

Po Tolo - Printable Technical Manual

Table of Contents

1. **Introduction**
 - Project Overview
 - System Components
 2. **Hardware Setup**
 - Arduino Mega and Custom Shield
 - Podium Assembly (Arduino, LEDs, Sensors)
 - Digiquad and Network Hardware
 - Wiring Diagrams
 3. **Network Configuration**
 - IP Addresses and Device Roles
 - Digiquad Setup (sACN/Art-Net)
 - TouchDesigner Network Configuration
 4. **Software and Code**
 - Arduino Code Upload
 - Required Libraries
 - TouchDesigner Project Setup
 - Arduino-TouchDesigner Integration
 5. **Troubleshooting**
 - Hardware Issues
 - Network Problems
 - Sensor Errors
 - Software and Compilation Errors
-

1. Introduction

Project Overview

The Po Tolo project, also known as Solar Shrine, is an interactive art installation featuring two podiums. Each podium contains an Arduino Mega, custom LED hand displays, and sonar sensors that react to hand proximity. The system generates audio and visual effects based on user interaction, controlled by a central TouchDesigner application.

System Components

- **Podiums (x2):** Each houses an Arduino, LEDs, and sensors.
- **Arduino Mega (x2):** The microcontroller brain for each podium.
- **Custom Arduino Shield:** Simplifies connections for all components.
- **LED Hand Displays:** WS2812B/WS2815 LED strips.
- **Sonar Sensors (x4):** HC-SR04 ultrasonic sensors for hand detection.

- **Digiquad (x4):** Network devices to control LED strips via sACN or Art-Net.
 - **Control Machine:** A Mac mini running TouchDesigner.
 - **Network Switch:** Connects all devices.
-

2. Hardware Setup

Arduino Mega and Custom Shield

The custom shield is designed to stack on top of the Arduino Mega, providing clean and robust connections for all external components.

Shield Features:

- Direct amplifier stacking.
- JST-XH connectors for sensors and LEDs.
- Perfboard section for custom circuits.
- Multiple power input options (12V Barrel Jack or Terminal Block).
- 3.5mm and screw terminal audio outputs.

Component Layout:

SOLAR SHRINE CUSTOM SHIELD - TOP VIEW (68.6mm x 53.4mm)

WWZMDiB XH-M543 AMPLIFIER		PERFBOARD
92mm x 68mm x 16mm		SECTION
TPA3116D2		20x15 holes
IN	OUT	0.1" spacing

SENSOR	SENSOR	LED	POWER	AUDIO	STATUS
1	2	STRIP	INPUT	OUT	LEDS
JST-XH	JST-XH	JST-XH	BARREL	3.5mm	PWR
4-pin	4-pin	3-pin	JACK	JACK	ACT

ARDUINO UNO HEADER PINS

DIGITAL 0-7

DIGITAL 8-13 + POWER

Podium Assembly

Each podium requires careful assembly of the Arduino, LED displays, and sonar sensors inside a waterproof enclosure.

Assembly Steps:

1. **Mount the Arduino/Shield:** Secure the stacked Arduino and custom shield inside the enclosure using standoffs.
2. **Connect Sensors:** Plug the HC-SR04 sensors into the 4-pin JST-XH connectors on the shield.
3. **Connect LEDs:** Plug the LED hand displays into the 3-pin JST-XH connector.
4. **Connect Power:** Connect a 12V power supply to either the barrel jack or the terminal block.
5. **Seal Enclosure:** Ensure all cable glands are tightened and the enclosure is properly sealed.

Shield in Enclosure (Photo):

This photo shows the Arduino Mega with the custom red protoshield installed inside the enclosure. Note the cable glands, wire strain relief, and mounting standoffs. Ensure wires are dressed to avoid pressure on the shield or Arduino headers.

12V Power Connection to Shield (Photo):

This photo illustrates the 12V power leads landed on the shield's power input terminal. Verify polarity per the terminal block markings before powering on. Keep power and data grounds common, and use fusing or an inline switch where appropriate.

Digiquad and Network Hardware

The Digiquads translate sACN/Art-Net data from TouchDesigner into signals for the LED strips.

Hardware:

- 4 x Digiquad devices
- 1 x Network Switch
- Ethernet cables

Connections:

1. Connect each Digiquad to the network switch.

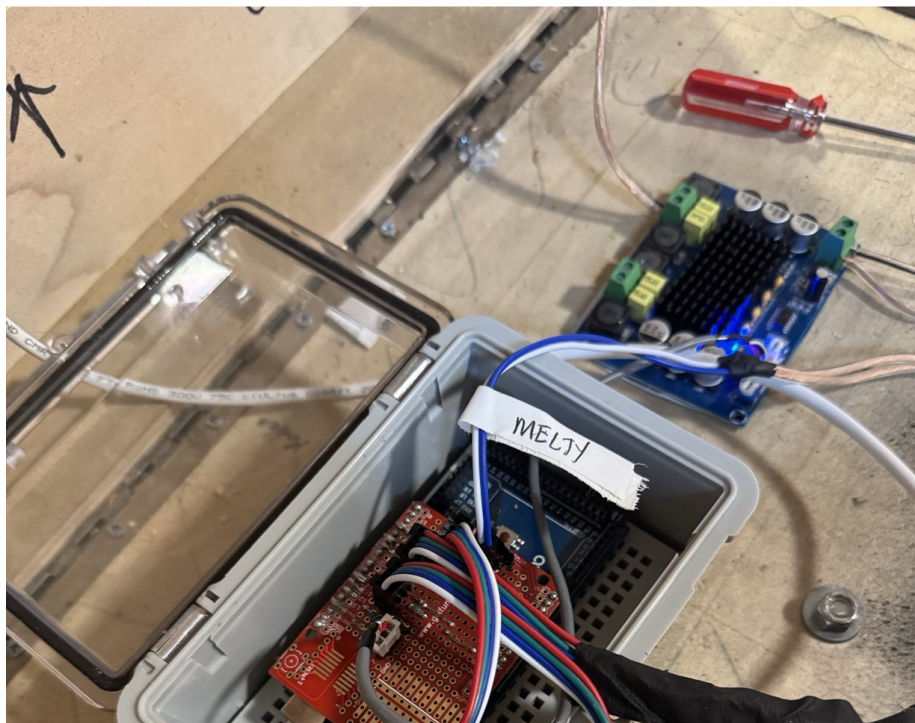


Figure 1: Shield enclosure inside waterproof box with cable routing

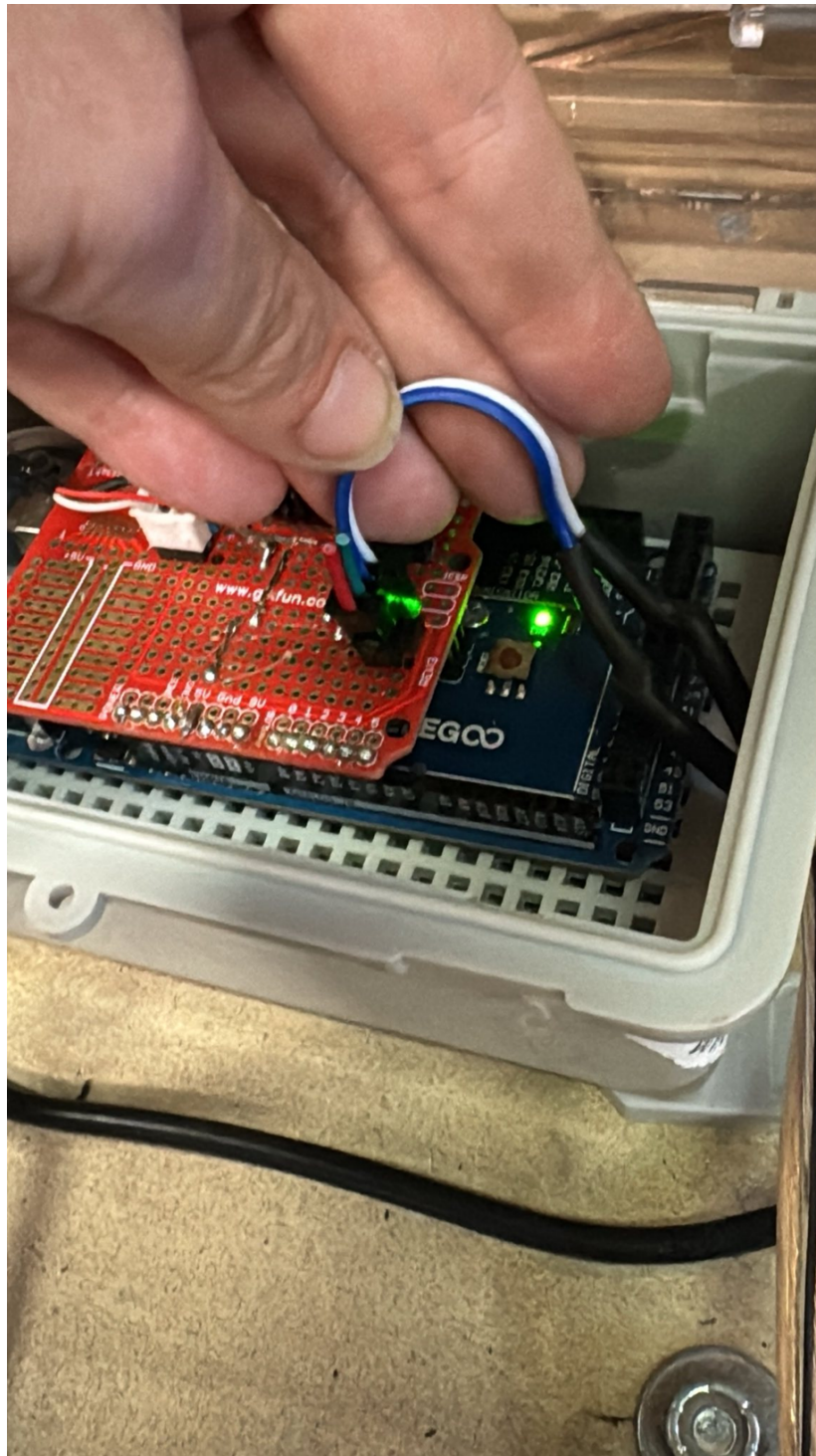


Figure 2: 12V power leads connected to the shield's power terminal

2. Connect the control Mac mini to the network switch.
3. Power on all devices.

Wiring Diagrams

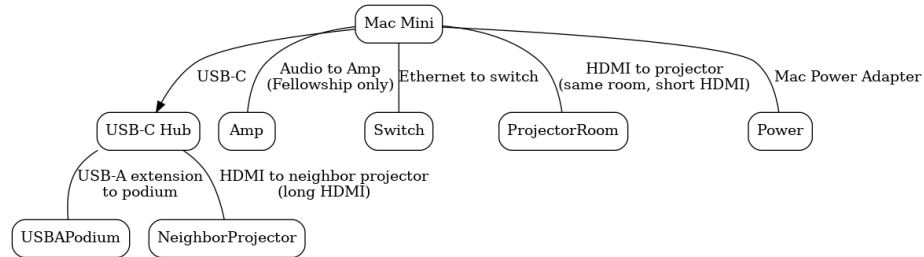


Figure 3: Wiring Diagram

Sensor Connector (JST-XH 4-pin):

Pin	Sensor 1 (Left)	Sensor 2 (Right)	Arduino Pin
1	VCC (5V)	VCC (5V)	5V
2	GND	GND	GND
3	Trig	Trig	Pin 9
4	Echo	Echo	Pin 10
1	VCC (5V)	VCC (5V)	5V
2	GND	GND	GND
3	Trig	Trig	Pin 5
4	Echo	Echo	Pin 6

LED Strip Connector (JST-XH 3-pin):

Pin	Function	Arduino Pin	Notes
1	VCC	External 5V/12V	From power input
2	GND	GND	Common ground
3	Data	Pin 3	Signal line

3. Network Configuration

IP Addresses and Device Roles

Device name	IP address	MAC address	Role
ANTWANES-MINI1	192.168.3.150		Control Machine 1
ANTWANES-MINI2	192.168.3.151		Control Machine 2
digi-threelegs	192.168.3.201	94:54:C5:FD:F0:4B	Digiquad
digi-snake	192.168.3.202	94:54:C5:FD:F6:EF	Digiquad
digi-pineapple	192.168.3.203	14:2B:2F:E5:6A:DB	Digiquad
digi-radish	192.168.3.200	94:54:C5:FE:10:C3	Digiquad

Digiquad Setup (sACN/Art-Net)

1. **Access Digiquad:** Open a web browser and navigate to the Digiquad's IP address (e.g., <http://192.168.3.201>).
2. **Sync Interfaces:** Click **Sync Interfaces** in the left menu.
3. **Network DMX Input:** Scroll to Realtime -> Network DMX input.
4. **Set Type:** Choose E1.31 (sACN) (recommended) or Art-Net.
5. **Configure sACN:**
 - Start universe: 1 (or as per your mapping)
 - DMX start address: 1
 - DMX mode: Multi RGB
6. **Save and Reboot:** Click **Save** and then reboot the Digiquad using Tools -> Reboot. **A reboot is mandatory after changing the protocol.**

Recommended Universe Mapping:

Universe	Device
1	digi-threelegs
2	digi-snake
3	digi-pineapple
4	digi-radish

TouchDesigner Network Configuration

- Use a **DMX Out CHOP**.
- Set **Protocol** to sACN or Art-Net to match the Digiquad configuration.
- For sACN, you can use **Multicast** for testing or **Unicast** with the specific Digiquad IPs for production.
- Set the **Universe** to match your mapping.

4. Software and Code

Arduino Code Upload

The main production code is located at: 01_MAIN_SYSTEM/solar_shrine_theremin/solar_shrine_theremin.

Upload Steps:

1. Connect the Arduino Mega to your computer via USB.
2. Open the `.ino` file in the Arduino IDE.
3. Select the correct Board (Arduino Mega) and Port from the **Tools** menu.
4. Click the “Upload” button.

Required Libraries

Install these libraries via the Arduino IDE’s Library Manager (**Tools -> Manage Libraries**):

- **FastLED**
- **ArduinoJson** (v7.x)
- **NewPing**
- **Mozzi** (Install from the .zip file available at the Mozzi website)

See `LIBRARY_INSTALLATION_TROUBLESHOOTING.md` for detailed help, especially for the `NewTone` library on macOS.

TouchDesigner Project Setup

The recommended method for connecting TouchDesigner to the Arduino is using a **Serial DAT** followed by a **JSON DAT**.

Network Structure:

1. **Serial DAT:**
 - **Port:** Your Arduino’s COM port.
 - **Baud Rate:** 9600
 - **Active:** On
2. **Select DAT:**
 - Selects the latest line from the Serial DAT to avoid processing old data.
 - **Select Rows:** by Index
 - **Start/End Row Index:** `me.inputs[0].numRows-1`
3. **JSON DAT:**
 - Connect the Select DAT to the JSON DAT.
 - The JSON DAT will automatically parse the data.

Arduino-TouchDesigner Integration

The Arduino sends JSON data over serial in this format: `{"left":27,"right":189,"hands_detected":false}`

You can access this data in TouchDesigner expressions:

- **Left Sensor:** `op('json_sensor').result['left']`
- **Right Sensor:** `op('json_sensor').result['right']`
- **Hands Detected:** `op('json_sensor').result['hands_detected']`

Use these expressions to drive parameters in your effects. For example, to control the scale of a `Transform TOP`:

```
1.0 + (op('json_sensor').result['left'] / 200.0) if op('json_sensor').result['hands_detected']
else 1.0
```

5. Troubleshooting

Hardware Issues

- **No Audio:** Check amplifier power, wiring, and audio source.
- **LEDs Not Working:** Verify `FastLED` library is installed, check LED strip type and power supply.
- **No Power:** Check the 12V power supply and all connections.

Network Problems

- **Cannot Access Digiquad:**
 - Confirm the Mac mini and Digiquad are on the same subnet (192.168.3.x).
 - Check all Ethernet cable connections.
 - Try pinging the Digiquad’s IP address.
- **No DMX from TouchDesigner:**
 - Ensure the DMX Out CHOP protocol matches the Digiquad setting.
 - Verify the universe mapping is correct.
 - Check for firewall issues on the Mac mini.
 - **Remember to reboot the Digiquad after changing its protocol.**

Sensor Errors

- **Sensor Reads 0 Consistently:**
 - **Cause:** Missing `pinMode()` setup in the Arduino `setup()` function.
 - **Fix:** Ensure `pinMode(trigPin, OUTPUT);` and `pinMode(echoPin, INPUT);` are present for each sensor.
- **“Stuck Echo” Problem (Sensor stops responding):**
 - **Cause:** Some HC-SR04 sensors don’t auto-reset.
 - **Fix:** Implement a reset fix in your sensor reading code by briefly setting the echo pin to an output and driving it low. See `SENSOR_TROUBLESHOOTING.md` for the code snippet.
- **NewPing and Mozzi Library Conflicts:**
 - **Cause:** Timer conflicts between the two libraries.
 - **Fix:** Use the direct `pulseIn()` method for reading sensors instead of the `NewPing` library.

Software and Compilation Errors

- **NewTone.h: No such file or directory:**
 - **Cause:** The `NewTone` library is not in the correct location for the Arduino IDE.
 - **Fix:** Follow the instructions in `LIBRARY_INSTALLATION_TROUBLESHOOTING.md`. This involves moving the library to `~/Documents/Arduino/libraries/` on macOS and changing `#include "NewTone.h"` to `#include <NewTone.h>`.
 - **JSON Errors in TouchDesigner:**
 - **“Failed to parse DAT as JSON”:** Check that the `Select DAT` is correctly configured to only pass the latest complete line of JSON.
 - **Data not updating:** Ensure the `Serial DAT` is active and connected to the correct COM port.
-