# Auto Clock Speed

| BUILD | PASSING | CRATES.IO | V0.1.10 | DOWNLOADS | 978 |
|---|---|---|---|---|---|

**ACS Upstream - autoclockspeed.org - Our crates.io - ACS Github Org** A utility to check stats about your CPU, and auto regulate clock speeds to help with either performance or battery life. This program is designed for Linux and Intel laptops, although it should theoretically work on AMD systems and sometimes desktops as well. If you encounter any issues or bugs, please refer to the wiki to see if there is a solution.

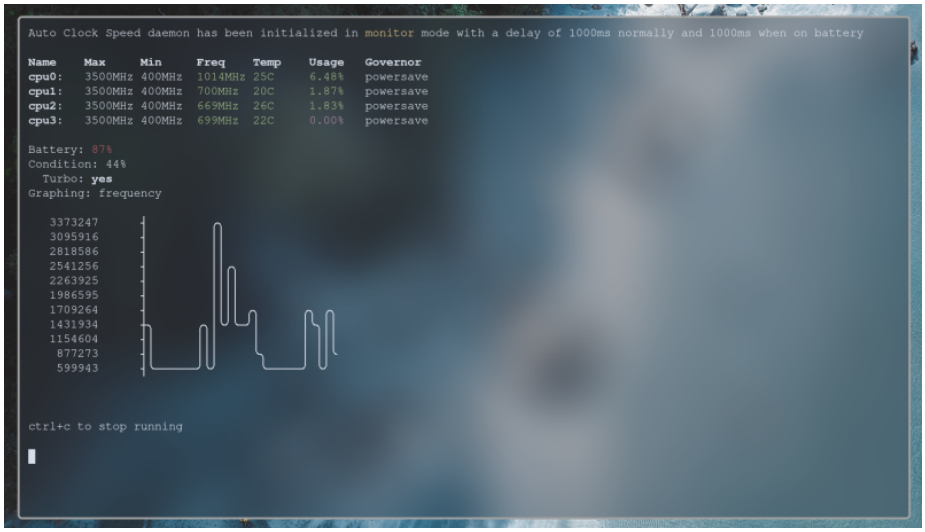



Figure 1: acs

## Goals

- First and foremost, this is a project to learn about Rust and Linux
- Secondly, try to improve upon AdnanHodzic's already amazing auto-cpufreq
- Add options to display raw output of governors, clockspeed, turbo, battery, etc. for use in scripts or display panels like polybar.

## Want to help? Yay! Welcome!

- Read our CONTRIBUTING.md for some helpful tips

- Find an issue - “good first issue” recommended
- Feel free to ask questions!

## Install Latest Release

If you have cargo on your machine, skip to step 3

1. Go to `rustup.rs` to install rust.
2. Setup rust

   ```
   rustup override set stable
   rustup update stable
   ```

3. Clone the project and install

   ```
   git clone https://github.com/JakeRoggenbuck/auto-clock-speed

   cargo install --path auto-clock-speed

   # This is needed to have the root version of acs match the local installed version
   sudo cp ~/.cargo/bin/acs /usr/bin/acs
   ```

Note: The latest release of acs can also be installed locally with the following

```
cargo install autoclockspeed
```

## Tested Devices

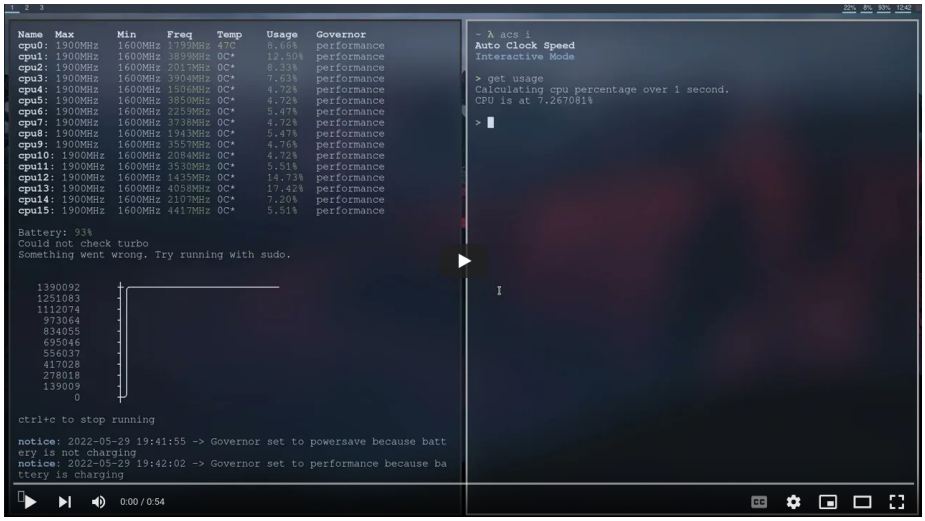
Auto clock speed has been tested to work on the following devices. If you have a device that is not listed please submit a pull request.

| Functionality | Description |
|---|---|
| Working | All parts of ACS are fully functional, the computer has enough data to make decisions on governor changes and can be run in edit mode |
| Mostly Working | ACS is unable to understand some data from the computer however certain data (like battery life, battery condition, temperature etc) which is non essential in making governor decisions, is missing |

| Functionality | Description |
|---|---|
| Barely Working | ACS is unable to be ran in edit mode due to missing data from the system, monit mode may still work however functionality is limited. If you have a system that falls under this category please open an issue |
| Borked | ACS cannot find any useful data. Please open an issue |

| Device Name | Functionality | Notes |
|---|---|---|
| Dell XPS 13 9360 | Working | |
| Dell Latitude 7480 | Working | |
| Steam Deck | Working | Edit mode not neccessary (use built in governor switcher) |
| Thinkpad T400 | Working | |
| Thinkpad X230 | Working | |
| Thinkpad W540 | Working | |
| ThinkPad X1 Extreme Gen 1 | Working | |
| Thinkpad P1 Gen 4 (Intel Core) | Working | |
| Thinkpad P14 Gen 2 (AMD) | Mostly Working | See #443 |

## In Action




## New Interactive Mode


Figure 2: image

## Systemd

In order to have auto-clock-speed start when you restart your computer you must follow these instructions

```
# IMPORTANT: Modify the service file (acs.service) in the
# project directory to include the path to the binary file
# (usually /home/username/.cargo/bin/acs)

# In the auto clock speed directory run this command to
# move the service file into your systemd directory
sudo cp acs.service /etc/systemd/system/

# Start and enable the service
sudo systemctl start acs
sudo systemctl enable acs

# Check service is up and running
systemctl status acs
```

## Systemctl command

The line after `[Service]` in `acs.service` is the command that will be run. You may want to add or remove arguments, mainly `--quiet`.

```
[Unit]
Description=Manages Clock Speed

[Service]
ExecStart=/home/your-user-here/.cargo/bin/acs run --no-animation --quiet

[Install]
WantedBy=multi-user.target
```

## Config

### Using default config

```
WARN: Using default config. Create file '/etc/acs/acs.toml' for custom config or run 'acs in
```

This warning recommends creating a config file, use the initconfig command to automatically create one for you!

```
sudo acs initconfig
```

### This is an example config

also the default settings if no config is provided

```
# acs.toml
powersave_under = 20
```

```
overheat_threshold = 80
active_rules = [ "battery_percent_rule", "lid_open_rule", "ac_charging_rule", "cpu_usage_rul
```

## Turn Off

If you would like to turn off auto-clock-speed, here are the steps. Note: This should be done during testing of acs run mode.

```
# Temporarily stop (only lasts until reboot)
sudo systemctl stop acs

# Permanently stop until turned on
sudo systemctl disable acs
```

## Uninstall

Here is how to uninstall the binary and the systemctl service.

```
# Remove local binary
cargo uninstall acs

# Remove system shared binary
rm /usr/bin/acs

# Remove systemctl entry
rm /etc/systemd/system/acs.service
```

## Example Usage

Here are some examples of how acs can be used.

```
# Monitor mode
acs monitor

# Run as root
sudo acs run

# Get all speeds
acs get speeds

# Select gov from dmenu
sudo acs set gov $(acs get available-govs --raw | dmenu)
```

## Detailed usage

Detailed usage can be found on our wiki

## Help

```
Automatic CPU frequency scaler and power saver

USAGE:
    acs <SUBCOMMAND>

FLAGS:
    -h, --help       Prints help information
    -V, --version    Prints version information

SUBCOMMANDS:
    daemon         Controls interaction with a running daemon
    get            Get a specific value or status
    help           Prints this message or the help of the given subcommand(s)
    initconfig     Initialize config
    interactive    Interactive mode for auto clock speed commands
    monitor        Monitor each cpu, it's min, max, and current speed, along with the goverr
    run            Run the daemon, this checks and edit your cpu's speed
    set            Set a specific value
    showconfig     Show the current config in use
```