

U noArduSimV2.x Schnelle Hilfe

The screenshot displays the U noArduSimV2.x software interface, which is used for simulating Arduino projects. The interface is divided into several key sections:

- CodeBereich (Code Area):** Located on the left, it shows the C++ code for the simulation. The code includes logic for controlling a servo motor, a stepper motor, and an analog sensor. A green highlight is visible on the line `stepper1.step(10); //step FORWARD`.
- VariablesBereich (Variables Area):** Located at the bottom left, it displays the current values of variables defined in the code. The variables and their values are: `backval= 22 = '.'`, `count= 23 = 0x17`, `tics= 46 = 0x2e`, `digital_level= 0 = 0`, `analog_level= 671`, `numchars= 4 = 0x4`, and `angle= 180`. A green highlight is visible on the line `angle= 180`.
- LaborBankBereich (Labor Bank Area):** The central part of the interface, it contains a detailed schematic of the Arduino Uno board. The board is labeled "UNO ARDUINO" and "ATMEGA328". It shows various components like the USB port, power pins, and digital/analog pins. A green highlight is visible on the "UNO ARDUINO" label.
- Statusleiste (Status Bar):** Located at the bottom, it provides real-time feedback. It includes a "Fly-over Hint" button and a message that says "REACHED A Run TEMPORARY BREAKPOINT".
- Symbolleiste (Symbol Bar):** Located at the bottom, it contains icons for various hardware components that can be added to the simulation, such as push buttons, resistors, LEDs, and sensors.

Additional details visible in the interface include a menu bar (File, Find, Execute, Options, Configure, VarRefresh, Windows, Help), a toolbar with various icons, and a RAM free status of 1679. The I/O section on the right shows various output devices like LEDs and motors, each with its own control panel.

CodeBereich:







```
/* Use File->Load Prog to
load a different Program
*/

int count;



void setup()
{
  count=0;
}

void loop()
{
  count=count+1;
  delay(100);
}

//the "int main()" below is IMPLICIT in Arduino
//but is shown here EXPLICITLY by UnoArduSim
int main()
{
  setup();
  while(true)
  {
    loop();
    serialEventRun();
  }
}
```

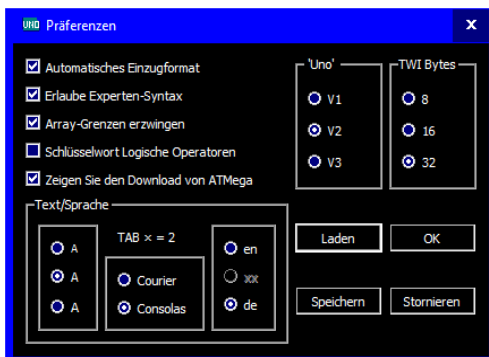
Schritt oder Ausführen mit , , , oder . Um **Halt an einer bestimmten Programmzeile**, zuerst c lecken , um diese Zeile zu markieren, und klicken Sie dann auf **Ausführen-Dort** . Um zu **stoppen, wenn eine bestimmte Variable geschrieben wird**, klicken Sie zuerst darauf, um sie zu markieren, und klicken Sie dann auf **Ausführen** .

Wechseln Sie zwischen den Funktionen, indem Sie auf klicken überall, dann benutze **PgDn** und **PgUp** (oder  und .

Setze Suchtext mit  und dann **zu diesem Text springt** mit  und .

Wechseln Sie zwischen '# include' Dateien mit  .

Präferenz:



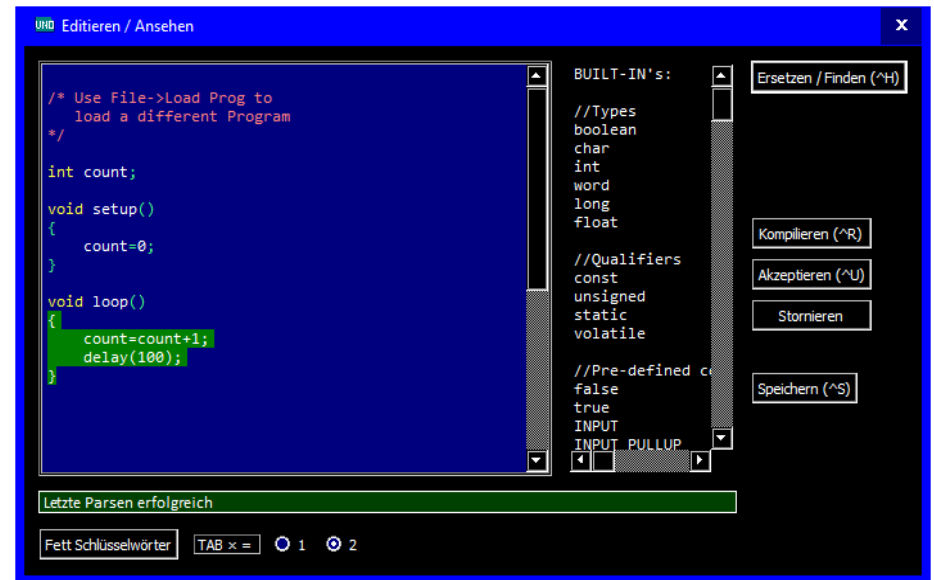
Konfigurieren | Präferenzen
Benutzereinstellungen festlegen, speichern und laden.

Alternative Sprache (n) eingestellt durch das Benutzergebietsschema und durch einen Zwei-Buchstaben-Code in der ersten Zeile der **myArduPrefs.txt** Voreinstellungsdatei

Editieren / Ansehen:

Um eine bestimmte Zeile zu öffnen, **doppelklicken Sie** auf die Zeile im **CodeBereich** oder verwenden Sie **Datei | Bearbeiten / Ansehen** (und es öffnet sich in der letzten markierten Zeile)

Der Tab-Einzug wird automatisch ausgeführt, wenn diese **Einstellung** aus **Configure | ausgewählt wird Einstellungen** - Sie können die Tabbreite auch einfach oder doppelt vergrößern.



Hinzufügen oder Löschen von Tabs zu einer Gruppe von Zeilen mit dem **Rechtspfeil** oder **TAB** und dem **Linkspfeil** (nach der ersten Auswahl einer Gruppe von 2 oder mehr aufeinander folgenden Zeilen) .

Um ein Element (nach der Einfügemarke) aus **der rechten Liste** der Eingezimmerten **hinzufügen** , doppelklicken Sie darauf.

Suchen (ctrl-F verwenden) , **Suchen / Ersetzen** (ctrl-H), **Zurücknehmen** (ctrl-Z) , **Wiederholen** (ctrl-Y)

Kompilieren und offen lassen (Strg-R) oder **Accept** (Strg-U) oder **Speichern** (Strg-S) zu schließen.

Finden Sie den **passenden Klammerpaarpartner** einer Klammer, **indem Sie darauf doppelklicken** - beide Klammern sowie der gesamte Text dazwischen werden hervorgehoben (wie im Bild oben).



Mit **ctrl-PgDn** und **ctrl-PgUp** springen Sie zum nächsten (oder vorherigen) Leerzeilenumbruch .

VariablenBereich:

```

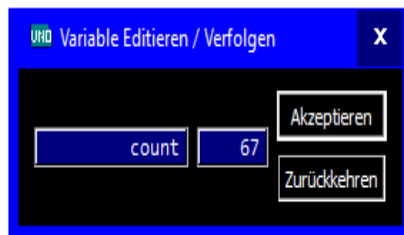
angle= 45
i= 8
k= 6
notefreq= 1046
dur= 0.12500
beats= 160
wholenote= 1500
quarternote= 375
msecs= 187
RingTones[0](-)
  RingTones[0](-)
  RingTones[0].frequency= 1046
  RingTones[0].duration= 0.12500
  
```

Klicken Sie auf (+) , um zu erweitern, oder auf (-) , um Arrays und Objekte zu reduzieren.

PgDn und **PgUp** (oder  und ) können Sie schnell springen zwischen Variablen .

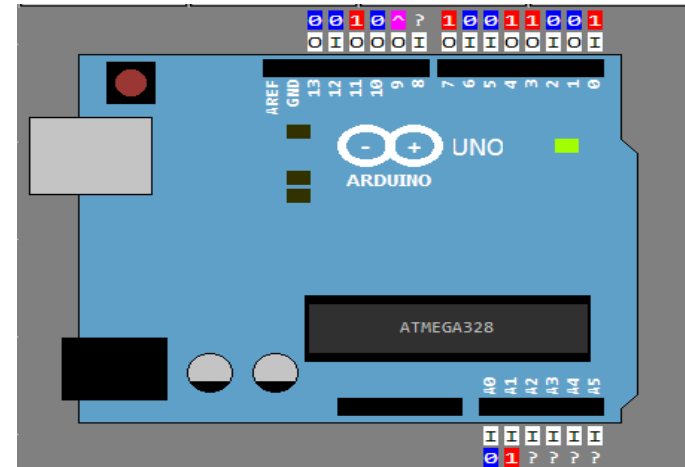
Verwenden Sie das **VarRefresh-** Menü, um die Aktualisierungshäufigkeit bei der Ausführung zu steuern.

Doppelklicken Sie auf eine Variable, um ihren Wert während der Ausführung zu verfolgen, oder ändern Sie sie während der (angehaltenen) Programmausführung in einen neuen Wert:

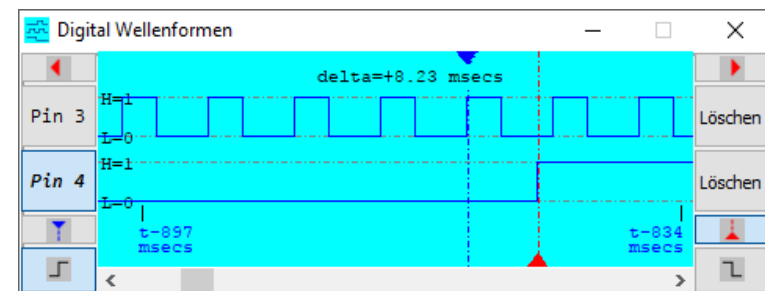


Oder **Einzelklick Um** eine Variable (oder ein **Objektelement** oder ein **Arrayelement**) **hervorzuheben** , verwenden Sie **Ausführen-Bis** , um die Ausführung bis zum nächsten **Schreibzugriff** auf diese Variable oder Position **fortzusetzen** .

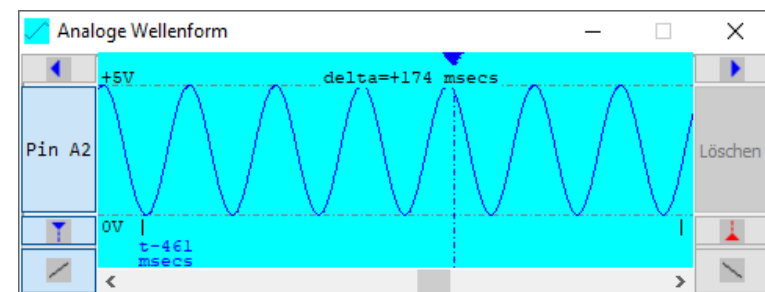
LaborBankBereich und 'Uno':



Klicken Sie mit der **linken** Maustaste auf einen Pin, um Pin Digital WaveForms zu erstellen (oder hinzuzufügen):



Klicken Sie mit der **rechten** Maustaste auf einen Pin, um ein Pin Analog WaveForm-Fenster zu erstellen:



Um **HINEINZOOMEN** und **RAUSZOOMEN** (Zoom ist immer auf den ACTIVE-Cursor zentriert) zu drehen, verwenden Sie das Mausrad oder die Tastenkombination **STRG-NACH-OBEN** und **STRG-AB-Pfeil** .

LaborBankBereich 'I/O' Geräte

Legen Sie die Nummern und Typen von jedem unter Verwendung von **Konfigurieren | 'I/O' Geräte**. Stellen Sie die Pins mit einem zweistelligen Wert zwischen 00 und 19 (oder A0-A5) ein.

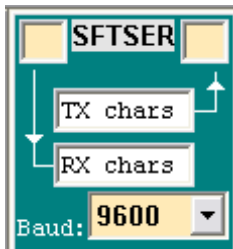
'Serial' Monitor ('SERIAL')



Geben Sie ein oder mehrere Zeichen in das obere Eingabefeld ('TX chars') ein und **drücken Sie Return**.

Doppelklicken Sie, um ein größeres Fenster für TX- und RX-Zeichen zu öffnen.

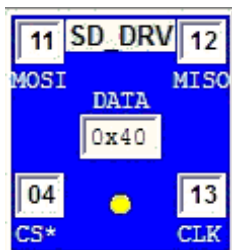
Software Serieller ('SFTSER')



Geben Sie ein oder mehrere Zeichen in das obere Eingabefeld ('TX chars') ein und **drücken Sie Return**.

Doppelklicken Sie, um ein größeres Fenster für TX- und RX-Zeichen zu öffnen

SD-Laufwerk ('SD DRV')

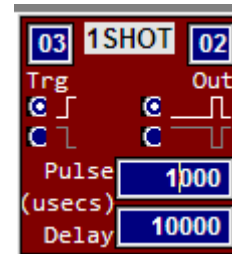


Eine kleine 8-MByte SD-Disk, die von SPI-Signalen gesteuert wird und in einem 'SD' -Unterverzeichnis im geladenen Programmverzeichnis gespiegelt wird (das bei fehlendem Programm erstellt wird)

Doppelklicken, um ein größeres Fenster zu öffnen sehen Verzeichnisse, Dateien und Inhalte

CS* niedrig zum Aktivieren.

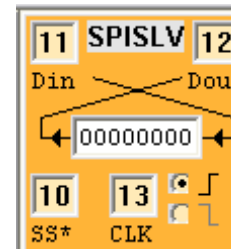
Generator Ein-Schuss ('1SHOT')



Ein digitaler One-Shot. Erzeugt nach einer bestimmten Verzögerung einen Impuls mit der gewählten Polarität auf "Out", entweder von einer steigenden oder einer fallenden Triggerflanke, die an seinem Trg- Eingang zu sehen ist. Einmal ausgelöst, werden nachfolgende Triggerflanken ignoriert, bis der Impuls auf "Out" vollständig abgeschlossen wurde.

'Pulse' und 'Delay' Werte (wenn ein 'S' angehängt ist) wird über den Schieberegler "I / O ____ S" skaliert

Schieberegister Sklave ('SRSLV')

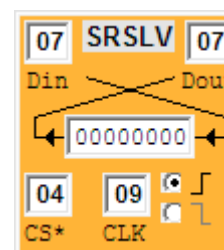


Ein einfaches Schieberegister.

Kantenübergänge am CLK lösen eine Verschiebung aus.

SS* niedrig, treibt MSB auf 'Dout'.

Konfigurierbarer SPI-Sklave ('SPISLV')

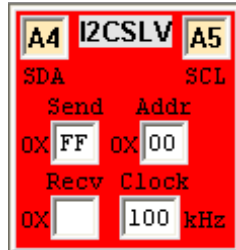


Ein moduskonfigurierbares SPI-Slave-Gerät ('MODE0', 'MODE1', 'MODE2' oder 'MODE3')

Doppelklicken, um ein größeres Fenster zu öffnen, um hex 'DATA' einzustellen / anzuzeigen und 'Recv'- Bytes.

SS* niedrig, treibt MSB auf MISO

Zweiadrigter I2C-Sklave ('I2CSLV')



Ein *Slave-Mode-Only*- I2C-Gerät.

Doppelklicken , um ein **größeres Fenster** zu öffnen
um die Bytes '**Send**' und '**Recv**' zu setzen /
anzuzeigen

Schrittmotor ('STEPR')

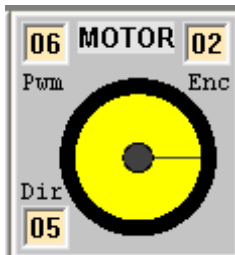
Akzeptiert Steuersignale **an 2 oder 4 Pins.** *'Schritte' müssen ein Vielfaches von 4 sein.*



Verwenden Sie '#include <Stepper.h>'. Um die Getriebeübersetzung in Ihrem Programm mit N zu **simulieren** , verwenden Sie einen Modulo-N-ähler, um zu bestimmen, wann '**Stepper.step ()**' tatsächlich aufgerufen werden soll.

Gleichstrommotor ("MOTOR")

Akzeptiert PWM-Signale an Pwm Pin, Pegelsignal an Dir und gibt 8 **Höhen** und 8 **Tiefs** pro Radumdrehung an Enc aus .



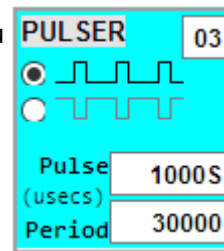
Die volle Geschwindigkeit beträgt ca. 2 Umdrehungen pro Sekunde.

ServoMotor ('SERVO')



Akzeptiert gepulste Steuersignale am angegebenen Pin. Wenn das Kontrollkästchen unten links aktiviert ist, wird das Servo zu einem Servo mit kontinuierlicher Rotation mit der gleichen maximalen Geschwindigkeit, aber jetzt stellt die PWM-Impulsbreite die Geschwindigkeit und nicht den Winkel ein

Digitaler Impulsgeber ('PULSER')

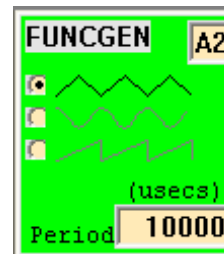


Erzeugt digitale Wellenformsignale auf dem angegebenen Pin.

Die minimale Dauer beträgt 50 Mikrosekunden, die minimale Impulsbreite 10 Mikrosekunden. Beide Werte (wenn ein S angehängt ist). wird über den Schieberegler "I / O ____ S" skaliert

Wählen Sie positive Impulse (0 bis 5 V) oder negative Impulse (5 V bis 0 V).

Analoger Funktionsgenerator ('FUNCGEN')



Generiert analoge Signale auf dem angegebenen Pin.

Die minimale 'Periode' beträgt 100 Mikrosekunden, skaliert vom Schieberegler 'I / O ____ S' (wenn ein 'S' angehängt ist).

Sinus-, Dreieck- oder Sägezahnwellenformen..

Piezelektrischer Lautsprecher ('PIEZO')



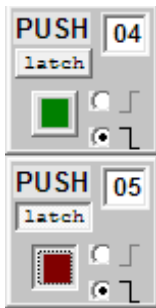
"Anhören" von Signalen an einem beliebigen 'Uno'-Pin.

Farbige LED ('LED')



R-, Y-, G- oder B-LED, die zwischen einem beliebigen 'Uno'-Pin und entweder Masse oder +5 V angeschlossen ist.

Druckknopf ('PUSH')



Ein normalerweise geöffneter **Momentan**-Druckknopf zu + 5V oder Boden

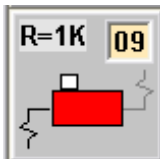
Ein normalerweise offenes **Rastgegen** Buttin bis + 5 V oder Masse
(Drücken Sie die **Latch** -Taste, um diesen Modus zu erhalten)

Analoger Schieberegler

Ein slidergesteuertes Potentiometer. 0-5V, um einen beliebigen 'Uno'-Pin zu steuern.







Schiebeschalterwiderstand ('R = 1K')



Ein hochziehen von 1 k-Ohm bis + 5 V ODER ein herunterziehen von 1 k-Ohm gegen Masse..

Menüs:

Datei:

 <u>Lade INO oder PDE Programm (Strg-L)</u>	Ermöglicht dem Benutzer, eine Programmdatei mit der ausgewählten Erweiterung auszuwählen. Das Programm wird sofort analysiert
<u>Editieren / Ansehen (Strg-E)</u>	Öffnet das geladene Programm zum Anzeigen / Editieren.
 <u>Speichern</u>	Speichern Sie den bearbeiteten Programminhalt zurück in die ursprüngliche Programmdatei.
<u>Speichern Als</u>	Speichern Sie den bearbeiteten Programminhalt unter einem anderen Dateinamen.
 <u>Nächste ('#include')</u>	Bringt den CodeBereich vor , um die nächste '#include' -Datei anzuzeigen
 <u>Vorherige</u>	Gibt die CodeBereichanzeige an die vorherige Datei zurück
<u>Ausgang</u>	Beendet UnoArduSim, nachdem der Benutzer daran erinnert wurde, geänderte Dateien zu speichern.

Konfigurieren:

<u>'I / O' Geräte</u>	Öffnet einen Dialog, in dem der Benutzer die gewünschten Typen und Nummern der gewünschten 'I / O' -Geräte auswählen kann. In diesem Dialogfeld können Sie auch 'I / O' Geräte in eine Textdatei speichern und / oder 'I / O' Geräte aus einer zuvor gespeicherten (oder bearbeiteten) Textdatei laden (einschließlich aller PIN-Anschlüsse und anklickbaren Einstellungen und Eingaben Werte).
------------------------------	---

Präferenz

Öffnet einen Dialog, in dem der Benutzer Präferenz wie das automatische Einrücken von Quellprogrammzeilen festlegen kann. Expertensyntax, Schriftartenwahl, größere Schriftgröße, Erzwingen von Array-Grenzen, Erlauben von logischen Operatorwörtern, ProgrammHerunterladen, Auswahl von 'Uno' Leiterplatte-Version und TWI-Pufferlänge (für I2C-Geräte).

Finden:

 <u>Nächste Funktion/Var</u>	Springe zur nächsten Funktion im CodeBereich (wenn sie den aktiven Fokus hat) oder zur nächsten Variablen im VariablenBereich (wenn sie stattdessen den aktiven Fokus hat).
 <u>Vorherige Funktion/Var</u>	Springe zur vorherigen Funktion im CodeBereich (wenn es den aktiven Fokus hat) oder zur vorherigen Variablen im VariablenBereich (wenn es stattdessen den aktiven Fokus hat).
 <u>Setze Suchtext (Strg-F)</u>	Aktivieren Sie das Bearbeitungsfeld Werkzeugleiste Suchen, um den Text zu definieren, nach dem als nächstes gesucht werden soll (und fügt das erste Wort aus der aktuell markierten Zeile im CodeBereich oder VariablenBereich hinzu , wenn einer davon den Fokus hat).
 <u>Finde den Nächsten Text</u>	Springt zum nächsten Text-Vorkommen im CodeBereich (wenn es den aktiven Fokus hat) oder zum nächsten Text-Vorkommen im VariablenBereich (wenn es stattdessen den aktiven Fokus hat).
 <u>Vorherigen Text Finden</u>	Zum vorherigen Text-Vorkommen im CodeBereich springen (wenn es den aktiven Fokus hat) oder zum vorherigen Text-Vorkommen im VariablenBereich (wenn es stattdessen den aktiven Fokus hat).

Ausführen:

 Schritt Hinein (F4)	Die Ausführung wird um eine Anweisung vorwärts oder <i>in eine aufgerufene Funktion fortgesetzt</i> .
 Schritt Über (F5)	Die Ausführung wird um eine Anweisung vorwärts oder <i>um einen vollständigen Funktionsaufruf</i> .
 Schritt Aus (F6)	Erweitert die Ausführung um <i>gerade genug, um die aktuelle Funktion zu verlassen</i> .
 Ausführen Dort (F7)	Führt das Programm aus <i>und hält an der gewünschten Programmzeile an</i> - Sie müssen zuerst auf eine gewünschte Programmzeile klicken, bevor Sie Ausführen-Dort verwenden.
 Ausführen Bis(F8)	Führt das Programm so lange aus, bis die Variable mit der aktuellen Hervorhebung im VariablenBereich aufschreibt (klicken Sie auf eine Variable , um die anfängliche Hervorhebung festzulegen).
 Ausführen (F9)	Führt das Programm aus.
 Halt (F10)	Stoppt die Programmausführung (<i>und friert die Zeit ein</i>).
 Zurücksetzen	Setzt das Programm zurück (alle Wertvariablen werden auf den Wert 0 zurückgesetzt und alle Zeigervariablen werden auf 0x0000 zurückgesetzt).
Animieren	Schreitet automatisch fortlaufende Programmzeilen <i>mit zusätzlicher künstlicher Verzögerung</i> und Hervorhebung der aktuellen Codezeile. Echtzeitbetrieb und Sounds sind verloren.
Zeitlupe	Verlangsamt die Zeit um den Faktor 10.

Optionen:

<u>Schritt Über Struktoren / Operatoren</u>	Fliege während eines Schrittes direkt durch Konstruktoren, Destrukturen und Operator-Überladungsfunktionen (dh es wird nicht innerhalb dieser Funktionen gestoppt).
<u>Register-Zuweisungsmodellierung</u>	Weisen Sie Funktion Locals zu, ATmega-Registern statt zu dem Stapel freizugeben (verursacht etwas reduzierte RAM-Auslastung).
<u>Fehler bei Nicht initialisiert</u>	Wird als Parse-Fehler überall dort angezeigt, wo Ihr Programm versucht, eine Variable zu verwenden, ohne zuvor ihren Wert (oder mindestens einen Wert in einem Array) initialisiert zu haben.
<u>Hinzugefügt 'loop()' Verzögerung</u>	Addiert 200 Mikrosekunden Verzögerung jedes Mal, wenn 'loop()' aufgerufen wird (falls keine weiteren Programmaufrufe an 'delay()' irgendwo vorhanden sind) - nützlich, um zu vermeiden, zu weit hinter Echtzeit zu fallen.
<u>Schritt Über Struktoren / Operatoren</u>	Fliege während eines Schrittes direkt durch Konstruktoren, Destrukturen und Operator-Überladungsfunktionen (dh es wird nicht innerhalb dieser Funktionen gestoppt).

VarAktualisieren:

<u>Erlaube Automatisches(-) Kontraktion</u>	Lassen Sie UnoArduSim angezeigte erweiterte Arrays / Objekte ausblenden, wenn sie in Echtzeit zurückfallen.
<u>Erlaube Reduktion</u>	Lassen Sie im VariablenBereich eine geringere Häufigkeit von Bildschirmaktualisierungen zu, um Flimmern zu vermeiden oder die CPU-Last zu reduzieren, wenn Sie hinter der Echtzeit zurückbleiben - dann werden die angezeigten Werte nur in regelmäßigen Abständen, aber auch bei angehaltenem Programm aktualisiert .
<u>Minimal</u>	Aktualisieren Sie die VariablenBereich -Anzeige nur 4 Mal pro Sekunde.
<u>Änderungen Hervorheben</u>	Markieren Sie den zuletzt geänderten Variablenwert (kann Verlangsamung verursachen).

Fenster:

<u>'Serial' Monitor</u>	Verbinden Sie ein seriell es E / A-Gerät mit den Pins 0 und 1 (falls nicht vorhanden), und ziehen Sie ein größeres ' Serial ' -Monitor TX / RX-Textfenster nach oben.
<u>Alles Wiederherstellen</u>	Stellen Sie alle minimierten untergeordneten Fenster wieder her.
<u>Pin DigitalWellenformen</u>	Stellen Sie ein minimiertes Pin DigitalWellenformen Fenster wieder her.
<u>Pin AnalogWellenform</u>	Stellen Sie ein minimiertes Pin AnalogWellenform Fenster wieder her.
<u>'Serial' Monitor</u>	Verbinden Sie ein seriell es E / A-Gerät mit den Pins 0 und 1 (falls nicht vorhanden), und ziehen Sie ein größeres ' Serial ' -Monitor TX / RX-Textfenster nach oben.

Hilfe:

<u>Schnellhilfe-Datei</u>	Öffnet die UnoArduSim_QuickHelp PDF-Datei.
<u>Vollständige Hilfedatei</u>	Öffnet die UnoArduSim_FullHelp PDF-Datei.
<u>Fehlerbehebung</u>	Wichtige Fehlerkorrekturen seit der letzten Version
<u>Änderung / Verbesserungen</u>	Wichtige Änderungen und Verbesserungen seit der letzten Version
<u>Über</u>	Zeigt Version, Copyright an.