

Comprehensive Guide — CharLib Setup & Usage in IIC-OSIC-Tools Docker

This document is a compact, professional, and beginner-friendly guide for Chipathon participants to install and use CharLib inside the IIC-OSIC-Tools Docker image. It includes a fast mirror for ngspice (GitHub, with --depth=1), a self-healing launcher, and clear separation between commands run on the HOST (your OS) and INSIDE the CONTAINER.

Legend & Conventions

- HOST = your machine's terminal (PowerShell on Windows, Terminal on macOS/Linux).
- CONTAINER = a shell inside the running Docker container.
- Code blocks are copy-paste ready. Replace placeholders like <container_id> with your values.

1) Prerequisites

- Docker Desktop installed and running.
- IIC-OSIC-Tools image available (e.g., hpretl/iic-osic-tools:chipathon).
- Stable internet connection.

2) Start the Docker Container (via Docker Desktop)

1. Open Docker Desktop.
2. From the Images tab, locate the image (e.g., hpretl/iic-osic-tools:chipathon) and click Run, OR if you already have a container, go to the Containers tab and click Start.
3. To enter the container as root, use your HOST terminal (PowerShell/Terminal). First, copy the container ID from Docker Desktop (Containers page → Copy ID icon), then run:

HOST — enter container as root

```
docker exec -it -u 0 <container_id> bash
```

3) Build and Install ngspice (shared) — INSIDE CONTAINER

Use the GitHub mirror and shallow clone for speed. Run the following INSIDE the container (root shell):

```
apt-get update
apt-get install -y git build-essential autoconf automake
libtool pkg-config \
    bison flex libx11-dev libreadline-dev

cd /tmp
git clone --depth=1 https://github.com/imr/ngspice.git
cd ngspice
./autogen.sh
mkdir build && cd build
../configure --with-ngshared --enable-xspice --enable-cider
--disable-debug
make -j$(nproc)
make install
ldconfig

# verify
ls -l /usr/local/lib/libngspice.so*
```

4) Create Self-Healing CharLib Launcher — INSIDE CONTAINER

This launcher automatically prepares a Python virtual environment, installs CharLib (with its compatible PySpice fork), sets ngspice-shared, and runs CharLib. Paste the following INSIDE the container:

```
cat > ~/run_charlib.sh <<'EOF'
#!/usr/bin/env bash
set -euo pipefail

# Keep venv isolated from global site-packages/PYTHONPATH
exported by the image
unset PYTHONPATH
export PYTHONNOUSERSITE=1

VENV="$HOME/.venvs/charlib"
PY="$VENV/bin/python"
PIP="$VENV/bin/pip"
CHARLIB="$VENV/bin/charlib"

# Create venv if missing
if [ ! -x "$PY" ]; then
```

```

python3 -m venv "$VENV"
fi
source "$VENV/bin/activate"

# Install CharLib if missing (pulls the correct PySpice
fork)
if ! "$PY" -c "import importlib;
importlib.import_module('charlib')" >/dev/null 2>&1; then
    "$PIP" install --upgrade pip
    "$PIP" install --no-cache-dir
    "git+https://github.com/stineje/CharLib"
fi

# ngspice shared env
export NGSPICE_LIBRARY_PATH=${NGSPICE_LIBRARY_PATH:-
/usr/local/lib/libngspice.so}
export LD_LIBRARY_PATH=/usr/local/lib:${LD_LIBRARY_PATH:-}
export PYSPICE_SIMULATOR=${PYSPICE_SIMULATOR:-ngspice-
shared}
# keep enabled if your setup requires it; safe default
export LD_PRELOAD=${LD_PRELOAD:-
/usr/local/lib/libngspice.so}

# quick sanity
"$PY" - <<'PY'
import PySpice, inspect
from PySpice import Circuit, Simulator, SpiceLibrary
from PySpice.Spice.NgSpice.Shared import NgSpiceShared
print("PySpice:", getattr(PySpice, "__version__", "unknown"))
print("PySpice path:", inspect.getfile(PySpice))
NgSpiceShared(); print("ngspice shared OK")
PY

# execute charlib with passthrough args
exec "$CHARLIB" "$@"
EOF
chmod +x ~/run_charlib.sh

```

5) Run CharLib — INSIDE CONTAINER

Basic usage:

```
~/run_charlib.sh --help
```

Run a job (replace path with your mounted project folder):

```
~/run_charlib.sh run  
/foss/designs/<your_charlib_job_folder>
```

6) Daily Workflow (After Reboot)

1. Open Docker Desktop → Containers → Start your container.
2. HOST: copy container ID from Docker Desktop → open a terminal and run:

HOST — enter container as root

```
docker exec -it -u 0 <container_id> bash
```

3. INSIDE CONTAINER:

```
~/run_charlib.sh run  
/foss/designs/<your_charlib_job_folder>
```

7) Troubleshooting

- Very slow git clone from SourceForge: use the GitHub mirror with --depth=1 (as above).
- PySpice version mismatch (ImportError on Circuit/Simulator): the launcher isolates venv, unsets PYTHONPATH, and installs the compatible fork automatically.
- ngSpice shared errors (undefined symbol ngSpice_Init): keep LD_PRELOAD=/usr/local/lib/libngspice.so enabled in the launcher.