T=90120* Ustawi aktualną datę systemu na 2 styczeń 2000 roku z godziną 1:02.

Komenda *TIME?*

Komenda zwraca czas jaki jest ustawiony w urządzeniu.

Komendy *CCW*, *CW*, *UP*, *DOWN*, *CWUP*, *CWDOWN*, *CCWUP*, *CCWDOWN*

Komendy te obowiązują tylko w trybie sterowania ręcznego. Ich wywołanie skutkuje wykonaniem jednego kroku silnika w danym kierunku:

- CW (clockwise) - w prawo
- CCW (counter clockwise) - w lewo
- UP - w góre
- DOWN - w dół

Komenda *INFO*

Wyświetla pełną informację o aktualnym położeniu dalmierza laserowego, a także całego systemu w przestrzeni.

Komendy $H=<\text{wartość}>$, $V=<\text{wartość}>$

Komendy $H=$ i $V=$ obowiązują w trybie ręcznym. Powoduje ustawienie docelowego położenia układu. Wartości H - położenie w płaszczyźnie poziomej powinny być z przedziału $(-180.0;180.0>$. Wartości V - położenie w płaszczyźnie pionowej powinny być z przedziału $<0.0;90.0>$.

Komendy $H?$, $V?$

Komendy $H?$ oraz $V?$ wypisują przeliczone wartości kątów w jakich położony jest dalmierz laserowy. H - płaszczyzna pozioma. V - płaszczyzna pionowa.

Komenda *!*

Gdy odebrany zostanie znak *!* wszystkie trwające pętle programowe są przerywane i układ przechodzi natychmiast w tryb stopu. Komenda ta powinna być wykorzystywana tylko w przypadku awarii.

Komenda *KONCZ*

Komenda *KONCZ* służy zakończeniu trybów pomiarowych. Po odebraniu tej komendy układ przerywa pomiar i przechodzi do pozycji końcowej, w której dany tryb jest prawidłowo zakańczany.

Komenda *LIST*

Komenda *LIST* obowiązuje w trybie transferu. Wypisuje listę plików znajdujących się na karcie SD.

Komenda *FILE*=<numer pliku>

Komenda *FILE*= obowiązuje w trybie transferu. Powoduje przesłanie zawartości pliku o wskazanym numerze.

Komenda *USUN*=<numer pliku>

Komenda *USUN* obowiązuje w trybie transferu. Powoduje usunięcie pliku o wskazanym numerze. Należy pamiętać o odświeżeniu listy plików po każdym usunięciu, gdyż zmienia ono numerację plików.

Komenda *KOMPAS?*

Komenda *KOMPAS?* wykonuje pełną procedurę pomiaru wektora natężenia pola magnetycznego.

Komenda *AKCEL?*

Komenda *AKCEL?* wykonuje pełną procedurę pomiaru wektora przyspieszenia.

Komenda *I2C?*

Komenda *I2C?* powoduje wykonanie procesu skanowania magistrali *I²C* i wypisuje dostępne adresy urządzeń podłączone do tej magistrali.

Komenda *SAVE*

Komenda *SAVE* obowiązuje w trybie kalibracji kompasu i trybie kalibracji akcelerometru. Wywołanie komendy powoduje zapisanie wartości skrajnych kalibracji do pliku i przechodzi w tryb stop.

Komenda *STOP*

Komenda *STOP* zmienia aktualny tryb pracy na *tryb stop*.

Komenda *AUTO*=<numer trybu>

Komenda *AUTO*= powoduje zapamiętanie trybu automatycznego - tj. tryb jaki będzie ustawiany automatycznie po odpowiednio długim okresie w trybie stop.

Komenda *NASTEPNY*=<numer trybu>

Komenda *NASTEPNY*= powoduje zapamiętanie trybu jaki będzie ustawiany po skończeniu *trybu automatycznego*.

Komenda *REBOOT*

Komenda *REBOOT* Powoduje programowy restart wszystkich peryferiów układu ATSAM7S, a następnie ponowne uruchomienie całego układu.

Komenda *PARAM=<wartość parametru>*

Komenda *PARAM=* Obowiązuje w trybie transferu. Komenda ta powinna wykonana być 6 razy. Ustala ona kolejno wartości następujących parametrów wykorzystywanych do generacji pliku z programem pomiaru:

1. Ustawienie początkowe w płaszczyźnie poziomej,
2. Ustawienie końcowe w płaszczyźnie poziomej,
3. Krok w płaszczyźnie poziomej z jakim układ będzie poruszał się z pozycji początkowej do końcowej,
4. Ustawienie początkowe w płaszczyźnie pionowej,
5. Ustawienie końcowe w płaszczyźnie pionowej,
6. Krok w płaszczyźnie pionowej z jakim układ będzie poruszał się z pozycji początkowej do końcowej.

Po wprowadzeniu ostatniego parametru układ rozpoczyna generację pliku z programem pomiaru.

Komenda *DIST*

Komenda *DIST* obowiązuje w trybie ręcznym. Po jej odebraniu układ emuluje naciśnięcie przycisku *Dist* dalmierza laserowego Leica DISTO Lite5 (włącza wskaźnik laserowy lub rozpoczyna pomiar odległości).

Komenda *PWR*

Działa w trybie ręcznym. Emuluje naciśnięcie przycisku włączającego i wyłączającego dalmierz laserowy Leica DISTO Lite5.

Komenda $D?$

Komenda $D?$ obowiązuje w trybie ręcznym. Powoduje wykonanie pełnej procedury pomiaru odległości: ewentualne włączenie dalmierza, pomiar odległości i wypisanie wyniku pomiaru.

Komunikaty wysyłane przez system

System w trakcie swojej pracy przesyła komunikaty przez moduł bluetooth w celu kontroli stanu w jakim się znajduje. Część komunikatów zapisywana jest w pliku wynikowym tworząc historię pomiarów. Historia pomiarów może być wykorzystywana m.in do tworzenia mapy pomieszczenia.

Najważniejsze komunikaty występujące w plikach wynikowych to:

- Komunikat w formacie: $B; \langle \text{wartość } X \rangle; \langle \text{wartość } Y \rangle; \langle \text{wartość } Z \rangle; \langle \text{kąt obrotu} \rangle$

Komunikat określa miejsce w jakim znajduje się robot względem ostatniej pozycji w której wykonywany był pomiar. Współrzędne podawane są w mm . Kątem obrotu nazywany jest kąt o jaki została obrócona platforma w stosunku do poprzedniego miejsca. Wartość kąta podawana jest w stopniach.

- Komunikat w formacie: $Z; A; \langle \text{wartość } X \rangle; \langle \text{wartość } Y \rangle; \langle \text{wartość } Z \rangle; K; \langle \text{wartość } X \rangle; \langle \text{wartość } Y \rangle; \langle \text{wartość } Z \rangle$

Komunikat podaje wyniki pomiarów wektora przyspieszenia i natężenia pola magnetycznego w pozycji zerowej.

- Komunikat w formacie: $C; \langle \text{stan cz1} \rangle; \langle \text{stan cz2} \rangle; \langle \text{stan cz3} \rangle; \langle \text{stan cz4} \rangle; \langle \text{ilość kroków } H \rangle; \langle \text{ilość kroków } V \rangle$

Komunikat informuje o stanie czujników i ilość kroków jaka została wykonana w obu płaszczyznach do momentu zmiany stanów czujników. Rozróżniane są cztery stany czujników:

- 0 oznacza, że czujnik krańcowy jest zwarty
 - 1 oznacza, że czujnik krańcowy został rozwarty (zbocze narastające)
 - 2 oznacza, że czujnik krańcowy został zwarty (zbocze opadające)
 - 3 oznacza, że czujnik krańcowy jest rozwarty
- Komunikat w formacie: *D;<odległość>;<czas>*
Komunikat informuje o wyniku pomiaru odległości. Odległość podana jest w *mm*. Czas jest to czas jaki minął od rozpoczęcia pomiaru odległości, mierzony w ilości ramek przesłanych z dalmierza laserowego. Wartość poniżej 300 świadczy o błędzie, który wystąpił podczas pomiaru i stanowi numer błędu dalmierza laserowego.
 - Komunikat w formacie: *P;<nr katalogu>;<nr pliku>;<rozmiar pliku>;<data utworzenia>;<nazwa pliku DOS>;<nazwa pliku>*

Komunikat informuje o zawartości karty SD. Nazwa pliku DOS jest to nazwa pliku skrócona do 8 znaków. Jeżeli nazwa pliku jest dłuższa niż 8 znaków dodatkowo zwracana jest pełna nazwa pliku.

Plik wynikowy

W trakcie trwania pomiarów system zapisuje zebrane dane do pliku wynikowego w karcie pamięci. W pliku wynikowym znajdują się wszystkie dane mogące służyć w dalszej analizie pomiarów. Dane te powinny być w odpowiedni sposób interpretowane. Najprostszym przykładem umożliwiającym poznanie pliku wynikowego jest przykładowy pomiar czterech punktów w przestrzeni.

Linie przykładowego pliku wynikowego	Znaczenie komunikatu
#Przykład pliku wynikowego	Linia rozpoczynająca się znakiem # powinna traktowana być jako komentarz nie mający znaczenia na wyniki pomiarów
T;72	Zegar systemowy podaje informację, że minęły 72 sekundy od północy 1 stycznia 2000 roku
B;10;20;30;45.500000	Podstawa robota znajduje się w odległości 10mm w osi x, 20mm w osi y, 30mm w osi z i obrócona jest o 45.5° w płaszczyźnie poziomej względem poprzedniego miejsca pomiarowego
Z;A;-0.73;-0.03;-0.26;K;-0.30;0.11;0.59	W pozycji zerowej wskazanie akcelerometru wyniosło: -0.73 w osi x, -0.03 w osi y, -0.26 w osi z, a wskazanie kompasu wyniosło: -0.3 w osi x, 0.11 w osi y, 0.59 w osi z
T;121	00:02:01 1 stycznia 2000r.
S;134.568000;-5.250000;100;50	Silniki krokowe wykonały 100 kroków w płaszczyźnie poziomej i 50 w płaszczyźnie pionowej od miejsca zerowego. Orientacyjna wartość kątów obrotu dalmierza laserowego wynosi: 134.568° w płaszczyźnie poziomej i -5.25° w płaszczyźnie pionowej
A;-0.706840;0.000000;-0.228296	Wskazanie akcelerometru
K;0.310345;0.212272;0.564966	Wskazanie kompasu

Linie przykładowego pliku wynikowego (c.d.)	Znaczenie komunikatu
D;652;47 T;1000 S;134.568000;-19.250000;18690;50 A;-0.706840;0.000000;-0.228296 K;0.310345;0.112272;0.664966	Odległość do mierzonego punktu wynosi 652mm. Wynik pomiaru pojawił się w 47 ramce danych z dalmierza laserowego licząc od momentu rozpoczęcia pomiaru.
D;255;148 T;1200 C;2;3;3;3;25526;50	Wartość pomiaru jest mniejsza od 300 co oznacza, że wystąpił błąd pomiaru. Kod błędu dalmierza laserowego Leica: 255. Nastąpiło zwarcie czujnika 1. Czujniki 2, 3 i 4 są rozwarte. Silniki wykonały 25526 kroków w płaszczyźnie poziomej i 100 kroków w płaszczyźnie pionowej od pozycji zerowej.
T;1220 C;1;3;3;3;26661;50 T;1500 S;33.912000;-6.500000;4710;900 A;-0.858958;0.116561;-0.003215 K;-0.243148;0.317668;0.486395	Nastąpiło rozwarcie czujnika 1 co oznacza, że system wykonał pełny obrót w płaszczyźnie poziomej. Ilość kroków na pełny obrót w płaszczyźnie poziomej wyniosła 26661.
D;1908;79	

Linie przykładowego pliku wynikowego (c.d.)	Znaczenie komunikatu
T;1700 S;34.128000;-6.500000;7050;900 A;-0.865472;0.116561;0.003215 K;-0.072414;0.412446;0.569728 D;1918;56 T;1750 C;2;3;3;3;25527;900 T;1800 C;1;3;3;3;26663;900 T;1820 C;3;3;3;2;0;5944 #KONIEC	 Nastąpiło rozwarcie czujnika 1. Ilość kroków na pełny obrót w płaszczyźnie poziomej wyniosła 26663 Nastąpiło zwarcie czujnika 4 co oznacza, że system znalazł się w pozycji końcowej. Ilość kroków w płaszczyźnie pionowej, od pozycji zerowej do końcowej wyniosła: 5944.

4.2 Oprogramowanie PC

W ramach niniejszej pracy zostało napisane oprogramowanie komputerowe umożliwiające kontrolę i podgląd stanu urządzenia. Program napisano w języku C++ z użyciem edytora CodeBlocks. Interfejs programu wykorzystuje bibliotekę wxWidgets.

wxWidgets

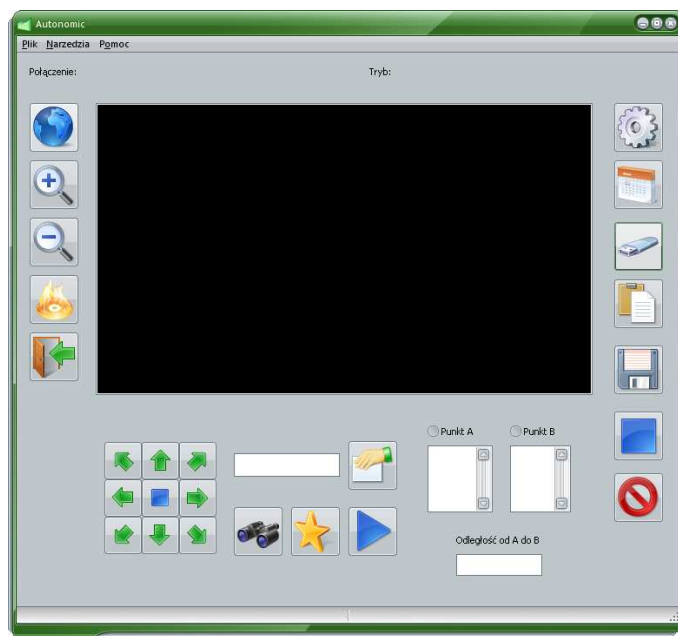
WxWidgets [18] jest projektem umożliwiającym tworzenie przenośnego oprogramowania na różne platformy systemowe. Prace nad projektem rozpoczęły się w 1992 roku na Uniwersytecie Edynburskim. Licencja typu open source umożliwia zastosowanie wxWidgets darmowo także w komercyjnym oprogramowaniu (w przeciwieństwie do bliźniaczego projektu: Qt). Aktualnie wxWidgets wspiera tworzenie oprogramowania w wielu językach programowania m.in.: C++, C#, Python, Perl, Java, Lua, Eiffel, Basic, Ruby dla systemów:

- Unix (GTK+, X11)
- Windows
- Mac OS X
- WinCE

Źródła programów korzystających z biblioteki wxWidgets kompilowane są do postaci binarnej docelowej platformy. Dzięki temu wygląd aplikacji jest typowy dla danej platformy systemowej.

Opis Programu

Napisany na potrzeby projektu program "Autonomic" pozwala na komunikację z robotem. Graficzny interfejs zwalnia użytkownika z wymogu poznania komend sterujących pracą systemu. Program pozwala także na przesyłanie wyników w trybie transferu. Na środku głównego okna programu znajduje się panel OpenGL, który



Rysunek 4.2: Główne okno programu Autonomic

można wykorzystać do wizualizacji pracy systemu. Dookoła panelu OpenGL znajdują się przyciski, dzięki którym możliwa jest interakcja z systemem. W trakcie pracy urządzenia wszystkie odebrane komunikaty są zapisywane w komputerze i mogą posłużyć do analizy w przypadku błędnego działania systemu. Głównym przeznaczeniem programu jest konwersja wyników pomiarów. Wyniki pomiarów mogą być konwertowane do formatów plików, obsługiwanych przez popularne programy statystyczne, naukowe i graficzne. Program Autonomic konwertuje wyniki pomiarów do następujących rodzajów plików:

- *Plik csv* - Jest to plik tekstowy dzięki któremu można dokonać eksportu zebranych danych do arkuszy kalkulacyjnych i baz danych. Pierwsza linijkę pliku

stanowią nagłówki kolumn oddzielone od siebie średnikami. W każdej następnej linii znajdują się informacje na temat zmierzonego punktu. Każda linia odpowiada jednemu punktowi pomiarowemu. Przykładowa struktura pliku:

T;D;N;H;V;Hmax;Vmax;Yaw;Pitch;X;Y;Z

1008;482;6;0;1100;26528;5930;-3,14159;0,0295794;-481,789;5,90002e-014;14,2552

1009;481;7;35;1100;26528;5930;-3,1333;0,0295794;-480,773;3,9856;14,2256

1010;484;6;70;1100;26528;5930;-3,12501;0,0295794;-483,722;8,02064;14,3143

1011;481;6;105;1100;26528;5930;-3,11672;0,0295794;-480,641;11,9557;14,2256

- *Plik xyz* - Jest to prosty plik tekstowy zapisujący dane chmur punktów. W każdej linii pliku znajdują się 3 współrzędne punktu (x, y i z) podane w mm, oddzielone od siebie spacją. W celu rozszerzenia możliwości analizy danych dodatkowo po parametrach x y i z umieszczone są 3 składowe koloru punktu: r g b. Kolor ten zależy od czasu w jakim dalmierz laserowy dokonał pomiaru. Przykładowa zawartość pliku xyz:

-298 0 12

-297 2 44

-296 4 32

-296 7 8

- Przykładowa zawartość pliku xyz zawierająca dodatkową informację o kolorze punktu:

-298 0 8 127 12 12

-297 2 8 227 12 12

-296 4 8 100 12 12

-2100 7 8 255 0 0

-2126 9 8 127 12 12

- *Plik obj* - dzięki któremu możliwy jest bezpośredni import pomiarów do biblioteki OpenGL, a także do większości trójwymiarowych programów graficznych. Jest to plik tekstowy. Eksport samych punktów pomiarowych wygląda bardzo podobnie do *pliku xyz* za wyjątkiem tego, że każda linia zaczyna się literą *v*. Format ten umożliwia łączenie punktów i tworzenie na ich podstawie brył przestrzennych.
- *Plik m* - umożliwiający szybki import wyników pomiarów w programie Matlab.
- *Obrazy graficzne w formacie png* - w celu łatwej oceny pomiarów przez użytkownika. Do każdego kompletnego pomiaru generowany jest szereg obrazów png:
 - widok panoramiczny, gdzie kolor odpowiada zmierzonej odległości do punktu, a pozycja piksela świadczy o ilości wykonanych kroków od punktu zerowego.
 - widok wzdłuż osi *z*, gdzie pozycja punktu na obrazie świadczy o jego zmierzonej pozycji *x* i *y*, a kolor punktu zmienia się wraz z pozycją *z* (widok z góry).
 - widok wzdłuż osi *x*, gdzie pozycja punktu na obrazie świadczy o jego zmierzonej pozycji *y* i *z*, a kolor punktu zmienia się wraz z pozycją *x* (widok z przodu).
 - widok wzdłuż osi *y*, gdzie pozycja punktu na obrazie świadczy o jego zmierzonej pozycji *x* i *z*, a kolor punktu zmienia się wraz z pozycją *y* (widok z boku).

Ostatnie trzy widoki tworzone są w dwóch wersjach:

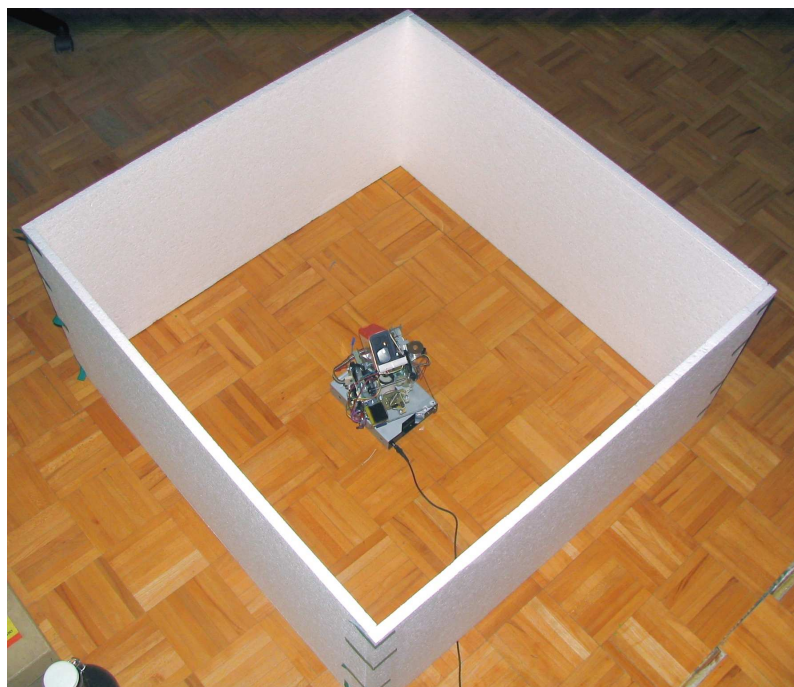
- Z punktami zaznaczonymi jako okrąg o promieniu 5 pikseli.
- Kolejnymi punktami połączonych z sobą cienką linią.

Program "Autonomic" dodatkowo umożliwia tworzenie programu pomiarów, a także manualne sterowanie systemem.

Rozdział 5

Przykładowe wyniki pomiarów

5.1 Pomiary przestrzeni referencyjnych

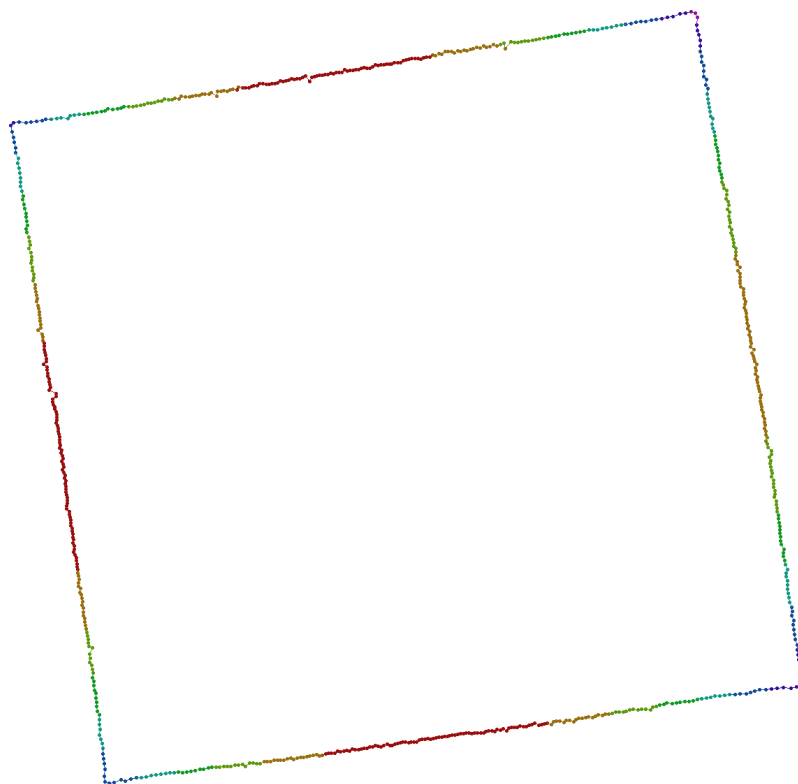


Rysunek 5.1: Konstrukcja o wymiarach wewnętrznych 100cm x 96cm x 50cm

Na potrzeby projektu stworzona została prosta, styropianowa konstrukcja. W trakcie trwania pomiarów górna część konstrukcji była zakrywana i tworzyła prostopadłościan o wymiarach 100 cm x 96 cm i wysokości 50cm. Wewnątrz konstrukcji wykonano dwa pomiary:

- pomiar w trybie nr 2 (dokładnym) z krokiem: 250 w płaszczyźnie poziomej, 200 w płaszczyźnie pionowej
- pomiar w trybie nr 12 (programowy). Program pomiaru wykonał jeden pełny obrót w płaszczyźnie poziomej z krokiem 35

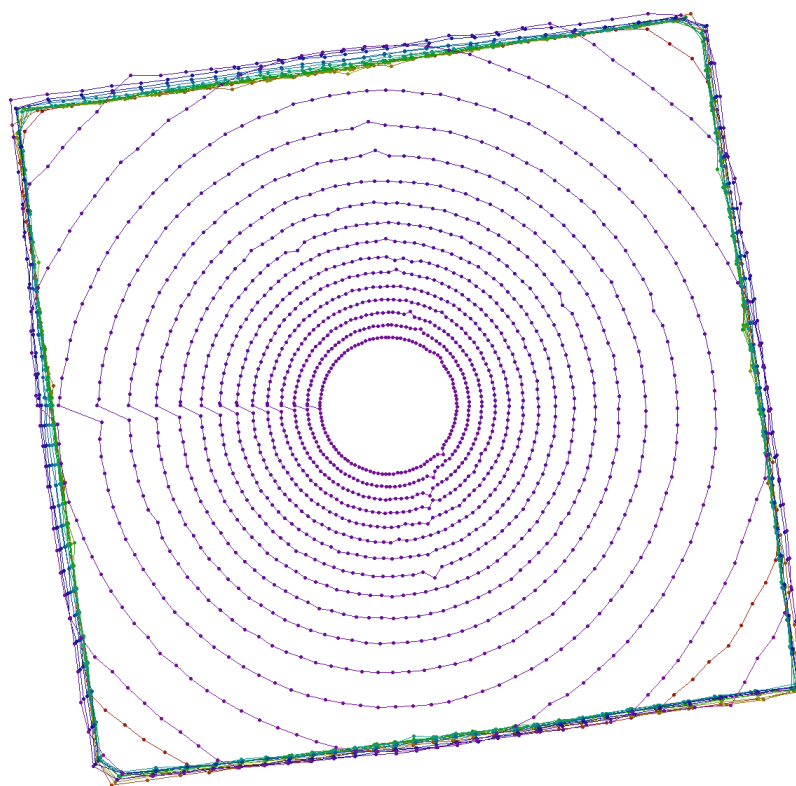
Pomiar w trybie 2 trwał 110 minut. Zostało zarejestrowane 3106 punktów. Pomiar w trybie 12 trwał 12 minut. Zostało zarejestrowane 759 punktów.



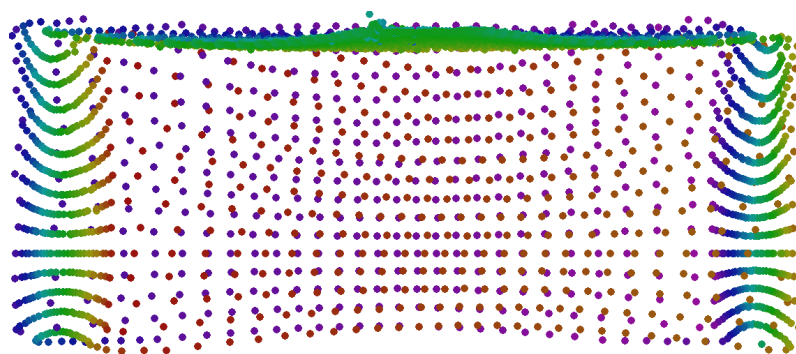
Rysunek 5.2: Automatycznie wygenerowany obraz png na podstawie pomiarów w trybie 12 - widok z góry



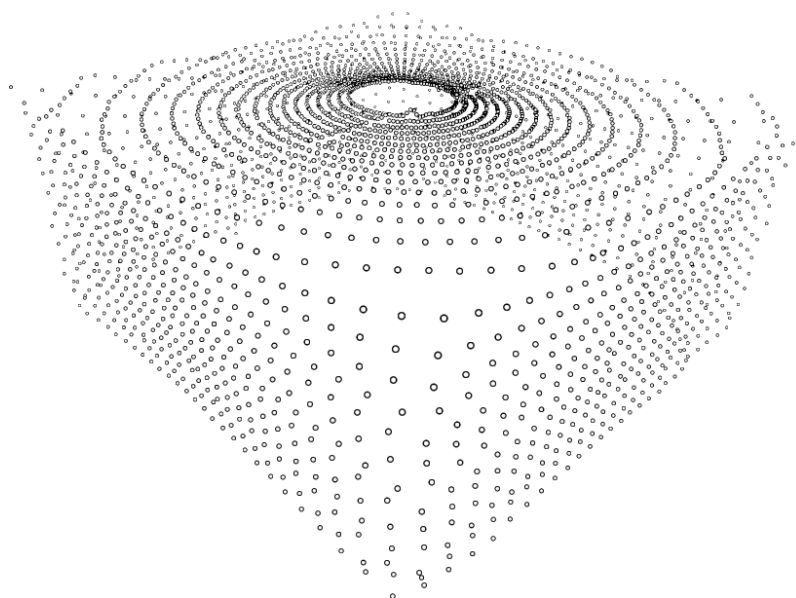
Rysunek 5.3: Automatycznie wygenerowany obraz png na podstawie pomiarów w trybie 2 - panorama odległości



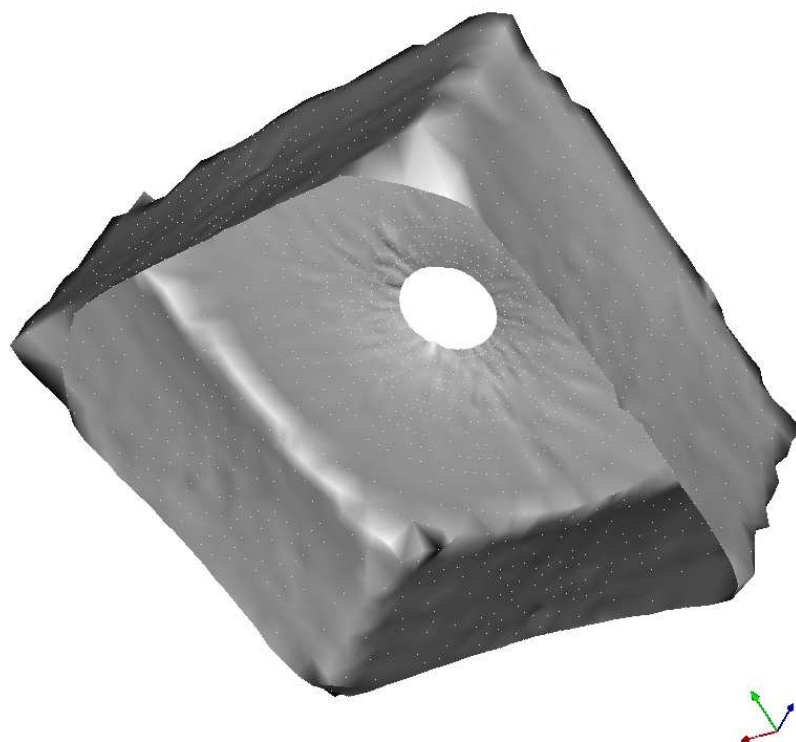
Rysunek 5.4: Automatycznie wygenerowany obraz png na podstawie pomiarów w trybie 2 - widok z góry



Rysunek 5.5: Automatycznie wygenerowany obraz png na podstawie pomiarów w trybie 2 - widok z boku



Rysunek 5.6: Wygenerowany plik obj oglądany w programie Meshlab



Rysunek 5.7: Chmura punktów z pliku xyz przekształcona w bryłę za pomocą programu VRMesh

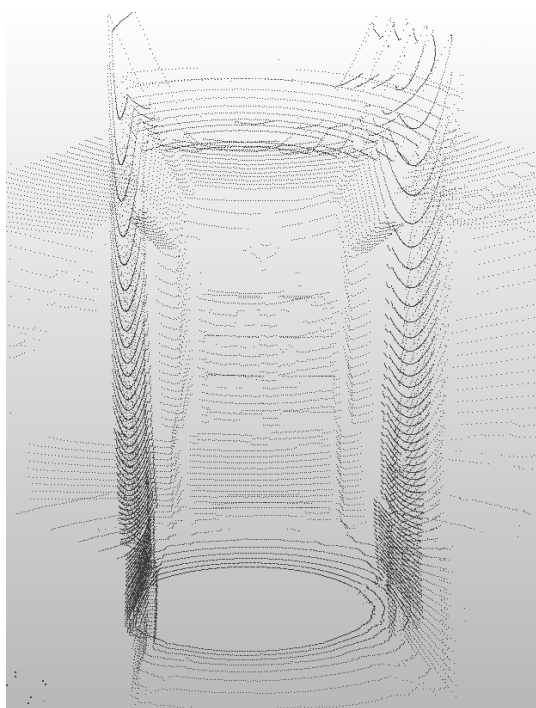
5.2 Pomiary pomieszczeń

Pomiary odbywały się w budynku mieszkalnym - piętrowym domu jednorodzinnym. Wszystkie pomiary odbywały się w trybie 2. Czas pomiarów zależny był od ustalonej ilości kroków silników co jaki miał pobierany być pomiar odległości. W pomieszczeniach tych w trakcie pomiarów przebywali ludzie. W wynikach pomiarów można zaobserwować pojedyncze punkty, które świadczą o ludzkiej obecności. Nie wpływa to znacząco na wyniki końcowe, które składają się z tysięcy punktów.

W pomieszczeniu łączącym trzy pokoje system zebrał pozycję 38603 punktów. Pomiar trwał 14 godzin.



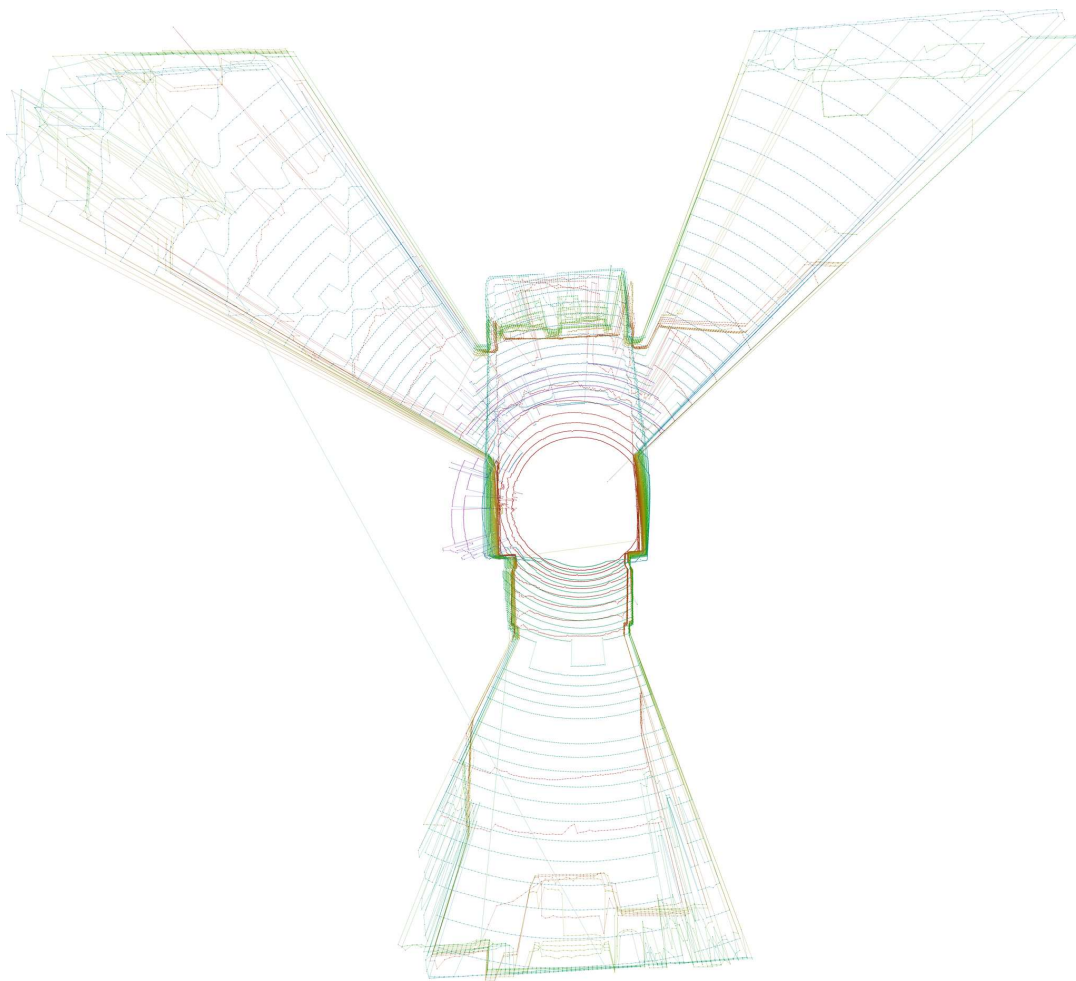
Rysunek 5.8: Pomieszczenie łączące 3 pokoje



Rysunek 5.9: Chmura punktów w programie Meshlab



Rysunek 5.10: Pomieszczenie łączące 3 pokoje - panorama odległości



Rysunek 5.11: Obraz png pomieszczenia łączącego 3 pokoje - widok z góry

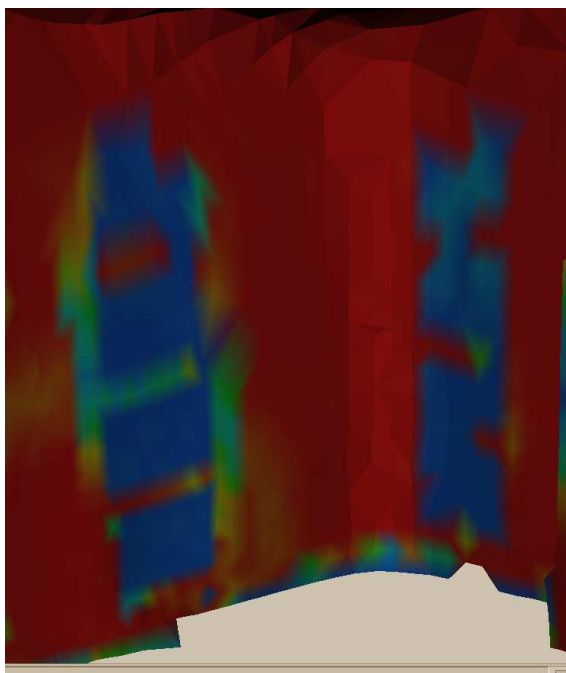
Kolejnym pomieszczeniem był hol, w którym wykonano pomiarów w dwóch miejscach.

- W pozycji 1 pomiar trwał 12 godzin, zarejestrowano 33672 punkty, a system dokonywał pomiaru odległości co 40 kroków w płaszczyźnie poziomej i 100 kroków w płaszczyźnie pionowej.
- W pozycji 2 pomiar trwał 6 godzin, zarejestrowano 16563 punkty, a system dokonywał pomiaru odległości co 40 kroków w płaszczyźnie poziomej i 200 kroków w płaszczyźnie pionowej.

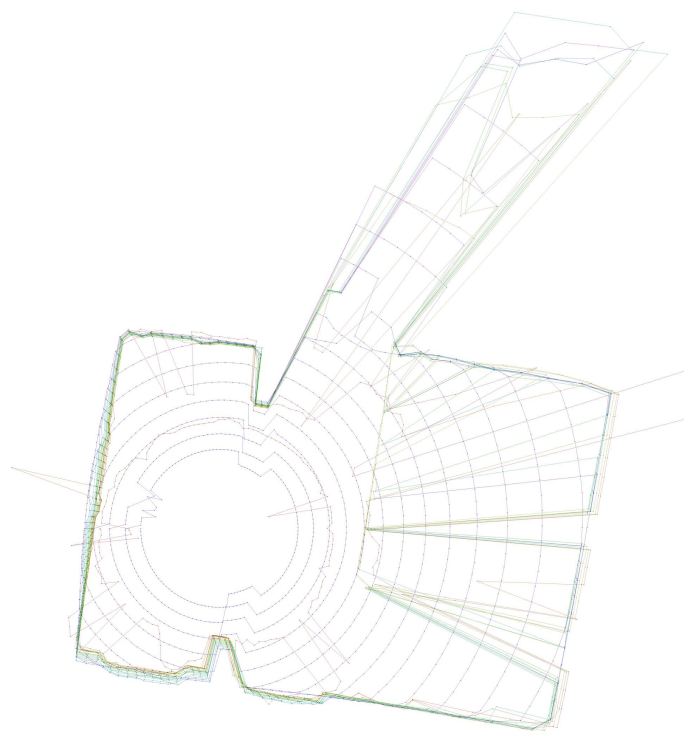
Dane uzyskane na podstawie badania czasu pomiaru odległości posłużyły do kolorowania chmury punktów. Przyspieszyło to analizę wyników przez użytkownika.



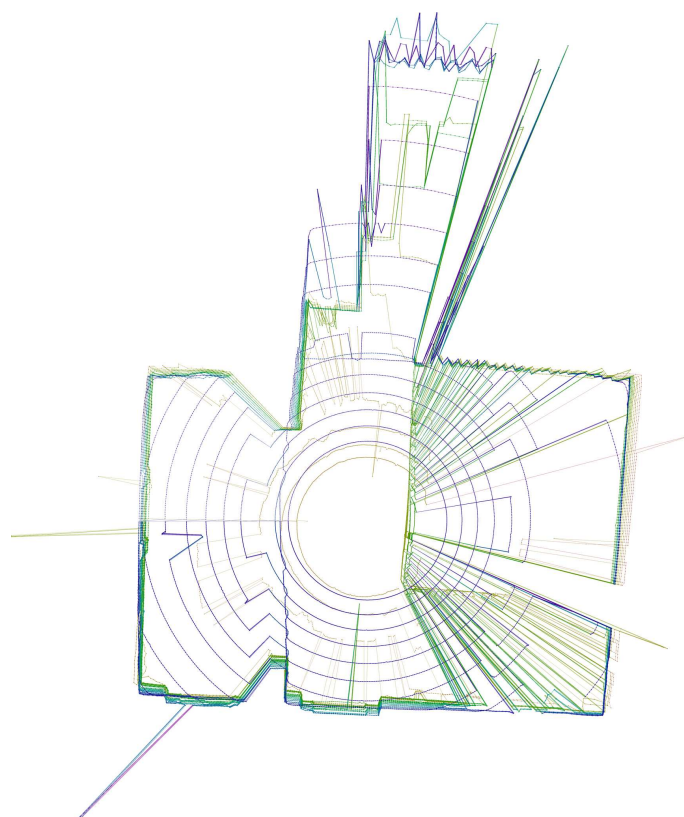
Rysunek 5.12: Fragment holu - drzwi



Rysunek 5.13: Kolorowa chmura punktów przekształcona w bryłę w programie VRMesh



Rysunek 5.14: Hol - pozycja 1 - widok z góry

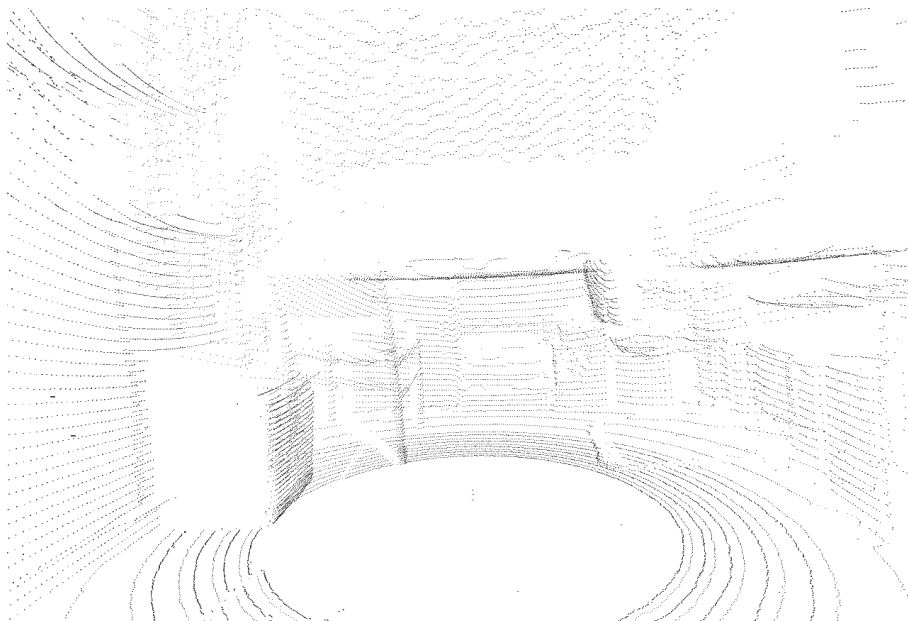


Rysunek 5.15: Hol - pozycja 2 - widok z góry

W pomieszczeniu biurowym pomiar trwał 24 godziny. Pomiar odległości dokonywany był co 40 kroków w płaszczyźnie poziomej i 100 kroków w płaszczyźnie pionowej. Zostało zarejestrowane 33689 punktów.



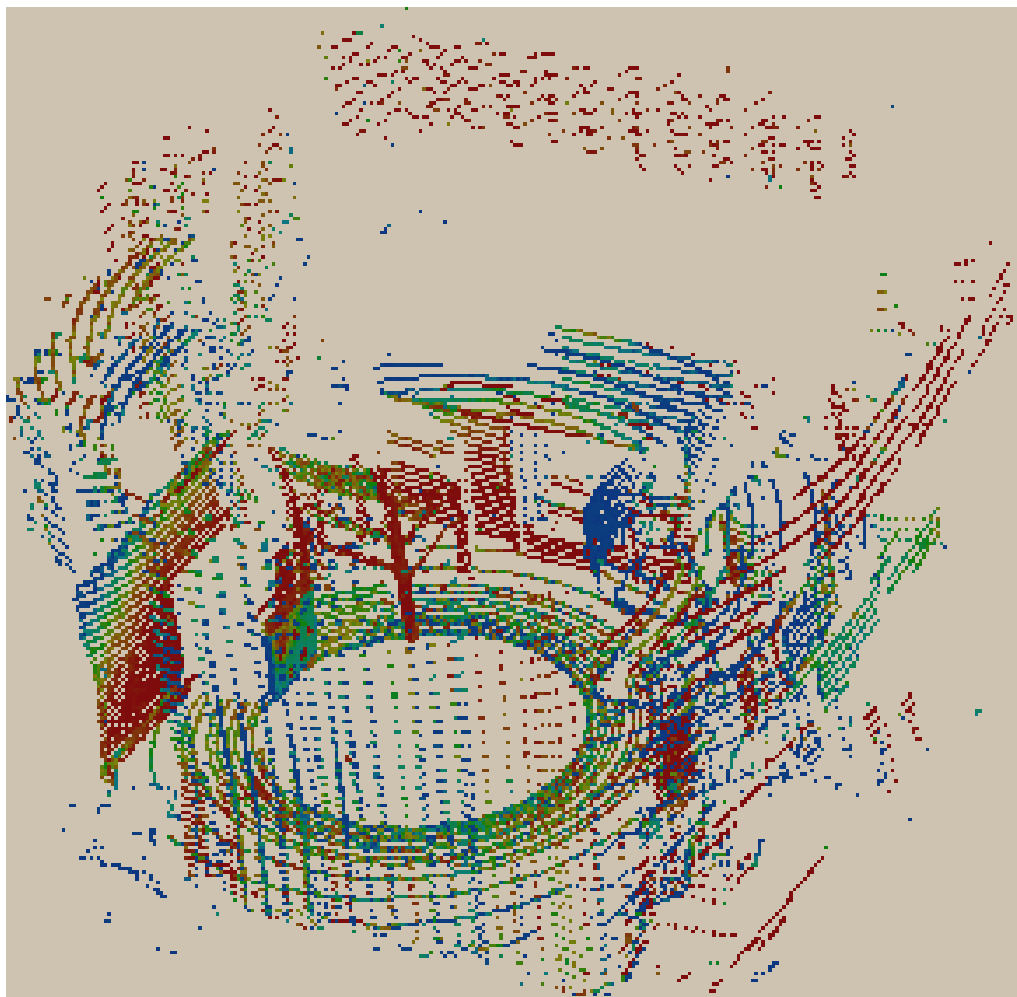
Rysunek 5.16: Pomieszczenie biurowe



Rysunek 5.17: Pomieszczenie biurowe - chmura punktów w programie Meshlab



Rysunek 5.18: Pomieszczenie biurowe - panorama odległości



Rysunek 5.19: Pomieszczenie biurowe - fragment kolorowej chmury punktów w programie VRMesh

Rozdział 6

Podsumowanie i wnioski końcowe

Wykonane na potrzeby pracy dyplomowej urządzenie spełnia wszystkie postawione założenia projektowe:

- Zaprojektowany system pozwala tworzyć elektroniczne mapy pomieszczeń.
- System do swojej pracy nie wymaga ludzkiej obsługi dzięki czemu zachowana jest pełna autonomiczność urządzenia.
- Bezprzewodowa komunikacja z urządzeniem umożliwia współpracę z użytkownikiem lub innymi urządzeniami.
- Stworzony na potrzeby projektu program komputerowy umożliwia komunikację z urządzeniem oraz konwersję wyników pomiarów.
- Konwersja map do popularnych formatów stosowanych w specjalistycznym oprogramowaniu daje szeroki wachlarz możliwości zastosowania układu.

Układ doskonale nadaje się do pomiarów pomieszczeń. Stworzone elektroniczne mapy mogą posłużyć innym urządzeniom do poruszania się i odnajdywania w przestrzeni. Mapy te mogłyby także być wykorzystywane przy inwentaryzacji, jak i wizualizacji nie tylko pomieszczeń ale i całych budynków.

Największą trudnością w codziennym wykorzystywaniu układu jest długi czas potrzebny do pomiaru pomieszczenia. Czas pomiaru jak i inne aspekty pracy układu

można łatwo zmienić wymieniając odpowiednie moduły na takie, które posiadają lepsze parametry. Przeszkodą do ich wprowadzenia w pierwszej wersji układu był zbyt duży koszt, gdy niepewny był efekt końcowy.

Bibliografia

- [1] A bird's eye view - of a leopard's spots. The Çatalhöyük 'map' and the development of cartographic representation in prehistory, The British Institute of Archaeology at Ankara 2006
- [2] Principles of airborne laser scanning, Karl Kraus, Technical University of Vienna, 2002
- [3] Skanowanie laserowe jako źródło informacji przestrzennych dotyczących lasów, Krzysztof Będkowski i Sławomir Mikrut, AGH 2007
- [4] A Perception-Driven Autonomous Urban Vehicle, Journal of Field Robotics vol. 25
- [5] The MIT - Cornell Collision and Why it Happened, Journal of Field Robotics vol. 25
- [6] DARPA Grand Challenge Winner: Stanley the Robot!, Steve Russell, Popular Mechanics 2006
- [7] 3D Mapping of Structured Environments, Jan Weingarten, ETH Zurich
- [8] Robotyzacja dyskretnych procesów produkcyjnych, Ryszard Zdanowicz, WPŚ 2009
- [9] GP2Y0A21YK0F - Distance Measuring Sensor Unit, SHARP 2007

- [10] Strona internetowa: DARPA Grand Challenge - <http://www.darpa.mil/grandchallenge/index.asp> (24.10.2009)
- [11] Laserowy pomiar wysokości lotu dla bezzałogowych obiektów latających, Cezary Kwiecień, Politechnika Śląska
- [12] Projektowanie systemów wbudowanych na przykładzie rodziny SAM7S z rdzeniem ARM7TDMI, Jacek Augustyn, IGSMiE PAN 2007
- [13] Low Cost $\pm 1.5g$ Tri Axis Accelerometer with Ratiometric Outputs, MEMSIC 2009.
- [14] Tri-axis Magnetic Sensor, With I^2C Interface, MEMSIC 2009.
- [15] Strona internetowa firmy Mercotac - <http://www.mercotac.com/html/products.html> (24.10.2009)
- [16] Strona internetowa firmy Transfer Multisort Elektronik - <http://www.tme.eu/lacznik-obrotowy-do-rj9-4p4c/arts/pl/b11/tel-0084.html> (24.10.2009)
- [17] Strona internetowa - <http://www.directindustry.com/prod/air-precision/slip-ring-sub-assembly-36953-253634.html> (24.10.2009)
- [18] Strona internetowa: wxWidgets - <http://www.wxwidgets.org/about/> (24.10.2009)
- [19] Strona internetowa: List of 3D API specifications and data formats - <http://local.wasp.uwa.edu.au/~pbourke/dataformats/> (6.11.2009)