

GRIFO_M_PBIT Simulation Mode - Launcher Guide

Overview

This document describes the launcher scripts for executing GRIFO_M_PBIT test in simulation mode without physical hardware.

Available Launchers

1. `run_simulate_simple.bat` (RECOMMENDED)

Best for: Quick testing with any Python installation

How it works:

- Uses the Python wrapper script to configure environment
- Works with system Python (3.7+) or embedded Python
- Automatically configures paths to use local test environment packages
- Bypasses uPlatSim.exe complexity

Usage:

```
cd
C:\src\___GitProjects\SXXXXXXX_PyBusMonitor1553\__OLD\__TEST_GENOVA\GrifoAutomati
cTestEnv
run_simulate_simple.bat
```

Requirements:

- Python 3.7 or higher available in PATH
- Or: Run with explicit Python interpreter (see alternative usage below)

Pros:

- ☒ Simple and straightforward
- ☒ Works with any Python 3.7+
- ☒ Easy to debug
- ☒ No dependency on uPlatSim.exe

Cons:

- ⚠ May have minor package version differences if using system Python

2. `run_simulate.bat` (PRODUCTION-LIKE)

Best for: Testing with exact production environment

How it works:

- Uses embedded Python from PlatformSimulator/bin/
- Executes via uPlatSim.exe in batch mode
- Creates temporary INI configuration for simulation mode
- Mimics production execution environment

Usage:

```
cd
C:\src\___GitProjects\SXXXXXXX_PyBusMonitor1553\__OLD\__TEST_GENOVA\GrifoAutomati
cTestEnv
run_simulate.bat
```

Requirements:

- uPlatSim.exe must be present in PlatformSimulator/bin/
- Embedded Python environment (python37.dll)

Pros:

- ☒ Uses exact production Python version (3.7)
- ☒ Identical package versions as production
- ☒ Closest to real test environment

Cons:

- ⚠ Depends on uPlatSim.exe supporting batch mode without hardware
- ⚠ More complex execution path

3. `python_simulate_wrapper.py` (MANUAL)

Best for: Advanced users, debugging, custom Python interpreters

How it works:

- Python script that configures environment paths
- Can be executed with any Python interpreter
- Provides detailed environment configuration output
- Verifies all required packages before test execution

Usage:

```
# Using system Python
python python_simulate_wrapper.py

# Using specific Python version
py -3.7 python_simulate_wrapper.py

# Using embedded Python (if available as standalone)
PlatformSimulator\bin\python.exe python_simulate_wrapper.py
```

```
# Direct execution (if Python associated with .py)
python_simulate_wrapper.py
```

Requirements:

- Python 3.7 or higher
- fpdf2, defusedxml, PIL, pyserial, pyvisa packages (available in env/site-packages/)

Pros:

- ☒ Maximum flexibility
- ☒ Detailed diagnostic output
- ☒ Easy to modify for custom scenarios
- ☒ Cross-platform compatible

Cons:

- ⚠ Requires manual Python invocation

Execution Flow

All launchers follow this pattern:

```
Launcher Script
↓
Configure Environment Paths
↓
Verify Dependencies
↓
Execute GRIFO_M_PBIT.py --simulate
↓
Simulation Mode Detection
↓
Import GRIFO_M_PBIT_mock.py
↓
setup_simulation()
↓
Run Test with Mock Objects
↓
Generate PDF Report
```

Which Launcher Should I Use?

Decision Tree:

1. Do you want the simplest option?

- YES → Use `run_simulate_simple.bat`

- NO → Continue

2. Do you need exact production environment?

- YES → Use `run_simulate.bat`
- NO → Continue

3. Do you want maximum control and diagnostics?

- YES → Use `python_simulate_wrapper.py` manually
- NO → Use `run_simulate_simple.bat`

Recommendations by Use Case:

- **First-time testing:** `run_simulate_simple.bat`
 - **Development/debugging:** `python_simulate_wrapper.py`
 - **Production validation:** `run_simulate.bat`
 - **CI/CD integration:** `python_simulate_wrapper.py` (scriptable)
 - **Quick test execution:** `run_simulate_simple.bat`
-

Troubleshooting

Error: "No Python interpreter found in PATH"

Solution:

```
# Option 1: Add Python to PATH temporarily
set PATH=C:\Path\To\Python;%PATH%
run_simulate_simple.bat

# Option 2: Use wrapper directly with full path
C:\Path\To\Python\python.exe python_simulate_wrapper.py

# Option 3: Use py launcher (if available)
py python_simulate_wrapper.py
```

Error: "ImportError: cannot import name 'FontFace'"

Cause: Using system Python with incompatible fpdf2 version

Solution:

```
# The wrapper script should handle this automatically by prioritizing
# local site-packages. If it persists:

# Check Python path priority
python python_simulate_wrapper.py
```

```
# Look for this output:  
#   Configuring environment paths:  
#     [+] ...\\TestEnvironment\\env\\site-packages  
  
# If site-packages is not first, check sys.path modification in wrapper
```

Error: "uPlatSim.exe not found"

Cause: Using `run_simulate.bat` without embedded environment

Solution:

```
# Use the simple launcher instead  
run_simulate_simple.bat  
  
# Or verify uPlatSim.exe location  
dir PlatformSimulator\\bin\\uPlatSim.exe
```

Test Runs but No PDF Report Generated

Check:

1. Console output for errors
2. Log file: `TestEnvironment\\LOG\\GRIFO_M_PBIT.log`
3. PDF reports folder: `TestEnvironment\\pdf_reports\\`

Common causes:

- fpdf2 package issues
 - File permission problems
 - Disk space
-

Output Files

After successful execution, find results in:

Log File:

```
TestEnvironment\\LOG\\GRIFO_M_PBIT.log
```

PDF Report:

```
TestEnvironment\\pdf_reports\\GRIFO_M_PBIT_<timestamp>.pdf
```

Report includes:

- Executive summary
 - Per-run detailed results with serial statistics
 - Aggregate statistics (pass/fail rates, timing)
 - Serial communication analysis (Errors, Fatal, Recycles)
 - Dedicated recycle events section
 - Test configuration and environment info
-

Advanced Usage

Custom Python Environment

```
# Create virtual environment with specific Python version
python -m venv grifo_test_env
grifo_test_env\Scripts\activate

# Install required packages from test environment
pip install fpdf2==2.7.8 defusedxml pillow pyserial pyvisa

# Run wrapper
python python_simulate_wrapper.py
```

Modifying Simulation Parameters

Edit `TestEnvironment\scripts\GRIFO_M_PBIT_mock.py`:

```
# Adjust PBIT duration (default: 15-25 seconds)
PBIT_TIME_MIN = 10.0 # Faster for testing
PBIT_TIME_MAX = 15.0

# Change scenario
SIMULATION_SCENARIO = 'pedestal_fail' # Or 'random_failures'

# Enable/disable serial message simulation
SIMULATE_SERIAL_ERRORS = True
SIMULATE_SERIAL_FATAL = True
SIMULATE_SERIAL_RECYCLE_EVENTS = True
```

Running from Different Directory

```
# Using absolute paths
cd C:\any\directory
```

```
python C:\src\...\GrifoAutomaticTestEnv\python_simulate_wrapper.py

# Or use batch file with full path
C:\src\...\GrifoAutomaticTestEnv\run_simulate_simple.bat
```

Continuous Integration Example

```
# GitHub Actions / GitLab CI example
test_simulation:
  script:
    - cd __OLD/__TEST_GENOVA/GrifoAutomaticTestEnv
    - python python_simulate_wrapper.py
  artifacts:
    paths:
      - TestEnvironment/pdf_reports/*.pdf
      - TestEnvironment/LOG/*.log
  when: always
```

Summary

Launcher	Simplicity	Production-Like	Flexibility	Recommended
run_simulate_simple.bat	★★★★★	★★★	★★★	☒ Yes
run_simulate.bat	★★★	★★★★★	★★	For production
python_simulate_wrapper.py	★★★★	★★★	★★★★★	For debugging

Quick Start: Use `run_simulate_simple.bat` for immediate testing.

Questions?

For issues or questions:

1. Check `TestEnvironment\LOG\GRIFO_M_PBIT.log` for detailed error messages
2. Review the Troubleshooting section above
3. Verify Python version: `python --version` (must be 3.7+)
4. Check package availability: `python -c "import fpdf; print(fpdf.__version__)"`

Related Documentation

- `README_SIMULATION.md` - Simulation mode overview and features
- `GRIFO_Test_Environment_Analysis.md` - Complete environment analysis
- `IMPLEMENTATION_SUMMARY.md` - Technical implementation details

