

SynShop Arduino Buildup

Presented by Charley Jones, PMP aka Dataman





SynShop

Arduino Buildup

Installing code.



SynShop Arduino Buildup

**Arduino allows users
to install libraries
through their home
Arduino folder.**



SynShop

Arduino Buildup

**Advantage is that
libraries are not lost
when moving to a
new version.**



SynShop
Arduino Buildup

Windows:
Create folder **Arduino**
in My Documents.
Paste **libraries** folder
from desktop.



SynShop
Arduino Buildup

Mac:

Create folder **Arduino**
in Home folder. Paste
libraries folder from
desktop.



SynShop
Arduino Buildup

Linux:

Create folder **Arduino**
in ~ (home) folder.

Paste **libraries** folder
from desktop.



SynShop
Arduino Buildup

Start Arduino 1.0.1
Observe:
File: Examples:
ArduinoCommander



SynShop
Arduino Buildup

**Building an app.
Step by step.**



SynShop
Arduino Buildup

Arduino Commander



SynShop

Arduino Buildup

**Permits serial
control of arduino.**



SynShop
Arduino Buildup

7 build stages with
increasing
functionality at each
stage.



SynShop
Arduino Buildup

Stage 1 LEDs



SynShop Arduino Buildup

Because its better to
know that the LEDs
really do work before
you try to debug code
that really does
work....

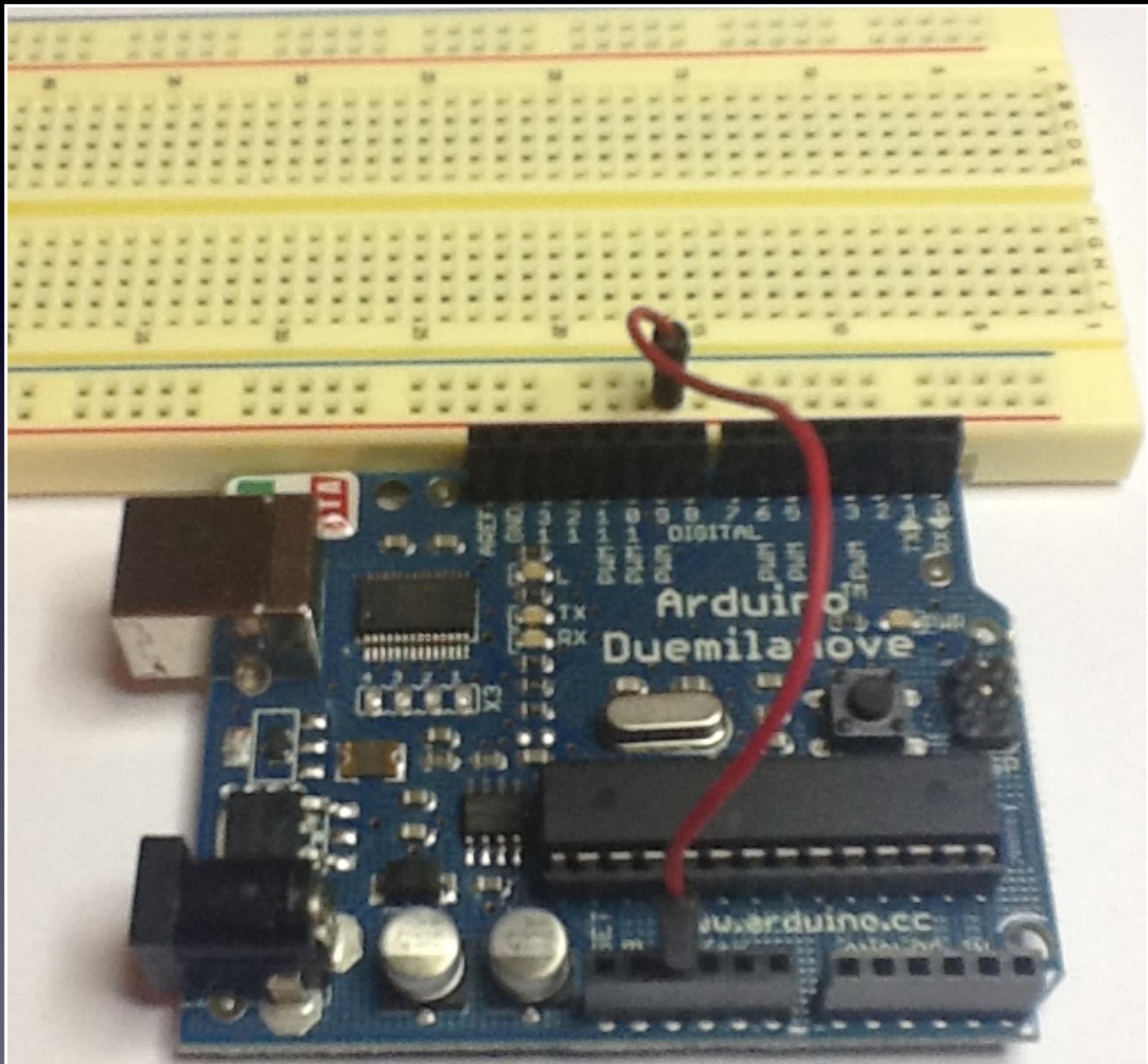


SynShop

Arduino Buildup

1

Start by
jumpering
+5v to
red rail.



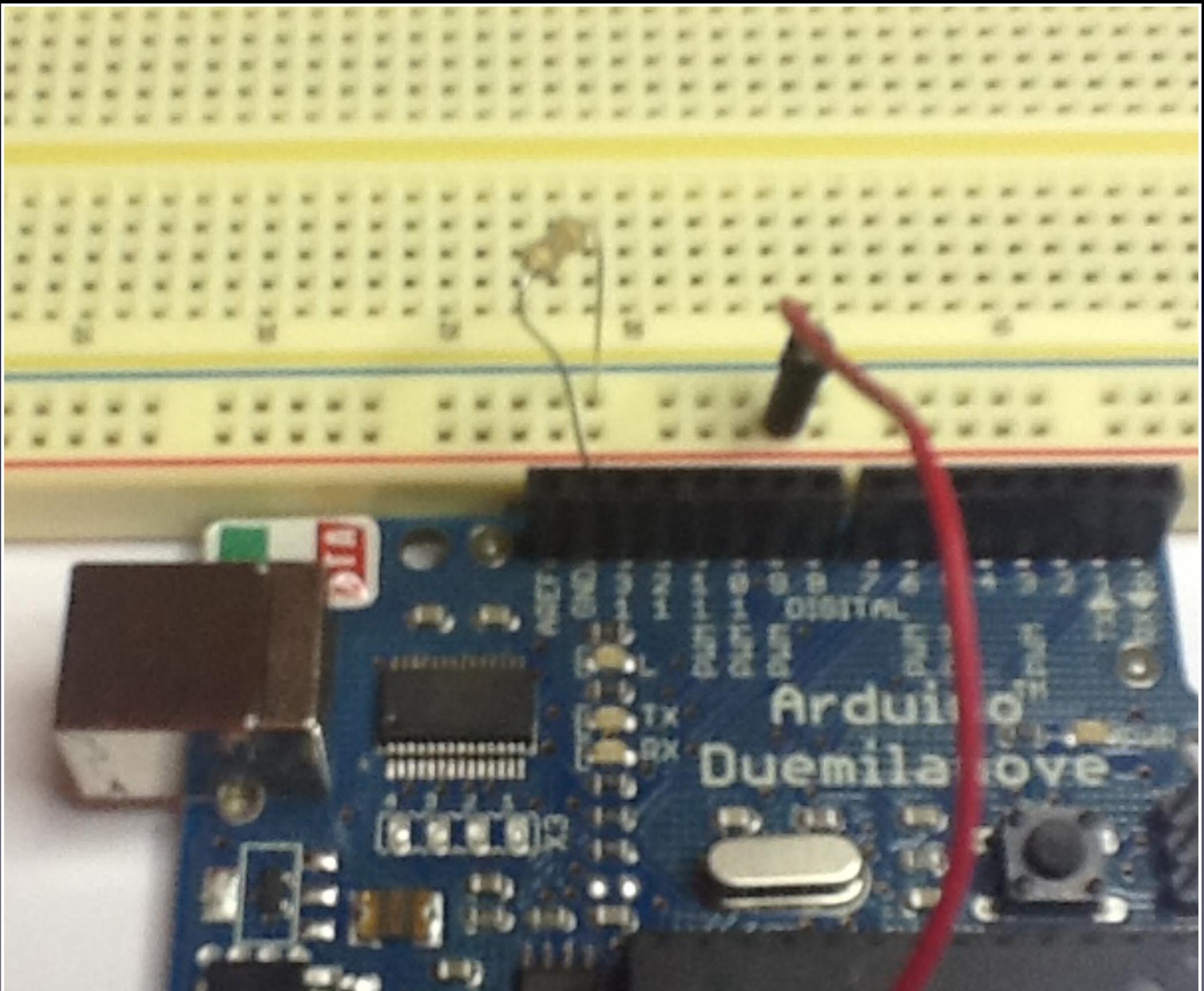


SynShop

Arduino Buildup

2

Insert 100 ohm resistor from gnd to board. While 100 ohms is not large, it will prevent arduino from burning out.



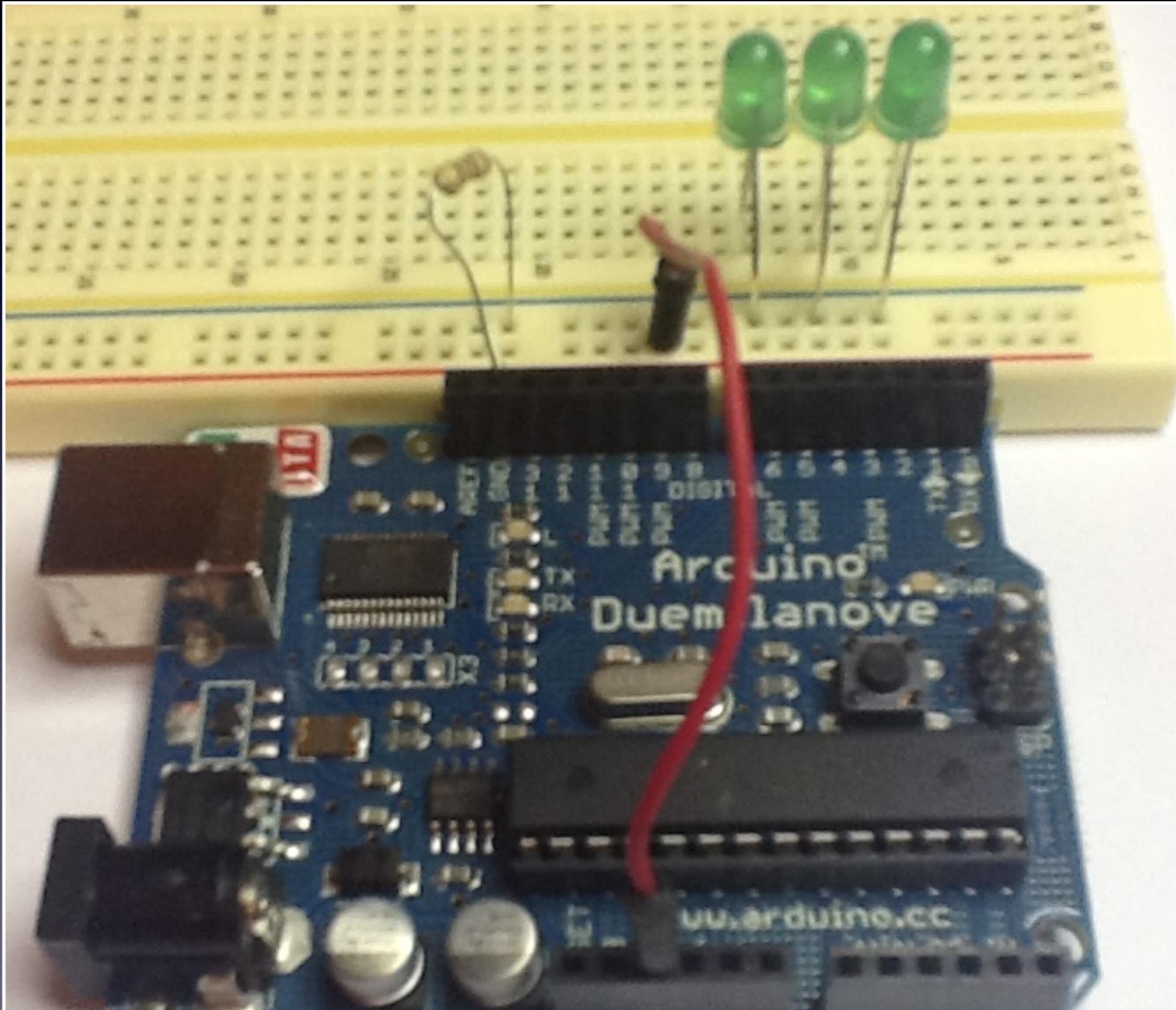


SynShop

Arduino Buildup

3

Insert 3x
LED from
resistor
rail,
negative,
to board,
positive.



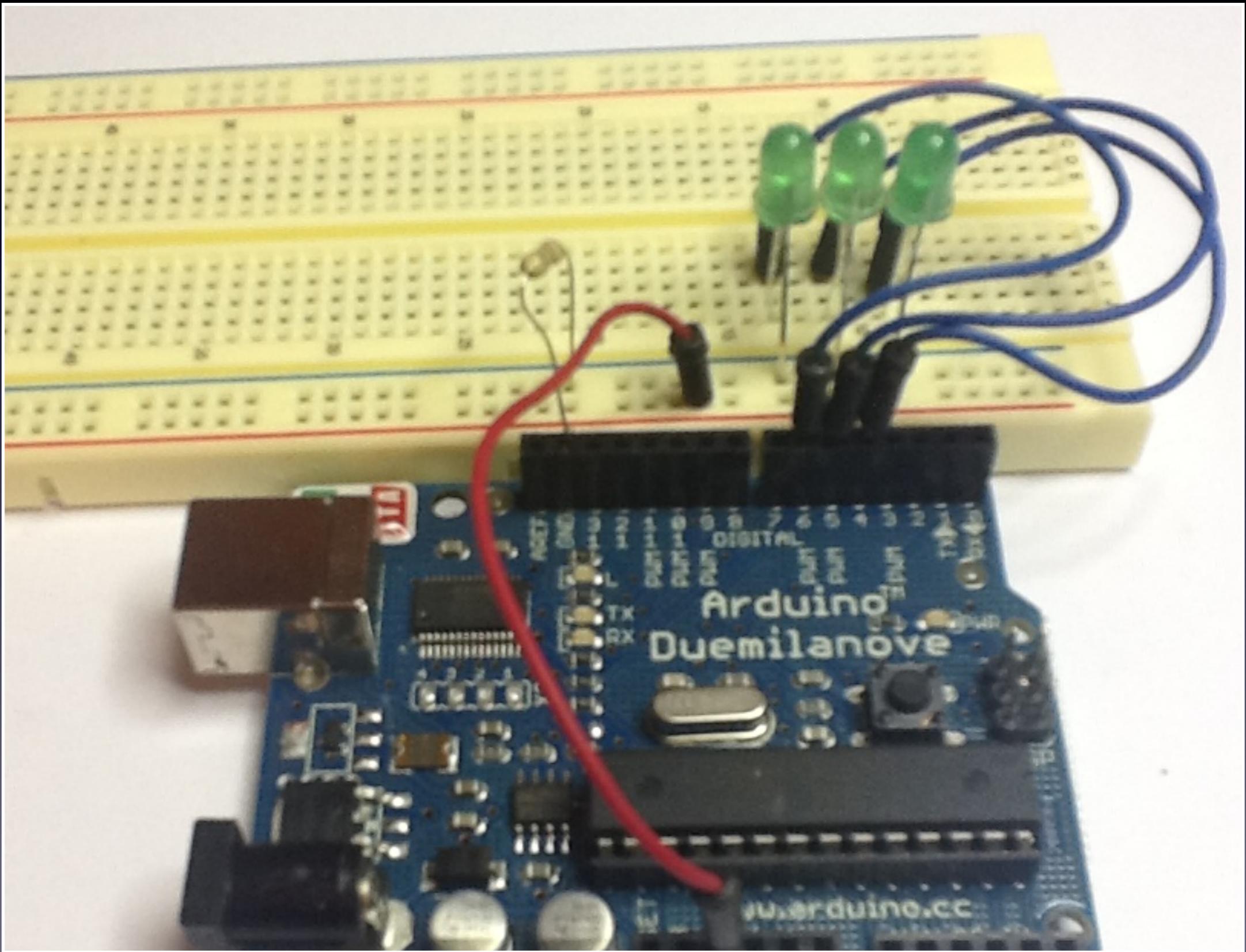


SynShop

Arduino Buildup

4

Jumper
LED
positives
to pins 4,
5, and 6.



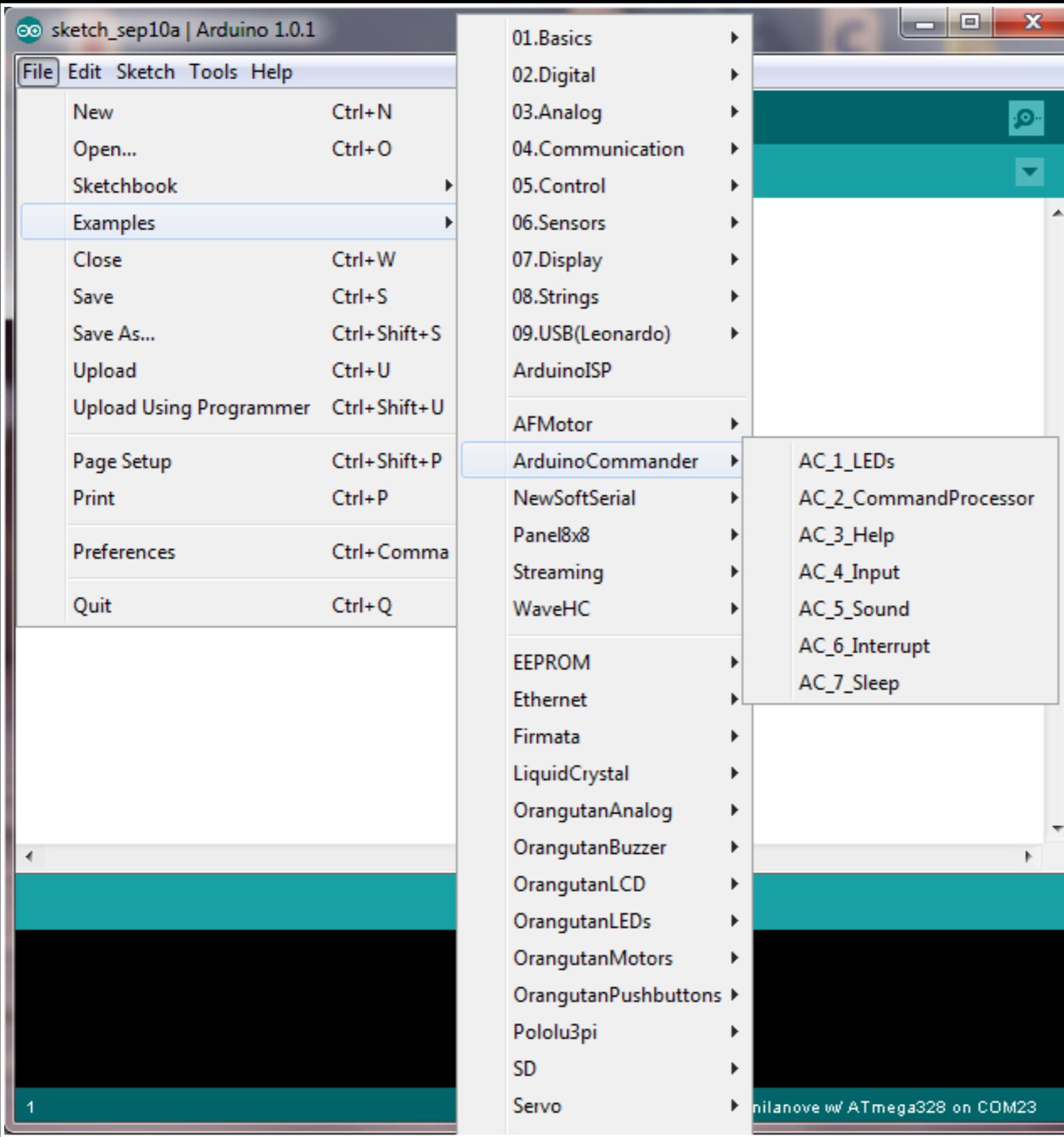


SynShop

Arduino Buildup

5

Load Arduino Buildup Stage 1.





SynShop

Arduino Buildup

6

**Globals,
setup, and
loop.**

```
int led = 6;

void setup() {
    // Initialize Output Pins
    for (int i=4;i<=6;i++) {
        pinMode(i, OUTPUT);
        digitalWrite(i, LOW);
    }
}

void loop() {
    // Loop through each LED, on, delay, off.
    if (++led>6) led=4;
    digitalWrite(led, HIGH);
    delay(1000);
    digitalWrite(led, LOW);
}
```



SynShop

Arduino Buildup

7

Compile
and
upload
code.
LEDs
should
begin
sequenc-
ing.

AC_1_LEDs | Arduino 1.0.1

File Edit Sketch Tools Help

AC_1_LEDs

```
// Arduino Buildup
// Arduino Commander

// Board Description
// 4, 5, 6 tied to gnd through resistor

// Globals
int led = 6;
```

Done uploading.

Binary sketch size: 1,124 bytes (of a 30,720 byte maximum)

14

Arduino Duemilanove w/ ATmega328 on COM23



SynShop
Arduino Buildup

We saw:
**Setup
Loop**



SynShop
Arduino Buildup

Stage 2 Command Processor



SynShop

Arduino Buildup

Concept of the message loop.



SynShop
Arduino Buildup

Behind every
Windows program
is a message loop
handler.



SynShop
Arduino Buildup

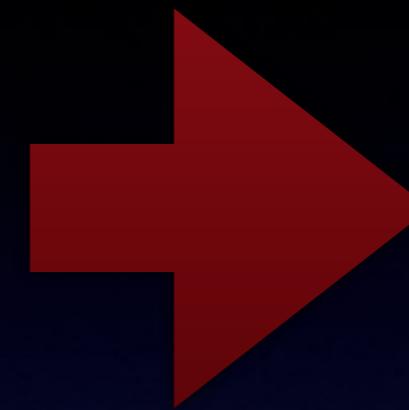
**Windows works by
passing messages.
The message loop
processes those
messages.**



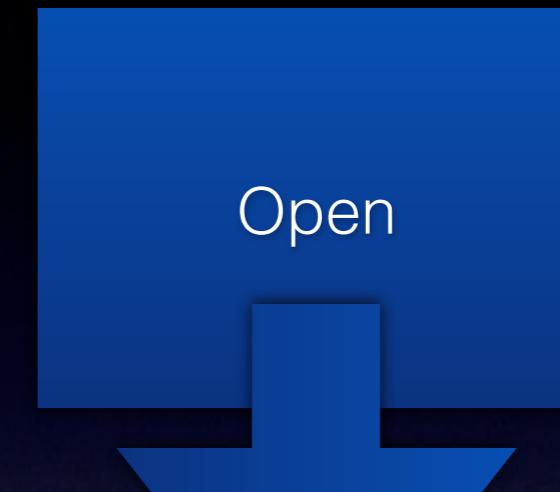
SynShop

Arduino Buildup

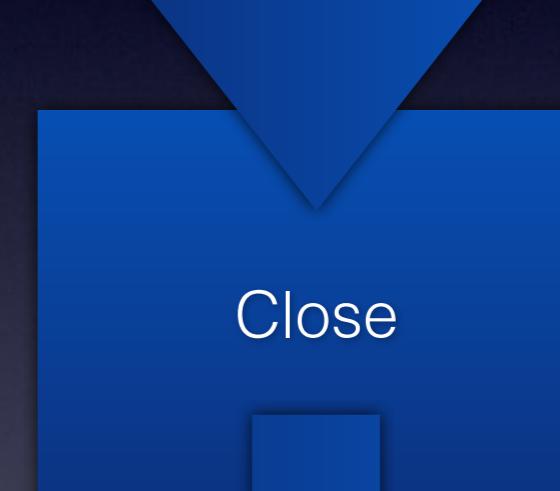
Close



Open



Close

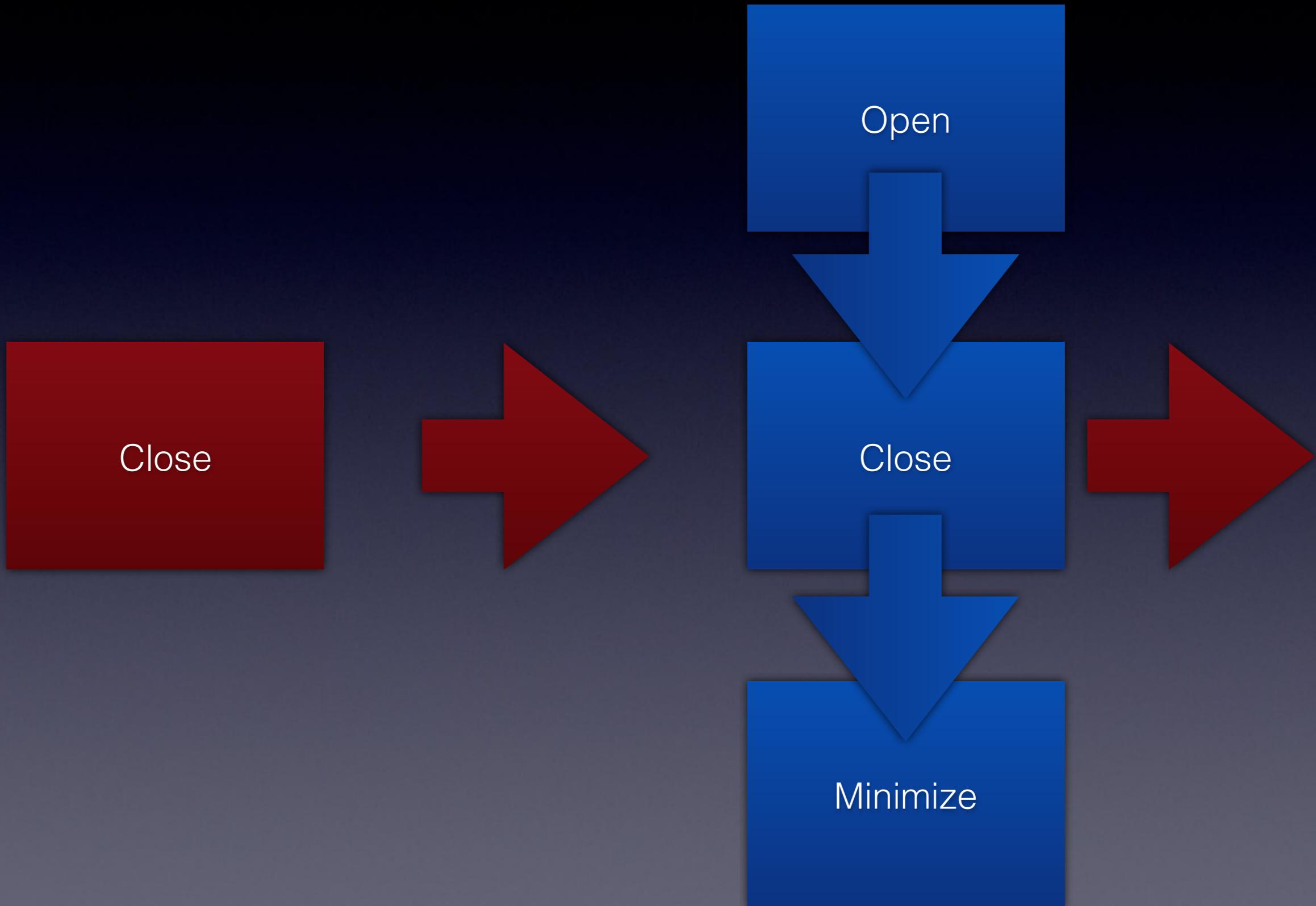


Minimize



SynShop

Arduino Buildup





SynShop

Arduino Buildup

It's a good solution,
and we'll adapt that to
our code to turn on
and off LEDs.



SynShop

Arduino Buildup

Problem:
We have to assemble
the incoming
command byte by
byte.



SynShop

Arduino Buildup

So it's a command
builder, command
processor problem....



SynShop

Arduino Buildup

1

Load Arduino Buildup Stage 2.

Globals and setup

```
// Defines
#define MAXCMD 50          // Max Command Length

// Includes
#include <string.h>

// Globals
char b;                  // Input buffer
char cmd[MAXCMD+1] = {" "}; // Command buffer

void setup() {
    // Initialize Output Pins
    for (int i=4;i<=6;i++) {
        pinMode(i,OUTPUT);
        digitalWrite(i,LOW);
    }
    // Start up Serial
    Serial.begin(9600);
    Serial.println("Arduino Commander v1.2");
    Ready();
}
```



SynShop

Arduino Buildup

2

Loop,
building
command
character
by
character.

```
void loop() {  
  
    // Read Serial Port, build command  
    if (Serial.available()) {  
        b = Serial.read();  
        // If /, then process command  
        if (b=='/') {  
            ProcessCommand();  
            return;  
        }  
        int i = strlen(cmd);  
        if (i < MAXCMD) {  
            cmd[i] = b;  
            cmd[i+1]= 0;  
            return;  
        }  
    }  
}
```



SynShop

Arduino Buildup

3

Process
the
command.

```
// Command Processor
void ProcessCommand() {
    Serial.println();
    Serial.println(cmd);

    // Command
    b = cmd[0] - '0';
    if (b>=4 && b<=6) {
        // valid Port Number
        Serial.println("");
        Serial.print("Port ");
        Serial.print(b, DEC);
        Serial.print(" set ");
        if (strcasecmp(cmd, "ON")) {
            digitalWrite(b, HIGH);
            Serial.println("ON");
        }
        else {
            digitalWrite(b, LOW);
            Serial.println("OFF");
        }
    Readv();
}
```



SynShop

Arduino Buildup

4

Syntax
error,
ready, and
clear
command.

```
        }

    Ready();
    return;
}

// Syntax Error
Serial.println("Syntax Error!");

Ready();
}

// Ready Prompt
void Ready() {
    Serial.println("");
    Serial.print("Ready >");
    ClearCMD();
}

void ClearCMD() {
    for (int i=0;i<MAXCMD;i++) cmd[i]=0;
}
```



SynShop

Arduino Buildup

5

Run the
code.

AC_2_CommandProcessor | Arduino 1.0.1

File Edit Sketch Tools Help

Upload

AC_2_CommandProcessor

```
Serial.println("OFF");  
}  
Ready();  
return;  
}  
  
// Syntax Error
```

Done uploading.

Binary sketch size: 3,172 bytes (of a 30,720 byte maximum)

1

Arduino Duemilanove w/ ATmega328 on COM23



SynShop

Arduino Buildup

6

Open a
terminal
window
and type:

4 ON/
Enter

Led 4
turns on.

Turn on
5 + 6

The screenshot shows a terminal window titled "COM23" running "Arduino Commander v1.2". The window has a "Send" button in the top right and a scroll bar on the right side. At the bottom, there are checkboxes for "Autoscroll", a dropdown for "No line ending", and a dropdown for "9600 baud". The terminal output is as follows:

```
Arduino Commander v1.2
Ready >
4 on
Port 4 set ON
Ready >
```



SynShop

Arduino Buildup

7

Issue
command:

4 OFF/
Enter

Led 4
turns off.

Turn off
5 + 6

```
6 on
Port 6 set ON
Ready >
4 off
Port 4 set OFF
Ready >
```

The screenshot shows a terminal window titled "COM23". The window has a "Send" button in the top right corner. At the bottom, there are three dropdown menus: "Autoscroll" (checked), "No line ending", and "9600 baud". The terminal window displays the following text:
6 on
Port 6 set ON
Ready >
4 off
Port 4 set OFF
Ready >



SynShop

Arduino Buildup

We saw:

Building command in
main loop.

Processing command
when eol detected.



SynShop
Arduino Buildup

Stage 3 Help



SynShop

Arduino Buildup

1

Load Arduino Buildup Stage 3

Code for
help verb.

```
// Command Processor
void ProcessCommand () {

    Serial.println();
    Serial.println(cmd);

    // HELP
    if (strcasecmp(cmd, "HELP")) {
        Syntax();
        return;
    }

    // Command
    b = cmd[0] - '0';
    if (b>=4 && b<=6) {
        // valid Port Number
        Serial.println("");
        Serial.print("Port ");
        Serial.print(b, DEC);
        Serial.print(" set ");
        if (strcasecmp(cmd, "ON")) {
            Serial.println("ON");
        }
        else if (strcasecmp(cmd, "OFF")) {
            Serial.println("OFF");
        }
    }
}
```



SynShop

Arduino Buildup

2

Help
code.

Called
from
bottom of
command
processor
and from
help
command.

```
Serial.println("Syntax Error!");

Syntax();
}

void Syntax() {
    Serial.println("");
    Serial.println("SYNTAX: COMMAND/");
    Serial.println("PORT (4-6) ACTION(ON/OFF) ie: 4 ON/");
    Ready();
}

// Ready Prompt
void Ready() {
    Serial.println("");
    Serial.print("Ready >");
    ClearCMD();
}

void ClearCMD() {
    for (int i=0;i<MAXCMD;i++) cmd[i]=0;
}
```



SynShop

Arduino Buildup

3

Run code.
Enter:
Help/
Observe
output.

Enter:
1 on/
Observe
syntax
help.

The screenshot shows a software window titled "COM23" which is a serial monitor for an Arduino. The window has a "Send" button and a scrollable text area. The text area displays the following output:

```
Arduino Commander v1.2
Ready >
help

SYNTAX: COMMAND/
PORT (4-6) ACTION (ON/OFF) ie: 4 ON/
Ready >
```

At the bottom of the window, there are three settings: "Autoscroll" (checked), "No line ending", and "9600 baud".



SynShop

Arduino Buildup

We saw:
**It's a good idea to
display command
syntax. You will
forget!**



SynShop
Arduino Buildup

Stage 4 Input



SynShop

Arduino Buildup

In this stage we
will add 3
switches.



SynShop

Arduino Buildup

We will read
switches and
combine values
for input.



SynShop
Arduino Buildup

Binary Math

Sw3 Sw2 Sw1

4 2 1

Sw3 + Sw1 = 5

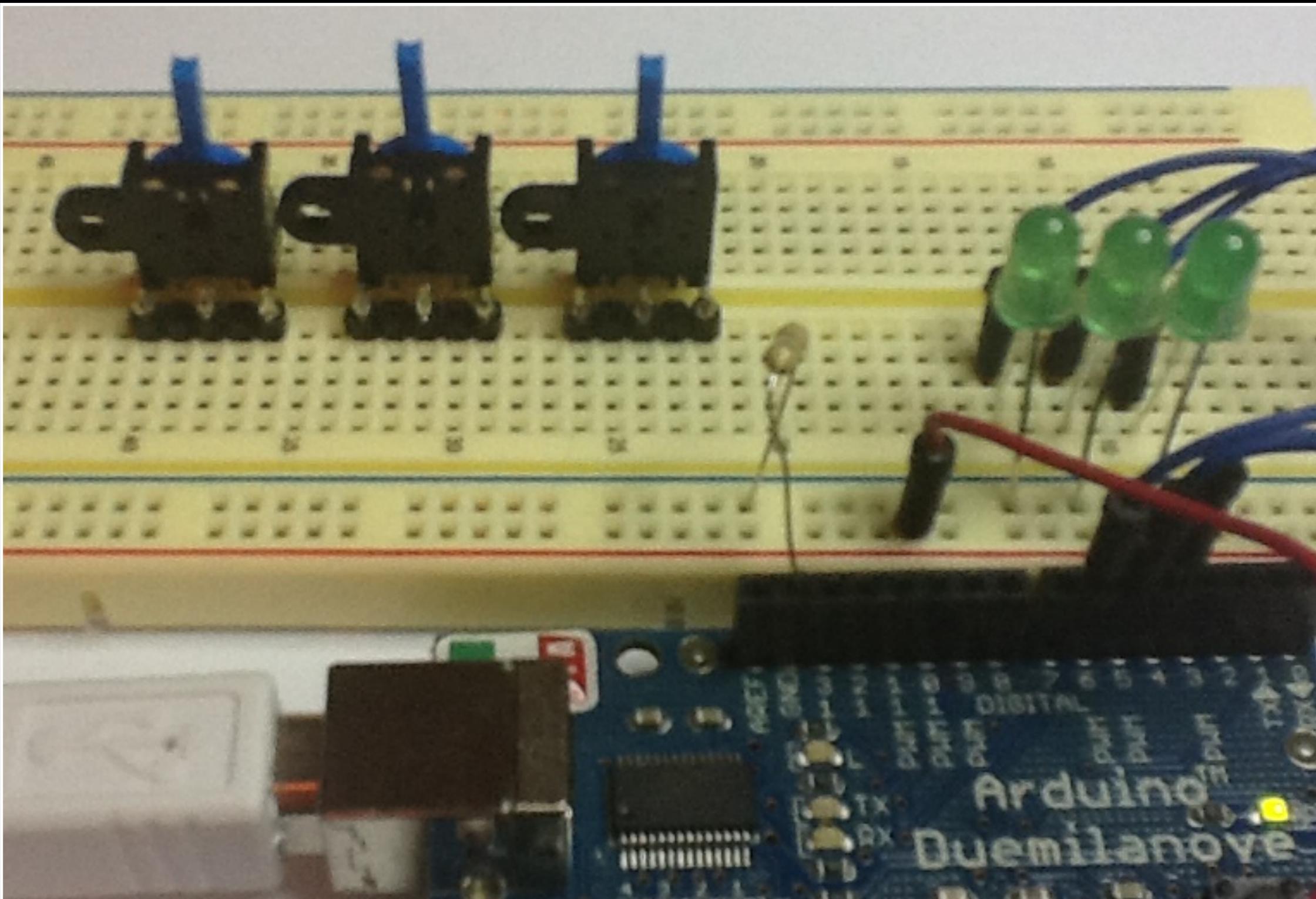


SynShop

Arduino Buildup

1

**Insert 3
switches
as shown.**



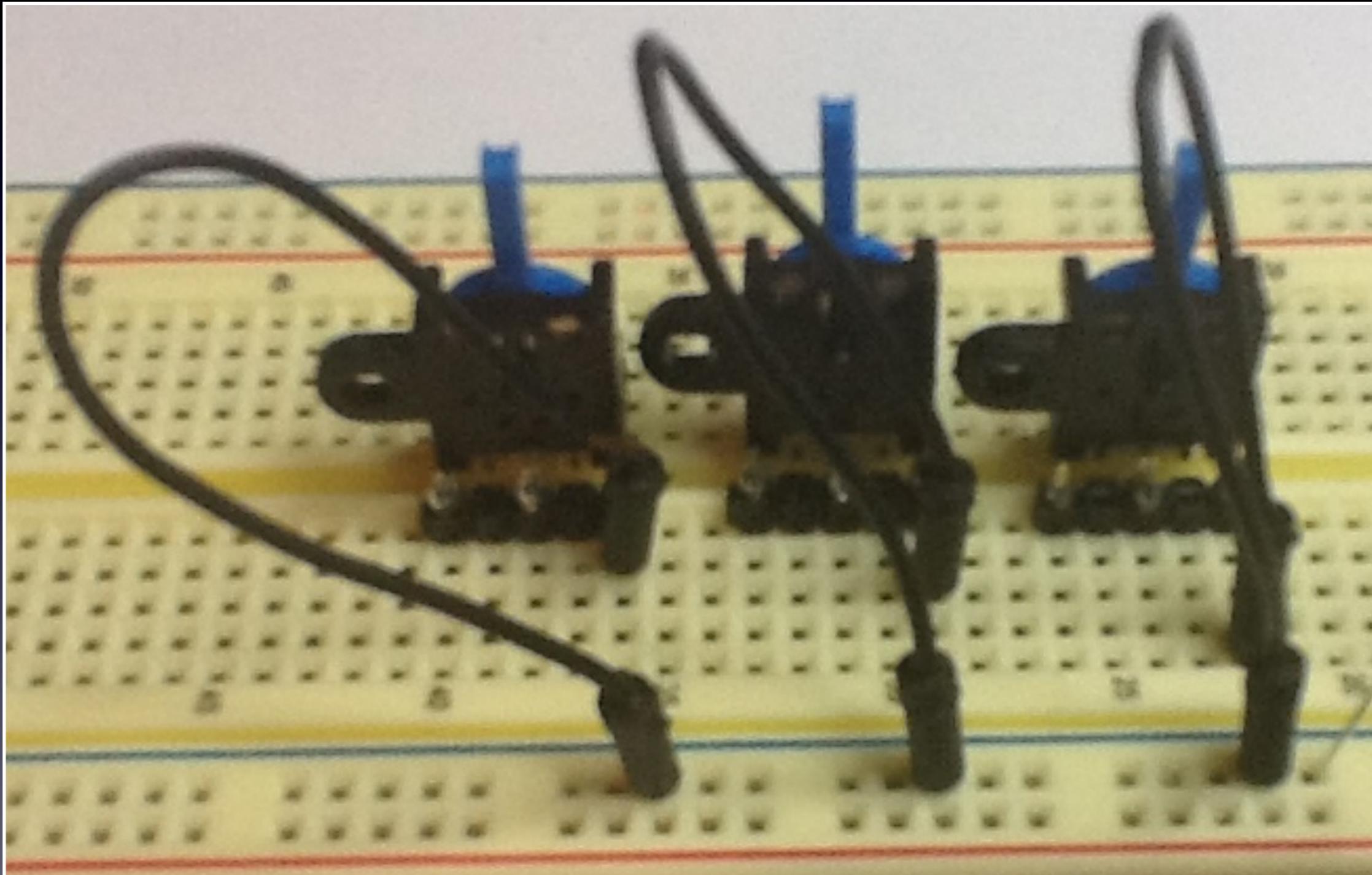


SynShop

Arduino Buildup

2

**Jumper
switch pin
3, right-
most pin,
to ground
rail.**



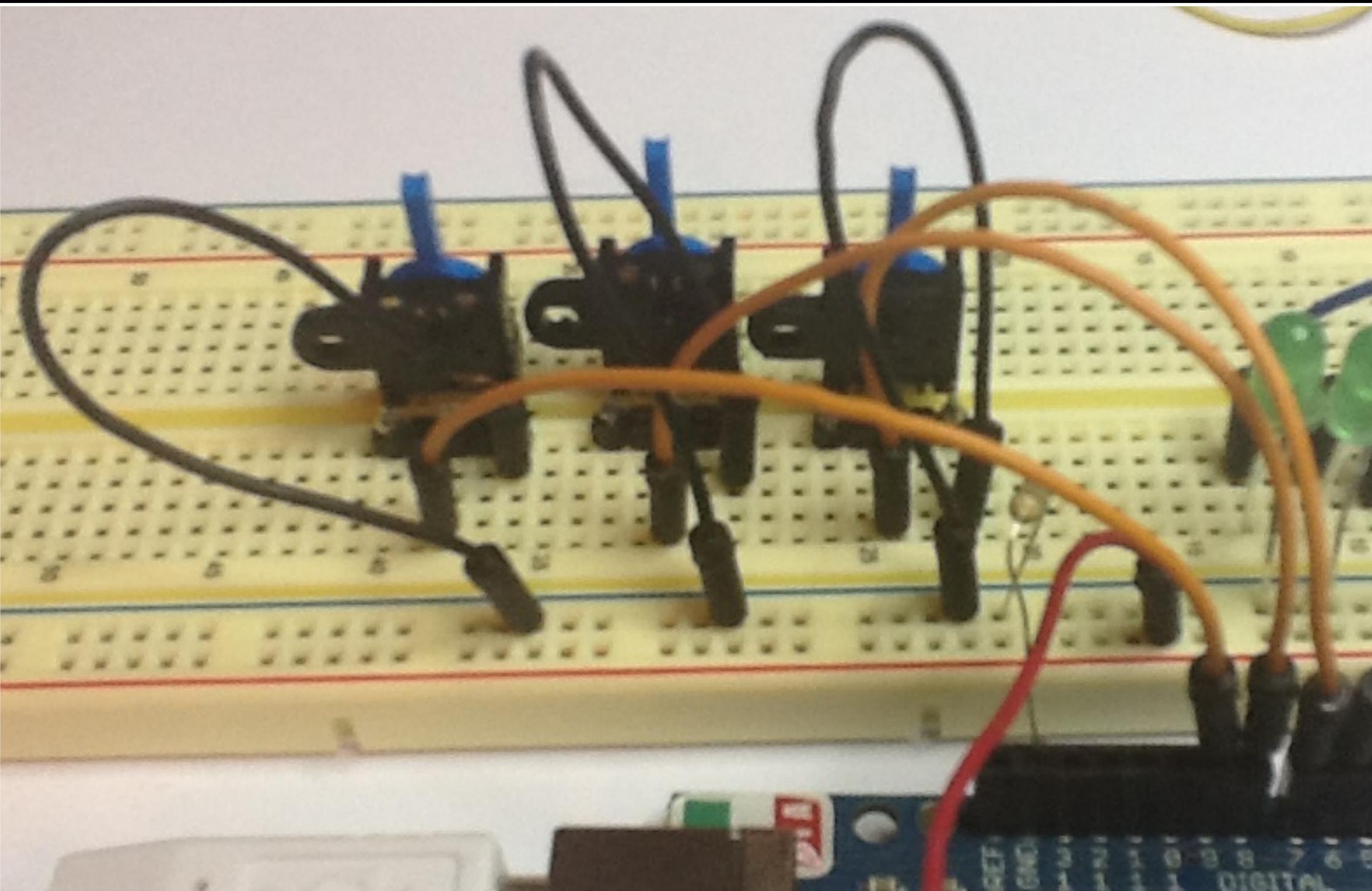


SynShop

Arduino Buildup

3

Jumper
switch pin
2, middle
pin, to
pins 7, 8,
and 9.





SynShop

Arduino Buildup

4

Load Arduino Buildup Stage 4.

Global
and setup
changes.

```
// Input buttons
int btn[3]={1,1,1};           // Button Status Register

void setup() {
    // Initialize Output Pins
    for (int i=4;i<=6;i++) {
        pinMode(i,OUTPUT);
        digitalWrite(i,LOW);
        pinMode(i+3,INPUT_PULLUP);
    }
    // Start up Serial
    Serial.begin(9600);
    Serial.println("Arduino Commander v1.2");
    Ready();
}
```



SynShop

Arduino Buildup

5

Input
polling at
bottom of
Loop.

```
// Input Polling
int dirty = 0;
for (int i=7;i<=9;i++) {
    b=digitalRead(i);
    if (b!=btn[i-7]) {
        btn[i-7]=b;
        dirty = 1;
    }
}
if (dirty) {
    Serial.println("");
    Serial.print("Input Value = ");
    Serial.println(btn[0] + btn[1]*2 + btn[2]*4);
}
}
```



SynShop

Arduino Buildup

6

Execute code.
Pull down switch 1.
Observe Input value 6
Predict values,
pull switches,
confirm output.

```
Arduino Commander v1.2
Ready >
Input Value = 6
Input Value = 7
```

Autoscroll No line ending 9600 baud



SynShop

Arduino Buildup

We saw:
**It's pretty simple to
add on new
functionality...**



SynShop
Arduino Buildup

Stage 5 Sound



SynShop
Arduino Buildup

The tone command
allows aruduiino to
make sound.

`tone(port, frequency [,duration]);`



SynShop Arduino Buildup

The team also published a table of notes and frequencies which I crunched down into code.



SynShop

Arduino Buildup

My revised code now
makes it possible to
send commands to the
arduino to play music.



SynShop

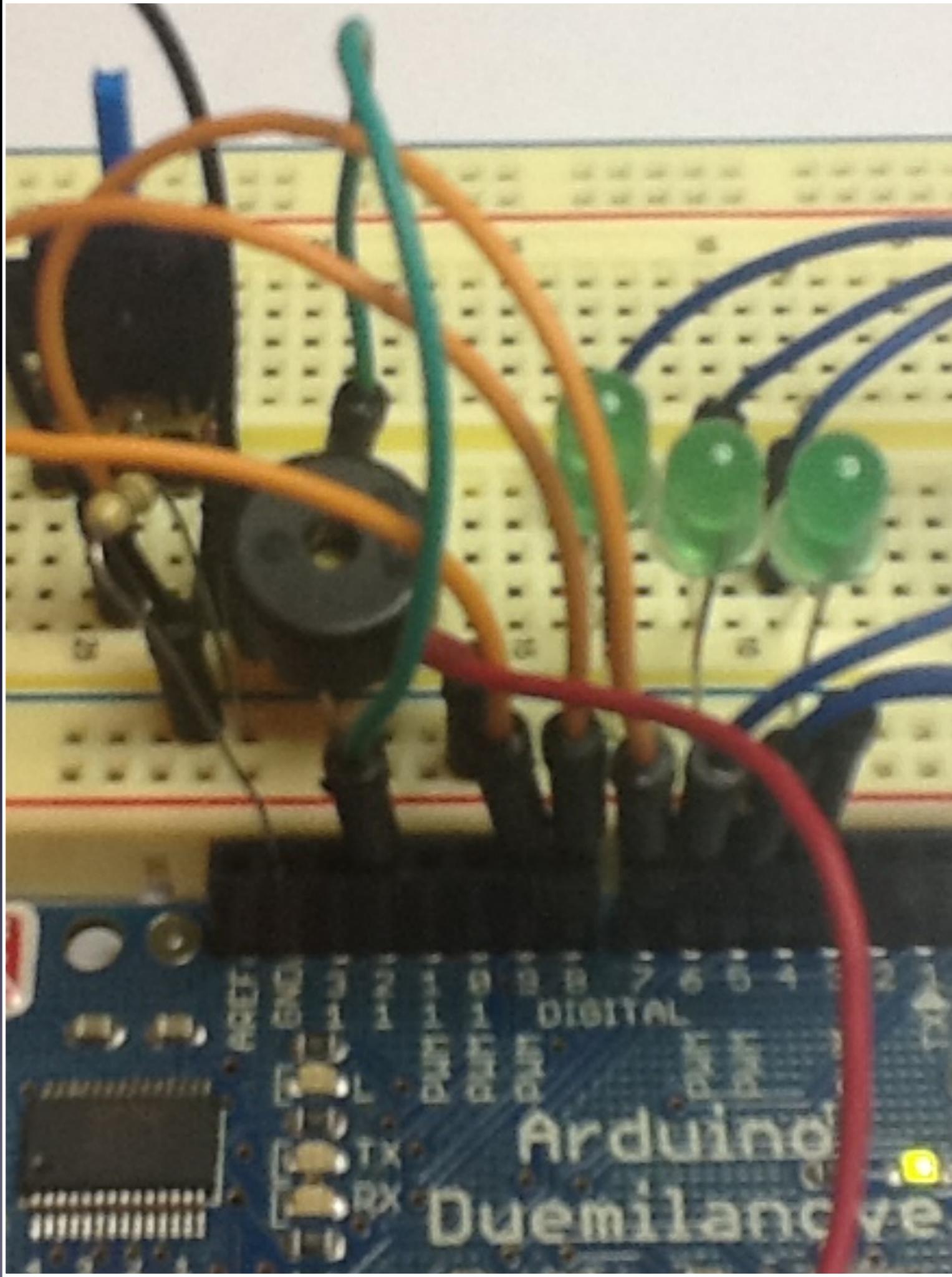
Arduino Buildup

**There's a lot of parts to
this code, so hang on!**



1

Insert
piezo
electric
speaker
from
ground to
board.
Jumper to
pin 12.





SynShop

Arduino Buildup

2

Load Arduino Buildup Stage 5.

Some defines.

Notes definitions

```
//: Board Description
//: 4,5,6 tied to gnd through resistor
//: 7,8,9 input switchs, N/O, through resistor to grnd
//: 12 piezo speaker through resistor to grnd

//: Defines
#define MAXCMD 50           // Max Command Length
#define MAXTIK 10000         // Note Duration
#define PTONE 12             // Port for Piezo

// Notes
char *note = 0;           // Pointer to next note
long ntick = 0;           // How long has not played
unsigned char notes[] = {
    'B', '0', ' ', 31, 0,
    'C', '1', ' ', 33, 0,
    'C', 'S', '1', 35, 0,
```



SynShop

Arduino Buildup

3

Hooking
the main
loop.

```
void loop() {
    // Play next note?
    if (note)
        if (++ntick > MAXTIK)
            PlayNote();
```

```
// PLAY
if (strcasecmp(cmd, "PLAY")) {
    note = cmd+2;
    return;
}
```

```
void Syntax() {
    Serial.println("");
    Serial.println("SYNTAX: COMMAND/");
    Serial.println("PORT (4-6) ACTION (ON/OFF) ie: 4 ON/");
    Serial.println("PLAY NOTE [NOTE [...]] ie: PLAY C1 C2 /");
    Ready();
}
```

Syntax

Hooking
command
processor



SynShop

Arduino Buildup

5

PlayNote Routine.

First part
figures
out if we
are done
and shuts
down note
playing.

```
void PlayNote() {
    char n[4]; // holds next note text
    n[3]=0;
    char c[4];
    c[3]=0;
    unsigned char *p; // pointer into note array
    int t; // tone;
    note+=3;
    if (note-cmd >= strlen(cmd)) {
        note = 0;
        noTone(PTONE);
        Ready();
        return;
    }
```



SynShop

Arduino Buildup

6

PlayNote Routine.

Second
part
figures
out what
frequency
to play.

```
// note to play
strncpy(n, note, 3);
// iterate through notes
for (p = notes; *p; p+=5) {
    strncpy(c, (char *)p, 3);
    if (strcasecmp(n, c)) {
        noTone(PTONE);
        delay(100);
        unsigned int itone = *(p+3) + (* (p+4) * 256);
        tone(PTONE, itone);
        ntick = 0;
        return;
    }
}
```



SynShop

Arduino Buildup

5

Enter
command:

Play d5 e5
c5 c4 g4 /

Note that
spacing is
important
as well as
final
space
before /.

Arduino Commander v1.2

Ready >

play d5 e5 c5 c4 g4

Ready >



SynShop Arduino Buildup

We saw:
**It's simple to add
sound.**
**Difficult part is
keeping your place!**



SynShop
Arduino Buildup

Stage 6 Interrupts



SynShop
Arduino Buildup

**It's not always
convenient to poll
for changes as we
did with the
switches.**



SynShop Arduino Buildup

We can attach code to certain pins and events. Code is executed as an interrupt.



SynShop
Arduino Buildup

**Special
considerations
for interrupt code.**



SynShop
Arduino Buildup

**Serial and delay
functions do not
work in interrupts.**



SynShop Arduino Buildup

**Any variables
shared with main
code marked
volatile.**

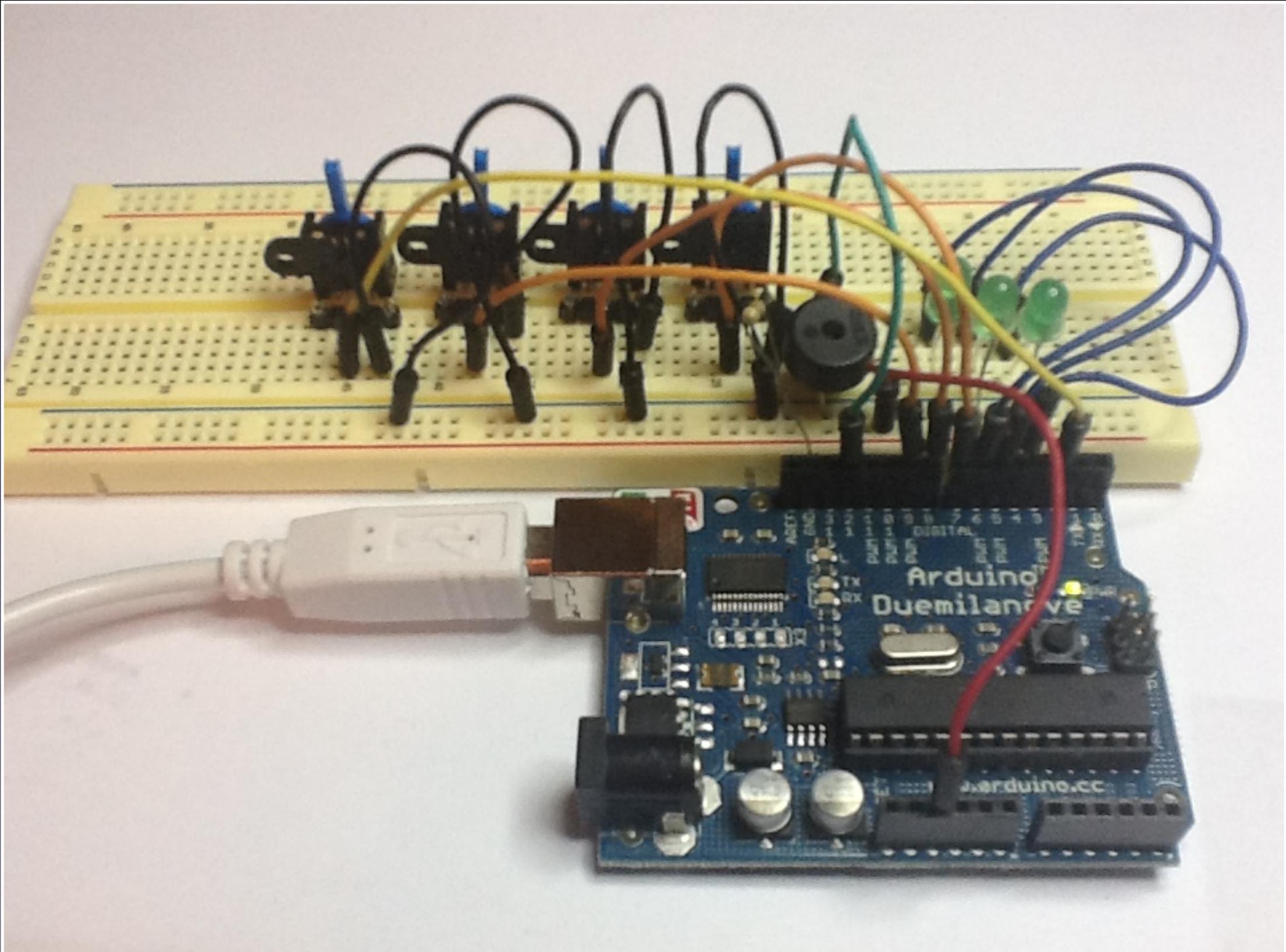


SynShop

Arduino Buildup

1

Add
another
switch
jumper
switch 3
to ground
and
switch 2
to pin 2.
This is
final build
config-
uration.





SynShop

Arduino Buildup

2

Load Arduino Buildup Stage 6.

Here we
define a
flag as
volatile
and
configure
the
interrupt.

```
// Interrupt Flag
volatile int iflag = 0;

void setup() {
    // Initialize Output Pins
    for (int i=4;i<=6;i++) {
        pinMode(i,OUTPUT);
        digitalWrite(i,LOW);
        pinMode(i+3,INPUT_PULLUP);
    }
    // Attach Interrupt
    pinMode(2,INPUT_PULLUP);
    attachInterrupt(0, Interrupt, FALLING);

    // Start up Serial
    Serial.begin(9600);
    Serial.println("Arduino Commander v1.2");
    Ready();
}
```



SynShop

Arduino Buildup

3

We check
and report
on the
interrupt
flag in the
main loop.

And finally
the
interrupt
routine.

```
void loop() {
    // Interrupt
    if (iflag) {
        iflag = 0;
        Serial.println("");
        Serial.println("Caught Interrupt");
        delay(1000);
        return;
    }

    // Play next note?
    if (note)
```

```
void Interrupt() {
    iflag = 1;
    cli();
}
```



SynShop

Arduino Buildup

4

Run code.

Activate
switch 3.
Observe:
Caught
Interrupt.

Arduino Commander v1.2

Ready >

Caught Interrupt



SynShop

Arduino Buildup

We saw:

Code can be attached
to interrupts. We can't
talk directly to serial,
but we can set a flag.



SynShop
Arduino Buildup

Stage 7 Sleep



SynShop

Arduino Buildup

**Atmel processors
have a low power
sleep mode.**



SynShop Arduino Buildup

All processing is suspended and we consume very little power.



SynShop

Arduino Buildup

We can be
awakened from
input on several
special pins.



SynShop

Arduino Buildup

1

Load Arduino Buildup Stage 7.

Sleep
command
and
process.

```
// SLEEP
if (strcasecmp(cmd, "SLEEP")) {
    set_sleep_mode(SLEEP_MODE_PWR_DOWN);
    sleep_mode();
    sleep_disable();
    delay(1000);
    Serial.flush();
    Serial.println("Ok! Ok! I'm Awake! I'm Awake!");
    Ready();
    return;
}
```



SynShop

Arduino Buildup

2

Execute.

Arduino Commander v1.2

Ready >ÿ

Enter:
sleep/

You may
see
garbage.



SynShop

Arduino Buildup

3

Execute.

Enter:
help/
Observe
no output.

Toggle
switch 3,
observe
output.

Arduino Commander v1.2

Ready >ÿC...»YY, CÃOK! Ok! I'm Awake! I

Ready >

Caught Interrupt



SynShop

Arduino Buildup

We saw:
We can sleep the
processor, waking it
up on input.



SynShop
Arduino Buildup

Bonus Stage



SynShop
Arduino Buildup

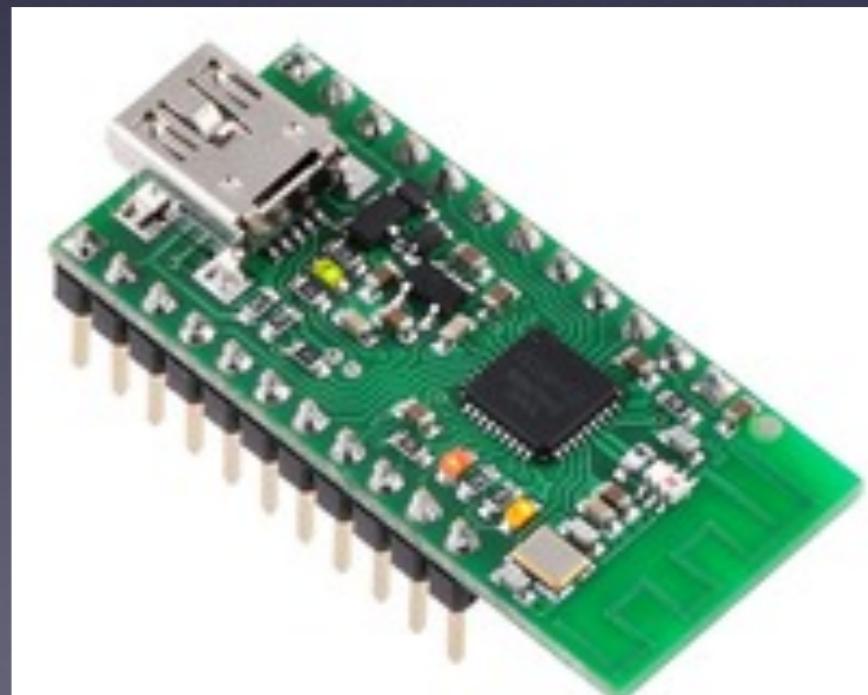
Stage 8 Wireless



SynShop

Arduino Buildup

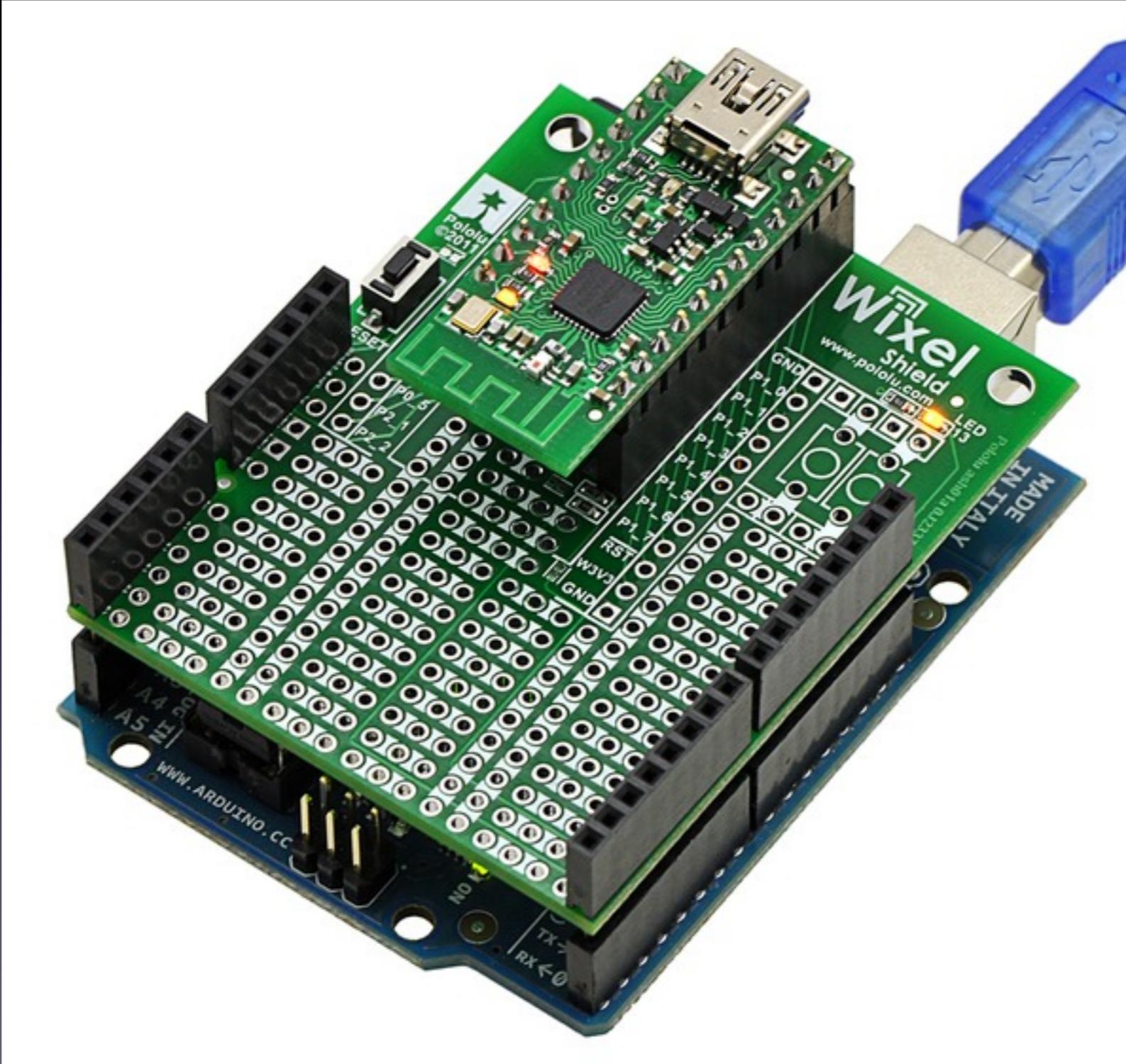
**Wixel is an easy to use
wireless solution from
Pololu.**





SynShop

Arduino Buildup





SynShop
Arduino Buildup

**With the Wixel Shield
and Arduino
Commander you can
remotely control
arduino.**

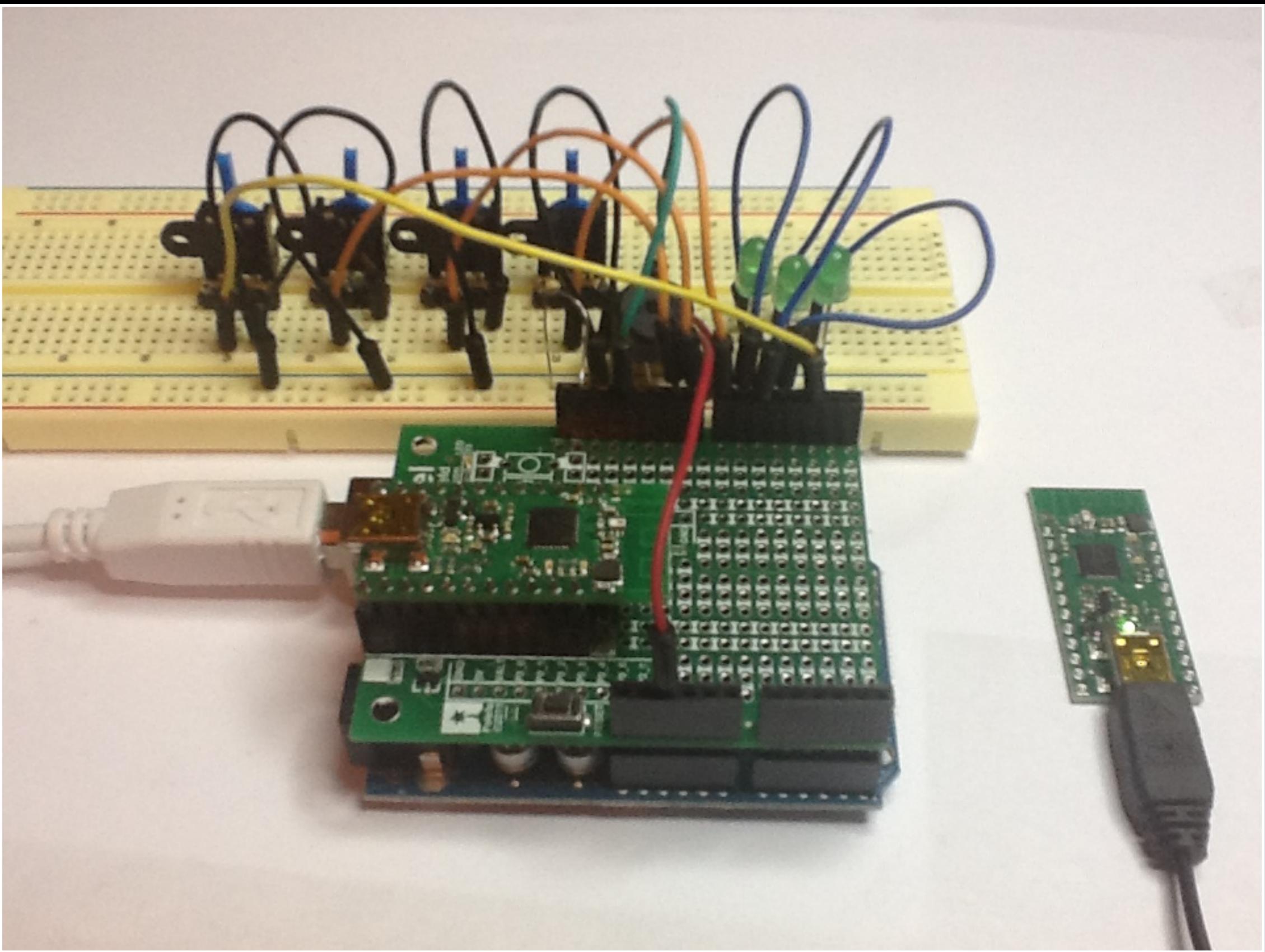


SynShop

Arduino Buildup

1

Live
Wireless
Arduino
Com-
mander.





SynShop Arduino Buildup

**And that's the
class!**



SynShop

Arduino Buildup

This entire course is published
github.com/Dataman/ArduinoBuildup

Or simply go to
github.com/dataman





SynShop

Arduino Buildup



Thank You!