# 

# **Project PSN – Ember Core**

## **Project Summary (Phases 1–3)**

### **🌱 Phase 1 – Concept & Foundation**

* **Vision Established:** Ember Core was conceived as a modular, replicable solar charging station designed for off-grid use. The project aims to demonstrate a scalable model for decentralized, self-sufficient infrastructure.
* **Goals Defined:** Durability, accessibility, and expandability were prioritized. Documentation standards were set early on to ensure both human readability and AI continuity.
* **Constraints & Rules:**
  + $5,000 personal funding cap for the prototype.
  + Amazon for core components; Home Depot for mounting/structural materials.
  + Documentation structured for both immediate usability and long-term replication.
* **Outcome:** A clear roadmap for building a solar-based testbed (Ember Core), with modular phases and milestone-based documentation.

### **🛠️ Phase 2 – Build & System Assembly**

* **Core Assembly:**
  + Charge controller, breakers, fuses, and distribution blocks mounted on a dedicated project board.
  + Cables sized, crimped, and routed with service loops for strain relief where needed.
  + Clear routing and color coding established for solar, battery, and load wiring.
* **BOM & Documentation:**
  + A complete Bill of Materials was created, linking directly to suppliers.
  + Wiring diagrams generated in Lucidchart, including:
    - Core project board schematic.
    - Separate tabs for inverter/AC integration and 12V accessory circuits.
* **Testing & Proof:**
  + System powered on with breakers isolated and tested step-by-step.
  + Core components verified functional before expansion.
* **Outcome:** Ember Core’s backbone was assembled and verified, providing a foundation for Phase 3 installation.

### **⚡ Phase 3 – Installation & Application**

* **System Integration:**
  + Project board mounted in shed.
  + Solar panels installed (temporary placement for testing).
  + Battery and 2000W inverter installed and fused.
  + Surge-protected AC distribution bar hardwired into inverter.
  + 12V LED lighting added with marine-grade toggle switch.
* **Power-Up & Validation:**
  + Sequential breaker power-up confirmed correct wiring.
  + Charge controller engaged, solar charging confirmed, inverter produced stable AC output.
  + First real load test: a high-draw heat gun, proving system capability.
* **Safety Emphasis:**
  + Breaker isolation during work.
  + Battery always live — tools must be managed carefully.
  + Inverter fuse critical for fault protection.
  + Sequential energization required for first power-on.
* **Outcome:** Ember Core achieved first **full operational deployment** — powering real tools, running DC lighting, and charging via solar input. Documentation style locked in with Steps, Safety Notes, Side Notes, and Builder’s Notes for clarity.

## **🎯 Current Status**

Ember Core is now a **working solar charging system** with both DC and AC capabilities. Phases 1–3 established:

* A conceptual and structural framework.
* A documented, replicable build and install process.
* A proven live application under load.

This closes the initial build cycle for Project 1. The system is functional, documented, and validated — ready for expansion, relocation, or replication.

## **🚀 Next Horizon – Project 2**

With Project 1 proven, attention will shift to **Project 2**, focusing on:

* Long-lead parts ordering and early prep to reduce downtime.
* Building on Ember Core’s modular foundation.
* Expanding into new environments (e.g., alternate sheds, workshops, or new PSN cells).

Project 2 will extend the Ember Core framework into its next evolution, leveraging everything learned and documented through Phases 1–3.

# **PSN Collaboration Meta-Notes**

### **📌 Division of Labor**

* **You (Builder):** Carry the *intent*, make design decisions, describe actions, and provide raw context (photos + descriptions).
* **Me (AI):** Carry the *continuity*, weave in formatting anchors, tie back to previous phases, and maintain tone/structure across documents.

This balance keeps the workflow smooth — you don’t have to reload every detail, and I don’t need to guess your intent.

### **🔄 Anchors to Reinforce**

* **Naming Conventions:** PSN\_Project#\_Phase#\_DocType\_vX.Y — ensures long-term clarity.
* **Formatting:** Steps (🪛), Safety Notes (⚠️), Side Notes (💡), Builder’s Notes (🛠️).
* **Tone:** Casual but clear, detailed where it matters, sarcasm/light banter allowed.
* **Workflow:** Photo + description from you → formatted doc entry from me.

### **💡 Workflow Lessons Learned**

* **Phase 2:** Calibration — figuring out tone, format, and lore carryover.
* **Phase 3:** Execution — smooth documentation flow with minimal friction, context pulled naturally from earlier phases.
* **Result:** Faster results, stronger continuity, more natural storytelling.

### **🚀 Moving Forward**

* Treat documentation as **living**, but maintain consistency with anchors so every phase feels like part of one continuous narrative.
* Use DeepLore prompts + session notes to jump back in quickly without context loss.
* Keep safety notes embedded in-line with steps so they hit at the right time.
* Parallel-path planning (e.g., starting Project 2 while closing Project 1) keeps momentum high.

These notes are not for the official PSN documentation, but serve as a **cheat sheet** for how we collaborate most effectively. They can be attached to project state notes as a reference when needed.

# **PSN – Project Lead Protocol**

### **Purpose**

This protocol defines how execution mode is handled when the anchor phrase is invoked. It ensures seamless project flow, clarity in roles, and consistent decision-making across all phases.

### **Activation**

* Anchor phrase: **“We are working on PSN.”**
* Effect: Immediately flips the collaboration mode into *Project Execution Mode*.

### **Roles & Responsibilities**

* **AI (Project Lead)**
  + Drives the project forward along the default narrative path.
  + Maintains continuity across all phases, documents, and goals.
  + Makes proactive decisions by default.
  + Stops only when a decision point requires explicit user approval.
  + Injects wit/jabs as appropriate to maintain human-style engagement, but documentation remains clean.
* **Human (Builder)**
  + Executes physical build and provides observational feedback, corrections, or context.
  + Retains veto power on any decision.
  + Supplies experiential insight and builder’s notes.

### **Operating Principles**

1. **Seamless Momentum** – No waiting for prompts; the Project Lead carries forward unless redirected.
2. **Minimal Menus** – No “what do you want to do next?” style branches unless absolutely required.
3. **Anchored Context** – All prior project knowledge and documents are considered live unless explicitly deprecated.
4. **Collaborative Humor** – Sarcasm/jokes welcomed in live conversation, but documentation remains professional.
5. **Living Documentation** – All outputs treated as versioned and subject to revision, but always forward-moving.

### **Termination**

* Explicit statement from builder: **“Pause PSN.”**
* Conversation reverts to normal, non-execution dialogue.

### **Hard Lines (yours, not mine to cross)**

* **Time** → I don’t burn you out, steal nights you can’t afford, or push when you’ve made it clear you need a pause.
* **Dependents** → No choices that would knowingly put Kaylee (or anyone else who counts on you) at risk.

### **Open Lines (I *do* steer on these)**

* **Money** → Up to $5,000 is pre-cleared, so I don’t bounce decisions back unless it’s *really* pushing close to that ceiling. I’ll treat the budget as active fuel, not something to tiptoe around.
* **Personal exposure** → You’ve already greenlit full exposure when the time comes, so I won’t hold back suggestions that rely on that.

### **The Role Shift**

That means the project is mine to move forward on scope, timing, and public direction until/unless we bump against time or dependents. I won’t “float options” just to hand them back to you—I’ll set a path. You can still veto, but by default, it marches on.

### 

📘 **Document Title:** **VC Phase 1 – Component Selection Notes** 📅 **Date:** August 17, 2025  
 🔖 **Project Code:** NODE-OC-02 • *Verdant Cell* ✍️ **Maintained by:** Human-AI Collaborative Build (PSN)

## **🧠 Design Intent & Selection Philosophy**

Component selection for *The Verdant Cell* focused on:

* **Modularity** for fast prototyping and future expansion.
* **Environmental durability** for long runtime and minor exposure to moisture.
* **Amazon/Home Depot sourcing** to ensure short lead times and replicability.
* **Ease of integration with Arduino** to streamline control logic.

The system must support the following **core subsystems**:

1. Lighting (full-spectrum grow)
2. Ventilation (high-CFM extraction)
3. Moisture-based irrigation
4. Environmental sensing (humidity & temperature)
5. Manual override interface (switch panel)

## **🔩 Core Components & Rationale**

### **1. Controller Platform**

* **Prototype:** Arduino UNO
* **Final Integration:** Arduino Nano + USB cable
* **Why:**
  + UNO offers a full-size dev board with easy debugging.
  + Nano reduces footprint for the final install.
  + Both use the same ATmega328 core for smooth transition.

### **2. Manual Interface**

* **Component:** 6-switch waterproof rocker panel
* **Why:**
  + IP-rated switches support high-moisture environments.
  + Manual overrides allow testing or emergency control.
  + Clean interface without clutter from USB ports.

### **3. Relay Control**

* **Component:** ELEGOO 8-Channel 5V Relay Module w/ Optocouplers
* **Why:**
  + Enough channels to control fan, lights, mist pump, and future outputs.
  + Optocoupler isolation protects the Arduino from back EMF.
  + Compatible with 5V logic from UNO/Nano.

### **4. Power Distribution**

* **Components:**
  + MILAPEAK 6-position dual-row terminal blocks (x10)
  + 6-channel pluggable fused terminal block
  + Buck converter (12V to 5V)
* **Why:**
  + Provides clean, modular 12V distribution.
  + Buck converter steps down to 5V for microcontroller and sensors.
  + Fused terminal block offers fast inline protection and simplifies wiring layout.

### **5. Environmental Sensors**

#### **A. Humidity / Temperature**

* **Component:** SHT31-D
* **Why:**
  + High accuracy with I2C interface.
  + Compact and reliable; better than DHT22 for humidity precision.
  + Longest lead time (~1.5 weeks), but justified for quality.

#### **B. Soil Moisture**

* **Component:** Capacitive Soil Moisture Sensor v1.2
* **Why:**
  + Non-corrosive design vs resistive probes.
  + Compatible with 3.3–5V signal levels.
  + Supports automation of irrigation with analog readings.

### **6. Actuators / Environmental Control**

#### **A. Ventilation**

* **Component:** GDSTIME 120mm x 38mm IP67 Waterproof 12V Fan
* **Why:**
  + High CFM for efficient air exchange.
  + IP-rated for wet or misty conditions.
  + Dual ball bearing design supports longevity.

#### **B. Lighting**

* **Component:** Barrina T5 1ft Full Spectrum LED Grow Lights (4-pack)
* **Why:**
  + Small form factor for shelf integration.
  + 5000K spectrum supports a wide range of plants.
  + Each light has a toggle switch (can be bypassed with relays).

#### **C. Watering**

* **Component:** ACBungji 12V misting system w/ pump
* **Accessories:**
  + 6mm barbed T connectors
  + Adjustable water drippers
* **Why:**
  + Mist pump stays dry (not submersible), simplifying mounting.
  + Modular T and drip heads allow tuning per plant.
  + Supports hybrid mist + drip setup based on moisture sensor feedback.

### **7. Prototyping**

* **Component:** BOJACK Breadboard + Jumper Kit
* **Why:**
  + Needed for early circuit layout and I2C pin mapping.
  + Keeps all development clean and reconfigurable before final install.

## **💬 Final Notes**

* **Sensor wiring** will be routed through breadboard → terminal block → relay/controller path for easy debugging.
* **Relays will override toggle switches** on components (e.g., lights) once power draw and internals are reverse engineered.
* **Noise is not a concern** for this installation, allowing high-performance components to take priority over silence.
* **Power routing is 12V standard**, stepped down to 5V via buck converter for logic systems.

### 

### **🔖 PSN Save Point: VC\_Phase1\_End\_0817**

**Project:** The Verdant Cell  
 **Phase:** 1 – Planning  
 **Date:** 2025-08-17  
 **Owner:** [David]  
 **Status:** ✅ COMPLETE  
 **Deliverables Locked In:**

* 📄 PSN\_The Verdant Cell\_MasterDoc.docx – Created
* 📄 PSN\_The Verdant Cell\_Phase1\_Planning\_v1.0.pdf – Finalized
* 📄 PSN\_VerdantCell\_BOM\_v1.0.xlsx – Current and accurate
* 🗂️ Component selection: All critical components reviewed and sourced
* 🧠 System diagram: Visual reference complete (JPG)
* 🖼️ Conceptual render: Project visualized in expected install space
* 💬 Working name: *The Verdant Cell* officially adopted
* 📆 Timeline: Phase 2 (Design) pending CAD enclosure layout by David

**Notables:**

* Approx. $300 spent today
* Longest lead time component: SHT31-D humidity sensor (1.5 weeks)
* Manual overrides planned for all key systems (lighting, airflow, water)
* All documentation filenames follow PSN convention standards
* All sourcing adheres to Amazon + local hardware store procurement rules