Magnetron Sputtering System - Technical Manual

Document Version: 1.0
Last Updated: August 21, 2025
System: Vacuum Sputter Control System
Hardware Platform: Arduino Mega 2560 R3

Table of Contents

System Overview

The Magnetron Sputtering System is controlled by an Arduino Mega 2560 R3 microcontroller that manages:

- **23 relay outputs** for pump, valve, and power control
- **4 digital inputs** for safety interlocks
- **4 analog inputs** for pressure monitoring
- **Serial communication** at 9600 baud for PC interface

Key Components:

- Main chamber with vacuum pumps
- Load-lock chamber
- Gas delivery system
- Safety interlock system
- Pressure monitoring system

Electronics Control Box

Figure: Electronics control box showing Raspberry Pi, Arduino Mega, relay banks, power supplies, and multi-pin feedthroughs.

This enclosure contains the main control electronics for the sputter system. It houses a Raspberry Pi 5 (host GUI and supervisory controller), an Arduino Mega 2560 R3 (real-time I/O and relay control), relay modules, DC power supplies, terminal blocks and multi-pin feedthroughs for chamber and auxiliary I/O. Wiring is bundled and labeled (ish); feeds for pumps, valves, sensors, and mains pass through the enclosure's feedthroughs. The Raspberry Pi runs the high-level GUI and coordination logic while the Arduino handles low-latency digital and analog I/O. The box also contains fusing, earth grounding, and isolation components as required for safe operation.

Operation Modes

The system operates in three distinct modes that control user access to functions and safety checks:

Normal Mode (Default)

Button Access:

- **Automatic Procedures Only:** PUMP, VENT, SPUTTER, VENT Load-lock, Load/Unload
- **Manual Controls:** Disabled (grayed out)

Safety System:

- **Full Safety Checks:** All safety conditions enforced
- **Interlock Protection:** Operations blocked if interlocks violated
- **Confirmation Dialogs:** Required for potentially dangerous operations

Use Case: Standard production operation with maximum safety protection

Manual Mode

Button Access:

- **All Buttons Enabled:** Both automatic procedures and manual controls
- **Full GUI Access:** Complete control over all system functions

Safety System:

- **Full Safety Checks:** All safety conditions enforced
- **Interlock Protection:** Operations blocked if interlocks violated
- **Confirmation Dialogs:** Required for potentially dangerous operations

Use Case: Setup, maintenance, and advanced operations requiring manual control

Override Mode ⚠️

Button Access:

- **All Buttons Enabled:** Complete unrestricted access to all functions
- **No Restrictions:** All manual and automatic controls available

Safety System:

- ⚠️ **ALL SAFETY CHECKS BYPASSED** ⚠️
- **No Interlock Protection:** Operations allowed even with violated interlocks
- **No Confirmation Dialogs:** Direct execution without safety prompts
- **No Condition Checking:** Analog thresholds and digital inputs ignored

Use Case: Emergency operation, system recovery, advanced troubleshooting

⚠️ **WARNING:** Override mode completely bypasses all safety systems. Use only when absolutely necessary and with full understanding of system hazards. Ensure proper PPE and safety protocols are followed.

Hardware Pin Assignments

Arduino Mega 2560 R3 Pin Allocation

Pin Range	Function	Count	Description
22-37	Relay Outputs	16	Primary relay control pins
38-41	Relay Outputs	4	Spare relay positions (wires only)
44	Relay Output	1	Scroll pump control
46	Relay Output	1	**CRITICAL: Mains power control**
48	Relay Output	1	Spare relay for future upgrades
45, 47, 49, 51	Digital Inputs	4	Safety interlock inputs
A1-A4	Analog Inputs	4	Pressure sensor inputs & turbo spin speed (%)

Analog inputs

	Analog PINS	Connects to	WIRE COLOR	NOTES
0	A0	nan	nan	nan
1	A1	Analog out A from the gauge controller	yellow	Load-lock pirani gauge
2	A2	Analog out B from the gauge controller	dark red	Chamber pirani gauge
3	A3	Analog out from the gauge controller	green	Ion gauge
4	A4	Analog in TMP speed	fat red	Analog speed output from Turbo controller.
5	A5	Not connected	nan	nan

Analog inputs

	DIGITAL In/Out PINS	Connects to	WIRE COLOR	Controls	Notes
0	22	Relay 1	Typically red	Mains power relay	Mains power relay to the DC/RF power supplies. Activates in sputter mode.
1	23	Relay 2	"	Rough valve	This is the valve between TMP & scroll
2	24	Relay 3	"	Vent valve	nan
3	25	Relay 4	"	Chamber rough valve	This pumps down the chamber with scroll
4	26	Relay 5	"	Load-lock rough	24V out to operate load-lock rough valve.
5	27	Relay 6	"	Load-lock vent	24V out to operate load-lock vent valve.
6	28	Relay 7	"	Load-lock Gate	24V out to operate load-lock gate valve.
7	29	Relay 8	"	Gas valve 1	24V out to operate Ar gas in valve.
8	30	Relay 9	"	Gas valve 2	24V out to operate N2 gas in valve.
9	31	Relay 10	"	Gas valve 3	24V out to operate O2 gas in valve.
10	32	Relay 11	"	Shutter 1	24V out to operate shutter 1 movement.
11	33	Relay 12	"	Shutter 2	24V out to operate shutter 2 movement.
12	34	Relay 13	"	Empty	nan
13	35	Relay 14	"	Ion gauge on/off	Pulsed in order to deliver signal to pressure gauge controller to turn on ion gauge. 5V.
14	36	Relay 15	"	Turbo gate valve	24V out to operate main gate valve.

	DIGITAL In/Out PINS	Connects to	WIRE COLOR	Controls	Notes
15	37	Relay 16	"	Turbo on/off	Connected to turbo controller ground 0V. Connecting pin 1 on EXDC controller to GND gives signal to operate turbo.
16	44	Scroll pump solid state relay +ve	red	nan	Activates the scroll pump relay when high (5V) to power he scroll & turn it on. This relay is in the electronics box.
17	45	Water switch signal	pink	nan	Connects to GND when water flowing.
18	46	nan	nan	nan	nan
19	47	Rod Home Switch signal	purple	nan	Connects to GND when rod home
20	48	nan	nan	nan	nan
21	49	Door Switch signal	orange	nan	This is +24V as the door switch is supplied with 24V. I have created transistor inversion circuit to deliver GND potential to pin 49 when door closed.
22	50	nan	nan	nan	nan

25 Pin D-sub connectors In/out to electronics box

S1 Inside Electronics Box

	Pin	Function	Wire Color	Notes
0	25 pin male S1	nan	nan	nan
1	1	Relay 7 on board 2, Turbo gate valve	brown	24V out to open the turbo gate valve via solenoid
2	2	Relay 5, board 1. Load-lock rough.	white	24V out to open solenoid to open load lock pump valve
3	3	Relay 2, board 1. Rough valve solenoid	black	24V out to open solenoid to open rough valve
4	4	Relay 3, board 2. Shut 2	red	24V out to open solenoid to open/close shutter on G2
5	5	Relay 4, board 2. Shut 1	green	24V out to open solenoid to open/close shutter on G1
6	6	Relay 7, board 1. LL valve open	yellow	24V out to open solenoid to open load lock gate valve I think

	Pin	Function	Wire Color	Notes
7	7	Relay 6, board 1. Load-lock vent valve	purple	24V out to open solenoid to open load lock vent valve
8	8	nan	nan	nan
9	9	nan	nan	nan
10	10	nan	nan	nan
11	11	nan	nan	nan
12	12	Relay 4, board 1. Chamber roughing valve solenoid	nan	24V out to open solenoid to open chamber rough valve
13	13	nan	nan	nan
14	14	nan	nan	nan
15	15	nan	nan	nan
16	16	+12V from power supply	nan	To power the relay board modules
17	17	GND/-ve from 12V power supply	nan	""
18	18	Relay 2, board 2. G3	orange	24V out to turn on G3 I think
19	19	Relay 1, board 2. G2	pink	24V out to turn on G2 I think
20	20	Relay 8, board 1. G1	grey	24V out to turn on G1 I think
21	21	Relay 3, board 1. Vent	light blue	Relay 3, board 1. 24V out to open vent valve
22	22	nan	nan	nan
23	23	nan	nan	nan
24	24	nan	nan	nan
25	25	nan	nan	nan

S2 Inside Electronics Box

	Connector	Connects to/labelled as	Wire color	Arduino PIN	Notes
0	25 pin female S2	nan	nan	nan	nan
1	1	Analog out from the gauge controller	green	A1	LabJack: AI2. There are 3 analog out from the gauge controller
2	2	Analog out A from the gauge controller	yellow	A2	LabJack: AI0. labelled 'analog out', 'analog out A', 'analog out B'.

	Connector	Connects to/labelled as	Wire color	Arduino PIN	Notes
3	3	Analog out B from the gauge controller	red	A3	LabJack: AI1. Analog input gauge reading
4	4	Rod home switch	purple	nan	LabJack: IO1. Must be digital input
5	5	Water switch	pink	nan	LabJack: IO0.
6	6	Door switch	Light blue	nan	5V out to door switch. Brown wire on outside
7	7	Door switch	orange	nan	Goes into LabJack: IO2. Must be door switch high/low signal
8	8	Into Relay 6, Board 2 - IG, switched pin	grey	nan	Must switch to connect these two 8-9 in order to give IG on signal
9	9	Into Relay 6, Board 2 - IG, common pin	black	nan	May be a pin to connect to PS -ve to turn on IG, just like with turbo
10	10	TMP connector PIN 1 On/off	red	nan	Into relay 8, board 2 (relay 16) switched pin
11	11	TMP 80V PS -ve PIN	black	nan	Into relay 8, board 2 (relay 16) common pin
12	12	nan	nan	nan	nan
13	13	Analog in TMP speed	red	A4	LabJack: AI4
14	14	GND	Fat Grey white stripe	nan	These 14 - 16 are the ground side of the analog outs from the gauge controller.
15	15	GND	Fat Grey white stripe	nan	"
16	16	GND	Fat Grey white stripe	nan	"
17	17	5V out to interlock switches	Fat Grey	nan	Rod Home switch
18	18	5V out to interlock switches	Fat Grey	nan	Water Switch.
19	19	GND	Fat Grey white stripe	nan	Door switch ground
20	20	GND	Fat Grey white stripe	nan	nan
21	21	nan	nan	nan	nan
22	22	nan	nan	nan	nan

	Connector	Connects to/labelled as	Wire color	Arduino PIN	Notes
23	23	nan	nan	nan	nan
24	24	nan	nan	nan	nan
25	25	nan	nan	nan	nan

S1 Outside Electronics Box

	Connector	Coming from	Wire color	Notes
0	25 pin female S2	nan	nan	nan
1	1	Analog out from the gauge controller	red	LabJack: AI2. There are 3 analog out from the gauge controller
2	2	Analog out A from the gauge controller	red	LabJack: AI0. labelled 'analog out', 'analog out A', 'analog out B'.
3	3	Analog out B from the gauge controller	red	LabJack: AI1. Analog input gauge reading
4	4	Rod home switch	red	LabJack: IO1. Must be digital input
5	5	Water switch	red	nan
6	6	Door switch	brown	nan
7	7	Door switch	green/yellow GND	nan
8	8	Remote I/O d-sub 9 pin from gauge controller	blue	nan
9	9	Remote I/O d-sub 9 pin from gauge controller	red	nan
10	10	nan	nan	nan
11	11	nan	nan	nan
12	12	nan	nan	nan
13	13	nan	nan	nan
14	14	Analog out from the gauge controller	black	nan
15	15	Analog out A from the gauge controller	black	nan
16	16	Analog out B from the gauge controller	black	nan

	Connector	Coming from	Wire color	Notes
17	17	Rod home switch	black	Green and fat grey wires input 5V to these from inside the box
18	18	Water switch	black	These have +5 V into black lines and black the +ve red go to AI 0 - 3 on LJ
19	19	Door switch	blue	nan
20	20	PROBS NC	nan	nan
21	21	nan	nan	nan

Relay Configuration

Primary Relay Assignments (Pins 22-37)

Relay #	Arduino Pin	Button Name	Function	Description
1	22	btnMainsPower	🔴 **Mains Power**	**CRITICAL: Magnetron power supply mains control**
2	23	btnValveBacking	Backing Valve	Backing pump isolation valve
3	24	btnValveVent	Vent Valve	Chamber vent valve
4	25	btnValveRough	Rough Valve	Rough pump valve
5	26	btnValveLoadLockRough	LL Rough Valve	Load-lock rough pump valve
6	36	btnValveLoadLockVent	LL Vent Valve	Load-lock vent valve
7	28	btnValveLoadLockGate	LL Gate Valve	Load-lock gate valve
8	29	btnValveGas1	Gas 1 Valve	Process gas 1 control
9	30	btnValveGas2	Gas 2 Valve	Process gas 2 control
10	31	btnValveGas3	Gas 3 Valve	Process gas 3 control
11	32	btnShutter1	Shutter 1	Target shutter 1
12	33	btnShutter2	Shutter 2	Target shutter 2
13	34	*Reserved*	*Future Use*	Available for expansion
14	35	btnIonGauge	Ion Gauge	Ion gauge power control
15	27	btnValveTurboGate	Turbo Gate Valve	Turbo pump gate valve
16	37	btnPumpTurbo	Turbo Pump	Turbo pump power

Spare Relay Positions (Pins 38-41)

Relay #	Arduino Pin	Status	Notes
17	38	Spare Wire	Physical relay not installed
18	39	Spare Wire	Physical relay not installed
19	40	Spare Wire	Physical relay not installed
20	41	Spare Wire	Physical relay not installed

Critical Safety and Special Relay Assignments

Function	Arduino Pin	Button Name	Relay #	Description
Scroll Pump	44	btnPumpScroll	21	Backing pump control
Spare 1	**46**	**btnSpare1**	**22**	**Available for future expansion**
Spare Relay	48	btnSpareRelay	23	Available for future upgrades

📌 **Note:** Pin 46 (btnSpare1) is wired but currently unassigned. Mains power control (btnMainsPower) is assigned to pin 22 (Relay #1).

🔴 CRITICAL SAFETY - Mains Power Relay (Pin 22)

Function: Controls mains voltage switch for magnetron sputtering power supplies

Safety Integration:

- Mains power relay (btnMainsPower) on pin 22 provides master power control
- Can be controlled through GUI during sputter procedure
- Subject to safety conditions defined in safety_conditions.yml
- Requires all critical interlocks satisfied before enabling

Critical Interlocks Required:

1. **Water Flow Interlock (Pin 45):** Cooling water must be flowing
2. **Rod Position Interlock (Pin 47):** Sample rod must be in safe position
3. **Door Interlock (Pin 49):** Chamber door must be closed

⚠️ **WARNING:** This relay controls lethal mains voltage to high-power magnetron supplies. The safety system enforces strict interlock requirements before allowing power to be enabled.

Digital Input Configuration

Safety Interlock Inputs

Input #	Arduino Pin	Function	Safe State	GUI Indicator	Description
0	45	Water Flow Interlock	LOW (switch closed)	indWater	Cooling water flow sensor
1	47	Rod Position Interlock	LOW (switch closed)	indRod	Sample rod position sensor
2	49	Door Interlock	LOW (switch closed)	indDoor	Chamber door closed sensor
3	51	Spare Interlock	LOW (switch closed)	*(none)*	Reserved for future use

Interlock Logic

Hardware Configuration:

- **Internal Pull-ups Enabled:** 20-50kΩ pull-up resistors to +5V
- **Active Low Logic:** Switches connect pin to ground when activated
- **Fast Response:** ~20-50ms switching time (no floating inputs)

Safe State Definition:

- `LOW (false)` at Arduino pin = Switch closed/activated = Interlock satisfied (safe to operate)
- `HIGH (true)` at Arduino pin = Switch open/disconnected = Interlock violated (unsafe, operations blocked)

Software Logic Inversion:

- Arduino firmware inverts readings: `digitalInputStates[i] = !digitalRead(pin)`
- Python receives: `true` = safe state, `false` = unsafe state
- This maintains consistent high-level logic while using reliable pull-up hardware

Visual Indicators:

- **Green:** Interlock OK (switch closed, safe state)
- **Red:** Interlock triggered (switch open, unsafe state)
- **Gray:** No connection/unknown state

Physical Switch Wiring:

```
Arduino Pin 45 ----[Water Switch]--- GND
Arduino Pin 47 ----[Rod Switch]----- GND
Arduino Pin 49 ----[Door Switch]---- GND
Arduino Pin 51 ----[Spare Switch]--- GND
```

IMPORTANT - Inverted Logic Operation:

- **Physical switches connect Arduino pins to GND when activated (safe state)**
- **NO connection to +5V required** - internal pull-ups provide the HIGH state
- **Switch closed → Pin LOW → Safe condition → Green indicator**
- **Switch open → Pin HIGH → Unsafe condition → Red indicator**

Benefits of Inverted Logic Design:

- **Improved Safety:** Open/broken wires default to unsafe state (fail-safe)
- **Lower Power Consumption:** No current flow through switches when closed
- **Fast Response:** Internal pull-ups eliminate floating inputs (~20-50ms switching)
- **Simplified Wiring:** Only one wire per switch (to GND), no +5V connection needed

Safety Integration:

- Water flow interlock prevents pump operations without cooling water flow
- Rod position interlock prevents process operations when rod in unsafe position
- Door interlock prevents all vacuum operations when chamber door open
- Spare interlock available for additional safety features

Analog Input Configuration

Pressure Monitoring Inputs

Input #	Arduino Pin	Label	GUI Display	Scale	Offset	Description
0	A1	Load-lock (Torr)	lcdAnalog1	1.0	0.0	Load-lock chamber pressure
1	A2	Chamber (Torr)	lcdAnalog2	1.0	0.0	Main chamber pressure
2	A3	Ion Gauge (Torr)	lcdAnalog3	1.0	0.0	Ion gauge pressure reading
3	A4	Turbo Spin (%)	lcdAnalog4	25.0	-12.5	Turbo spin speed (0.5-4.5V → 0-100%)

Analog Signal Processing

ADC Resolution: 10-bit (0-1023 counts)
Input Voltage Range: 0-5V DC
Scaling Formula: `Display_Value = (ADC_Reading × Scale) + Offset`
Update Rate: 700ms polling interval

Pressure Sensor Integration:

- Raw ADC values converted to engineering units
- Configurable scaling and offset per channel
- Real-time display on GUI LCD widgets
- Safety system monitoring for out-of-range conditions

Pin Assignment Summary

Complete Pin Allocation Table

Arduino Pin	Function	Direction	Signal Type	Connected Device
22	Relay 1	Output	Digital	🔴 **Mains Power (CRITICAL)**
23	Relay 2	Output	Digital	Backing Valve
24	Relay 3	Output	Digital	Vent Valve
25	Relay 4	Output	Digital	Rough Valve
26	Relay 5	Output	Digital	LL Rough Valve
36	Relay 6	Output	Digital	LL Vent Valve
28	Relay 7	Output	Digital	LL Gate Valve
29	Relay 8	Output	Digital	Gas 1 Valve
30	Relay 9	Output	Digital	Gas 2 Valve
31	Relay 10	Output	Digital	Gas 3 Valve
32	Relay 11	Output	Digital	Shutter 1
33	Relay 12	Output	Digital	Shutter 2
34	Relay 13	Output	Digital	*Reserved*
35	Relay 14	Output	Digital	Ion Gauge
27	Relay 15	Output	Digital	Turbo Gate Valve

Arduino Pin	Function	Direction	Signal Type	Connected Device
37	Relay 16	Output	Digital	Turbo Pump
38	Relay 17	Output	Digital	Spare Wire
39	Relay 18	Output	Digital	Spare Wire
40	Relay 19	Output	Digital	Spare Wire
41	Relay 20	Output	Digital	Spare Wire
44	Relay 21	Output	Digital	Scroll Pump
46	Relay 22	Output	Digital	Spare (btnSpare1)
48	Relay 23	Output	Digital	Spare Relay
45	Digital Input 0	Input	Digital	Water Flow Interlock
47	Digital Input 1	Input	Digital	Rod Position Interlock
49	Digital Input 2	Input	Digital	Door Interlock
51	Digital Input 3	Input	Digital	Spare Interlock
A1	Analog Input 0	Input	Analog	Load-lock Pressure
A2	Analog Input 1	Input	Analog	Chamber Pressure
A3	Analog Input 2	Input	Analog	Ion Gauge Pressure
A4	Analog Input 3	Input	Analog	Turbo Spin Speed (%)

Reserved/Unused Pins

Available for Future Expansion:

- Digital pins: 0-21, 42-43, 50, 52-53
- Analog pins: A5-A15
- Serial pins: 0 (RX), 1 (TX) - reserved for PC communication
- SPI pins: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS) - available if needed

Note: Pins 46 and 48 have been allocated for critical mains power control and spare relay expansion respectively.

End of Technical Manual Section 1

This document will be expanded with additional sections covering software architecture, safety systems, operational procedures, and maintenance protocols.