



[Manuale introduttivo: guida ai manuali, vision e roadmap.]

Sistema Domotico
su piattaforma
Arduino



AUTORE: ANDREA LANDO

REVISIONE 1.0.0

DATA ULTIMA REVISIONE: 09.05.2026

Sommario

Prefazione	3
La Missione del Progetto GitHub DomoManager.....	3
Una Visione Condivisa	4
Guida ai Documenti della Suite DomoManager	5
Struttura, Contenuti e Percorso di Lettura	5
1. Analisi dei rischi legati alle tecnologie	5
2. Comparazione tra Domotica filare, Domotica Wireless e raccomandazioni per impianti ibridi	5
3. Comparazione tra Server Domotici locali	6
4. Architettura, funzionalità e panoramica del sistema	7
5. Librerie Dispositivi	7
6. Classi di Base	8
7. Configurazioni di Base.....	9
Percorso di Lettura Consigliato.....	10
Conclusione	10
Governance della Community GitHub – Progetto DomoManager.....	11
Struttura, Ruoli, Processi e Visione Collaborativa	11
Introduzione	11
Conclusione	14
Roadmap Strategica 2026–2030.....	16
Introduzione	16
2026 – Consolidamento delle Basi e Stabilità Operativa.....	16
2027 – Espansione Funzionale e Architettura Ibrida	16
2028 – Ecosistema di Plugin e Modularità Avanzata	17
2029 – Automazione Intelligente e Manutenzione Predittiva	17
2030 – Scalabilità, Certificazioni e Adozione Professionale	18
2031 – Maturità, Ecosistema Completo e Sostenibilità a Lungo Termine.....	18
Conclusione	19
Disclaimer Legale – Limitazione di Responsabilità	20

Prefazione

Negli ultimi anni il settore della domotica ha vissuto una trasformazione radicale: da un insieme di soluzioni frammentate, spesso orientate al mercato consumer, si è evoluto verso un ecosistema più maturo, più integrato e sempre più vicino ai criteri dell'automazione industriale. La diffusione di dispositivi intelligenti, protocolli wireless avanzati, piattaforme software modulari e sistemi di supervisione sempre più potenti ha reso possibile la realizzazione di impianti complessi, affidabili e capaci di migliorare la qualità della vita, l'efficienza energetica e la sicurezza degli edifici.

Tuttavia, questa crescita non è priva di sfide: la domotica moderna richiede continuità di servizio, resilienza ai guasti, sicurezza dei dati, sostenibilità nel tempo, riduzione dell'impatto RF e una progettazione che tenga conto non solo delle funzionalità, ma anche della salute degli ambienti e delle persone e non per ultimo, della sicurezza e della privacy. In un contesto in cui molti sistemi sono nati per il mercato consumer, spesso dipendenti dal cloud o da ecosistemi chiusi, diventa sempre più evidente la necessità di soluzioni professionali, aperte, affidabili e realmente controllabili (e limitabili) dall'utente.

È proprio da questa esigenza che nasce **DomoManager**, un progetto pubblico e aperto, sviluppato su GitHub, che ambisce a diventare un punto di riferimento per chi desidera una domotica moderna, sostenibile e progettata con criteri ingegneristici. DomoManager parte da una base solida: un controller embedded real-time, privo di sistema operativo, con logiche deterministiche, ridondanza nativa e un'architettura pensata per garantire continuità operativa anche in condizioni critiche. Ma la sua forza non risiede solo nella tecnologia: risiede nella **community**.

La Missione del Progetto GitHub DomoManager

DomoManager non è solo un software: è un progetto aperto, collaborativo e orientato alla sostenibilità tecnica. La sua missione è chiara:

1. Democratizzare la domotica professionale

Rendere accessibili tecnologie tipiche dell'automazione industriale — affidabilità, determinismo, ridondanza — anche a installatori, maker e progettisti che non dispongono di budget elevati o infrastrutture proprietarie.

2. Creare un ecosistema sostenibile nel tempo

Il progetto nasce per durare.

Non dipende da servizi cloud esterni, non richiede licenze obbligatorie, non vincola l'utente a hardware proprietario.

La sostenibilità è intesa come:

- sostenibilità economica (costi prevedibili e contenuti),
- sostenibilità tecnica (manutenzione semplice, documentazione chiara),
- sostenibilità ambientale (riduzione dell'impatto RF, efficienza energetica),
- sostenibilità operativa (continuità di servizio e ridondanza).

3. Bilanciare tecnologia, salute e benessere

DomoManager promuove un approccio progettuale che riduce l'esposizione a emissioni RF superflue, privilegiando soluzioni cablate e architetture ibride.

L'obiettivo non è demonizzare il wireless, ma usarlo in modo intelligente, consapevole e misurato.

4. Valorizzare la community come motore del progetto

Il repository GitHub è il cuore pulsante del progetto.

La community contribuisce con:

- codice,
- test,
- documentazione,
- idee,
- feedback,
- integrazioni,
- traduzioni,
- esempi di impianti reali.

DomoManager vuole diventare un progetto **costruito dalla community per la community**, dove ogni contributo migliora la qualità del sistema e ne amplia le possibilità.

5. Ridurre la complessità per installatori e tecnici

Uno dei problemi più diffusi nella domotica moderna è la complessità crescente: configurazioni frammentate, protocolli eterogenei, manutenzione difficile, aggiornamenti rischiosi.

DomoManager nasce per semplificare:

- configurazione chiara e leggibile,
- logiche deterministiche,
- manutenzione prevedibile,
- architettura modulare,
- documentazione professionale.

6. Offrire una piattaforma realmente aperta e trasparente

Il codice è pubblico, leggibile, verificabile.

Non esistono componenti nascosti, telemetrie occulte o dipendenze non dichiarate.

Ogni scelta architetturale è documentata e motivata.

Una Visione Condivisa

Questa prefazione introduce quindi un percorso strutturato: prima l'analisi dei rischi, poi la comparazione delle tecnologie, quindi la valutazione dei sistemi domotici locali e infine la presentazione di DomoManager come risposta concreta alle esigenze emerse.

Il documento non si limita a confrontare piattaforme: propone una visione più ampia, una domotica progettata con criteri professionali, attenta alla salute degli ambienti, alla sostenibilità, alla sicurezza dei dati e alla qualità della vita delle persone.

DomoManager vuole essere un progetto che non solo funziona, ma che può essere **capito, mantenuto, esteso e adottato** da chiunque abbia competenze tecniche di base.

Un progetto che cresce grazie alla community e che vuole diventare un punto di riferimento per chi cerca una domotica che non sia solo “smart”, ma anche **solida, sicura e responsabile**.

Guida ai Documenti della Suite DomoManager

Struttura, Contenuti e Percorso di Lettura

Questa guida presenta in modo chiaro e strutturato tutti i documenti che compongono la suite tecnica di DomoManager. Ogni documento è progettato per essere **autonomo, modulare e riutilizzabile**, ma insieme formano un percorso completo che va dall'analisi dei rischi fino agli esempi pratici di implementazione.

La suite è pensata per:

- progettisti,
- installatori,
- integratori,
- sviluppatori,
- utenti avanzati,
- membri della community GitHub.

1. Analisi dei rischi legati alle tecnologie

Documento: DomoManager - *Analisi dei rischi legati alle tecnologie.pdf*

Contenuti principali

- Analisi dei rischi tecnologici (hardware, software, rete, cloud).
- Tabella cronologica incidenti 2020–2025.
- Rischi per la salute e impatto RF.
- Rischi di interruzione di servizio.
- Classificazione dei dispositivi in base alla resilienza.

A chi è rivolto

- A chi vuole capire come nasce l'esigenza di Domo Manager.
- Progettisti, installatori, responsabili tecnici.
- Chi deve valutare l'affidabilità di un impianto, chi deve proporre una tipologia di impianto ad un cliente.

Quando usarlo

- In fase di progettazione preliminare.
- Per valutare protocolli, architetture e dispositivi.

2. Comparazione tra Domotica filare, Domotica Wireless e raccomandazioni per impianti ibridi

Documento: DomoManager - *Sistemi ibridi domotici.pdf*

Contenuti principali

- Principi generali: cablato vs wireless
- Analisi per tipologia di dispositivo
- Rischi specifici delle tecnologie radio
- Rischi specifici dei sistemi filari
- Linee guida per combinare filare e wireless
- Matrice rischio–impatto
- Strategie di riduzione RF
- Scalabilità, sicurezza e manutenzione
- Best practice per impianti professionali
- Conclusione tecnica e raccomandazioni

A chi è rivolto

- Progettisti e installatori
- Integratori di sistemi
- Chi deve scegliere la tecnologia più adatta
- Chi realizza impianti complessi o multi-ambiente

Quando usarlo

- Durante la definizione dell’infrastruttura fisica dell’impianto
- In fase di progettazione esecutiva
- Per valutare l’impatto RF e la sostenibilità
- Per ottimizzare costi, affidabilità e manutenzione

3. Comparazione tra Server Domotici locali

Documento: DomoManager - *Comparazione tra Server Domotici locali.pdf*

Contenuti principali

- Analisi tecnica di:
 - DomoManager
 - Home Assistant
 - OpenHAB
 - Node-RED
 - Hubitat
- Rating finale basato su 14 parametri tecnici.
- Interpretazione dei risultati.
- Ruoli ideali per ogni piattaforma.

A chi è rivolto

- Progettisti, integratori, utenti avanzati.
- Chi deve scegliere il “cervello” dell’impianto.

Quando usarlo

- Durante la selezione della piattaforma di controllo.
- Per confrontare soluzioni consumer vs professionali.

4. Architettura, funzionalità e panoramica del sistema

Documento: DomoManager – Architettura, funzionalità e panoramica del sistema.pdf

Contenuti principali

- Architettura interna del controller.
- Moduli, protocolli e interfacce.
- Logiche deterministiche.
- Configurazione e messa in servizio.
- Manutenzione e diagnostica.
- Integrazione con sistemi esterni.

A chi è rivolto

- Installatori, integratori, sviluppatori.
- Chi deve configurare o mantenere un impianto.

Quando usarlo

- Durante l'installazione.
- Per la manutenzione periodica.
- Per sviluppare integrazioni o plugin.

5. Librerie Dispositivi

Documento: DomoManager – Librerie dispositivi.pdf

Contenuti principali

- Panoramica delle librerie integrate Descrizione delle librerie built-in disponibili nel core di DomoManager: I/O digitali, sensori analogici, bus cablati, Modbus, attuatori, sensori virtuali.
- Struttura delle librerie dispositivo Modello architetturale comune: reader, writer, init, update deterministico, diagnostica.
- Configurazione delle librerie Come definire dispositivi, parametri, mapping e binding tramite FrontendConfig.
- Estensione del sistema con nuove librerie Linee guida per sviluppare librerie aggiuntive mantenendo coerenza, modularità e sostenibilità.
- Best practice operative Approcci consigliati per garantire stabilità, leggibilità, assenza di hardcoded values e diagnostica affidabile.
- Template di libreria Struttura generica riutilizzabile per creare nuove librerie compatibili con il Device Manager.

A chi è rivolto

- Installatori Che devono integrare dispositivi certificati, sensori, attuatori o moduli bus.
- Progettisti Che devono valutare compatibilità, architetture e modalità di integrazione hardware.
- Sviluppatori e utenti avanzati Che desiderano estendere il sistema con nuove librerie o adattare quelle esistenti.
- Membri della community GitHub Che contribuiscono con driver, moduli o dispositivi aggiuntivi.

- Produttori che desiderano avere i propri dispositivi inseriti nella libreria built-in di DomoManager

Quando usarlo

- Per integrare nuovi dispositivi Quando si aggiungono sensori, attuatori, moduli bus o dispositivi non ancora supportati.
- Per replicare configurazioni collaudate Utilizzando librerie built-in già testate e documentate.
- Per progettare architetture hardware Valutando quali librerie usare per ingressi, uscite, sensori, protocolli cablati o wireless.
- Per formare nuovi tecnici o sviluppatori Introducendo la struttura modulare delle librerie e il loro ciclo di vita.
- Per contribuire al progetto Creando nuove librerie secondo le linee guida ufficiali.

6. Classi di Base

Documento: DomoManager – Classi di Base.pdf

Contenuti principali

- Architettura modulare a tre colonne — FrontendConfig, DMFrontendEngine e DomoManager costituiscono la pipeline principale di orchestrazione, definendo configurazione, logica e runtime industriale. *«FrontendConfig è la sorgente unica di verità... DMFrontendEngine è il coordinatore del ciclo... DomoManager è il runtime industriale».*
- Motori funzionali indipendenti — WeatherEngine, PowerEngine, PowerSupervisorEngine, HVACEngine, SecurityEngine, WatchEngine, AveragesEngine, AutomationEngine. Ogni motore opera con update(), legge/scrive nel Buffer e non ha dipendenze dirette.
- Buffer come Single Source of Truth — struttura centrale con change-tracking, routing dinamico, aree virtuali, diagnostica e atomicità delle operazioni. *«Il Buffer è il fulcro dell'intera architettura... tutto ruota attorno a esso».*
- Astrazione hardware completa — GenericSensorReader, DMWiredSensors, DeviceProfiles, DMPLC e ModbusManager permettono di integrare qualsiasi dispositivo senza modificare il core.
- DMAEE e AEEProtocol — framework di scambio dati tipizzato, con change-tracking, scaling, minDelta, serializzazione JSON, integrazione con Bridge, MQTT e automazioni.
- Servizi infrastrutturali — DMFncs, TimeManager, OwnerManager, IpManager, Diagnostic, Watchdog, RS485Node, HotStandbyManager: garantiscono robustezza, sicurezza e comportamento deterministico.
- Classi di attuazione fisica — Heater, Valve, Pump, Shutter, IrrigationSystem: driver deterministici con state machine, timeout, protezioni e callback verso DomoManager.

A chi è rivolto

- Sviluppatori di motori funzionali che devono comprendere ciclo di vita, update deterministico e interazione con il Buffer.
- Integratori hardware e protocolli che lavorano con Modbus, RS485, DeviceProfiles, GenericSensorReader e DMPLC.
- Architetti software che valutano estendibilità, separazione dei livelli, sicurezza e robustezza del runtime.

Quando usarlo

- Durante la progettazione di nuovi moduli o motori che devono integrarsi nel ciclo deterministico.
- Per estendere il sistema con nuovi dispositivi tramite DeviceProfiles, GenericSensorReader o DMPLC.
- Per comprendere le fondamenta architetture prima di intervenire su automazioni, buffer, protocolli o pipeline di orchestrazione.
- Per attività di diagnostica avanzata basate su DMLogger, Diagnostic, Watchdog e change-tracking del Buffer.

7. Configurazioni di Base

Documento: DomoManager – Configurazioni di base.docx **Contenuti principali**

- **Architettura generale del sistema** — descrizione dei tre livelli fondamentali: Main, FrontendEngine e Setup.
- **Pipeline Modbus TCP** — flusso di comunicazione tra dispositivi, buffer e automazioni.
- **Mappatura dispositivi** — registrazione deterministica dei moduli Modbus e validazione diagnostica.
- **Definizione del buffer** — struttura dati centrale con aree fisiche e virtuali, soglie, toggle e split-out.
- **Creazione automazioni** — regole, scene, sequenze e automazioni dinamiche.
- **Appendice tecnica** — strutture dati, flussi operativi e diagnostica integrata.

A chi è rivolto

- Sviluppatori e integratori che devono comprendere la logica interna del sistema.
- Installatori che configurano dispositivi e buffer.
- Progettisti che analizzano la pipeline Modbus e la diagnostica.

Quando usarlo

- Durante la configurazione iniziale del firmware.
- Per aggiungere nuovi dispositivi o automazioni.
- Per validare la coerenza tra hardware e buffer.

Percorso di Lettura Consigliato

Profilo	Sequenza consigliata	Obiettivo
Principiante / Installatore base	Documento Introduttivo → Configurazioni di Base	Comprendere architettura generale e primi passi di configurazione
Installatore esperto	Configurazioni di Base → Configurazioni Estese	Approfondire diagnostica, pipeline Modbus e automazioni
Progettista	Analisi dei Rischi → Sistemi Ibridi → Comparazione Server Domotici → Configurazioni Estese → Librerie Dispositivi	Valutare architetture, protocolli, resilienza e compatibilità
Sviluppatore / Integratore	Configurazioni di Base → Configurazioni Estese → Librerie Dispositivi	Comprendere pipeline, buffer, engines e DeviceProfiles
Community GitHub / Contributor	Documento Introduttivo → Librerie Dispositivi → Configurazioni Estese	Contribuire con librerie, moduli, profili e miglioramenti

Conclusione

Questa guida permette di navigare facilmente l'intera suite documentale, garantendo un approccio modulare, professionale e coerente con la filosofia del progetto DomoManager: **chiarezza, sostenibilità, affidabilità e collaborazione aperta.**

Governance della Community GitHub – Progetto DomoManager

Struttura, Ruoli, Processi e Visione Collaborativa

Introduzione

DomoManager è un progetto pubblico, aperto e collaborativo, sviluppato su GitHub con l'obiettivo di creare una piattaforma domotica professionale, sostenibile e accessibile. Per garantire una crescita ordinata, trasparente e duratura, il progetto adotta un modello di governance chiaro, ispirato alle migliori pratiche dell'open-source moderno.

La governance non è un insieme di regole rigide, ma un **ecosistema di processi, ruoli e responsabilità** che permette alla community di contribuire in modo efficace, mantenendo al tempo stesso la qualità, la sicurezza e la coerenza del progetto.

1. Principi Fondamentali della Governance

La governance di DomoManager si basa su cinque principi cardine:

1.1 Trasparenza

Ogni decisione tecnica, ogni modifica al codice e ogni discussione rilevante avviene pubblicamente su GitHub. Nessuna parte del progetto è nascosta o gestita privatamente.

1.2 Meritocrazia Aperta

Chi contribuisce in modo costante e di qualità ottiene maggiore influenza nelle decisioni. Il merito è misurato attraverso:

- contributi tecnici,
- documentazione,
- supporto agli utenti,
- revisione del codice,
- proposte architetturali.

1.3 Sostenibilità

Ogni scelta deve essere sostenibile nel tempo:

- manutenzione semplificata,
- codice leggibile,
- dipendenze minime,
- architettura stabile,
- documentazione chiara.

1.4 Neutralità Tecnologica

Il progetto non favorisce vendor o ecosistemi proprietari. Le scelte tecniche sono basate su criteri oggettivi: affidabilità, sicurezza, scalabilità, salute, costi e competenze richieste.

1.5 Community-Driven

La community non è un accessorio: è il motore del progetto. Ogni contributo è valorizzato e ogni voce è ascoltata.

2. Struttura dei Ruoli nella Community

La governance definisce ruoli chiari, ognuno con responsabilità specifiche.

2.1 Founder & Lead Architect

Responsabilità principali:

- definizione della visione tecnica del progetto,
- approvazione delle modifiche architetture,
- gestione della roadmap,
- supervisione della qualità del codice,
- coordinamento dei maintainer.

Il Lead Architect garantisce che il progetto rimanga coerente con la sua missione originaria: **una domotica professionale, sostenibile, aperta e affidabile.**

2.2 Maintainer

Sono membri della community con permessi avanzati sul repository.

Responsabilità:

- revisione delle pull request,
- gestione delle issue,
- supporto agli utenti,
- verifica della qualità del codice,
- applicazione delle linee guida di sicurezza e stile.

I maintainer sono nominati in base al merito e alla continuità dei contributi.

2.3 Contributor

Chiunque contribuisca al progetto tramite:

- codice,
- documentazione,
- test,
- traduzioni,
- idee,
- segnalazioni di bug,
- miglioramenti alla UX o alla configurazione.

Non richiede permessi speciali: basta aprire una issue o una pull request.

2.4 Community Tester

Ruolo fondamentale per la stabilità del progetto.

Responsabilità:

- testare nuove versioni,
- verificare la compatibilità con hardware reale,
- segnalare regressioni,

- proporre miglioramenti.

2.5 Utenti Finali e Installatori

Anche chi non contribuisce direttamente al codice partecipa alla governance tramite:

- feedback,
- richieste di funzionalità,
- segnalazioni di problemi reali,
- condivisione di casi d'uso.

3. Processi di Contributo

La governance definisce processi chiari per mantenere ordine e qualità.

3.1 Issue Tracking

Ogni problema, proposta o discussione parte da una issue GitHub. Le issue sono classificate in:

- **bug**,
- **feature request**,
- **enhancement**,
- **documentation**,
- **discussion**,
- **security** (gestite in modo riservato).

3.2 Pull Request

Ogni modifica al codice deve passare attraverso una PR.

Requisiti:

- codice leggibile e commentato,
- test (quando applicabile),
- documentazione aggiornata,
- motivazione tecnica chiara,
- rispetto delle linee guida di stile.

3.3 Code Review

Ogni PR deve essere approvata da almeno un maintainer. Per modifiche critiche, è richiesta l'approvazione del Lead Architect.

3.4 Roadmap e Versioning

La roadmap è pubblica e aggiornata periodicamente. Il versioning segue un modello semantico:

- **MAJOR** – cambiamenti architetturali,
- **MINOR** – nuove funzionalità,
- **PATCH** – bugfix e ottimizzazioni.

4. Linee Guida per la Qualità del Codice

Per garantire sostenibilità e manutenzione nel tempo:

- evitare complessità inutili,
- preferire soluzioni deterministiche,
- documentare ogni modulo,
- mantenere compatibilità retroattiva quando possibile,
- evitare dipendenze superflue,
- privilegiare architetture modulari,
- sviluppare codice “generico”, parametrizzabile dall’esterno.

5. Visione a Lungo Termine della Community

La governance non è statica: evolve con il progetto.

Obiettivi futuri:

5.1 Creare un ecosistema di plugin ufficiali e community

Per integrare nuovi protocolli, dispositivi e logiche.

5.2 Costruire una documentazione professionale multilivello

- onboarding per principianti,
- manuali tecnici per installatori,
- reference per sviluppatori.

5.3 Sviluppare un programma di certificazione dei contributori

Per garantire qualità e continuità.

5.4 Organizzare incontri periodici

- call tecniche,
- workshop,
- hackathon,
- revisioni roadmap.

5.5 Garantire la sostenibilità economica del progetto

Attraverso:

- donazioni volontarie,
- crowdfunding,
- sponsor etici,
- partnership non vincolanti.

Conclusione

La governance della community GitHub di DomoManager è progettata per garantire:

- **ordine,**
- **qualità,**
- **trasparenza,**
- **sostenibilità,**

- **partecipazione,**
- **crescita condivisa.**

DomoManager non è solo un progetto software: è una visione comune, costruita insieme, per una domotica più professionale, più sana, più sostenibile e più libera.

Roadmap Strategica 2026–2030

Evoluzione Tecnica, Comunitaria e Architettuale del Progetto DomoManager

Introduzione

La roadmap 2026–2030 definisce la visione a medio-lungo termine del progetto DomoManager, delineando gli obiettivi tecnici, organizzativi e comunitari che guideranno lo sviluppo della piattaforma nei prossimi anni. Non è un elenco rigido di funzionalità, ma un **percorso strategico**, costruito per garantire:

- sostenibilità tecnica,
- continuità operativa,
- qualità del codice,
- crescita della community,
- riduzione della complessità,
- attenzione alla salute e all’impatto RF,
- adozione professionale su larga scala.
- abbassamento dei costi di implementazione.
- utilizzo consapevole di dispositivi certificati e standard.

La roadmap è pubblica, aperta ai contributi e soggetta a revisione annuale.

2026 – Consolidamento delle Basi e Stabilità Operativa

Obiettivi principali

- Rafforzare l’architettura core embedded real-time.
- Stabilizzare le funzionalità esistenti.
- Migliorare la documentazione e l’onboarding.

Milestone 2026

- **Refactoring del core** per modularità e leggibilità.
- **Documentazione multilivello:**
 - guida rapida,
 - manuale tecnico,
 - reference API.
- **Test suite automatizzata** per garantire stabilità nelle release.
- **Prima release LTS (Long Term Support).**
- **Creazione del sito ufficiale** con documentazione integrata.
- **Avvio del programma Community Tester.**

Focus 2026

Stabilità, chiarezza, basi solide. Il progetto deve essere facile da capire, installare e mantenere.

2027 – Espansione Funzionale e Architettura Ibrida

Obiettivi principali

- Estendere le capacità del sistema senza comprometterne la semplicità.
- Migliorare l'integrazione con protocolli cablati e wireless.
- Ridurre ulteriormente l'impatto RF.

Milestone 2027

- **Supporto avanzato per bus cablati** (RS485, Modbus, Ethernet).
- **Ottimizzazione del layer wireless** con strategie di riduzione RF.
- **Dashboard locale minimale** per diagnostica e manutenzione.
- **Sistema di logging avanzato** con buffer circolare e diagnostica predittiva.
- **Integrazione con sistemi di supervisione esterni** (Home Assistant, Node-RED, OpenHAB).
- **Prima versione del configuratore visuale** (beta).

Focus 2027

Potenziare senza complicare. L'utente deve avere più possibilità, non più complessità.

2028 – Ecosistema di Plugin e Modularità Avanzata

Obiettivi principali

- Rendere DomoManager estensibile tramite plugin ufficiali e community.
- Separare nettamente core, moduli e integrazioni.

Milestone 2028

- **Plugin System v1** con sandboxing e API stabili.
- **Repository ufficiale dei plugin** (GitHub + registry).
- **SDK per sviluppatori** con esempi e template.
- **Integrazione con protocolli industriali avanzati** (BACnet, KNX via gateway).
- **Ottimizzazione della ridondanza hot-standby**.
- **Strumenti di migrazione da sistemi consumer** (Zigbee, Wi-Fi, Thread).

Focus 2028

Creare un ecosistema. DomoManager deve diventare una piattaforma, non solo un controller.

2029 – Automazione Intelligente e Manutenzione Predittiva

Obiettivi principali

- Introdurre automazioni intelligenti senza dipendere dal cloud.
- Migliorare la resilienza e la manutenzione predittiva.

Milestone 2029

- **Motore di automazione avanzato** con:
 - condizioni complesse,
 - scenari dinamici,
 - logiche temporali evolute.
- **Analisi predittiva locale** basata su pattern (no cloud).

- **Sistema di notifiche locale e remota** (MQTT, Telegram, email).
- **Ottimizzazione energetica integrata.**
- **Strumenti per audit e sicurezza** (firmware signing, integrity check).
- **Supporto per cluster di controller** (multi-node).

Focus 2029

Intelligenza locale, zero dipendenze esterne. La casa deve funzionare anche senza Internet.

2030 – Scalabilità, Certificazioni e Adozione Professionale

Obiettivi principali

- Portare DomoManager a livello professionale certificabile.
- Supportare installazioni complesse e multi-edificio.

Milestone 2030

- **Certificazioni elettriche e industriali** (ove applicabile).
- **Supporto per impianti multi-sito** con sincronizzazione sicura.
- **Strumenti per installatori professionali:**
 - configuratore avanzato,
 - report automatici,
 - diagnostica remota.
- **Manuali professionali** per progettisti e integratori.
- **Partnership con produttori hardware open-friendly.**

Focus 2030

Portare DomoManager nel mondo professionale. Non solo hobby, ma impianti reali, affidabili e certificabili.

2031 – Maturità, Ecosistema Completo e Sostenibilità a Lungo Termine

Obiettivi principali

- Rendere DomoManager un progetto maturo, stabile e sostenibile.
- Consolidare la community come fondamento del progetto.

Milestone 2031

- **DomoManager 3.0 – Release di maturità.**
- **Ecosistema completo di plugin ufficiali e community.**
- **Governance evoluta con comitato tecnico.**
- **Modello di sostenibilità economica** basato su:
 - donazioni,
 - crowdfunding,
 - sponsor etici,
 - servizi opzionali (non obbligatori).
- **Strumenti di formazione certificata** per installatori e sviluppatori.

- **Supporto a lungo termine (LTS) garantito.**

Focus 2031

Un progetto maturo, stabile, sostenibile e guidato dalla community. DomoManager deve diventare un riferimento nel panorama della domotica locale.

Conclusione

La roadmap 2026–2031 non è un semplice elenco di funzionalità: è una **visione strategica**. DomoManager vuole diventare:

- **affidabile come un PLC,**
- **flessibile come un framework open-source,**
- **sostenibile come un progetto community-driven,**
- **sicuro e sano per gli ambienti in cui viviamo,**
- **accessibile a installatori, progettisti e appassionati.**

È un percorso ambizioso, ma realistico, costruito su basi solide e sulla forza della community.

Disclaimer Legale – Limitazione di Responsabilità

Il software **DomoManager**, la relativa documentazione tecnica e tutti i moduli associati (inclusi ma non limitati a: *DM.hpp*, *DMFrontend.hpp*, *DMSetup.hpp*, *DMDiagnostic.hpp*, automazioni, configurazioni e profili hardware) sono forniti “**così come sono**”, senza alcuna garanzia espressa o implicita.

L'autore e i collaboratori **non assumono alcuna responsabilità** per:

- danni diretti o indiretti
- malfunzionamenti
- perdita di dati
- interruzioni di servizio
- danni a cose, impianti o persone
- utilizzo improprio o non conforme del software
- modifiche non autorizzate al codice o alla configurazione
- integrazioni con dispositivi di terze parti non certificati

L'utente accetta che:

L'installazione, la configurazione e l'uso del software avvengono sotto la propria esclusiva responsabilità.

Il sistema è progettato per applicazioni domotiche e di automazione non critiche. Non deve essere utilizzato in contesti dove un malfunzionamento potrebbe causare:

- rischi per la sicurezza personale
- danni strutturali
- interruzioni di servizi essenziali
- condizioni pericolose per persone o animali

L'utente è tenuto a:

- verificare la correttezza delle configurazioni
- testare ogni automazione prima dell'uso reale
- assicurarsi che i dispositivi collegati siano conformi alle normative vigenti
- implementare adeguate misure di sicurezza elettrica e informatica

L'autore declina ogni responsabilità derivante da:

- modifiche al codice sorgente
- utilizzo in ambienti industriali o critici
- integrazioni con hardware non supportato
- errori di cablaggio o installazione
- uso improprio delle automazioni o delle scene

L'uso del software implica l'accettazione integrale del presente disclaimer.

DOMO Manager è ciò che davvero conta:

la continuità, la sicurezza, la sostenibilità e la qualità dell'ambiente in
cui viviamo ogni giorno...