

## 1. Úvod

Tento dokument popisuje objektovou strukturu modulárního emulátoru architektury RISC-V. Kapitoly obsahují názorné diagramy a popis nejdůležitějších částí aplikace, zároveň zde také uvádíme argumenty, proč byl určitý postup nebo návrhový vzor zvolen.

Informace uvedené v tomto dokumentu čtenáři usnadní případnou úpravu, nebo rozšíření aplikace.

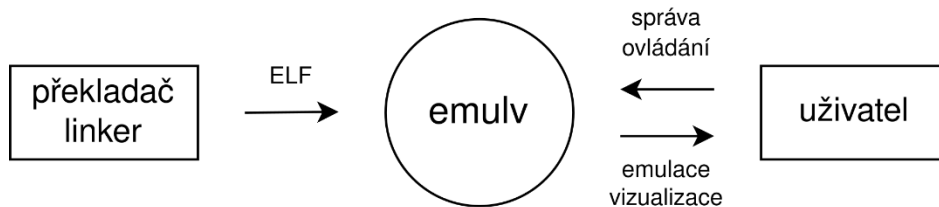
### 1.1 Účel systému

Dodávaný software je modulární emulátor platformy RISC-V s grafickým uživatelským rozhraním, který umožní studentům nahradit práci s fyzickým zařízením používaným a dostupným v omezeném počtu při výuce.

## 2. Kontext a architektura systému

### 2.1 Kontext systému

Následující diagram znázorňuje kontext tohoto systému:

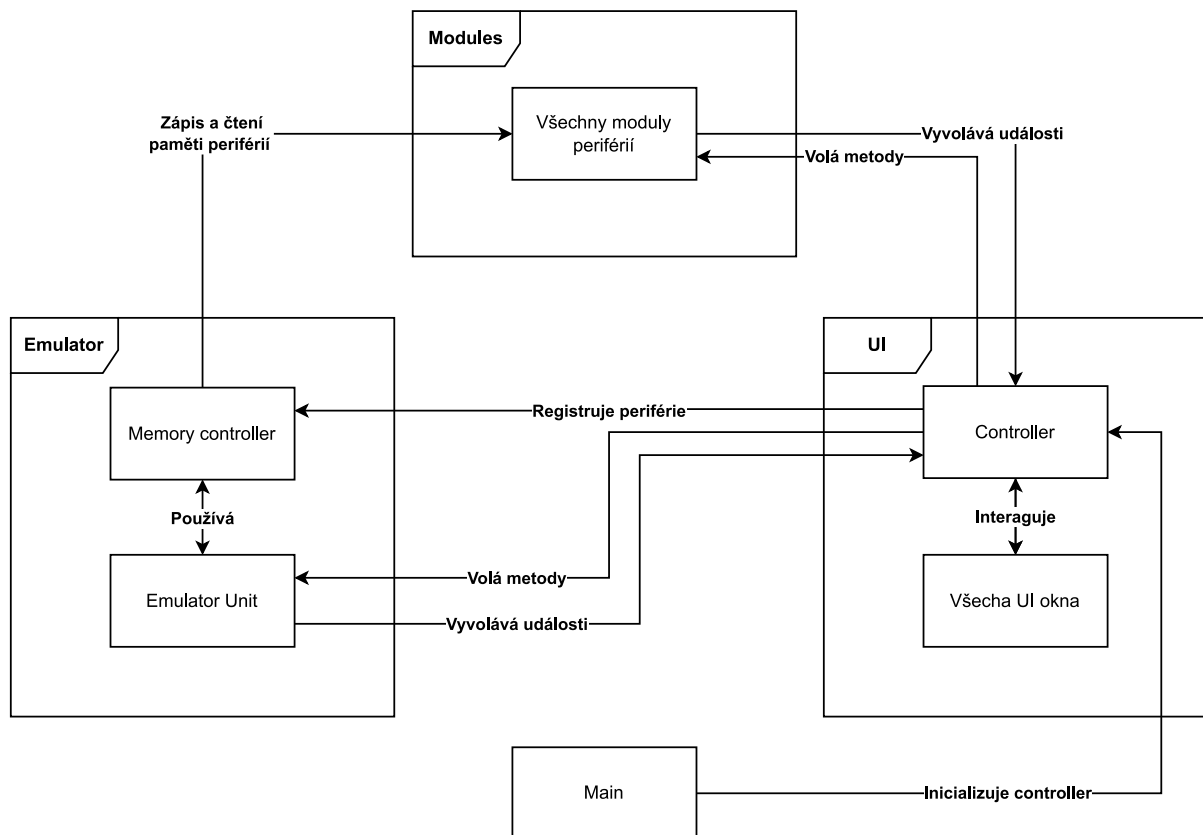


### 2.2 Architektura systému

Vysokoúrovňový popis architektury této aplikace je znázorněn pomocí následujícího diagramu komponent.

Aplikace se skládá ze tří hlavních balíčků. S balíčkem UI je komunikováno výhradně pomocí **event emitteru** (observer pattern), výhoda tohoto postupu je kompletní odstínění tříd emulátoru a modulů periférií od tříd uživatelského rozhraní. UI controller pouze obsluhuje jednotlivé události na základě kterých uživatelské rozhraní aktualizuje.

Celkově je aplikace tedy „dvouvrstvá“ architektura, skládající se z **frontendu** a **backendu**, kde balíček UI tvoří frontend, a backend je reprezentován balíčky Emulator, Modules.



## 2.3 Zvolená technologie, programovací jazyk ad., důvody

Celá aplikace je naprogramována v jazyku C++. Zvolené knihovny jsou následující:

- **Libriscv**
  - zajišťuje emulaci instrukční sady RV-32G
  - Dostupné z: <https://github.com/fwsGonzo/libriscv>
- **RISC-V Disassembler**
  - Zajišťuje disassembly ELF souboru, tento výstup je následně zobrazen v GUI
  - Dostupné z: <https://github.com/michaeljclark/riscv-disassembler>
- **Nlohmann/json**
  - Knihovna pro parsování konfiguračního JSON souboru
  - Dostupné z: <https://github.com/nlohmann/json>
- **cxx-eventemitter**
  - Knihovna poskytující jednoduchý event emitter pro C++.
  - Dostupné z: <https://github.com/heapwolf/cxx-eventemitter>

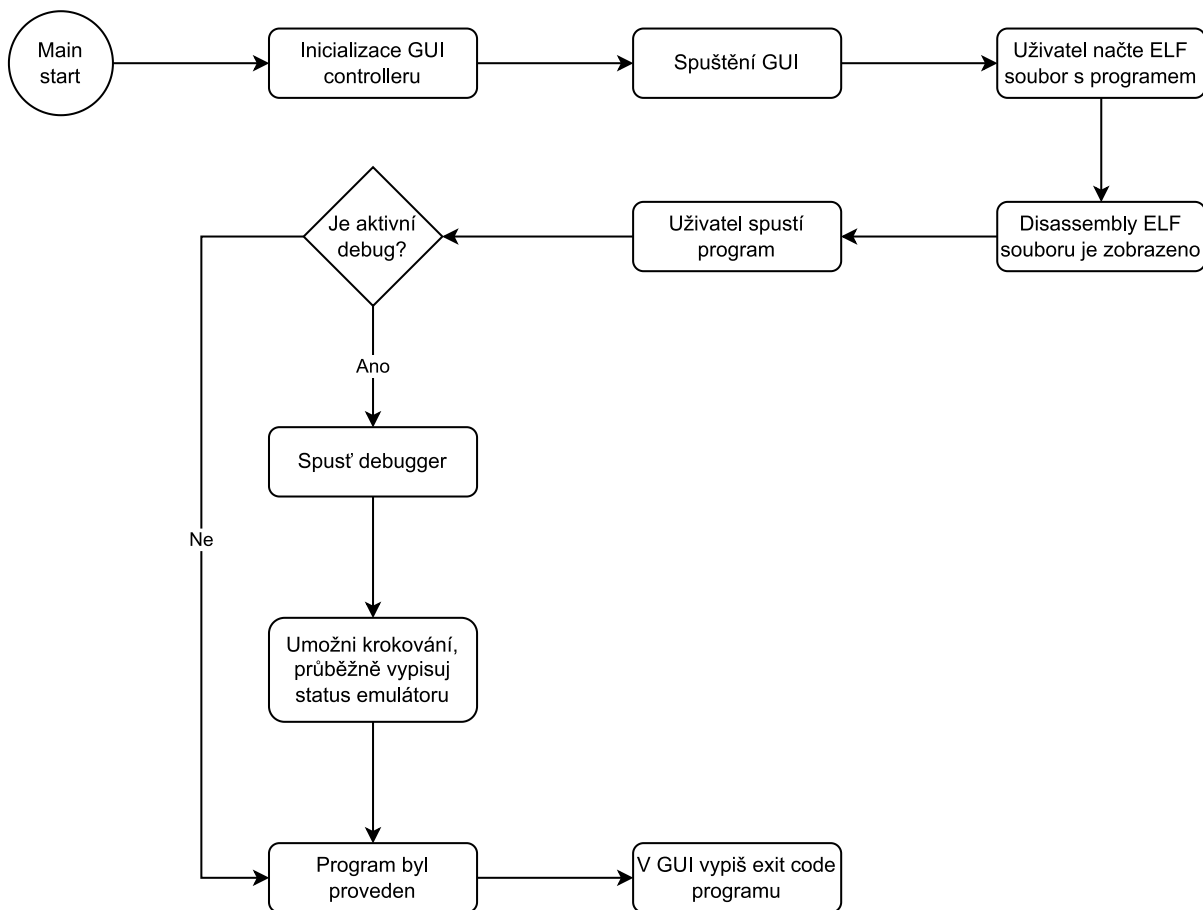
### 3. Typy informací zpracovávané systémem

Pro použití aplikace jsou potřeba následující soubory:

- **Vstupní soubor – Spustitelný program ve formátu ELF**
  - Program v instrukční sadě RV32G, je poskytován uživatelem
- **Konfigurační soubor – Ve formátu JSON**
  - Uživatel má možnost tento soubor upravit, umožňuje nastavit chování periférií a vlastnosti emulátoru

### 4. Návrh systému

Následující flow diagram popisuje hlavní činnost navrhnutého systému:



#### 4.1 Přehled důležitých komponent/tříd/modulů

Tato kapitola popisuje nejdůležitější třídy a komponenty této aplikace.

##### 4.1.1 Balíček Emulator

Představuje jakousi „procesorovou jednotku“, obsahuje třídu **EmulatorUnit.cpp**, která pokrývá veškerou činnost spojenou se spuštěním emulovaného programu. Nejdůležitější metody jsou zde **Execute()** a **Debug()**, obě slouží ke spuštění programu, metoda **Debug()** umožňuje program spustit v debug režimu, který umožňuje krokovat programem.

### 4.1.2 Balíček Modules

Obsahuje všechny moduly periférií, jako například GPIO, nebo UART.

Klíčové je zde rozhraní **PeripheralDevice.h**, pomocí kterého je možné naprogramovat další moduly. Poskytuje sadu metod, které by měly implementovat všechna periferní zařízení.

### 4.1.3 Balíček UI

Všechny soubory spojené s knihovnou Qt jsou umístěny v tomto balíčku. Snaha byla tento balíček udělat co nejméně závislý na ostatních částech aplikace (třeba pomocí event emitteru, viz předchozí kapitoly).

Nejdůležitější třídou tohoto balíčku je třída **Controller.cpp**, do které je centralizována obsluha komponent grafického rozhraní.