12Blocks

12blocks is a simple drag and drop style IDE for the Arduino, and many other hardware platforms.
While providing a simple easy to use interface, it incorporates many other functions not found in any other programmers.

The program uses standard Arduino boards with no modification to the bootloader, but provides debugging in real-time, along with the ability to modify variables on the fly.

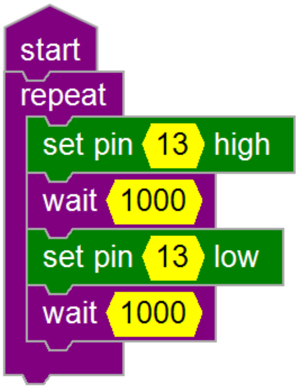
It also features multiple instances, or put simply running more than one sketch at the same time.

Simplicity

Take the basic 'Blink' example

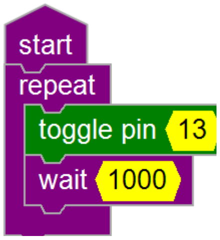
```
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most
  // Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Blink code



```
start
repeat
  set pin 13 high
  wait 1000
  set pin 13 low
  wait 1000
```

12Block Blink



```
start
repeat
  toggle pin 13
  wait 1000
```

12Block Toggle

The standard Arduino IDE doesn't have a 'toggle' function, however 12blocks does, and it makes it so much simpler.

Variables

To change the timing of the blink program, you modify the delay(1000) to say delay(500) for both, press upload and wait while it uploads, and then it will run at the new flash rate.
For 12Blocks simply click in the yellow part of the 'wait' box and type 500 press enter and it changes, repeat for the second wait and that's it.

Debug

For the Arduino IDE debugging or passing the values of the variables requires adding a serial statement, and placing Serialprint statements with the variable you are interested in, then after uploading you need to open up the 'Serial Monitor' from the menu. (why they didn't incorporate an auto opening function like Picaxe, will forever be a mystery)

If you want to make a change, then you need to change, upload, and re-open the terminal.
It does get a little tiresome if you have a few tweaks to make.

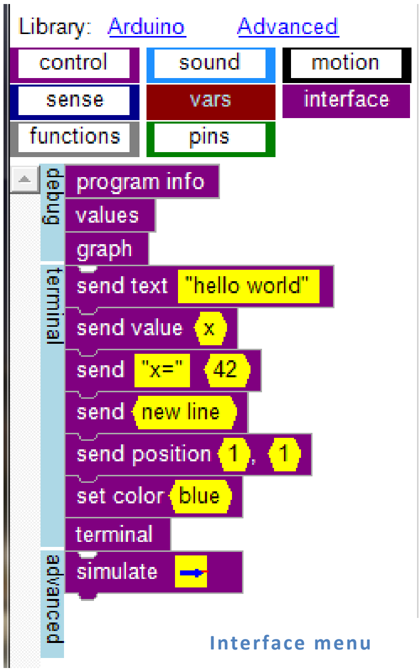
The other issue with this, is there is no means to manipulate where the content appears. It just keeps scrolling down continuously, with no ability to have a stationary display.

While this isn't so bad for some situations, it is hard to monitor easily.

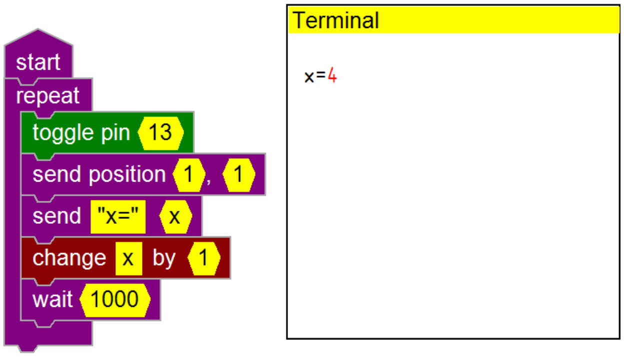
```
int x =0;

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most
Arduino boards:
  pinMode(13, OUTPUT);
  Serial.begin(57600);

}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  Serial.println (x,DEC);   // print the value of x
  delay(1000);              // wait for a second
  x = x + 1;                // increment by 1
}
```

Blink with debug

In 12Blocks you have a couple of options
You can use the ‘Terminal’ block and with the ‘send position’ block keeps the data in the same spot.



Interface menu



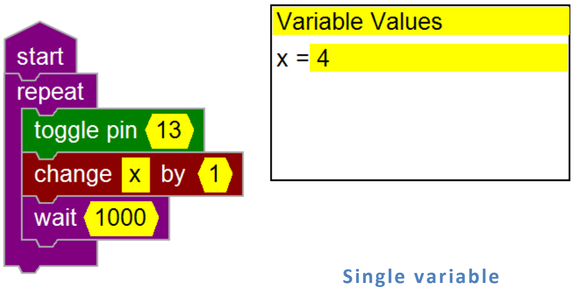
```
start
repeat
  toggle pin 13
  send position 1 , 1
  send "x=" x
  change x by 1
  wait 1000
```

```
Terminal

x=4
```

Clicking in the yellow area of a block allows the content to be modified.

Terminal window

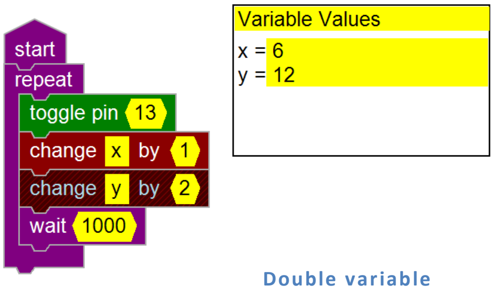
You can also use the ‘values’ block to have a display of the variables automatically increment.



```
start
repeat
  toggle pin 13
  change x by 1
  wait 1000
```

```
Variable Values
x = 4
```

Single variable

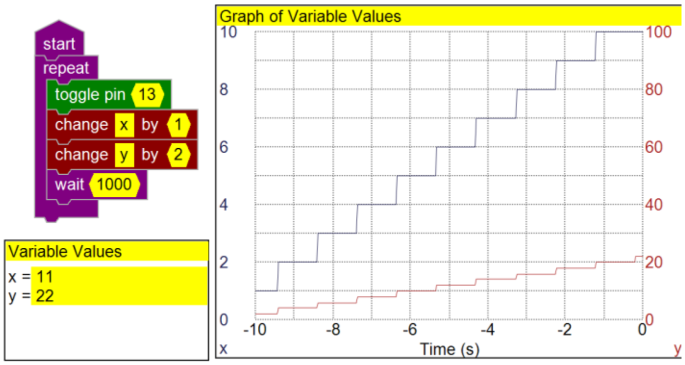


```
start
repeat
  toggle pin 13
  change x by 1
  change y by 2
  wait 1000
```

```
Variable Values
x = 6
y = 12
```

Double variable

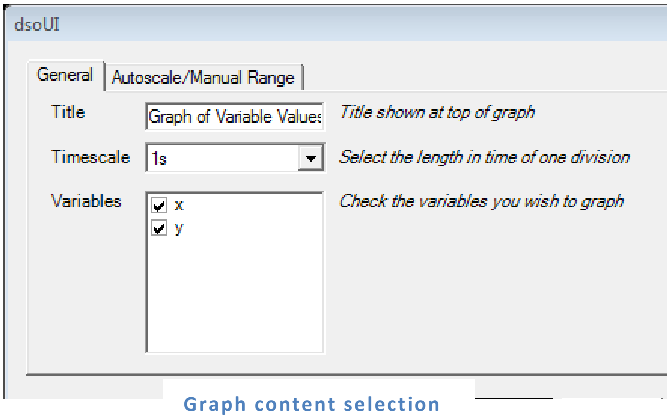
The third option is to use the ‘graph’ block to display the values.



```
start
repeat
  toggle pin 13
  change x by 1
  change y by 2
  wait 1000
```

```
Variable Values
x = 11
y = 22
```

Double variable graphing



Title: Graph of Variable Values — *Title shown at top of graph*

Timescale: 1s — *Select the length in time of one division*

Variables: ☑ x ☑ y — *Check the variables you wish to graph*

Graph content selection

Clicking on the graph window allows the variables to be selected, along with the timescale.
The Autoscale/Manual Range tab sets which variable to use and the option to use autoscale or set the lower and upper limits.
You can even change the Title!

Program stepping

One of the other things missing from Arduino IDE is the ability to control the program by stopping and manually stepping each line of code.
Often I have struggled to find where the program was jumping to, due to a mistake somewhere in my code.

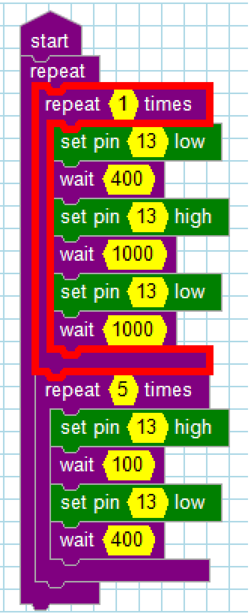
12Blocks allows this function and is as simple as setting the breakpoint (where it stops for you) and then stepping using the F6 function key or mouse.

In this example I have two loops that flash the Led as an error code.
The first loop turns the led on for 1 sec, then off for 1 sec (Long pulse), and then flashes it 5 times.

The sketch is currently stopped on the repeat for the long pulse, and the next step (using F6) will make the led low.

The keyboard is being scanned, so you can hold down the F6, or press for each step.
Any delays will be affected by how you advance the sketch, so keep this in mind if the timing is critical.

For The Shed readers, this code is something I used for error codes, or to give feedback to the user for the mode they are in.



```
start
repeat
  repeat 1 times
    set pin 13 low
    wait 400
    set pin 13 high
    wait 1000
    set pin 13 low
    wait 1000
  repeat 5 times
    set pin 13 high
    wait 100
    set pin 13 low
    wait 400
```

two loops