

Pre-Build Preparation Guide

Purpose: Comprehensive preparation before starting a custom build

Date: 2025-01-03

Status: Preparation checklist and workflow

■ Goal

Prepare everything before starting the build to ensure:

- ■ All configurations are correct
- ■ All tools are ready
- ■ Monitoring is set up
- ■ Efficient workflow established

■ Pre-Build Checklist

1. System Status ■

- [] **Docker running:** `docker info` should work
- [] **No build running:** `docker ps -a | grep moode-builder` should be empty
- [] **Latest image checked:** Know which image is current
- [] **Disk space:** At least 20GB free for build

2. Configuration Verification ■

- [] **User config:** `FIRST_USER_NAME=andre` in `imgbuilder/pi-gen-64/config`
- [] **Password:** `FIRST_USER_PASS=0815` in config
- [] **SSH enabled:** `ENABLE_SSH=1` in config
- [] **Display config:** `config.txt.overwrite` has correct settings
- [] **Services:** All custom services in place
- [] **Scripts:** All custom scripts in place

3. Tools Ready ■

- [] **Build tool:** `./tools/build.sh` exists and is executable
- [] **Monitor tool:** `./tools/monitor.sh` exists and is executable
- [] **Test tool:** `./tools/test.sh` exists and is executable
- [] **Toolbox:** `./tools/toolbox.sh` works

4. GitHub MCP Setup ■

- [] **GitHub token:** Added to `.env`
- [] **GitHub MCP active:** Can search repositories
- [] **Ready to research:** Can check moOde source code if needed

■ Pre-Build Verification Commands

Run these before starting:

```
# 1. Check Docker
docker info

# 2. Check no build running
docker ps -a | grep moode-builder

# 3. Check build tool
cd ~/moodeaudio-cursor
./tools/build.sh --status

# 4. Check disk space
df -h | grep "(/$|/Users)"

# 5. Check configuration
grep "FIRST_USER_NAME" imgbuild/pi-gen-64/config
grep "ENABLE_SSH" imgbuild/pi-gen-64/config

# 6. Check latest image
ls -lht imgbuild/deploy/*.img | head -1
```

■ Build Workflow

Step 1: Pre-Build (Before Starting)

1. **Run verification commands** (see above)
2. **Check all configurations** are correct
3. **Ensure no build is running**
4. **Have monitoring ready**

Step 2: Start Build

```
cd ~/moodeaudio-cursor
./tools/build.sh --build
```

Step 3: Monitor Build

Option A: Using monitor tool

```
./tools/monitor.sh --build
```

Option B: Manual monitoring

```
# Watch logs
docker-compose -f docker-compose.build.yml logs -f

# Check status
docker ps -f name=moode-builder
```

Step 4: Post-Build

```
# Validate build
./tools/build.sh --validate

# Check image
ls -lh imgbuild/deploy/moode-r1001-arm64-*.img
# Deploy if ready
```

```
./tools/build.sh --deploy
```

■ Monitoring & Efficiency Tools

Available Tools

1. **`./tools/build.sh`**
 - `--build` - Start build
 - `--monitor` - Monitor build progress
 - `--status` - Check build status
 - `--validate` - Validate build image
 - `--deploy` - Deploy to SD card
 - `--cleanup` - Cleanup old images
2. **`./tools/monitor.sh`**
 - `--build` - Monitor build
 - `--pi` - Monitor Pi status
 - `--all` - Monitor everything
3. **`./tools/toolbox.sh`**
 - Interactive menu for all tools

Efficiency Tips

- ■ Use `toolbox.sh` for interactive workflow
- ■ Use `build.sh --monitor` for automated monitoring
- ■ Use GitHub MCP to research issues if needed
- ■ Keep build logs for reference

■ Using GitHub MCP for Build Preparation

Now that GitHub MCP is active, I can:

- ■ **Research moOde source code** if issues arise
- ■ **Check for updates** in moOde repository
- ■ **Find solutions** in GitHub issues/PRs
- ■ **Read documentation** from GitHub
- ■ **Compare configurations** with official moOde

Example uses:

- "Check moOde repository for display configuration"
- "Find solutions for build errors"
- "Compare our config with official moOde"

■ Ready to Build?

Before starting, confirm:

1. ■ All checklist items completed

2. ■ Verification commands passed
3. ■ Tools are ready
4. ■ Monitoring is set up
5. ■ GitHub MCP is active

■ Next Steps

1. **Run pre-build verification** (commands above)
2. **Review configurations** one more time
3. **Start build** when ready: `./tools/build.sh --build`
4. **Monitor progress** using tools
5. **Use GitHub MCP** if issues arise

Ready to proceed? Let me know when you want to start the build! ■