

BASIC USER & CODING MANUAL

Vetus Pars

How to set up, run chip tests, and write practical test programs

Plain-language edition

Based on the current project firmware

Manual revision: 1.0 · 5 September 2026

PURPOSE: This edition is for everyday users and people learning to create tests. It leaves out most internal firmware and repair-level details, which are documented in the companion Firmware Manual.

Quick start

1. Make sure no test is running. Wait until the previous test has completely ended and the result or menu screen is displayed.
2. Choose the correct chip from Chip Select, or use Type Select to narrow the list.
3. Check the chip name, pin count, orientation, and socket position. Pin 1 is always at the top of the ZIF socket.
4. Insert the chip and lock the ZIF socket.
5. The selected chip information appears immediately. Press Button 2 (OK) to test, or Button 1 (Back) to leave.
6. Follow any messages on the display. Button 1 is Back, Button 2 is OK, and Button 3 is Down.
7. Read the PASSED or FAILED result and wait for the test to completely end.
8. When the socket pins have returned to high impedance, unlock the socket and remove the chip.

SAFETY: Never insert, remove, or rotate a chip while a test is running. When a test completely ends, Vetus Pars removes power from the test board and returns all socket pins to high impedance. A wrong chip selection or wrong orientation can drive signals into the wrong pins during a test.

What Vetus Pars does

Vetus Pars is a stand-alone chip tester. A saved chip profile tells it how many pins the chip has, which test program to run, and which messages to show. During a test it can set pins high or low, read pins, generate a clock, step through addresses, repeat sections of a test, and show progress or instructions.

- Run saved chip tests with or without a computer.
- Search and browse saved chips.
- Show chip information, test messages, images, and code.
- Load or run tests from VetusGUI over USB.
- Store chip definitions, test code, strings, images, fonts, and user settings.

Part 1 — User manual

1. Controls and screens

The three buttons along the front edge of the PCB are shown from left to right:

Button 1 (Back)	Button 2 (OK)	Button 3 (Down)
Returns to the previous screen.	Selects the highlighted option or starts the test.	Moves the highlight to the next option.

The current on-device menu is intentionally small:

- Welcome — identifies Vetus Pars and shows the firmware version during startup.
- Main Menu — contains Chip Select, Type Select, USB Mode, and Brd Scan.
- Chip Select — browse all saved chip profiles. Selecting a chip immediately shows its name, type, pin count, and package image.
- Type Select — opens the Search by Type screen, where the saved-chip list can be filtered by DRAM, SRAM, CPU, 7400 Series, I/O, or Other.
- USB Mode — allows VetusGUI to communicate with the tester.
- Brd Scan — reads the attached add-on board descriptor. It shows the board name when a valid add-on board is found; otherwise it reports “Basic board or no board connected.”

2. Setting up a chip

1. Find the exact part number printed on the chip.
2. In Vetus Pars, select the matching chip profile. Similar names are not always electrically compatible.
3. Confirm that the profile pin count matches the physical package.
4. With no test running, place the chip at the top of the ZIF socket. Pin 1 is always at the top of the socket.
5. Match the chip notch or pin-1 mark to the top of the ZIF socket, then close the lever gently. If it needs unusual force, stop and check alignment.

PIN 1: Pin 1 is always at the top of the ZIF socket. Never reverse the chip.

Mounting hardware

- Use the supplied vinyl standoffs to mount the display and the ZIF test board.
- Use the four supplied washers with the four larger mounting holes in the display assembly.
- Do not overtighten the mounting hardware or bend either circuit board.

Ribbon cables

Disconnect USB and the 5 V supply before installing or reseating either ribbon cable. The display and test board are intentionally shown upside down in these illustrations so you can see the connectors clearly. In both pictures, the blue reinforcement strip at each end of the cable faces upward toward you.

Display cable

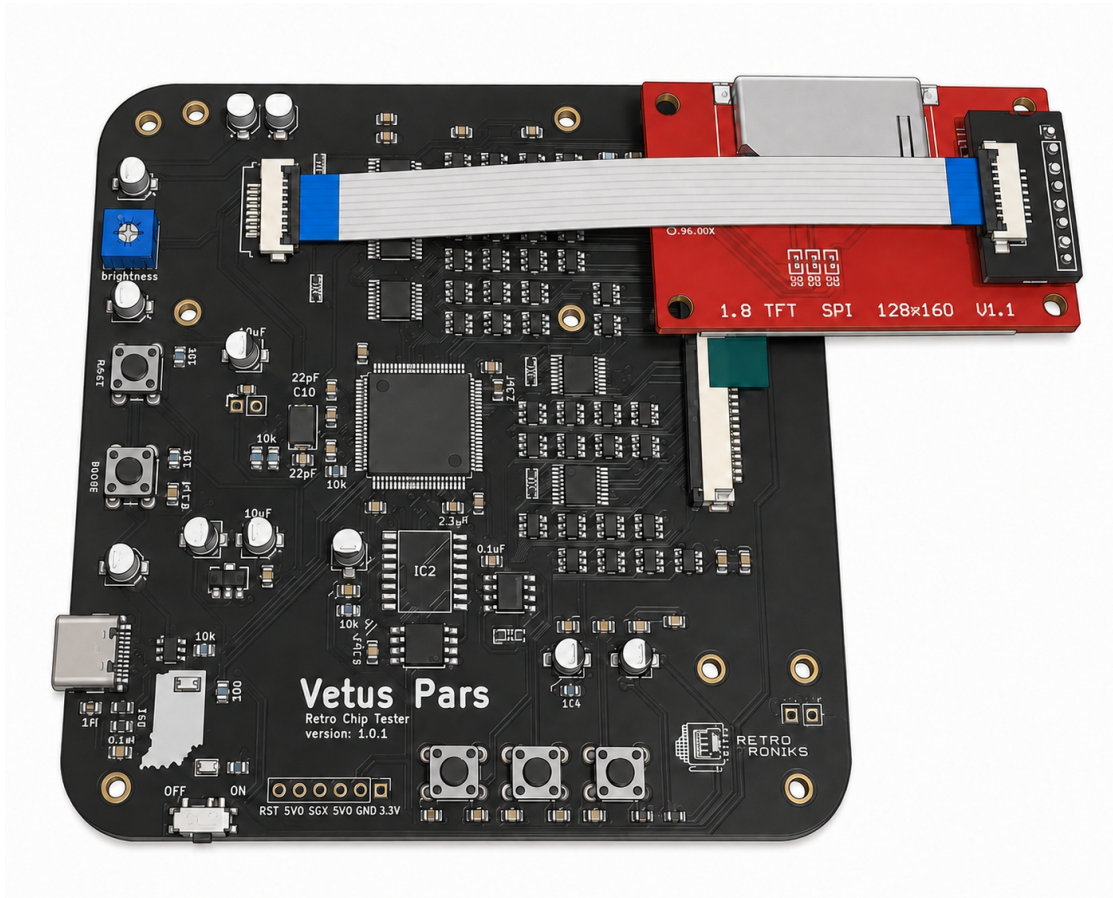


Figure 1. Upside-down installation view of the display cable. Keep both blue reinforcement strips facing upward as shown.

1. Open both connector latches gently.
2. Insert each end straight and completely, with the blue strip facing upward as shown.
3. Make sure the cable is not crooked and both ends are inserted evenly.
4. Close both latches and gently check that the cable cannot slide out.

ZIF test board cable

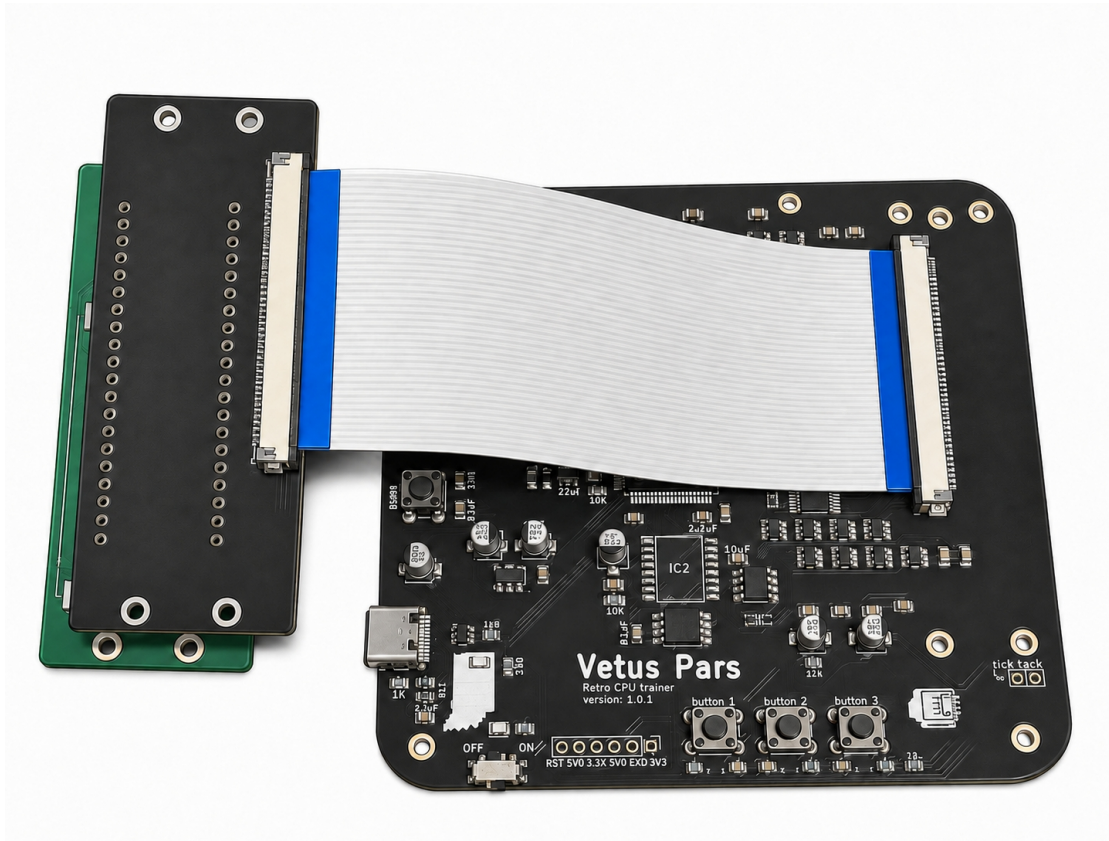


Figure 2. Upside-down installation view of the wide cable between the main board and ZIF test board. Keep both blue reinforcement strips facing upward as shown.

1. Open the wide-cable latch on each board.
2. Route the cable in a broad curve as pictured; do not crease, sharply fold, or twist it.
3. Insert each end straight and completely, with the blue strips facing upward as shown.
4. Close both latches and check that both ends are seated evenly before turning the boards over and mounting them.

ORIENTATION: The display and ZIF test board are upside down only to show cable installation. Turn them into their normal mounted positions after both cables are locked in place.

3. Running a saved test

1. Open Chip Select, or use Type Select to open Search by Type and narrow the list.
2. Highlight the correct chip and press Button 2 (OK).
3. The chip information appears immediately. Verify the name, type, pin count, and package image.
4. Press Button 2 (OK) again to start the test, or press Button 1 (Back) to leave without testing.

5. Follow any display prompts. Button 1 is Back, Button 2 is OK, and Button 3 is Down.
6. Wait for the final PASSED or FAILED screen and for the test to completely end. Button 1 will also end the test and go back to chip info.
7. Once the test has ended, the cleanup routine returns all socket pins to their safe high-impedance state and discharges the lines. The chip may then be removed, or Button 1 (Back) may be pressed to return.

Understanding results

Result	Meaning
PASSED	The programmed checks completed without the test setting a failure result.
FAILED	The test program marked at least one check as failed.
No result / frozen test	The program may be waiting for a button, stuck in a loop, or missing its ending command. Power down safely and review the test code.
Unexpected display	Recheck the selected profile, chip orientation, and saved strings.

REMINDER: A PASS only means the selected test passed. It does not prove every possible function, timing limit, or analog characteristic of the chip.

Part 2 — Using VetusGUI with the PCB

4. Connecting VetusGUI

1. Connect the Vetus Pars PCB to your computer with USB.
2. Start VetusGUI.
3. On the Vetus Pars board, choose USB Mode from the Main Menu.
4. In VetusGUI, choose the Vetus serial port. Use Refresh if no Vetus port is listed.
5. Select Connect and confirm that the GUI shows the connected device and serial number.

USB MODE: Keep the PCB on USB Mode while VetusGUI is transferring chip files, images, fonts, temporary tests, or settings. Fonts may be added while not in USB mode in the case of full reset.

Chip files and the tester library

- Chip Files are files stored on your computer. Use Add Files, Load, Delete File, Clear, or New File to manage this list.
- Use Send to Vetus for a temporary test without permanently adding the chip.
- After sending to Vetus, use Save to Vetus to store the current chip definition, or Batch Save to Vetus for a group of files. Batch will auto save all the files in the list to Vetus Pars.
- Vetus Chips shows chip definitions already stored on the connected PCB. Use Refresh, Load Selected, or Delete Selected as needed.
- The chip editor contains Chip Name, Type, No. Pins, Image Index, test code, and eight chip strings. Save the local file after editing.

Testing from VetusGUI

1. Load or create the correct chip file and verify its name, type, pin count, image index, strings, and code.
2. Make sure the chip is oriented with pin 1 at the top of the ZIF socket and close the lever.
3. Choose Send to Vetus to transfer the temporary test.
4. Use Button 2 (OK) on the PCB when prompted to run it, or Button 1 (Back) to leave.
5. Wait until the test completely ends before inserting or removing a chip.

Images and fonts

- Open Image Manager to create or load an image, preview or edit it, load an image slot from Vetus, save to a Vetus image slot, refresh slots, or delete an image. Images 1-14 cannot be deleted, they can however be edited and the slot saved to. All other slots can be saved to or deleted as desired.
- Open Font Manager to create or open a font, edit glyphs, preview it, and save it to a Vetus font slot. Saved fonts are not editable and new fonts must be added via files which can be made within Vetus Pars.
- After saving fonts, choose the active default font in Vetus Settings.

DO NOT INTERRUPT: Do not disconnect USB or power while saving, deleting, erasing, or updating chip data.

5. GUI settings and factory reset

Settings are changed in VetusGUI, not from the Vetus Pars board menus. Connect the tester, open USB Mode on the board, and use the GUI settings window.

- Show logo at startup — turn the startup logo on or off.
- Logo Speed — choose Slow, Medium, or Fast.
- Default Font — choose one of the fonts currently saved on Vetus Pars.
- AUX on — enable or disable the auxiliary output formerly called fan control.

- LEDs on — enable or disable the indicator LEDs.

Factory reset remains a board startup action. To invoke it, hold Button 2 and Button 3 while the unit starts. Use it only when you intentionally want to reset the unit. This deletes all fonts and images as well.

Part 3 — Basic coding manual

6. How a test program works

A test is a list of VetusGUI commands. The GUI validates each command and its required arguments, then converts it to the four 32-bit words used by the tester. Use the command names shown in VetusGUI exactly.

`COMMAND` required arguments

The tester runs one command at a time until ENDTEST. Some commands pause, repeat, jump, call a function, or wait for a button.

A safe program shape

```
ASAFE
PASSFAIL 1
//configure pins and run checks
//use PASSFAIL 0 when a check fails
ASAFE
ENDTEST
```

CODING STYLE: Use comments in your VetusGUI source, choose clear register purposes, keep loops short, and put cleanup commands on every normal exit path.

7. Pins: input, output, high, and low

Chip pin numbers are logical package numbers from 1 to the selected profile's pin count. The selected pin count is essential because it tells Vetus Pars how those numbers map onto the 40-pin socket.

GUI command	Use
ASAFE	Set all socket pins to high impedance.
SETPIN pin level	Set one pin low with 0 or high with 1.
RDPIN pin register	Read one pin into the numbered output register.
SETIN pin	Set one pin as an input.

GUI command	Use
SETOUT pin	Set one pin as an output.
PHIZ pin	Set one pin to high impedance.
TOGPIN pin direction	Toggle one pin.
MULTIO p1 p2 p3	Set three pins as outputs.
MULTII p1 p2 p3	Set three pins as inputs.
MULTIHI p1 p2 p3	Set three pins high.
MULTILO p1 p2 p3	Set three pins low.
ALLDIR sideA sideB	Set direction for both package sides using bitfields.
ALLEVEL sideA sideB	Set levels for both package sides using bitfields.

```
// Drive pin 2 high, then read pin 3 into register 0
SETPIN 2 0
SETOUT 2
SETIN 3
SETPIN 2 1
DELUS 10
RDPIN 3 0
```

SAFE ORDER: When possible, set the intended low/high value before changing a pin to output. Never let the tester and another device drive opposite levels onto the same wire.

8. Registers and values

VetusGUI provides registers 0 through 31. A destination argument uses the register number, such as 0. To use a register value as an input, write #R followed by its number, such as #R0.

GUI command	Use
REGL0D register value	Load a value into a register.
REGINC register	Add one.
REGDEC register	Subtract one.

GUI command	Use
ADD left right outputReg	Add two values.
SUBTR left right outputReg	Subtract two values.
MULT left right outputReg	Multiply two values.
DIV left right outputReg	Perform whole-number division.
RNG min max outputReg	Generate a value between min and max.
AND / OR / XOR	Perform a bitwise operation and store the result.
NOT value outputReg	Invert the bits.
SHL / SHR value amount outputReg	Shift a value left or right.

```

REGL0D 0 5
REGINC 0
ADD #R0 10 1

```

MATH SAFETY: Never divide by zero. Keep shift amounts within the valid width of the value.

9. Comparing values

IF, WHILE, and ENDUNTIL use a numeric operator. VetusGUI uses either number or symbols for the operator.

1 - NE	not equal !=
2 - LT	less than <
3 - GT	greater than >
4 - LE	less than or equal <=
5 - GE	greater than or equal >=
6 - DIV2	DIV2 even-value test

```

RDPIN 3 0
IF 0 #R0 1
SHOSTR 0 5 40
ENDIF

```

There is no ELSE command. Use a second IF with the opposite numeric operator when two outcomes are needed.

10. Repeating work and moving through a test

GUI command	Basic use
GOSTRT	Return to the start of the code.
LOOP count ... ENDLOOP	Repeat a block a fixed number of times.
BRKLOOP	Break the current loop.
WHILE operator value1 value2 ... ENDWHL	Repeat while the numbered comparison is true.
DUNTIL ... ENDUNTIL operator value1 value2	Repeat until the numbered comparison is true.
JMP line	Jump to an instruction line.
FUNC name ... ENDFUNC	Define a named function.
CALL name	Call a named function.

```

REGL0D 0 0
LOOP 8
REGINC 0
// perform one test step here
ENDLOOP

```

- Never use a LOOP count of zero.
- Keep every loop and conditional properly closed.
- Make sure ENDTEST remains reachable on every normal path.

11. Addresses, clocks, and arrays

Address groups

Address commands let one number control several pins as a bus. First map each address bit to a chip pin. Then write or read the bus.

```
DEFADDR 0 0x08070605 0x0C0B0A09
SETADDR 8 0x34 1
```

- DEFADDR defines up to eight packed chip pins for an address group.
- SETADDR writes an address; direction 1 means output and 0 means input.
- GETADDR reads the requested address bits into an output register.
- DEF8, SET8, and GET8 provide reusable 8-bit address groups.

Clock

CLKST starts a repeating clock on one logical pin at the requested kHz value. Use CLKSTP before cleanup.

```
CLKST 4 100
DELMS 10
CLKSTP
```

Arrays

Arrays A and B can each hold up to 32 values. They are useful for expected data, pin lists, or small lookup sequences.

GUI command	Basic use
ARRADD array index value	Add or place a value in array 0 or 1.
ARRGET array index outputReg	Copy an array item into a register.
PIN2ARR array lower4 upper4	Add packed chip-pin numbers to an array.

12. Timing, display, messages, and buttons

GUI command	Basic use
DELUS / DELMS duration	Pause for microseconds or milliseconds.
BTNWT button	Wait for Button 2, or 3. Button 1 will always leave the test
BTNPR button line	Jump to a line when the selected button is pressed.

GUI command	Basic use
FILSCR color	Fill the screen with an RGB565 color.
TXTCOL foreground background	Set text colors.
SHOSTR string x y	Draw a saved string.
PROG percent fill background	Draw a progress bar.
UDSCRN	Refresh the display.
LEDON / LEDOFF	Turn the test LED on or off.
BLINKON speed / BLINKOFF	Start or stop LED blinking.
PASSFAIL value	Use 1 for PASSED and 0 for FAILED. All tests start as failed unless specified.
ENDTEST	End the test, clear the loaded program, return every socket pin to its safe high-impedance state, discharge the lines, and show the Pass/Fail screen.

Saved strings

A chip profile can store up to eight short display strings. SHOSTR chooses the string number and its screen position. Strings can include printf-style number formatting and register markers.

```
// Saved string: Data = %02lX #R1
SHOSTR 0 5 40
UDSCRN
```

- Use #R0, #R1, and similar markers to choose register values.
- Keep strings short enough for the display and stored string slot.
- Use compatible numeric formats for the stored value.
- Use UDSCRN after drawing when an explicit refresh is needed.

Add-on board commands

GUI command	Basic use
BRD_ON / BRD_OFF	Turn add-on-board power on or off.

GUI command	Basic use
BRD_SEND_CMD command	Send a command to the board.
BRD_GET_RSLT outputReg	Wait for a board response.

13. Two complete examples

Example A — one output and one input

```

ASAFE
PASSFAIL 1
SETOUT 2
SETIN 3
SETPIN 2 1
DELUS 10
RDPIN 3 0
IF NE #R0 1
PASSFAIL 0
ENDIF
ENDTEST

```

This drives pin 2 high, checks whether pin 3 becomes high, and marks the test failed if it does not.

Example B — check both logic states

```

ASAFE
PASSFAIL 1
SETOUT 2
SETIN 3
RDPIN 3 0
IF NE #R0 0
PASSFAIL 0
ENDIF
SETPIN 2 1
DELUS 10
RDPIN 3 0
IF NE #R0 1
PASSFAIL 0
ENDIF
ENDTEST

```

This checks that the input follows both low and high. Real chips may need enable pins, power pins, longer settling time, or an inverted expected result.

14. Command quick reference

These 76 commands are shared by the current GUI compiler and firmware. SEVSEG is visible in the GUI panel but is not compilable, so it is excluded.

GUI category	Commands
Control Flow	GOSTRT, LOOP, ENDLOOP, BRKLOOP, WHILE, ENDWHL, JMP, DOUNTIL, ENDUNTIL, IF, ENDIF, FUNC, ENDFUNC, CALL, PASSFAIL, ENDTEST
GPIO	SETPIN, RDPIN, SETIN, SETOUT, MULTIO, MULTII, MULTIHI, MULTILO, ASAFE, ALLDIR, ALLEVEL, PHIZ, TOGPIN
Timing	DELUS, DELMS, CLKST, CLKSTP, BTNWT, BTNPR
Memory	DEFADDR, SETADDR, GETADDR, REGLOD, REGINC, REGDEC, ARRADD, PIN2ARR, ARRGET, DEF8, SET8, GET8
Math	ADD, SUBTR, MULT, DIV, RNG, AND, OR, XOR, NOT, SHL, SHR
Add-on Boards	BRD_ON, BRD_OFF, BRD_GET_ID, BRD_SEND_CMD, BRD_SEND_STR, BRD_WAIT_RESP, BRD_WAIT_STR, BRD_SEND_DATA, BRD_WAIT_DATA
Display	LEDON, LEDOFF, BLINKON, BLINKOFF, FILSCR, SHOSTR, TXTCOL, PROG, SHOWIMG, UDSCRN

Basic test checklist

- Correct chip profile and pin count
- Pin 1 at the top of the ZIF socket
- ASAFE at the start
- PASSFAIL initialized to 1 for a passing test
- Output levels chosen before output direction where possible
- Inputs configured before reading
- Numeric comparison operators used
- No loop count of zero
- Button numbers limited to 1, 2, or 3
- Failure conditions set PASSFAIL to 0
- CLKSTP used when a clock was started
- ENDTEST is required to automatically stop the test.

EDITION NOTE: This manual intentionally teaches the supported basics. Numeric opcode IDs, raw USB packets, storage addresses, circuit-level add-on-board details.

