

# PROJECT BLUEPRINT: BỘ ĐIỀU KHIỂN ROBOT HÀN 6-DOF THƯƠNG MẠI

Phiên bản: 1.0

Ngày lập: 01/02/2026

Kiến trúc: Hybrid PC-Based (PC làm Não - MCU làm Cơ Bắp)

## 1. TẦM NHÌN & MỤC TIÊU (EXECUTIVE SUMMARY)

Xây dựng một bộ điều khiển robot hàn hồ quang (Arc Welding) 6 bậc tự do thương mại "Made in Vietnam". Hệ thống thay thế các giải pháp ngoại nhập đắt tiền bằng cách sử dụng phần cứng phổ thông mạnh mẽ kết hợp với thuật toán điều khiển tiên tiến.

- Mục tiêu Kỹ thuật:** Đạt được chuyển động nội suy đường thẳng (MOVL) mượt mà với biên dạng S-Curve, xử lý được đa nghiệm động học, và hỗ trợ tính năng hàn nâng cao (Weaving, Seam Tracking).
- Chiến lược Cốt lõi:** Sử dụng kiến trúc "Tách biệt" (Split-Architecture):
  - PC (Windows/C#):** Xử lý tính toán nặng (Toán học & Quỹ đạo).
  - MCU (Teensy 4.1):** Xử lý thời gian thực cứng (Phát xung & An toàn).

## 2. STACK CÔNG NGHỆ (THE WINNING STACK)

Hệ thống được xây dựng dựa trên việc tích hợp các thư viện mã nguồn mở hàng đầu thế giới:

### A. Tầng Toán học Lõi (Core Math Engine) - C++

- Robotics Library (RL) [BSD License]:**
  - Chức năng:** Giải quyết bài toán Động học ngược (Inverse Kinematics - IK).
  - Lý do chọn:** Hỗ trợ bộ giải **Analytical IK** cho cấu trúc Puma 560. Điều này cho phép kiểm soát 8 bộ nghiệm cấu hình (Configuration Flags: Arm/Elbow/Wrist) để tránh robot bị vặn xoắn bất ngờ.
- Ruckig (Community Version) [MIT License]:**
  - Chức năng:** Tạo quỹ đạo thời gian thực (OTG) với biên dạng S-Curve (Jerk-limited).
  - Lý do chọn:** Hỗ trợ chế độ **Phase Synchronization** (bắt buộc để đi đường thẳng MOVL) và **Arbitrary Target States** (cho phép cập nhật mục tiêu tức thì để làm Seam Tracking).

### B. Tầng Thực thi Phần cứng (Execution Layer) - Firmware

- grblHAL Core [GPLv3]:**

- **Chức năng:** Nhận xử lý G-code và thuật toán tạo xung bước (Step Generation).
- **Cấu hình:** Kích hoạt chế độ **6 Trục độc lập (XYZABC)** và tối ưu hóa bộ đệm (Buffer) lớn.
- **Phần cứng: Teensy 4.1** (Chip ARM Cortex-M7 600MHz) hoặc STM32F4/H7.
  - **Lý do chọn:** Sức mạnh xử lý đủ để phát xung tần số cao (>300kHz) cho 6 trục mà không bị nghẽn (jitter).

## C. Tầng Giao diện & Ứng dụng (Application Layer) - C#

- **Ngôn ngữ:** C# (WPF/.NET 6+).
- **Hiển thị 3D:** Helix Toolkit (Hiển thị mô hình robot và đường hàn thời gian thực).
- **Giao tiếp:** C++/CLI Wrapper (Cầu nối giữa C# và C++ Core).

## 3. LỘ TRÌNH THỰC THI (EXECUTION ROADMAP)

Dự án được triển khai theo 3 giai đoạn cuốn chiếu:

### Giai đoạn 1: The Virtual Controller (Mô phỏng & Toán học)

*Mục tiêu: Chứng minh toán học đúng trên máy tính trước khi kết nối phần cứng.*

1. **Xây dựng Core Math DLL (C++):**
  - Tích hợp RL & Ruckig vào một project C++ duy nhất.
  - Viết hàm `CalculatePath(StartPose, EndPose)` trả về danh sách các góc khớp theo chu kỳ (ví dụ 10ms).
2. **Xây dựng "Cầu nối" (Bridge):**
  - Viết lớp **C++/CLI Wrapper** để ứng dụng C# có thể gọi trực tiếp các hàm tính toán của C++.
3. **Phát triển Simulator (WPF):**
  - Tạo giao diện với thanh trượt (Sliders) điều khiển XYZ.
  - Dùng Helix Toolkit để hiển thị chuyển động.
  - **KPI:** Robot trên màn hình di chuyển thẳng (MOVL) mượt mà, đúng tư thế tay.

### Giai đoạn 2: The Physical Link (Kết nối Phần cứng)

*Mục tiêu: Điều khiển động cơ thực tế quay mượt mà, không rung giật.*

1. **Chuẩn bị Phần cứng:**
  - Sử dụng board Teensy 4.1 hoặc STM32F4/H7.
  - Biên dịch và nạp firmware `grblHAL` với cấu hình: `#define N_AXIS 6` và tăng `RX_BUFFER_SIZE` lên 4096 bytes.
2. **Phát triển Module Streaming (C#):**
  - Viết driver giao tiếp Serial sử dụng thuật toán "**Character Counting**" (đếm byte) thay vì "Send-Wait-OK".
  - Đảm bảo luồng dữ liệu xuống MCU luôn liên tục, tránh hiện tượng "đói dữ liệu" (Starvation).

### 3. Cấu hình "Planner Bypass":

- Cài đặt tham số \$11 (Junction Deviation) cực lớn và giới hạn gia tốc ảo cao trong grblHAL để Firmware không can thiệp vào quỹ đạo S-Curve đã tính toán bởi Ruckig.

## Giai đoạn 3: Welding Intelligence (Công nghệ Hàn)

Mục tiêu: Hoàn thiện tính năng công nghệ.

### 1. Calibration (Hiệu chuẩn):

- Xây dựng quy trình đo đặc kích thước thực tế của cánh tay robot.
- Cập nhật thông số DH (Denavit-Hartenberg) vào file cấu hình XML của Robotics Library.

### 2. Weaving (Hàn lắc):

- Cài đặt thuật toán dao động hình Sin (Sine Wave) chồng lên quỹ đạo Ruckig.

### 3. Seam Tracking (Bám đường hàn):

- Tích hợp cảm biến Laser/Vision.
- Sử dụng tính năng cập nhật mục tiêu thời gian thực của Ruckig để bù trừ sai lệch đường hàn.

## 4. CHIẾN LƯỢC QUẢN LÝ RỦI RO KỸ THUẬT

Rủi ro Tiềm ẩn	Giải pháp Kỹ thuật
Đường hàn bị cong	Cấu hình Ruckig bắt buộc sử dụng chế độ Synchronization::Phase.
Robot bị rung/giật	<ol style="list-style-type: none"><li>Sử dụng S-Curve (Jerk Control).</li><li>Tăng kích thước Buffer grblHAL.</li><li>Áp dụng chiến lược "Planner Bypass".</li></ol>

<b>Độ trễ (Latency)</b>	<p>1. Tối ưu hóa C++/CLI Wrapper (tránh copy dữ liệu thừa).</p> <p>2. Sử dụng giao thức Stream đếm byte (Token Bucket).</p>
<b>Đa nghiệm (Vặn tay)</b>	Sử dụng bộ giải Analytical IK của RL và ép buộc các cò cầu hình (Elbow Up/Down) cố định trong suốt đường hàn.
<b>Giới hạn phần cứng</b>	Nếu dùng STM32 cũ (F103), phải nâng cấp lên F4/H7 hoặc Teensy 4.1 để đủ RAM cho Buffer.