

The image shows two Arduino IDE windows side-by-side. The left window is titled "sketch\_jun23b | Arduino 1.8.13" and contains a blank sketch with the following code:

```
1 void setup() {  
2     // put your setup code here, to run once:  
3 }  
4  
5 void loop() {  
6     // put your main code here, to run repeatedly:  
7 }  
8 }
```

The right window is titled "Blink | Arduino 1.8.13" and contains the standard Blink example sketch. The code is as follows:

```
1 /*  
2  * Blink  
3  *  
4  * Turns an LED on for one second, then off for one second, repeatedly.  
5  *  
6  * Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO  
7  * it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to  
8  * the correct LED pin independent of which board is used.  
9  * If you want to know what pin the on-board LED is connected to on your Arduino  
10 * model, check the Technical Specs of your board at:  
11 * https://www.arduino.cc/en/Main/Products  
12 *  
13 * modified 8 May 2014  
14 * by Scott Fitzgerald  
15 * modified 2 Sep 2016  
16 * by Arturo Guadalupi  
17 * modified 8 Sep 2016  
18 * by Colby Newman  
19 *  
20 * This example code is in the public domain.  
21 *  
22 * http://www.arduino.cc/en/Tutorial/Blink  
23 */  
24  
25 // the setup function runs once when you press reset or power the board  
26 void setup() {  
27     // initialize digital pin LED_BUILTIN as an output.  
28     pinMode(LED_BUILTIN, OUTPUT);  
29 }  
30  
31 // the loop function runs over and over again forever  
32 void loop() {  
33     digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)  
34     delay(1000);                      // wait for a second  
35     digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW  
36     delay(1000);                      // wait for a second  
37 }
```

sketch\_jun23b | Arduino 1.8.13

**sketch\_jun23b**

```

1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly
8 }
9

```

**Blink**

```

1 /*
2   Blink
3
4   Turns an LED on for one second, then off for one second, repeating
5   Most Arduinos have an on-board LED you can control. On the Uno it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is the correct LED pin independent of which board is used.
6   If you want to know what pin the on-board LED is connected to on your model, check the Technical Specs of your board at:
7   https://www.arduino.cc/en/Main/Products
8
9   modified 8 May 2014
10  by Scott Fitzgerald
11  modified 2 Sep 2016
12  by Arturo Guadalupi
13  modified 8 Sep 2016
14  by Colby Newman
15
16  This example code is in the public domain.
17
18  http://www.arduino.cc/en/Tutorial/Blink
19 */
20
21
22 // the setup function runs once when you press reset or power
23 void setup() {
24   // initialize digital pin LED_BUILTIN as an output.
25   pinMode(LED_BUILTIN, OUTPUT);
26 }
27
28 // the loop function runs over and over again forever
29 void loop() {
30   digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH)
31   delay(1000);                      // wait for a second
32   digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making
33   delay(1000);                      // wait for a second
34 }
35
36
37

```

1 Arduino Nano, AT

linedancer\_node | Arduino 1.8.13

**linedancer\_node**

```

1 const uint8_t dipPins[] = {pin_addr_1, pin_addr_2, pin_addr_3, pin_addr_4}; // DIP Switch Pins
2 uint8_t boardAddress; // board address from DIP switch
3
4 // variables for distance signals and filtering
5 long distA = 0; // raw distance
6 long distB = 0;
7 long distA_filtered = 0; // filtered distance signal (two stage exponential filter)
8 long distB_filtered = 0;
9 long oldDistA = 0; // distance value from previous loop
10 long oldDistB = 0;
11 long aHyst = 0; // hysteresis from filtered signal
12 long bHyst = 0;
13
14 // variables for step calculation
15 uint16_t closeness = 60; // 55 cm as detection distance for wall side / 60 for no wall
16 long deviation = round(closeness * 0.6); // how large is the change in distance data from one loop to the other - larger
17 long thresh = 4; // threshold for detection of change
18 uint16_t dipSteps = 1650; // maximum steps the stepper can travel to dip into the water
19 uint16_t minSteps = 750; // minimum steps the stepper should travel to dip into the water
20
21 // variables for homing purposes
22 const long initSteps = 600; // steps to travel to the start position after homing
23 const long maxHomingDistanceInSteps = 2000;
24 const long homingSpeedInSteps = 50;
25 const uint8_t directionTowardHome = -1;
26
27 // variable for several states and eventualities while program is executed
28 boolean homed = false;
29 boolean allowMotionA = false;
30 boolean allowMotionB = false;
31 boolean movedA = false;
32 boolean movedB = false;
33
34 // several debug sets
35 boolean debug = true; // view single program steps and high level operation = setup -> homing -> distance sensing
36 boolean debug2 = false; // view only distance data in serial monitor
37 boolean debugCurves = false; // view filtered distances and hysteresis in the signal plotter
38
39 boolean RS_Comms = false; // set true if RS485 communication is used -> implement later
40
41 //use to only run with one stepper motor
42 boolean mot_a = true;
43 boolean mot_b = false;
44
45 // individual timers and intervals for sensor reading
46 elapsedMillis sensorATimer;
47 elapsedMillis sensorBTimer;
48 uint8_t sensorAInterval = 40; // interval to read the sensor in (40 equals 25hz refresh rate)
49 uint8_t sensorBInterval = 52; //52 was ok
50 uint8_t switchInterval = 10; // micorsecond delay in sensor read function
51
52 void setup() {
53   Serial.begin(115200);
54   if (debug) {
55     Serial.println();
56   }
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105

```

Done Saving.

```

at java.lang.Thread.dumpStack(Thread.java:1336)
java.lang.Exception: Stack trace
at java.lang.Thread.dumpStack(Thread.java:1336)

```

304 Arduino Nano, ATmega328P (Old Bootloader) on /dev/cu.wchusbserial1430