|  |  |
| --- | --- |
| Univerzita Pavla Jozefa Šafárika v Košiciach  Prírodovedecká fakultaNázov fakultyNázov vysokej školy | |
| Komponentové a udalosťami riadené programovanie zariadení na platforme Arduino | |
|  | |
|  | |
| 2018 | Bc. Patrik Pekarčík |

|  |  |
| --- | --- |
| Univerzita Pavla Jozefa Šafárika v Košiciach  Prírodovedecká fakultaPrírodovedecká fakulta | |
| Komponentové a udalosťami riadené programovanie zariadení na platforme ArduinoKomponentové a udalosťami riadené programovanie zariadení na platforme Arduino | |
| **Diplomová práca**Diplomová práca | |
| Študijný program: | InformatikaInformatika |
| Pracovisko (katedra/ústav): | Ústav InformatikyÚstav Informatiky |
| Vedúci diplomovej práce: | RNDr. František Galčík, PhD.RNDr. František Galčík, PhD. |
| Konzultant diplomovej práce: (nepovinný) | Titul Meno Priezvisko, HodnosťTitul Meno Priezvisko, Hodnosť |
|  |  |
| KošiceKošice 20182018 | Bc.Bc. PatrikPatrik PekarčíkPekarčík |

**Zadanie záverečnej práce**

Zadanie záverečnej práce (ďalej len „zadanie“) je dokument, ktorým vysoká škola stanoví študentovi študijné povinnosti v súvislosti s vypracovaním záverečnej práce. Zadanie spravidla obsahuje: typ záverečnej práce, názov záverečnej práce, meno, priezvisko a tituly študenta, meno, priezvisko a tituly školiteľa, v prípade externého školiteľa meno, priezvisko a tituly konzultanta, školiace pracovisko, meno, priezvisko a tituly vedúceho pracoviska, anotáciu záverečnej práce, jazyk, v ktorom sa práca vypracuje, dátum schválenia zadania.

**Poďakovanie (nepovinné)**

Na tomto mieste môže byť vyjadrenie poďakovania napr. vedúcemu práce resp. konzultantom za pripomienky a odbornú pomoc pri vypracovaní práce. Nie je zvykom ďakovať za rutinnú kontrolu, menšiu spoluprácu alebo všeobecné rady. Vyjadrenie poďakovania v prípade využitia inej práce sa uskutočňuje formou citácie na konci hlavného textu práce a odkazy na citáciu sa musia uviesť aj na zodpovedajúcich miestach v texte.

|  |
| --- |
| Abstrakt v štátnom jazyku |
| Abstrakt obsahuje informáciu o cieľoch práce, jej stručnom obsahu a v závere abstraktu sa charakterizuje splnenie cieľa, výsledky a význam celej práce. Súčasťou abstraktu je  3 - 5 kľúčových slov. Abstrakt sa píše súvisle ako jeden odsek a jeho rozsah je spravidla 100 až 500 slov. |

|  |
| --- |
| Abstrakt v cudzom jazyku |
| Text abstraktu v svetovom jazyku je potrebný pre integráciu do medzinárodných informačných systémov (napr. The Network Digital Library of Theses and Dissertations). Ak nie je možné jazykovú verziu umiestniť na jednej strane so slovenským abstraktom, je potrebné umiestniť ju na samostatnú stranu (cudzojazyčný abstrakt nemožno deliť a uvádzať na dvoch stranách). |

Obsah

[Obsah 5](#_Toc507452026)

[Zoznam ilustrácií 7](#_Toc507452027)

[Zoznam tabuliek 8](#_Toc507452028)

[Zoznam skratiek a značiek 9](#_Toc507452029)

[Slovník termínov 10](#_Toc507452030)

[Úvod 11](#_Toc507452031)

[1 Internet vecí 12](#_Toc507452032)

[1.1 Hardvér 13](#_Toc507452033)

[1.2 Softvér 13](#_Toc507452034)

[1.3 Mikrokontrolér v porovnaní so single-board počítačom 14](#_Toc507452035)

[2 Platforma Arduino 15](#_Toc507452036)

[2.1 Parametre mikrokontroléra 16](#_Toc507452037)

[2.2 Programovanie pre mikrokontrolér 17](#_Toc507452038)

[2.3 Existujúce riešenia pre platformu Arduino 17](#_Toc507452039)

[2.3.1 Arduino EventManager 17](#_Toc507452040)

[2.3.2 Quantum Leaps Modeling Tool 18](#_Toc507452041)

[2.3.3 ARTe (Arduino Real-Time extension) 18](#_Toc507452042)

[2.3.4 Cayenne 18](#_Toc507452043)

[3 Komponentové a udalosťami orientované programovanie 19](#_Toc507452044)

[3.1 Definovanie udalosti 19](#_Toc507452045)

[3.2 Definovanie komponentu 19](#_Toc507452046)

[3.3 Vzťah komponentu s udalosťou 19](#_Toc507452047)

[4 Integrované vývojové prostredie 20](#_Toc507452048)

[4.1 Rapid application development 20](#_Toc507452049)

[5 Architektúra komponentového a udalosťami orientovaného riešenia ACProg 21](#_Toc507452050)

[5.1 Project bootstrap 21](#_Toc507452051)

[5.1.1 Plánovač udalostí 21](#_Toc507452052)

[5.1.2 Trvalá pamäť EEPROM 21](#_Toc507452053)

[5.2 Komponenty 21](#_Toc507452054)

[5.3 Konfigurácia projektu 21](#_Toc507452055)

[5.4 Generátor knižnice pre Arduino 22](#_Toc507452056)

[6 IDE pre projekt ACProg 23](#_Toc507452057)

[6.1 Používateľské požiadavky 23](#_Toc507452058)

[6.1.1 Grafický návrh IDE 23](#_Toc507452059)

[6.2 Technologický návrh 23](#_Toc507452060)

[6.2.1 Framework Java Swing 23](#_Toc507452061)

[6.2.2 Docking framework 23](#_Toc507452062)

[6.3 Architektúra IDE 23](#_Toc507452063)

[6.4 Implementácia IDE 23](#_Toc507452064)

[6.4.1 Use case: Manažment komponentov 23](#_Toc507452065)

[6.4.2 Use case: Ponuka dostupných komponentov 23](#_Toc507452066)

[6.4.3 Use case: Kompilácia a spustenie projektu 24](#_Toc507452067)

[6.4.4 Use case: Syntaktická analýza kódu 24](#_Toc507452068)

[6.4.5 Use case: Volitelny UX pre programatora 24](#_Toc507452069)

[7 Vzorové komponenty pre ACProg 25](#_Toc507452070)

[7.1 Časovač 25](#_Toc507452071)

[7.2 Digitálny výstup 25](#_Toc507452072)

[7.3 Digitálny vstup 25](#_Toc507452073)

[7.4 Analógový vstup s threshold alert 25](#_Toc507452074)

[7.5 Komunikácia pomocou MQTT 25](#_Toc507452075)

[7.6 Rádiová komunikácia pomocou 433MHz 25](#_Toc507452076)

[Záver 26](#_Toc507452077)

[Resumé 27](#_Toc507452078)

[Zoznam použitej literatúry 28](#_Toc507452079)

[Prílohy 29](#_Toc507452080)

Zoznam ilustrácií

Obr. 1 Schéma reprezentujúca IoT 12

Obr. 2 Doska Arduino (vľavo) a rozširujúca doska shield (vpravo) 15

Obr. 3 Arduino IDE, v červenom kruhu je vyznačené tlačilo spustenia na doske. 16

Obr. 4 21

Zoznam tabuliek

Tab. 1 2

Tab. 2 2

Zoznam skratiek a značiek

μ **micro**, 10-6

SI **S**ystème **I**nternational

V **volt**, základná jednotka napätia v sústave SI

Slovník termínov

**Dizertácia** je rozsiahla vedecká rozprava, v ktorej sa na základe vedeckého výskumu a s použitím (využitím) bohatého dokladového materiálu ako i vedeckých metód rieši zložitý odborný problém.

**Font** je súbor, obsahujúci predpisy na zobrazenie textu v danom písme, napr. na tlačiarni. To čo vidíme je písmo; font je súbor a nevidíme ho.

**Meter** (m) je vzdialenosť, ktorú svetlo vo vákuu prejde za časový interval 1/299 792 458 sekundy.

**Proces** je postupnosť či rad časovo usporiadaných udalostí tak, že každá predchádzajúca udalosť sa zúčastňuje na determinácii nasledujúcej udalosti.

Úvod

Hlavnú textovú časť záverečnej práce tvorí: úvod, jadro, záver, resumé (povinné iba v prípade, ak je práca vypracovaná v inom ako štátnom jazyku), zoznam použitej literatúry.

V úvode autor stručne a výstižne charakterizuje stav poznania alebo praxe v oblasti, ktorá je predmetom záverečnej práce a oboznamuje čitateľa s významom, cieľmi a zámermi práce. Autor v úvode zdôrazňuje, prečo je práca dôležitá a prečo sa rozhodol spracovať danú tému.

# Internet vecí

Pojem internetu vecí (IoT) pokrýva viac ako len koncept alebo technológiu. Je to nový prístup, ktorý ovplyvňuje vývoj aktuálnych technológií, aplikácií a vízií. Kvôli rozsiahlosti ešte nemáme presnú definíciu pre IoT ako také. Viaceré vedecké články, sa zhodujú v tom, že kľúčovým prvkom IoT je prepojenie fyzického a digitálneho sveta, pomocou senzorov a aktuátorov. Okrem toho je pre IoT zariadenia dôležité vedieť komunikovať s inými zariadeniami, počítačmi či ľuďmi pomocou počítačovej siete [1], pre následné vykonávanie akcií. Nasledujúca schéma reprezentuje IoT v komunikačnej postupnosti.

obrazok

Obr.  Schéma reprezentujúca IoT

Popis schémy začneme senzormi a budeme pokračovať v smere toku dát. Senzory sú harvérové zariadenia, ktoré zberajú informácie z fyzického sveta a menia ich na digitálny signál. Signál smeruje do mikrokontroléra, ten vykonáva lokálne spracovanie. Lokálne spracovanie spočíva v prevode signálu na digitálne dáta. Okrem tohto prevodu môže vykonávať akcie a ihneď ovplyvňovať fyzický svet. Po lokálnom spracovaní sa dáta uložia na lokálne úložisko pre ich budúce analýzy. Keďže hovoríme o internete vecí, tak je na miesto rozmýšľať o online zdieľaní dát s ostatnými zariadeniami pomocou počítačovej siete. IoT server je zariadenie na sieti, ktoré prijíma dáta zo všetkých senzorov a vykonáva cloud spracovanie. Pozícia cloud spracovania je veľmi podobná lokálnemu spracovaniu. Hlavným rozdielom je množstvo spracovávaných dát. Kým pre lokálne spracovanie máme nízke výkonnostné prostriedky pri cloud spracovaní sú k dispozícii gigabajty pamäte s výkonnými procesormi, ktoré nám umožňujú vykonávať náročné spracovania prijatých dát. Všetky spracované dáta sa následne ukladajú na cloud úložisku. Schéma predstavuje aký potenciál nám poskytuje IoT. V mnohých IoT projektoch sa však stáva, že nevyužijú všetky časti v schéme.

## Hardvér

Základným stavebným prvkom lokálneho spracovania v IoT je mikrokontrolér, ktorý riadi prácu zariadenia. Na mikrokontrolér sú pripojené senzory, aktuátory a obvody podporujúce komunikáciu s ďalšími zariadeniami. Mikrokontrolér je elektronický čip obsahujúci obvody pre zavedenie a beh softvéru. Menovite to sú RAM, ROM, CPU. Avšak parametre týchto obvodov sú relatívne nízke oproti bežne dostupným počítačom. Napríklad mikrokontrolér, ktorý používajú aj dosky Arduino, ATMEGA328 disponuje 32kB ROM a 2kB RAM so 16-bitovým 20MHz CPU v cene približne dvoch dolárov.

**Senzory** sú elektronické súčiastky pomocou, ktorých mikrokontrolér získava informácie o fyzickom svete v digitálnej informácii, napr. odmeranie elektrického odporu. Komunikácia senzora s mikrokontrolérom spočíva v čítaní elektrického napätia, ktoré mikrokontrolér spracuje do digitálnej podoby a prevedie do meraných jednotiek. Napríklad pri meraní elektrického odporu by sme prijaté elektrické napätie prepočítali na digitálnu verziu v ohmoch.

**Aktuátory** sú elektronické súčiastky ovplyvňujúce fyzický svet. Zaraďujeme tu motory, svetlá, reproduktory, a pod. Tieto súčiastky sú riadené informáciami prijatými od mikrokontrolérov.

## Softvér

Na riadenie mikrokontroléra je potrebný obslužný softvér napálený v jeho pamäti ROM. Tento softvér je zavedený pri privedení elektrického prúdu do mikrokontroléra. Pomocou neho mikrokontrolér číta hodnoty vstupných pinov (zo senzorov) a riadi výstupné piny (k autátorom). Najčastejším programovacím jazykom pre mikrokontroléry je upravená verzia jazyka C. Netreba zabudnúť, že mikrokontroléry poskytujú len rádovo 2kB operačnej pamäte pre beh vytvorené programy.

Okrem samotných programov pre mikrokontrolér je dôležitý aj podporný softvér. Podporný softvér slúži programátorom na uľahčenie vývoja. Základom je kompilátor, pomocou, ktorého vieme previesť program. Integrované vývojové prostredia tiež radíme medzi podporný softvér. Jeho hlavnou úlohou je odbremeniť programátorov od opakovaných, krokov a poskytnúť jednoduchú konfiguráciu bez väčšej potreby znalosti celého systému. Integrované vývojové prostredia taktiež veľmi prispievajú k rýchlejšiemu prototypovaniu nových nápadov.

## Mikrokontrolér v porovnaní so single-board počítačom

Hlavnou návrhovou črtou mikrokontroléra je komunikácia so senzormi, aktuátormi a pod. Pre takú komunikáciu je dôležité presné časovanie prenosu signálov a tú nám mikrokontroléry priamo poskytujú. Pri single-board počítačoch je hlavnou črtou vyšší výpočtový výkon za nízku cenu. Výkon na týchto počítačoch sa pohybuje okolo 1GB RAM s ARM 1,2Ghz procesorom. Tieto parametre nám už dovoľujú spustiť na procesore odľahčený operačný systém, avšak nevýhodou operačných systémov je, že prenechávame časovanie a plánovanie na nich a preto je takmer nemožná priama stabilná komunikácia so senzormi a aktuátormi. Single-board počítače často poskytujú rôzne mikrokontroléry implementujúce komunikačné protokoly používané v elektronike (napr. I2C). Pomocou nich vieme pomocou single-board počítačov komunikovať so senzormi. Avšak počet takých senzorov na trhu je veľmi nízky.

// TODO: napisat nieco malo o fog comutingu (cloud computing v malom).

# Platforma Arduino

Projekt Arduino sa začal v roku 2003 ako program pre študentov v Interaction Design Institute Ivrea v Ivreu v Taliansku s cieľom poskytnúť novým a profesionálnym používateľom lacný a jednoduchý spôsob vytvárania zariadení, ktoré interagujú s prostredím pomocou senzorov a aktuátorov. Bežné príklady takýchto zariadení určené pre začiatočníkov zahŕňajú jednoduché roboty, termostaty a detektory pohybu.

Arduino je hardvér a softvér s otvoreným zdrojovým kódom, licencovaný licenciami GNPL Lesser General Public License (LGPL) a GNU General Public License (GPL). Licencie umožňujú výrobu dosiek Arduino a distribúciu softvéru každému. Dosky Arduino sú špecifické vyvedenými pinmi na ich okraji. Toto vyvedenie je kľúčové pre jednoduché prototypovanie embeded zariadení. Vďaka štandardizovanému tvaru a vyvedeniu pinov dosky Arduino je možné vytvárať rozširujúce dosky tzv. Arduino shield. Arduino shield poskytuje rovnaký tvar, rozloženie pinov a dodáva rôzne funkcionality, či už sú to senzory, aktuátory alebo kompletné riešenie pre embeded systém. Dosky Arduino majú vstavané sériové komunikačné rozhrania, vrátane univerzálnej sériovej zbernice (USB).

obrazok

Obr.  Doska Arduino (vľavo) a rozširujúca doska shield (vpravo)

Softvérová časť projektu Arduino spočíva v integrovanom vývojovom prostredí (IDE) a programovacom jazyku postavenom na c++. Integrované vývojové prostredie bolo postavené na projekte Language Processing. Keďže väčšina doskiek Arduino má vstavané rozhranie USB tak kompikácia s napálením projektu z IDE do zariadenia je na jedno kliknutie.

obrazok

Obr.  Arduino IDE, v červenom kruhu je vyznačené tlačilo spustenia na doske.

## Parametre mikrokontroléra

Základným prvkom dosky Arduino je mikrokontrolér. Pri originálnych doskách Arduino sa používajú mikrokontroléry Atmel ATmega. Na čipe mikrokontroléra sa nachádzajú obvody procesora (CPU), operačnej pamäte (RAM), programovej pamäte (ROM), trvalej pamäte (EEPROM). Pre komunikáciu s ostatnými zariadeniami sú z mikrokontroléra vyvedené vstupno výstupné piny. Výkonnostné hodnoty týchto parametrov sú pomerne nízke a preto je potrebné na ne brať ohľad pri návrhu softvéru. Tabuľka č. 1 zobrazuje konkrétne parametre niekoľkých modelov dosiek Arduino pre predstavu nízkych parametrov. Pre ilustráciu výkonnostného rozdielu medzi mikrokontrolérom a single-board počítačom, sme v poslednom riadku tabuľky vypísali parametre single-board počítača Raspberry PI. Pri single-board počítači si môžeme všimnúť absenciu analógových vstupno výstupných pinov.

Tab.  Porovnanie parametrov pre rôzne modely dosiek Arduino.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model Arduino** | **CPU Model** | **ROM (kB)** | **RAM (kB)** | **EEPROM (kB)** | **CPU frekvencia (MHz)** | **Analógové / digitálne piny** |
| UNO | ATmega323P | 32 | 2 | 1 | 16 | 6/14 |
| Mega 2560 | ATmega2560 | 256 | 8 | 4 | 16 | 16/54 |
| Nano | ATmega168 | 16 | 1 | 0,512 | 16 | 8/14 |
| 101 | Intel Curie | 196 | 24 | - | 32 | 6/14 |
| Gemma | ATtiny85 | 8 | 0,5 | 0,5 | 8 | 1/3 |
| Raspberry PI (singe-board) | ARMv8 64bit quad-core | SD slot | 512 MB | SD slot | 1200 | 0/40 |

## Programovanie pre mikrokontrolér

## Existujúce riešenia pre platformu Arduino

Stále rastúca komunita sa venuje zariadeniam Arduino. S tým je spojený aj vývoj rôznych frameworkov pre túto platformu, ktorých cieľom je zjednodušiť vývoj programátorom. Riešenia aké sme našli vyhľadávaním na fórach Arduino komunity, rozdeľujeme do kategórií online a offline riešení.

**Online** sú riešenia, ktoré do zariadenia nainštalujú zavádzač. Úlohou zavádzača je preposielať všetky údaje zo senzorov na sieť, kde bude vykonané cloud spracovanie. Vo všetkých analyzovaných online riešeniach bola úplne vynechaná časť lokálneho spracovania a uloženie dát.

**Offline** sú riešenia bežiace priamo na Arduino zariadení. Tieto riešenia využívajú najmä lokálne spracovanie s lokálnym uložením.

V nasledujúcich sekciách priblížime vybrané existujúce riešenia, ktoré sa svojou povanou najviac približujú nami navrhnutému riešeniu pre programovanie arduino zariadení.

### Arduino EventManager

Knižnica Arduino EventManager[12] patrí do skupiny offline riešení. Pri využití tejto knižnice sa periodicky vykonávaná funkcia loop() nevytvára. Namiesto nej v inicializačnej funkcii setup() zaregistrujeme obslužné funkcie (angl. callback) a programujeme spracovanie udalostí. Pri registrovaní funkcie určujeme na zmenu akého pinu má byť udalosť vyvolaná.

### Quantum Leaps Modeling Tool

Modelovací nástroj Quantum Leaps Modeling Tool zaradzujeme do skupiny offline riešení. Aplikácie v tomto nástroji modelujeme pomocou stavového automatu. Konečno stavový automat je matematický model výpočtu. Je to abstraktný stroj, ktorý môže byť v danom čase presne v jednom stave z konečného počtu stavov. Konečno stavový automat sa môže zmeniť z jedného stavu na druhý v reakcii na niektoré externé vstupy.

### ARTe (Arduino Real-Time extension)

### Cayenne

Aplikácia Cayenne[14] zaradzujeme do kategórie online riešení. Na dosku arduino je nahraná špeciálna aplikácia. Táto aplikácia automatizovane odosiela na servery Cayenne svoj aktuálny stav vstupných pinov. Doska taktiež čaká na inštrukcie zo serverov Cayenne pre nastavenie výstupných pinov. Vo webovom nástroji Cayenne vidíme reálny stav vstupných pinov, dokážeme nastavovať rôzne akcie pre zopnutie výstupných pinov. Príkladom akcie môže byť nasledovné: Ak stúpne teplota (teplotný senzor je na pine 5) nad 30C, tak zapni klimatizáciu (zapni 5V na pine 4).

# Komponentové a udalosťami orientované programovanie

## Definovanie udalosti

## Definovanie komponentu

## Vzťah komponentu s udalosťou

# Integrované vývojové prostredie

## Rapid application development

# Architektúra komponentového a udalosťami orientovaného riešenia ACProg

obrazok

Obr.

## Project bootstrap

### Plánovač udalostí

### Trvalá pamäť EEPROM

## Komponenty

## Konfigurácia projektu

## Generátor knižnice pre Arduino

Pri snahe zachovať syntax a princípy programovania Arduino zariadení, sme sa rozhodli nevytvoriť úplne nový programovací jazyk. Vybrali sme sa smerom vytvorenia Arduino knižnice podľa popisu komponentov. Toto vytvorenie sa nazýva generovanie zdrojového kódu. Generátor dostáva na vstupe zoznam komponentov, kde komponenty majú určený typ (modul). Generátor pridá do výslednej knižnice zdrojové kódy abstrakcie modulu, ak tam ešte nie sú, a v hlavnej časti behu programu vytvorí jeho inštanciu. Inštancie komponentov budú globálne a používateľ v nich bude môcť programátorsky meniť konfigurácie za behu. Okrem komponentov generátor vytvorí aj tzv. jadro nášho riešenia. Jadro spočíva v inicializácii komponentov a následnom plánovaní vykonávania jednotlivých udalostí. Jadro poskytuje programátorovi aj aplikačné rozhranie na komunikáciu so stálou pamäťou EEPROM.

# IDE pre projekt ACProg

## Používateľské požiadavky

### Grafický návrh IDE

## Technologický návrh

### Framework Java Swing

### Docking framework

## Architektúra IDE

Rozdelenie namespacov. Nacrt hlavnych okien a prechody medzi nimi. Popis vlastnosti a spojenia Generatora s IDE.

## Implementácia IDE

### Use case: Manažment komponentov

Spomenut skupinovy zoznam komponentov s presuvanim medzi skupinami. Spomenut aj property editor

### Use case: Ponuka dostupných komponentov

Popis nacitania vsetkych dostupnych komponentov a implementacia vizualnej stranky.

### Use case: Kompilácia a spustenie projektu

Popis prepojenia na generátor, nasledne spustenie kompilacie cez arduino. Pre vystup spomenut konzolu v gui.

### Use case: Syntaktická analýza kódu

Popis vsetkeho co bude hotove

### Use case: Volitelny UX pre programatora

# Vzorové komponenty pre ACProg

## Časovač

## Digitálny výstup

## Digitálny vstup

## Analógový vstup s threshold alert

## Komunikácia pomocou MQTT

## Rádiová komunikácia pomocou 433MHz

Záver

V závere je potrebné v stručnosti zhrnúť dosiahnuté výsledky vo vzťahu k stanoveným cieľom.

Resumé

Ak je záverečná práca napísaná v cudzom jazyku, musí obsahovať resumé v slovenskom jazyku v rozsahu spravidla 10 % rozsahu záverečnej práce.

Zoznam použitej literatúry

|  |  |
| --- | --- |
| [1] | Giusto, D.; Iera, A.; Morabito, G.; Atzori, L.;, The Internet of Things, 20th Tyrrhenian Workshop on Digital Communications, Springer, New York, NY, 2010. |

Prílohy

1. CD médium – diplomová práca v elektronickej podobe, prílohy v elektronickej podobe.
2. Používateľská príručka
3. Systémová príručka

Táto časť diplomovej práce je povinná a obsahuje zoznam všetkých príloh vrátané elektronických nosičov. Názvy príloh v zozname musia byt’ zhodné s názvami uvedenými na príslušných prílohách. Tlačené prílohy majú na prvej strane identifikačné údaje – informácie zhodné s titulnou stranou diplomovej práce doplnené o názov príslušnej prílohy (Systémová príručka, Používateľská príručka). Identifikačné údaje sú aj na priložených diskoch alebo disketách. Ak je médií viac, sú označené aj číselne v tvare I/N, kde I je poradové číslo a N je celkový počet daných médií.

Každá príloha začína na novej strane a je označená samostatným písmenom alebo číslom (Príloha A, Príloha B, ... alebo Príloha 1, Príloha 2, ...). Číslovanie strán príloh nadväzuje na číslovanie strán v hlavnom texte.