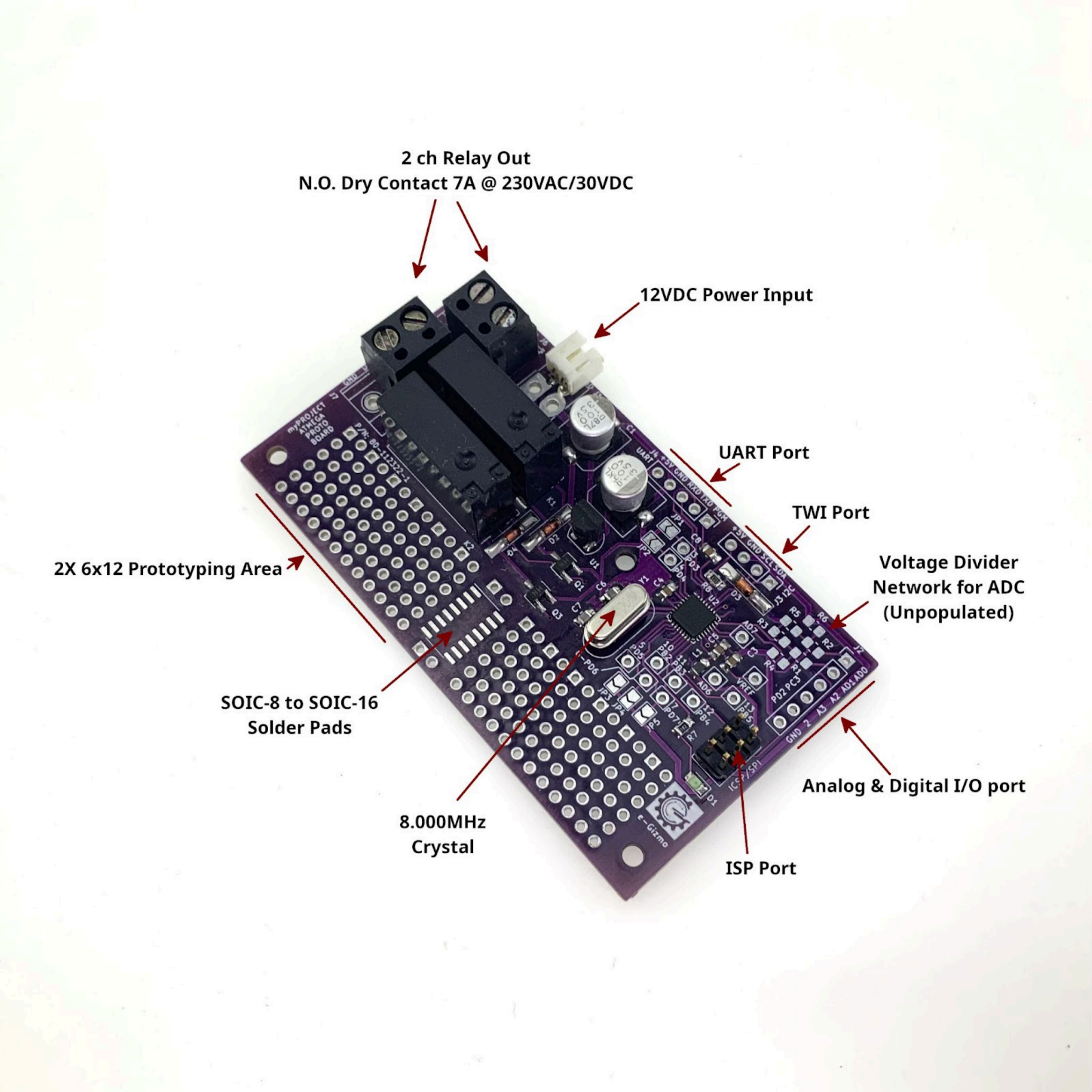
Technical Manual:

ATmega88P Quick Prototyping Board

Beyond Arduino



1. Introduction

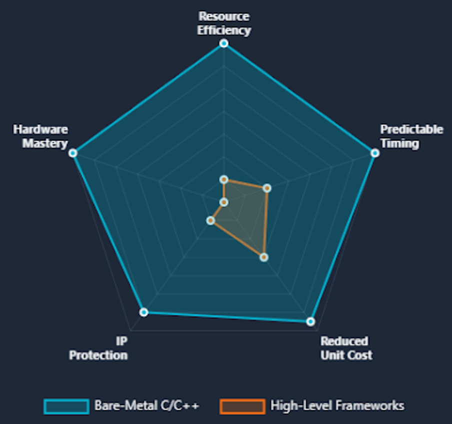
This prototyping board is a deployable, simple prototype controller board based on the **ATmega88P** microcontroller. While many entry-level platforms rely on abstraction layers, this board is designed to be programmed in **bare-metal C/C++**, allowing developers to interface directly with the hardware registers.

ATmega88P Prototype Board

A deployable, bare-metal C/C++ development platform for commercial embedded systems.

Level Up: The Bare-Metal Advantage

Transitioning from high-level abstractions to bare-metal C/C++ is critical for commercial products. It ensures microsecond predictable timing, optimizes the 8KB/1KB memory architecture, and reduces unit costs for production runs.



Core Specifications

12VDC
POWER INPUT

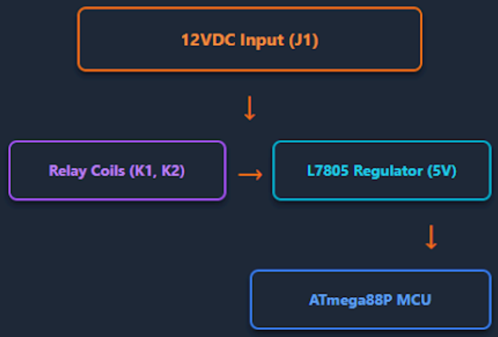
8.0MHz
CRYSTAL OSC

7A
RELAY RATING

5V
LOGIC RAIL

Power Architecture

The board accepts 12VDC which directly drives the heavy-duty relay coils, while an onboard L7805 regulator drops the voltage to a stable 5V for the MCU and logic headers.



12VDC Input (J1)

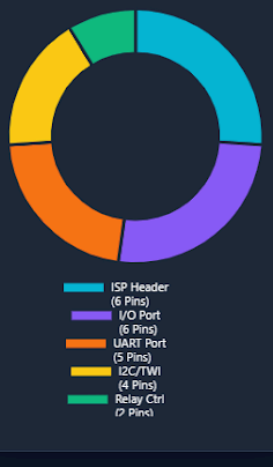
Relay Coils (K1, K2)

L7805 Regulator (5V)

ATmega88P MCU

Header Allocation

Distribution of connectivity pins across the board's integrated headers.



Peripheral Integration & Prototyping

High-Voltage Relays

- ✓ **K1 (PB0):**SPST, Active High via DTC143Z
- ✓ **K2 (PB1):**SPST, Active High via DTC143Z
- ✓ **Rating:**7A @ 250VAC / 30VDC
- ✓ **Terminals:**Accessible via J6, J7

Connectivity Headers

- **TWI/I2C (J3):**PC4 (SDA), PC5 (SCL)
- **UART (J4):**PD0 (RXD), PD1 (TXD)
- **ISP-6 (J5):**MISO, MOSI, SCK, RST

Custom Prototyping Area

The board provides dedicated space for expanding hardware functionality without requiring a separate breadboard.

Through-Hole Grid	Two 6x12 Arrays
SMD Pads	Two SOIC-16
Free I/O Pads	JP1 to JP5

1.1 Level up to Pro; The Advantage of Bare-Metal C/C++

Transitioning from high-level frameworks to bare-metal C/C++ is a critical step for developers intending to move from a prototype to a **commercial product**. Key advantages include:

- **Resource Efficiency:** High-level abstractions often include unnecessary overhead. Writing in bare-metal C/C++ allows you to optimize memory usage (Flash and SRAM), which is vital on the ATmega88P's 8KB/1KB architecture.
- **Predictable Timing:** In commercial industrial or automotive applications, deterministic behavior is mandatory. Removing background library processes ensures your interrupts and control loops execute with microsecond precision.
- **Reduced Unit Cost:** By optimizing code for efficiency, you can often use a lower-spec (and cheaper) MCU for the final production run without sacrificing performance.
- **IP Protection and Portability:** Bare-metal code is generally easier to audit for security vulnerabilities and more straightforward to port to other architectures if the global supply chain necessitates a component swap.
- **Hardware Mastery:** Programming in C/C++ forces a deep understanding of the datasheet, ensuring that the final product utilizes hardware features like Sleep Modes and Power Reduction Registers to maximize battery life or thermal efficiency.

2. Hardware Specifications

- **Microcontroller:** ATmega88P-20M (QFN/TQFP package)
- **External Power Input:** 12VDC via screw terminal (Minimum 500mA recommended).
- **On-board Regulation:** L7805 Linear Regulator provides a stable +5V logic rail.
- Clock Frequency: 8.000MHz External Crystal Oscillator (**Y1**).
- **Flash Programming:** Recommended tool is the **e-Gizmo USBASP**.

3. Integrated Peripherals & Headers

3.1 Relay Channels (K1, K2)

The board features two integrated relays for switching external loads.

- **Control Pins:**
  - **Relay 1:** PB0 (Active High via DTC143Z transistor)
  - **Relay 2:** PB1 (Active High via DTC143Z transistor)
- **Contact Rating:** 7A @ 250VAC / 30VDC.
- **Access:** Individual SPST contacts are accessible via dedicated 2-pin screw terminals (**J6**,**J7**).

3.2 User Interface

- **Function LED:** A green indicator LED (**D1**) is connected to **PB2**. It can be used for status signaling or debugging.

3.3 Connectivity Headers

- **TWI/I2C Port (J3):** 4-pin header (SDA, SCL, GND, +5V).
- **UART/Programming Port (J4):** 5-pin header (PGM, TXD, RXD, GND, +5V).
- **AVR-ISP-6 (J5):** Standard 2x3 ISP header for firmware flashing.
- **I/O Port (J2):** 6-pin header providing access to:
    - ADC0, ADC1 (Analog inputs)
    - PC2, PC3 (Digital I/O)
    - PD2 (Digital I/O / External Interrupt)
    - GND reference.

4. Prototyping Area

The board includes a dedicated expansion area for custom circuitry:

- **Through-Hole Grid:** Two 6x12 solder pad arrays for standard DIP components.
- **SMD Pads:** Two SOIC-16 footprints for surface-mount IC integration.
- **Free I/O:** Remaining unused ATmega88P pins are routed to labeled solder pads (**JP1** through **JP5**) for easy wire-wrapping or jumpering.

5. Pinout Reference Table

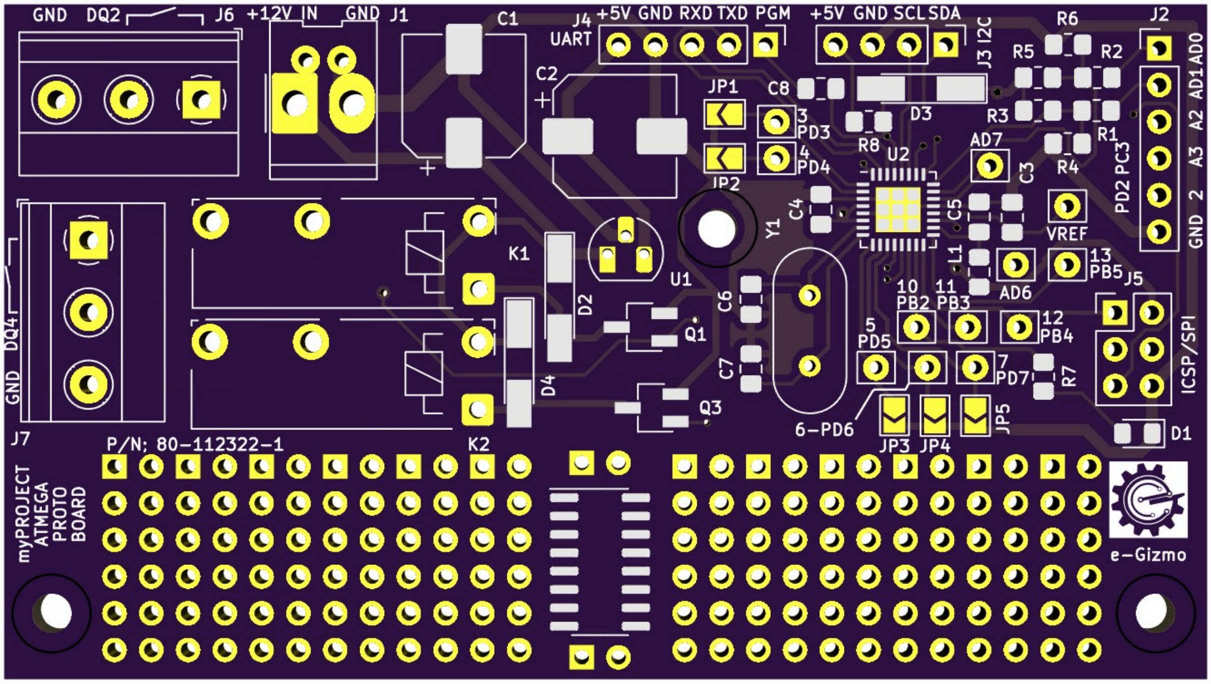
Function	**Pin Name**	**Schematic Label**	**Note**
Relay 1	PB0	PB0	Drives K1
Relay 2	PB1	PB1	Drives K2
User LED	PB2	PB2	Green LED (D1)
UART RX	PD0	RXD	Header J4
UART TX	PD1	TXD	Header J4
I2C SDA	PC4	SDA	Header J3
I2C SCL	PC5	SCL	Header J3
ISP MISO	PB4	MISO	Header J5
ISP MOSI	PB3	MOSI	Header J5
ISP SCK	PB5	SCK	Header J5

6. Power Considerations

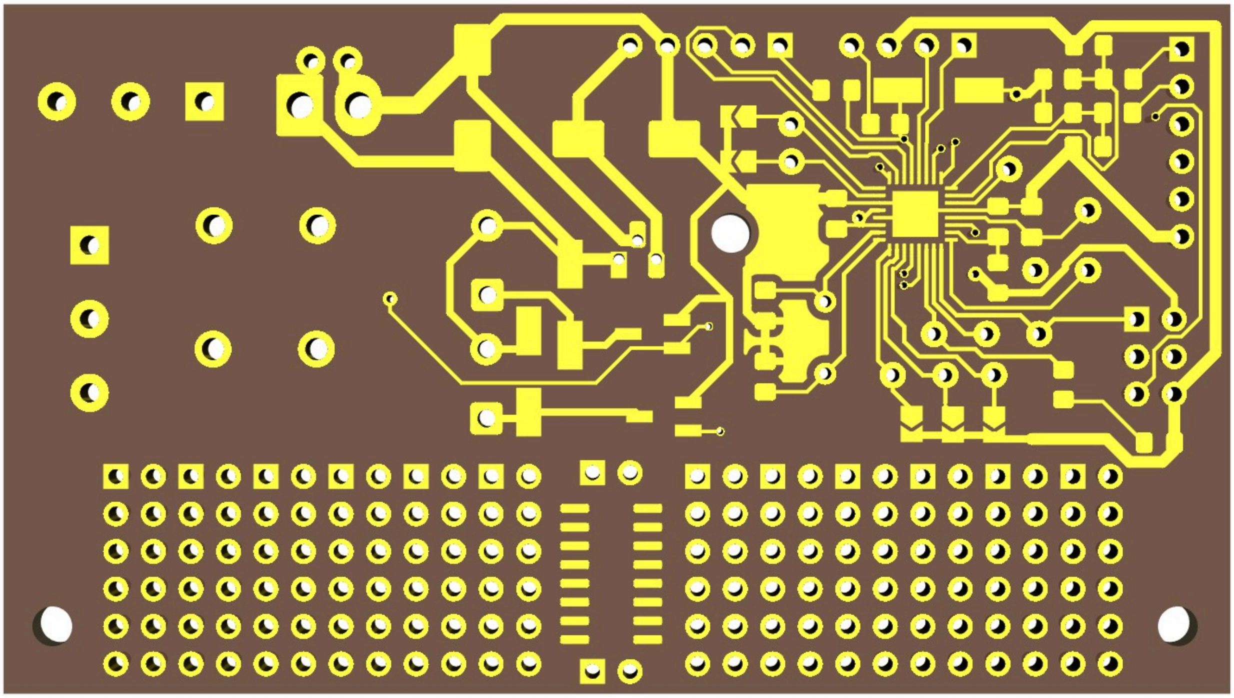
The board requires a 12VDC input at **J1**. This voltage directly powers the relay coils (**K1**, **K2**) while the **L7805** regulator (**U1**) drops the voltage to 5V for the MCU and logic headers. Ensure your power supply can handle the combined current draw of the relays (approx. 40-70mA each) plus any external sensors attached to the headers.

7. PCB

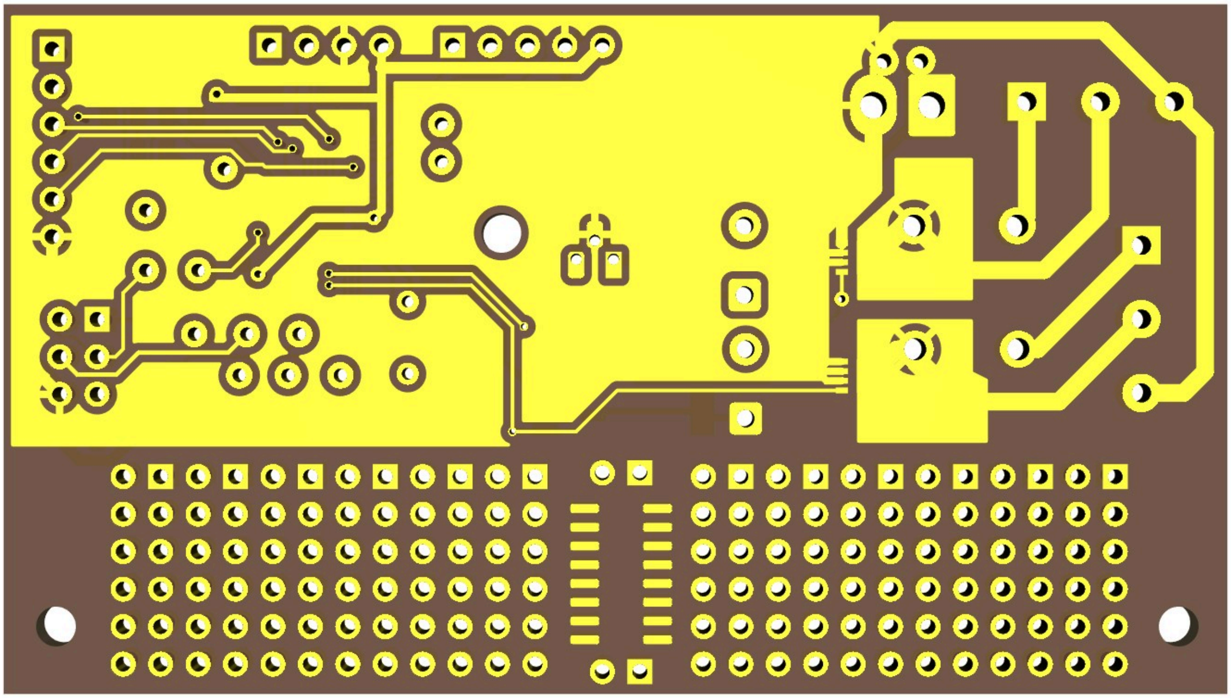
Parts Placement



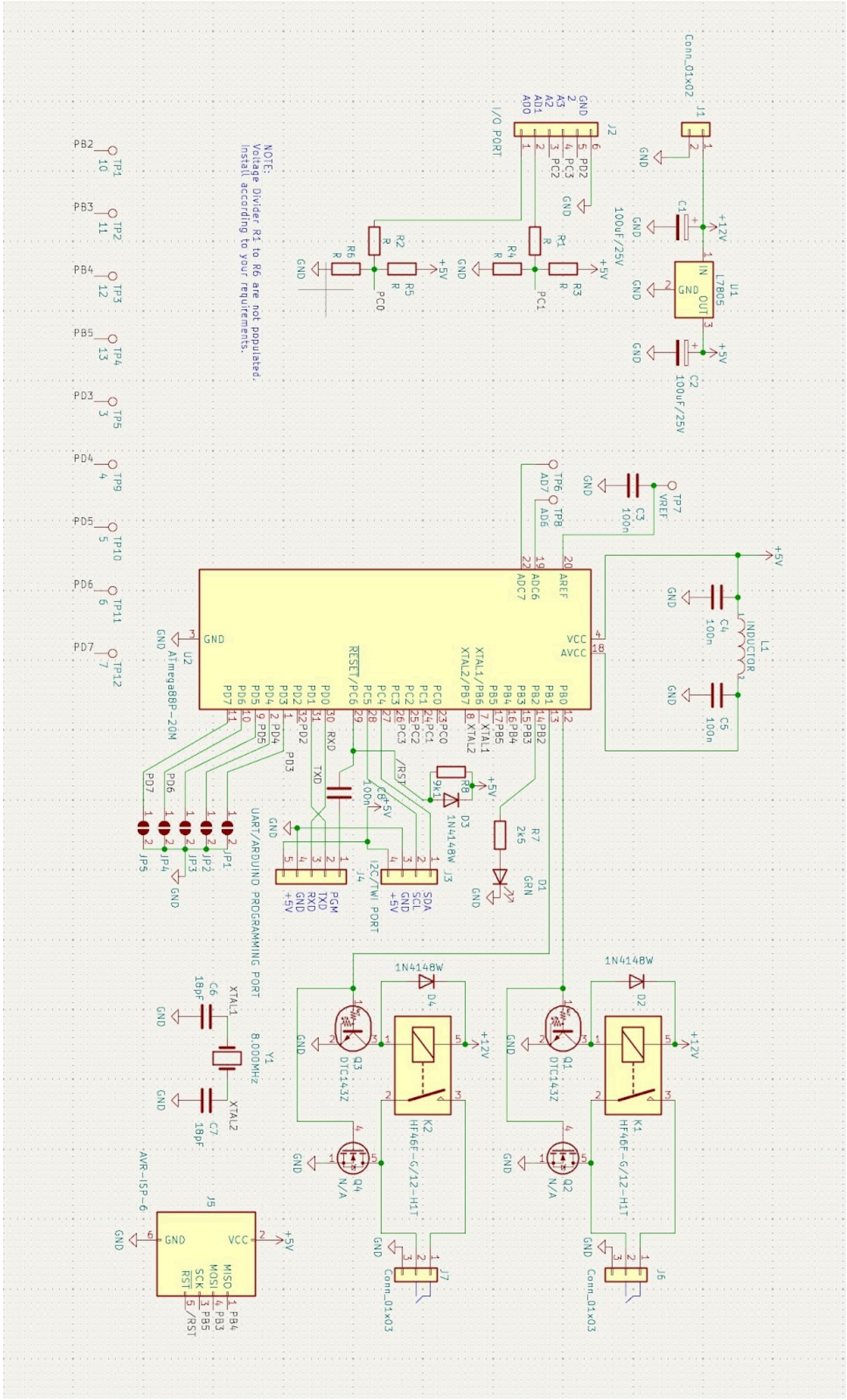
PCB Pattern Top Side



PCB Pattern Bottom Side



8. Schematic Diagram



8. Links

8.1 **Microchip Studio Integrated Development Environment (IDE)**

Microchip Studio for AVR® and SAM Devices | Microchip Technology

8.2 **How to Enable ATmega88P in Arduino IDE**

Still can't shake Arduino out of your system?
GitHub - MCUdude/MiniCore: Arduino hardware package for ATmega8, ATmega48, ATmega88, ATmega168, ATmega328 and ATmega328PB · GitHub

Installation instructions are detailed in the MiniCore documentation.

8.3 e-Gizmo USBASP kit

e-Gizmo USBASP AVR Programming Tool with 6-pin ISP Interface

- ○