



Quail Hatchery - community package

HACS integration bundle | v0.1.0 public beta



Track incubation batches, hatch outcomes, archives and permanent bird tags in Home Assistant. This bundle prepares the owner-supplied Pyscript, all 27 helpers, the complete dashboard, a reusable YAML card, artwork, JSON storage and a notification blueprint.

Two ways to get it

HACS: add [Tryon-Electronics/tryon-quail-hatchery](#) as an **Integration** custom repository. **Manual:** use the source ZIP and copy its `custom_components/tryon_quail_hatchery` folder into `/config/custom_components/`. Both methods then follow the setup flow on page 2.

What is automated

The integration prepares its bundled files and seeds an empty `batches.json` only when none exists. It preserves hatch records and user-edited files. You still enable YAML packages, register the dashboard, install the dashboard dependencies, and choose your phone notification action.

Validation status: package/source and simulated lifecycle tests pass. The full Home Assistant/Pyscript/frontend installation still needs a live-instance test. This is a custom-repository public beta, not default HACS catalog inclusion.



Install and enable the dashboard

HACS dependency	Required for
Pyscript	Tracker services, local JSON storage; configure Allow all imports.
Mushroom	Title and template cards.
Auto Entities	Dynamic batch, archive and tag lists.
Card Mod	Dashboard card styling.
Browser Mod	Batch and tag editor popups; configure its integration/browser connection.

1. Install and configure the dependencies above. In the Pyscript integration options, enable **Allow all imports** for the tracker's local file access. If Pyscript was initially configured in YAML, enable the option there instead; do not mix configuration methods.
2. Add the repository to HACS as an **Integration**, download Tryon Quail Hatchery, and restart Home Assistant. In **Settings > Devices & services**, add **Tryon Quail Hatchery**, read the setup screen, and confirm.
3. Review the installation notification for conflicts. Merge these entries into your existing configuration.yaml without duplicating existing homeassistant or lovelace keys:

```
homeassistant:
  packages: !include_dir_named packages

lovelace:
  dashboards:
    quail-hatchery:
      mode: yaml
      title: Quail Hatchery
      icon: mdi:egg-easter
      show_in_sidebar: true
      filename: dashboards/quail-hatchery.yaml
```

4. Check configuration, restart, then open **Quail Hatchery**. Add a test batch and verify active-batch sensors and the editor popups. Sensors refresh at startup, hourly, and after changes.

For a storage dashboard instead, paste the complete dashboard YAML into its raw editor. For one existing-dashboard card, paste **cards/quail-hatchery-card.yaml** into a Manual card. This is a reusable YAML card assembled from the supplied dashboard, not a separate JavaScript card repository.

The regular script is /config/pyscript/incubator.py, so no Pyscript apps entry is required. Existing package layouts may differ: page 4 covers migration and merge-named includes.



Files, JSON and opt-in notifications

Path under /config/	Contents
pyscript/incubator.py	Tracker and its 14 exposed services.
packages/incubator.yaml	27 helper definitions.
dashboards/quail-hatchery.yaml	Complete dashboard YAML.
cards/quail-hatchery-card.yaml	Reusable main dashboard section.
www/tryon-quail-hatchery/quail-family-bg.png	Supplied hatchery artwork.
blueprints/automation/incubator/incubator_notifications.yaml	Daily summary and candling/lockdown/hatch-day reminders.
incubator_data/batches.json	Your live batch/tag JSON; initialized as [] only if absent.
incubator_data/batches.backup.json	Previous source saved by the tracker during valid writes/migrations.
tryon_quail_hatchery/examples/batches.example.json	Fictional example; never seeded into live storage.

Create your notification automation

Open **Automations & scenes > Blueprints**, choose **Incubator - Daily Status & Milestone Alerts**, then enter your own notify service(s), preferred time and daily-summary setting. The supplied automation referenced a blueprint without its implementation; this package includes a new compatible blueprint using the same path and input name.

```
- alias: Incubator - Daily Status & Milestone Alerts
  initial_state: false
  use_blueprint:
    path: incubator/incubator_notifications.yaml
  input:
    phone_notification_target:
      - notify.mobile_app_your_phone
    notification_time: "09:00:00"
    daily_summary: true
```

Replace the example notify service before enabling. The blueprint sends once at the chosen time, reports milestones due today, and skips notifications on data errors. It does not resend missed past milestones or monitor incubator temperature/humidity.



Existing installs, updates and recovery

Keep helpers separate from hatch records

Helper type	Count	Purpose
input_text	11	Batch names, dates, notes and editor/tag IDs.
input_number	11	Egg counts, incubation days, milestone days and results.
input_select	4	Tag picker, color, sex and status.
input_boolean	1	Batch archive editor state.

Home Assistant creates these entities from `packages/incubator.yaml`. Adding helper descriptions to `batches.json` would not create them. Keep your batch/tag records in JSON and enable one copy of the helper YAML.

Migrate an existing hatchery safely

Back up your data, script, helpers, dashboard and automations first. Use only one `incubator.py` script and one set of helper IDs. The installer preserves existing files that differ and writes proposed replacements with a **.quail-update** suffix. Those staged files are not automatically executed or loaded.

If packages use **!include_dir_merge_named**, wrap the shipped helper file beneath an **incubator:** key. If packages already use a selected-file include, retain that convention. Remove duplicate legacy package/UI helpers only after reviewing your backup.

Updates and JSON recovery

HACS updates the integration; restart to prepare its new bundle. Unedited installer-owned files get `.quail-backup` copies before replacement. Edited files remain untouched with `.quail-update` candidates. Check configuration and restart again when YAML changes. `batches.json` is outside the integration directory and is never overwritten by the installer.

If JSON is malformed, the tracker reports a data-status error and keeps existing dynamic sensors. Restore the backup only after saving the current file, then run **pyscript.incubator_refresh**. The supplied legacy Survival migration is corrected so earlier hatch counts are preserved.

Quick checks and troubleshooting

Missing helpers: verify the package include, duplicates and restart. **Unknown cards:** check Mushroom/Auto Entities/Card Mod resources and reload the browser. **No editor popup:** check Browser Mod connection. **No tracker services:** check Pyscript imports and logs. **No phone alert:** verify your notify service, enabled automation, daily time and data-status sensor.

Removal leaves config and data in place. Disable the alert automation, remove the dashboard/helper configuration and rename the script so it does not end in `.py`; check configuration and restart. Retain JSON for recovery.

Source, dependency links and tests: github.com/Tryon-Electronics/tryon-quail-hatchery. Support: tryonelectronics.com/support.