

How to Use u64-remote (C++)

This tool sends a **.prg** file from your Linux machine to a **Commodore 64 Ultimate / Ultimate-64** over the network and **runs it remotely**.

Prerequisites (one-time)

On the C64 Ultimate

- Ultimate firmware supports the REST API
- Network enabled
- Password set (or empty if disabled)
- You already know it's reachable (e.g. 10.0.0.250) ← *Edit to reflect your private IP*

You can verify manually from Linux:

```
curl http://10.0.0.250/v1/version ← Edit to reflect your private IP
```

If that responds → the device is ready.

On Linux (Pop!_OS / Debian / Ubuntu)

Required packages: *(Install only if needed)*

```
sudo apt install -y build-essential cmake libcurl4-openssl-dev
```

Build the project

```
cd /home/eric/development/c++/u64-remote/u64-remote-cpp
mkdir -p build
cd build
cmake ..
make -j
```

This produces:

```
build/u64-remote
```

Credentials file (recommended)

Create a credentials JSON file (example):

 creds.json (creds = credentials)

```
{
  "address": "http://10.0.0.250", ← Edit to reflect your private IP
  "password": "YOUR_C64U_PASSWORD"
}
```

You already keep yours in:

```
/home/eric/development/c++/u64-remote/json_examples/
```

That's perfectly fine.

Basic usage (most common)

Run a PRG using credentials

```
./u64-remote \  
  --creds /home/eric/development/c++/u64-remote/json_examples/creds.json \  
  myprogram.prg
```

What happens internally

1. Reads credentials
2. Calls:

```
GET /v1/version
```


(sanity check)
3. Uploads PRG:

```
POST /v1/runners:run_prg
```
4. PRG runs immediately on the C64U

Run without a creds file (manual)

```
./u64-remote \  
  --address http://10.0.0.250 \- Edit to reflect your private IP  
  --password YOUR_C64U_PASSWORD \  
  myprogram.prg
```

Useful for scripting or quick tests.

Auto-discover the C64 Ultimate (recommended feature)

Auto-discovery is a new added feature. However, it is recommended you use a static private-ip.

If you **don't want to hard-code the IP**:

```
./u64-remote --discover myprogram.prg
```

What discovery does

- Scans local IPv4 subnets
- Probes:

```
http://<ip>/v1/version
```

- If **one device found** → uses it automatically
- If **multiple found** → prompts you to choose

This directly solves:

“Do I need to preset the IP?”

No — discovery handles it.

Useful diagnostics

Show help

```
./u64-remote --help
```

Verbose/debug (if enabled in your build)

```
./u64-remote --debug --discover myprogram.prg
```

Test connectivity manually

```
curl http://10.0.0.250/v1/version- Edit to reflect your private IP
```

Typical workflow (real-world)

```
# Build your C64 program
cl65 -t c64 mygame.c -o mygame.prg

# Send and run it
./u64-remote --discover mygame.prg
```

No SD cards.




No USB swapping.

No reboot.

Common mistakes (and fixes)

Problem	Cause	Fix
Program exits immediately	No args	Provide .prg
Connection error	Wrong IP/password	Use <code>--discover</code>
Permission denied	REST password mismatch	Fix creds
Terminal closes	Launched by double-click	Run from shell

What this tool is *not*

-  Not a debugger
-  Not a file browser
-  Not a replacement for Ultimate menu

It is intentionally:

A fast, scriptable PRG launcher

Summary (mental model)

Think of `u64-remote` as:

`scp` + `run` for a Commodore 64 Ultimate

You compile on Linux → push → execute instantly. You can also compile on a Windows PC., however since I am not a Windows user I cannot provide any advice.