

# Sada realizovaných testů

Aplikace je testována kombinací jednotkových testů, monkey testing, automatizovaného testování GUI podle scénářů a spouštěním vlastních testovacích programů v emulátoru.

Periferní zařízení jsou testovány jednotkovými testy a vlastními testovacími programy a grafické uživatelské rozhraní aplikace je testováno pomocí monkey testing, automatizovanými testy GUI podle scénářů a vlastními testovacími programy.

Veškeré testy lze nalézt v repozitáři v adresáři **test**.

## Jednotkové testy

### Periferní zařízení GPIO

Periferní zařízení GPIO je testováno 29 jednotkovými testy ověřující správnou funkčnost periferie dle specifikace<sup>1</sup> výrobce fyzického zařízení.

Testy pokrývají veškerou funkčnost periferie GPIO, tedy konfiguraci pinů, nastavování výstupní hodnoty pinů a získávání vstupní hodnoty pinů.

Testují se resetované hodnoty registrů, hodnoty registrů po konfiguraci pinů, hodnoty registrů úrovně pinů a návratové hodnoty funkcí vracející stav konfigurace pinu a úroveň pinu.

Názvy testů vystihují to, jakou vlastnost či funkcionalitu testují.

### Periferní zařízení UART

Periferní zařízení UART je testováno 14 jednotkovými testy ověřující správnou funkčnost periferie dle specifikace<sup>1</sup> výrobce fyzického zařízení.

Testy pokrývají veškerou funkčnost periferie UART, tedy konfiguraci, příjem a odesílání zpráv.

Testují se resetované hodnoty registrů, hodnoty registrů po konfiguraci, během příjmu a odesílání zprávy.

Názvy testů vystihují to, jakou vlastnost či funkcionalitu testují.

## Vlastní testovací programy

Byly napsány testovací programy v jazyce C ověřující funkčnost periférií GPIO a UART. V programech se napřímo zapisuje do registrů periférií mapovaných v paměti a také se z nich čte.

---

<sup>1</sup> <https://dl.sipeed.com/LONGAN/Nano/DOC/>

## Periferní zařízení GPIO

Čtením z registrů periferie GPIO a zapisováním do nich se konfiguruje piny portů, nastavují se výstupní hodnoty pinů a čte se vstupní hodnota pinu. Ověří se tak veškerá potřebná práce s periferií.

## Periferní zařízení UART

Existují dva testovací programy pro periferii UART, jeden pro příjem zpráv a druhý pro odesílání zpráv.

Zápisem do registrů periferie UART se UART nakonfiguruje jako vysílač s délkou slova 8 bitů a poté se ve smyčce odesílají 4 byty zápisem do datového registru UART, když je UART připraven, což se zjišťuje čtením příznaku v registru UART.

Zápisem do registrů periferie UART se UART nakonfiguruje jako přijímač s délkou slova 8 bitů a poté čeká, až lze číst data, ověřováním příznaku v registru UART a poté data přečte z datového registru UART a hodnotu vrátí jako návratovou hodnotu programu.

## Monkey testing

Během vývoje a na konci vývoje byla aplikace testována pomocí monkey testing, tedy pokusným testováním aplikace s cílem najít chybové chování aplikace.

## Automatizované testy GUI

Pro rychlé ověření správného chování grafického rozhraní byly vytvořeny dva automatizované jednoduché testovací scénáře (TS).

Jednotlivé testovací kroky (TC) nejsou nezávislé, je využíváno toho, že jsou vykonávány sekvenčně.

### TS\_01\_01

Rychlý *smoke test* pro otestování **Happy Day Scenaria**.

Předpoklady ... Aplikace je spuštěna, hlavní okno je viditelné

TC_01_01_01	
Krok	Očekávaný výsledek
Kliknout na <i>File &gt; Open</i>	Zobrazí se dialog výběru souboru
Vybrat soubor <i>memory_experiment.elf</i>	Soubor se úspěšně načte
Kliknout na tlačítko <i>Run</i>	Emulace se spustí a dokončí
	V registrech se zobrazí výpis
Nastavit hodnotu koncové adresy výpisu paměti na 0x2000002F	Hodnota 0x2000002F je nastavena
Kliknout na tlačítko pro potvrzení rozsahu paměti, poté na radiobutton <i>Dec</i>	Ve výpisu paměti se zobrazí tři řádky
	V řádku 20000010 má byte 0 hodnotu <b>111</b> a na odpovídající pozici v ASCII sloupci je znak <b>o</b> .
TC_01_01_02	
Krok	Očekávaný výsledek
Kliknout na <i>Tools &gt; Select configuration</i>	Zobrazí se dialog výběru souboru
Vybrat soubor <i>config.json</i>	Objeví se periferie <i>GPIO</i> a <i>UART1</i>
Kliknout na <i>File &gt; Open</i>	Zobrazí se dialog výběru souboru
Vybrat soubor <i>uart_test_transmit.elf</i>	Soubor se úspěšně načte
Kliknout na tlačítko <i>Run</i>	Emulace se spustí a dokončí
	V záložce <i>UART1</i> se ve výpisovém okně objeví <b>aaaa</b>

## TS\_02\_01

Rychlý *smoke test* pro otestování **viditelnosti a povolení ovládání prvků**.

Předpoklady ... Aplikace je spuštěna, hlavní okno je viditelné

TC_02_01_01	
Krok	Očekávaný výsledek
Žádný	Výpisy registrů a paměti <b>nejsou ovladatelné</b>
	Periferie <b>nejsou viditelné</b>
TC_02_01_02	
Krok	Očekávaný výsledek
Žádný	Tlačítka <i>Run</i> a <i>Debug</i> jsou <b>viditelná a neovladatelná</b>
	Tlačítka <i>Terminate</i> , <i>Step</i> a <i>Continue</i> <b>nejsou viditelná</b>
TC_02_01_03	
Krok	Očekávaný výsledek
Kliknout na <i>File &gt; Open</i>	Zobrazí se dialog výběru souboru
Vybrat soubor <i>gpio_test.elf</i>	Soubor se úspěšně načte
	Tlačítka <i>Run</i> a <i>Debug</i> jsou <b>ovladatelná</b>
TC_02_01_04	
Krok	Očekávaný výsledek
Kliknout na <i>Tools &gt; Select configuration</i>	Zobrazí se dialog výběru souboru
Vybrat soubor <i>config.json</i>	Objeví se periferie <i>GPIO</i> a <i>UART1</i> , prvky v nich jsou <b>neovladatelné</b>
TC_02_01_05	
Krok	Očekávaný výsledek
Kliknout na tlačítko <i>Run</i>	Tlačítko <i>Terminate</i> je <b>viditelné a ovladatelné</b>
	Tlačítka <i>Step</i> a <i>Continue</i> jsou <b>viditelná a neovladatelná</b>
	Výpisy registrů a paměti jsou <b>neovladatelné</b>
	Periferie jsou <b>ovladatelné</b> .

TC_02_01_06	
Krok	Očekávaný výsledek
Přidat breakpoint na první řádek disassembly	Zobrazí se breakpoint na prvním řádku
Kliknout na tlačítko <i>Debug</i>	Emulace se spustí a zastaví na prvním řádku
	Tlačítka <i>Terminate</i> , <i>Step</i> a <i>Continue</i> jsou <b>viditelná a ovladatelná</b>
	Výpisy registrů a paměti a periferie jsou <b>ovladatelné</b>
TC_02_01_07	
Krok	Očekávaný výsledek
Kliknout na <i>File &gt; Open</i>	Zobrazí se dialog výběru souboru
Vybrat soubor <i>gpio_test.elf</i>	Soubor se úspěšně načte
Kliknout na tlačítko <i>Run</i>	Emulace se spustí
Kliknout na tlačítko <i>Terminate</i>	Emulace se ukončí
	Tlačítka <i>Run</i> a <i>Debug</i> jsou <b>ovladatelná</b>
	Tlačítka <i>Terminate</i> , <i>Step</i> a <i>Continue</i> <b>nejsou viditelná</b>
	Výpisy registrů a paměti jsou <b>ovladatelné</b>
	Periferie jsou <b>viditelné a neovladatelné</b>
TC_02_01_08	
Krok	Očekávaný výsledek
Kliknout na <i>Tools &gt; Clear configuration</i>	Periferie <b>nejsou viditelné</b>

## TS\_02\_02

Rychlý *smoke test* pro otestování **výpisu paměti**.

Předpoklady:

- Aplikace je spuštěna, hlavní okno je viditelné
- Byl vybrán soubor *memory\_experiment.elf*
- Byla spuštěna a dokončena jeho emulace
- Ve výpisu paměti je vypsán rozsah 0x20000010 až 0x2000001F

TC_02_02_01	
Krok	Očekávaný výsledek
Žádný	Ve výpisu paměti je jeden řádek.
	Byte 0 má hodnotu <b>6F</b> a na odpovídající pozici v ASCII sloupci je znak <b>o</b>
	Ostatní byty mají hodnotu <b>00</b> a na odpovídající pozici v ASCII sloupci je znak <b>tečky</b>
TC_02_02_02	
Krok	Očekávaný výsledek
Kliknout na radiobutton <i>Dec</i> u výpisu paměti	Ve výpisu paměti je jeden řádek.
	Ostatní byty mají hodnotu <b>000</b> a na odpovídající pozici v ASCII sloupci je znak <b>tečky</b>

## TS\_02\_03

Rychlý *smoke test* pro otestování **výpisu registrů**.

Předpoklady:

- Aplikace je spuštěna, hlavní okno je viditelné
- Byl vybrán soubor *memory\_experiment.elf*
- Byla spuštěna a dokončena jeho emulace

TC_02_03_01	
Krok	Očekávaný výsledek
Žádný	Ve výpisu registrů jsou vypsány hodnoty registrů <b>x0</b> až <b>x31</b> a <b>pc</b> v <b>hexadecimálním</b> formátu
TC_02_03_02	
Krok	Očekávaný výsledek
Kliknout na radiobutton <i>Dec</i> u výpisu registrů	Ve výpisu registrů jsou vypsány hodnoty registrů <b>x0</b> až <b>x31</b> a <b>pc</b> v <b>dekadickém</b> formátu

## TS\_02\_04

Rychlý *smoke test* pro otestování **disassembly**.

Předpoklady ... Aplikace je spuštěna, hlavní okno je viditelné, byl vybrán soubor *fib.rv64.elf*

TC_02_04_01	
Krok	Očekávaný výsledek
Žádný	V disassembly je načteno <b>18086</b> řádků
TC_02_04_02	
Krok	Očekávaný výsledek
Žádný	Adresa první instrukce je <b>000101B8</b>
	Adresa poslední instrukce je <b>00021C4C</b>
TC_02_04_03	
Krok	Očekávaný výsledek
Žádný	První instrukce je <b>auipc gp,16384 #0x141b8</b>
	Poslední instrukce je <b>j 61440 #0x30c4c</b>
TC_02_04_04	
Krok	Očekávaný výsledek
Přidat breakpoint na první řádek	Zobrazí se breakpoint na prvním řádku
Přidat breakpoint na druhý řádek	Zobrazí se breakpoint na druhém řádku
TC_02_04_05	
Krok	Očekávaný výsledek
Kliknout na breakpoint na prvním řádku	Breakpoint na prvním řádku zmizí
Kliknout na breakpoint na druhém řádku	Breakpoint na druhém řádku zmizí
TC_02_04_06	
Krok	Očekávaný výsledek
Přidat breakpoint na první a druhý řádek	Zobrazí se breakpoint na prvním a druhém řádku
Kliknout na <i>File &gt; Open</i>	Zobrazí se dialog výběru souboru
Vybrat soubor <i>fib.rv64.elf</i>	Soubor se úspěšně načte
	Disassembly neobsahuje žádné breakpointy