

HASMASTER LAUNCH GUIDE

HASMaster Overview

Launch edition: 2026-04-15

HasMaster Themes and Plugins	Environment	Fans	AI and Assistants	Image and Video Generation	AI Servers
Journeys	Security	Heaters	Backup and Recovery	Inference Engines	Voice Assistants
Automate	Surveillance	Humidifiers & Dehumidifiers	Developer Tools	Vision and Detection	DIY Projects
Q1 – Automation Goal	Convenience	Thermostats & HVAC	Document and Knowledge Management	Voice Networks	3D Printing
Q2 – Automation Inputs and Outputs	Automations	Water Heaters	Fitness and Wellness	Bluetooth / BLE Networks	Custom Projects
Q3 – Automation Tools	Device Control	Covers	Media and Entertainment Management	Thread Mesh Networks	Electronics & Wiring
Q4 – Validation and Fallbacks	Voice Control	Blinds & Shades	Network and Infrastructure Management	Wi-Fi and Wired Network	HASMaster
Q5 – Save and Continue	Organization	Garage Doors	Productivity and Scheduling	Z-Wave Mesh Networks	Prompt Library
Define Constraints	Content	Gates	Recipes and Meal Planning	Zigbee Mesh Networks	Structure
Q1 – Ownership and Budget	Fitness	Energy & Storage	Security and Identity Management	Protocols	Theme and Plugin Overview
Q2 – Platform Preferences	Scheduling	Batteries	Storage and File Management	HTTP / REST APIs	User Guides
Q3 – Privacy Expectations	Resilience	EV Charging	Voice and Speech Processing	Matter Protocol	References
Q4 – Network and Protocol Fit	Energy	Generators	Firmware	MQTT Protocol	Command References
Q5 – Skill and Effort	Integration	Solar / PV	Cloud OTA	NFC Tags and QR Codes	Pinout Diagrams
Q6 – Save and Continue	Networks	Lighting	Home Assistant Managed OTA	Z-Wave Protocol	Programming
Fix It	Entertainment	Bulbs & Fixtures	Manual Update	Zigbee Protocol	Setup
Q1 – Observed Symptom	Audio	Dimmers	Protocol OTA	Servers	Core Applications
Q2 – Affected Scope	Media	LED Strips	Platforms	Home Assistant	Infrastructure
Q3 – Steps Already Tried	TV & Gaming	Scenes & Effects	Home Automation Platforms	Home Lab Server	Platforms
Q4 – Fix and Prevention	Development	Media	Integration and API Platforms	Network Attached Storage (NAS)	Troubleshooting
Q5 – Save and Continue		Receivers & Amps	Smart Home Ecosystems	Web Servers	Firmware & Recovery
Inspired Design		Speakers	Voice Assistant Platforms		Maintenance
Q1 – Problem to Solve		Streaming Devices			
		TVs & Displays			
		Network			
		Security			
		Access Control			
		Alarm Panels			
		Locks			
		Sirens			
		Sensors			
		Air Quality			
		Contact			
		Diagnostics			

Launch edition: 2026-04-15

CORE	USE CASES	DEVICES	SOFTWARE	INFRASTRUCTURE	GUIDES
Guides	Safety	Climate	Applications	AI Infrastructure	AI
HasMaster Themes and Plugins	Environment	Fans	AI and Assistants	Image and Video Generation	AI Servers
Journeys	Security	Heaters	Backup and Recovery	Inference Engines	Voice Assistants
Automate	Surveillance	Humidifiers & Dehumidifiers	Developer Tools	Vision and Detection	DIY Projects
Q1 - Automation Goal	Convenience	Thermostats & HVAC	Document and Knowledge Management	Voice	3D Printing
Q2 - Automation Inputs and Outputs	Automations	Water Heaters	Fitness and Wellness	Networks	Custom Projects
Q3 - Automation Tools	Device Control	Covers	Media and Entertainment Management	Bluetooth / BLE Networks	Electronics & Wiring
Q4 - Validation and Fallbacks	Voice Control	Blinds & Shades	Network and Infrastructure Management	Thread Mesh Networks	HASMaster
Q5 - Save and Continue	Organization	Garage Doors	Productivity and Scheduling	Wi-Fi and Wired Network	Prompt Library
Define Constraints	Content	Gates	Recipes and Meal Planning	Z-Wave Mesh Networks	Structure
Q1 - Ownership and Budget	Fitness	Energy & Storage	Security and Identity Management	Zigbee Mesh Networks	Theme and Plugin Overview
Q2 - Platform Preferences	Scheduling	Batteries	Storage and File Management	Protocols	User Guides
Q3 - Privacy Expectations	Resilience	EV Charging	Voice and Speech Processing	HTTP / REST APIs	References
Q4 - Network and Protocol Fit	Energy	Generators	Firmware	Matter Protocol	Command References
Q5 - Skill and Effort	Integration	Solar / PV	Cloud OTA	MQTT Protocol	Pinout Diagrams
Q6 - Save and Continue	Networks	Lighting	Home Assistant Managed OTA	NFC Tags and QR Codes	Programming
Fix It	Entertainment	Bulbs & Fixtures	Manual Update	Z-Wave Protocol	Setup
Q1 - Observed Symptom	Audio	Dimmers	Protocol OTA	Zigbee Protocol	Core Applications
Q2 - Affected Scope	Media	LED Strips	Platforms	Servers	Infrastructure
Q3 - Steps Already Tried	TV & Gaming	Scenes & Effects	Home Automation Platforms	Home Assistant	Platforms
Q4 - Fix and Prevention	Development	Media	Integration and API Platforms	Home Lab Server	Troubleshooting
Q5 - Save and Continue		Receivers & Amps	Smart Home Ecosystems	Network Attached Storage (NAS)	Firmware & Recovery
Inspired Design		Speakers	Voice Assistant Platforms	Web Servers	Maintenance
Q1 - Problem to Solve		Streaming Devices			
Q2 - Priority Areas		TVs & Displays			
Q3 - Relevant		Network			
		Security			
		Access Control			
		Alarm Panels			
		Locks			
		Sirens			
		Sensors			
		Air Quality			
		Contact			
		Diagnostics			
		Electrical			
		Energy			

HASMaster taxonomy view

What HASMaster Is

HASMaster is a self-hosted WordPress platform for designing, operating, and maintaining home automation knowledge. It combines a block theme, a structured content plugin, household workspaces, journey pages, and optional AI guidance.

Tagline: Life Automated.

Why It Exists

Home automation is powerful but fragmented. People buy devices before they have a design, follow disconnected forum advice, and lose the operating knowledge needed to maintain the system later.

HASMaster gives the work a structure:

- Design before buying.
- Organize content by outcome.
- Connect devices, software, guides, and infrastructure.
- Give households a private working library.
- Use AI where it helps, without giving up control of the site.

The Lifecycle

Lifecycle	Meaning
Design	Define goals, constraints, and the system you are building toward
Operate	Run the system with clear content, compatible components, and guided workflows
Maintain	Keep documentation, backups, prompts, and component knowledge current

SCORE

HASMaster organizes use cases by the benefit they deliver.

Letter	Benefit	Example
S	Safety	Detect risk and alert the household
C	Convenience	Reduce repeated manual actions
O	Organization	Lower household mental load
R	Resilience	Handle outages and failure states
E	Entertainment	Improve comfort, media, and delight

What Buyers Get

Package	Includes
Theme Only	Block theme, templates, patterns, directory UI, taxonomy landing UI, journey navigation UI
Theme + Plugin	Theme plus six content types, taxonomies, household library, account workspace, admin tools, pipelines, and backups
Full Bundle	Theme + Plugin plus AI assistant, journey-aware routing, stage prompts, BYOK endpoint support, and prompt governance

Core Capabilities

HASMaster

[Videos](#)
[Journeys](#)
[Use Cases](#)
[Infrastructure](#)
[Software](#)
[Devices](#)
[Guides](#)
[About](#)

[Guides](#) > HASMaster, Theme and Plugin Overview

HASMaster Plugin — Marketing Page

Log in with a household-enabled account to save this item to your home library.

Last updated: 2026-03-08. Structure follows: wp-marketing-docs-best-practices.md (15-section sequence).

HASMaster Plugin is the operating layer for the HASMaster environment: it adds the structured content model (custom post types + taxonomies), the guided journey engine, and the admin tools that keep everything consistent over time. For the presentation layer (styling, layouts, directory UX), see: HASMaster Block Theme.

Section 1: Navigation Bar

[Links: Features](#) | [How it Works](#) | [Pricing](#) | [Docs](#) | [Support](#)
Primary CTA: [Get HASMaster Plugin]
Behavior: Sticky on scroll

Section 2: Hero

Headline:

The Schema + Journey Engine for Home Automation Knowledge

Subheadline:

Turn "I want to automate X" into a structured scenario with constraints, components, a confirmed recommendation, and setup-ready documentation.

Primary CTA: [Get the Plugin] **Secondary CTA:** [View Live Demo]

Hero visual: [Screenshot: journey step page with progress + confirmed outcomes]

Trust strip (below hero):

- Purpose-built content types for home automation (not generic blogging)
- Taxonomy spine that stays consistent as your content grows

HASMaster plugin page

Theme

- Gutenberg-native block theme.
- Directory filters, tags, and result headers.
- Journey step navigation.
- Taxonomy landing layouts.
- Purpose-built patterns for guides, devices, applications, use cases, infrastructure, and journey pages.

Core Plugin

- Six public content types: devices, use cases, guides, software, infrastructure, and videos.
- Shared and type-specific taxonomy spine.
- Compatibility badges.
- Household ownership and roles.
- Account and My Home workspaces.
- Content pipelines.

- Maintenance tools.
- Runtime registries for prompt and source health.

AI Assistant

- AI chat blocks and frontend assistant.
- Journey-aware routing across six stages.
- Stage-specific prompt templates.
- Bring-your-own-key provider model.
- Compatible with OpenAI-style endpoints, including LiteLLM and Ollama gateways.
- Prompt governance and provenance expectations.

Journeys

The screenshot shows a web page for HASMaster. The navigation bar includes links for Videos, Journeys, Use Cases, Infrastructure, Software, Devices, Guides, and About. The main heading is "HASMaster Theme + Plugin Demo — Get Inspired Journey".

Purpose

This page demonstrates how the HASMaster block theme, plugin control plane, orchestrator, and agent workflow guide a user from frustration to a confirmed automation goal.

Live Journey

Open Get Inspired Journey

Journey Sequence

1. Frustration capture and confirmation
2. Focus area selection (S.C.O.R.E) and confirmation
3. Inspiration option selection and confirmation
4. Goal capture and confirmation
5. Final summary and Save Progress to user profile

Architecture Flow (Mermaid)

```

flowchart TD
  U[User in Get Inspired page] --> UI[Block Theme UI + Assistant Composer]
  UI --> ORCH[ai-assistant/chat Thin Orchestrator/]
  ORCH --> PROMPTS[Runtime Prompt Cache\nwp-content/hasmaster-runtime-cache/prompts]
  ORCH --> REG[Runtime Source Registry Cache\nwp-content/hasmaster-runtime-cache/source-registry]
  ORCH --> LG[LangGraph Worker\n/workflows/hasmaster-render]
  LG --> PHASE[Phase Router\nfrustration/focus/inspiration/goal]
  PHASE --> LLM[LiteLLM / Model]
  PHASE --> RULES[Deterministic Transition Rules\nconfirm_summary + select_inspiration_options]
  PHASE --> LINKS[Registry-bound Inspiration Links\n(no invented URLs)]
  LG --> ORCH
  ORCH --> UI
  UI --> SAVE[Save Progress action]
  SAVE --> PROFILE[User Profile persistence]
  subgraph N8N_LangChain[LangChain + n8n execution model]
    ORCH --> N8N[n8n workflows (task orchestration)]
    N8N --> LG
  end
  
```

HASMaster journey page

Journeys turn browsing into a guided path:

Inspired Design -> Define Constraints -> Select Components -> Set Up -> Automate -> Fix It

Each stage can combine examples, AI guidance, user confirmations, and saved scenario state.

Content and Taxonomy

HASMaster separates content by job:

- Use cases explain the outcome.
- Devices explain hardware fit.
- Software explains application fit.
- Infrastructure explains operating layers.
- Guides explain how to do the work.
- Videos carry video-first stories and demos.

Taxonomy connects the pieces so users can browse by capability, domain, compatibility, setup difficulty, platform, and benefit.

Privacy and Operating Model

HASMaster is self-hosted. Content, households, user data, and AI sessions live on the buyer's WordPress infrastructure. The AI add-on is BYOK, so token usage and provider choice stay with the site owner.

Best Fit

HASMaster is strongest for:

- Home automation enthusiasts who want structure before buying more gear.
- Creators who publish guides, devices, use cases, and videos.
- Smart home consultants who need repeatable client knowledge bases.
- Home lab operators who want a self-hosted, AI-ready publishing system.

Short Launch Message

HASMaster turns home automation knowledge into a structured, searchable, household-aware WordPress platform. Start with the theme for presentation, add the plugin for the full content model, and use the full bundle when you want AI-guided journeys on your own infrastructure.