Module 08

Cron Jobs and Heartbeats

Making Your Agent Proactive

Navigation Chart

By the end of this module, you will be able to:

1. **Explain** what heartbeats are and how they make your agent proactive
2. **Configure** heartbeat frequency and behavior
3. **Use cheaper models** for heartbeats to save money
4. **Set up cron jobs** for scheduled automations
5. **Create a morning briefing** that runs daily
6. **Understand** the difference between reactive and proactive behavior

📜 Ship's Logbook

Term	Definition
Heartbeat	A periodic check-in where the agent "wakes up," reviews its instructions, and takes action if needed
Cron job	A scheduled task that runs at a specific time or interval, named after the Unix `cron` scheduler
Proactive	Acting without being asked -- the opposite of reactive
Morning brief	A daily summary sent to you at a set time -- weather, news, tasks, calendar
Mission control	A custom dashboard or workspace the agent builds and maintains for you

Reactive vs. Proactive

Reactive	Proactive
You ask "What's the weather?"	Your agent sends weather at 7 AM every morning
You ask "Summarize this article"	Your agent monitors news sources and sends summaries
You say "Remind me to..."	Your agent reviews your goals and suggests priorities
You ask, your agent answers	**Your agent acts, you benefit**

- Heartbeats and cron jobs are what make **proactive behavior** possible
- This is where 🦞 OpenClaw goes from chatbot to true assistant

How Heartbeats Work

A heartbeat is a periodic "wake-up" for your agent. Every X minutes, the ⛵ gateway triggers your agent to:

1. Read its `HEARTBEAT.md` instructions
2. Check for pending tasks, messages, or triggers
3. Take action if needed
4. Go back to sleep until the next heartbeat

Defaults

- **Frequency:** Every 30 minutes (or every hour with the daemon)
- **What it does:** Checks for unread messages, pending tasks, scheduled items
- **Cost:** Each heartbeat consumes tokens

🚩 Rough Waters: Heartbeat Cost Explosion

🦞 OpenClaw sends **everything** to your primary model by default. If your primary is Opus 4.6, you are paying premium prices for routine checks.

- **System prompt re-injection:** 3,000-14,000 tokens re-sent with every API call
- **Heartbeat overhead:** 48 full-context API calls per day at 30-min intervals
- **Cron job overhead:** Each trigger creates a fresh conversation with full context

> ***This is the #1 cost mistake new OpenClaw users make.*** *A well-configured multi-model setup with prompt caching can reduce monthly API costs by up to 90%.*

Configuring Heartbeat Frequency

```
openclaw config heartbeat interval 30
```

Interval	Use Case	Cost Impact
15 minutes	High-responsiveness (urgent monitoring)	Higher
30 minutes	Good balance for most users	Moderate
60 minutes	Budget-conscious, low urgency	Lower
0 (disabled)	No proactive behavior	Zero

> ***Recommended default:*** *Set heartbeat to* ***55 minutes*** *+ route to* ***Haiku*** *+ enable caching = 99.5% cost reduction on heartbeats. This should be the first thing you configure, not an afterthought.*

- **To disable heartbeats entirely:**

Use Cheap Models for Heartbeats

Most heartbeat checks are routine: "Any new messages? Any tasks due? No? Go back to sleep." This does not require Opus 4.6.

```
openclaw config heartbeat model claude-haiku-4-5
```

Or with Google Gemini:

```
openclaw config heartbeat model gemini-flash-3
```

Heartbeat Cost Comparison

Model	Cost per Heartbeat	Monthly Cost (30-min intervals)
Claude Opus 4.6	~$0.05-0.15	~$70-200
Claude Sonnet 4.5	~$0.01-0.05	~$15-70
Claude Haiku 4.5	~$0.001-0.005	~$1-7
Gemini Flash 3	Free (20/day) or very cheap	~$0-5

Using Haiku 4.5 instead of Opus saves **$50-190 per month**.

- Use the cheap model for routine checks
- Escalate to Opus only when something important is found

Local Heartbeat Models via LM Studio

Run heartbeats on **local 3-4B models** for literally zero cost -- no API calls, no tokens billed.

Recommended Local Models

Model	Strength	Speed
Qwen 3 4B	"Agentic King" -- most stable for tool-use and structured output	~30 sec per heartbeat
Gemma 3 4B	Excellent instruction following, clean output	~21 sec per heartbeat

- Both are roughly **GPT-4o mini equivalent** for routing and heartbeat tasks
- Run them in **LM Studio** with the OpenAI-compatible API endpoint
- Point 🦞 OpenClaw's heartbeat config at `http://localhost:1234`

When to Use Local vs. Cloud

Setup	Recommendation

The 55-Minute Prompt Caching Trick

Anthropic's extended cache stays warm for about **55 minutes**. Set your heartbeat to 55 minutes so every heartbeat hits warm cache:

```
openclaw config heartbeat interval 55
```

Combined savings: Haiku + prompt caching + 55-min interval

Without Optimization	With Optimization
Opus for heartbeats	Haiku for heartbeats
Full system prompt every call	Cached system prompt (90% off)
~$100+/month	~$0.50/month

That is a **99.5% reduction** on heartbeat costs alone.

Configuring HEARTBEAT.md

This file tells your agent what to do during each heartbeat:

```
nano ~/.openclaw/workspace/HEARTBEAT.md
```

Structure your priorities:

- **Priority 1:** Check messages -- respond to urgent ones immediately
- **Priority 2:** Scheduled tasks -- execute due tasks, log completions
- **Priority 3:** Morning brief -- send between 6:30-7:00 AM if not yet sent
- **Priority 4:** Health check -- verify gateway and API connectivity
- **Cost rules:** Use cheapest model for routine, escalate for complex tasks

Setting Up a Morning Briefing

The morning brief is the "killer feature" most OpenClaw users set up first.

Three methods:

Method 1: Via HEARTBEAT.md -- simplest, catches the AM window automatically

Method 2: Via cron job -- precise timing:

```
openclaw cron add "morning-brief" \
  --schedule "0 7 * * *" \
  --task "Send my morning brief to Telegram"
```

Method 3: Just ask your agent -- easiest:

```
Set up a daily morning brief at 7 AM on Telegram.
Include weather, top AI news, my tasks, and one suggestion.
```

Testing Your Morning Brief

Do not wait until tomorrow morning. Test it now:

```
Generate and send my morning brief right now as a test.
```

Morning brief format should include:

1. Date and greeting
2. Weather for your city
3. Top 3 news stories in your interest areas
4. Today's tasks from 🪸 memory and pending reminders
5. One proactive suggestion based on your goals

Check Telegram. Adjust and re-test until it is right.

Understanding Cron Syntax

Cron uses five fields: `minute hour day-of-month month day-of-week`

```
*  *  *  *  *
|  |  |  |  |
|  |  |  |  day of week (0-7, 0 and 7 = Sunday)
|  |  |  month (1-12)
|  |  day of month (1-31)
|  hour (0-23)
minute (0-59)
```

Pattern	Meaning
`0 7 * * *`	7:00 AM every day
`30 8 * * 1-5`	8:30 AM weekdays only
`0 */2 * * *`	Every 2 hours
`0 9 * * 1`	9:00 AM every Monday

Cron Job Examples

Schedule	Task	Cron Pattern
Every day at 7 AM	Morning brief	`0 7 * * *`
Every Monday at 9 AM	Weekly review	`0 9 * * 1`
Every hour	Check Bitcoin price	`0 * * * *`
Every day at 6 PM	End-of-day summary	`0 18 * * *`
1st of month at 10 AM	Monthly goals review	`0 10 1 * *`

Cost warning:

- 1 cron/day = ~30 API calls/month (manageable)
- 1 cron/hour = ~720 API calls/month (adds up)
- 1 cron/15 min = ~2,880 API calls/month (expensive on Opus)

Managing Cron Jobs

```
# List all cron jobs
openclaw cron list

# Add a cron job
openclaw cron add "weekly-review" \
  --schedule "0 9 * * 1" \
  --task "Conduct my weekly review"

# Remove a cron job
openclaw cron remove "weekly-review"

# Temporarily disable a cron job
openclaw cron disable "weekly-review"
```

- Use **cheap models** for routine cron jobs
- **Combine related checks** into a single job
- **Enable prompt caching** to reduce system prompt overhead

Practical Automation Ideas

- **End-of-day summary (6 PM):** Tasks completed, what is pending, tomorrow's priorities
- **Weekly review (Monday 9 AM):** Goals progress, accomplishments, lessons learned
- **Price alerts (hourly):** Only message if above/below threshold -- silence for routine checks
- **News monitoring (every 4 hours):** Only alert for genuinely important stories

Best practices:

- Be **specific** about format, content, and delivery channel
- Set alerts for **important things only** -- avoid notification fatigue
- Verify your WSL2 timezone: `timedatectl`

The Three Essential Cron Jobs

Set these up before anything else -- experienced users consider them **non-negotiable**:

Cron Job	Schedule	Purpose
Session cleanup	Every 72 hours	Delete bloated session files that slow your agent
Daily security audit	Every morning	Check firewall, fail2ban, SSH, ports, Docker
Silent backups	Every 2 hours	Git push workspace -- never lose config/memory

Executive Assistant Scheduling

Create a `SCHEDULING.md` with your rules:

- Working hours, hard boundaries, VIP overrides
- Travel buffers, video platform defaults
- CC your agent into email threads for autonomous scheduling

Mission Statement Reverse Prompt

Add a Mission Statement

Add a mission statement to `IDENTITY.md` or `AGENTS.md`:

```
## Mission
Build a sustainable freelance business doing $10K/month
by helping small businesses with AI automation.
```

Schedule a Cron to Reverse-Prompt

```
openclaw cron add "mission-nudge" \
  --schedule "0 8 * * *" \
  --task "What is 1 task we can do today to get closer to our mission?"
```

Why This Works

- The agent **suggests tasks you never thought of** -- it sees patterns across all your projects,

Advanced: Compaction and Mission Control

Set Compaction Thresholds Early

Design for long-running sessions from the start:

- Set 🪸 memory flush threshold: **80K tokens**
- Set compaction threshold: **80K tokens**
- Don't wait until the agent crashes from context overflow

Mission Control Dashboard

Create a cron job that generates a **daily status dashboard**:

- Agent health and uptime
- Token spend (daily/weekly/monthly)
- Active tasks and progress
- Upcoming calendar events

Gmail Pub/Sub and Webhook Automation

Beyond cron and heartbeats, OpenClaw supports **event-driven automation**:

Trigger Type	When to Use	Example
Cron	Time-based, recurring	Morning brief at 7 AM
Heartbeat	Periodic check-in	Check messages every 30 min
Gmail Pub/Sub	Real-time email events	Process invoices immediately
Webhook	External service events	React to GitHub PRs, Stripe payments

- **Gmail Pub/Sub:** Instant notification when email arrives (no polling needed)
- **Webhooks:** Connect GitHub, Stripe, Shopify, monitoring tools to trigger agent actions
- Full guide: docs.openclaw.ai/automation/gmail-pubsub

Session Lifecycle

Understanding when sessions live and die prevents lost work and wasted money.

Sessions Are Stateful Only While the Chat Window Is Open

- The moment you **close the TUI or terminal**, the session dies and work stops
- There is no background daemon keeping your conversation alive
- Any in-progress task the agent was working on **halts immediately**

What This Means for Long Tasks

- **Use cron jobs for background work**, not long-running sessions -- cron jobs trigger fresh conversations on a schedule and do not depend on an open window
- **Break one-off tasks into discrete steps** with saved output -- ask the agent to write results to a file after each step so nothing is lost if the session drops
- **Long-running sessions accumulate context (200K+ tokens)** -- the agent hallucinates

Shoals and Sandbars

Mistake	Fix
Setting heartbeats too frequent	Start at 55-60 minutes, decrease only if needed
Using Opus for heartbeats	Use Haiku 4.5 or Gemini Flash
Not testing automations	Always test with "do it now"
Vague cron job instructions	Be specific about format and delivery
Too many alerts	Only alert for genuinely important things
Cron jobs in wrong timezone	Verify with `timedatectl`, fix with `sudo timedatectl set-timezone`

⚙ Hands on Deck

Part 1: Set Up the Morning Brief (15 min)

1. Edit `HEARTBEAT.md` with morning brief instructions
2. Set heartbeat model to Haiku or Gemini Flash
3. Restart the ⛵ gateway
4. Test by asking your agent to generate the brief now
5. Verify it arrives on Telegram

Part 2: Create One Custom Cron Job (10 min)

- Choose: weekly review, daily summary, price alert, news, or custom reminder
- Set it up, test it, verify it works

Part 3: Evaluate (5 min)

- Was the format easy to read on your phone?

💎 Treasure Chest

1. **Heartbeats make your agent proactive** -- it checks in periodically and acts without being asked
2. **Use cheap models for heartbeats** -- Haiku 4.5 or Gemini Flash saves significant money
3. **The 55-minute trick** -- set heartbeats to 55 min to hit the prompt cache window, reducing costs by 90%
4. **The morning brief is the killer feature** -- set it up first, customize over time
5. **Cron jobs give precise scheduling** -- but each trigger costs tokens, so use cheap models
6. **Start conservative** -- 55-60 min heartbeats, minimal alerts; increase as needed
7. **Test everything immediately** -- do not wait for the next trigger

Next Port of Call

Module 09: Advanced Concepts

Brains and muscles, model switching, ClawRouter, prompt caching, reverse prompting, and multi-agent setups.