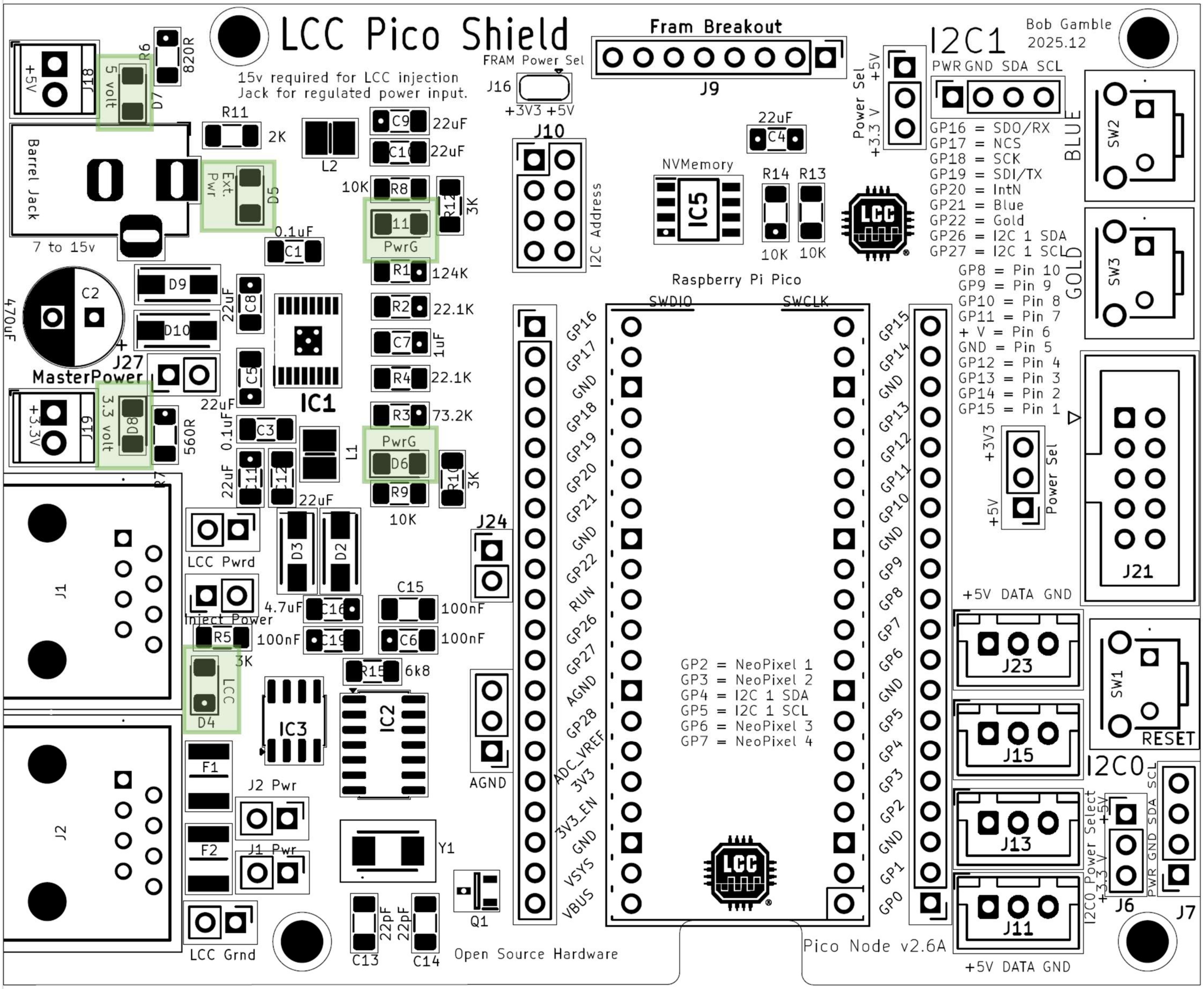
LCC Pico Node User Guide

Version 2.6

This version utilizes the MCP2815 and MCP2562 discrete hardware components (chips) for the LCC interface.

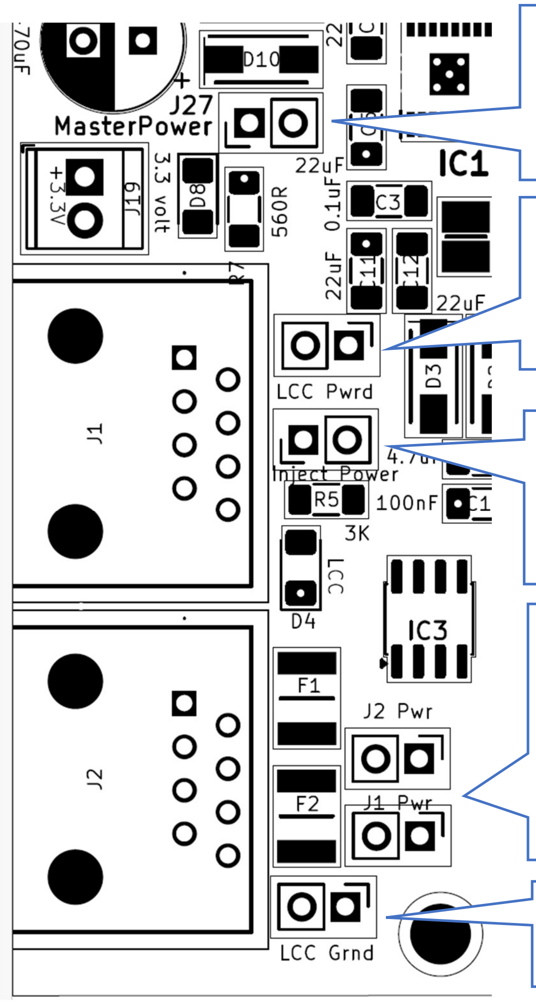
The intent of this node is to control up to four strings of NeoPixels and/or other peripheral components on a model railroad layout. It may also be adapted to several other uses as there is facility to connect to all the exposed pins on the Raspberry Pi Pico microcontroller. J11, J12, J15, and J23 have power and ground pins in line to facilitate NeoPixel string connections. While it will work with either the Pico or Pico2, the latter is recommended due to enhanced capabilities.

There are no resistors on the Reset, Blue, and Gold button lines.



The node may be powered from the barrel jack or the LCC port. The barrel jack power should be at least seven (7) volts to power the microcontroller or fifteen (15) volts [or your preferred LCC voltage] if using the node to inject power to LCC. Amperage should be sufficient to drive any external devices plus about half an amp for the node plus another 1.5 amps if injecting power to LCC. There are six power LEDs. Five should be lit to indicate that the node is properly powered, six if barrel jack powered.

Node power configuration is accomplished with five jumpers located behind the LCC ports.



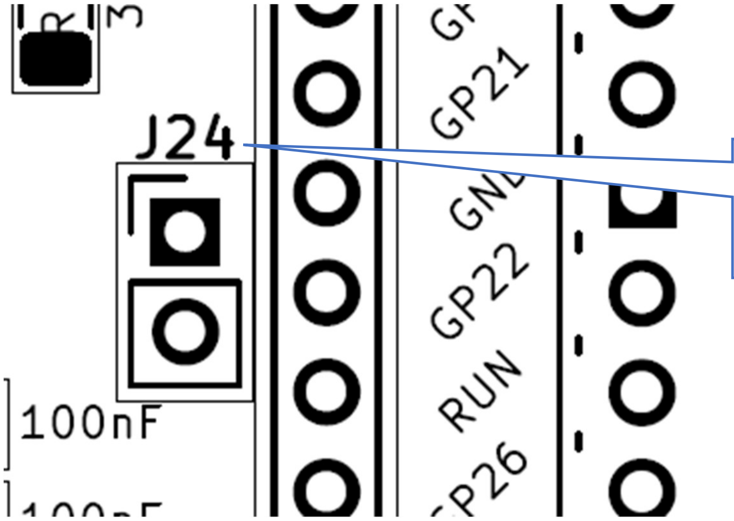
Master Power jumper should be bridged to power the node. A normally closed button may be plugged in here as a means to do a hardware reset should there be a problem with the power initialization due to a slow ramping power supply.

LCC Pwrd jumper should be bridged along with the LCC Grnd jumper if the node is to be powered from the LCC Port. Note that this will not provide power to the stepper driver which is powered exclusively from the barrel jack.

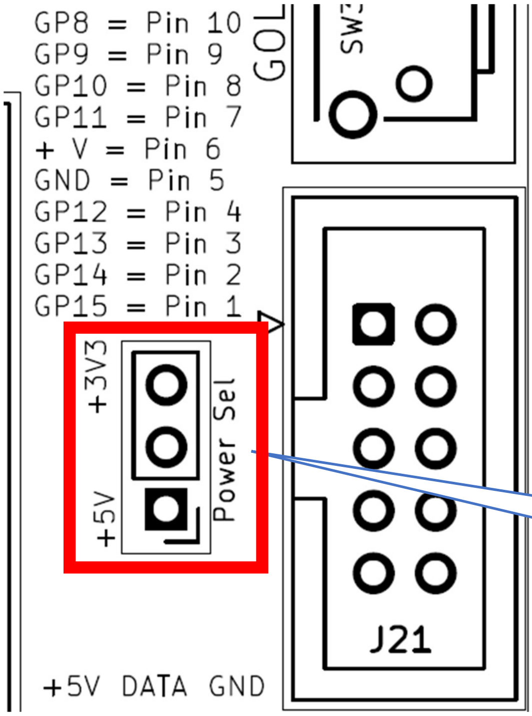
Inject Power should be bridged along with the LCC Grnd jumper if the node is intended to inject power to one or both of the LCC ports. 15 volts of at least 1.5 amps should be applied to the barrel jack in order to inject power.

J1 Pwr and J2 Pwr jumpers allow power isolation of one or both LCC ports. Generally, both jumpers would be bridged. If power injection or utilization was only desired in one direction, then the opposing jumper would be opened to isolate that leg. The isolated leg should be powered downstream as opening the jumper will cut off any power from the node.

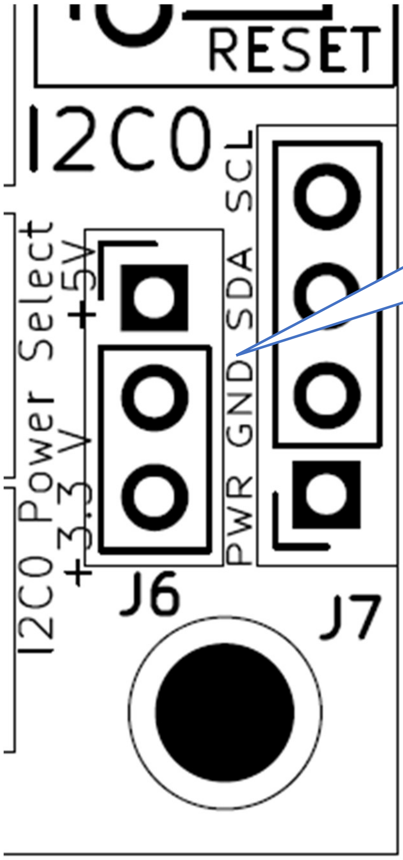
LCC Grnd jumper should be bridged if the node is intended to inject power to or draw power from one or both of the LCC ports.



J24 jumper may be attached to a normally open button to function as a Reset as it is in parallel to the on board Reset button.



The power applied to the middle pins of the IDC connector may be selected by bridging the respective power selection jumpers. Pin assignments are noted on the PCB.



The power applied to the I2C may be selected by bridging the power selection jumper.

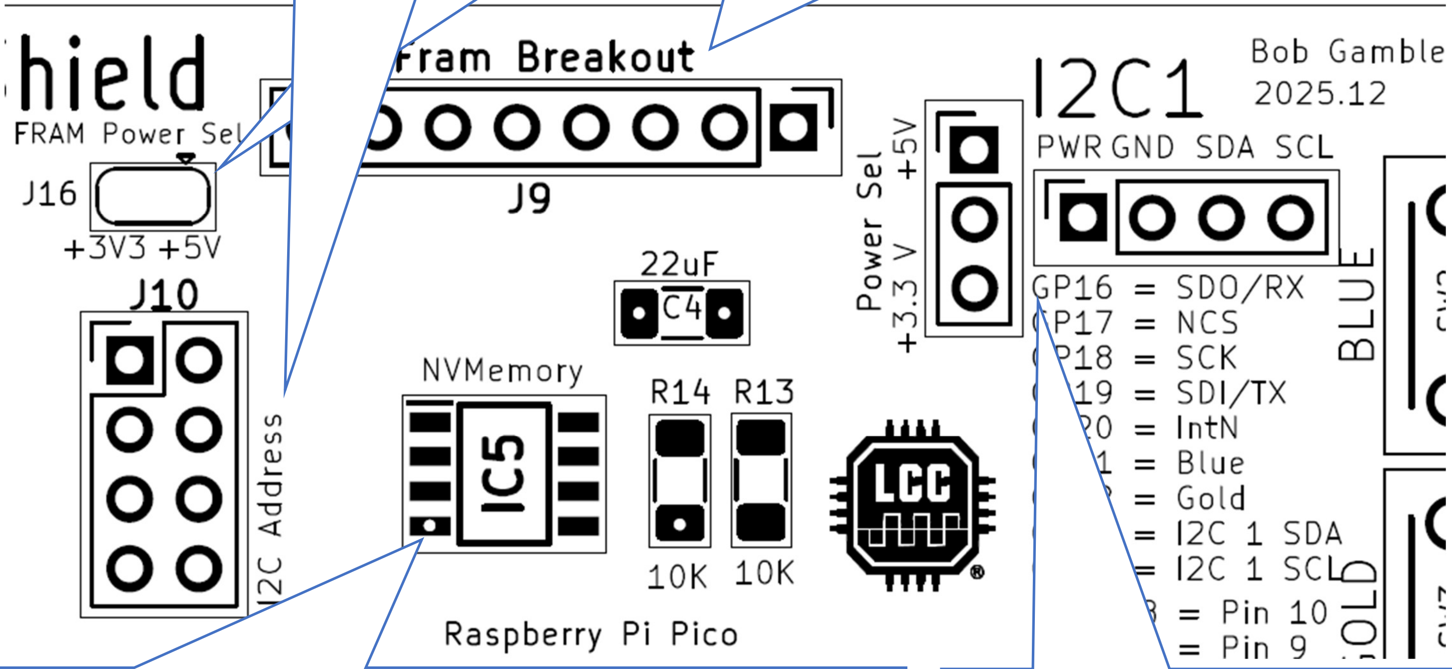
A similar jumper is adjacent to the I2C1 header.

NVM Storage

Either 3.3V or 5V may be selected by these solder jumpers. The selection will apply to the soldered NVM module or FRAM Breakout header

I2C address may be selected by the lower three pairs of headers. The selection will apply to the soldered NVM module or FRAM Breakout header

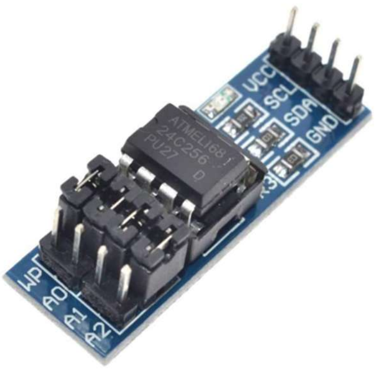
An Adafruit FRAM breakout board may be attached to this header.



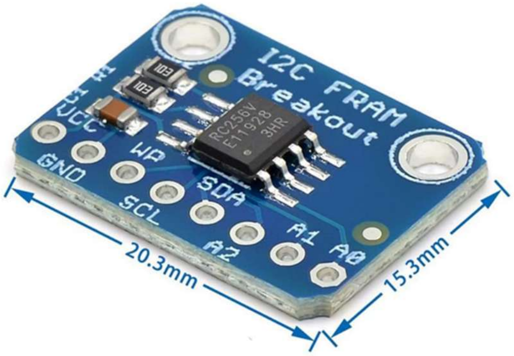
I2C nonvolatile memory may be populated if desired to reduce strain on the Pico flash. Parts compatible with these should work.

BR24G512F-3AGTE2	I2C BUS(2-Wre) 512K
MB85RC256VPF-G-BCERE1	256kbit FRAM with I2C

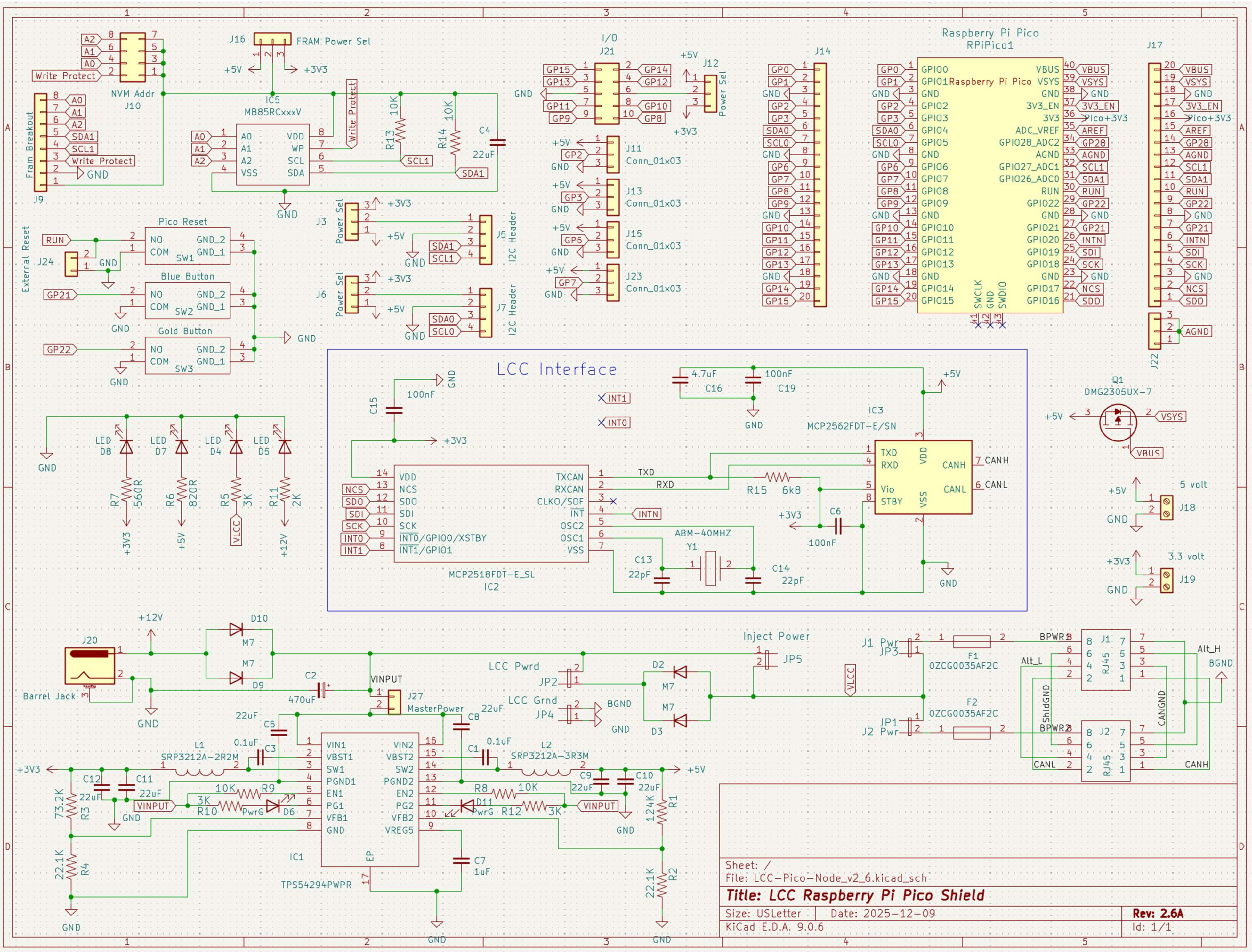
I2C modules including NVM may be attached to this header. Either 3.3V or 5V may be selected by the adjacent jumpers.



External NVM may be attached to either the I2C1 header, the FRAM Breakout header, or soldered down to the IC5 footprint. If either of the latter two choices are used, the I2C Address jumpers may be used to select non-default addresses or implement write protection. Otherwise, these would be functions of the module plugged into the I2C1 header.



A typical part would be a module utilizing an AT25C256 chip (left) or RC256V FRAM (right).



Sheet: /
File: LCC-Pico-Node_v2_6.kicad_sch

Title: LCC Raspberry Pi Pico Shield

Size: USLetter   Date: 2025-12-09   **Rev: 2.6A**
KiCad E.D.A. 9.0.6   Id: 1/1

Bill of Materials

Qty	Value	Reference	Mfr. #	Description
5	0.1uF, 100nF	C1,C3,C6,C15,C19	C1206C104K3RAC7411	Multilayer Ceramic Capacitors MLCC - SMD/SMT 25V 0.1uF X7R 1206 10%
1	470uF	C2	REF1016471M035K	Aluminum Electrolytic Capacitors - Radial Leaded Radial Leaded Aluminum Electrolytic Capacitor
7	22uF	C4,C5,C8,C9,C10,C11,C12	CL31A226MOCLNNC	Multilayer Ceramic Capacitors MLCC - SMD/SMT 22uF+/-20% 16V X5R 3 1206
1	1uF	C7	CL31B105KAHNFNE	Multilayer Ceramic Capacitors MLCC - SMD/SMT 1uF+/-10% 25V X7R 32 1206
2	22pF	C13,C14	C1206C220J3HACTU	Multilayer Ceramic Capacitors MLCC - SMD/SMT 25V 22pF X8R 1206 5%
1	4.7uF	C16	CL31B475KOHNFNE	Multilayer Ceramic Capacitors MLCC - SMD/SMT 4.7uF+/-10% 16V X7R 1206
4	M7	D2,D3,D9,D10	M7	Rectifiers Diode, SMA, 1000V, 1A
2	0ZCG0035AF2C	F1,F2	0ZCG0035AF2C	Resettable Fuses - PPTC
1	TPS54294PWPR	IC1	TPS54294PWPR	Switching Voltage Regulators 4.5-18Vin Dual 2A Ou t Sync SD Cnvrtr A A 595-TPS54294PWP
1	MCP2518FDT-E_SL	IC2	MCP2518FDT-E/SL	CAN Interface IC Stand-alone Low Power CAN FD Controller w/SPI Interface Grade1
1	MCP2562FDT-E/SN	IC3	MCP2562FDT-E/SN	CAN Interface IC CAN Flexible Data Rate Transceiver
1	MB85RCxxxV	IC5	BR24G512F-3AGTE2	EEPROM I2C BUS(2-Wre) 512K SOP8 EEPROM
2	RJ45	J1,J2	MTJ-880X1	Modular Connectors / Ethernet Connectors 8P8C TYPE 0 GLD FLSH BLK
1	SRP3212A-2R2M	L1	SRP3212A-2R2M	Power Inductors - SMD Ind,3.2x2.5x1mm,2.2uH+/-20%,4A,shd AEC-Q200
1	SRP3212A-3R3M	L2	SRP3212A-3R3M	Power Inductors - SMD Ind,3.2x2.5x1mm,3.3uH+/-20%,3.1A,shd AEC-Q200
1	DMG2305UX-7	Q1	DMG2305UX-7	MOSFETs P-Ch ENH FET -20V 52mOh -5A
1	124K	R1	ERA-8AEB1243V	Thin Film Resistors - SMD 1206 124Kohm 25ppm 0.1% AEC-Q200
2	22.1K	R2,R4	ERA-8AEB2212V	Thin Film Resistors - SMD 1206 22.1Kohm 25ppm 0.1% AEC-Q200
1	73.2K	R3	ERA-8AEB7322V	Thin Film Resistors - SMD 1206 73.2Kohm 25ppm 0.1% AEC-Q200
3	3K	R5,R10,R12		
1	820R	R6	RT1206FRE07820RL	Thin Film Resistors - SMD 1/4W 820 ohm 1% 50ppm
1	560R	R7	RT1206FRE07560RL	Thin Film Resistors - SMD 560 OHM 1% 50PPM 1/4W
4	10K	R8,R9,R13,R14		
1	2K	R11	RT1206FRE072KL	Thin Film Resistors - SMD 1/4W 2K ohm 1% 50ppm
1	6k8	R15	RT1206DRD076K8L	Thin Film Resistors - SMD 6.8 kOhms 250 mW 0.5 % 1206
1	Raspberry Pi Pico	RPiPico1		
3	Pico Reset	SW1,2,3	TS11-674-43-BK-160-RA-D	Tactile Switches 6 x 7.4 mm 4.3 mm 160 gf Black Right Angle Through Hole SPST
1	ABM-40MHZ	Y1	ABM3-40.000MHZ-B2-T	Crystals Xtal 5032 2-SMD 40MHz Tol +/-20ppm Stab +/-50ppm -20C - 70C 18pF 25 Ohms
4	LED	D4,D5,D7,D8	156120VS75000	Single Color LEDs WL-SMRW SMD RvMt Mon Wtrclr Rect 1206BtGn
2	PwrG	D6,D11	156120VS75000	Single Color LEDs WL-SMRW SMD RvMt Mon Wtrclr Rect 1206BtGn

Connections:

J3,J6,J12	3.3/5 Volt Power Selection
J5,J7	I2C Header
J9	Fram Breakout header
J10	NVM Address
J11,J13,J15,J23	NeoPixel Headers
J14,J17	Pico Pinout Conn_01x20
J16	FRAM Power Select
J18	5 volt power out
J19	3.3 volt power out
J20	Barrel Jack power input

J21	10 pin IDC connector I/O
J22	AGND headers
J24	External Reset header
J27	MasterPower bridge header
JP1	J2 LCC Power enable
JP2	Power from LCC enable
JP3	J1 LCC Power enable
JP4	LCC Ground enable
JP5	Inject Power to LCC enable

Software

Board:

The author uses the RP2040 board library by Earle Philhower

https://arduino-pico.readthedocs.io/en/latest/install.html

For LCC support:

https://github.com/openlcb/OpenLCB_Single_Thread

Or

https://github.com/JimKueneman/OpenLcbCLib

CAN Hardware:

ACAN2517 Arduino library.

External Storage:

I2C_EEPROM Arduino library by Rob Tillaart

FRAM_I2C Arduino library by Rob Tillaart

NeoPixel:

NeoPixelBus by Makuna Arduino library (preferred)

Or

NeoPixelConnect Arduino library by Alan Yorinks.

Other preferred libraries for the RPi Pico:

Display

```
#include <TFT_eSPI.h>      // Hardware-specific display library

#include "my_bb_captouch.h"    //capacitive touch driver
```

Servos

```
#include <ServoEasing.hpp>  // great library for getting slow servo action, including bounces

#include <PCA9685_servo_driver.h>

#include <PCA9685_servo.h>
```

Stepper Motor

```
#include "AccelStepper.h"         // by Mike McCauley
```