

NanoArc ACE Mira

KiCad Wiring Workbook

RevA low-voltage controller reference for schematic capture, net naming, and breadboard translation

Document role	Bench reference while drawing the KiCad schematic and wiring RevA modules
Scope	ACE/control electronics, panel/HID, storage, ADC/sensor simulation, low-voltage fault logic, and connector placeholders
Out of scope	Energy-storage construction, high-current output conductors, charger design, electrode construction, and operating procedures
Source baseline	NanoArc Master Technical Architecture RevA Freeze 1.2, Control System and HID Manual RevA, Missing Items BOM v2, current shipment/cart notes
Generated	2026-07-10

Use this PDF as the wiring truth source while creating the KiCad schematic. The goal is not to route a PCB yet. The goal is to make a clean schematic hierarchy with stable net names, then build/test one RevA module at a time.

Safety boundary: this workbook intentionally stops at low-voltage controller logic and simulated/conditioned interfaces. Do not use it as a construction guide for an energized output stage. Any future Weld Energy Stage signals must enter ACE only through deliberately conditioned sense/control interfaces.

Contents

- 1. How to use this workbook
- 2. KiCad project structure and sheet hierarchy
- 3. Net naming rules and global labels
- 4. RevA block map
- 5. Nano Every carrier and direct pins
- 6. I2C backbone, addresses, and FRAM breakout
- 7. MCP23017 GPIO expanders
- 8. Panel input matrix and rotary encoder
- 9. Panel output LEDs, buzzer, and ULN2803A
- 10. SPI ADC, precision reference, and analog simulators
- 11. Fault and safety-support logic
- 12. Protection, decoupling, connectors, and test points
- 13. Parts-to-circuit map
- 14. KiCad 6 practical commands and symbol search terms
- 15. KiCad step-by-step build procedure
- 16. Breadboard translation and staged test checklist
- Appendix A. DIP pin maps
- Appendix B. Net dictionary
- Appendix C. Symbol and footprint notes

1. How to use this workbook

This document is a practical companion to the NanoArc ACE RevA architecture. It answers one question: when you open KiCad, what symbols, sheets, nets, connectors, and footprints should you place so the schematic becomes a reliable wiring plan?

RevA Mira is treated as a bench controller. The firmware can be debugged over serial while the OLED is deferred. I2C expanders, FRAM, ADC, simulated inputs, panel switches, talkbacks, and low-voltage fault paths can all be captured and tested without any energized Weld Energy Stage hardware.

Recommended workflow

Step	Action	Result
1	Create the KiCad hierarchy first.	The schematic becomes the named wiring contract instead of a drawing made after the fact.
2	Place modules by sheet: Nano, I2C, panel input, panel output, ADC, fault logic, machine I/O, power/protection.	Each breadboard/perfboard section has a matching schematic section.
3	Use global labels only for true cross-sheet signals.	Direct nets such as I2C_SDA, SPI_MISO, FAULT_SUM, and +5V_LOGIC are unambiguous.
4	Add connectors/test headers deliberately.	Every wire leaving a sheet has a named connector pin or test point.
5	Build one block at a time.	Failures stay local: Nano boot, I2C scan, one MCP23017, keypad, LEDs, ADC, then loopbacks.

Rule: do not draw the entire thing and then power it as one giant breadboard. Capture and test one block at a time.

2. KiCad project structure and sheet hierarchy

Create a normal KiCad 6 project. The root schematic is only a block diagram with hierarchical sheets and shared rails. Do not cram the entire controller into the root page.

```
NanoArc_ACE_Mira/  
  NanoArc_ACE_Mira.kicad_pro  
  NanoArc_ACE_Mira.kicad_sch          # Root sheet only  
  01_Nano_Carrier.kicad_sch  
  02_I2C_Backbone.kicad_sch  
  03_Panel_Input_0x20.kicad_sch  
  04_Panel_Output_0x21.kicad_sch  
  05_SPI_ADC_Analog.kicad_sch  
  06_Fault_Logic.kicad_sch  
  07_Machine_IO_0x22.kicad_sch  
  08_Power_Protection.kicad_sch  
  lib/  
    NanoArc_Modules.kicad_sym  
    NanoArc_FRAM.pretty/
```

Root sheet blocks

Root block	Purpose	Main cross-sheet nets
01_Nano_Carrier	Nano Every socket, service serial, direct safety pins, SPI/I2C roots	+5V_LOGIC, GND, I2C_SDA, I2C_SCL, SPI_MOSI, SPI_MISO, SPI_SCK, ADC_CS, direct I/O nets
02_I2C_Backbone	I2C pullups, FRAM breakout, I2C expansion header	I2C_SDA, I2C_SCL, FRAM_WP
03_Panel_Input_0x20	MCP23017 at 0x20, keypad matrix, encoder	I2C_SDA, I2C_SCL, EXPANDER_INT, KEY_*, ENC_*
04_Panel_Output_0x21	MCP23017 at 0x21, ULN2803A, LEDs, buzzer, OLED placeholder	TB_*, BUZZER, OLED_RST

Root block	Purpose	Main cross-sheet nets
05_SPI_ADC_Analog	MCP3208, REF5025 adapter, MCP6004, analog simulator inputs	SPI_*, ADC_CS, VREF_2V5, AIN*
06_Fault_Logic	Comparators, latch, monostable pulse-window test fixture	FAULT_SUM, TRIP_OV, TRIP_OC, HW_INHIBIT, PULSE_CMD
07_Machine_IO_0x22	MCP23017 at 0x22 for low-voltage machine loopbacks and placeholders	CHARGER_READY, BLEED_CONFIRM, FAN_ENABLE, SERVICE_KEY, etc.
08_Power_Protection	Decoupling, rail bulk caps, test points, TVS/ESD placeholders	+5V_LOGIC, GND, connector nets

3. Net naming rules and global labels

Use stable net names. The schematic, firmware constants, breadboard labels, and eventually perfboard labels should all use the same names. Avoid local labels such as signal1 or button5.

Conventions

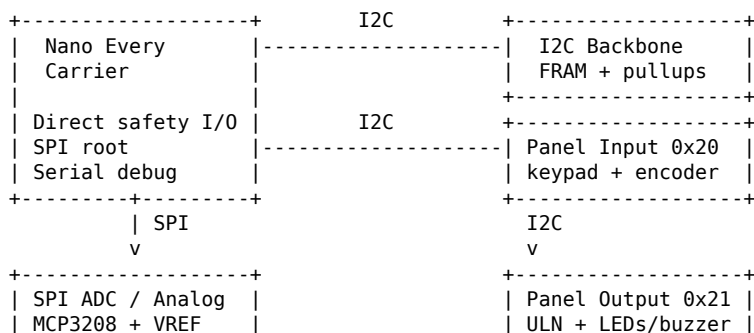
Pattern	Use	Examples
ALL_CAPS	Project net names	FAULT_SUM, I2C_SDA, TB_READY
Prefix + role	Group related nets	KEY_ROW0, KEY_COL3, ENC_A, AIN5
_N suffix	Active-low signal only when needed	RESET_N, INT_N if drawn active-low
Address in sheet name	I2C expander identity	Panel_Input_0x20, Panel_Output_0x21
+5V_LOGIC	Logic rail	Do not call this VCC everywhere unless the KiCad symbol forces it
GND	Common low-voltage controller ground	Use one GND symbol family consistently

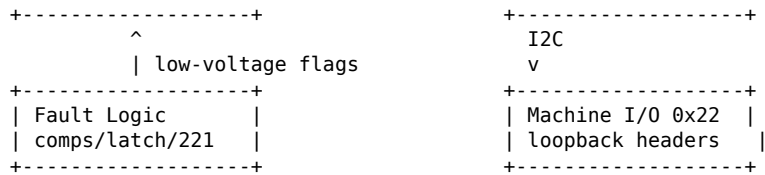
Global labels to create first

Power: +5V_LOGIC, GND, VREF_2V5
I2C: I2C_SDA, I2C_SCL, EXPANDER_INT
SPI: SPI_MOSI, SPI_MISO, SPI_SCK, ADC_CS
Direct input: FAULT_SUM, TRIP_OV, ARM_SW, HARD_SAFE_SW, GRIP_SW, TRIGGER_SW, OUTPUT_FB, HW_INHIBIT, ENCLOSURE_INTLK, HANDPIECE_PRESENT, TRIP_OC
Direct out: PULSE_CMD, CHARGE_ENABLE
Panel: KEY_ROW0..KEY_ROW3, KEY_COL0..KEY_COL4, ENC_A, ENC_B, ENC_SW
Talkbacks: TB_SAFE, TB_CHARGE, TB_READY, TB_ARMED, TB_OUTPUT, TB_FAULT, TB_ENERGY, TB_OPR_ERR
Analog: AIN0_VBANK, AIN1_VIN_AUX, AIN2_ICHG_MON, AIN3_IOUT_MON, AIN4_TBANK, AIN5_TSTAGE, AIN6_TACE, AIN7_THAND

4. RevA block map

The RevA schematic should show the controller as a collection of boards/modules. A module can be a breadboard today and a perfboard tomorrow. This keeps KiCad useful even before a routed PCB exists.





The Machine I/O sheet is a low-voltage placeholder and loopback interface. It must not become a dumping ground for raw pulse-current wiring.

5. Nano Every carrier and direct pins

The Nano Every carrier is the first sheet to draw and the first hardware to build. It owns serial debug, power entry to the logic system, and the direct pins that should not be hidden behind I2C expanders.

Symbols and footprints

Item	KiCad symbol	Footprint / representation	Notes
Arduino Nano Every	Use Arduino_Nano or custom Nano Every module symbol	Two 1x15 2.54 mm socket strips	Represent as a removable module, not as the ATmega4809 chip.
Nano sockets	Connector_Generic:Conn_01x15 x2	Socket_Strip_Straight_1x15_P2.54mm or equivalent	Two Samtec SSW-115-01-F-S sockets on the carrier.
Service UART	Conn_01x04 or Conn_01x06	PinHeader_1x04/1x06_P2.54mm	Label TX, RX, GND, +5V if used; serial monitor is the temporary display.
Test points	TestPoint	TestPoint_THTPad_D1.0mm or pin header	Put test points on +5V_LOGIC, GND, I2C_SDA/SCL, SPI, and each direct safety net.

Frozen direct pin map for RevA schematic

Nano pin	Net	Direction	Why direct
D2	FAULT_SUM	Input	Aggregated fault path; should remain visible to firmware immediately.
D3	TRIP_OV	Input	Independent over-voltage trip/simulator input.
D4	ARM_SW	Input	Physical arm request/state input.
D5	HARD_SAFE_SW	Input	Hard-safe/abort/inhibit switch path.
D6	GRIP_SW	Input	Grip or handpiece intent input/simulator.
D7	TRIGGER_SW	Input	Trigger request input; firmware authorizes, operator only requests.
D8	OUTPUT_FB	Input	Low-voltage feedback or simulator state for output-command verification.
D9	PULSE_CMD	Output	Low-voltage command line only; no raw output-stage wiring.
D10	ADC_CS	Output	SPI chip select for MCP3208.
D11	SPI_MOSI	Output	SPI MOSI.
D12	SPI_MISO	Input	SPI MISO.
D13	SPI_SCK	Output	SPI clock.
A0	HW_INHIBIT	Input	Hardware inhibit/supervisor state input.
A1	CHARGE_ENABLE	Output	Low-voltage enable/simulator line.
A2	ENCLOSURE_INTLK	Input	Interlock/simulator input.
A3	EXPANDER_INT	Input	Shared interrupt from MCP23017 expanders, optional in early firmware.
A4	I2C_SDA	Bidirectional	I2C data.
A5	I2C_SCL	Output	I2C clock.
A6	HANDPIECE_PRESENT	Input	Low-voltage presence/simulator input.
A7	TRIP_OC	Input	Independent current-trip/simulator input.

Direct input schematic pattern

For early breadboard/testing, draw each direct input as a connector or switch to a known state. The exact pull-up/pull-down policy should match firmware assumptions; the important KiCad point is that each direct signal has a visible default state and a test point.

Example direct input pattern:

```
+5V_LOGIC --- 4.7k/10k --- NET_NAME --- switch/jumper --- GND
                        |
                        +--- test point
                        +--- Nano direct input pin
```

6. I2C backbone, addresses, and FRAM breakout

Draw the I2C bus as its own sheet. This keeps pullups and address choices in one place. Every I2C device then connects to the same I2C_SDA/I2C_SCL global labels.

I2C backbone wiring

Net	Connection	Notes
I2C_SDA	Nano A4, FRAM SDA, MCP23017 SDA pins	One bus shared by all I2C devices.
I2C_SCL	Nano A5, FRAM SCL, MCP23017 SCL pins	One bus shared by all I2C devices.
SDA pullup	4.7 kOhm from I2C_SDA to +5V_LOGIC	Fit once on ACE carrier/backbone unless module pullups force adjustment.
SCL pullup	4.7 kOhm from I2C_SCL to +5V_LOGIC	Fit once on ACE carrier/backbone unless module pullups force adjustment.
EXPANDER_INT	Nano A3 plus MCP23017 INT pins if used	Can be omitted in first scanner test and added later.

I2C pullup rule: one deliberate pullup network is easier to debug than many accidental parallel pullups hidden on modules.

I2C address table

Device	Address	Address pins	Sheet
FRAM breakout	0x50	A2=0, A1=0, A0=0	02_I2C_Backbone
MCP23017 panel input	0x20	A2=0, A1=0, A0=0	03_Panel_Input_0x20
MCP23017 panel output	0x21	A2=0, A1=0, A0=1	04_Panel_Output_0x21
MCP23017 machine I/O	0x22	A2=0, A1=1, A0=0	07_Machine_IO_0x22

FRAM breakout representation

Treat the Adafruit FRAM breakout as a module with pins, not as the bare FRAM chip. Use the generated KiCad 6 library included separately.

FRAM pin	Wire to	RevA note
VCC	+5V_LOGIC	Power from the logic rail.
GND	GND	Common controller ground.
WP	GND or FRAM_WP	Tie low for normal writes, or route to a labeled jumper/control net if desired.
SCL	I2C_SCL	Shared I2C clock.
SDA	I2C_SDA	Shared I2C data.
A2	GND	Default address 0x50.
A1	GND	Default address 0x50.
A0	GND	Default address 0x50.

Generated FRAM symbol:
MOD_Adafruit_FRAM_1895_I2C

Generated footprints:

NanoArc_FRAM:Adafruit_FRAM_1895_I2C_HeaderOnly
NanoArc_FRAM:Adafruit_FRAM_1895_I2C_ModuleOutline

Use ModuleOutline when you want the schematic-to-board connection to remind you that the physical object is a breakout. Use HeaderOnly for pure perfboard wiring.

7. MCP23017 GPIO expanders

The MCP23017 devices are on backorder for a short time, but the schematic can be completed now. Use one identical symbol type three times with different address pin straps and reference designators.

Common MCP23017 wiring

MCP23017 pin	Name	Wire to	Notes
9	VDD	+5V_LOGIC	Add a 100 nF decoupling cap close to the package.
10	VSS	GND	Ground.
12	SCL	I2C_SCL	Shared I2C clock.
13	SDA	I2C_SDA	Shared I2C data.
18	RESET	+5V_LOGIC via pullup or direct for RevA	Optional reset test point/jumper.
20	INTA	EXPANDER_INT or local TP	Optional shared interrupt.
19	INTB	EXPANDER_INT or local TP	Optional shared interrupt.
15/16/17	A0/A1/A2	Address straps	See address table.

Panel input expander at 0x20

Port pin	Net	Purpose
GPA0	KEY_COL0	Keypad column 0
GPA1	KEY_COL1	Keypad column 1
GPA2	KEY_COL2	Keypad column 2
GPA3	KEY_COL3	Keypad column 3
GPA4	KEY_COL4	Keypad column 4
GPA5	KEY_ROW0	Keypad row 0
GPA6	KEY_ROW1	Keypad row 1
GPA7	KEY_ROW2	Keypad row 2
GPB0	KEY_ROW3	Keypad row 3
GPB1	ENC_A	Encoder phase A
GPB2	ENC_B	Encoder phase B
GPB3	ENC_SW	Encoder push switch
GPB4	PANEL_PRESENT	Panel presence jumper/input
GPB5	SPARE_IN0	Spare input
GPB6	SPARE_IN1	Spare input
GPB7	SPARE_IN2	Spare input

Panel output expander at 0x21

Port pin	Net	Purpose
GPA0	TB_SAFE	SAFE talkback LED
GPA1	TB_CHARGE	CHARGE talkback LED
GPA2	TB_READY	READY talkback LED
GPA3	TB_ARMED	ARMED talkback LED
GPA4	TB_OUTPUT	OUTPUT talkback LED
GPA5	TB_FAULT	FAULT talkback LED

Port pin	Net	Purpose
GPA6	TB_ENERGY	ENERGY talkback LED
GPA7	TB_OPR_ERR	OPR ERR talkback LED
GPB0	OLED_RST	Display reset placeholder; OLED deferred
GPB1	BUZZER_CMD	Buzzer command to ULN/input driver
GPB2	LAMP_TEST	Lamp test / diagnostic
GPB3	PANEL_DIM_OR_ENABLE	Panel enable/dim placeholder
GPB4-GPB7	SPARE_OUT*	Spare outputs

Machine I/O expander at 0x22

Port pin	Net	Purpose
GPA0	CHARGER_READY	Low-voltage status/simulator
GPA1	CHARGER_FAULT	Low-voltage status/simulator
GPA2	BLEED_CONFIRM	Low-voltage status/simulator
GPA3	FAN_TACH_OR_GOOD	Fan-good/tach placeholder
GPA4	AUX_INTERLOCK	Additional interlock/simulator
GPA5	SERVICE_KEY	Service mode input
GPA6	POWER_MODULE_PRESENT	Module-present/simulator
GPA7	FAULT_LATCH_VALID	Latch-status feedback
GPB0	BLEED_ENABLE	Low-voltage command placeholder
GPB1	FAN_ENABLE	Fan command placeholder
GPB2	CONTACT_TEST_ENABLE	Low-energy contact-test enable placeholder
GPB3	FAULT_LATCH_RESET	Reset command for fault latch
GPB4-GPB7	MACHINE_SPARE*	Spare machine I/O

8. Panel input matrix and rotary encoder

The panel input sheet is built around MCP23017 address 0x20. RevA can scan a diode-less matrix if firmware assumes one key at a time. If later you need reliable multi-key rollover, add per-key diodes in a future revision.

4x5 keypad matrix

	COL0	COL1	COL2	COL3	COL4
ROW0	VERB	NOUN	+	-	CLR
ROW1	7	8	9	KEY REL	RSET
ROW2	4	5	6	PRO	ENTR
ROW3	1	2	3	0	SPARE

Switch schematic pattern

Each tactile key goes between one KEY_ROWx net and one KEY_COLx net.
Do not connect one side of every key to ground; it is a matrix, not 20 direct buttons.

Example:

SW_VERB between KEY_ROW0 and KEY_COL0
 SW_NOUN between KEY_ROW0 and KEY_COL1
 SW_7 between KEY_ROW1 and KEY_COL0

Rotary encoder

Encoder terminal	Wire to	Notes
A	ENC_A	MCP23017 0x20 GPB1. Use pullups in firmware or external pullups if needed.

Encoder terminal	Wire to	Notes
B	ENC_B	MCP23017 0x20 GPB2.
C/common	GND	Common for mechanical encoder contacts.
Switch 1	ENC_SW	MCP23017 0x20 GPB3.
Switch 2	GND	Encoder push switch common.

9. Panel output LEDs, buzzer, and ULN2803A

The panel output expander should not directly carry every indicator and sounder load. Use the ULN2803A as an 8-channel low-side driver for the talkback LEDs, and optionally a separate channel/driver path for the buzzer depending on current and spare channels.

Talkback list

Talkback	Net	Meaning in panel
SAFE	TB_SAFE	System safe/inhibited state indication
CHARGE	TB_CHARGE	Charge-related state/placeholder
READY	TB_READY	Ready condition
ARMED	TB_ARMED	Armed state indication
OUTPUT	TB_OUTPUT	Output command/feedback indication
FAULT	TB_FAULT	Fault present
ENERGY	TB_ENERGY	Energy availability/placeholder state
OPR ERR	TB_OPR_ERR	Operator error/invalid entry

LED driver pattern

Preferred RevA pattern using ULN2803A low-side sinks:

```
+5V_LOGIC -> LED resistor -> LED anode -> LED cathode -> ULN2803 OUTn
MCP23017 output TB_* -----> ULN2803 INn
ULN2803 GND -----> GND
```

Result: the expander controls logic-level inputs; the ULN sinks lamp current.

ULN2803 channel assignment

ULN channel	Input net from MCP23017	Output load
1	TB_SAFE	SAFE LED cathode
2	TB_CHARGE	CHARGE LED cathode
3	TB_READY	READY LED cathode
4	TB_ARMED	ARMED LED cathode
5	TB_OUTPUT	OUTPUT LED cathode
6	TB_FAULT	FAULT LED cathode
7	TB_ENERGY	ENERGY LED cathode
8	TB_OPR_ERR	OPR ERR LED cathode

If the buzzer also needs a driver channel, either use a separate small transistor/driver in a later revision, or temporarily drive it through a spare expander output only if the current budget is acceptable. In KiCad, label the command net BUZZER_CMD and the load side BUZZER_LOAD so the design can change without renaming firmware logic.

10. SPI ADC, precision reference, and analog simulators

The MCP3208 ADC sheet allows firmware to read simulated voltages, current monitors, temperatures, and reference values before real hardware exists. This is a controller/simulator interface, not an output-stage measurement guide.

MCP3208 pin map

MCP3208 pin	Net	Purpose
1	AIN0_VBANK	Bank-voltage simulator / scaled placeholder
2	AIN1_VIN_AUX	Aux/input-voltage simulator
3	AIN2_ICHG_MON	Charge-current monitor simulator
4	AIN3_IOUT_MON	Output-current monitor simulator
5	AIN4_TBANK	Bank temperature simulator
6	AIN5_TSTAGE	Stage temperature simulator
7	AIN6_TACE	ACE board temperature simulator
8	AIN7_THAND	Handpiece temperature simulator
9	DGND	GND
10	ADC_CS	SPI chip select from Nano D10
11	SPI_MOSI	Data in from Nano
12	SPI_MISO	Data out to Nano
13	SPI_SCK	SPI clock from Nano
14	AGND	GND / analog ground in RevA
15	VREF_2V5	Precision reference input
16	+5V_LOGIC	ADC supply

REF5025 and MCP6004 roles

Part	Sheet role	KiCad representation
REF5025AID	2.5 V precision reference for ADC VREF	SOIC-8 symbol plus SOIC-to-DIP adapter symbol/footprint for RevA build
MCP6004-I/P	Quad op-amp for low-rate analog buffering/filtering/simulator conditioning	DIP-14 in socket
10 k trimmers/pots	Manual analog simulator sources for firmware bring-up	Potentiometer symbols from +V/ref to GND with wiper to AINx through protection/filter as chosen
100 nF caps	Local decoupling	One at MCP3208 VDD, REF5025, MCP6004 supply pins
10 uF cap	Local rail bulk	Near analog board power entry

For the first KiCad pass, draw analog channels as simulator inputs and headers. Do not invent final sensor scaling networks until the measurement requirements are frozen.

11. Fault and safety-support logic

This sheet captures the low-voltage safety-support and test-fixture logic used by ACE. It is intentionally separated from the output-energy hardware. Keep the schematic focused on command authorization, fault reporting, latching, and simulator/test signals.

Main parts

Part	Circuit role	KiCad note
LMV393IDR comparators	Independent threshold detectors for trip/simulator conditions	SOIC-8 on DIP adapter in RevA. Use open-collector output notation and pullups.
CD74HC221E	Independent non-retriggerable maximum-window / pulse-clamp test fixture	DIP-16 in socket. Keep RC timing components visible and labeled as test-fixture values.
SN74HC573AN	Fault latch / fault-source snapshot	DIP-20 in socket. Inputs from fault causes; outputs to low-voltage readback/header.

Part	Circuit role	KiCad note
1N4148	Small-signal steering/isolation	Use for logic steering/OR-ing where needed; label orientation clearly.

Fault-sheet net responsibilities

Net	Source/sink	Meaning
FAULT_SUM	Fault logic to Nano D2	Aggregate fault present. Draw as open-drain/open-collector style if multiple sources can pull it.
TRIP_OV	Comparator/test fixture to Nano D3	Independent over-voltage trip/simulator.
TRIP_OC	Comparator/test fixture to Nano A7	Independent over-current/current-trip simulator.
HW_INHIBIT	Hardware logic to Nano A0	Hardware inhibit state input.
PULSE_CMD	Nano D9 to clamp/test fixture	Low-voltage command line only.
OUTPUT_FB	Test fixture to Nano D8	Low-voltage feedback/simulator state.
FAULT_LATCH_RESET	Machine I/O 0x22 output	Reset/clear latch after firmware acknowledges.
FAULT_LATCH_VALID	Machine I/O 0x22 input	Latch has valid captured fault data.

Do not connect ACE directly to raw pulse-current conductors. For RevA, this sheet is allowed to contain only low-voltage command, feedback, trip, latch, and simulator/test-fixture signals.

12. Protection, decoupling, connectors, and test points

The power/protection sheet is where you make the design buildable and debuggable. It should show rails, decoupling, connector entry points, and where protection devices will eventually fit.

Decoupling plan

Capacitor	Where	Purpose
100 nF ceramic	One per IC power pin pair, physically close	High-frequency local decoupling.
10 uF electrolytic	One per module/board power entry	Local rail bulk for each breadboard/perfboard branch.
100 uF electrolytic	Main logic rail or larger branch	Board-level bulk capacitance.

Protection placeholders

Part	What to draw	When to physically fit
SMBJ5.0A	TVS from +5V_LOGIC to GND on the power/protection sheet	Fit after rail layout is defined; not important for the first breadboard boot.
TPD4E05U06DQAR	4-line ESD array symbols at connector/harness entries	Prefer future PCB or careful adapter; tiny package is not breadboard friendly.
1N4148	Logic steering/protection helpers	Fit where the fault/switch logic explicitly needs them.

Minimum test points

Power: TP_5V_LOGIC, TP_GND, TP_VREF_2V5
I2C: TP_I2C_SDA, TP_I2C_SCL
SPI: TP_SPI_MOSI, TP_SPI_MISO, TP_SPI_SCK, TP_ADC_CS
Direct: TP_FAULT_SUM, TP_ARM_SW, TP_HARD_SAFE_SW, TP_TRIGGER_SW,
TP_PULSE_CMD, TP_OUTPUT_FB, TP_HW_INHIBIT
Expander: TP_EXPANDER_INT, TP_MCP0_RESET, TP_MCP1_RESET, TP_MCP2_RESET
Panel: TP_KEY_ROW0..3, TP_KEY_COL0..4 if practical

13. Parts-to-circuit map

This section maps the purchased parts to the sheet where they belong. Use it as a sanity check when placing symbols.

Part	Qty	Sheet	Circuit purpose
Arduino Nano Every ABX00028	1	01_Nano_Carrier	Main ACE controller.
Samtec 1x15 socket strips	2	01_Nano_Carrier	Removable Nano mounting.
MCP23017-E/SP	3	03/04/07	I2C GPIO expansion: panel input, panel output, machine I/O.
ICS-628-T 28-pin sockets	4	03/04/07	MCP23017 sockets, three fitted plus spare.
TL1105 tactile switches	25	03_Panel_Input_0x20	4x5 keypad plus spares.
Bourns PEC11R encoder	1	03_Panel_Input_0x20	Rotary input and push switch.
Davies 1226-J knob	1	Panel mechanical	Encoder knob.
White 3 mm diffused LEDs	10/15	04_Panel_Output_0x21	Talkback indicators plus spares.
PUI 5 V buzzer	1	04_Panel_Output_0x21	Audible feedback.
ULN2803A	1	04_Panel_Output_0x21	Low-side LED/sounder driver.
ICS-318-T 18-pin socket	2	04_Panel_Output_0x21	ULN2803A socket plus spare.
Adafruit FRAM breakout	1	02_I2C_Backbone	Persistent storage: calibration, profiles, alarms, counters.
MCP3208-CI/P	1	05_SPI_ADC_Analog	8-channel SPI ADC.
ICS-316-T 16-pin sockets	3	05/06	MCP3208 and CD74HC221 sockets.
REF5025AID	1	05_SPI_ADC_Analog	2.5 V ADC reference.
SOIC-8 to DIP-8 adapters	pack	05/06	Adapters for REF5025 and LMV393IDR.
MCP6004-I/P	1	05_SPI_ADC_Analog	Quad op-amp for low-rate analog buffering/simulation.
ICS-314-T 14-pin sockets	2	05_SPI_ADC_Analog	MCP6004 socket plus spare.
LMV393IDR	3	06_Fault_Logic	Comparator threshold detectors.
CD74HC221E	2	06_Fault_Logic	Monostable max-window / clamp test fixture.
SN74HC573AN	1	06_Fault_Logic	Fault latch.
ICS-320-T 20-pin sockets	2	06_Fault_Logic	SN74HC573 socket plus spare.
1N4148 diodes	many	06/08	Small-signal steering/isolation.
1 k resistors	10+	04/various	LED current limits and small series resistors.
4.7 k resistors	10+	02/various	I2C pullups and logic pulls.
100 k resistors	10+	various	Weak pulls, bias, safe defaults.
100 nF ceramics	30	all IC sheets	Local IC decoupling.
10 uF electrolytics	several	08_Power_Protection	Module/branch bulk caps.
100 uF electrolytics	several	08_Power_Protection	Board-level logic rail bulk.
SMBJ5.0A	3	08_Power_Protection	5 V logic rail transient clamp placeholder.
TPD4E05U06DQAR	10	08_Power_Protection	4-line ESD arrays for connector entries; PCB/fine adapter later.
Perfboards	kit	physical build	Modular soldered RevA boards after breadboard proof.
Flux pen / braid / solder	tools	assembly	Soldering and rework.

14. KiCad 6 practical commands and symbol search terms

This section is the no-drama KiCad workflow while you are staring at the schematic editor. Exact library names can vary by KiCad install, but these search terms and object types are stable.

Common KiCad 6 actions

Need	KiCad action	What to choose
Place a symbol	Press A or use Place Symbol	Search by part family: MCP23017, MCP3208, ULN2803, 74HC573, 74HC221, LMV393, MCP6004.
Place a resistor/cap/diode	Press A	Device:R, Device:C, Device:CP, Device:D, Device:LED.
Place power	Press P	+5V or create/use +5V_LOGIC label, GND, PWR_FLAG where ERC requires.
Name a wire on one sheet	Place local label	Use only inside a sheet for short local runs.

Need	KiCad action	What to choose
Name a wire across sheets	Place global label	Use for I2C_SDA, SPI_MISO, FAULT_SUM, +5V_LOGIC, etc.
Make a sheet	Place Hierarchical Sheet	Use the numbered sheet names from this workbook.
Tie off unused IC pins	Place No Connect flag	Only after verifying the pin is truly unused.
Check schematic	Inspect -> Electrical Rules Checker	Fix real errors; document intentional warnings with notes/no-connects.
Assign footprints	Tools -> Assign Footprints	Use DIP footprints for DIP ICs, header/socket footprints for modules.

Useful symbol/footprint search terms

Circuit object	Symbol search	Footprint search / note
Nano Every module	Arduino_Nano, Conn_01x15 x2, or custom module	PinSocket_1x15_P2.54mm / Socket_Strip_1x15. Verify spacing against Nano.
MCP23017	MCP23017	Package_DIP:DIP-28_W7.62mm or equivalent.
MCP3208	MCP3208	Package_DIP:DIP-16_W7.62mm.
ULN2803A	ULN2803	Package_DIP:DIP-18_W7.62mm.
SN74HC573	74HC573, 74573	Package_DIP:DIP-20_W7.62mm.
CD74HC221	74HC221, 74221	Package_DIP:DIP-16_W7.62mm. Verify exact symbol pins.
MCP6004	MCP6004	Package_DIP:DIP-14_W7.62mm.
LMV393IDR	LMV393, LM393 compatible symbol	SOIC-8 on adapter for build; schematic may show comparator symbol.
REF5025AID	REF5025 or voltage reference symbol	SOIC-8 on adapter for build.
FRAM breakout	MOD_Adafruit_FRAM_1895_I2C	NanoArc_FRAM:Adafruit_FRAM_1895_I2C_ModuleOutline.
Tactile switch	SW_Push	Button_Switch_THT:SW_PUSH_6mm or matching TL1105 footprint if making PCB.
LED	LED	LED_THT:LED_D3.0mm.
Buzzer	Buzzer, Speaker, Device:Buzzer	PinHeader or buzzer THT footprint; verify pitch.
Test point	TestPoint	TestPoint_THTPad or PinHeader_1x01.

KiCad library warning: a symbol name matching the part number does not guarantee the footprint or pin numbering is correct. Before wiring any physical IC, compare the KiCad symbol pins against the chip datasheet and the DIP notch orientation.

15. KiCad step-by-step build procedure

The goal is to produce a schematic that can generate a clean BOM-like part list, a wire checklist, and eventually a PCB/perfboard layout. Follow these steps in order.

Step 1 - Create libraries

- Create project symbol library lib/NanoArc_Modules.kicad_sym.
- Add the generated FRAM library to the project symbol libraries.
- Add NanoArc_FRAM.pretty to project footprint libraries with nickname NanoArc_FRAM.
- Do not use random internet symbols unless you verify pin order and footprint.

Step 2 - Root sheet

- Place hierarchical sheet blocks for 01 through 08.
- Add only labels/ports that really cross sheets.
- Put a note on the root sheet: RevA low-voltage ACE only; no energized output-stage wiring.

Step 3 - Nano carrier

- Place Nano module or two 1x15 connectors representing the Nano socket.
- Label all direct pins from the direct pin map.

- Place service UART header and power/test headers.
- Add 100 nF and local bulk cap placeholders near logic power entry.

Step 4 - I2C backbone

- Place I2C pullup resistors to +5V_LOGIC.
- Place FRAM breakout module symbol.
- Tie FRAM A0/A1/A2 low for 0x50 unless intentionally changed.
- Tie WP low or route it as FRAM_WP with a clear default.

Step 5 - GPIO expanders

- Place MCP23017 #1 on Panel Input 0x20 and strap A0/A1/A2 = 0/0/0.
- Place MCP23017 #2 on Panel Output 0x21 and strap A0/A1/A2 = 1/0/0 as physical pins A0 high, A1/A2 low.
- Place MCP23017 #3 on Machine I/O 0x22 and strap A0/A1/A2 = 0/1/0.
- Add decoupling to each expander and label reset/interrupt nets.

Step 6 - Panel

- Draw the key matrix as switches between row and column nets.
- Add encoder A/B/common and push switch.
- Draw talkback LEDs and ULN2803A channels.
- Add buzzer command/load representation, even if physical driver changes later.

Step 7 - ADC and fault logic

- Draw MCP3208 with SPI pins connected to Nano nets.
- Draw VREF_2V5 from REF5025 to MCP3208 VREF.
- Draw analog simulator headers/pots rather than final sensor scaling.
- Draw comparators, latch, and monostable as low-voltage logic/test fixture only.

Step 8 - ERC cleanup

ERC warning type	What to do
Unconnected pin	Either wire it, mark No Connect, or add a note explaining why.
Input pin not driven	Add pullup/pulldown, connector source, or PWR_FLAG if it is a power rail.
Power pin not driven	Add correct power symbol/PWR_FLAG only where appropriate.
Multiple output drivers	Check open-drain/open-collector nets and use correct symbol pin types if possible.
Hidden power pin confusion	Prefer explicit power pins or project symbols with visible power pins.

16. Breadboard translation and staged test checklist

Once the schematic exists, the breadboard is no longer a guess. Each test is a small, named subset of the schematic.

Build/test sequence

Checkpoint	Hardware present	Test result expected
1. Nano boot	Nano socket/carrier, USB serial	Serial boot message, no peripherals required.
2. Direct pin sketch	Jumpers/switches on direct pins	Firmware reports each input change over serial.
3. I2C scanner	I2C pullups only	No stuck bus; scanner runs.
4. FRAM	FRAM breakout	Scanner sees 0x50; write/read test passes.

Checkpoint	Hardware present	Test result expected
5. One MCP23017	Panel input expander at 0x20	Scanner sees 0x20; register test passes.
6. Key matrix	A few switches first, then full matrix	Single key reports correct row/column and label.
7. Encoder	PEC11R encoder	A/B direction and push switch reported correctly.
8. Output expander	0x21 and one LED	One talkback toggles under serial command.
9. ULN/LED bank	ULN2803A and all talkbacks	Lamp test toggles all eight LEDs.
10. ADC	MCP3208, VREF, one pot	ADC returns stable values; serial displays raw and scaled.
11. Fault logic	Low-voltage comparators/latch fixture	Simulated trip causes FAULT_SUM and latch state.
12. Machine I/O	0x22 loopback headers	Firmware can read/write placeholder lines.

Wire label format

Use the label maker or tape flags on jumper bundles. The labels should match KiCad net names exactly.

Good labels:

I2C_SDA
I2C_SCL
ADC_CS
TB_FAULT
KEY_ROW2
GND

Bad labels:

green wire
switch thing
pin 5 maybe

Appendix A. DIP pin maps

MCP23017 PDIP-28

1 GPB0	28 GPA7
2 GPB1	27 GPA6
3 GPB2	26 GPA5
4 GPB3	25 GPA4
5 GPB4	24 GPA3
6 GPB5	23 GPA2
7 GPB6	22 GPA1
8 GPB7	21 GPA0
9 VDD	20 INTA
10 VSS	19 INTB
11 NC	18 RESET
12 SCL	17 A2
13 SDA	16 A1
14 NC	15 A0

MCP3208 PDIP-16

1 CH0	16 VDD
2 CH1	15 VREF
3 CH2	14 AGND
4 CH3	13 CLK
5 CH4	12 DOUT
6 CH5	11 DIN
7 CH6	10 CS/SHDN
8 CH7	9 DGND

ULN2803A DIP-18

1 IN1	18 OUT1
2 IN2	17 OUT2
3 IN3	16 OUT3
4 IN4	15 OUT4
5 IN5	14 OUT5
6 IN6	13 OUT6
7 IN7	12 OUT7
8 IN8	11 OUT8
9 GND	10 COM

CD74HC221E DIP-16 - check datasheet before final wiring

The 221 family has timing/control pins where exact naming differs by symbol library. In KiCad, choose the symbol matching CD74HC221E and verify each pin against the datasheet before soldering. Label the timing components clearly as a low-voltage test fixture.

SN74HC573AN DIP-20 - check library pin names

Use the standard 74HC573 latch symbol. Keep D inputs, Q outputs, LE, and OE visible. Add explicit pull/default states where firmware expects a known latch state.

Appendix B. Net dictionary

Net	Meaning / owner
+5V_LOGIC	Main RevA low-voltage logic rail.
GND	Low-voltage controller ground.
VREF_2V5	Precision ADC reference generated by REF5025.
I2C_SDA/I2C_SCL	Shared I2C bus for FRAM and three MCP23017s.
EXPANDER_INT	Optional shared interrupt from MCP23017 devices to Nano A3.
SPI_MOSI/SPI_MISO/SPI_SCK	SPI bus for MCP3208 ADC.
ADC_CS	Chip select for MCP3208.
FAULT_SUM	Aggregate fault line to Nano D2.
TRIP_OV	Over-voltage trip/simulator input to Nano D3.
TRIP_OC	Over-current/current-trip simulator input to Nano A7.
ARM_SW	Physical arm request/state input to Nano D4.
HARD_SAFE_SW	Hard-safe/abort/inhibit input to Nano D5.
GRIP_SW	Grip/handpiece operator-intent input to Nano D6.
TRIGGER_SW	Trigger request input to Nano D7.
OUTPUT_FB	Low-voltage output feedback/simulator to Nano D8.
PULSE_CMD	Low-voltage command output from Nano D9.
HW_INHIBIT	Hardware inhibit state input to Nano A0.
CHARGE_ENABLE	Low-voltage command/simulator output from Nano A1.
ENCLOSURE_INTLK	Interlock/simulator input to Nano A2.
HANDPIECE_PRESENT	Presence/simulator input to Nano A6.
KEY_ROW0..3	Panel keypad rows.
KEY_COL0..4	Panel keypad columns.
ENC_A/ENC_B/ENC_SW	Rotary encoder phases and switch.
TB_SAFE..TB_OPR_ERR	Panel talkback command nets.
BUZZER_CMD	Panel buzzer command net.
AIN0_VBANK..AIN7_THAND	ADC analog input simulator/telemetry nets.
CHARGER_READY/CHARGER_FAULT	Low-voltage machine status placeholders.
BLEED_CONFIRM/BLEED_ENABLE	Low-voltage machine status/command placeholders.
FAN_TACH_OR_GOOD/FAN_ENABLE	Fan status/command placeholders.
SERVICE_KEY	Service mode input.
POWER_MODULE_PRESENT	Low-voltage module-present placeholder.
FAULT_LATCH_VALID/FAULT_LATCH_RESET	Fault latch handshake nets.

Appendix C. Symbol and footprint notes

Breakout boards

Breakout boards are represented as module symbols. Do not draw the internal breakout circuit unless NanoArc is replacing the breakout with an onboard bare chip in a later revision.

SOIC-to-DIP adapters

For REF5025AID and LMV393IDR, the schematic can show the actual chip symbol, but the footprint/build note should say SOIC-8 on DIP-8 adapter for RevA. If you want maximum clarity, add an adapter connector symbol beside it or use a custom module symbol that includes the adapter.

DIP sockets

The socket is usually not a separate schematic symbol unless you are documenting mechanical assembly. In the BOM fields, add a note such as Socket: ICS-316-T. On a future PCB, the IC footprint is still a DIP footprint; the socket is an assembly part.

Perfboard-first design

Because RevA is a bench prototype, the first KiCad schematic should prioritize readable net names, test points, and module connectors over dense layout. A correct schematic makes a breadboard/perfboard build possible; a pretty board layout can wait.

Final pre-wiring checklist

Question	Pass condition
Does every cross-sheet wire have a global label?	Yes; no mystery local wires crossing pages.
Does every module have +5V_LOGIC and GND visible?	Yes; no hidden rail confusion.
Does every IC have 100 nF decoupling placed in the schematic?	Yes.
Do all I2C devices have unique address straps?	Yes: FRAM 0x50, expanders 0x20/0x21/0x22.
Are the MCP23017 RESET pins defined?	Yes; tied high or routed with pullup/test point.
Are the OLED nets deferred rather than deleted?	Yes; OLED_RST placeholder exists, display can be serial-stubbed.
Are raw high-current/output-stage conductors absent?	Yes; only low-voltage simulator/conditioned nets appear.