



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# Vytvoření výukové aplikace řešící blokové diagramy bezporuchovosti (RBD)

## Ročníkový projekt

*Studijní program:* B2646 – Informační technologie  
*Studijní obor:* 1802R007 – Informační technologie  
*Autor práce:* **Jan Špecián**  
*Vedoucí práce:* Ing. Josef Chudoba, Ph.D.



Tento list nahradte  
originálem zadání.

## Prohlášení

Byl jsem seznámen s tím, že na můj ročníkový projekt se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasaahuje do mých autorských práv užitím mého ročníkového projektu pro vnitřní potřebu TUL.

Užiji-li ročníkový projekt nebo poskytnu-li licenci k jeho využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Ročníkový projekt jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mého ročníkového projektu a konzultantem.

Současně čestně prohlašuji, že texty tištěné verze práce a elektronické verze práce vložené do IS STAG se shodují.

6. 5. 2019

Jan Špecián

# Vytvoření výukové aplikace řešící blokové diagramy bezporuchovosti (RBD)

## Abstrakt

Práce je zaměřena na tvorbu desktopové aplikace pro tvorbu RBD diagramů a spojených výpočtů a vizualizací.

**Klíčová slova:** RBD

## Poděkování

Tímto bych rád poděkoval Ing. Josef Chudobovi, Ph.D. za věnovaný čas v konzultacích a odborné vedení plné trpělivosti a s tím spojené nabyté zkušenosti.

# Obsah

Seznam zkratek . . . . .	8
<b>1 Přehled existujících softwarových nástrojů</b>	<b>10</b>
<b>2 Teoretický úvod</b>	<b>11</b>
2.1 Distribuční funkce . . . . .	11
2.2 Exponenciální rozdělení . . . . .	11
2.3 Spolehlivost a střední doba mezi poruchami . . . . .	11
2.4 Analýza blokového diagramu bezporuchovosti (RBD) . . . . .	11
2.5 Základní zapojení bloků . . . . .	12
<b>3 Návrh desktopové aplikace .NET</b>	<b>13</b>
3.1 Objektová struktura . . . . .	13
3.2 Pomocné třídy . . . . .	13
<b>4 Průběh vývoje</b>	<b>14</b>
4.1 Rozdělení projektu na subprojekty . . . . .	14
4.2 . . . . .	14
<b>5 Testování</b>	<b>15</b>
<b>6 Návod k použití</b>	<b>16</b>
<b>7 Závěr</b>	<b>17</b>

## Seznam obrázků

2.1	Příklad sériového zapojení komponent . . . . .	12
2.2	Příklad paralelního zapojení komponent . . . . .	12

## Seznam zkratek

**JSON** JavaScript Object Notation  
**LINQ** Language Integrated Query



# Úvod

U každého systému je velmi důležitá jeho funkční spolehlivost během doby jeho životnosti. Každý systém, pokud má existovat a fungovat co nejdéle a přitom bez závad, nebo alespoň s jejich co nejmenším počtem, musí splňovat jednu zásadní vlastnost, a tou je spolehlivost. Požadavek na dostatečně velkou a často až maximální spolehlivost námi užívaných systémů má tudíž zcela zásadní význam z hlediska bezpečnostního, ekonomického i ekologického.

Cílem ročníkového projektu je navrhnout a implementovat desktopovou aplikaci pro tvorbu a jednoduchou vizualizaci RBD diagramů a výpočet parametrů spolehlivosti. Zobrazit střední dobu do poruchy pro každý blok a poskytnout možnost vizualizace distribuční funkce pro každý blok v kombinaci sériového a paralelního zapojení bloků.

# 1 Přehled existujících softwarových nástrojů

## 2 Teoretický úvod

### 2.1 Distribuční funkce

### 2.2 Exponenciální rozdělení

### 2.3 Spolehlivost a střední doba mezi poruchami

#### **Střední doba mezi poruchami**

Základní veličinou pro měření spolehlivosti systému je střední doba mezi poruchami (MTBF, Mean Time Between Failure). Obvykle je udávána v hodinách. Čím vyšší je hodnota MTBF, tím vyšší je spolehlivost produktu.[3] Je statistická veličina používaná ke kvantifikaci spolehlivosti součásti, či celého výrobku. Určuje se pro výrobek nebo zařízení, které se opravuje. [3]

#### **Spolehlivost**

Spolehlivost je schopnost systému nebo součásti vykonávat požadované funkce za daných podmínek po určené časové období [4]

$$Spolehlivost = e^{-\left(\frac{Time}{MTBF}\right)}$$

### 2.4 Analýza blokového diagramu bezporuchovosti (RBD)

Analýza blokového diagramu bezporuchovosti (RBD - Reliability Block Diagram) je metoda analýzy systému. Diagram RBD je grafická reprezentace logické struktury systému v podobě podsystémů a/nebo součástí. To umožňuje, aby byly cesty úspěchu (funkceschopného stavu) reprezentovány tak, jak jsou bloky (podsystémy/součásti) logicky propojeny.[1]

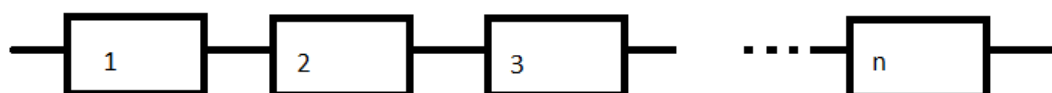
Blokové diagramy jsou mezi prvními úkoly dokončenými během etapy vymezení produktu. Mají být vypracovány jako součást vývoje počáteční koncepce. Práce na nich mají být zahájeny, jakmile existuje vymezení programu, a mají být dokončeny jako součást analýzy požadavků a mají se neustále rozšiřovat do větších úrovní

podrobnosti, jakmile budou k dispozici data, aby bylo možné činit rozhodnutí a provádět optimalizace nákladů a přínosů.[2]

## 2.5 Základní zapojení bloků

### Sériové zapojení

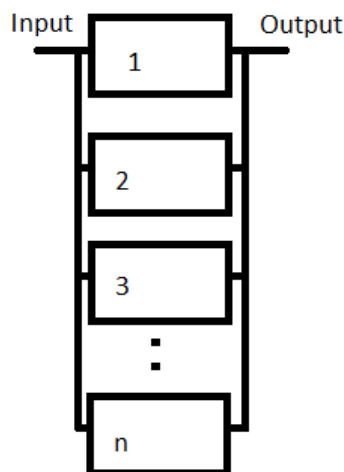
Při poruše jedné komponenty dojde k poruše celého systému. Systém je v bezporuchovém stavu, pokud všechny jeho komponenty nemají poruchu.[5]



Obrázek 2.1: Příklad sériového zapojení komponent

### Paralelní zapojení

K poruše celého systému dochází pokud jsou v poruše všechny jeho komponenty. Bezporuchový stav trvá, dokud je alespoň jedna komponenta v bezporuchovém stavu. Z hlediska odhadu pravděpodobnosti představuje paralelní systém nejlepší variantu pro odhad pravděpodobnosti bezporuchového stavu.[5]



Obrázek 2.2: Příklad paralelního zapojení komponent

## **3 Návrh desktopové aplikace .NET**

### **3.1 Objektová struktura**

### **3.2 Pomocné třídy**

## 4 Průběh vývoje

### 4.1 Rozdělení projektu na subprojekty

### 4.2

## 5 Testování

Pro testování funkčních bloků byla použita výchozí knihovna pro Unit testování v prostředí .NET pro desktopové aplikace MSTest. Za pomoci testování jsem došel ke správným výsledkům za pomoci připravené konfigurace a tím jsem ušetřil práci manuálním testováním. Další nespornou výhodou testování je odhalení chyb při změně tím, že testovací metody odhalí neočekávané výsledky.

Testované byly třídy pro výpočet distribuční funkce.

## 6 Návod k použití

Pro spuštění aplikace pro vývoj je potřeba mít nainstalované Visual Studio 2017 a novější. V přiloženém CD ve složce SpecianPRJ spusťte soubor SpecianPRJ.sln. Pro standartní spuštění aplikace stačí otevřít soubor s příponou .exe.

Pro obě varianty spuštění je nutným předpokladem nainstalovaný plný .NET Framework 4.6.1 a novější.

### Založení nového diagramu

### Uložení a otevření nového diagramu

### Přidání prvku

### Výpočty



## 7 Závěr

## Literatura

- [1] 28.6.2007, Prof. Ing. Václav Legát, DrSc., Zdroj: Verlag Dashöfer
- [2]
- [3] <http://gabben.wbs.cz/mtbf1.pdf>
- [4] IEEE 90
- [5]