**Must-watch video with details about the printed circuit board:**  
<https://www.youtube.com/watch?v=-_hQa9JtouU>

To open the board, you need the **Sprint Layout 6.0** program.

I highly recommend watching the previous parts, where the circuit, simulation, and all stages of development are explained in detail.

* Part 1: <https://www.youtube.com/watch?v=9bNtSEOOno0>
* Part 2: <https://www.youtube.com/watch?v=7C9bg4Z8BRs>
* Part 2.1: <https://www.youtube.com/watch?v=fEoLM9fryu8>
* Part 3: <https://www.youtube.com/watch?v=qYMWOKI5Wsg>
* Part 4: <https://www.youtube.com/watch?v=D-SWTAUGvdE>
* Part 5: <https://www.youtube.com/watch?v=5y6lBQpGDro>

**Project Presentation: MyWeld V2.0 (Auto)**

**Project Author:** Aka Kasyan  
**Development Date:** 04/2025

**IMPORTANT!** This is a non-commercial project. The author distributes it **FOR FREE**, provided that copyrights are respected. Anyone can download and replicate it for a "thank you."

The current version of the project as of 04/05 is **V2.0**, and the software version is also **V2.0**. The initial version, **V1.0**, is considered outdated by the author. It is fully functional but lacks the AUTO mode.

**Project Description:**

A device for contact (spot) welding of batteries based on the **Atmega 328 (Arduino Nano)**.

The device is a control system for spot welding with an integrated power section. The welding current source consists of two **2.7V 3000F supercapacitors** connected in series, resulting in a total voltage of **5.4V** when fully charged.

Current switching is handled by powerful **MOSFET transistors**.

**Components:**

The device consists of several modules:

1. Main printed circuit board (motherboard)
2. Arduino Nano board
3. Encoder with a button (KY-040)
4. 2004 LCD display with an I2C interface module

**Features and Settings:**

The device has two operating modes: **manual** and **automatic**. All settings and parameters work identically in both modes.

1. **Manual Mode (MAN):** Welding is controlled by an external button or pedal.
2. **Automatic Mode (AUTO):** Welding is triggered by the short-circuiting of the welding electrodes. In AUTO mode, an additional parameter **S** (delay before welding) is introduced. After detecting electrode contact, the system waits for time **S** before initiating the welding pulse. A buzzer emits a short sound when contact is detected. If the electrodes are disconnected during time **S**, welding will not start.

The device can operate in **single-pulse** or **dual-pulse** mode, with a delay between pulses. To use single-pulse mode, set the second pulse time **P2** to **0ms**.

**Functionality:**

The **2004 display** shows the following parameters:

* **First line:** **P1** (first welding pulse time) and **T** (delay between pulses).
* **Second line:** **P2** (second pulse time). All values are in milliseconds, adjustable from **0-50ms** in **1ms** steps.

**Note:** There is an additional **10ms delay** between **P1** and **P2**, which is added to the **T** delay. This is a safety measure to ensure proper system operation at low delays.

To use a single pulse, simply set **P2** to **0ms**.

On the second line of the display, the **Light** parameter controls the backlight brightness (0-5, where 0 is the dimmest).

The **third line** shows the **MODE (AUTO/MAN)**. In AUTO mode, the **S** parameter (delay after electrode contact) appears, adjustable from **0.3 to 2 seconds**.

**IMPORTANT:** In MAN mode, the **S** parameter should not appear in the software or simulator. However, some low-quality displays may fail to erase it. This does not affect functionality, as the parameter cannot be edited in MAN mode.

The **bottom line** displays a voltmeter showing the actual power supply voltage.

To select a parameter, rotate the encoder. The selected parameter is indicated by an arrow. To edit, press and hold the encoder button—a beep will sound, and the parameter will blink. Rotate the encoder to adjust, then press and hold the button again to save.

**Working Principle and Circuit Description:**

The heart of the circuit is the **ATMEGA 328** microcontroller, implemented on an **Arduino Nano** board.

Power from the supercapacitor battery (5.4V) is first fed to a **boost converter**, which outputs **12-15V**. This voltage is necessary for the **gate driver (TC4422)**, ensuring fast switching of the power transistors.

The boost converter is built around the **XR2981** IC. The control system (Arduino, etc.) is powered by the 15V line, stepped down to 5V by a linear regulator (no heatsink required).

The start button for manual mode uses **optical isolation** to avoid noise interference from long wires (e.g., pedal cables).

Capacitors are used throughout the circuit for noise immunity, and electrolytic capacitors compensate for voltage drops during welding.

* **Pin A2:** Voltmeter monitoring the 15V line.
* **Pin A1:** Voltmeter monitoring the power supply voltage.
* **Pin A3:** Monitors voltage at the transistor drains to detect electrode contact (for AUTO mode).
* **Pin D9:** Controls display backlight brightness.

**Modification for the 2004 LCD display with I2C:**  
Locate the transistor on the I2C module that controls the backlight. Desolder its base resistor and connect a new resistor (1kΩ to 3.3kΩ) from the base to **Pin D9** on the Arduino.

Additional LEDs indicate:

* Output pulses from the Arduino.
* Presence of 15V and 5V power (the latter is also indicated by the Arduino's onboard LED).

**IMPORTANT!**

The control system **does not include a charging or balancing circuit for the supercapacitors**.

**Protections:**

The device includes several electronic and software protections:

1. **Self-test:** Performed automatically upon startup (takes ~10-12 seconds). Tests the display, buzzer, backlight, memory, and protections.
2. **Low-voltage protection:** If the supercapacitor voltage drops below **4.5V**, welding is disabled. The display shows "LOW" until voltage normalizes.
3. **Gate voltage protection:** If the boost converter output is **<10V or >18V**, the display shows "DC 15V PROBLEM! DANGER! CONTROL VOLTAGE ERR." The buzzer beeps periodically, and the device becomes unresponsive until restarted.

**Note:** Protections monitor conditions for **1.5 seconds** before triggering to avoid false alarms during welding voltage drops.

Additional protections include:

* Zener diodes for overvoltage on the gates.
* Pull-down resistors to ensure transistors turn off if the control system fails.
* Reverse polarity protection.
* Suppressors for back-EMF (optional, not soldered by default).

**Testing and Component Selection:**

1. Power the device and adjust the **multiturn resistor R6** so the voltmeter matches the actual supply voltage.
2. Test all settings and protections:
   * Lower the supply voltage below 4.5V to trigger low-voltage protection.
   * Disable the 15V line to trigger gate voltage protection (requires a restart to reset).
3. Use an oscilloscope to verify pulse timing and shape. Add **10ms** to the measured delay between pulses.
4. For welding, use **16 AWG or thicker cables**. Start with **2-3ms pulses** for 0.1-0.12mm strips.
   * **Wear safety goggles** and shield the transistors during welding.

**Test Results:**

The project was tested in real-world conditions and simulations. With **2x 2.7V 3000F supercapacitors**, welding currents reached **1700-2000A** (using 6 mid-range transistors). With recommended transistors (e.g., **IRL40SC228**), currents up to **2500A** are achievable, suitable for nickel strips up to **0.4mm**.

For thicker materials (e.g., **1mm copper or aluminum**), use **2S2P/2S3P supercapacitor configurations** and **10-12 IRL40SC228 transistors**.

**Transistor requirements:**

* N-channel MOSFETs with **V\_DS ≥ 25-30V** and **R\_DS(on) ≤ 1.8mΩ**.
* Use **genuine transistors** to avoid failure during welding.

**Potential Issues:**

No issues will arise if:

* The assembly is correct.
* The original software is used unmodified.
* Genuine components are employed.

**Important for the Arduino sketch:**

* The sketch is provided in **DOCX** (Word) and **PAGES** (Mac) formats.
* **Do not copy-paste the code**—directly upload the original file to avoid errors.
* The sketch has **394 lines**. Removing empty lines or spaces may cause display glitches (but won't affect functionality).

**Future Development:**

* Upgraded **double-sided PCBs** with better converters, input protections, and shielding.
* Separate driver and power boards (to be released for free after testing).
* The power board will support up to **12 IRL40SC228 transistors** with a **300A fuse**.

**No plans to migrate to STM32/ESP32**—the ATMEGA328 is more than sufficient. Fancy features (counters, multilingual support, etc.) are deemed unnecessary for this application.

**Updates and Contact:**

All updates will be available on the author's official channels:

* YouTube:
  + <https://www.youtube.com/@akakasyanshorts>
  + <https://www.youtube.com/c/akakasyan>
  + <https://www.youtube.com/@KitShop1/featured>
  + <https://www.youtube.com/c/KasyanTV>
* Dzen: <https://dzen.ru/akakasyan>

**Email for inquiries:** [artur.kasyan@mail.ru](https://mailto:artur.kasyan@mail.ru/)  
**Subject line for project-related questions:** "MyWeld"

**Author:** Aka Kasyan  
**Development Date:** 04/2025

**IMPORTANT!** This is a **non-commercial project**, distributed **for free** under the condition of respecting copyright. Anyone can replicate it for a "thank you."

For ideas or commercial inquiries, contact the author via email.