

*34374 IoT Hardware and PCB Design*  
**Final Project - Closed Loop Stepper Control  
with Power Out Position Retention**

Ulrik Helk (s203826), Victor Hald (s183710)

November 2025

## 1 Introduction

Most consumer stepper motor solutions for motorized curtains, 3D printers, etc. are not natively customizable for a variety of wireless applications. The proposal in this report is a device that can be mounted directly to the back of a NEMA17 stepper motor. It incorporates an MCU for wireless connectivity, position control, and uses the USB-C PD (Power Delivery) interface for input power. The uniqueness of the device lies in its compact form factor, ZigBee/Thread connectivity, and the input power interface. This allows for a variety of applications for the consumer market, and mitigates the electronics expertise needed for traditional stepper motor solutions. This means that consumers who invest in the device can focus more on the physical aspects of their home projects instead of the technical electronic solution. It is an all in one solution, that can be easily integrated in existing home automation solutions, such as Philips Hue. The USB PD powered nature of the device also allows it to be easily used for portable use cases, as users can plug the device directly into external USB power banks. An example of this could be a portable sliding camera rig.

The device incorporates the following main components:

- TMC2130 - Driver
- MT6835 - Encoder
- ESP32-H2 - MCU
- USB-C 16 - Power and programming - AP33771C
- AP33771C - USB PD (power negotiation)
- TPS62933 - Buck converter TPS62933
- XC6220 - LDO
- Battery
- Battery charger IC - MCP73871

## 2 System Block Diagram

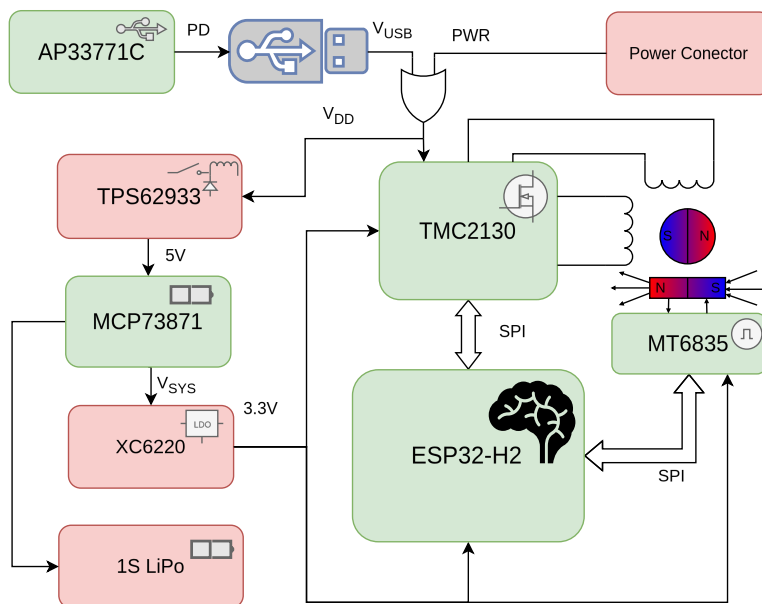


Figure 1: Block diagram of the system

### 3 Component Justification

### 3.1 MCU: ESP32-H2-MINI-1

The ESP32-H2 was selected primarily for its native support of IEEE 802.15.4, enabling Zigbee and Thread connectivity. This ensures the device is Matter compatible, a critical requirement for modern consumer home automation integration (e.g., Philips Hue, Apple Home). Furthermore, its RISC-V architecture offers sufficient processing power for closed-loop PID control algorithms while maintaining low power consumption. While an STM32L series MCU likely would provide better low power performance, the pre integrated antenna of the ESP32 outweighed the power consumption aspect. Our product is unable to reach  $\mu\text{A}$  levels in any case given the need for polling of the encoder over SPI.

### 3.2 Stepper motor driver: TMC2130

The TMC2130 is the industry standard for silent stepper operation due to its Stealth-Chop technology. For a home automation devices (e.g., curtains), acoustic noise is a primary concern. Additionally, it supports SPI configuration, allowing the MCU to dynamically adjust current and micro-stepping settings based on load, which is essential for the closed-loop system. Cheaper options like A4988 were considered, however the advanced proprietary control methods of the silent TMC2130 integrates nicely with smart home applications where noise matters and limit switches are unavailable.

### 3.3 Encoder: MT6835

To achieve true closed-loop control, a high-precision magnetic encoder is required. The MT6835 offers 21-bit resolution and an SPI interface. Unlike optical encoders, it is resistant to dust and vibration, making it a great choice for direct mounting on the motor shaft. This allows the system to recover position after a stall or power loss event. An external quadrature mechanical encoder IC was considered, but given that it would result in a less integrate-able product we chose the magnetic based encoder.

### 3.4 USB-C interface

A 16-pin USB-C connector is utilized to support both high-power delivery and data for firmware updates. Because USB-C eliminates the need for proprietary power bricks, it aligns with EU regulations and user convenience.

### 3.5 USB PD negotiator: AP33771C

The AP33771C provides an interface to request high voltage (up to 28 V) from standard USB PD (Power Delivery) chargers. This allows the NEMA17 motor to operate at higher torque and efficiency compared to standard 5 V USB power. Voltage selection is done via a side switch, allowing the user to select between 12 V and 20 V.

### 3.6 Buck converter: TPS54202

With the input voltage potentially reaching 20 V via PD, a high-efficiency step-down converter is needed to generate the 5 V rail. The TPS54202 offers high efficiency ( $> 90\%$ ) and a compact footprint, minimizing thermal generation on the PCB. We consciously oversized the buck converter as the ESP32 can consume a surprising amount of power, although we recognize that a smaller one might have sufficed.

### 3.7 LDO: XC6220

The XC6220 LDO regulates the 5 V rail down to 3.3 V for the ESP32-H2 and logic. A low-dropout regulator was chosen over a second buck converter to minimize switching noise, ensuring clean power for the sensitive ADC and RF sections of the MCU.

### 3.8 Battery

A small single-cell Li-Po battery is included specifically for the "Power Out Position Retention" feature. It is not intended to drive the motor, but to power the MCU and encoder long enough to save the current state to non-volatile memory or perform low-power emergency parking of the motor. It is connected via a standard JST-PH 2-pin connector.

### 3.9 Connectors

USB C connector was chosen for the USB given the general migration towards a unification of USB devices. The unique power capabilities of it was another important factor. JST PH connectors were chosen based on two parameters. (A) current rating. The TMC2130 is rated for 2 A RMS per coil. This is right in line with the capabilities of the JST PH, as it is rated for 2 A per pin. (B) dimensions. Our design is rather compact

with a strict predefined footprint (nema17) and the 2 mm pitch was manageable for our design.

### 3.10 Battery charger IC: MCP73871

The MCP73871 is a linear battery charger with Power Path Management. This feature allows the system to run directly from the USB PD input while simultaneously charging the battery. If USB power is cut, the IC switches the load to the battery without resetting the MCU, preserving the RAM and position data.

## 4 Power Budget & Battery Life Analysis

### 4.1 Main Components Power Draw

Component	Rail	Current (Max)	Power
NEMA17 Motor	20 V	1.0 A	20.0 W
TMC2130 Logic	5 V	15 mA	0.075 W
ESP32-H2 (RX/TX)	3.3 V	123 mA	0.4 W
MT6835 Encoder	3.3 V	20 mA	0.066 W
XC6220 & Misc	3.3 V	50 $\mu$ A	165 $\mu$ W
MCP73871 & Misc	5.0 V	0.3 mA	1.5 mW
AP33771C & Misc	20.0 V	6 mA	120 mW
TPS54202 & Misc	20.0 V	45 $\mu$ A	0.9 mW
LEDs & Misc	3.3 V	10 mA	0.033 W
<b>Total (Peak)</b>			$\approx$ 20.69 W

Table 1: Estimated power draw in active state. Note for the ESP32-H2 it is assumed that the 802.15.4 antenna is using the highest signal strength (18.0 dBm)

Component	Rail	Current (Max)	Power
NEMA17 Motor	20 V	0.0 A	0.0 W
TMC2130 Logic	5 V	15 mA	0.075 W
ESP32-H2	3.3 V	17 mA	0.056 W
MT6835 Encoder	3.3 V	20 mA	0.066 W
XC6220 & Misc	3.3 V	50 $\mu$ A	165 $\mu$ W
MCP73871 & Misc	5.0 V	0.3 mA	1.5 mW
AP33771C & Misc	20.0 V	6 mA	120 mW
TPS54202 & Misc	20.0 V	45 $\mu$ A	0.9 mW
LEDs & Misc	3.3 V	10 mA	0.033 W
<b>Total (Peak)</b>			$\approx$ 0.35 W

Table 2: Estimated power in idle state

Component	Rail	Current (Max)	Power
NEMA17 Motor	20 V	0.0 A	0.0 W
TMC2130 Logic	0 V	0 mA	0.0 W
ESP32-H2	3.3 V	8 mA	0.0264 W
MT6835 Encoder	3.3 V	20 mA	0.066 W
LEDs & Misc	3.3 V	10 mA	0.033 W
XC6220 & Misc	3.3 V	50 $\mu$ A	165 $\mu$ W
MCP73871 & Misc	5.0 V	0.3 mA	1.5 mW
AP33771C & Misc	20.0 V	0 mA	0 W
TPS54202 & Misc	20.0 V	0 $\mu$ A	0.0 W
<b>Total (Peak)</b>			$\approx 125.4$ mW

Table 3: Estimated power in ultra low power state

## 4.2 Battery Life

For the system to operate as intended, the battery mode must be in the idle state and not the ultra low power state. This is due to the encoder not being multi-turn and thus the MCU must continuously poll the encoder to recognize a full turn. Assuming the use of a Samsung 18650 with 2900 mAh the current consumption of ultra low power state 38 mA thus the product can be used in  $\frac{2900}{38} = 76.31$  hours or 3.18 days.

# 5 Final product

## 5.1 3D render

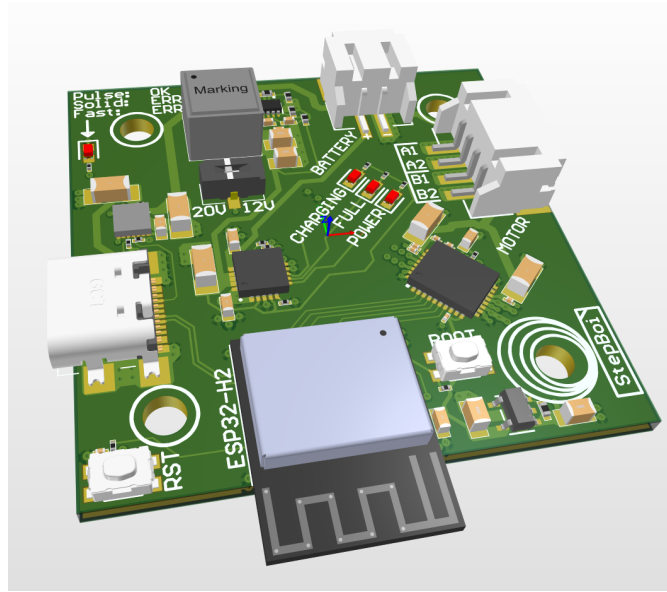


Figure 2: 3D render

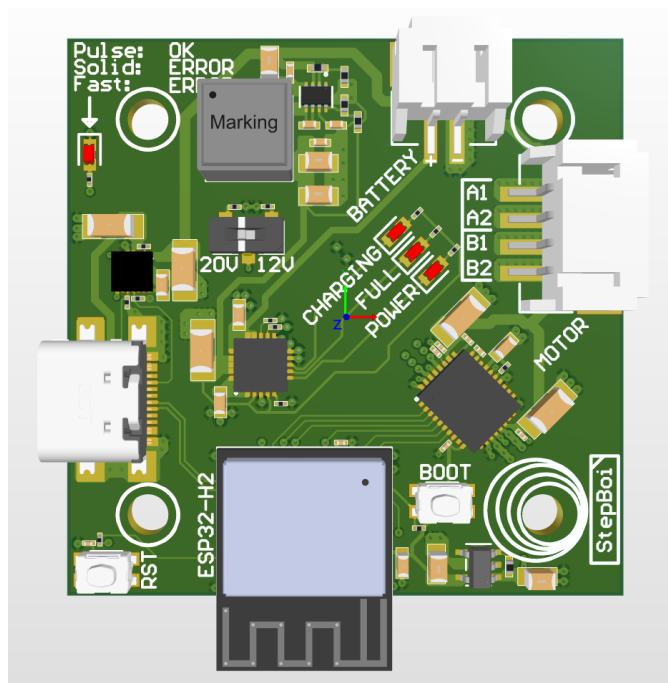


Figure 3: Top view

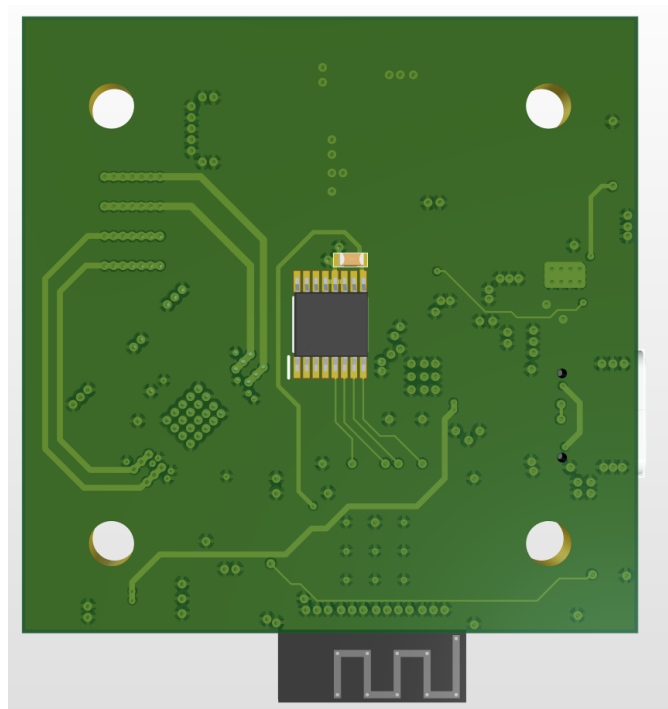


Figure 4: Bottom view