Simulator for Arduino

Table of contents

Introduction

Simulator for Arduino is an Arduino Simulator which may be downloaded or purchased from www.arduino.com.au. The Free version has a 1minute delay on opening sketches and is code limited to 100 lines. The Pro Version has no limitations.

The benefits and features of an Arduino Simulator are:

- The ability to teach and demonstrate the inner workings of an Arduino sketch
- Test out a sketch without the hardware, or prior to purchasing hardware
- Debug a sketch, step through or run with a breakpoint
- Speed up Arduino designs
- Demonstrate a project to a potential customer
- Find out number overruns such as setting a byte to 256, assigning a new value to a const variable, or use library routines without including the library

The SIMFORARDUINO Pro Version license for 10 users is designed for training groups with 10 students.

What's new

v0.97 03 August

1. Add CallStack window
2. Improve EEPROM dramatically - add save/load clear and fill
3. Add F3 Find to Sketch, variables and edit
4. Add Leonardo and Mega 75%
5. Add ON/OFF for objects (in call stack window)
6. Add inline Simulate(Uno) and Breakpoint commands
7. Add load/save language
8. fix a = b*c+d
9. add ?: and log10 operator
10. fix cos(a+b)+d
11. Add save and load hardware settings
12. Check all menu functions and fix Run F9
13. Remove back step
14. Fix break
15. Improve looping if/else/while

Free Version Differences

The Free Version is identical to the Pro Version, apart from the addition of a timer delay window and a Code Limit of 100 lines. The Free version is supplied as a demo version to allow users to trial the Simulator for Arduino program. The logic and operation of the Simulator for Arduino program is identical apart from the timer delay and code limit. In addition to the Pro version, upgrades are offered for a minimum of 1 month.

Simulator or Arduino

Simulator or Arduino is not an option. Simulator and Arduino or even Arduino and Simulator reads better for us. The Simulator is an add-on product to the Arduino development kit and has been designed as a training/ demo/debugging tool but not as a development or compiling tool. We fully support the Arduino project and encourage new users to buy and try a real Arduino first.

To develop code for Arduino, we recommend using the Arduino Uno together with the Arduino IDE. This Simulator has not been designed to check for syntax errors, and we recommend only using the Simulator for debugging when there is a difficult problem with an Arduino sketch.

There are many Arduino clones around, and while these may be sometimes slightly cheaper, we recommend using original Arduino kits since these work better than the clones and support for the Arduino project leads to more cool new development boards.

Getting Started

Getting Started.

Download the setup.exe file. When finished, run this program to install the Simulator for Arduino program.

System requirements

The System Requirements are a PC Pentium II or later.

The Simulator for Arduino program will run on the following systems:

WINDOWS

- Windows 2000
- Windows XP
- Windows 7

MAC

- Windows using Parallels
- Windows using VMWare VM systems

LINUX

- Wine on Linux (windows emulator)

Untested on Windows Vista,98,ME and 95. If the Simulator does run on any of these systems, please let us know by sending an email to arduinoshields@gmail.com

Getting help

For help, please email arduinoshields@gmail.com and please attach the sketch.

Before emailing, please check that the sketch compiles in the Arduino IDE first.

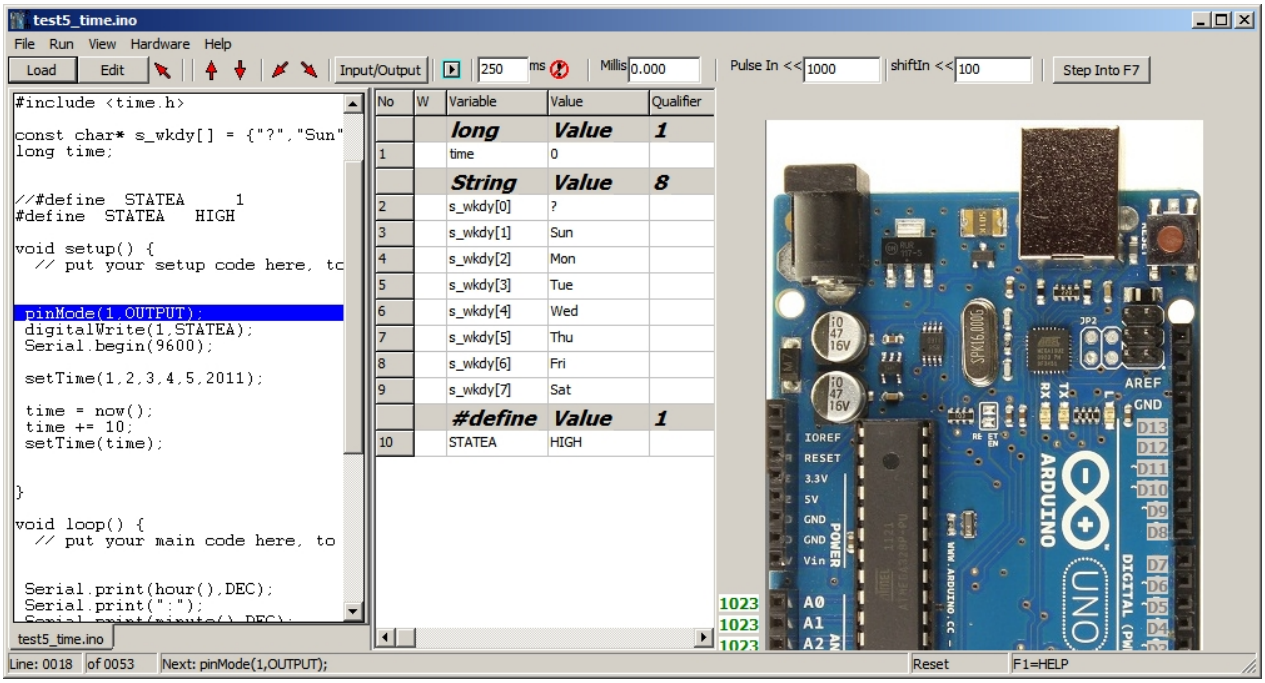
Using the IDE

The Simulator for Arduino IDE has several sections. These are:

- The Menu system which contains options and settings - refer to the Menu System for more info
- The Shortcut Toolbar provides single click action for commonly used operations
- The Program list box
- The Variables area
- The Arduino picture

Note that the order of the Program window, Variables are and Arduino picture may be swapped by Selecting View|Arduino on Left. The preferred method is to have the Program Window on the left so that the work flow is input,data and output.

When the Simulator for Arduino IDE is resized, the Variables window will disappear first since this is the least important window. If the main screen is made smaller, the Program window will then disappear and the Arduino Screen will always be on top. This allows for sketches to be simulated as they would appear for real.



Program Window

The Program window area shows the program listing, and the cursor will always be in the vertical center of the screen. The Program Window allows for the program to be stepped through. When selected, the F8 key may be used to step through the program.

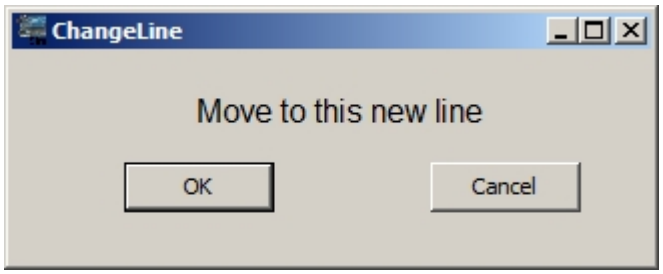
```
// Saved by Simulator for Arduino V0.92
// Simulate Uno
This is a test /* Text */
/* Text */ This is a test
This is a test /* Te
xt */ This is a test
xt */ This is a test /* Te */
int y = 23 ;

char x = 1;
int y = 23 ;
long fd = 34;
String r ="1234";
#define dghs 3456

void setup() {
 pinMode(13,OUTPUT) ;
 pinMode(12,INPUT) ; digitalWrite(12,HIGH) ;
 Serial.begin(9600) ;
}

void loop() {
 y = digitalRead(12) ;
 Serial.print(y) ;
 digitalWrite(13,y) ;
 if (y==HIGH) Serial.println("Hi") ;
 if (y==LOW)  Serial.println("Lo") ;
 delay(500) ;
}
```

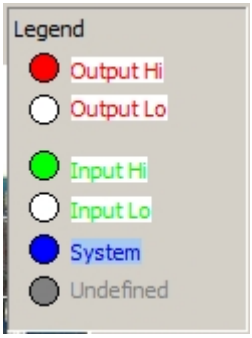
A new line may be selected by click anywhere in the program window. If the *Ask on Line Change* option is Checked, then a Dialog will ask for confirmation that the program should jump to the new Line.



The Statusbar shows the Line number, Last Line number and the next code to be executed. If the AutoStep time is set to less than 10ms, the Status-Bar will not update until AutoStep is stopped.

The first line may be used to define the Arduino board type using the // Simulate Uno or // Simulate Mega command in the first line. The Simulator for Arduino program will look for these commands inside the comments on the first line only. Using commands inside comments allows the Arduino IDE to still compile the sketch without any changes. The UnoR3 and MegaR3 pictures are included in the same directory as the Simulator for Arduino program.

Legend



The Legend may be turned on by selecting View | Legend

The Legend shows the colours and states of the Digital output. The AnalogWrite value of a digital pin will be displayed to the left of the digital pin when the analogWrite() routine is used.

BreakPoint

A breakpoint can be set by right clicking anywhere in the program window.

The pop-up menu has one option which can be clicked to add or clear the breakpoint.

If a BreakPoint is added, the breakpoint line will be shown in red. This is commonly used with the AutoStep or F9 command. The breakpoint can also be saved inside a comment using the syntax // breakpoint(20) - this will set a breakpoint on line 20

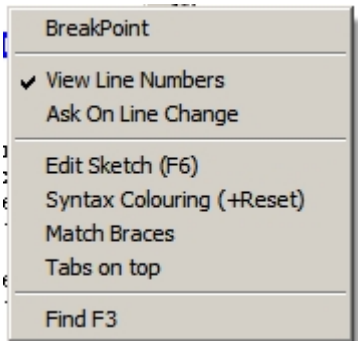
```
// Saved by Simulator for Arduino V0.92
// Simulate Uno
This is a test /* Text */
/* Text */ This is a test
This is a test /* Te
xt */ This is a test
xt */ This is a test /* Te */
int y = 23 ;

char x = 1;
int y = 23 ;
long fd = 34;
String r ="1234";
#define dghs 3456

void setup() {
 pinMode(13,OUTPUT) ;
 pinMode(12,INPUT) ; digitalWrite(12,HIGH) ;
 Serial.begin(9600) ;
}

void loop() {
 y = digitalRead(12) ;
 Serial.print(y) ;
 digitalWrite(13,y) ;
 if (y==HIGH) Serial.println("Hi") ;
 if (y==LOW)  Serial.println("Lo") ;
 delay(500) ;
}
```

Program Popup Menu



The Program Area Popup menu has four options:

- Breakpoint - right click to add a breakpoint in red
- Clear Breakpoint - clear all breakpoints

- Line Numbers - turn on the line numbers in the Program Window - this is useful for checking subroutines
- Ask on Line Change - if a new line is clicked in the program window, this option allows for a confirmation window before changing the line number

- Edit Sketch - allows the sketch to be edited in the Edit Window
- Syntax Colouring - show comments as light gray, #if hash commands in medium grey, definitions in green, constants in Olive and subroutines as blue
- Match Braces - find the opening or closing bracket for the current line
- Tabs On Top - places the tab with the filename at the top if checked or else the bottom
- Find - open the Find window to find text inside the sketch

Bare Minimum

Bare Minimum will load the bare minimum needed for a new sketch. Select Run|Edit Sketch and the press the rightmost button to load the Bare Minimum to start writing a sketch.

Loading a Sketch

To load a sketch, select File|Load Sketch, press the Load Sketch button or press F4. Navigate to the correct folder and select the *.ino file (or *.pde for previous Arduino versions).

Running a Sketch

After a sketch has been loaded, the first executable line of code will be highlighted. Now press the Step Into button or the down arrow. Alternately, the F7 or F8 button may be pressed. The F9 button allows the program to be autorun with a 250ms interval between steps. This interval can be reduced to suit.

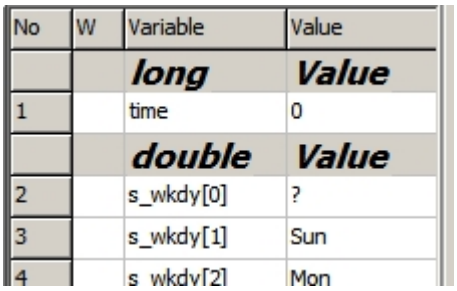
The middle section of the screen displays the variables grid, and these are organised into Bytes, Chars, Int (eger)s, Long, Strings and Defines. Any value in the grid can be changed by selecting it and typing in a new number.

Variables Area

The Variables area shows the status of all the variables used in program and their current value. The Value may be changed by clicking in the relevant cell and changing the value.

The Qualifier column typically shows if the value is a const, unsigned, static or volatile type. The qualifier

number in a heading row (shown by the gray shading) shows how many variables of that type have been defined.



No	W	Variable	Value
		long	***Value***
1		time	0
		double	***Value***
2		s_wkdy[0]	?
3		s_wkdy[1]	Sun
4		s_wkdy[2]	Mon

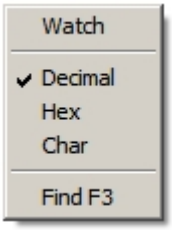
When returning from a subroutine, the stack will be returned the the original state, and any variables defined in the subroutine or passed as arguments will be cleaned or removed.

Right Click in the Variables area to bring up the Pop-up Menu.

The second column is the watch column, and clicking in this column will show a w. The Pop-up menu can then be selected to show only Watched variables. This is useful when only a few variables are relevant amongst several hundred. The gray area in the W column may be click to show or unshow all the variables for that type.

Arrays can be folded by clicking in the second or Watch column and a + sign will show that the Array has been folded. When folded, only the first element of each array will be shown. Alternately, the array may be folded at any element of the array and in this case, the elements above will be shown, but the elements below will be hidden. Click the + again to unfold the array

Variables Popup Menu



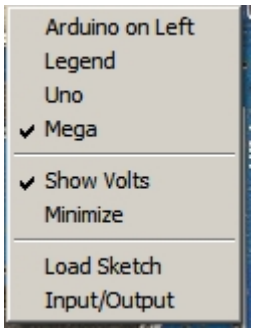
The Variables Area Popup menu has four options:

- Watch - only display variables which have a w or + in the second column of the grid
- Decimal - display Variables value in decimal
- Hex - display Variables value in hexadecimal
- Char - display Variables value in Character format and hexadecimal
- Find - open the Find window to find text inside the variables area

Arduino Area

The Arduino picture area allows for the digital inputs and outputs to be displayed along with the Analog input and output values. The digital pins are defined up to D53.

Arduino Popup Menu



The Arduino Area Popup menu has four options:

- Arduino on Left - display the picture of the Arduino on the left and the Program window on the right
- Legend - display the digital pin legend on the top right of the Arduino picture
- Uno - set the Arduino board to a Uno type
- Mega - set the Arduino board to a Mega type

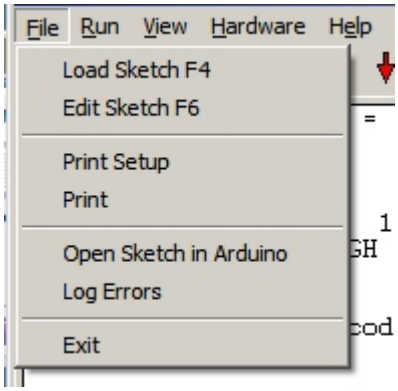
- Show Volts - this option allows for the analog number to be displayed as volts in the format x.xx (accurate to 0.01V)
- Minimize - turn off the Tool bar and menu to make the Arduino picture as minimal as possible. the program can be resized to then only show the Arduino picture and the Title (which shows the sketch name).

- Load Sketch - load a sketch into the program window - useful in minimize mode
- Input/Output - show the input/output window - useful in minimize mode

Menu System

The Menu system has five sub-menus. These menus allow the Simulator to be customised.

File Menu



The File Menu Items are:

- Load Sketch F4 - Load a Sketch into the Program Window
- Edit Sketch F6 - Edit a Sketch in the Edit Window

- Print Setup - open the Printer Setup Dialog Box
- Print - print a scaled picture of the program

- Open Sketch in Arduino - this opens the sketch in the Arduino IDE assuming Arduino is the default program to open sketches

- Log Errors - all errors will be logged to the ErrorLog + date.txt file Note that Errors here mean program and simulator errors

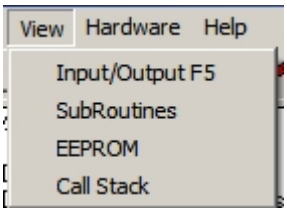
- Exit - close the program

Run Menu

The Run Menu Items are:

- Reset - reset the program to the start or Setup routine
- Step Into - step one line through the program, and step into any subroutines
- Step Over - step one line and step over any subroutines
- Step Out of - Step out of a subroutine
- Run - perform a step once every time period which is 250ms by default -see Shortcut toolbar

View Menu

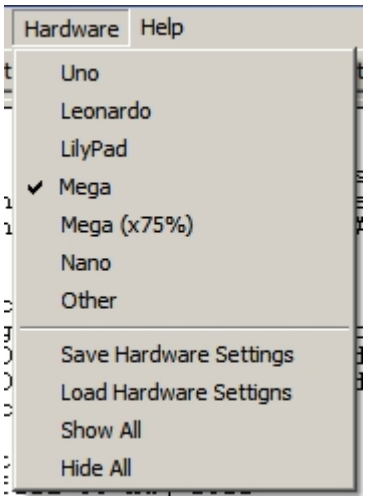


The View Menu Items are:

- Simulation data - open or Close the Simulation input/output window
- Subroutines - View the Subroutines and Line Numbers
- EEPROM - View the EEPROM data
- Open the Call Stack Window

Hardware

- Save Hardware Settings

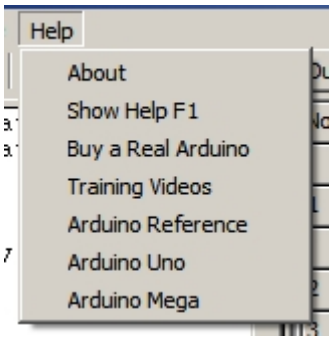


The Hardware Menu Items are:

- Uno - display the Arduino picture as an Arduino Uno board
- Leonardo- display the Arduino picture as an Arduino Leonardo board
- LilyPad - display the Arduino picture as an Arduino LilyPad board
- Mega - display the Arduino picture as an Arduino Mega board
- Mega (x75%) - a smaller version for laptops
- Nano - display the Arduino picture as an Arduino Nano board
- Other - open up any picture file preferably 350 pixels wide to overlay on the Arduino picture

- Save Hardware Settings - save hardware settings to a .txt file which can be edited later
- Save Hardware Settings - load hardware settings from a .txt file
- Show All - Show all the pins, pin names and analog values for the Arduino board
- Hide All - Hide all the pins, pin names and analog values for the Arduino board

Help



The Help Menu items are :

- About - Show the Program version
- Show Help F1 - show this help - Note: this can also be activated by pressing F1
- Buy a real Arduino - open the Buy Arduino page
- Training Videos - watch all the Youtube videos related to the Simulator and Shields
- Arduino Reference - display the Arduino language reference page
- Arduino Uno - display the Arduino Uno Page
- Arduino Mega - display the Arduino Mega Page

The Help | About screen shows the current software version.

The F (for Free version) after the version number indicates the Simulator is identical to the Pro Version but has a delay timer window any time a new sketch is loaded or edited. The Free version has been provided as demo version to allow the Simulator for Arduino program to be evaluated prior to purchase. It is anticipated that after a week of use, it will be more economically viable to purchase the Pro Version than continue to run the Free Version.

Another limitation is a Code limit of 100 lines which will be displayed on the Delay Timer screen.

Dialog Windows

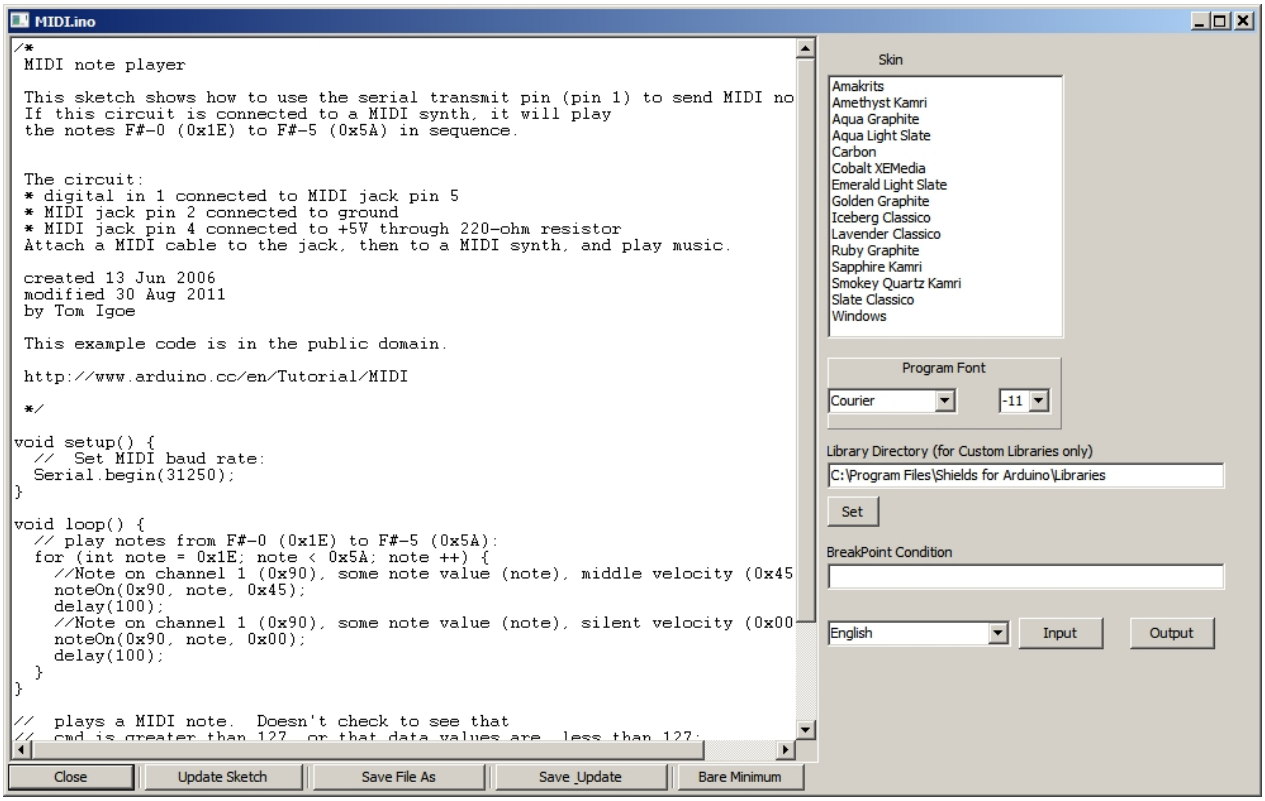
The Simulator has several dialog windows which can display more information when activated. These are:

- The Edit Window
- The Input/Output Window
- View Subroutines
- Error Message

Edit Window

The Edit Sketch Window allows for sketches to be edited and updated or reloaded into the Program window

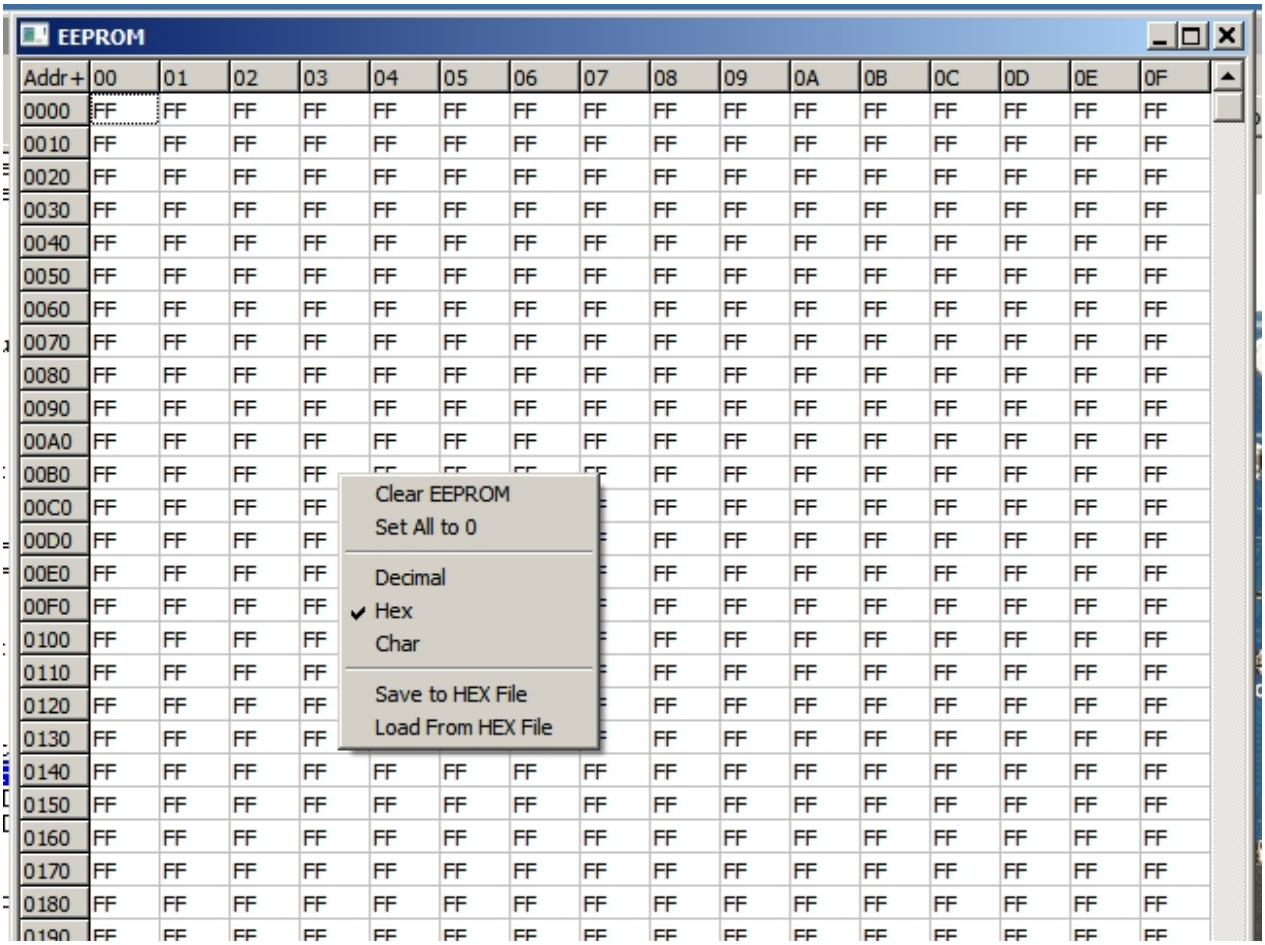
- Close - closes this window and resumes simulating the program
- Update Sketch - save this edited sketch back to the program window
- Save File As - save this sketch to a new .ino file
- SaveUpdate - save the sketch under the same name and update the sketch in the program window
- Bare Minimum - load the bar minimum code skeleton - see here for more info
- Font type - changes the font type of the main program window
- Font size - changes the font size of the main program window
- Skins - this will change and save the skin for the whole Simulator program. The default skin is Windows.
- Library Directory - set for using custom libraries
- BreakPoint condition - set to break on a certain condition
- Language - change between English, French, Italian and Other for titles only - also Input and output the language text



EEPROM Window

The EEPROM window allows for viewing and modifying of the EEPROM memory. Right click to bring up the popup menu which has several options:

- Clear EEPROM - set all EEPROM memory to FF
- Set All to 0 - all EEPROM memory to 00
- Decimal - display Variables value in decimal
- Hex - display Variables value in hexadecimal
- Char - display Variables value in Character format and hexadecimal
- Save to Hex File - save the EEPROM memory to a hex file
- Load from Hex File - load the EEPROM memory from a hex file

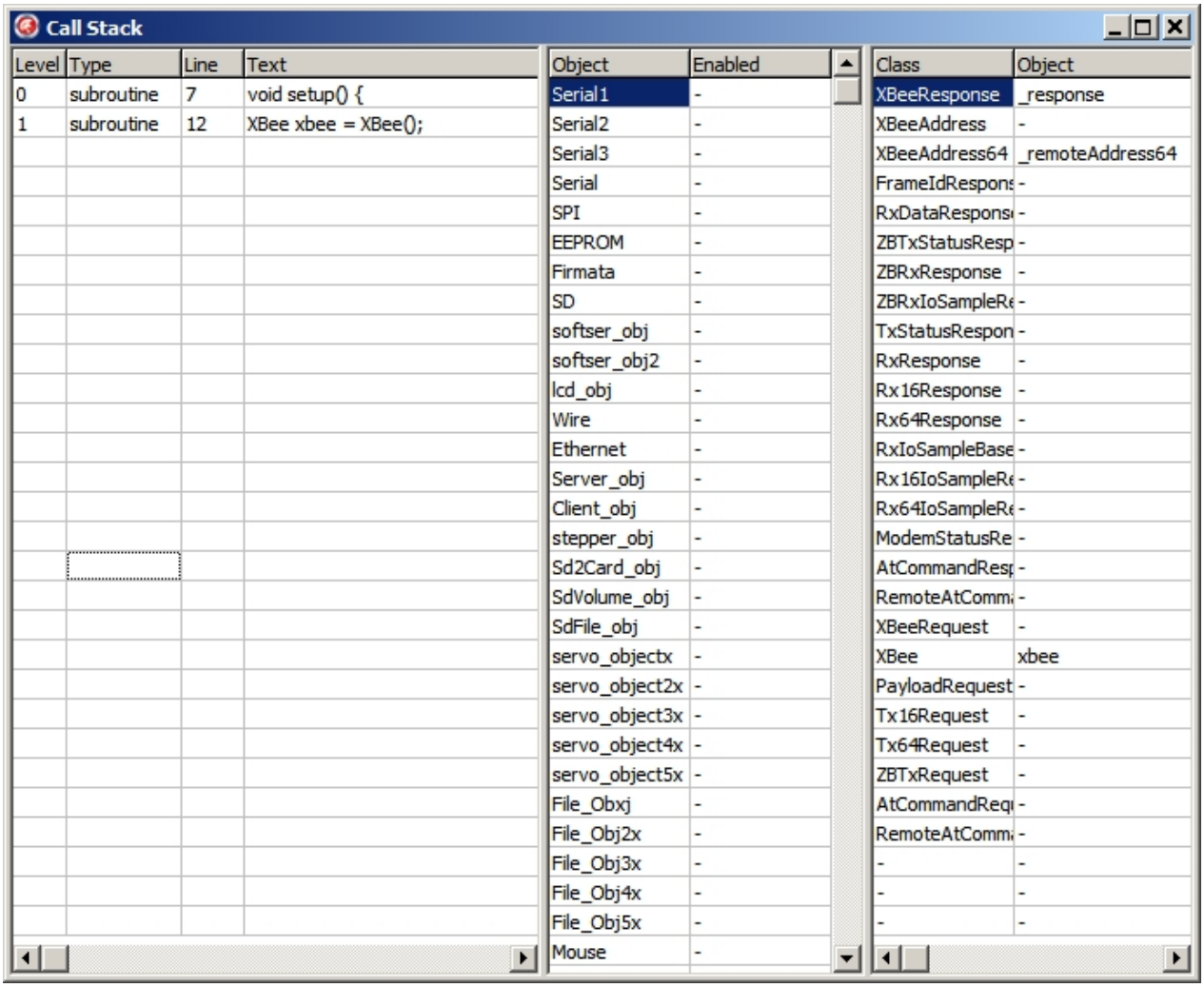


Call Stack

The Call Stack displays the different levels of sketch loops in the left grid

The middle grid shows the Arduino objects, names and Enabled status.

The right grid shows classes found and objects derived from these classes.



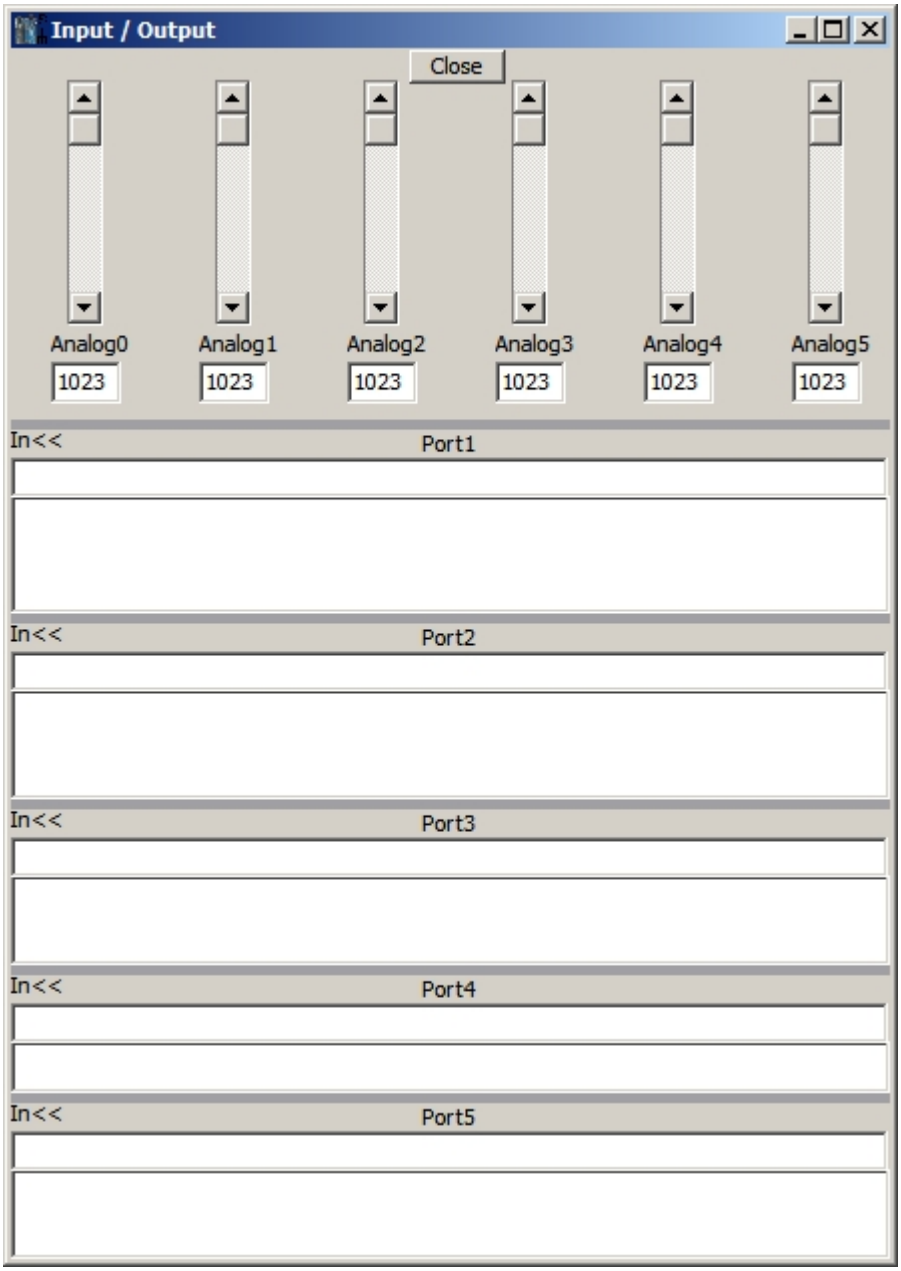
Input/Output Window

The Simulation Window allows for data to be simulated for the six analog inputs for the Uno or 16 analog inputs for the Mega. The window may be resized to be smaller, and the scrollbars and text boxes will automatically resize to suit.

The Analog Value may be changed by moving the respective scrollbar, or entering a new value in the white Edit box.

Input and Output data is displayed in five ports. Each port has its own line for input data and a box for output data. Each Port can be expanded or contracted to suit the display and resized.

The ports are assigned in the order they are setup.



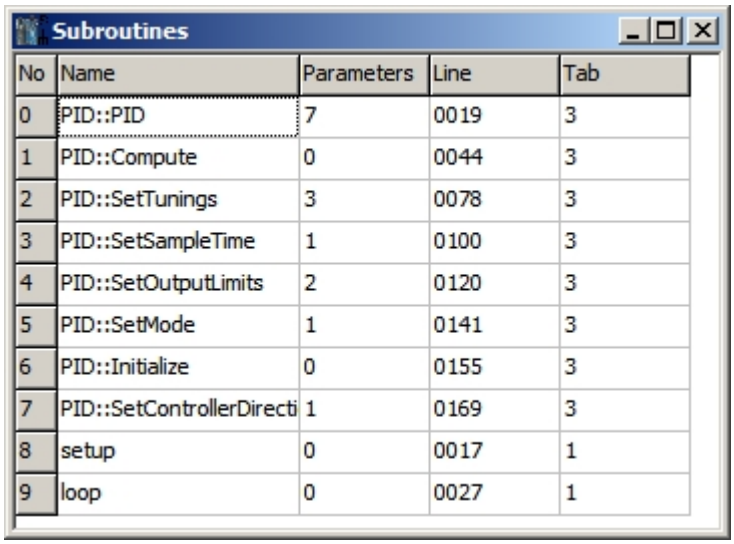
Simulation Popup Menu

The Simulation Window Popup Menu has two items:

- Show All Ports - to view the five ports and reset the Splitter bars if needed
- Hide Unused Ports - close all ports except the opened ports

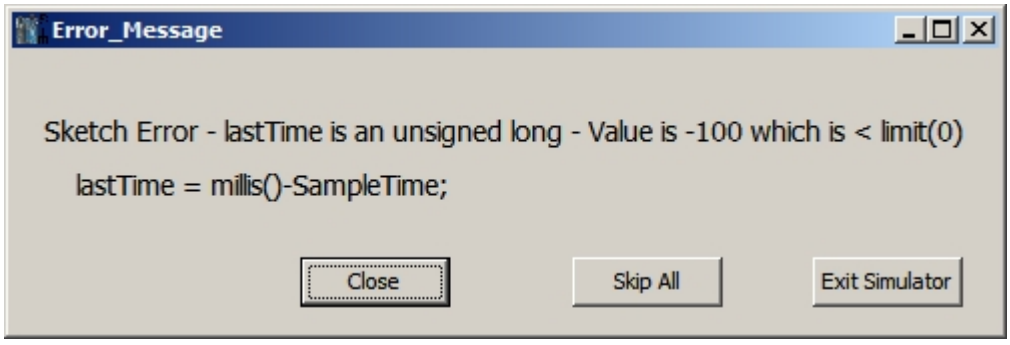
View Subroutines

The View Subroutines window allows for the Subroutines to be viewed along with the number of parameters, the line number and the tab. The line numbers may be viewed in the program window by selecting Line Numbers



No	Name	Parameters	Line	Tab
0	PID::PID	7	0019	3
1	PID::Compute	0	0044	3
2	PID::SetTunings	3	0078	3
3	PID::SetSampleTime	1	0100	3
4	PID::SetOutputLimits	2	0120	3
5	PID::SetMode	1	0141	3
6	PID::Initialize	0	0155	3
7	PID::SetControllerDirecti	1	0169	3
8	setup	0	0017	1
9	loop	0	0027	1

Error Message



The Error message screen displays any code which the Simulator cannot process or which the Simulator decides if an illegal operation such as:

- the above operation, outside the setup() and loop() routines
- setting a digital pin which is set to input
- changing a const(ant) value
- setting a variable to a value outside its limits
- addressing an array element before the start or past the end of the array

The File|Log Errors option allows these errors to be logged to an ErrorLog +date.txt file.

The Email button will only be shown if the Error could be related to internal Simulator logic. If the Error message starts with Sketch Error, the Simulator has guessed that it is more likely the error is in the Sketch. If this is not the case, please email the sketch to arduinoshields@gmail.com

Shortcut Toolbar

The Shortcut Toolbar provides single click actions for commonly used operations.

Button Functions:

- Load - loads a sketch from the last used folder into the Simulator
- Edit - Edit a sketch
- Left up arrow - Reset
- Up Arrow - go back without executing any code

- Down Arrow - Step Over - single step jumping over any subroutines
- Left Down Arrow - Step out of subroutines excluding setup and loop
- Right Down Arrow - Single Step and Step into any subroutines
- Simulation - show or close the Simulation input/output window
- Run - AutoStep the program one instruction every xx ms The ms value can be edited. This time is the delay between line execution and is not the actual speed of the program since each line may take 1-100ms to execute. If the AutoStep time period is set to faster than 10ms, the statusbar will not update to increase speed.
- Abort - abort if the program is stuck in any loop
- Millis - the run time in milliseconds. This value can be changed at any time. Each program step takes 1us which is an approximation
- PulseIn - this value is the result of the PulseIn function and can be changed to suit
- Step Into (F7) - this button allows for the program to be single stepped

The Milli seconds variable is displayed in a text box. This value can be adjusted by typing in a new value. Each line of program execution takes 1us or 0.001. The commands Delay and delayMicroseconds will add the accurate number of milliseconds or microseconds to the Millis Text box.

PulseIn provides the return data for the pulseIn() command. This value must be a number.

Shortcut keys

Several keyboard shortcuts have been added. These are

- F2 for Reset
- F3 for Find
- F4 for Load a sketch
- F5 to open the Simulation Window
- F6 - Edit a sketch in the Edit Window
- F7 to Step Into
- F8 to Step Over
- Shift F8 to Step out of a subroutine
- F9 to Run or AutoStep
- F10 find (to be removed later)

Example Sketches

Several sketches have been included for testing the Simulator. These are the Official Arduino examples in the folders 01 Basics to 19 Wire.

In a typical install, the sample sketches will be found in this folder:
C:\Program Files\Shields for Arduino

Test Sketches

The test sketches are made up of issues with the Simulator and are used to do quick checks on patch versions of the Simulator. The major releases of the Simulator are fully tested against all the sample sketches.

Language Processing

Keywords

The following words are defined as keywords and should not be redefined in a sketch:

defined(ARDUINO)	1
defined(__AVR__)	1
defined(__AVR_ATMEGA168__)	1 for Uno or 0 for Mega
defined(NUM_ANALOG_INPUTS)	6 for Uno or 16 for Mega

FALSE	0
TRUE	1
LOW	0
HIGH	1
ARDUINO	100
TOTAL_ANALOG_PINSX	
TOTAL_PORTS	14
INPUT	0
OUTPUT	1
TOTAL_PINSX	20
SD_CARD_TYPE_SD1	1
SD_CARD_TYPE_SD2	2
SD_CARD_TYPE_SD3	3
PI	3.14159265358979323846264338327 95
HALF_PI	1.57079632679489661923132169163 9
TWO_PI	6.28318530717958647692528676655 9
DEG_TO_RAD	0.01745329251994329576923690768 4886
RAD_TO_DEG	57.29577951308232087679815481410 5
NUM_ANALOG_INPUTS	6 for Uno or 16 for Mega

PINA-L (no PINI)
DDRA-L (no DDRI)
PORTA-L (no PORTI)

Custom Libraries

The Simulator now has the ability to load custom libraries and partially use them.

To use a custom library, there are two methods. The first is to copy the header file and cpp file to the same directory as the sketch, and then the Simulator will find these and load them in separate tabs.

The second method which is the preferred method is to setup the library directory and then the Simulator will automatically find the library header and source file and load them in separate tabs in the Program Window. This method is better since the Simulator will find the files in the same place as the Arduino IDE, and will save the effort of copying the files. The Library Directory can be setup by pressing F6 to load the Edit Sketch page and at the right hand side is an Edit Box which allows the Library directory to be setup.

Registry Settings

Most settings are saved in the registry under the key
SOFTWARE\SimForArduino x.xx where x.xx is the version number

The Registry settings can be viewed by running the RegEdit program or a similar utility.

Version Info

v0.97 03 August
1. Add CallStack window
2. Improve EEPROM dramatically - add save/load clear and fill
3. Add F3 Find to Sketch, variables and edit
4. Add Leonardo and Mega 75%
5. Add ON/OFF for objects (in call stack window)
6. Add inline Simulate(Uno) and Breakpoint commands
7. Add load/save language
8. fix a = b*c+d
9. add ?: and log10 operator
10. fix cos(a+b)+d
11. Add save and load hardware settings
12. Check all menu functions and fix Run F9
13. Remove back step
14. Fix break
15. Improve looping if/else/while

v0.96 22 April
1. Make Input/Output ports open as 1,2,3,4,5 so LCD, SPI and I2C can be used together
2. Start implement custom libraries - add four test samples sketchs
3. Add breakpoint condition and library directory to Edit Sketch screen
4. Add two more fonts
5. Add divide by zero check
6. Add recursive checks and max include checks
7. Stop input.output window resizing when reset is click
8. Improve minimize mode and add lights to LEDs
9. Add finger to reset button, pulse to crystal and USB plug
10. Add Structure, two dim arrays (without = yet) and strcmp
11. Add class and abject processing for up to 10 objects
12. Convert /t and .n to tab and linefeeds (tabs do not quite line up yet)
13. Add PIN, PORT and DDR processing
14 Add many more keywords
15. Remove Option Delay and Tone

v0.95 27th Feb
1. Run/Stop button changes state when Abort is pressed or error is found
2. Add syntax colouring where possible
3. Improved the Uno/Mega selection - Uno picture will only shows pins 0-13
4. Added a font selection for the program listbox
5. BUG - Fixed reset issue with Line Numbers on - caused by attempted parse of commented line
6. String addition not recognised - "This" + "That" prints as is
7. LCD Scroll goes past characters allowed
8. Match Brace option Added
9. Add Windows Themes - still to test if it works on a separate PC
10. Improve variable area to have individual +- for folding arrays
11. Improve error processing so sketch errors are different from possible Simulator errors
12. if/While loops need to check for commented lines
13. Include file directory sometimes loses the directory (possible issue)
14. Increase digital pins to 53
15. Add a button area to the reset switch and crystal

v0.94 28th January 2012
v0.93 6th January 2012
v0.92 - 30 December 2011
v0.91 - 16 December 2011
v0.80 - v0.90 1-Dec to 16-Dec-11 first test releases

Credits

This project has been a collaboration with many people.

Credit and thanks must go to these people for all their help: Marco Stuurman, Zeljko Frankovic, Pete Lunt, Mark Grass Sr, Serge Desjardins, Peter Brouggy, Halam Rose, Larry Vatland, Raimondas Butauskas, Jason Snow, J-Pierre Romanogli, Hamilton Elliott, Graeme Caie, Michael Moore, Filipe Oliveira, Leon Rozengarten, Robert Lopez,Mauro Abbattista,Todd Radack,Alain Herben, George Vrynios, Neko San, Mauro Abbattista, Alain Herben, Victor Aguilar, David Williams, Janis Gkatzaras, Rodrigo Amaya, Ed Ross, David Cox, Shane Stenton, Freddie Snijman, Andres j. Ogayar, Anxionnaz Yannick, Jeandaniel Planterose, Donald Dempsey, Enrique Condes, Peter Brown, Mladen Bruck, Jesse Carneiro