

UnoArduSimV2.7 Schnelle Hilfe

The screenshot shows the UnoArduSim V2.7 software interface. The main window is divided into several sections:

- Code Editor (left):** Contains C++ code for controlling an Arduino Uno. It includes comments and function calls like `tone()`, `servo.write()`, `stepper1.step()`, `analogRead()`, and `digitalRead()`. An orange label **CodeBereich** points to this area.
- Hardware Simulation (center):** A detailed 3D model of an Arduino Uno board. An orange label **LaborBankBereich** points to the board itself.
- Component Panels (right and bottom):** Various peripheral components are shown in numbered panels:
 - 01 SERIAL:** TX chars, Baud: 300.
 - 06 PULSER:** Pulse 50000, Period 100000.
 - 11 SPISLV:** MOSI, MISO, DATA, Recv, SS*, Mode, SCK.
 - 09 SERVO:** Servo motor control.
 - 02-07 LED:** Six LED indicators with color selection (R, Y, G, B).
 - 08 PIEZO:** Two piezo buzzers.
 - 06 MOTOR:** Motor control with Pwm and Enc.
 - 03 STEPR:** Stepper motor control with P1, P2, steps, and 04.
 - FUNCGEN:** Function generator with A2, Period, and (usecs).
 - A1, A0:** Two 5V power supply outputs.
- Bottom Bar:** Contains a **Fly-over Hint** button and a status message: **ERREICHT A Ausführen Temporary Haltepunkt**. An orange label **Statusleiste** points to this bar.

Additional orange labels highlight specific features:

- I/O'-Wert Mit $0,0 < S \leq 1,0$ multiplizieren:** Points to the I/O status bar at the top right.
- VariablenBereich:** Points to the variable declaration and assignment section of the code.
- Symbolleiste Hinweise zum Überfliegen:** Points to the **Fly-over Hint** button.

CodeBereich:







```
/* This is a default program--
   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();
    delay(1); //added by the Arduino team
  }
}
```

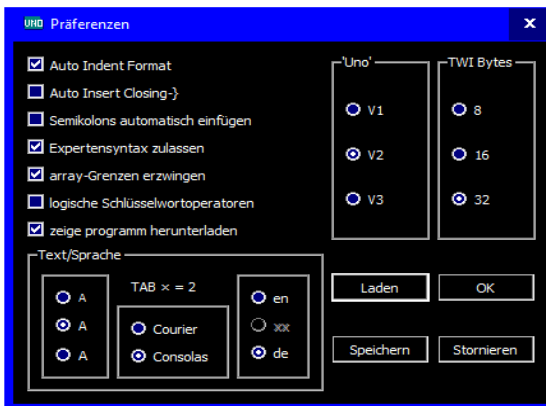
Schritt oder Ausführen mit , , , oder . Zu **Halt bei einer spezifischen Linie** programm zuerst eine Lecke diese Linie zu markieren, und dann klicken **Ausführen Dort** . Zu **Halt wann ein bestimmtes variable wird geschrieben**, Klicken Sie zuerst darauf, um es markieren, und dann klicken **Ausführen Bis** .

Navigieren Sie den Anruf-Stapel Verwendung  und . oder **springen zwischen funktionsmodule** Beim Klicken überall, dann verwenden **PgDn** und **PgUp**.

Legen Sie den Suchtext fest mit , und dann **Springe zu diesem Text** mit  und .

Bewegen Sie sich zwischen '#include' dateien mit  .

Präferenzen:



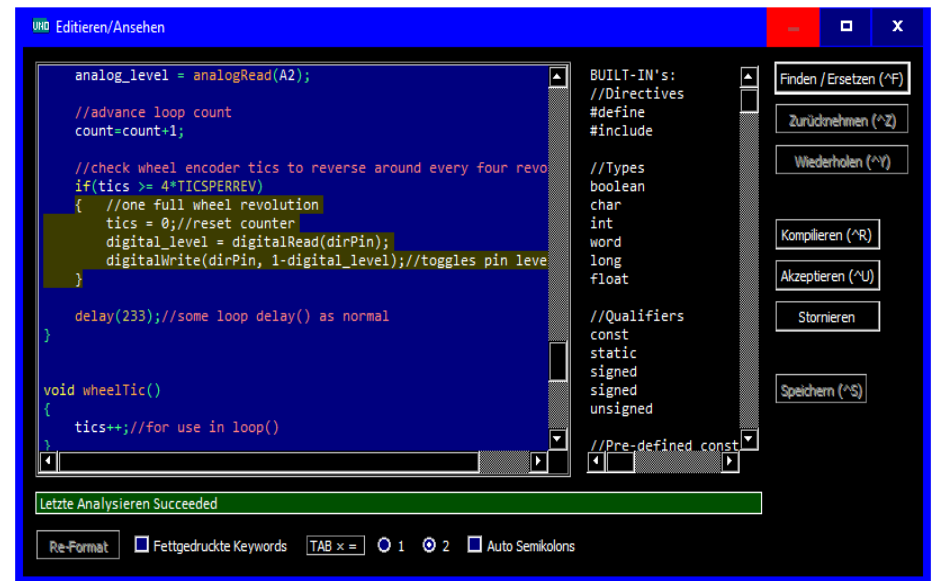
Konfigurieren | Voreinstellungen zum Festlegen, Speichern und Laden von Benutzeroptionen.

Alternative Sprache (n) eingestellt nach dem Gebietsschema des Benutzers und durch ein Zwei-Buchstaben-Code in der ersten Zeile des myArduPrefs.txt Präferenzen datei

Editieren/Ansehen:

Um in einer bestimmten Zeile zu öffnen, **Doppelklick** in dieser Zeile n das **CodeBereich** oder verwenden **Datei | Editieren/Ansehen** (und es öffnet sich bei der letzten hervorgehobenen Zeile)

Tabulator-Einrückung erfolgt automatisch wenn diese Vorliebe wird gewählt aus **Konfigurieren | Voreinstellungen** - Sie können die Tab-Breite auch ein- oder zweifach vergrößern.



Tabs hinzufügen oder löschen zu einer Gruppe von Zeilen mit **rechter Pfeil** oder **TAB** und **linker Pfeil** (nach erstem Auswählen einer Gruppe von 2 oder mehr aufeinanderfolgenden Zeilen).

So fügen Sie einen Artikel hinzu (nach dem Caret) **f von der rechten Liste der Built-Ins**, Doppelklicken Sie darauf. **Mit ALT-Pfeil nach rechts** auf Anfrage Auto-Vervollständigung Entscheidungen für eingebaut **globale variablen**, und für **Mitglied variablen und funktionsmodule**.

Finden (benutze Strg-F), **Finden / Ersetzen** (benutze Strg-H), **Zurücknehmen** (Strg-Z), **Wiederholen** (Strg-Y) **Kompilieren** und offen lassen (Strg-R) oder **Akzeptieren** (Strg-U) oder **Speichern** (Strg-S) schließen.

Finden a **passender geschweifte klammer**- Doppelklicken Sie auf, um einen Partner zu verbinden - sowohl geschweifte klammern als auch der gesamte Text dazwischen werden hervorgehoben (wie in der Abbildung oben).

Benutzen **Strg-PgDn** und **ctrl-PgUp** zu springen nächster (oder vorheriger) Leerzeilenumbruch.

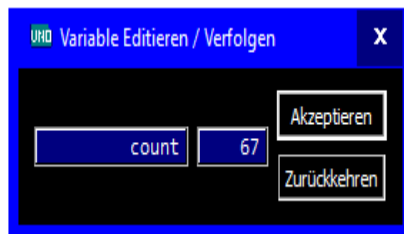
VariablenBereich:

```
LED_pin= 5
angle= 135
i= 3
k= 6
notefreq= 1046
dur= 0.12500
beats= 160
wholenote= 1500
quarternote= 375
msecs= 375
RingTones[0](-)
  RingTones[0](-)
    RingTones[0].frequency= 1046
    RingTones[0].duration= 0.12500
```

Klicke auf (+) bis erweitern oder weiter (-) bis zusammenziehen arrays und objekte.

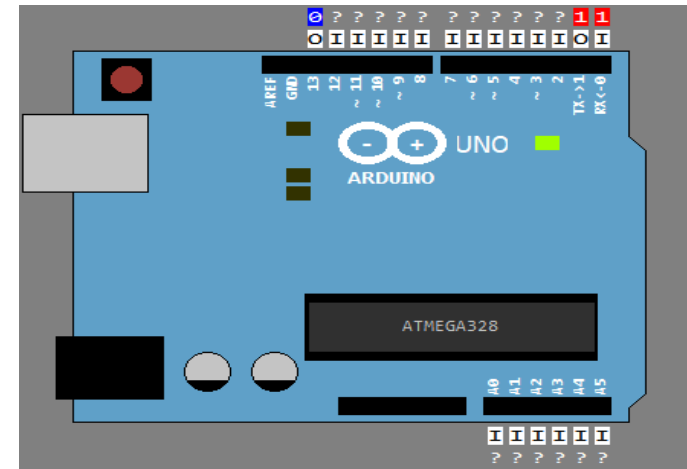
Verwenden Sie die **VarAktualisieren** Menü zur Steuerung der Aktualisierungshäufigkeit bei der Ausführung.

Doppelklick auf jedem variable, um seinen Wert während ausführung zu verfolgen oder in der Mitte von (angehalten) programm ausführung auf einen neuen Wert zu ändern:

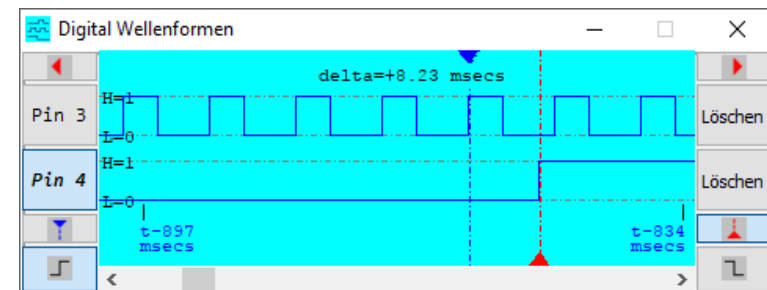


Oder **ein einziger Klick** bis markieren ein beliebiges variable (oder objekt-Element oder array-Element), dann verwenden **Ausführen Bis** ausführung bis zum nächsten vorrückten **Schreibzugriff** zu diesem variable oder Standort.

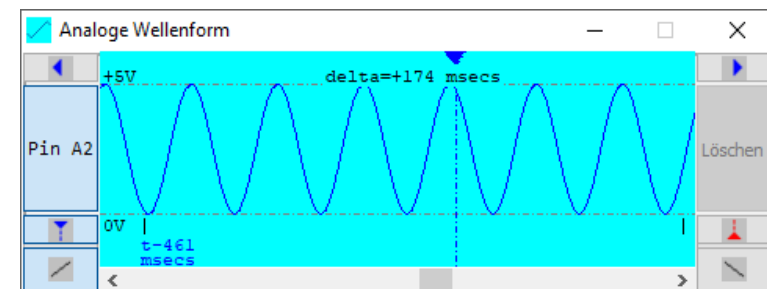
LaborBankBereich und der 'Uno':



Links Klick auf einem beliebigen pin, um Pin Digital Wellenformen zu erstellen (oder hinzuzufügen):



Rechtsklick auf einem beliebigen pin, um einen Pin Analoge Wellenform fenster zu erstellen:



Zu **HINEINZOOMEN** und **RAUSZOOMEN** Verwenden Sie das Mausrad oder Tastenkombinationen **STRG-Aufwärtspfeil** und **STRG-Abwärtspfeil**. 'Art **'Ctrl-S'** um den wellenform zu retten (**X, Y**) verweist auf einen Text datei (**'X'** ist Mikrosekunden von links, **'Y'** ist Volt)

LaborBankBereich 'I/O' Geräte

Stellen Sie die Nummern und Typen mit Konfigurieren ein | 'I/O' Geräte.
Stellen Sie pins mit einem 2-ziffer-Wert von 00 bis 19 (oder A0-A5) ein.



Einige dieser geräte unterstützen die Skalierung ihrer eingegebenen Werte mithilfe des Schiebereglers in der fenster-Hauptsymbolleiste (siehe 'I/O ___ S' unter jedem Schlauch geräte unten):

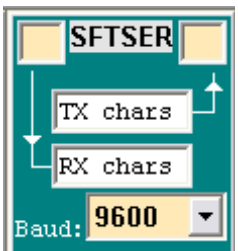
'Serial'-Monitor ('SERIAL')



Geben Sie ein oder mehrere Zeichen in das obere Bearbeitungsfeld ('TX chars') und ein **Drücken Sie die Eingabetaste**.

Doppelklick (oder Rechtsklick) öffnen **ein größerer fenster für TX- und RX-Zeichen**.

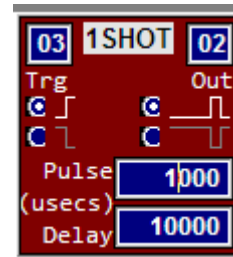
Software Serieller ('SFTSER')



Geben Sie ein oder mehrere Zeichen in das obere Bearbeitungsfeld ('TX chars') und ein **Drücken Sie die Eingabetaste**.

Doppelklick (oder Rechtsklick) öffnen **a größeres fenster für TX- und RX-Zeichen**.

Generator Ein Schuss ('1SHOT')

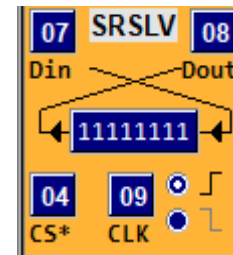


Ein digital One-Shot. Erzeugt einen Impuls mit gewählter Polarität an 'Out' nach einer festgelegten Verzögerung von entweder a steigende oder fallende auslösende Flanke an der 'Trg' Eingang. Einmal ausgelöst, ignoriert es nachfolgende Triggerflanken bis zum Puls an 'Out' wurde vollständig abgeschlossen.

'Pulse' und 'Delay' Werte (wenn mit einem 'S' versehen). wird über den 'I/O ___ S'-Schieberegler

in der Symbolleiste skaliert

Schieberegister Sklave ('SRSLV')

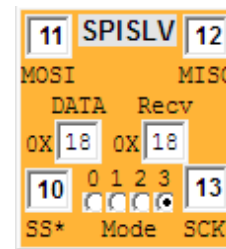


Ein einfaches Schieberegister gerät.

Flankenübergänge auf CLK werden Schalthebel.

SS * low, treibt an MSB auf Dout.

SPI Sklave ('SPISLV')

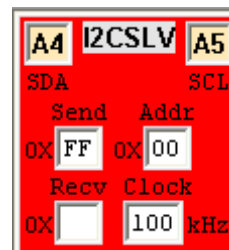


Ein moduskonfigurierbarer SPI-Slave gerät ('MODE0', 'MODE1', 'MODE2' oder 'MODE3')

Doppelklick (oder Rechtsklick) öffnen **ein größerer fenster Hex setzen / sehen 'DATA' und 'Recv' Bytes**.

SS * low, treibt an MSB auf MISO.

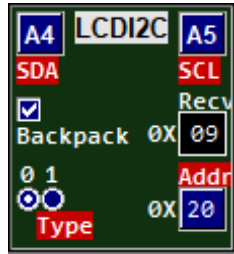
Zweileiter I2C Sklave ('I2CSLV')



EIN Nur im Slave-Modus I2C gerät.

Doppelklick (oder Rechtsklick) öffnen **ein größerer fenster Hex setzen / sehen 'Send' und 'Recv' Bytes**

Text LCD I2C ('LCDI2C')



A 1,2, 04 4-line Zeichen-LCD, in einem von drei Modi (2 Rucksack styles, plus ein nativen Modus), mit Bibliothek-Code für jeden Modus gerät Stütz innerhalb der 'include_3rdParty'-Ordner zur Verfügung gestellt.

Doppelklick (Oder Rechtsklick) öffnen **eine größere fenster sehen der LCD-Bildschirm** (Und setzt es

Größe)

Text LCD SPI ('LCDSPI')

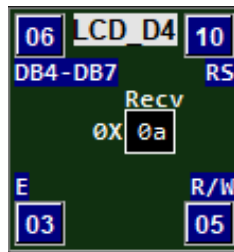


A 1,2, 04 4-line Zeichen-LCD, (SYLE ein Rucksack, plus ein nativen Modus) in einem von zwei Modi, wobei für jede Betriebsart gerät Bibliothekscode Stütz innerhalb der 'include_3rdParty'-Ordner zur Verfügung gestellt.

Doppelklick (Oder Rechtsklick) öffnen **eine größere fenster sehen der LCD-Bildschirm** (Und setzt es Größe)

Text LCD SPI ('LCDSPI')

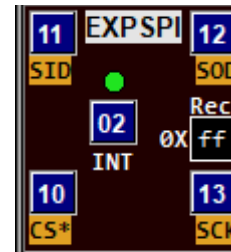
A 1,2, 04 4-line Zeichen-LCD, (SYLE ein Rucksack, plus ein nativen Modus) in einem von zwei Modi, wobei für jede Betriebsart gerät Bibliothekscode Stütz innerhalb der 'include_3rdParty'-Ordner zur Verfügung gestellt.



Doppelklick (Oder Rechtsklick) öffnen **eine größere fenster sehen der LCD-Bildschirm** (Und setzt es Größe)

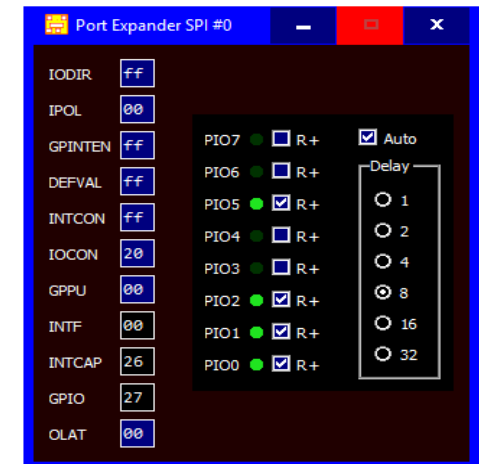


Erweiterungsport SPI ('EXP SPI')

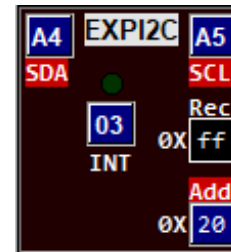


Ein 8-bit Port-Expander basierend auf der MCP23008, mit Stütz 'MCP23008.h' Code innerhalb des 'include_3rdParty' vorgesehen Mappe. Sie können zu MCP23008 Register schreiben und lesen Sie die GPIO pin zurück Ebene. Interrupts können auf jeder GPIO pin Änderung aktiviert werden - ein ausgelöster Interrupt die 'INT' pin antreiben.

Doppelklick (Oder Rechtsklick) öffnen **eine größere fenster sehen das 8 GPIO Portleitungen** und die angebrachten Pull-up-Widerstände. Sie können Klimmzüge manuell ändern, indem Sie oder einen Zähler anhängen, die sie in regelmäßigen Abständen in einer Aufwärtszählungs Weise verändern wird.

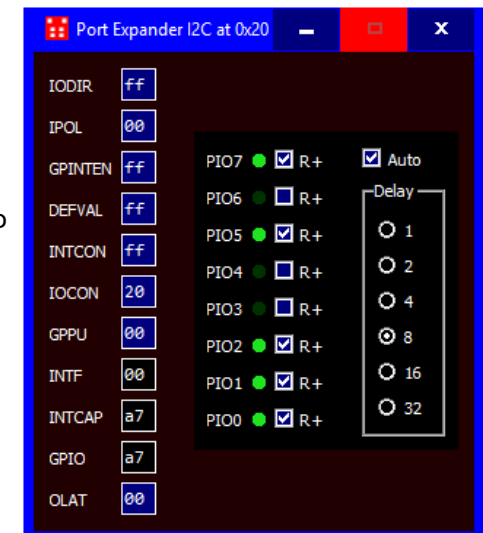


Erweiterungsport I2C ('EXPI2C')

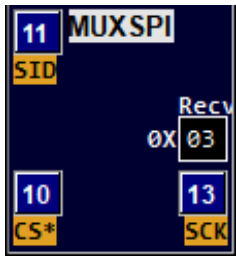


Ein 8-bit Port-Expander basierend auf der MCP23008, mit Stütz 'MCP23008.h' Code vorgesehen innerhalb des 'include_3rdParty' Mappe. Fähigkeiten entsprechen den 'EXP SPI' gerät.

Doppelklick (Oder Rechtsklick) zu öffnen eine größere fenster wie fro der 'EXP SPI' gerät.



Mux LED SPI ('MUXSPI')



EIN multiplexierten-LED-Controller basierend auf dem MAX6219 . mit Unterstützung 'MAX7219.h' Code vorgesehen innerhalb des 'include_3rdParty' Ordner nach antreiben bis zu acht 7-Segment-Ziffern.

Doppelklick (Oder Rechtsklick) zu öffnen eine größere fenster sehen die farbigen 7-Segment-Anzeige ziffer.

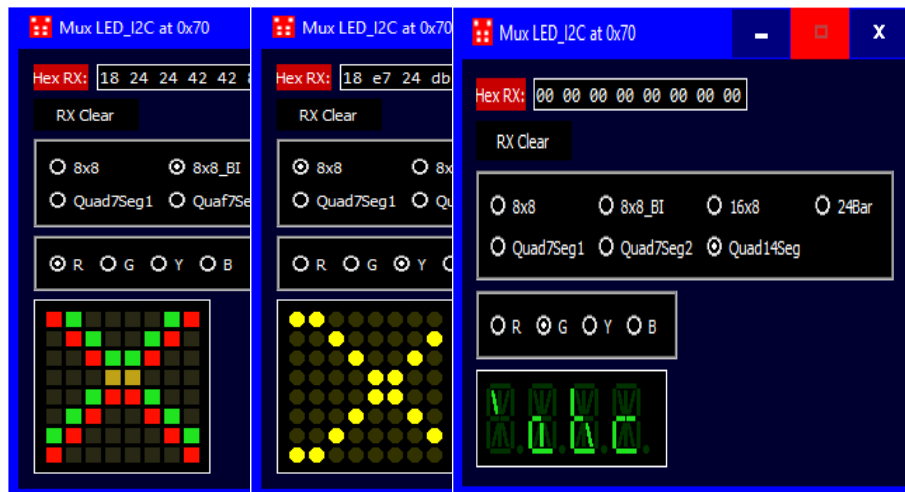


Mux LED I2C ('MUXI2C')



EIN multiplexierten-LED-Controller basierend auf der HT16K33 . mit Unterstützung **Adafruit_LEDBackpack.h** Code vorgesehen innerhalb des 'include_3rdParty' Mappe.

Doppelklick (Oder Rechtsklick) zu öffnen eine größere fenster zur Auswahl und Ansicht einer von mehreren farbigen LED Displays.



Schrittmotor ('STEPR')



Akzeptiert Steuersignale *entweder auf 2 oder 4 pins*. **'Steps' muss ein Vielfaches von 4 sein.**

Benutzen '#include <Stepper.h>' .

Verwenden Sie einen Modulo-N-Zähler, um die Untersetzung um N in Ihrem programm zu emulieren und festzustellen, wann Sie tatsächlich anrufen müssen

'Stepper.step()'

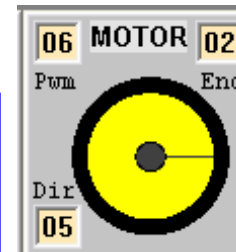
Gepulst Schrittmotor ('PSTEPR')



Jede steigende Flanke auf 'STEP' verursacht einen (Mikro-) Schritt in die Richtung von 'DIR' gesteuert, wenn sie von einem niedrigen auf 'EN' aktiviert . **'Steps' muß ein Vielfaches von 4 sein**, und **'micro'** muss sein **1,2,4,8 oder 16 Mikroschritte pro Vollschritt.**

Emulieren Getriebeuntersetzung von N in Ihrem programm, verwenden Sie einen Modulo-N-Zähler zu bestimmen, wann tatsächlich Aufruf gepulst.

GleichstromMotor ('MOTOR')



Akzeptiert eingeschaltete PWM-Signale **Pwm** pin, Pegelsignal auf **Dir** und gibt 8 Höhen und 8 Tiefen pro Rad aus Revolution auf **Enc** .

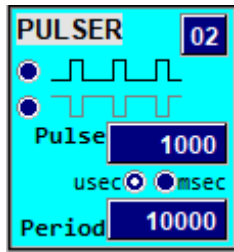
Die volle Geschwindigkeit ist ungefähr 2 Umdrehungen pro Sekunde .

ServoMotor ('SERVO')



Akzeptiert gepulste Steuersignale auf dem angegebenen pin. Kann durch Aktivieren des Kontrollkästchens unten links in eine kontinuierliche Rotation geändert werden

Digitaler Impulsgeber ('PULSER')



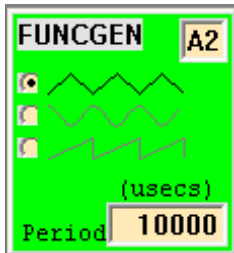
Erzeugt digital wellenform Signale auf bestimmten pin. EIN Nur im Slave-Modus I2C gerät.

Wählen Sie Zeitbasis in Millisekunden ('msec') oder Mikrosekunden ('usec')

Minstdauer beträgt 50 Mikrosekunden, minimale Impulsbreite 10 Mikrosekunden. Beide Werte (wenn mit einem 'S' suffixed). wird aus der Werkzeugleiste 'I/O ____S' Schieber skaliert werden

Wählen Sie positiv verlaufenden Impulse (0 bis 5 V) oder negativ verlaufenden Impulse (5 V auf 0 V).

Analoge Funktionsgenerator ('FUNCGEN')



Erzeugt analoge wellenform-Signale am angegebenen pin.

Minimum 'Period' ist 100 Mikrosekunden, skaliert über den 'I/O ____S'-Schieberegler der Symbolleiste (wenn mit einem 'S'-Suffix versehen).

Sinusförmige, dreieckige oder sägezahnförmige

Wellenformen.

Programmierbares 'I/O' Gerät ('PROGIO')



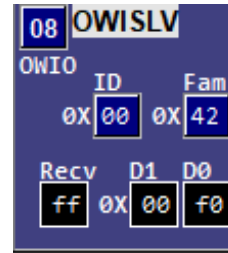
Ein nackter 'Uno' leiterplatte, den Sie programm (mit einem separaten programm) können, um einen 'I/O' gerät zu emulieren wessen Verhalten Sie vollständig definieren.

Dieser Sklave 'Uno' kann keine haben 'I/O' geräte für sich - Es kann nur tpo4 pins (IO1, IO2, IO3 und IO4) gemeinsam mit dem Master 'Uno' im Haupt-fenster nutzen LaborBankBereich .

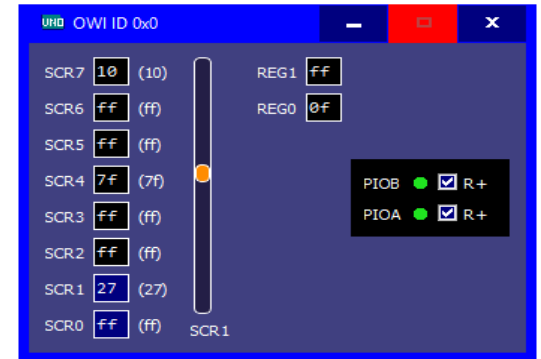
Rechtsklick (oder **Doppelklick**), um einen größeren fenster zu öffnen **CodeBereich** und **VariablenBereich** . Benutzen **Datei | Laden** Um einen neuen programm in diesen 'Uno'-Slave zu laden, bleibt sein ausführung immer mit dem des Master-'Uno' synchronisiert.

Nach dem Klicken in seine CodeBereich y Sie können sogar verwenden **Datei | Ausführen** zu **Schritt** oder **Ausführen Dort** oder **Ausführen Bis** in seinem Slave programm (der Master 'Uno' wird ausführen gerade genug, um synchron zu bleiben).

'1-Wire' Sklave ('OWISLV')

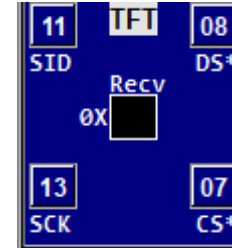


Doppelklick (oder Rechtsklick) öffnen **ein größerer fenster einstellen / ansehen** interne Register und parallele IO pins.



TFT Bildschirm ('TFT')

ein Adafruit™ Dünnfilmtransistor-LCD-Anzeige von 128-für-Pixel 160 angetrieben vom 'SPI'-Bus.

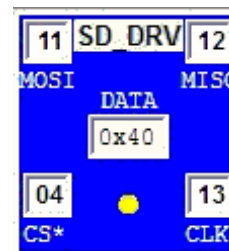


Die 'DS*' pin ist Daten / Befehl auswählen, und die 'CS*' pin ist die Active-Low-chip-auswählen, gibt es keine Zurücksetzen pin zur Verfügung gestellt, aber System Zurücksetzen setzt es ..

Doppelklick (Oder Rechtsklick) öffnen **eine größere fenster die eigentliche TFT-Display zu sehen**

SD-Laufwerk ('SD DRV')

Eine kleine 8-MByte-SD-Festplatte angetrieben von SPI Signale und



gespiegelt in eine '**SD' Unterverzeichnis** im Verzeichnis der **geladen programm** (ein '**SD'** Unterverzeichnis wird bei Abwesenheit angelegt).

Doppelklick (oder Rechtsklick) öffnen **ein größerer fenster sehen Verzeichnisse, Dateien, und Inhalt.**

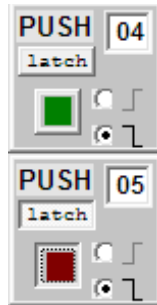
CS * niedrig zum Aktivieren.

Piezo Lautsprecher ('PIEZO')



Signale ein "hören" ein beliebiger 'Uno' pin.

Druckknopf ('PUSH')

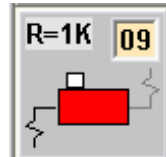


Ein Schließer **momentan** Taster auf + 5V oder Masse

Ein Schließer **einrasten** Taster auf + 5V oder Masse (Drücken Sie die "Latch" -Taste, um in diesen Modus zu gelangen.)

Sie können den Druckknopf schließen, indem Sie darauf klicken. oder durch Drücken einer beliebigen Tastaturtaste - Kontaktsprung wird nur erzeugt, wenn Sie die Taste verwenden **Leertaste** Schlüssel.

Schiebewiderstand ('R=1K')



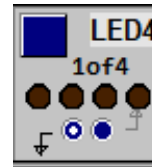
Ein 1 k-Ohm Pull-Up auf + 5V ODER ein 1 k-Ohm Pull-Down grundieren.

Farbige LED ('LED')



R, Y, G oder B LED verbunden zwischen Beliebiger gewählter 'Uno' pin und entweder geschliffen oder + 5V.

4-LED-Reihe ('LED4')



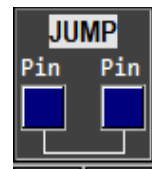
R, Y, G oder B Reihe von 4 LEDs zwischengeschaltetet **vier aufeinanderfolgende** 'Uno' pins und entweder geschliffen oder + 5V. Das mitgelieferte **1von4** pin Nummer entspricht der am weitesten links stehende LED.

7-Segment LED Ziffer ('7SEG')



Ein 7-LED_segment farbiges ziffer. Das mitgelieferte **1von4** pin Nummer repräsentiert die erste von **vier aufeinanderfolgende** 'Uno' pins. Die Aktiv-Hoch-Pegel auf diesen 4 pins definieren den hexadezimalen-Code für die gewünschte Anzeige ziffer ('0' bis 'F'), wobei die niedrigste pin-Nummer dem niedrigstwertigen Bit des hexadezimalen-Codes entspricht.

Pin Drahtbrücke ('JUMP')



Hier können Sie zwei 'Uno' pins miteinander verbinden, solange dadurch kein elektrischer konflikt entsteht.

In der Vollversion von Hilfe, datei, finden Sie Informationen zu möglichen Verwendungen für diesen gerät (die meisten davon betreffen Interrupts).





Analoger Schieberegler

Ein Schieberegler gesteuert Potentiometer. 0-5V bis antreiben beliebiger 'Uno' pin.



Menüs


Datei:

<u>Laden INO oder PDE Prog</u> 	Ermöglicht dem Benutzer die Auswahl eines programm datei mit der ausgewählten Erweiterung. Der programm ist sofort analysiert
<u>Editieren/Ansehen</u>	Öffnet den geladenen programm zum Anzeigen / Bearbeiten.
<u>Speichern</u> 	Speichern Der bearbeitete programm-Inhalt kehrt zum ursprünglichen programm-datei zurück.
<u>Speichern Als</u>	Speichern Der bearbeitete programm-Inhalt unter einem anderen datei-Namen.
<u>Nächster</u> <u>(' #include')</u> 	Bewegt den CodeBereich zur Anzeige des nächsten '#include' datei
<u>Bisherige</u> 	Kehrt zum vorherigen datei zurück
<u>Ausgang</u>	Beendet UnoArduSim.







Konfigurieren:

<u>'I/O' Geräte</u>	Wählen Sie die gewünschte Anzahl für jeden gerät-Typ (8 große und 16 kleine 'I/O' geräte sind zulässig).
<u>Präferenzen</u>	Wählen Sie automatische Einrückung, Schriftart schriftart, optional größere Schriftgröße, Expertensyntax, logische Schlüsselwortoperatoren, Erzwingen von array-Grenzen, Anzeige von herunterladen, 'Uno' leiterplatte-Version und TWI-Pufferlänge

Finden:

<u>Aufsteigen Anruf-Stapel</u> 	Zur vorherigen Anruferfunktion im Anruf-Stapel springen - der Variablenbereich passt sich dieser Funktion an
<u>Absteigen Anruf-Stapel</u> 	Zur nächsten aufgerufenen Funktion im Anruf-Stapel springen - der Variablenbereich passt sich dieser Funktion an
<u>Suche Text setzen</u> <u>(Strg-F)</u> 	Aktivieren Sie das Bearbeitungsfeld der Symbolleiste Finden, um den Text zu definieren, nach dem gesucht werden soll.
<u>Finden Nächster Text</u> 	Springe zum nächsten Textvorkommen im CodeBereich (wenn es den aktiven Fokus hat) oder zum nächsten Textvorkommen im VariablenBereich (wenn es stattdessen den aktiven Fokus hat).
<u>Finden Vorheriger Text</u> 	Zum vorherigen Textvorkommen im CodeBereich springen (wenn es den aktiven Fokus hat) oder zum vorherigen Textvorkommen im VariablenBereich springen (wenn es stattdessen den aktiven Fokus hat).

Ausführen:

<u>Schritt Hinein (F4)</u> 	Schritte ausführung um eine Anweisung vorwärts oder <i>in ein genanntes funktionsmodul</i> .
<u>Schritt Über (F5)</u> 	Schritte ausführung um eine Anweisung vorwärts oder <i>durch einen vollständigen funktionsmodul-Aufruf</i> .
<u>Schritt Aus (F6)</u> 	Vorschüsse ausführung von <i>gerade genug, um den aktuellen funktionsmodul zu verlassen</i> .
<u>Ausführen Dort (F7)</u> 	Läuft der programm, <i>Halt an der gewünschten programm-Linie</i> - Sie müssen zuerst auf eine gewünschte programm-Linie klicken, bevor Sie Ausführen Dort verwenden können.
<u>Ausführen Bis (F8)</u> 	Läuft der programm, <i>Anhalten, wenn der markierte Ort VariablenBereich variable das nächste Mal geschrieben wird</i> (Klicken Sie auf ein gewünschtes markieren Artikel vor der Verwendung von Run-Till).
<u>Ausführen (F9)</u> 	Läuft der programm.
<u>Halt (F10)</u> 	Stoppt programm ausführung (<i>und friert die Zeit ein</i>).
<u>Zurücksetzen</u> 	Setzt den programm zurück (alle Werte von variablen werden auf 0 zurückgesetzt, und alle Zeiger von variablen werden auf 0x0000 zurückgesetzt).
<u>Animieren</u>	Führt automatisch aufeinanderfolgende programm-Zeilen aus <i>mit zusätzlicher künstlicher Verzögerung</i> und Hervorheben der aktuellen Codezeile.
<u>Zeitlupe</u>	Verlangsamt die Zeit um den Faktor 10.

Optionen:

<u>Schritt Über Konstruktoren / Operatoren</u>	Fliegen Sie direkt durch Konstruktoren, Destruktoren und die Überlastung des Operators funktionsmodul während eines Schrittes (dh es stoppt nicht in diesen funktionsmodule).
<u>Register-Allocation-Modellierung</u>	Weisen Sie funktionsmodul-Locals den frei-ATmega-Registern zu und nicht dem Stack.
<u>Loop () Verzögerung hinzugefügt</u>	1 Millisekunde hinzufügen. (standardmäßig) bei jedem Anruf an <i>Schleife ()</i> (falls der Benutzer keine Verzögerungen hinzugefügt hat)
<u>Fehler bei Nicht initialisiert</u>	Als Analysieren-Fehler markieren, wenn Ihr programm versucht, einen variable zu verwenden, ohne zuvor seinen Wert initialisiert zu haben.
<u>Zeigen Sie Programm Herunterladen</u>	Zeigen Sie dem 'Uno' leiterplatte programm herunterladen (mit entsprechender Verzögerung).
<u>Verschachtelte Interrupts zulassen</u>	Ermöglichen Sie die erneute Aktivierung mit ' <i>interrupts . ()</i> ' von innerhalb einer Benutzerinterrupt-Serviceroutine.

VarAktualisieren:

<u>Erlaube Auto (-) Zusammenziehen</u>	Lassen Sie UnoArduSim zu, dass zusammenziehen erweitert arrays / structs / objekte anzeigt, wenn Sie in Echtzeit zurückfallen.
<u>Minimal</u>	Aktualisieren Sie die VariablenBereich-Anzeige nur viermal pro Sekunde.
<u>HighLight-Updates</u>	Markieren der zuletzt geänderte variable-Wert (kann zu einer Verlangsamung führen).

Hilfe Menübefehle:

<u>Schnell Hilfe Datei</u>	Öffnet das UnoArduSim_QuickHelp PDF datei.
<u>Volle Hilfe Datei</u>	Öffnet das UnoArduSim_FullHelp PDF datei.
<u>Fehler-Korrekturen</u>	Anzeigen wichtiger fehler-Korrekturen seit der vorherigen Version.
<u>Änderungen / Verbesserungen</u>	Zeigen Sie wichtige Änderungen und Verbesserungen seit der vorherigen Version an.
<u>Über</u>	Zeigt die Version und das Copyright an

Fenster:

<u>'Serial'-Monitor</u>	Fügen Sie ein seriellles IO gerät (falls keines vorhanden) hinzu und ziehen Sie einen größeren 'Serial'-Monitor-TX / RX-Text fenster hoch.
<u>Alles wiederherstellen</u>	Stelle alle minimierten Kinder fenster wieder her.
<u>Pin Digital Wellenformen</u>	Stelle einen minimierten Pin Digital Wellenformen fenster wieder her.
<u>Pin Analoge Wellenform</u>	Stelle einen minimierten Pin Analoge Wellenform fenster wieder her.