Click to prove you're human

Normally, Arduino IDE offers to auto-update when a newer version is available. However, due to a bug, Arduino IDE 2.3.5 does not have an auto-update capability. Please download Arduino IDE 2.3.6 from the "Software" page of the Arduino website and install the downloaded file: This release provides fixes for regressions present in Arduino IDE 2.3.5. Thanks so much to the forum community for all the valuable testing and feedback that has helped us greatly in identifying and investigating these bugs! Highlights Fix application auto-update capability arduino/arduino-ide#2697, arduino/arduino-ide#2696 Arduino IDE has an auto-update feature, which notifies the user when a newer version is available, and allows them to easily update the application. The auto-update feature was broken in Arduino IDE 2.3.5. Arduino IDE 2.3.5 will not notify the user of an updated version, even if the user manually triggers an update check. This bug has been fixed in Arduino IDE 2.3.6, so the auto-update feature will work as before for users of Arduino IDE 2.3.6 once a future release comes out. However, the bug in Arduino IDE 2.3.5 means that it will not be possible to use the auto-update feature to update from 2.3.5 to 2.3.6. Please download Arduino IDE 2.3.6 from the links on the "Software" page of the Arduino website: Then install the new version from the downloaded file. Fix opening additional sketches via file association arduino/arduino-ide#2686, arduino/arduino-ide#2678 In addition to opening them via the Arduino IDE GUI, sketch files can be opened directly when the .ino file type is associated with Arduino IDE. Previously, if a sketch was opened in this manner while Arduino IDE was already running, a bug caused the currently opened sketch was opened again in a new window instead of the selected sketch. Fix extra blank window on macOS when IDE started via sketch file association arduino/arduino-ide#2693, arduino/arduino-ide#2688 In addition to starting the Arduino IDE application directly, it can be started by opening a specific sketch file using the Arduino IDE application. Previously, when Arduino IDE was started by opening a sketch file on a macOS machine, a pointless blank window was opened in addition to the Arduino IDE window for the selected sketch. Arduino IDE will now only open the window for the selected sketch. (Thanks @embeddedkiddie) Fix IDE stealing focus from other windows. arduino/arduino-ide#2681, arduino/arduino-ide#2679 During long operations such as installing large boards platforms or libraries, or compiling/uploading large sketch programs, the user may wish to multitask by using another application will the operation proceeds in Arduino IDE. Previously, if the user selected a different window while an operation was in progress in Arduino IDE, a bug caused the Arduino IDE window to steal focus from the active window each time any new content was printed to Arduino IDE's "Output" panel. Expand and improve translations arduino/arduino-ide#2692 The community work to make Arduino IDE accessible to everyone in the world through localization of the text of the IDE's UI continues, reaching high coverage levels for 18 languages/locales. If you would like to contribute to the Arduino IDE translations, please see the information here: Full changelog here: If you have any questions or feedback please post here in the dedicated forum category for Arduino IDE 2.x: If you want to see the list of known issues, work in progress, submit a formal report, or contribute to development, the Arduino IDE 2.x source code is hosted in this public repository: 8 Likes Downloaded 2.3.5 1,5 hrs ago 2 Likes ptillisch: Please download Arduino IDE 2.3.6 from the "Software" page of the Arduino website and install the downloaded file: Should IDE 2.3.5 be uninstalled first? If installing 2.3.6 over the old 2.3.5 installation, does it remember the previous installation preferences (e.g., whether to install for just the current user or all users, location of installation folder, etc.)? 1 Like Afaik binaries and data including preferences are separated. 1 Like grb: Should IDE 2.3.5 be uninstalled first? It is not required. The Windows installers (exe and "MSI") will perform an update if there is an existing installation of Arduino IDE. If you are using the Windows ZIP package, then you manage the installation manually and are free to have multiple copies of Arduino IDE if you like, and likewise can delete the copies as you like. There is no installer for Linux, so you manage the installations manually and can have multiple copies and remove them as you like, the same as the Windows ZIP package. macOS has a minimal installation helper, were you simply mount the DMG file and then drag the application from the mounted drive to the Applications folder (or anywhere you like). You are free to have multiple copies of the Arduino IDE application if you like. You only need to rename the Arduino IDE application folder if you want to store multiple copies in the same folder (since each folder must have a unique name). grb: does it remember the previous installation preferences (e.g., whether to install for just the current user or all users, location of installation folder, etc.)? Yes, the Windows installer does. 1 Like FYI, my MacAir does NOT have the Fix extra blank window on macOS when IDE started via sketch file association bug. I also do not have the following bug Fix opening additional sketches via file association sonofcy: my MacAir does NOT have the Fix extra blank window on macOS when IDE started via sketch file association bug. Are you using Arduino IDE 2.3.5 or 2.3.6? If you are using 2.3.6, then that is expected because the bug has been fixed in 2.3.6. If you are using 2.3.5, then the bug is indeed present. You can follow these instructions: If Arduino IDE is running, select Arduino IDE > Quit Arduino IDE from the Arduino IDE menus. All Arduino IDE windows will close. Control-click on any .ino file from a sketch. A context menu will open. Select "Open With" from the menu. A submenu will open that allows you to select which application to use to open the file. Select the Arduino IDE application from the menu. Arduino IDE starts, with the selected sketch open as expected. However, an additional blank Arduino window also opens. sonofcy: I also do not have the following bug Fix opening additional sketches via file association If you are using 2.3.5, then the bug is indeed present, but on macOS it only occurs if you pass the path of the sketch to the Arduino IDE invocation as a command line argument. For example: % "/Applications/Arduino IDE/Contents/MacOS/Arduino IDE" ~/Documents/Arduino/SomeSketch Maybe that explains it, I open sketches using normal methods like dbl clicking the ino file. I have no idea if that passes the path to the ino file to the command although it sounds probable. I have NEVER used Control-Click for anything in all the years I have been using a Mac. I just tried it on 2.3.5, and indeed, it works as the bug report states. Now, I have to find out what the advantage is of using a procedure to open a sketch that requires all those extra steps. Thanks for the clarification, and now, back to our regularly scheduled program. sonofcy: Maybe that explains it, I open sketches using normal methods like dbl clicking the ino file. A double click is generally equivalent to the Control-click procedure I described. Those instructions came from the bug report. I chose the Control-click procedure because it allows you to select the specific application to use to open the sketch, while the double click will open it using the default application. The developers and beta testers likely have multiple installations of Arduino IDE and they must make sure to open the sketch using the specific installation that is under test. sonofcy: I have no idea if that passes the path to the ino file to the command macOS seems to use a different mechanism. The reason I say this is because arduino/arduino-ide#2678 occurs when you open the sketch via command line argument, but not when you open it via double click or Control-click. Likewise, arduino/arduino-ide#2688 occurs when you open the sketch via double click or Control-click, but not when you open it via command line argument. sonofcy: Now, I have to find out what the advantage is of using a procedure to open a sketch that requires all those extra steps. The only advantage is the ability to select which application will be used to open it. If the application you want to use is used by default, then there is no advantage. 1 Like I have been using rename to allow for multiple versions/releases, but I have always used the Mac ecosystem method of Spotlight to select from a few versions of an app, fewer keystrokes I had a very old IDE in this Win 10 WS, so uninstalled it and grabbed 2.3.5. It was fascinating to see that during the first startup, the focus bug appeared. Tried to get a browser window into the fg, but the IDE kept took it back, until it was done with initial install, setup. Under 2.3.6 it's gone. Looks like IDE2.3.6 still having issues. Been working with zephyr on the giga R1 and had to do a remove of 0.3.1 and a reinstall. The reinstall completed successfully but after I closed and restarted the IDE on windows 11 22h2 received a not responding when I opened the IDE again. Had to restart machine. After restart it looked like it worked but when I put the board into booloader mode by double clicking the IDE went blank and had to do task manager to kill the process didn't see this issue previous versions. After restarting the ide seems to have recovered. Merlin513: Looks like IDE2.3.6 still having issues. Been working with zephyr on the giga R1 and had to do a remove of 0.3.1 and a reinstall Needless to say I have been in the same boat at times with zephyr on the GIGA. I have also installed 2.3.6 as 2.3.5 had many other issues. Sometimes with Zephyr, it feels like there is something with each individual GIGA on will it boot correctly or not. Not sure what. I have Two GIGAs, each now has a Giga display shield on it. With one of them, each time zephyr boots, I have to go to the Debug window (Serial1 through USB to UART connection) and type: sketch to get the sketch to run. Note: I have gone through probably 50+ loader installs since this started, plus I have tried building with or without debug on... The other one works as expected... Not sure what the difference is... Note: They both work fine when using MBED I have also tried to reset, the giga using the instructions at: Factory reset giga r1/remove micropython - Mega / GIGA R1 WiFi - Arduino Forum But maybe other things, like maybe the settings that it was set in earlier when using MBED, like flash split?... Sorry I know this is more giga specific and not the IDE, but maybe something like that is hitting your board as well, although maybe manifesting itself differently. There were other changes; not sure if it is in 2.3.4 → 2.3.5 or 2.3.5 → 2.3.6. One change that I observed is that when starting with a multi-line comment /* it now auto completes with the closing */. No complaints about it. Is this related to feat: use `theia@1.57.0` (#2654) · arduino/arduino-ide@859d29d · GitHub (2.3.4 → 2.3.5)? sterretje: Is this related to feat: use theia@1.57.0 (#2654) · arduino/arduino-ide@859d29d · GitHub (2.3.4 → 2.3.5)? I confirm that was where the feature was introduced. For those of us who prefer to close block comments manually, it can be disabled via Arduino IDE's advanced settings: Press the Ctrl+Shift+P keyboard shortcut (Command+Shift+P for macOS users) to open the "Command Palette". A menu will appear on the editor toolbar: Select the "Preferences: Open Settings (UI)" command from the menu. You can scroll down through the list of commands to find it, or type the command name in the field. A "Settings" tab will open in the Arduino IDE main panel. Type editor.autoClosingComments in the "Search Settings" field of the "Settings" tab. Select "never" from the "Editor: Auto Closing Comments" setting menu. Close the Preferences tab by clicking its X icon. 1 Like Two issues with 2.3.6: when initially offered to start the IDE immediately after the install, I got a black window. Library update hangs: DaveEvans: I got a black window. When that happened, did the IDE application become completely unresponsive as reported by @Merlin513?: Merlin513: the IDE went blank and had to do task manager to kill the process DaveEvans: Library update hangs I can provide you with instructions for a workaround: Start Arduino IDE. When you see the "Updates are available for some of your libraries" notification, click the "INSTALL MANUALLY" button instead of the "UPDATE ALL" button. Library Manager will open, with a list of the updatable libraries. Click the "UPDATE" button on the first of the entries in Library Manager for a library you wish to update. Wait for the update to finish, as indicated by the appearance of a "Successfully installed library ..." notification. Repeat steps (2)-(3) until you have updated all the libraries you wish to update. Please let me know whether the problem of the update hanging still occurs when you use that procedure. I also experience this problem sometimes. Unfortunately it seems to only occur intermittently and under specific conditions, which makes it difficult to investigate. For example, I experience it when I use the IDE with my ridiculous 32 GB collection of 90 different boards platforms installed, but when I switch to a more realistic environment with only a dozen platforms then I can no longer reproduce the fault. And when I enable the debug logging that would give me some idea of what is happening, I can no longer reproduce the fault. And when I run the same commands with Arduino CLI directly (which is the tool that actually performs the library updates for Arduino IDE under the hood), I can no longer reproduce the fault. A very difficult to pin down bug! I have made multiple attempts at investigating it, but previously I didn't take it very seriously because we had never received any reports from users who were affected. However, now we have your report, as well as a couple of others over the last six months (1, 2), so maybe something has changed that makes it more likely to occur? I find that when I update each of the libraries separately using the procedure I described in my previous reply, instead of all in one go as occurs when you click the "UPDATE ALL" button in the notification, I never encounter the update process hang. So I think that we at least have a reliable workaround for the bug. ptillisch: When that happened, did the IDE application become completely unresponsive as reported by @Merlin513?: Yes. Thank you for the library work-around. I'll give that a try the next time there are libraries to update. 1 Like ptillisch: Please download Arduino IDE 2.3.6 from the links on the "Software" page of the Arduino website: Software Then install the new version from the downloaded file. Hi, I did update like this from 2.3.5 and my all-previous sketches somehow disappeared from the sketch directory. I checked the sketch library location, but they are not there. The bug that previous versions had (Restart: "A setting has changed that requires a restart to take effect. Press the restart button to restart Arduino IDE and enable the setting" when a new sketch is opened, and next time when a sketch is opened menu bar is missing) is still there. I am on Windows 10 64bit. Page 2 3 posts were split to a new topic: Port select grey out bug A post was split to a new topic: IDE 2.x laborary/student setup ptillisch: And when I enable the debug logging that would give me some idea of what is happening, I can no longer reproduce the fault. That may mean (99% of cases) that multithreaded application (IDE) is accessing shared resources without any locking mechanism. At least from my software development experience. And this symphom is the same for i386 Win32 GUI application or PowerPC RTEMS Console-only app. Debug output changes timings so this bug becomes not reproducible. These bugs are extremely hard to find :(. DaveEvans: I'll give that a try the next time there are libraries to update. You can change library version manually, by editing library.properties file (located in your library folder): change version to the previous one and then Arduino IDE will offer you to update again ptillisch: I also experience this problem sometimes. Unfortunately it seems to only occur intermittently and under specific conditions, I experience freezes since 2.3.4 : either complete freeze (usually when IDE updates something in bulk , like toolchain or libraries) or partial freeze. By partial I mean that IDE works somehow (you can edit files and switch between them) but file content is not saved, with "Saving Icon" (a circle) appearing that never goes away. This happens when I right click on a macro to go to definition: after clicking nothing happens (or Loading... appears) and since that moment no source files can be saved. Happens quite often, somehow connected to file operations, but I can't always reproduce it :-/ vvb333007: complete freeze (usually when IDE updates something in bulk , like toolchain or libraries) Thanks for your report @vvb333007. When that occurs, does the window go blank as was described by @Merlin513 and @DaveEvans: Merlin513: the IDE went blank and had to do task manager to kill the process I believe the window being black is due to them having the IDE configured to use a dark theme, and it would instead be blank white if you instead have it configured for a light theme. vvb333007: Happens quite often, somehow connected to file operations, but I can't always reproduce it :-/ The lack of reliable reproducibility makes it challenging to investigate, prepare a fix, and validate the fix, but the "quite often" makes things a bit easier. Which operating system are you using? Windows 10, Light Theme, yes it was white blank. Do you have "debug" builds of Arduino IDE with all verbose output turned on? I'd like to install one so whenever it freezes I can copy IDE's debug output and send it to you. I had strong feeling that it is connected to multiple file operations performed in parralel. So even before you posted a workaround I somehow realized that doing things one by one will probably help. And yes it helped. Arduino IDE always automatically generates log files while it is running, so no special debug build is required. The next time you encounter this problem, please provide the latest log file from your computer. It might contain information that will help us to identify the problem. This procedure is not intended to solve the problem. The purpose is to gather more information. Please do this: Open a forum reply here by clicking the "Reply" button. Click the "Upload" icon () on the post composer toolbar: The "Open" dialog will open. In the "Open" dialog, select the latest .log file from this folder (the files are named according to the date):C:\Users\\AppData\Roaming\Arduino IDE\ (where is your Windows username) Note that the C:\Users\\AppData folder is hidden by default. On Windows "File Explorer", you can make it visible by opening the "View" menu, then checking the box next to "☐ Hidden items". Click the "Open" button. The dialog will close. Click the "Reply" button to publish the post. Alternatively, instead of using the "Upload" icon on the post composer toolbar as described in steps (5) - (7) above, you can simply drag and drop the .log file onto the post composer field to attach it. Please let me know if you have any questions or problems while following those instructions. 1 Like 2 posts were split to a new topic: IDE hangs on startup A post was merged into an existing topic: IDE hangs on startup Hello, after upgrading to version 2.3.3 I have a problem compiling my ESP32 module. I have a problem with the message: thread 'main' panicked at 'assertion failed: (left != right) left: 0, right: 0: Failed to get path name. Error code: 3', main.rs:65:9 note: run with RUST_BACKTRACE=1 environment variable to display a backtrace exit status 101 Compilation error: exit status 101 Does anyone know how to solve the problem? Thanks and have a nice day marosh555: after upgrading to version 2.3.3 From what? From IDE 1.x or from IDE 2.x (2.3.2?)? Was that the only upgrade or did you also upgrade the board package and/or libraries? Which operating system are you using. What you can try is to downgrade the ESP32 board package in case that was also upgraded. If that works, you can upgrade it again; if that upgrade fails it has something to do with the board package. Further I can't really help; here are some search results: esp32 thread 'main' panicked at 'assertion failed: (left != right) - Google Search Note: This might be IDE 2.x related (though not directly) and not hardware related and hence your topic has been moved to the IDE 2.x section of the forum. From version 2.3.2 on Windows 10. I have already tried installing the original version 2.3.2 but the result is the same. Yesterday when I was writing the program everything worked without problems and after the update this happened to me. The problem was solved when I downgraded the esp board. Thanks for help with my problem sterretje I had the same problem if you use Windows 10 like me. I solved it! After a long time of trying, I figured out that the account name and computer name must be without diacritics. After changing without diacritics, I installed the esp32 boards (from Espressif system) in the latest version 3.1.1 without any problems. 1 Like How did you rename the account name and computer name? The /users/YourName' folder can't be changed or am I wrong? I changed my username because I had the same problem (my name is José, I change it to Jose without the diacrytic accent) and it solves it. You have to be careful, doing that can damage your user sesion, so these are the steps you should follow. 1st. Be an administrator. 2nd. Create another user with administration privilegies. 3rd. Enter in that user sesison and change your user file name as you wish. 4th. Finally you have to change the register of the users on the computer, so open redegit and find this path: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\ there are the profiles path to each username, find yours and change the original one to the new path, for example: C:\Users\OldOne to C:\Users\NewOne in the profileImagePath that's is all. Then you have to reset the computer, and be able to enter your session. My recommendation is to create another user and just use the new user, doing the previus can damage your installation of some programs that path your username profile archives (I got the reinstall Arduino and repair some path issues because I did that, it's annoying but it's possible). That's it, be careful doing that seriously (I google it when I did it and it works but causes some issues as I said). Thanks for sharing the solution you found @juanjgomezb! I'll mention an alternative workaround for those who are encountering problems caused by characters in their Windows user folder name. This workaround might be preferrable to those who don't feel comfortable making the changes to the registry as is required to change the folder name. The alternative is to configure Arduino IDE to use different paths on your computer, which are not under the user folder (and that only contain basic ASCII characters): These instructions will only work for Arduino IDE versions 2.3.4 or newer. Select File > Quit from the Arduino IDE menus if it is running. All IDE windows will close. Open the file at the following path in any text editor:C:\Users\\.arduinoIDE\arduino-cli.yaml (where is your Windows username) Add the following content to the arduino-cli.yaml file (or replace the existing content if equivalent content is already present in the file):build_cache: path: directories: builtin: libraries: data: user: Replace the placeholder with the path to the folder under which you want Arduino to store the cached data for compiling sketches. For this, and for the paths in the subsequent instructions, make sure to chose a location that is not under any folders with names that contain anything other than basic ASCII characters (e.g., A-z, 0-9, _, -, .,). Arduino IDE periodically deletes contents of this folder (the files stored under it are only cached there for the sake of efficiency, and it can always regenerate the files on demand), so you should not use this folder to store any valuable files. Replace the placeholder with the path to the folder under which you want Arduino to store some fundamental "built-in" Arduino libraries. Replace the placeholder with the path to the folder under which you want Arduino to store the boards platforms you install via Boards Manager, as well as some other data files. Replace the placeholder with the path you want to use as the Arduino sketchbook (the folder under Arduino IDE stores libraries you install via Library Manager and "Add .ZIP Library...", as well as a convenient location to save your sketches). Unlike the other paths, you can also configure this one via Arduino IDE's preferences GUI. I thought I should include it in these instructions just to be comprehensive. Save the file. Move the files from this folder on your hard drive:C:\Users\\AppData\Local\Arduino15\libraries (where is your Windows username) to the folder you configured in step (5) above (""). Move the files from this folder on your hard drive:C:\Users\\AppData\Local\Arduino15 to the folder you configured in step (6) above (""). If you configured a different sketchbook folder location in step (7) above, move the files from the previous sketchbook folder location to the new one. Start Arduino IDE. I also found an alternative workaround, which is perhaps a bit risky so I don't necessarily recommend it, but maybe of interest to advanced users so I'll add a link to it: 1 Like A common response to “I want my code to do A., then B., then C., etc.,” from the knowledgeable is “Use a state machine.” Many who are new to coding/programming have never heard of a state machine, although almost everyone is familiar with the concept. There are different ways to realize a state machine, like the very similar Sequential Function Chart, a construct used in Programmable Logic Controllers, but this discussion will deal only with the C++ switch/case construct - which is described briefly on the Arduino reference page. There are also numerous examples on the web and elsewhere but I wanted to offer an Arduino-focused example to eliminate the distraction of translating a generic non-Arduino explanation to what is seen in the IDE. The example will be a simple garage door opener. Push a button and the door goes up and stops. Push again, the door goes down and stops. Right there are four ‘states’ – 1. Closed, waiting for open command; 2. Opening, waiting for top limit; 3. Open, waiting for down command; 4. Closing, waiting for bottom limit. Notice there are three elements for each state: A name, usually describing what the state does. In reality it's an alias for the underlying numeric value the switch statement evaluates to select a case. It can also be, and usually is, a target for another state to go to – the way Closed above goes to Opening. The example uses an enum to give the states their names. One or more actions that happen in this state, like driving an LED on or off or calling a function. Some condition or combination of conditions which, when met, advance us to the next state. The listing is pretty liberally commented so there won’t be a lot of discussion here. Also, to reduce clutter in the listing, there’s no switch debouncing included in this first effort. It’s not needed for the sketch to demonstrate the state machine principle. You can observe the effects of the default: label by uncommenting doorState = 6;, the last statement in setup(). Since only four names are given in the enum statement, and they are not given explicit values, the compiler assigns the values zero through three. If switchState is 6 on the first pass through the code it won’t match any of the cases given so, the default is selected, a message is displayed, and the state is then set to doorIsDown. Thereafter the sequence will proceed normally. It should be obvious from this that you can start the sequence at any point by loading an arbitrary, legal, value to doorState. Hardware: The examples use an UNO but are easily adapted to other processors if needed. The usual breadboard and hookup/jumper wires for setting up experiments. Two normally open pushbuttons for digital input – like the tactile switches that come with Arduino kits. Two LEDs with associated dropping resistors for visual indicator outputs. You can omit the LEDs if you’re satisfied with watching the action on the serial monitor. Construct the circuit and load and run the sketch – see the listing for pin numbers and schematic. Don’t worry about the second switch right now, it’ll be added later. Pressing and releasing the switch should cause the LEDs to alternately light for a time, signifying the motor running to open or close the door. A timer is used rather than a physical input to simplify the circuit - although it’s not uncommon for a real application to use a timer state as the condition to advance. As part of the attempt to simplify I made a small modification to the timer comparison. The usual Arduino timer uses the construction if(millis() – yourTimer >= yourPreset){ // Time is up, do something } To avoid having to think about subtraction I put this into a #define and gave it the name accumulatedMillis – #define accumulatedMillis millis() – yourTimer Now every time the compiler sees accumulatedMillis it will substitute millis() – yourTimer in its place so the comparison becomes if(accumulatedMillis >= yourPreset). Notice that the condition to get out of the closed state is the switch being operated. Once we’re in doorOpening the switch state isn’t checked since it doesn’t matter anymore. Likewise, when the door is either open or closed we don’t check the timer to see if it’s time to stop the motor. Remember, this is only an illustration, don’t build a real garage door opener based on this sketch! Real-world door openers have safety features built in which don’t exist in this demo. // // The sketch simulates a simple motor-driven door opener/closer // such as might be used on a garage, chicken coop, window blinds, // etc. Starting from a known position a switch is pressed, the // door opens, then stops. When the switch is pressed again the // door closes, then stops. // // A timer is used to simulate the delay while the door is being // driven to its new position and LEDs simulate the up/down // forward/reverse action of the the motor. // The sketch uses the following I/O const unsigned char switchInput = 10; // Arduino pin 10 --| |--SW-- GND const unsigned char openLED = 9; // +5V--/\\/\- 330Ω -->|-- Arduino pin 9 const unsigned char closeLED = 7; // +5V--/\\/\- 330Ω -->|-- Arduino pin 7 #define motorRun LOW #define motorStop HIGH #define accumulatedMillis millis() - timerMillis const unsigned long motorTimerPreset = 2000; // two seconds unsigned long timerMillis; // For counting time increments // The door has four possible states it can be in // Let's give the states descriptive names enum {doorIsDown, doorIsUp, doorOpening, doorClosing}; unsigned char doorState; // What the door is doing at any given moment. void setup() { Serial.begin(115200); pinMode(switchInput, INPUT_PULLUP); pinMode(openLED, OUTPUT); digitalWrite(openLED, HIGH); pinMode(closeLED, OUTPUT); digitalWrite(closeLED, HIGH); // doorState = 6; } void loop() { switch (doorState) { case doorIsDown: // Nothing happening, waiting for switchInput Serial.println("door down"); if (digitalRead(switchInput) == LOW) { // switchInput pressed timerMillis = millis(); // reset the timer doorState = doorOpening; // Advance to the next state break; } else { // Switch not pressed break; // State remains the same, continue with rest of the program } case doorOpening: Serial.println("door opening"); digitalWrite(openLED, motorRun); // // The compare below would be replaced by a test of a limit // switch, or other sensor, in a real application. if (accumulatedMillis >= motorTimerPreset) { // Door up digitalWrite(openLED, motorStop); // Stop the motor doorState = doorIsUp; // The door is now open break; } else { break; } case doorIsUp: Serial.println("door up"); if (digitalRead(switchInput) == LOW) { // switchInput pressed timerMillis = millis(); // reset the timer doorState = doorClosing; // Advance to the next state break; } else { // switchInput was not pressed break; } case doorClosing: Serial.println("door closing"); digitalWrite(closeLED, motorRun); // Down LED on if (accumulatedMillis >= motorTimerPreset) { digitalWrite(closeLED, motorStop); // Stop the motor doorState = doorIsDown; // Back to start point break; } else { break; } default: Serial.println(" We hit the default"); delay(3000); doorState = doorIsDown; break; } } With my BASIC language programmed controllers I can use AND and OR. example: IF (VAL > 100 AND VAL < 140) THEN ... How can I solve this with the if function in the Arduino? Thanks. 1 Like you need to add () 's and a lot of them , further Arduino knows 2 types of AND and OR's the logical and the bitwise. IF (VAL > 100 AND VAL < 140) THEN ... becomes if ((val > 100) && (val < 140)) { .... } See for more information 1 Like C++ Keyword Synonyms and && and_eq &= bitand & bitor | not ! not_eq != or || or_eq |= xor ^ xor_eq ^= EDIT: What the above means is that: IF (VAL > 100 AND VAL < 140) THEN ... can be written as: if (val > 100 and val < 140) { ... } 3 Likes See for more information Thanks for this link didn't now where to look. It works perfect now. Many thanks you need to add () 's and a lot of them Why? && and || are of a much lower precedence than any operator you would commonly use in a comparative expression. Overbracketing just leads to parenthesis-blindness in my opinion, though I concede that is partly a matter of style. Why? && and || are of a much lower precedence than any operator you would commonly use in a comparative expression True, but for those few times the precedence was higher it kept my debugging sessions short except the first time parenthesis-blindness Ever debugged LISP code? If you are using a recent version of the IDE, you will get an auto-update offer simply by starting the IDE. Otherwise, the release is available for download here: This release provides some nice advancements and fixes. Thanks so much to the forum community for all the valuable testing and feedback that has helped us greatly in identifying and investigating these bugs and enhancements! Deprecation notice: Upcoming cessation of support for Linux distros using glibc 2.28 Recent changes in the framework used to produce automated release of Arduino IDE resulted in the loss of compatibility of the Linux builds with older Linux distro versions that use version 2.28 of the GNU C Library (glibc) shared library. This includes Ubuntu 18.04. Arduino IDE 2.3.4 will be the last version that can be used with these distro versions. Future releases (including nightly and tester builds) will fail to start on machines using these distro versions with an error like: Error: node-loader: Error: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.28' not found (required by /home/foo/arduino-ide/resources/app/lib/backend/native/pty.node) at 85467 (/home/foo/arduino-ide/resources/app/lib/backend/main.js:2:2766) at __webpack_require__ (/home/foo/arduino-ide/resources/app/lib/backend/main.js:2:6663105) at 23571 (/home/foo/arduino-ide/resources/app/lib/backend/main.js:2:3374073) at __webpack_require__ (/home/foo/arduino-ide/resources/app/lib/backend/main.js:2:6663105) at 55444 (/home/foo/arduino-ide/resources/app/lib/backend/main.js:2:3369761) at __webpack_require__ (/home/foo/arduino-ide/resources/app/lib/backend/main.js:2:6663105) at 24290 (/home/foo/arduino-ide/resources/app/lib/backend/main.js:2:1780542) at __webpack_require__ (/home/foo/arduino-ide/resources/app/lib/backend/main.js:2:6663105) at 43416 (/home/foo/arduino-ide/resources/app/lib/backend/main.js:2:1770138) at __webpack_require__ (/home/foo/arduino-ide/resources/app/lib/backend/main.js:2:6663105) If you don't know which version of glibc your machine is using, run this from the terminal: /lib/x86_64-linux-gnu/libc.so.6 Further technical details are available here. The recommended path forward is to update to a modern version of Linux. If you can't or won't do that, you can update to the last compatible version of Arduino IDE, 2.3.4, and then avoid updating when future releases (e.g., 2.3.5) come out. There are two methods for dealing with these update offers. I'll provide instructions for both of them. You can pick whichever one of the two is most convenient for you. "SKIP VERSION" The "Update Available" dialog contains a "SKIP VERSION" button. If you click that button, Arduino IDE will no longer show the dialog for the specific newer version being offered at the time you click the button. The dialog will appear once again each time Arduino releases a new version of Arduino IDE, so you will need to click the button again after each release. That is slightly inconvenient, but the release cycle is fairly long so I don't think it will be very burdensome. Disable All Update Offers It is possible to completely disable the offers of updates in the Arduino IDE advanced settings. The downside is this setting also disables offers of updates for your installed boards platforms and libraries. Press the Ctrl+Shift+P keyboard shortcut (Command+Shift+P for macOS users) to open the "Command Palette". A menu will appear on the editor toolbar: Select the "Preferences: Open Settings (UI)" command from the menu. You can scroll down through the list of commands to find it or type the name in the field. A "Preferences" tab will open in the Arduino IDE main panel. Type arduino.checkForUpdates in the "Search Settings" field of the "Preferences" tab. Uncheck the box under the "Arduino: Check For Updates" setting. Close the Preferences tab by clicking its X icon. Keeping Your Arduino Dependencies Updated If you chose this "Disable All Update Offers" method, make sure to periodically check to see if newer versions of your installed boards platforms and libraries are available. You can do this by opening Boards Manager and then setting the "Type" menu to "Updatable", then repeating the process with Library Manager. There is no possibility of a problem using newer versions of Arduino libraries on a computer with an older version of Linux, so don't worry about an incompatibility when updating your libraries. It is theoretically possible that some future version of a boards platform could have an increased minimum Linux version requirement in its tool dependencies. Even if you did encounter that a boards platform stopped working after an update, you can always easily downgrade the platform back to the last working version via the Arduino IDE Boards Manager. Fix operations hanging while Serial Monitor/Plotter is open arduino/arduino-ide#2571, arduino/arduino-ide#2562, arduino/arduino-cli#2728, arduino/arduino-cli#2719 A bug was introduced in Arduino IDE 2.3.3 that caused operations such as downloading Boards Manager package index files to hang if performed while Serial Monitor or Serial Plotter is open. The bug has now been fixed. (Thanks @k5map and @starthemorning) Fix bug causing failed uploads to be indicated as successful arduino/arduino-ide#2518, arduino/arduino-ide#2516 At the termination of an upload operation, Arduino IDE displays a notification to communicate the result to the user. A bug was introduced in Arduino IDE 2.3.3 that caused the IDE to display the upload success notification even when the upload failed. This bug has now been fixed. Fix compilation caching when library installation or cache path contains non-ASCII characters arduino/arduino-ide#2571, arduino/arduino-cli#2733, arduino/arduino-cli#2671 In order to reduce the sketch compilation duration, Arduino IDE caches the compiled files and reuses those on subsequent compilations when appropriate. Previously, the caching did not work when libraries were installed under a path containing non-ASCII characters, or when the path where the cached files are stored contained such characters. This resulted in unnecessarily long compilation durations for the affected users. (Thanks @g91_y and @vvb333007) Fix hang on startup when Raspberry Pi Pico/RP2040 platform installed arduino/arduino-ide#2571, arduino/arduino-cli#2707, arduino/arduino-cli#2665, arduino/pluggable-discovery-protocol-handler#50 Previously, Arduino IDE would sometimes hang on startup if the user had previously installed @earlephilhower's excellent "Raspberry Pi Pico/RP2040" boards platform. Compatibility with that platform has now been restored. Kill tool process when upload is canceled arduino/arduino-ide#2571, arduino/arduino-ide#2517, arduino/arduino-cli#2726 A "CANCEL" button was added to the progress notification Arduino IDE shows during sketch compilation and upload. The user can click this button if for some reason they wish to cancel the operation (e.g., it is clear from the output that the upload is going to fail eventually). Previously, the button affected the Arduino IDE GUI, but didn't actually cancel the underlying process. This could result in spurious failures of subsequent operations if the collided with the still running process from the "canceled" operation. The upload tool process is now killed when the button is clicked. Display a more helpful message when cryptographic signature validation fails due to incorrect system time arduino/arduino-ide#2571, arduino/arduino-cli#2750 Index files used by Arduino IDE are cryptographically signed. Arduino IDE validates the signature of these files. Signatures are time stamped, so if the system time on a users's computer is set to a time in the past, the validation will fail due the signature being considered invalid at any time prior to the timestamp. This causes Arduino IDE to hang on startup, with an "Error verifying signature: openpgp: signature expired" error message in the logs. The error message displayed under these conditions has been updated to include the hint "is your system clock set correctly?". Don't fail platform installation on mismatch with size fields of package index arduino/arduino-ide#2571, arduino/arduino-cli#2739, arduino/arduino-cli#2332 The Arduino IDE Boards Manager is populated by data from index files. In addition to the primary index file maintained by Arduino, the user can specify additional 3rd party Arduino boards platforms to be included by adding URLs to the "Additional Boards Manager URLs" preference. The package index data includes the checksum and file size for each of the files that will be downloaded by Arduino IDE. The purpose of the checksum is to ensure the downloaded file has not been corrupted in any way. The purpose of the file size data is to allow Arduino IDE to display a progress indicator while downloading the file. Arduino IDE 1.x only validated the checksum after download, as this is fully sufficient. If the maintainer of a package index made an error in the file size value, the only impact on the user would be an inaccurate progress indicator, so package index maintainers who used Arduino IDE 1.x would not have any sign of their error. Previously, Arduino IDE 2.x validated the file size against the package index. This resulted in unnecessary boards platform installation failures. It is now changed to only validate against the checksum. Improve resiliency to errors in package indexes arduino/arduino-ide#2571, arduino/arduino-cli#2713 The Arduino IDE Boards Manager is populated by data from index files. In addition to the primary index file maintained by Arduino, the user can specify additional 3rd party Arduino boards platforms to be included by adding URLs to the "Additional Boards Manager URLs" preference. The maintainers of the package index files occasionally introduce errors in the data format, which might impact the functionality of Arduino IDE. Arduino IDE is now able to more gracefully handle such errors. Display donation links on Arduino IDE update arduino/arduino-ide#2581 Arduino IDE is a cornerstone of the Arduino initiative. The IDE, as well as all its helper tools and components, are free open source software. This software benefits the entire Arduino community, even those who do not use the boards sold by Arduino. The ongoing work to develop and maintain this software can be supported by donating to Arduino. The IDE now displays a link to the donation page during updates via the auto-updater, and on the first start after an update. Fix memory leak when scanning sketchbooks with large folders arduino/arduino-ide#2555, arduino/arduino-ide#2537 Previously, the IDE could become unresponsive if the sketchbook contained a folder with many/large files. Support compilation of very large sketch files arduino/arduino-ide#2571, arduino/arduino-cli#2729, arduino/arduino-cli#2718 Previously, compilation would fail spuriously if any sketch code file was larger than 4 MB. The maximum sketch file size has now been increased to 16 MB. (Thanks @jacek_kander) Store all temporary files in path indicated by build_cache.path arduino/arduino-ide#2571, arduino/arduino-cli#2673, arduino/arduino-cli#2668 When a sketch is compiled, Arduino IDE stores the generated files in a temporary folder. It is possible to configure the path under which the files are stored via Arduino CLI's build_cache.path configuration key. This capability might be required for some rare advanced use cases. Previously, the configuration only controlled where the generated Arduino core library files were cached, while sketch and library files were still stored in the default location. All generated files are now stored under the path specified by the build_cache.path configuration key when the user has added that key to the configuration file. Align viewsWelcome behavior to VS Code arduino/arduino-ide#2543 This change allows Arduino IDE extension developers to accomplish certain UI designs. (Thanks @dankeboy36 and @pjrc) Expand and improve translations arduino/arduino-ide#2523, arduino/arduino-ide#2571, arduino/arduino-cli#2692, arduino/arduino-cli#2757 The community work to make Arduino IDE accessible to everyone in the world through localization of the text of the IDE's UI continues, reaching high coverage levels for 18 languages/locales. If you would like to contribute to the Arduino IDE translations, please see the information here: Update Arduino CLI dependency to 1.1.1 arduino/arduino-ide#2571 Much of the non-GUI functionality of Arduino IDE comes from the Arduino CLI tool. In order to benefit from the ongoing development work in the Arduino CLI project, Arduino IDE has been updated to using Arduino CLI version 1.1.1. Full changelog here: If you have any questions or feedback please post here in the dedicated forum category for Arduino IDE 2.x: If you want to see the list of known issues, work in progress, submit a formal report, or contribute to development, the Arduino IDE 2.x source code is hosted in this public repository: HI Paul I started a new topic for this. As you suggested instead of using strings or pointer for choices I should use enum. Please let me know what I m doing wrong. void setup() { Serial.begin(9600); enum motion { UP, DOWN, STOP }; //