

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: AUTOMATYKA I ROBOTYKA
SPECJALNOŚĆ: ROBOTYKA

PRACA DYPLOMOWA
INŻYNIERSKA

Projekt robota usługowego do zastosowań
domowych

Project of a service robot for home purposes

AUTOR:

Albert Lis

PROWADZĄCY PRACĘ:

dr inż. Mateusz Cholewiński,

Wydział Elektroniki, Katedra Cybernetyki i Robotyki

OCENA PRACY:

Spis treści

| | | |
|--------|--|----|
| 1. | Wstęp | 5 |
| 1.1. | Wprowadzenie | 5 |
| 1.2. | Cel i zakres pracy | 5 |
| 2. | Wstęp teoretyczny | 6 |
| 3. | Praca z szablonem | 7 |
| 3.1. | Organizacja szablonu | 7 |
| 3.2. | Kompilacja szablonu | 8 |
| 4. | Zalecenia dotyczące formatowania | 10 |
| 4.1. | Rozmiar i układ treści na stronach dokumentu | 10 |
| 4.2. | Strona tytułowa | 12 |
| 4.3. | Krój i wielkość czcionek | 13 |
| 4.4. | Formatowanie bloków tekstu | 14 |
| 4.5. | Opisy tabel i rysunków | 16 |
| 4.6. | Przypisy dolne | 17 |
| 4.7. | Formatowanie spisu treści | 17 |
| 4.8. | Formatowanie list wyliczeniowych i wypunktowań | 17 |
| 4.9. | Wzory matematyczne | 19 |
| 5. | Redakcja pracy | 20 |
| 5.1. | Układ pracy | 20 |
| 5.2. | Styl | 21 |
| 6. | Uwagi techniczne | 22 |
| 6.1. | Rysunki | 22 |
| 6.2. | Wstawianie kodu źródłowego | 24 |
| 6.3. | Wykaz literatury oraz cytowania | 25 |
| 6.4. | Indeks rzeczowy | 26 |
| 6.5. | Inne uwagi | 27 |
| 7. | Podsumowanie | 29 |
| 7.1. | Sekcja poziomu 1 | 29 |
| 7.1.1. | Sekcja poziomu 2 | 29 |
| 7.2. | Sekcja poziomu 1 | 29 |
| 8. | Konstrukcja | 30 |
| 8.1. | Wybór mikrokontrolera | 30 |
| 9. | Wstęp teoretyczny | 32 |

| | |
|---|----|
| Literatura | 33 |
| A. Tytuł dodatku | 34 |
| B. Opis załączonej płyty CD/DVD | 35 |

Skróty

IoT (ang. Internet of Things)

DMA (ang. Direct Memory Acces)

ADC (ang. Analog-to-digital converter)

PLL (ang. Phase-locked loop)

API (ang. Application programming interface)

HAL (ang. Hardware Abstraction Layer)

SPL (ang. Standard Peripheral Libraries)

(ang.)

(ang.)

Rozdział 1

Wstęp

1.1. Wprowadzenie

Robotyka jest obecnie prężnie rozwijającą się dziedziną nauki. Niskie ceny mikrokontrolerów oraz duża konkurencyjność firm na rynku powodują przenikanie urządzeń robotycznych z zastosowań specjalnych do życia codziennego. Urządzenia te, mogą oszczędzać zasoby ludzkie w codziennych prostych czynnościach. Dodatkowo projekty takie jak Arduino [1] pozwalają na tworzenie tych urządzeń bez specjalistycznej wiedzy. Kolejnym czynnikiem dynamizującym popularyzację automatyzacji i robotyki jest stopniowe wprowadzanie sieci 5G [3]. Pozwoli ona na wykorzystanie potencjału IoT oraz znaczną automatyzację działania urządzeń robotycznych. W tej pracy skupiono się na budowie urządzenia wspomagającego prace sprzątające.

Na rynku istnieje wiele konstrukcji robotów sprzątających. Skupiają się one głównie na pracy jako odkurzacze. Natomiast liczba autonomicznych robotów myjących podłogi jest zdecydowanie mniejsza. Głównie są to proste roboty mopujące. Robot przedstawiony w pracy ma za zadanie wypełnić lukę między małymi i prostymi urządzeniami, a dużymi do zastosowań profesjonalnych.

1.2. Cel i zakres pracy

Celem jest zaprojektowanie, zbudowanie i zaprogramowanie autonomicznego robota myjącego podłogi do zastosowań niekomercyjnych.

Rozdział pierwszy opisuje wstęp teoretyczny

Rozdział 2

Wstęp teoretyczny

Robot mobilny - robot, który potrafi zmieniać swoje położenie w przestrzeni. Może być robotem autonomicznym, czyli takim który realizując swoje zadanie porusza się bezkolizyjnie w wyznaczonym środowisku oraz robi to bez ingerencji operatora. Roboty mobilne można podzielić na kategorie przedstawione w tabeli 9.1.

Tab. 2.1: Kategorie robotów mobilnych

| Narzędzie | Wersja | Opis | Adres |
|--------------|--------|---|---|
| MiKTeX | 2.9 | Zalecana jest instalacja Basic MiKTeX z dystrybucji 32 lub 64 bitowej. Brakujące pakiety będą się doinstalowywać podczas kompilacji projektu. | http://miktex.org/download |
| TexnicCenter | 2.02 | Można pobrać 32 lub 64 bitową wersję | http://www.texniccenter.org/download/ |
| SumatraPDF | 3.1.1 | Można pobrać 32 lub 64 bitową wersję | http://www.sumatrapdfreader.org/download-free-pdf-viewer.html |
| JabRef | 3.3 | Można pobrać 32 lub 64 bitową wersję | http://www.fosshub.com/JabRef.html |

Rozdział 3

Praca z szablonem

3.1. Organizacja szablonu

Szablon składa się z pliku głównego, plików z kodem kolejnych rozdziałów i dodatków (włączanych do kompilacji w dokumencie głównym), katalogów z plikami grafik (włączanymi do rysunków w rozdziałach), pliku ze skrótami (opcjonalny), pliku z danymi bibliograficznymi (plik `dokumentacja.bib`). Taki „układ” zapewnia porządek oraz pozwala na selektywną kompilację rozdziałów. Wyjaśniając to dokładniej, podczas tworzenia szablonu przyjęto następującą konwencję:

- Plikiem głównym jest plik `Dyplom.tex`. To w nim znajdują się deklaracje wszystkich używanych stylów, definicje makr oraz ustawień, jak również polecenie `\begin{document}`.
- Teksty rozdziałów są redagowane w osobnych plikach o nazwach zawierających numer rozdziału. Pliki te zamieszczone są w katalogu głównym (tym samym, co plik `Dyplom.tex`). I tak `rozdziel01.tex` to plik pierwszego rozdziału (ze Wstępem), `rozdziel02.tex` to plik z treścią drugiego rozdziału itd.
- Teksty dodatków są redagowane w osobnych plikach o nazwach zawierających literę dodatku. Pliki te, podobnie do plików z tekstem rozdziałów, zamieszczane są w katalogu głównym. I tak `dodatekA.tex` oraz `dodatekB.tex` to, odpowiednio, pliki z treścią dodatku A oraz dodatku B.
- Pewnym wyjątkiem od reguły nazewnictwa w przypadku plików z tekstem rozdziałów i dodatków jest plik `skroty.tex`. Jest to plik, w którym zamieszczono wykaz użytych skrótów. W jego nazwie nie występuje żaden numer czy porządkowa litera.
- Każdemu rozdziałowi i dodatkowi towarzyszy katalog przeznaczony do składowania dołączanych w nim grafik. I tak `rys01` to katalog na pliki z grafikami dołączanymi do rozdziału pierwszego, `rys02` to katalog na pliki z grafikami dołączanymi do rozdziału drugiego itd. Podobnie `rysA` to katalog na pliki z grafikami dołączanymi w dodatku A itd.
- W katalogu głównym zamieszczany jest plik `dokumentacja.bib` zawierający bazę danych bibliograficznych.

Szablon przygotowano w systemie Windows stosując kodowanie `cp1250`. Można go wykorzystać również w innych systemach i przy innych kodowaniach. Jednakże wtedy konieczna jest korekta dokumentu `Dyplom.tex` odpowiednio do wybranego przypadku. Korekta ta polegać może na zamianie polecenia `\usepackage[cp1250]{inputenc}` na polecenie `\usepackage[utf8]{inputenc}` oraz konwersji kodowania istniejących plików ze źródłem latexowego kodu (plików o rozszerzeniu `*.tex` oraz `*.bib`).

Samo kodowanie plików może być źródłem paru problemów. Chodzi o to, że użytkownicy pracujący z edytorami tekstów pod linuxem mogą generować pliki zakodowane w UTF-8 bez BOM lub z BOM, a pod windowsem – pliki z kodowaniem znaków `cp1250` zakodowanych

Tab. 3.1: Pliki źródłowe szablonu oraz wyniki kompilacji

| Źródła | Wyniki kompilacji |
|--|--------------------------------|
| Dokument.tex - dokument główny | Dyplom.bbl |
| Dokument.tcp – szablon projektu MiKTeX | Dyplom.blg |
| rozdzial01.tex – plik rozdziału 01 | Dyplom.ind |
| ... | Dyplom.idx |
| dodatekA.tex – plik dodatku A | Dyplom.lof |
| ... | Dyplom.log |
| rys01 – katalog na rysunki do rozdziału 01 | Dyplom.lot |
| - fig01.png – plik grafiki | Dyplom.out |
| - ... | Dyplom.pdf – dokument wynikowy |
| ... | Dyplom.synctex |
| rysA – katalog na rysunki do dodatku A | Dyplom.toc |
| - fig01.png – plik grafiki | Dyplom.tps |
| - ... | *.aux |
| ... | Dyplom.synctex |
| dokumentacja.bib – plik danych bibliograficznych | |
| Dyplom.ist – plik ze stylem indeksu | |
| by-nc-sa.png – plik z ikonami CC | |

w ANSI. A z takimi plikami różne edytory różnie sobie radzą (w szczególności edytor TeXnicCenter czasami z niewiadomego powodu traktuje zawartość pliku jako UTF8 lub ANSI – chyba sprawdza, czy w bufore nie ma jakichś znaków specjalnych i na tej podstawie interpretuje kodowanie). Bywa, że choć wszystko wygląda OK to jednak kompilacja latexowa „nie idzie”. Problemem mogą być właśnie pierwsze bajty, których nie widać w edytorze.

Kodowanie znaków jest istotne również przy edytowaniu bazy danych bibliograficznych (pliku dokumentacja.bib). Aby bibtex poprawnie interpretował polskie znaki plik dokumentacja.bib powinien być zakodowany w ANSI, CR+LF (dla ustawień jak w szablonie). Do konwersji kodowania można użyć Notepad++ (jest tam opcja „konwertuj” - nie mylić z opcją „koduji”, która przekodowuje znaki, jednak nie zmienia sposobu kodowania pliku).

3.2. Kompilacja szablonu

Kompilację szablonu może uruchomić na kilka różnych sposobów. Wszystko zależy od używanego systemu operacyjnego, zainstalowanej na nim dystrybucji latexa oraz dostępnych narzędzi. Zazwyczaj kompilację rozpoczyna się wydając polecenie z linii komend lub uruchamia się ją za pomocą narzędzi zintegrowanych środowisk.

Kompilacja z linii komend polega na uruchomieniu w katalogu, w którym rozpakowano źródła szablonu, następującego polecenia:

```
> pdflatex Dyplom.tex
```

gdzie pdflatex to nazwa kompilatora, zaś Dyplom.tex to nazwa głównego pliku redagowanej pracy. W przypadku korzystania ze środowiska TeXnicCenter należy otworzyć dostarczony w szablonie plik projektu Dyplom.tcp, a następnie uruchomić kompilację narzędziami dostępnymi w pasku narzędziowym.

Aby poprawnie wygenerowały się wszystkie referencje (spis treści, odwołania do tabel, rysunków, pozycji literaturowych, równań itd.) kompilację pdflatex należy wykonać dwukrotnie, a czasem nawet trzykrotnie, gdy wygenerowane mają zostać odwołania do pozycji literaturowych oraz wykazu literatury.

Wygenerowanie danych bibliograficznych zapewnia kompilacja `bibtex` uruchamiana po kompilacji `pdflatex`. Można to zrobić z linii komend:

```
> bibtex Dyplom
```

lub wybierając odpowiednią pozycję z paska narzędziowego wykorzystywanego środowiska. Po kompilacji `bibtex` na dysku pojawi się plik `Dyplom.bbl`. Dopiero po kolejnych dwóch kompilacjach `pdflatex` dane z tego pliku pojawią się w wygenerowanym dokumencie. Podsumowując, po każdym wstawieniu nowego cytowania w kodzie dokumentu uzyskanie poprawnego formatowania dokumentu wynikowego wymaga powtórzenia następującej sekwencji kroków kompilacji:

```
> pdflatex Document.tex
> bibtex Document
> latex Document.tex
> latex Document.tex
```

Szczegóły dotyczące przygotowania danych bibliograficznych oraz zastosowania cytowań przedstawiono w podrozdziale 6.3.

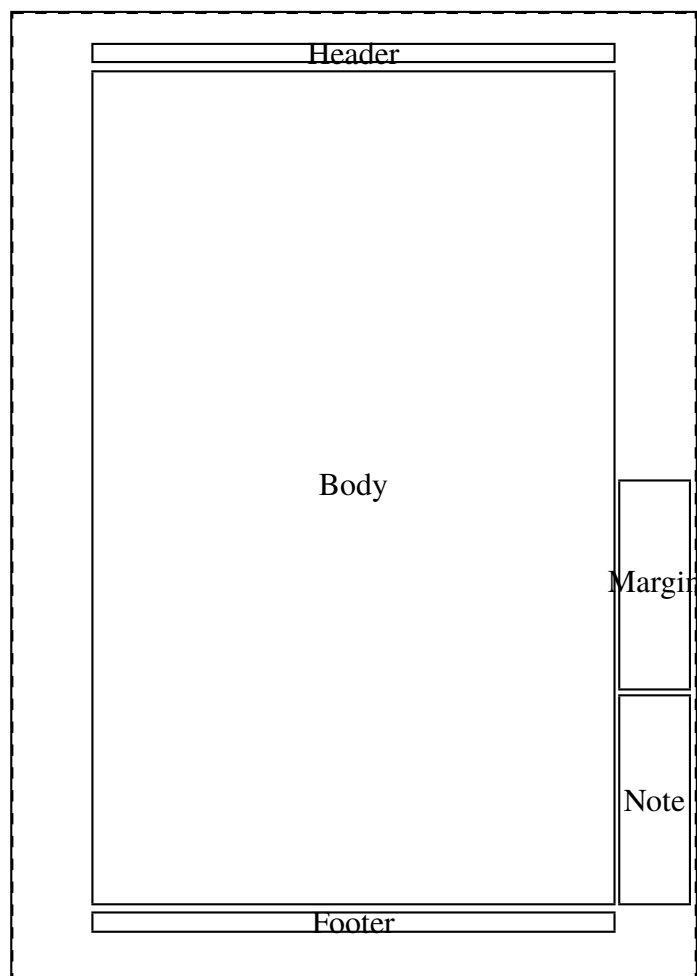
W głównym pliku zamieszczono polecenia pozwalające sterować procesem kompilacji poprzez włączanie bądź wyłączanie kodu źródłowego poszczególnych rozdziałów. Włączanie kodu do kompilacji zapewniają instrukcje `\include` oraz `\includeonly`. Pierwsza z nich pozwala włączyć do kompilacji kod wskazanego pliku (np. kodu źródłowego pierwszego rozdziału `\include{rozdzial01.tex}`). Druga, jeśli zostanie zastosowana, pozwala określić, które z plików zostaną skompilowane w całości (na przykład kod źródłowy pierwszego i drugiego rozdziału `\includeonly{rozdzial01.tex,rozdzial02.tex}`). Brak nazwy pliku na liście w poleceniu `\includeonly` przy jednoczesnym wystąpieniu jego nazwy w poleceniu `\include` oznacza, że w kompilacji zostaną uwzględnione referencje wygenerowane dla tego pliku wcześniej, sam zaś kod źródłowy pliku nie będzie kompilowany.

W szablonie wykorzystano klasę dokumentu `memoir` oraz wybrane pakiety. Podczas kompilacji szablonu w `MikTeXu` wszelkie potrzebne pakiety zostaną zainstalowane automatycznie (jeśli `MikTeX` zainstalowano z opcją dynamicznej instalacji brakujących pakietów). W przypadku innych dystrybucji `latexowych` może okazać się, że pakiety te trzeba doinstalować ręcznie (np. pod `linuxem` z `TeXLive` trzeba doinstalować dodatkową zbiorczą paczkę, a jeśli ma się menadżera pakietów `latexowych` - to pakiety `latexowe` można instalować indywidualnie).

Jeśli w szablonie będzie wykorzystany indeks rzeczowy, kompilację źródeł trzeba będzie rozszerzyć o kroki potrzebne na wygenerowanie plików pośrednich `Dokument.idx` oraz `Dokument.ind` oraz dołączenia ich do finalnego dokumentu (podobnie jak to ma miejsce przy generowaniu wykazu literatury). Szczegóły dotyczące generowania indeksu rzeczowego opisano w podrozdziale 6.4.

Rys. 4.1: Układ strony nieparzystej dla dokumentu klasy memoir

Dashed lines represent the actual page size after trimming the stock.



Lengths are to the nearest pt.

| | |
|-------------------------------------|----------------------------------|
| <code>\stockheight = 845pt</code> | <code>\stockwidth = 598pt</code> |
| <code>\pageheight = 845pt</code> | <code>\pagewidth = 598pt</code> |
| <code>\textheight = 727pt</code> | <code>\textwidth = 455pt</code> |
| <code>\trimtop = 0pt</code> | <code>\trimedged = 0pt</code> |
| <code>\uppermargin = 52pt</code> | <code>\spinemargin = 71pt</code> |
| <code>\headheight = 14pt</code> | <code>\headsep = 10pt</code> |
| <code>\footskip = 24pt</code> | <code>\marginparsep = 6pt</code> |
| <code>\marginparpush = 7pt</code> | <code>\columnsep = 10pt</code> |
| <code>\columnseprule = 0.0pt</code> | |

Rys. 4.2: Rzeczywisty układ strony nieparzystej w tym dokumencie

Rzeczywisty układ strony zastosowany w niniejszym dokumencie przedstawiono na rysunku 4.2. Lewy i prawy margines są takie same, więc strony parzyste i nieparzyste wyglądają podobnie, z dokładnością do umiejscowienia notatek marginesowych. Taki rezultat zapewniło zastosowanie poniższych komend.

```
\setlength{\headsep}{10pt}
\setlength{\headheight}{13.6pt}
\setlength{\footskip}{\headsep+\headheight}
\setlength{\uppermargin}{\headheight+\headsep+1cm}
\setlength{\textheight}{\paperheight-\uppermargin-\footskip-1.5cm}
\setlength{\textwidth}{\paperwidth-5cm}
\setlength{\spinemargin}{2.5cm}
```

```

\setlength{\foremargin}{2.5cm}
\setlength{\marginparsep}{2mm}
\setlength{\marginparwidth}{2.3mm}
\checkandfixthelayout[fixed]
\linespread{1}
\setlength{\parindent}{14.5pt}

```

4.2. Strona tytułowa

Według ogólnouczelnianych zaleceń (tj. logotypu Politechniki Wrocławskiej) strona tytułowa powinna być zredagowana z użyciem czcionki *garamond*. W oficjalnym wzorcu (patrz rysunek 4.3) nie rozróżniono, czy dotyczy on pracy inżynierskiej czy magisterskiej. Nie uwzględniono również miejsca na nazwę specjalności ani kierunku oraz zapomniano o nazwisku promotora, jednostce, dacie i ocenie. Za to określono (zgrubnie) położenie słów kluczowych i streszczenia. Ponieważ brakujące dane pojawiały się we wzorcach stron tytułowych stosowanych w codziennej praktyce na Wydziałach, nie wiadomo do końca, czy oficjalny szablon należy stosować w 100 procentach. Dlatego w niniejszym dokumencie zastosowano własny wzorzec strony tytułowej (używany od lat) oraz podano wymagania odnośnie wzorca z logotypu uczelnianego.

Rys. 4.3: Oficjalny szablon strony tytułowej pracy dyplomowej, <http://www.logotyp.pwr.edu.pl/Default.aspx?page=PracaDyplomowa> [dostęp dnia 20.04.2016]

Wymagania co do wielkości znaków na stronie tytułowej są następujące:

- według uczelnianego logotypu

Nazwa jednostki organizacyjnej: Garamond 16 pt
 Napis "PRACA DYPLOMOWA INŻYNIERSKA": Garamond 32 pt
 Tytuł pracy: Garamond 16 pt
 Autor: Garamond 14 pt
 Słowa kluczowe: Garamond 12 pt
 Krótkie streszczenie: Garamond 10 pt

- według wzorca użytego w niniejszym dokumencie

POLITECHNIKA WROCŁAWSKA (Garamond 22pt 24pt)
 WYDZIAŁ ELEKTRONIKI (Garamond 22pt 24pt)
 KIERUNEK: JAKIŚ KIERUNEK (Garamond 14pt 16pt)
 SPECJALNOŚĆ: JAKAŚ SPECJALNOŚĆ (Garamond 14pt 16pt)
 PRACA DYPLOMOWA (Garamond 24pt 26pt)
 INŻYNIERSKA (Garamond 24pt 26pt)
 Tytuł pracy w języku polskim (Garamond 16pt 18pt)
 Title in English (Garamond 16pt 18pt)
 AUTOR: (Garamond 16pt 18pt)
 Imię Nazwisko (Garamond 14pt 16pt)
 PROWADZĄCY PRACĘ: (Garamond 16pt 18pt)
 tytuł, Imię Nazwisko, Jednostka (Garamond 14pt 16pt)
 OCENA PRACY: (Garamond 16pt 18pt)
 WROCŁAW, 2015 (Garamond 16pt 18pt)

W szablonie zastosowano pakiet `ebgaramond`. Dostarcza on klonu czcionki `garamond`, jednak bez kształtu `slanted` i z pewnymi brakami. Na przykład zamiast literki „l” w zbiorze `EBGaramond08 Italic` renderuje się samo „l” (braku tego nie ma zbiór `EBGaramond12`). Zaletą pakietu w porównaniu do innych jest to, że generalnie dobrze obsługiwane są w nim polskie znaki oraz że pakiet ten można znaleźć w różnych dystrybucjach `latexa` (`MikTeX` instaluje go automatycznie).

4.3. Krój i wielkość czcionek

Główny tekst pracy powinien być zredagowany z wykorzystaniem czcionki `Times`, typ normalny, o wysokości 12pt, z odstępem między liniami równym 14.5pt. Istnieje możliwość zmiany odstępu między liniami za pomocą komendy `\linespread`, jednak zaleca się pozostawienie tego odstępu jak w niniejszym dokumencie (`\linespread{1}`). Wymagania odnośnie kroju pisma pozostałych elementów (nagłówków, stopek itp.) zamieszczono w tabeli 4.1.

W szablonie zastosowano czcionkę `texgyre-termes` (dostarcza ją pakiet `tgtermes`). Czcionka ta jest klonem czcionki `Times`, w którym obsługiwane jest środkowoeuropejskie kodowanie znaków (podobnie jak w przypadku czcionki `ebgaramond`, dzięki czemu polskie literki nie są zlepkami dwóch znaków lecz pojedynczymi znakami).

Wszelkie przykłady źródeł kodu (fragmenty programów, komendy linii poleceń), nazwy plików i uruchamianych programów powinny być pisane czcionką maszynową. W szablonie czcionką maszynową jest `tlxtt`. Czcionka ta obsługuje polskie znaki. Dostarcza ją pakiet `txfonts`, który należy wcześniej zainstalować (`MikTeX` zainstaluje go automatycznie podczas pierwszej kompilacji szablonu).

Jeśli w pracy zostaną użyte otoczenia matematyczne, to w dokumencie wynikowym pojawią się dodatkowe czcionki (domyślne `latexowe` czcionki do wyrażeń matematycznych). Dzięki zastosowaniu opcji `extrafontsizes` w klasie `memoir` nie dość, że otrzymuje się większe

Tab. 4.1: Zestawienie czcionek elementów podziału dokumentu, tekstu wiodącego, nagłówka i stopki oraz podpisów (Rozm. – rozmiar czcionki, Odst. – baselineskip)

| Element | Przykład | Czcionka | Rozm. | Odst. |
|----------------------|-------------------------------|------------------------------------|--------|--------|
| Nr rozdziału | Rozdział 1 | <code>\huge \bfseries</code> | 25pt | 30pt |
| Tytuł rozdziału | Wstęp | <code>\Huge \bfseries</code> | 30pt | 37pt |
| Nr i tytuł sekcji | 1.1. Wprowadzenie | <code>\Large \bfseries</code> | 17pt | 22pt |
| Nr i tytuł podsekcji | 1.1.1. Cel szczegółowy | <code>\large \bfseries</code> | 14.5pt | 18pt |
| Tytuł podpodsekcji | Założenia | <code>\normalsize \bfseries</code> | 12pt | 14.5pt |
| Tytuł paragrafu | Podstawy Opis ... | <code>\normalsize \bfseries</code> | 12pt | 14.5pt |
| Tekst wiodący | Niniejszy dokument ... | <code>\normalsize</code> | 12pt | 14.5pt |
| Nagłówek strony | 3.2. Czcionka wiodąca ... | <code>\small \itshape</code> | 11pt | 13.6pt |
| Stopka strony | Imię Nazwisko: ... | <code>\small</code> | 11pt | 13.6pt |
| Podpisy tabel | Tab. 3.1: Zestawienie ... | <code>\small</code> | 11pt | 13.6pt |
| Podpisy rysunków | Rys. 3.1: Oficjalny ... | <code>\small</code> | 11pt | 13.6pt |

czcionki (30pt), to jeszcze zamiast Computer Modern do wzorów matematycznych jest stosowana czcionka Latin Modern (wywodząca się z Computer Modern). Stąd lista wszystkich użytych czcionek może być następująca:

```
EBGaramond12-Regular
GaramondNo8-Reg-Norml
TeXGyreTermes-Regular-Normalna
TeXGyreTermes-Bold-Pogrubiona
TeXGyreTermes-Italic-Normalna
t1xtt-Nomal
LMMathItalic12-Regular
LMMathSymbols10-Regular
LMMathExtension10-Regular
LMRoman8-Regular
```

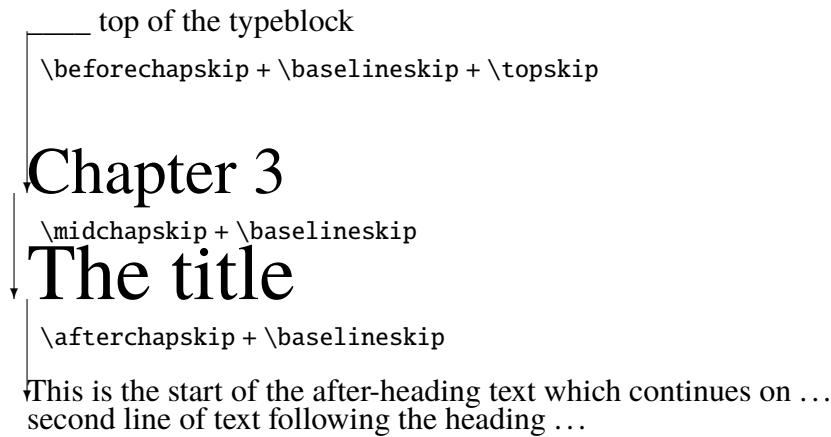
Aby wykorzystać te czcionki poza systemem LaTeX, wystarczy pobrać je spod adresów (ważnych na dzień 1.04.2016): <https://www.ctan.org/tex-archive/fonts/cm/ps-type1/bakoma/ttf/?lang=en>, <http://www.gust.org.pl/projects/e-foundry/latin-modern>, <http://www.gust.org.pl/projects/e-foundry/tex-gyre>, <https://bitbucket.org/georgd/eb-garamond/downloads>, a następnie zainstalować w systemie. Dzięki temu można będzie np. edytować rysunki używając dokładnie tej samej czcionki, co czcionka użyta w dokumencie.

4.4. Formatowanie bloków tekstu

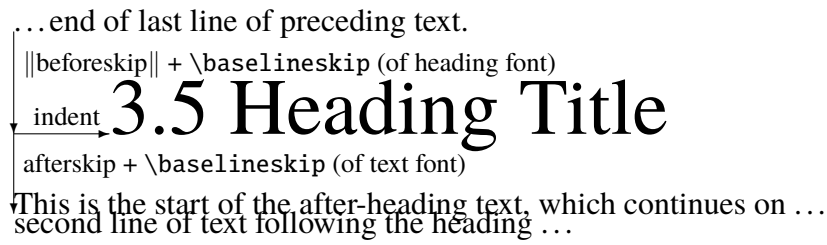
Każdy rozdział pracy powinien rozpoczynać się od nowej strony. Jej wygląd powinien być kontrolowany parametrami pokazanymi na rysunku 4.4. W niniejszym szablonie (dokument klasy memoir z opcją [12pt]) przyjęto następujące wartości tych parametrów:

- `\beforechapskip` (50.0pt) + `\baselineskip` of `\huge` (30pt) + `\topskip` (12.0pt) = 92pt (3.246cm)
- `\midchapskip` (20.0pt) + `\baselineskip` of `\Huge` (37pt) = 57 pt (2.011cm)
- `\afterchapskip` (40.0pt) + `\baselineskip` of `\normalsize` (14.5pt) = 54.5pt (1.923cm)

Nieco kłopotów może sprawić dobre ustawienie na stronie tytułów nienumerowanych rozdziałów oraz list generowanych automatycznie (Skróty, Spis treści, Spis rysunków, Spis tabel,



Rys. 4.4: Parametry sterujące wielkościami odstępów na stronie z tytułem rozdziału



Rys. 4.5: Kontrola ustawień odległości w tytułach kolejnych sekcji

Indeks rzeczowy). W szablonie w tym celu zdefiniowano nowy styl rozdziału komendami jak niżej (w szablonie są to komendy zamarkowane)

```
\newlength{\linespace}
\setlength{\linespace}{-\beforechapskip-\topskip+\headheight+\topsep}
\makechapterstyle{noNumbered}{%
\renewcommand\chapterheadstart{\vspace*{\linespace}}
}
```

oraz dokonano przełączenia stylów rozdziałów komendami `\chapterstyle{nonumbered}` oraz `\chapterstyle{default}` podczas dołączania do dokumentu wymienionych nienumerowanych rozdziałów i list. Aby „podnieść do góry” tytuły nienumerowanych rozdziałów (gdyby jest to rzeczywiście konieczne) wystarczy odmarkować wspomniane komendy.

Tytuły rozdziałów, sekcji, podsekcji itd. nie powinny kończyć się kropką. Odległości pomiędzy tekstem wiodącym a tytułem sekcji powinien być regulowany parametrami pokazanymi na rysunku 4.5. Rozmiar `\baselineskip` zależy od rozmiaru czcionki (zobacz tabela 4.1), zaś `before skip` i `secskip` od poziomu sekcji. W niniejszym szablonie przyjęto następujące wartości tych parametrów (są to wartości dobierane elastycznie podczas kompilacji):

- `indent` = 14.5pt
- `parskip` = 0.0pt
- `beforesecskip` = -18.08334pt plus -5.16667pt minus -1.03331pt
- `aftersecskip` = 11.88335pt plus 1.03331pt
- `beforesubsecskip` = -16.79167pt plus -5.16667pt minus -1.03331pt
- `aftersubsecskip` = 7.75pt plus 1.03331pt
- `beforesubsubsecskip` = -16.79167pt plus -5.16667pt minus -1.03331pt

- `aftersubsubsecskip` = 7.75pt plus 1.03331pt

W szablonie obowiązują również następujące wartości parametrów odpowiedzialnych za odstępy pomiędzy pływającymi figurami, tekstami oraz tekstem i figurą:

- `floatsep` = 12.0pt plus 2.0pt minus 2.0pt
- `intextsep` = 14.0pt plus 4.0pt minus 4.0pt
- `textfloatsep` = 20.0pt plus 2.0pt minus 4.0pt

Pierwsza linia pierwszego akapitu w bloku (po tytule rozdziału, sekcji, podsekcji, podpodsekcji) nie może mieć wcięcia. Pierwsze linie w kolejnych akapitach już powinny mieć wcięcie równe 14.5pt. Tekst w akapitach powinien być wyrównany z obu stron.

Strony powinny być numerowane numeracją ciągłą (sekwencja arabskich cyfr). Numery stron powinny być umieszczone w ich stopkach (tj. tak jak w niniejszym dokumencie). Wyjątkiem są tutaj pierwsze strony rozdziałów oraz strona tytułowa – na nich numery nie powinny się pojawić.

W pracy należy dbać o poprawność redakcyjną zgodnie z zaleceniami:

- nie zostawiać znaku spacji przed znakami interpunkcji („powiedziano , że ...” -> „powiedziano, że ...”),
- kropki po skrótach, które nie są jednocześnie kropkami kończącymi zdanie należy sklejać z kolejnym wyrazem znakiem tyldy, np. jak tutaj (`np.~jak tutaj`) lub wstawiać za nimi ukośnik, np. jak tutaj (`np.\ jak tutaj`)
- nie zapominać o dobrym sformatowaniu wyliczenia (należy zaczynać małymi literami lub dużymi oraz kończyć przecinkami, średnikami i kropkami – w zależności od kontekstu danego wyliczenia),
- nie zostawiać samotnych literek na końcach linii (można je „skleić” z wyrazem następnym stosując znaczek tilde, jak w~przykładzie).
- nie zostawiać pojedynczych wierszy na końcu lub początku strony (należy kontrolować „sieroty” i „wdowy”),
- nie zostawiać odstępu pomiędzy tekstem a nawiasami czy znakami cudzysłowów (znaki te powinny przylegać do tekstu, który obejmują „jak w tym przykładzie”),
- wyrazy obcojęzyczne powinny być pisane czcionką italic wraz ze skrótem oznaczającym język, w szczególności ma to zastosowanie przy rozwijaniu skrótów, np. OGC (ang. Open Geospatial Consortium),
- każdy zastosowany skrót powinien zostać rozwinięty podczas pierwszego użycia, później może już występować bez rozwinięcia (skrót i jego rozwinięcie powinny trafić również do wykazu Skrótów, jeśli taki wykaz jest dołączany do dokumentu).

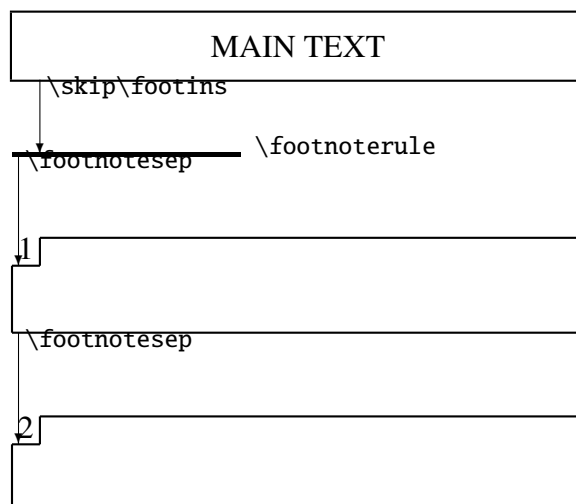
4.5. Opisy tabel i rysunków

Podpisy powinny być umieszczane pod rysunkami lub nad tabelami wraz z etykietą składającą się ze skrótu Rys. lub Tab. oraz numeru. Podpisy te nie powinny mieć końcowej kropki. Numery występujący w podpisach powinny zaczynać się numerem rozdziału, po którym następuje kolejny numer rysunku lub tabeli w obrębie rozdziału. Etykieta powinna kończyć się dwukropkiem, po którym następuje tekst podpisu. Numer rozdziału powinien być rozdzielony kropką od kolejnego numeru w rysunku bądź tabeli w rozdziale (liczniki tabel i rysunków są rozłączne). Należy pamiętać o tym, żeby w całej pracy tabele miały podobny wygląd (rodzaj czcionki, ewentualne pogrubienia w nagłówku itp.).

4.6. Przypisy dolne

Istnieje możliwość zamieszczania przypisów na dole strony, choć nie jest to zalecane (przykładowo ¹). Sposób parametryzowania ich wyglądu pokazano na rysunku 4.6. W szablonie wykorzystano następujące, domyślne wartości tych parametrów:

```
\footins = 12pt \footnotesep = 8pt
\baselineskip = 10pt note separation = 40pt
rule thickness = 0.4pt
rule length = 0.25 times the \textwidth
```



Rys. 4.6: Parametry sterujące przypisami dolnymi

4.7. Formatowanie spisu treści

W klasie memoir istnieją komendy pozwalające dość dobrze zarządzać wyglądem spisu treści. Na rysunku 4.7 pokazano, za pomocą jakich parametrów można wpływać na finalną jego postać. W szablonie wykorzystano następujące, domyślne ich wartości:

```
indent = 18pt
numwidth = 28pt
\@tocrmarg = 31pt
\@pnumwidth = 19pt
\@dotsep = 4.5
```

4.8. Formatowanie list wyliczeniowych i wypunktowań

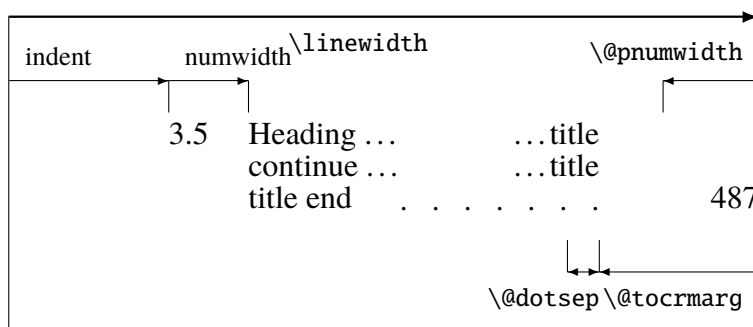
Standardowo sposób formatowania list można parametryzować jak pokazano na rysunku 4.8. Jednak czasem trudno poradzić sobie z niektórymi rzeczami, jak np. znakami wypunktowania. Dlatego w szablonie wykorzystano pakiet `enumi`. Pozwala on na łatwe zarządzanie wyglądem list. W szablonie zastosowano następujące globalne ustawienia dla tego pakietu:

```
\usepackage{enumitem}
\setlist{noitemsep,topsep=4pt,parsep=0pt,partopsep=4pt,leftmargin=*}
\setenumerate{labelindent=0pt,itemindent=0pt,leftmargin=!,label=\arabic*..}
```

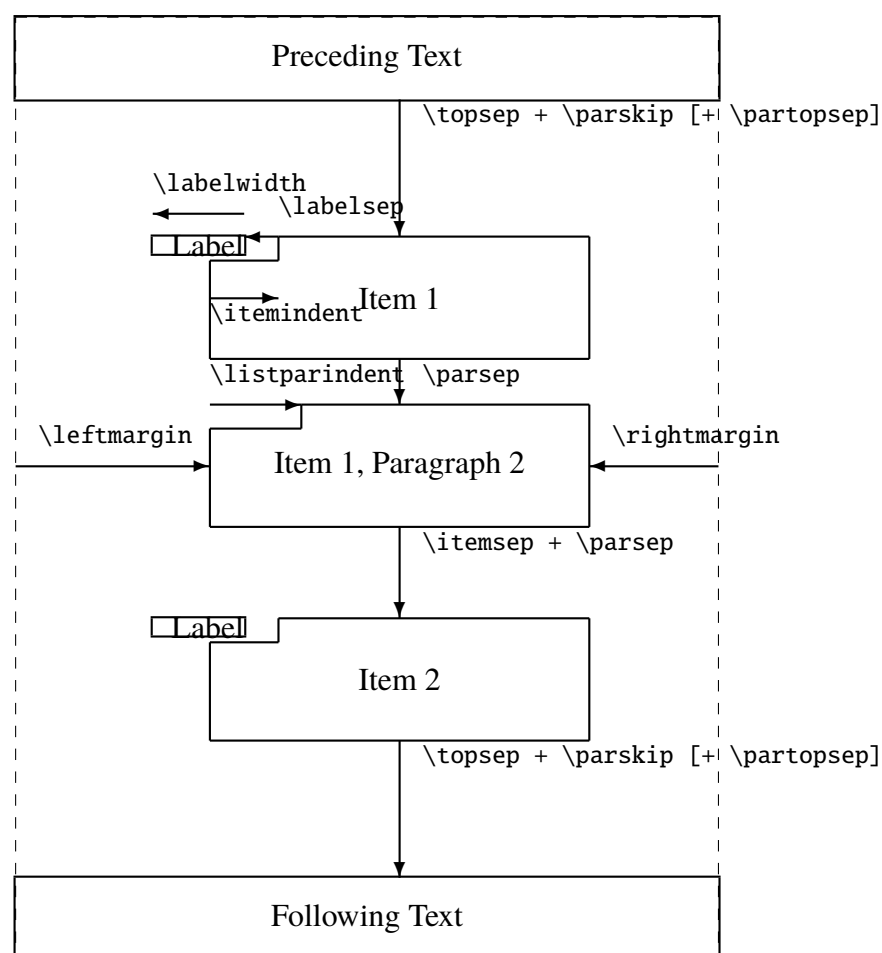
¹Tekst przypisu

```
\setlistdepth{4}
\setlist[itemize,1]{label=$\bullet$}
\setlist[itemize,2]{label=\normalfont\bfseries\textendash}
\setlist[itemize,3]{label=$\ast$}
\setlist[itemize,4]{label=$\cdot$}
\renewlist{itemize}{itemize}{4}
```

W podrozdziale 5.2 pokazano przykład wykorzystania możliwości komend oferowanych w pakiecie `enumi`.



Rys. 4.7: Parametryzacja wyglądu spisu treści



Rys. 4.8: Parametryzacja list wyliczeniowych i wypunktowań

4.9. Wzory matematyczne

Wzory matematyczne, jeśli mają być osobnymi formułami, powinny być wycentrowane, z numeracją umieszczoną na końcu linii i ujętą w okrągłe nawiasy (zobacz równanie (4.1)). Numery równań powinny zawierać numer rozdziału oraz kolejny numer równania w obrębie rozdziału (podobnie jak przy numerowaniu rysunków i tabel). Spełnienie tych warunków zapewnia otoczenie `equation`. Nie wszystkie formuły trzeba numerować (nienumerowane wzory można osiągnąć stosując otoczenie `\equation*`). Właściwie należy numerować tylko te, do których tworzy się jakieś odniesienia w tekście. Jeśli wzory umieszczane są w linijce tekstu, to można zastosować otoczenie matematyczne inline, jak w przykładzie $\int_0^{10\nu\sum i} x dx$ (wyprodukowanym komendą `\int_{0}^{10\nu\sum i}{x dx}`). Tylko że wtedy może dojść do rozszerzenia odstępów pomiędzy liniami tekstu (aby zmieścił się wzór).

$$\int_0^{10\nu\sum i} x dx \quad (4.1)$$

Rozdział 5

Redakcja pracy

5.1. Układ pracy

Standardowo praca powinna być zredagowana w następującym układzie:

| |
|---------------------------------|
| Strona tytułowa |
| Strona z dedykacją (opcjonalna) |
| Spis treści |
| Spis rysunków (opcjonalny) |
| Spis tabel (opcjonalny) |
| Skróty (wykaz opcjonalny) |
| 1. Wstęp |
| 1.1 Cel i zakres pracy |
| 1.2 Układ pracy |
| 2. Kolejny rozdział |
| 2.1 Sekcja |
| 2.1.1 Podsekcja |
| Nienumerowana podpodsekcja |
| Paragraf |
| ... |
| #. Podsumownie i wnioski |
| Literatura |
| A. Dodatek |
| A.1 Sekcja w dodatku |
| ... |
| \$. Zawartość płyty CD/DVD |
| Indeks rzeczowy (opcjonalny) |

Spis treści – powinien być generowany automatycznie, z podaniem tytułów i numerów stron. Typ czcionki oraz wielkość liter spisu treści powinny być takie same jak w niniejszym wzorcu.

Spis rysunków, Spis tabel – powinny być generowane automatycznie (podobnie jak Spis treści). Elementy te są opcjonalne (robienie osobnego spisu, w którym na przykład są tylko dwie pozycje specjalnie nie ma sensu).

Wstęp – pierwszy rozdział, w którym powinien znaleźć się opis dziedziny, w jakiej osadzona jest praca, oraz wyjaśnienie motywacji do podjęcia tematu. W sekcji „Cel i zakres” powinien znaleźć się opis celu oraz zadań do wykonania, zaś w sekcji „Układ pracy” – opis zawartości kolejnych rozdziałów.

Podsumowanie – w rozdziale tym powinny być zamieszczone: podsumowanie uzyskanych efektów oraz wnioski końcowe wynikające z realizacji celu pracy dyplomowej.

Literatura – wykaz źródeł wykorzystanych w pracy (do każdego źródła musi istnieć odpowiednie cytowanie w tekście). Wykaz ten powinien być generowany automatycznie.

Dodatki – miejsce na zamieszczanie informacji dodatkowych, jak: Instrukcja wdrożeniowa, Instrukcja uruchomieniowa, Podręcznik użytkownika itp. Osobny dodatek powinien być przeznaczony na opis zawartości dołączonej płyty CD/DVD. Założono, że będzie to zawsze ostatni dodatek.

Indeks rzeczowy – miejsce na zamieszczenie kluczowych wyrazów, do których czytelnik będzie chciał sięgnąć. Indeks powinien być generowany automatycznie. Jego załączanie jest opcjonalne.

5.2. Styl

Zasady pisania pracy (przy okazji można tu zaobserwować efekt wyrównania wpisów występujących na liście wyliczeniowej uzależnione od długości etykiety):

1. Praca dyplomowa powinna być napisana w formie bezosobowej („w pracy pokazano ...”). Taki styl przyjęto na uczelniach w naszym kraju, choć w krajach anglosaskich preferuje się redagowanie treści w pierwszej osobie.
 2. W tekście pracy można odwołać się do myśli autora, ale nie w pierwszej osobie, tylko poprzez wyrażenia typu: „autor wykazał, że ...”.
 3. Odwołując się do rysunków i tabel należy używać zwrotów typu: „na rysunku pokazano ...”, „w tabeli zamieszczono ...” (tabela i rysunek to twory nieżywotne, więc „rysunek pokazuje” jest niepoprawnym zwrotem).
 4. Praca powinna być napisana językiem formalnym, bez wyrażeń żargonowych („sejwowanie” i „downloadowanie”), nieformalnych czy zbyt ozdobnych („najznamienitszym przykładem tego niebywałego postępu ...”)
 5. Piszac pracę należy dbać o poprawność stylistyczną wypowiedzi
 - trzeba pamiętać, do czego stosuje się „liczba”, a do czego „ilość”,
 - nie „szereg funkcji” tylko „wiele funkcji”,
 - redagowane zdania nie powinny być zbyt długie (lepiej podzielić zdanie wielokrotnie złożone na pojedyncze zdania),
 - itp.
 6. Zawartość rozdziałów powinna być dobrze wyważona. Nie wolno więc generować sekcji i podsekcji, które mają zbyt mało tekstu lub znacząco różnią się objętością. Zbyt krótkie podrozdziały można zaobserwować w przykładowym rozdziale 7.
 7. Niedopuszczalne jest pozostawienie w pracy błędów ortograficznych czy tzw. literówek – można je przecież znaleźć i skorygować automatycznie.
10005. Niedopuszczalne jest pozostawienie w pracy błędów ortograficznych czy tzw. literówek – można je przecież znaleźć i skorygować automatycznie.

Rozdział 6

Uwagi techniczne

6.1. Rysunki

W niniejszym szablonie numeracja rysunków odbywa się automatycznie według następujących reguł: rysunki powinny mieć numerację ciągłą w obrębie danego rozdziału, sam zaś numer powinien składać się z dwóch liczb rozdzielonych kropką. Pierwsza liczbą ma być numer rozdziału, drugą – kolejny numer rysunku w rozdziale. Przykładowo: pierwszy rysunek w rozdziale 1 powinien mieć numer 1.1, drugi – numer 1.2 itd., pierwszy rysunek w rozdziale 2 powinien mieć numer 2.1, drugi – numer 1.2 itd.

Rysunki powinny być wyśrodkowane na stronie wraz z podpisem umieszczonym na dole. Podpisy nie powinny kończyć się kropką. Czcionka podpisu powinna być mniejsza od czcionki tekstu wiodącego o 1 lub 2 pkt (w szablonie jest to czcionka rozmiaru small). Ponadto należy zachowywać odpowiedni odstęp między rysunkiem, podpisem rysunku a tekstem rozdziału. W przypadku korzystania z szablonu odstępny te regulowane są automatycznie. Podpis i grafika muszą stanowić jeden obiekt. Chodzi o to, że w edytorach tekstu typu Office podpis nie scala się z grafiką i czasem trafia na następną stronę, osieracając grafikę. Korzystającym z niniejszego szablonu i otoczenia `\figure` takie osierocenie nigdy się nie zdarzy.

Do każdego rysunku musi istnieć odwołanie w tekście (inaczej mówiąc: niedopuszczalne jest wstawienie do pracy rysunku bez opisu). Odwołania do rysunków powinny mieć postać: „Na rysunku 3.3 przedstawiono...” lub „... co ujęto na odpowiednim schemacie (rys. 1.7)”. Jeśli odwołanie stanowi część zdania, to wtedy wyraz „rysunek” powinien pojawić się w całości. Jeśli zaś odwołanie jest ujęte w nawias (jak w przykładzie), wtedy należy zastosować skrót „rys.”. Jeśli do stworzenia obrazka wykorzystano jakieś źródła, to powinny one być zacytowane w podpisie tegoż rysunku.

Należy pamiętać o tym, że „rysunki” to twory nieżywotne. W związku z tym nie mogą „pokazywać”. Dlatego „rysunek 1.1 pokazuje ...” jest stylistycznie niepoprawne. Zamiast tego zwrotu trzeba użyć „na rysunku 1.1 pokazano ...”.

Rysunki można wstawiać do pracy używając polecenia `\includegraphics`. Zalecane jest, aby pliki z grafikami były umieszczane w katalogach odpowiadających numerom rozdziałów czy literom dodatków: `rys01`, `rysA` itd. Sposób wstawiania rysunków do pracy zademonstrowano na przykładzie rysunków 6.1 i 6.2.

Listing 6.1: Kod źródłowy przykładów wstawiania rysunków do pracy

```
\begin{figure}[ht]
  \centering
  \includegraphics[width=0.3\linewidth]{rys05/kanji-giri}
  \caption{Dwa znaki kanji - giri}
  \label{fig:kanji-giri}
\end{figure}
```

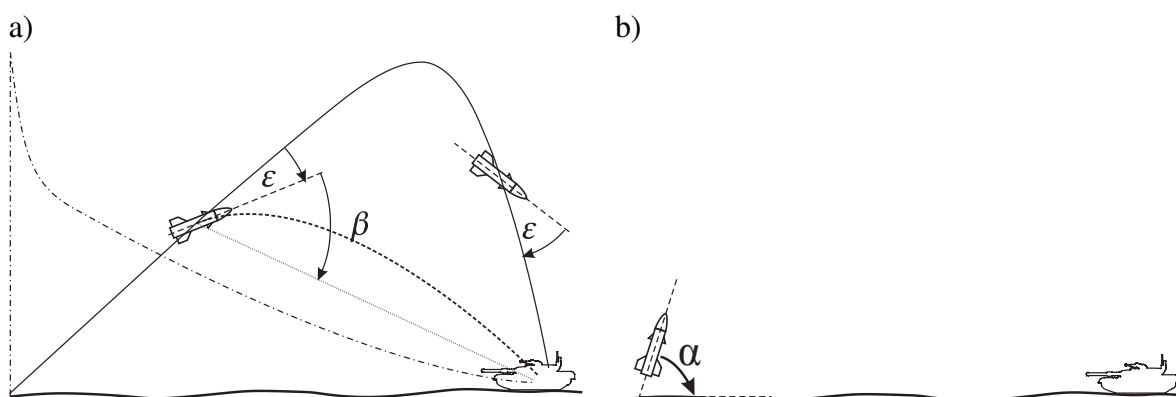
```

\begin{figure}[htb]
\centering
\begin{tabular}{@{}ll@{}}
a) & b) \\
\includegraphics[width=0.475\textwidth]{rys05/alfa1} & \\
\includegraphics[width=0.475\textwidth]{rys05/beta1} & \\
\end{tabular}
\caption{Wyznaczanie trajektorii lotu rakiety:  
a) trzy podejścia, b) podejście praktyczne}
\label{fig:alfabeta}
\end{figure}

```

義理

Rys. 6.1: Dwa znaki kanji – giri



Rys. 6.2: Wyznaczanie trajektorii lotu rakiety: a) trzy podejścia, b) podejście praktyczne

Grafiki wektorowe powinny być dostarczone w plikach o formacie pdf. Rozmiar strony w pliku pdf powinien być troszeczkę większy niż zamieszczona na nim grafika (proszę spojrzeć na przykłady grafik wykorzystanych w niniejszym szablonie). Chodzi o to, aby na rysunku nie pojawiała się niepotrzebna biała przestrzeń. Grafiki rastrowe (głównie zrzuty z ekranu bądź zdjęcia) powinny być dostarczane w plikach o formacie png z kompresją bezstratną. Zastosowanie kompresji stratnej, jak jpg, wprowadza niepotrzebne artefakty. Podobnie jak w przypadku grafik wektorowych, grafiki rastrowe nie powinny mieć białych marginesów.

Na rysunkach nie powinno stosować się 100% czarnego wypełnienia, bo robią się plamy przebijające się przez kartkę. Zamiast tego wypełnienie powinno być ok. 90% czerni.

Czcionka na rysunkach nie może być większa od czcionki wiodącej tekstu (jedyne wyjątek to np. jakieś nagłówki). Należy stosować czcionkę kroju Arial, Helvetica bądź tego samego kroju co czcionka dokumentu (`texgyre-termes`).

Jeśli na jednym rysunku pojawić się ma kilka grafik, to zamiast stosować `subfigure` lub inne otoczenia należy wstawić grafiki w tabelę, opisać ją indeksami a) i b), a potem odnieść się do tego w podpisie (rys. 6.2). Czasem pomaga w pozycjonowaniu rysunków użycie komendy:

```

\top{\vskip3ex\hbox{\includegraphics[width=0.475\textwidth]{nazwa}}}

```

Na rysunkach nie wolno nadużywać kolorów oraz ozdóbników (wiele narzędzi do tworzenia diagramów dostarcza grafikę z cieniowaniem, gradacją kolorów itp. co niekoniecznie przekłada się na czytelność rysunku).

Podczas robienia zrzutów z ekranu należy zadbać o to, by taki zrzut był czytelny po wydrukowaniu. Czyli aby pojawiające się literki były wystarczająco duże, a przestrzenie bez treści – relatywnie małe. Przystępując do robienia zrzutu trzeba odpowiednio wyskalować elementy na ekranie. Na przykład robiąc zrzut z przeglądarki FF najpierw należy wcisnąć CTR+0 (domyślne skalowanie), potem CTR+ (zmniejszenie skali o stopień). Potem dobrze jest zawęzić okno przeglądarki tak, by interesująca treść wypełniła je w całości. Jeśli na obserwowanej stronie jest zbyt dużo pustych obszarów, to należy je jakoś zawęzić (sterując wielkością okna przeglądarki lub aktywnymi elementami interfejsu użytkownika). Zrzut bowiem wcale nie musi być odwzorowaniem 1:1 domyślnego układu obserwowanych elementów. Ważne jest, by na zrzucie z ekranu pokazać interesujący, opisywany fragment i żeby ten fragment był czytelny.

Czasem problemem jest tworzenie zrzutów z ekranu, gdy występują na nim dane wrażliwe. Istnieją dwa sposoby na radzenie sobie z tym problemem. Pierwszy polega na zastąpieniu w systemie danych rzeczywistych danymi testowymi – wygenerowanymi tylko do celów prezentacji. Zrzut robi się wtedy na bazie danych testowych. Drugi polega na wykonaniu zrzutu z ekranu, na którym pokazano dane rzeczywiste, i następnie zamianie tych danych już w pliku graficznym za pomocą odpowiedniego edytora (np. gimp). Czyli oryginalny zrzut z ekranu należy otworzyć w edytorze, a potem nadpisać oryginalny tekst własnym tekstem. Konieczne jest wtedy dobranie odpowiednich czcionek aby nie było widać wprowadzonych zmian.

Uwaga: takie manipulowanie zrzutami jest usprawiedliwione jedynie w przypadku konieczności ochrony danych wrażliwych czy też lepszego pokazania wybranych elementów. Nie może to prowadzić generowania fałszywych rezultatów!!!

6.2. Wstawianie kodu źródłowego

Kod źródłowy można wstawiać jako blok tekstu pisany czcionką maszynową. Używa się do tego otoczenie `\lstlisting`. W atrybutach otoczenia można zdefiniować tekst podpisu wstawianego wraz z numerem nad blokiem, etykietę do tworzenia odwołań, sposób formatowania i inne ustawienia. Zaleca się stosowanie w tym otoczeniu następujących parametrów:

```
\begin{lstlisting}[label=list:req1,caption=Initial HTTP Request,
                    basicstyle=\footnotesize\ttfamily]
```

Szczególnie przydatne podczas wstawiania większej ilości kodu źródłowego jest zastosowanie parametru `basicstyle=\footnotesize\ttfamily`. Dzięki niemu zmniejsza się czcionka, a przez to na stronie można zmieścić dłuższe linijki kodu. Użycie tak zdefiniowanego parametru nie jest jednak sztywnym zaleceniem. Wielkość czcionki można dobierać do potrzeb.

Listing 6.2: Initial HTTP Request

```
GET /script/Articles/Latest.aspx HTTP/1.1
Host: www.codeproject.com
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml
User-Agent: Mozilla/5.0 ...
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US...
Accept-Charset: windows-1251, utf-8...
```

Można też sformatować kod bez stosowania numerowanego podpisu (wtedy nie zamieszcza się `caption` na liście atrybutów).


```
GET /script/Articles/Latest.aspx HTTP/1.1
Host: www.codeproject.com
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml
User-Agent: Mozilla/5.0 ...
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US...
Accept-Charset: windows-1251,utf-8...
```

Istnieje możliwość wstawiania kodu źródłowego w bieżącej linii tekstu. Można to zrobić na kilka sposobów:

- korzystając z polecenia `\texttt` ustawiającego czcionkę maszynową, jak w przykładzie tutaj (efekt zastosowania komendy `\texttt{tutaj}`). Problemem jednak mogą okazać się znaki podkreślenia i inne znaki kontrolne.
- korzystając z otoczenia `\verb` zapewniającego wypisanie kodu czcionką maszynową jak w przykładzie tutaj (efekt zastosowania komendy `\verb|tutaj|`). Problemem jest to, że polecenie `\verb` nie potrafi łamać dłuższego tekstu.
- korzystając z polecenia `\lstin` umożliwiającego wypisanie kodu czcionką ustawianą w opcjach jak w przykładzie tutaj (efekt komendy `\lstset{basicstyle=\ttfamily}\lstinline{tutaj}`) lub tutaj (efekt komendy `\lstinline[basicstyle=\ttfamily]=tutaj=`).

6.3. Wykaz literatury oraz cytowania

Cytowania powinny być zamieszczane w tekście z użyciem komendy `\cite{}`. Jej argumentem powinien być klucz cytowanej pozycji (lub lista kluczy rozdzielonych przecinkiem bez spacji, jeśli takich pozycji w danym miejscu cytuje się więcej) jaki jest używany w bazie danych bibliograficznych (plik dokumentacji `.bib`). Po kompilacji `bibtex` i `pdflatex` w tekście pojawia się właściwy odsyłacz do pozycji w wykazie literatury (ujęty w kwadratowe nawiasy – zgodnie z tym, co definiuje styl `plabrv.bst`), zaś w samym wykazie (rozdział Literatura) – zacytowana pozycja. Przykładem cytowania jest: „dobrze to opisano w pracach [?, ?]” (gdzie zastosowano komendę `\cite{JS07,SQL2}`).

Co do zawartości rekordów bibliograficznych - style `bibtex`owe potrafią „skracać” imiona (czyli wstawiać, jeśli taka wola, inicjały zamiast pełnych imion). Niemniej dobrze jest od razu przyjąć jakąś konwencję. Proponuje się, aby w rekordach od razu wstawiane były inicjały zamiast pełnych imion.

Niekiedy tytuły prac zawierają wyrazy z dużymi i małymi literami. Takie tytuły należy brać w podwójne nawiasy klamrowe, aby `bibtex` nie zamienił ich na postać, w której poza pierwszą literą pozostałe są małe.

Jeśli jakiś cytowany zasób pochodzi z Internetu, to jego rekord w pliku `bib` powinien wyglądać jak niżej.

```
@INPROCEEDINGS{SQL2,
  title={{A MySQL-based data archiver: preliminary results}},
  author={Bickley, M. and Slominski, Ch.},
  booktitle = {{Proceedings of ICALEPCS07}},
  month = oct,
  day = {15--19},
  year={2007},
  note={\url{http://www.osti.gov/scitech/servlets/purl/922267}
    [dostęp dnia 20 czerwca 2015]}
}
```

A to inny przykład rekordu danych bibliograficznych:

```
@TechReport{JS07,
  author = {Jędrzejczyk, J. and Śródka, B.},
  title = {Segmentacja obrazów metodą drzew decyzyjnych},
  year = {2007},
  institution = {Politechnika Wrocławska, Wydział Elektroniki}
}
```

6.4. Indeks rzeczowy

Generowanie indeksu po trosze wygląda jak generowanie wykazu literatury – wymaga kilku kroków. Podczas pierwszej kompilacji `pdflatex` generowany jest plik z rozszerzeniem `*.idx` (zawierający „surowy indeks”). Następnie, bazując na tym pliku, generowany jest plik z rozszerzeniem `*.ind` zawierający sformatowane dane. Ten krok wymaga uruchomienia odpowiedniego narzędzia oraz zastosowania pliku z definicją stylu `Dyplom.ist`. W kroku ostatnim dokonuje się kolejnej kompilacji `pdflatex` (dzięki niej w wynikowym dokumencie pojawi się Indeks rzeczowy). Domyślnie Indeks rzeczowy zostanie sformatowany w układzie dwukolumnowym.

Oczywiście aby to wszystko zadziało w kodzie szablonu należy umieścić odpowiednie komendy definiujące elementy indeksu rzeczowego (`\index`) oraz wstawiające sformatowany Indeks rzeczowy do dokumentu wynikowego (`\printindex`). Więcej informacji o tworzeniu indeksu rzeczowego można znaleźć na stronie <https://en.wikibooks.org/wiki/LaTeX/Indexing>. Poniżej przedstawiono przykłady komend użytych w szablonie do zdefiniowania elementów indeksu rzeczowego:

- `\index{linia komend}` – pozycji główna.
- `\index{generowanie!-- indeksu}` – podpozycja.

Generowanie pliku `*.ind` można inicjować na kilka sposobów:

- poprzez wydanie odpowiedniego polecenia bezpośrednio w linii komend
`makeindex Dyplom.idx -t Dyplom.ilg -o Dyplom.ind -s Dyplom.ist`
- poprzez odpalenie odpowiedniego narzędzia środowiska. Na przykład w `TeXnicCenter` definiuje się tzw. `output profiles`:

```
makeindex "%tm.idx" -t "%tm.ilg" -o "%tm.ind" -s "%tm.ist"
```

a samo generowanie pliku `*.ind` zapewni wybranie pozycji menu `Build/Makeindex`.

- korzystając z odpowiednio sparametryzowanych pakietów i komend wewnątrz kompilowanego dokumentu (czyli od razu przy okazji jego kompilacji).

```
\DisemulatePackage{imakeidx}
\usepackage[noautomatic]{imakeidx}
% jeśli chcemy, by indeks by generowany automatycznie programem makeindex:
%\usepackage[makeindex]{imakeidx}
% a tak ponoć można przekazać opcje do programu generującego indeks:
%\makeindex[options=-s podrecznik -L polish -M lang/polish/utf8]
%\makeindex[options=-s podrecznik]
\makeindex
```

Niestety, `makeindex` jest narzędziem, które umieszcza część pozycji w grupie `Symbols`, a nie w grupach związanych z literkami alfabetu (w związku z czym indeksowany element zaczynający się od polskiej literki trafia do grupy `Symbols`, jak np. `\index{Światło}`. Jeśli chce się zamieszczać w indeksie symbole matematyczne, to dobrze jest to robić jak w następującym przykładzie: `\index{$asterisk@$ast$}` czy też `\index{c@$mathcal{C}$}`, tj. dostarczając przy okazji klucz do sortowania. Lepiej w tym względzie radzą sobie inne

narzędzia, jak `texindy` lub `xindy` dostępne pod linuxem. Korzystając z nich uzyskuje się grupy polskich literek w indeksie rzeczowym (hasła zaczynające się od polskich literek już nie trafiają do grupy Symbols). Przykład polecenia wydanego z linii komend, w którym wykorzystano `texindy` zamieszczono poniżej (zakładamy kodowanie plików w UTF8, można dla niniejszego szablonu zmienić na `cp1250`):

```
texindy -L polish -M lang/polish/utf8 Dyplom.idx
```

To polecenie wygeneruje `Dyplom.ind` o zawartości:

```
\begin{theindex}
  \providecommand*\lettergroupDefault[1]{}
  \providecommand*\lettergroup[1]{%
    \par\textbf{#1}\par
    \nopagebreak
  }

  \lettergroup{G}
  \item generowanie
    \subitem -- indeksu, 27
    \subitem -- wykazu literatury, 27

  \indexspace

  \lettergroup{L}
  \item linia komend, 27

  \indexspace

  \lettergroup{Ś}
  \item \textit{Świat} \LaTeX, 28

\end{theindex}
```

Aby mieć większą kontrolę automatyczne generowanie indeksu zostało w niniejszym szablonie wyłączone (indeks trzeba wygenerować samemu, wydając polecenie `makeindex` lub zalecane `texindy`).

6.5. Inne uwagi

Dobrym sposobem na kontrolę błędów występujących podczas kompilacji jest wstawianie linijki `\end{document}` w wybranym miejscu dokumentu. Jest to szczególnie przydatne w przypadkach, gdy błędy te są trudne do zidentyfikowania (gdy wygenerowane przez kompilator numery linii z błędami nie są tymi, w których błędy występują). Wystarczy wtedy przestawić wspomnianą linijkę do kolejnych miejsc, aż znajduję to miejsce, gdzie występuje problem.

Aby osiągnąć apostrofy maszynowe (czyli takie złożone z samych kresek) należy użyć polecenia `"{}jak tutaj{}"` (podwójny apostrof i podwójny apostrof z na wszelki wypadek umieszczonymi nawiasami klamrowymi, nawiasy są potrzebne z tej racji, iż podwójny apostrof przed niektórymi literkami zamienia je na literki z akcentami). W efekcie otrzymamy "jak tutaj". Jeśli natomiast apostrofy mają być drukarskie (czyli złożone z kropek i kresek), to należy użyć polecenia `„,jak tutaj”` (dwa pojedyncze przecinki i dwa pojedyncze apostrofy). W efekcie otrzymamy „jak tutaj”. Można też użyć znaków apostrofów odpowiednio zakodowanych „jak tutaj”, tylko że czasem trudno pisać się takie apostrofy w środowiskach kompilacji projektów latexowych.

Oto sposoby ustawienia odstępów między liniami:

- używając komendy `\linespread{...}` (akceptowalne), przy czym atrybutem tej metody jest współczynnik zależny od wielkości czcionki. Dla czcionki wiodącej 12pt odstęp półtora linii osiągnie się komendą `\linespread{1.241}`. Dla innych czcionek wiodących wartości tego parametru są jak w poniższym zestawieniu.

```
10pt 1.25 dla \onehalfspacing
      1.667 for \doublespacing,
          ponieważ „basic ratio” = 1.2
          (\normalfont posiada \baselineskip rozmiaru 12pt)
11pt 1.213 dla \onehalfspacing oraz 1.618 dla \doublespacing,
      ponieważ „basic ratio” = 1.236
          (\normalfont posiada \baselineskip rozmiaru 13.6pt)
12pt 1.241 dla \onehalfspacing oraz 1.655 dla \doublespacing,
      ponieważsince „basic ratio” is 1.208
          (\normalfont has a \baselineskip of 14.5pt)
```

Kłopot w tym, że raz ustawiony odstęp będzie obowiązywał do wszystkich czcionek (nie działa tu żaden mechanizm zmiany współczynnika w zależności od wielkości czcionki akapitu).

- używając pakietu `setspace` (niezalecane). Ponieważ klasa `memoir` emuluje pakiet `setspace`, w preambule dokumentu należałoby umieścić:

```
\DisemulatePackage{setspace}
\usepackage{setspace}
```

a potem można już sterować odstęp komendami:

```
\singlespacing
\onehalfspacing
\doublespacing
```

Ten sposób pozwala na korzystanie z mechanizmu automatycznej zmiany odległości linii w zależności od wielkości czcionki danego akapitu.

- korzystając bezpośrednio z komend dostarczonych w klasie `memoir` (zalecane):

```
\SingleSpacing
\OnehalfSpacing
\DoubleSpacing
```

Ten sposób również pozwala na korzystanie z mechanizmu automatycznej zmiany odległości linii w zależności od wielkości czcionki danego akapitu.

Na koniec jeszcze uwaga o rozmiarze pliku wynikowego. Otóż `pdflatex` generuje pliki pdf, które zazwyczaj mogłyby być nieco lepiej skompresowane. Do lepszego skompresowania tych plików można użyć programu `ghostscript`. Wystarczy w tym celu wydać komendę (pod windowsami):

```
gswin64 -sDEVICE=pdfwrite -dCompatibilityLevel=1.4 -dNOPAUSE -dQUIET -dBATCH
-sOutputFile=Dyplom-compressed.pdf Dyplom.pdf
```

Rozdział 7

Podsumowanie

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

7.1. Sekcja poziom 1

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Nam id nulla a adipiscing tortor, dictum ut, lobortis urna. Donec non dui. Cras tempus orci ipsum, molestie quis, lacinia varius nunc, rhoncus purus, consectetur congue risus.

7.1.1. Sekcja poziom 2

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Sekcja poziom 3

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Paragraf 4 Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

7.2. Sekcja poziom 1

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Rozdział 8

Konstrukcja

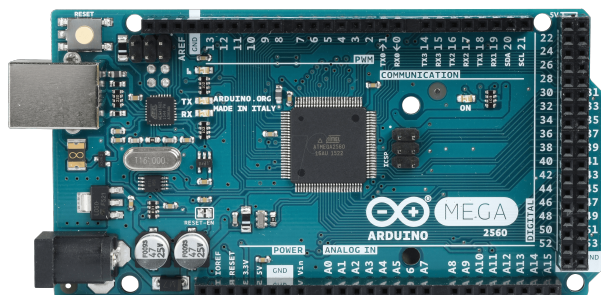
8.1. Wybór mikrokontrolera

W przypadku robota autonomicznego istotną jego częścią jest jednostka logiczna, która nim steruje. Powinna być wystarczająco wydajna aby umożliwić szybkie podejmowanie decyzji na podstawie odczytów z czujników oraz stanu wewnętrznego robota. W przypadku braku zewnętrznego sterowania przez operatora, robot sam powinien unikać kolizji oraz decydować o kierunku poruszania się.

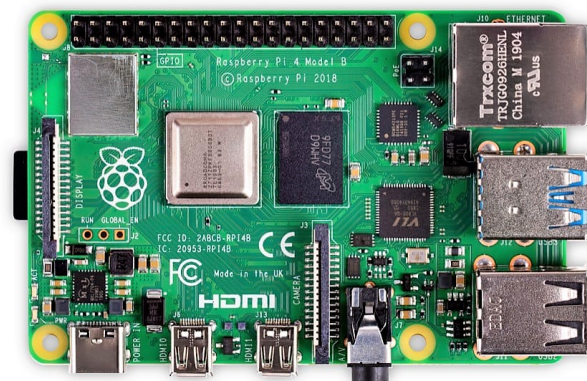
Obecnie dostępnych jest wiele rodzajów mikrokontrolerów, a konkurencyjność zapewnia niskie ceny zakupu. To sprawia, że są one w zasięgu finansowym przeciętnego człowieka. Najbardziej rozpowszechnioną platformą jest seria Arduino [1]. Posiada ona dużą ilość użytkowników i dzięki temu można łatwo uzyskać wsparcie w przypadku problemu z platformą. Przykładową gotową płytką z 8-bitowym procesorem ATmega2560 jest Arduino Mega 8.1. Posiada zegar o maksymalnej częstotliwości 16Mhz, 54 piny cyfrowe (w tym 15 PWM i 6 wspierających przerwanie), 16 pinów analogowych i 6 timerów. Cena w przypadku nieoryginalnej wersji płytki wynosi około 7\$ (28zł)

Kolejnym przykładem rodziny mikrokontrolerów z dużym wsparciem użytkowników jest Raspberry Pi. Przykładową płytką jest Raspberry Pi 4 model B 8.1. Posiada 4-rdzeniowy, 64-bitowy procesor o taktowaniu GHz, od 1 do 4GB pamięci RAM oraz 40 pinów cyfrowych. Zaletą tej płytki jest jej wysoka wydajność i możliwość wgrania pełnoprawnego systemu operacyjnego. Przykładem może być system operacyjny Rasbian [4]. Jest to wersja Linuxa podobna do systemu Debian. Minusem jest cena która dla najnowszego modelu wynosi około 50\$ (200zł) natomiast dla starszej wersji około 40\$ (160zł).

a)

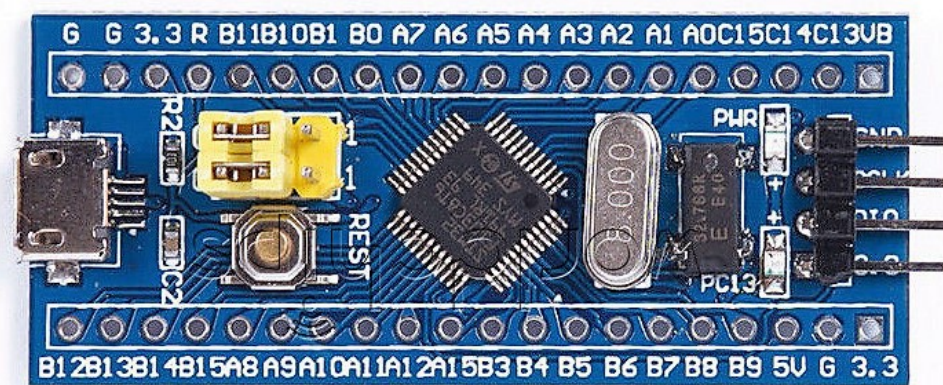


b)



Rys. 8.1: Mikrokontrolery: a) Arduino Mega, b) Raspberry Pi

Trzecią alternatywą łączącą pozytyw obu wymienionych wcześniej platform są mikrokontrolery STM32. Przykładową płytką jest STM32F103 Blue Pill 8.2. Posiada 32 konfigurowalne piny, 16 może obsługiwać zewnętrzne przerwania, 10 pinów połączonych z przetwornikiem ADC. Dodatkowo 18 może pracować z napięciem 5V (sam procesor pracuje na napięciu 3.3V) co może być przydatne zważywszy na fakt, że większa część gotowych modułów pracuje w logice 5V. Procesor wyposażony jest również w kontroler DMA, który pozwala na pomiary ADC oraz komunikację bez użycia procesora. Maksymalna nominalna częstotliwość taktowania wynosi 72MHz i dzięki pętli PLL może być łatwo konfigurowana. W przypadku niewielkiego braku mocy obliczeniowej istnieje możliwość łatwego overclockingu do 128MHz kosztem braku komunikacji przez USB. Procesor wyposażony jest także w 4 timery 16-bitowe, 4-kanalowe. Posiada także kilka możliwości wyboru API. Od wysokopoziomowego STM32duino, opartego na wspomnianym wcześniej Arduino, przez HAL oraz starsze SPL kończąc na systemie czasu rzeczywistego FreeRTOS [2]. Ogromnym plusem jest niska cena. Płytkę kosztuje około 1.5\$ (6zł).



Rys. 8.2: Mikrokontroler STM32F103 Blue Pill

Uwzględniając wskazane powyżej informacje wybrano do projektu płytkę STM32F103 Blue Pill.

Rozdział 9

Wstęp teoretyczny

Robot mobilny - robot, który potrafi zmieniać swoje położenie w przestrzeni. Może być robotem autonomicznym, czyli takim który realizując swoje zadanie porusza się bezkolizyjnie w wyznaczonym środowisku oraz robi to bez ingerencji operatora. Roboty mobilne można podzielić na kategorie przedstawione w tabeli 9.1.

Tab. 9.1: Kategorie robotów mobilnych

| Narzędzie | Wersja | Opis | Adres |
|--------------|--------|---|---|
| MiKTeX | 2.9 | Zalecana jest instalacja Basic MiKTeX z dystrybucji 32 lub 64 bitowej. Brakujące pakiety będą się doinstalowywać podczas kompilacji projektu. | http://miktex.org/download |
| TexnicCenter | 2.02 | Można pobrać 32 lub 64 bitową wersję | http://www.texniccenter.org/download/ |
| SumatraPDF | 3.1.1 | Można pobrać 32 lub 64 bitową wersję | http://www.sumatrapdfreader.org/download-free-pdf-viewer.html |
| JabRef | 3.3 | Można pobrać 32 lub 64 bitową wersję | http://www.fosshub.com/JabRef.html |

Literatura

- [1] Arduino framework. <https://www.arduino.cc/>.
- [2] Freertos. <https://www.freertos.org>.
- [3] Strona polskiego ministerstwa cyfryzacji. <https://www.gov.pl/web/5g>.
- [4] System operacyjny raspbian. <https://www.raspberrypi.org/downloads/raspbian/>.

Dodatek A

Tytuł dodatku

Zasady przyznawania stopnia naukowego doktora i doktora habilitowanego w Polsce określa ustawa z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz.U. nr 65 z 2003 r., poz. 595 (Dz. U. z 2003 r. Nr 65, poz. 595)). Poprzednie polskie uregulowania nie wymagały bezwzględnie posiadania przez kandydata tytułu zawodowego magistra lub równorzędnego (choć zasada ta zazwyczaj była przestrzegana) i zdarzały się nadzwyczajne przypadki nadawania stopnia naukowego doktora osobom bez studiów wyższych, np. słynnemu matematykowi lwowskiemu – późniejszemu profesorowi Stefanowi Banachowi.

W innych krajach również zazwyczaj do przyznania stopnia naukowego doktora potrzebny jest dyplom ukończenia uczelni wyższej, ale nie wszędzie.

Dodatek B

Opis załączonej płyty CD/DVD

Tutaj jest miejsce na zamieszczenie opisu zawartości załączonej płyty. Należy wymienić, co zawiera.