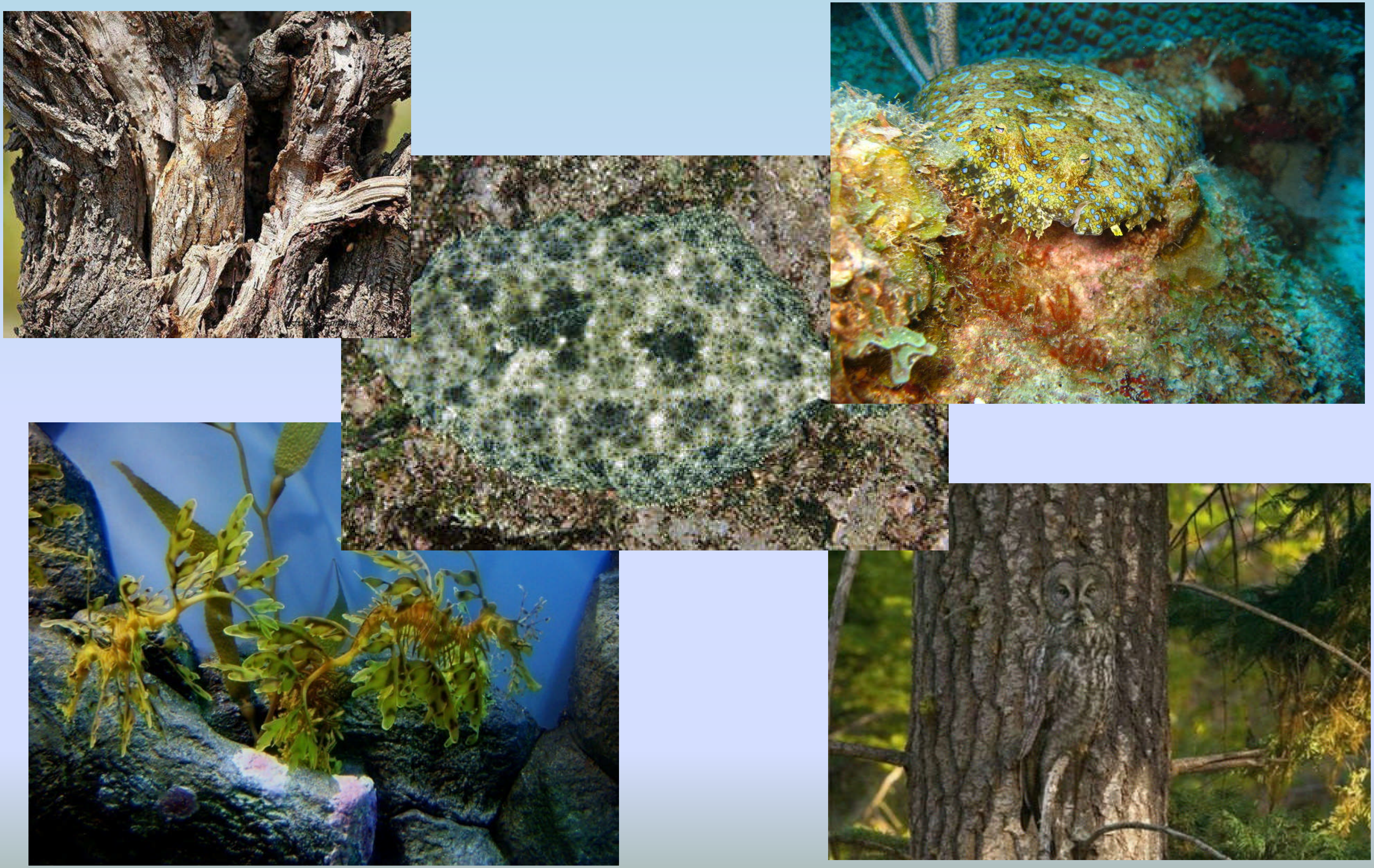
# Control for Scale Model Animation Including DCC

2ND Div PNR

**Presented by Geoff Bunza**

1.4

# What Makes Something Appear to Be Alive?



# ***Animated*** **versus** ***Animation***

You can create ***animated*** features

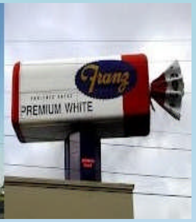
Blinking lights, Body movements, etc.

When creating an ***animation*** consider the whole:

- Movement
- Light
- Sound
- Synchronization
- Story

# *Movement*

- **Circular** (likely continuous)

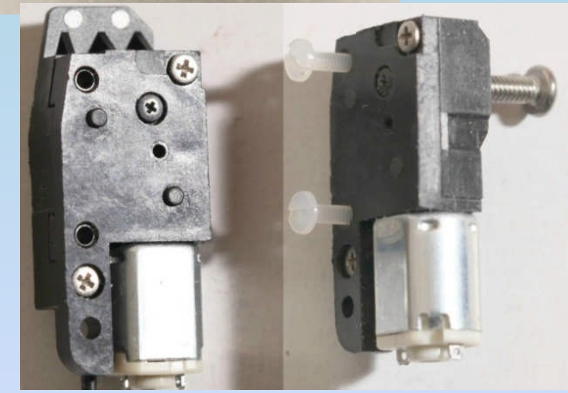


- **Linear** (movement along a line, possibly with pauses and reversals)
- **Steered** (forced path by track, trough, or other means)

- **Guided** (Intelligently steered without obvious constraints)

- **Random** (arbitrary, chaotic, non-repeating)

# *Drives for Movement*



Right Angle Drive Gearmotors

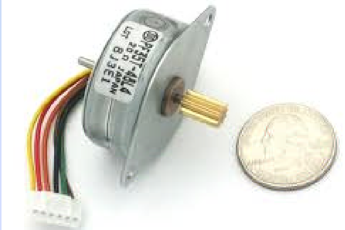
Vibrator Motors

HiTec HS-55

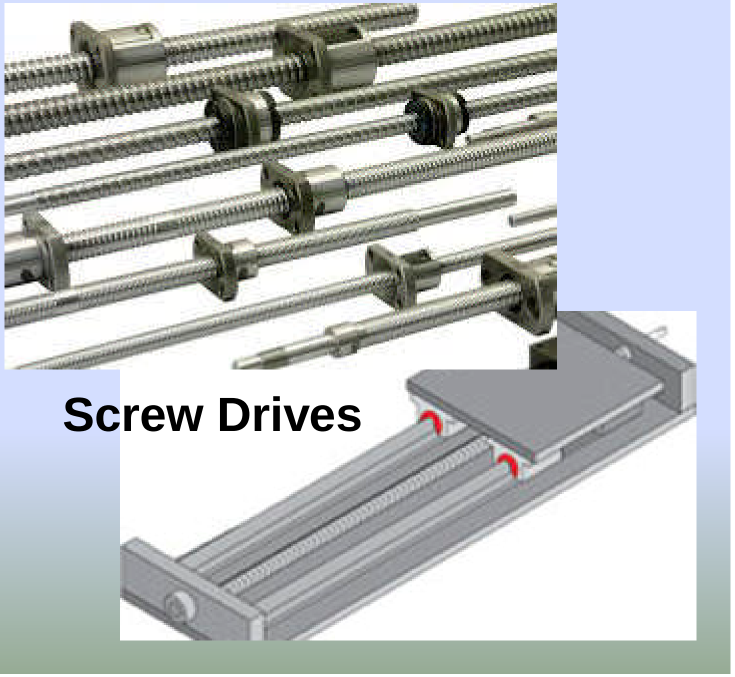
LS-0003

Servo Motors

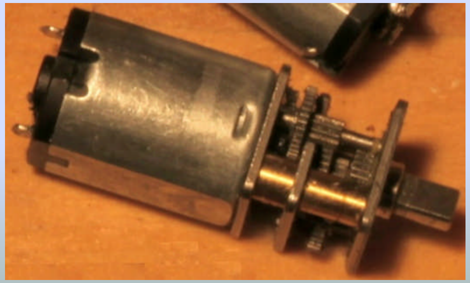
Planetary Gearmotors



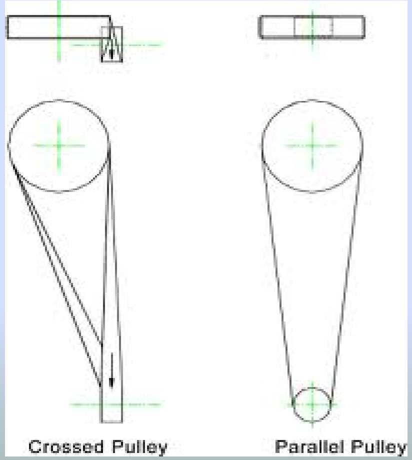
Micro Stepper Motor



Screw Drives



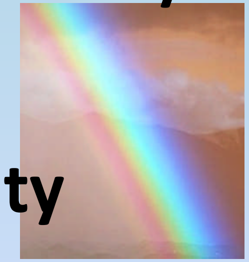
Gearmotor

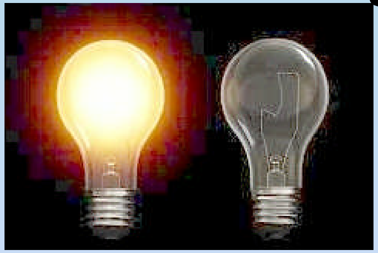


Belt or String Drive

# *Lights*

- Background/Point Sources vs. Large Scene Lighting



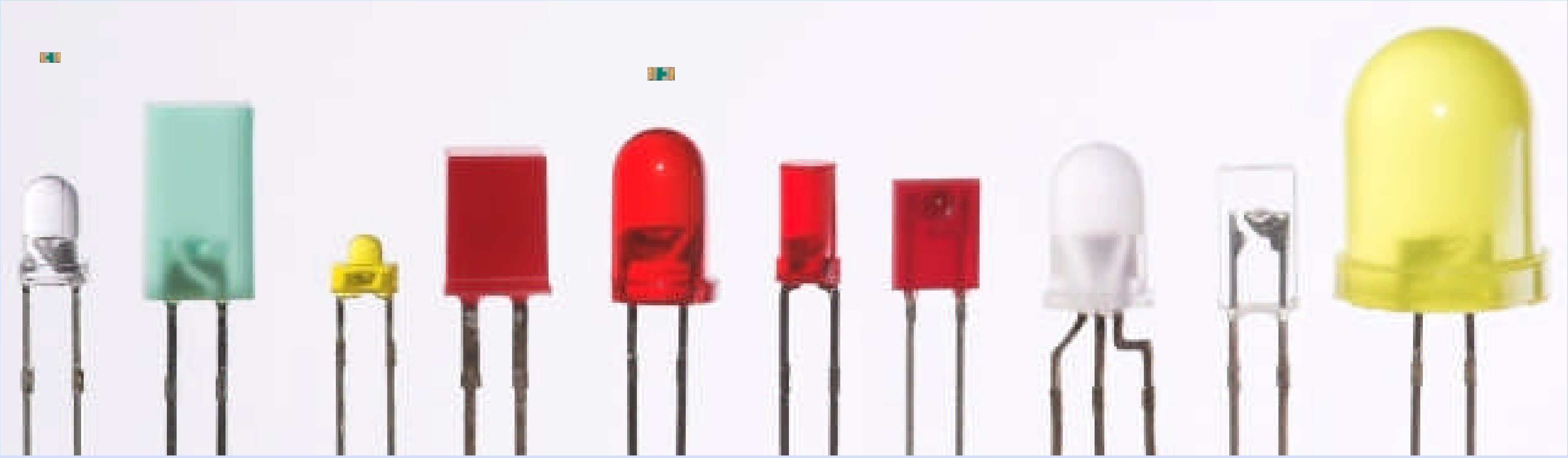


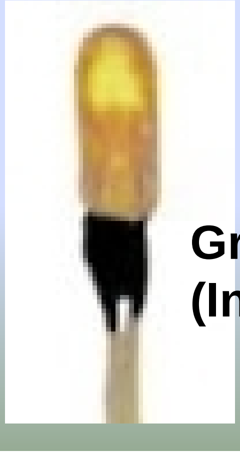
- Color
- Intensity
- Modulation/Change/Turn-on-off

- Scale

- Coverage (point, line, circle, streak, etc.)
- Placement for view (direct, indirect, silhouette, etc)
- Power routing

# *Types of Light Emitting Diodes (LEDs) You Might Know*





**Grain of Wheat (Incandescent)**

# Sound, Sounds, and Noisemakers

- **Background**
  - *The unseen sounds obvious and appropriate for the scene (birds, traffic, etc.)*

- **Foreground**
  - *A collection of sounds appropriate for a near view not attributed to a single point source (machinery, fans, water flowing, conversations)*

- **Ambiance**
  - *Using sound to create a feeling for a scene (rainfall, fog horns, wind, etc.)*

- **Spot**
  - *A specific sound easily matched to a specific source*

*(hammer, saw, crane, etc.)*

- **Mobile**
  - *Sound generated by a specific source moving in a scene (locomotive,car, etc.)*

# *Sound and Noisemakers*

- **DCC Sound Decoders**
  - *Sound clips can be uploaded and replaced, subbing for bells, whistles, etc*
- **Reprogrammable DCC Sound Decoders**
  - *Sounds can be replaced but also the internal sequencing can be reprogrammed (**Digitrax, Zimo**?)*
- **Commercial Sound Players**
  - *Triggerable, configurable, higher quality sound (**Pricom Dream Player**)*
- **Sound Modules**
  - *Commercial Sound Modules (**ITT Products HQ Series Sound Modules**)*
  - *Module sound players: mp3,WAV, & AD4 players **(WTV020-SD-16P Module**)*
- **Mp3 players**
  - *CD & USB (Thumb) Drive packaged players*
- **Other Issues: Sound Level, Speakers, Enclosures, & Placement**

# *Timing, Cueing, & Synchronization*

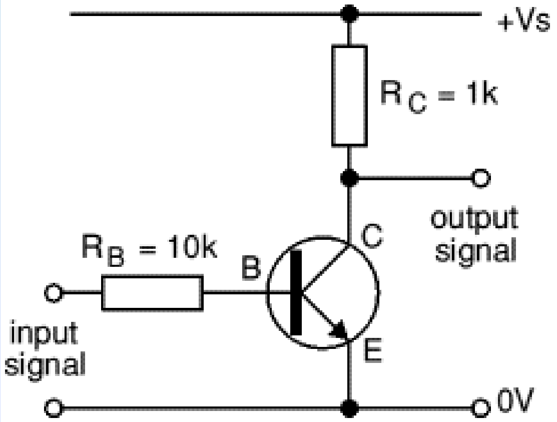
- **Consider Animation as a sequence of actions**
  - **Move this…**
  - **Play these sounds…**
  - **Set this speed…**
  - **Turn this light on… off… on… …**
  - **Wait so long…**
  - **Wait until something happens…**
- **How does an action start ?**
- **What are the delay times among actions ?**
- **How are the actions coordinated ?**
- **How does an action end ?**

# ***Triggers & Cueing***

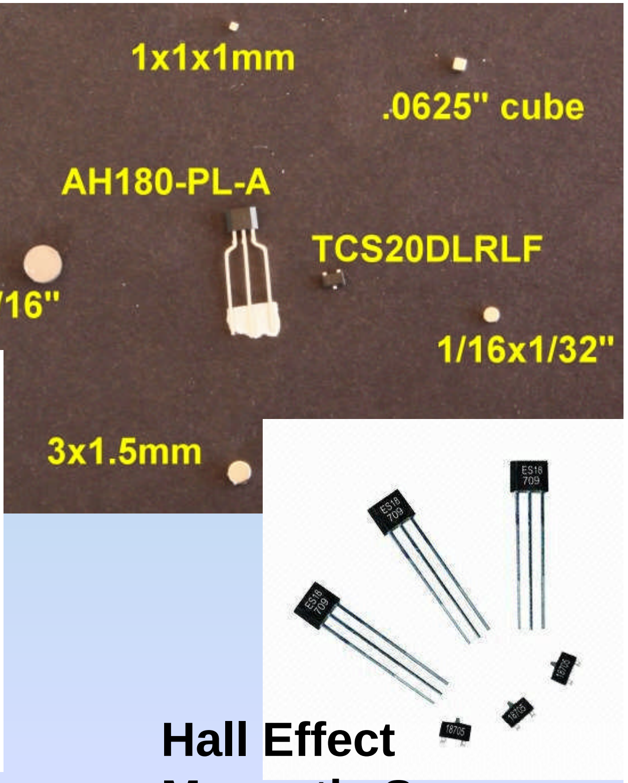
- A ***Trigger*** is a detectable change
  - Switch closure: ***metal contacts, push button, switch…***
  - A sensor change: ***optical sensor, magnetic sensor…***
  - The end of a timer: ***mechanical or electrical or program***
- A ***Timer*** is some mechanism that has a known delay
- A ***Sequencer*** provokes one or more actions at different time intervals
- ***Cueing*** is the coordinated ***start*** of multiple actions– like movement start and sound generation
- Action ends with:
  - Completion of sequence
  - End of time period
  - Ending trigger
  - No end – continuous loop

# *Triggers*

Metal Contacts/Switches



Transistor Switch
Hi In – Low Out



Hall Effect
Magnetic Sensors

PIR Motion Sensor

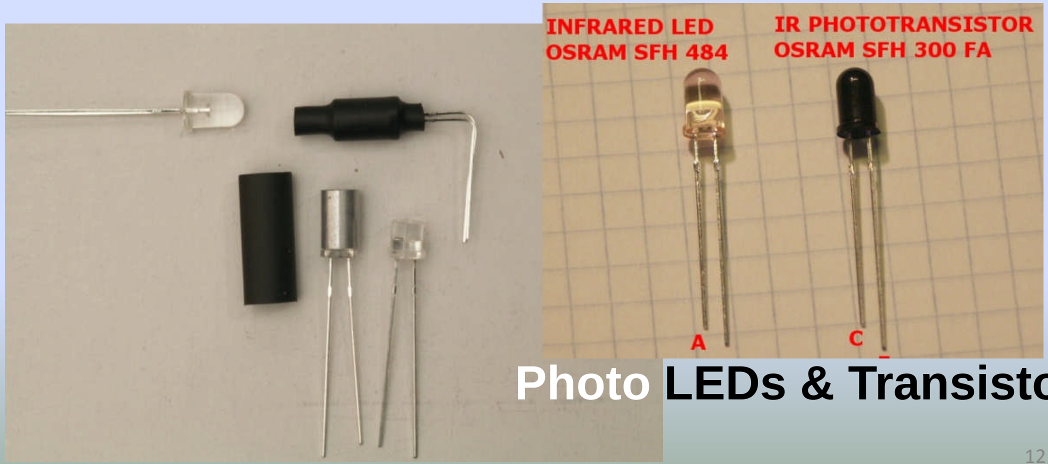
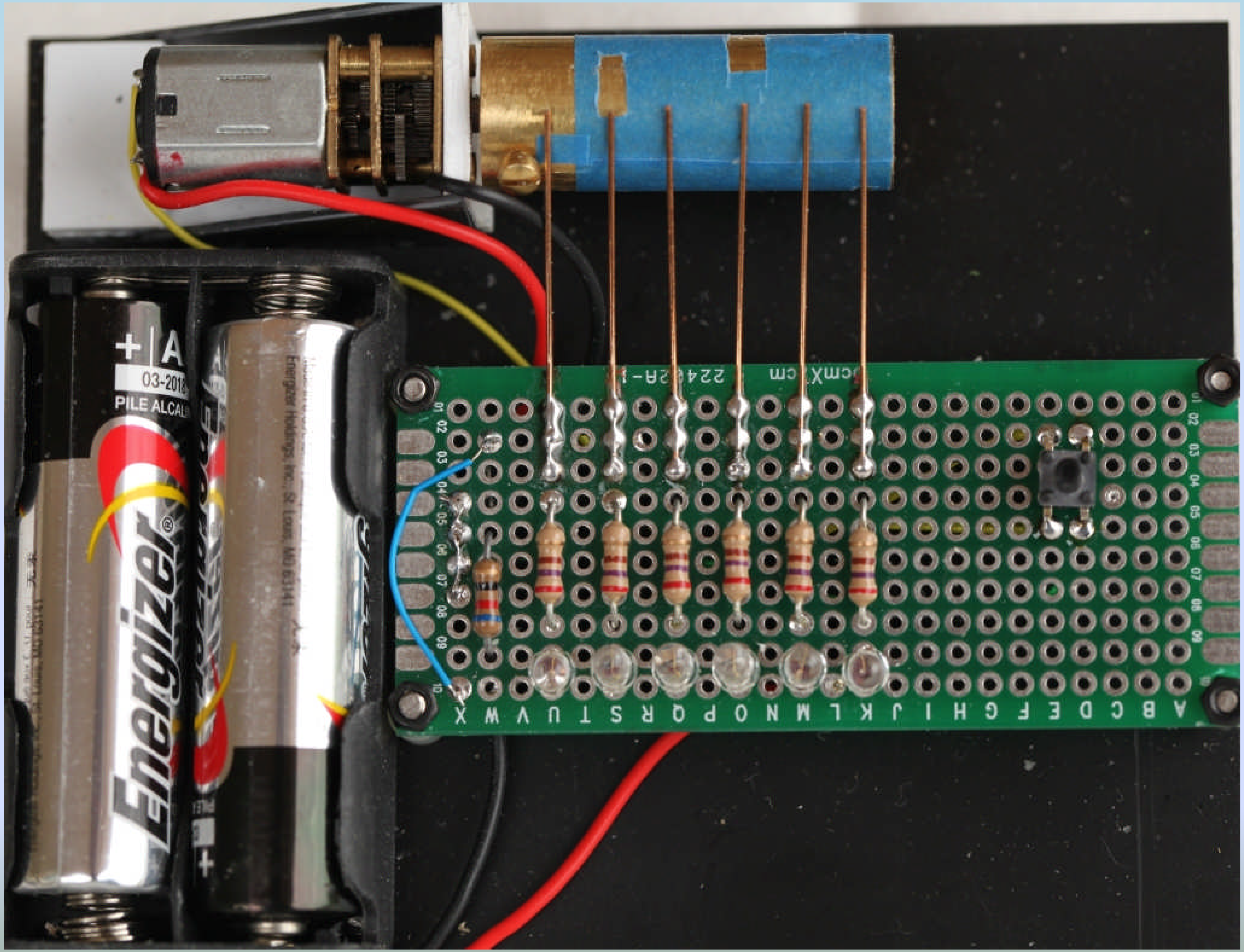
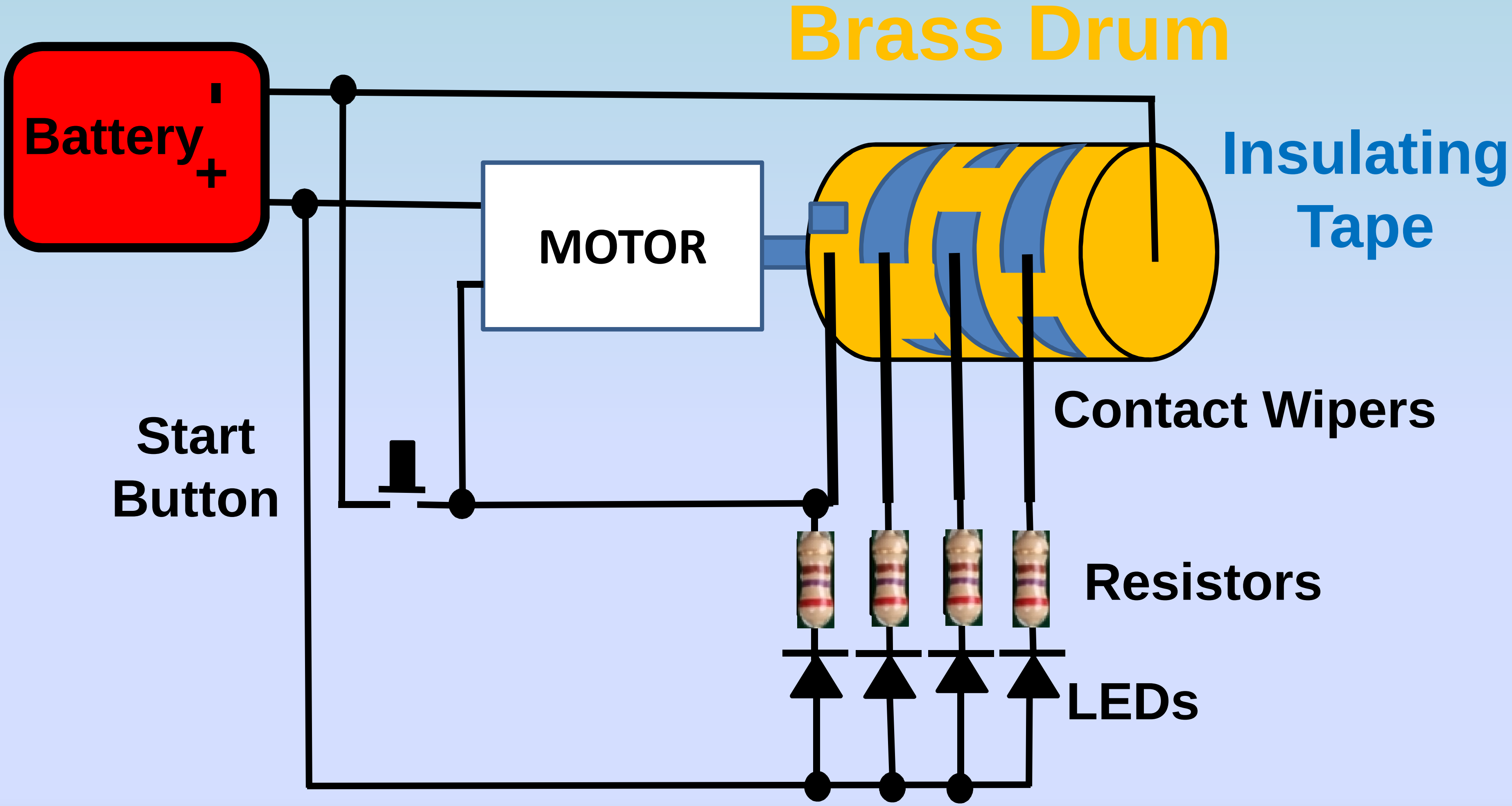


Photo LEDs & Transistors

# *Mechanical Timing & Sequencing*

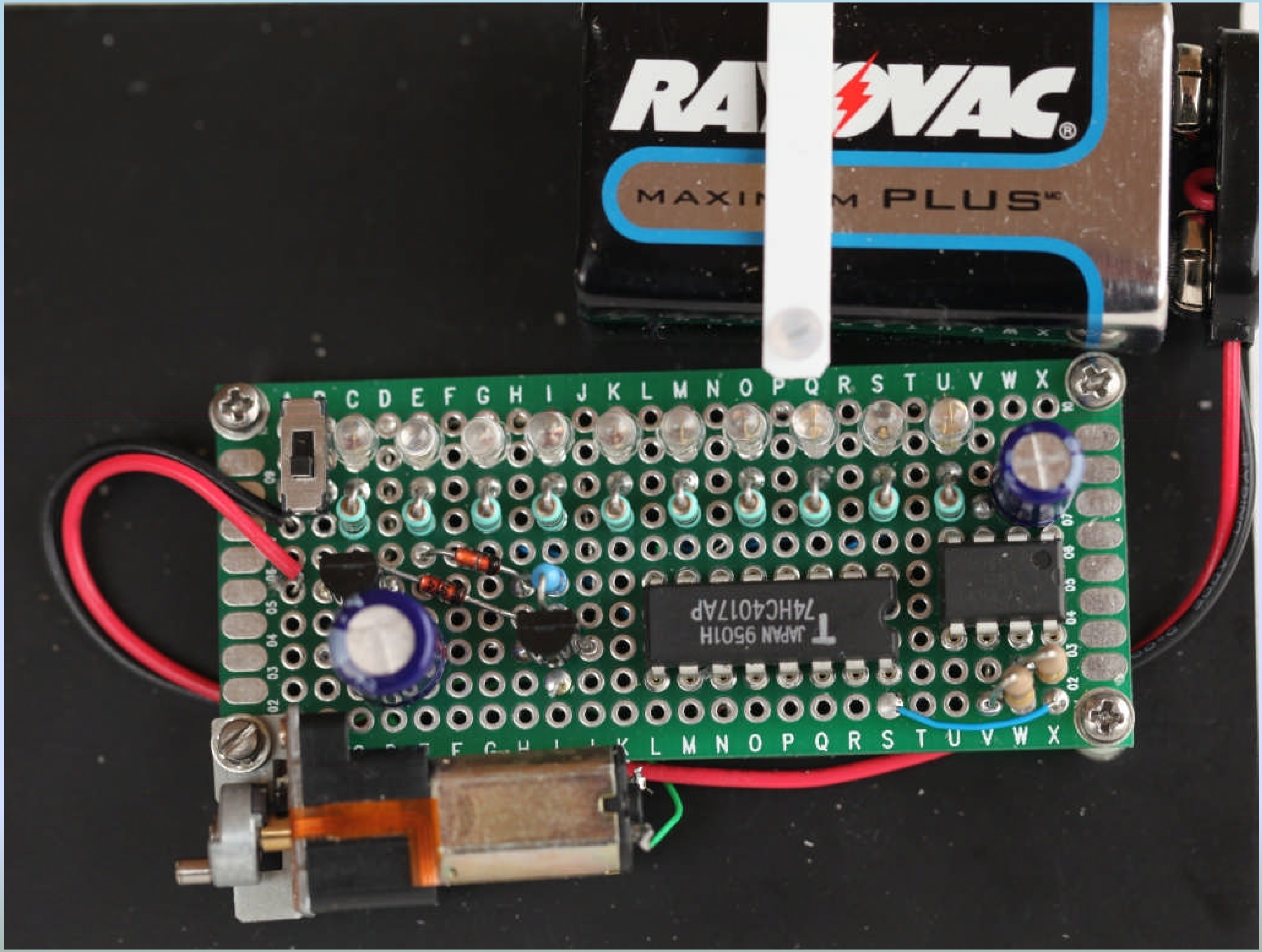


# *Mechanical Timing & Sequencing*

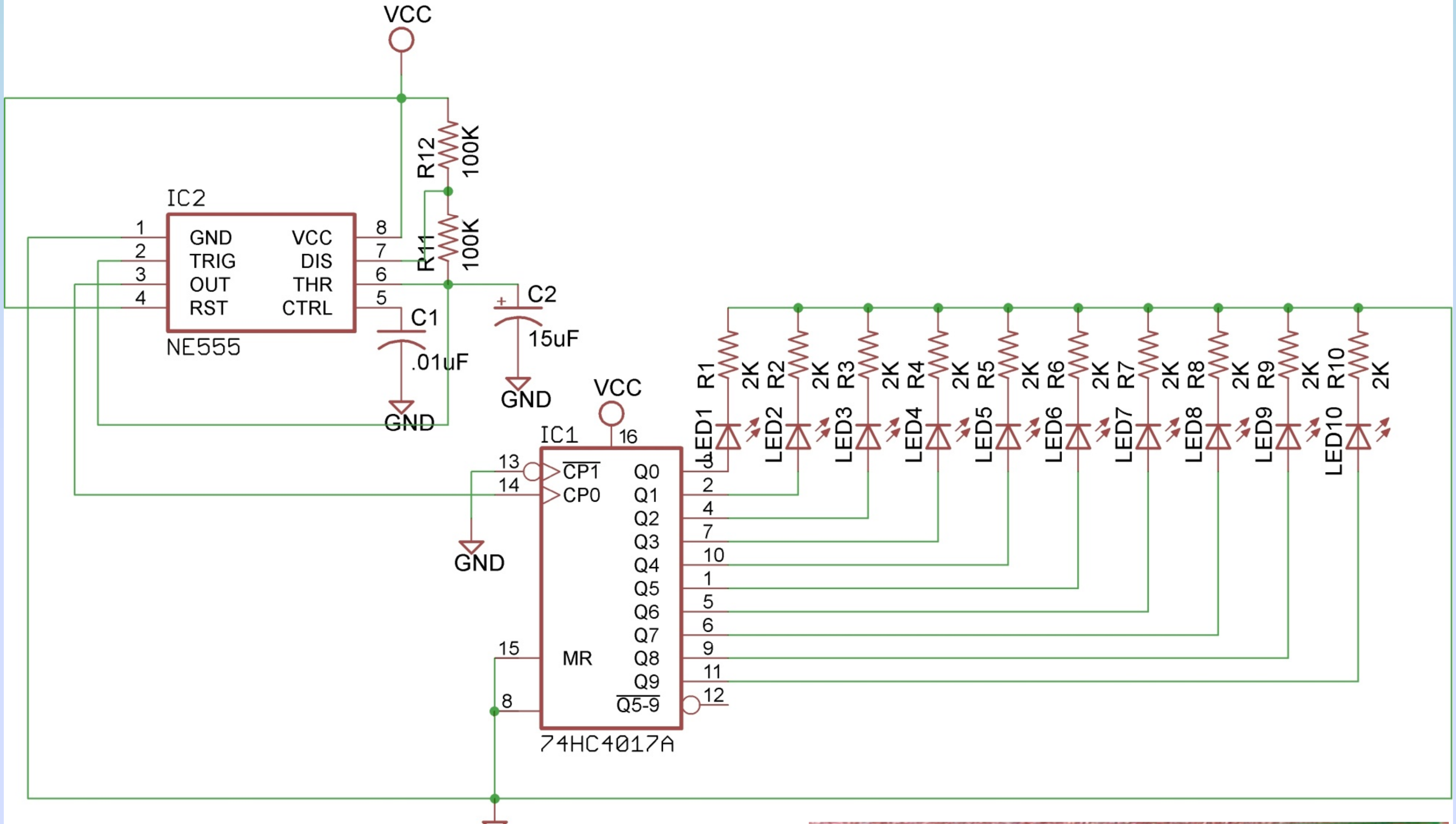


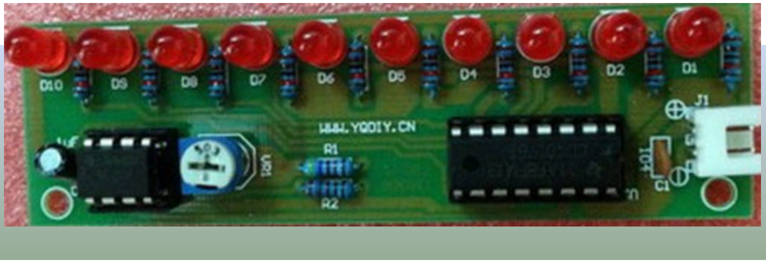
***Start Button Moves Drum Off "Stopping" Tape***
***Wipers Close the Circuit to the LEDs***

# *Electronic Timing & Sequencing*



# Simple Electronic Sequencing





Ebay item-$3.75:     http://tinyurl.com/l7reunu

# *Take a Picture*

THE GREAT ATLANTIC & PACIFIC TEA CO.
A&P
"Where Economy Rules"
UNION REFRIGERATOR TRANSIT CO.
23054

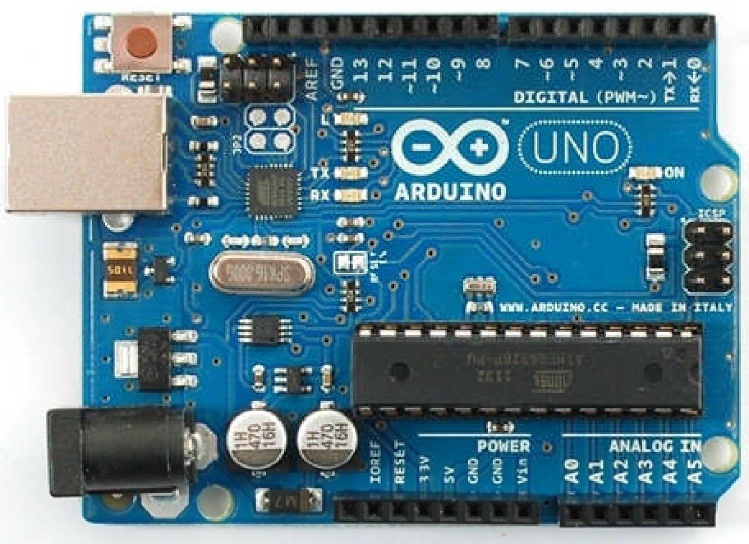
**Take a picture… Wait for train to pass !**

# *Programmed Timing & Sequencing*

- **Why Bother?**
  - **Way More Flexibility Than ANY other Method**
  - **Low Cost**
  - **Easy to Modify/Change**

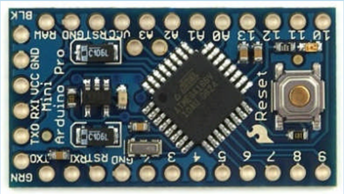
- **Multiple triggers**
- **Multiple Kinds of Triggers**
- **Lights, Motors, Relays, Servos, Sensors**
- **Pseudo Random Sequencing**
- **Different Sequences Can Run Simultaneously**
- **DCC Library Already Written**
- **Support For Sound Generators (MP3, WAV & AD4)**
- **Additional Tutorials, Books, Libraries, Examples**

# *Programmed Timing & Sequencing*



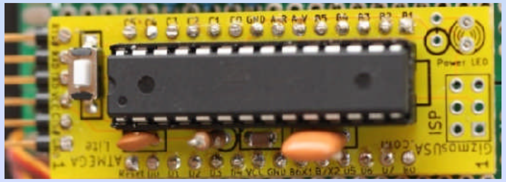
**Arduino Uno $11-$30 Qty1**

**Arduino Pro Mini**

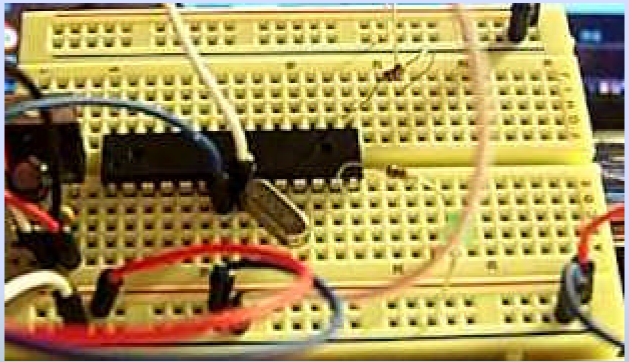


**Digispark**

**~~$3.84~~ $3.68!! Qty 1**
**http://tinyurl.com/oamynj5**



**Atmel 328P on SurplusGizmos.com**
**Atmel Lite Kit  $6.75 Kit Only**



**Solderless Breadboard**

**Atmel ATMega328P**
**Mouser.com $2.24 Qty1**

**Alternatives:**
**MC PIC Processor**
**TI LaunchPad**

http://www.arduino.cc
http://sparkfun.com
http://www.adafruit.com

# The Essentials for Programmed Sequencing

**http://tinyurl.com/oamynj5**

**Arduino Pro Mini-$3.68**

http://tinyurl.com/ld6sgy3

**USB to Pro Mini Cable-$8.48 (Need One)**

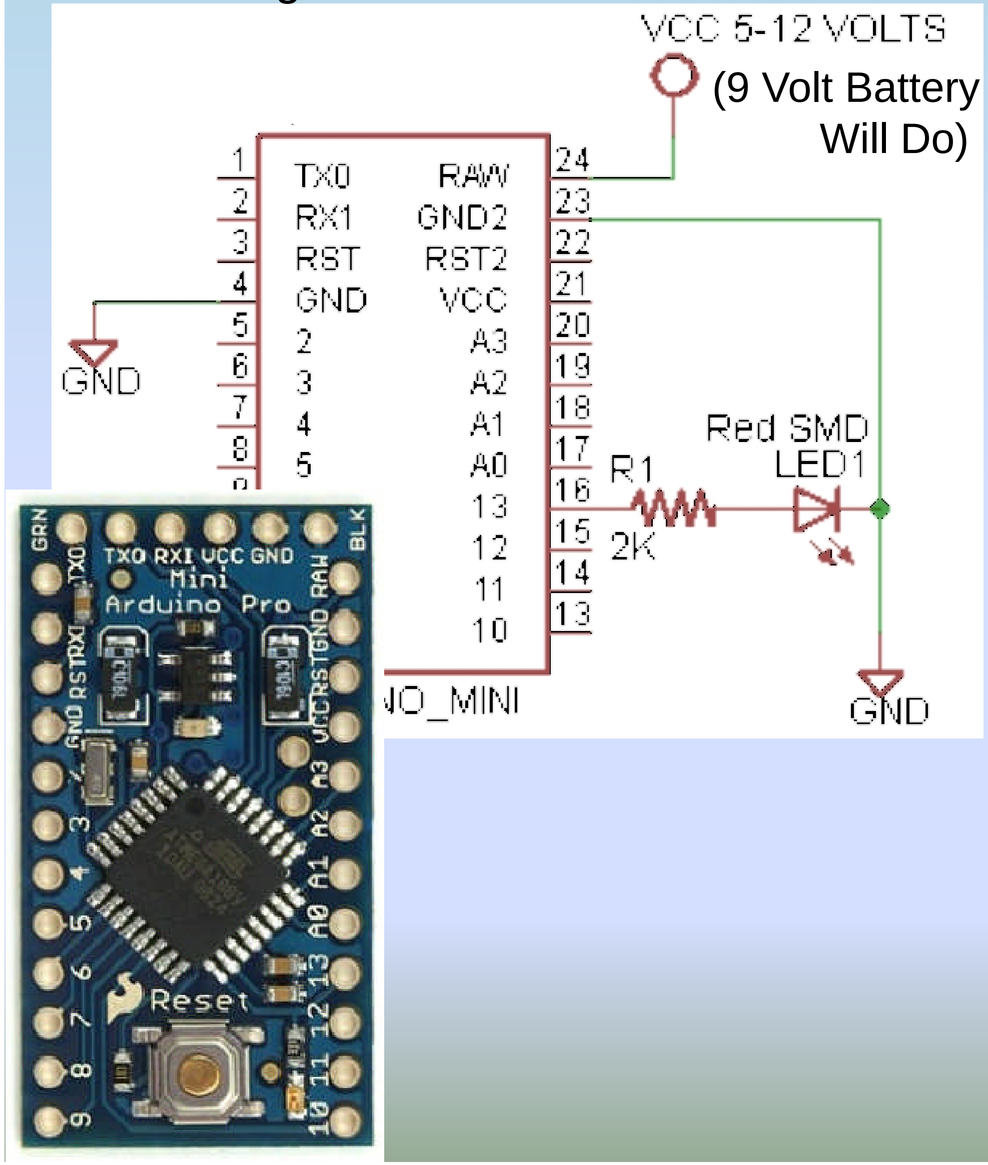
**PLUS**

http://arduino.cc/en/Main/Software
http://www.arduino.cc/

**Download Software for FREE**

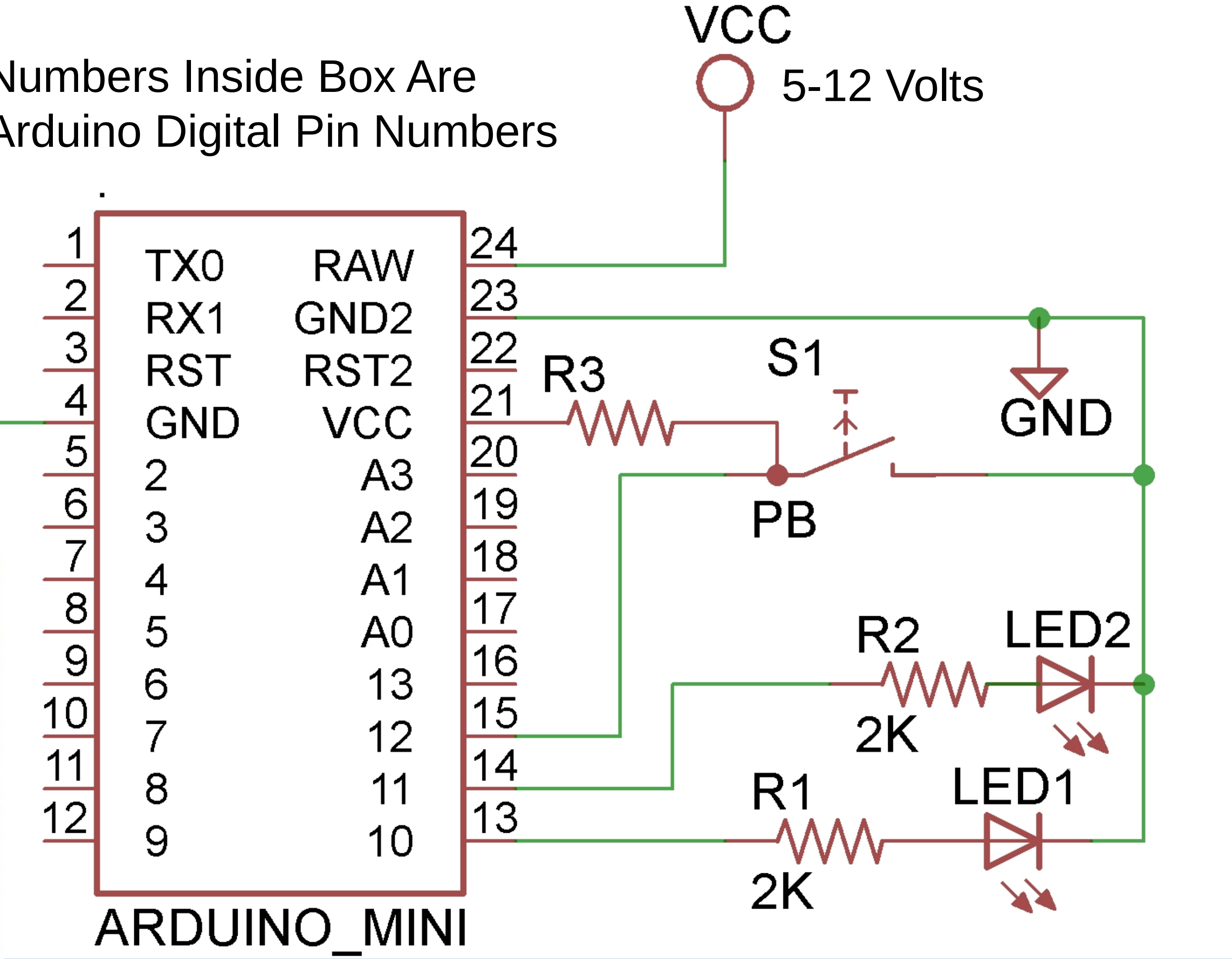
# *Example: Make a LED Blink On and Off*

Numbers Inside The Box Are
Arduino Digital Pin Numbers

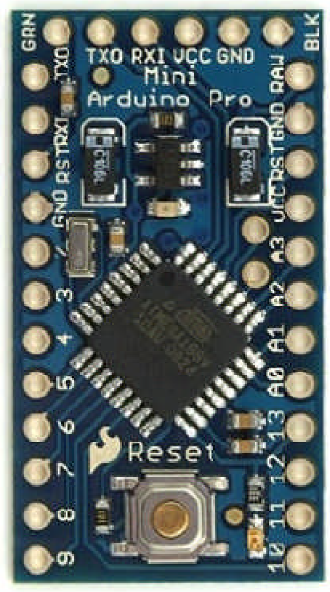


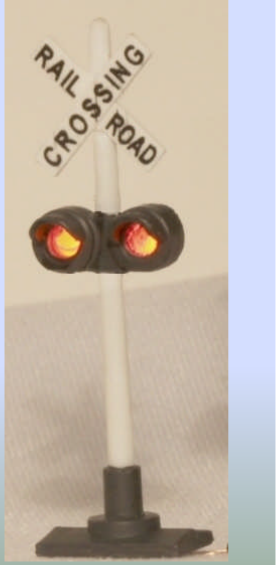
```
// the setup section runs once on power up
void setup()
 {

   // Set up  DigitalPin 13 for Output
   pinMode(13, OUTPUT);
 }

   // the loop section repeats forever
void loop()   //this names this section
 {

    // turn the LED on (sets Voltage HIGH)
   digitalWrite(13, HIGH);

   delay(1000);     // wait 1000 milliseconds

    // turn LED off (sets Voltage LOW)
   digitalWrite(13, LOW);

   delay(1000);     // wait one second
 }
```

# *Example: Triggered Crossbucks Flashers*



Numbers Inside Box Are
Arduino Digital Pin Numbers





# *Example: Triggered Crossbucks Flashers*

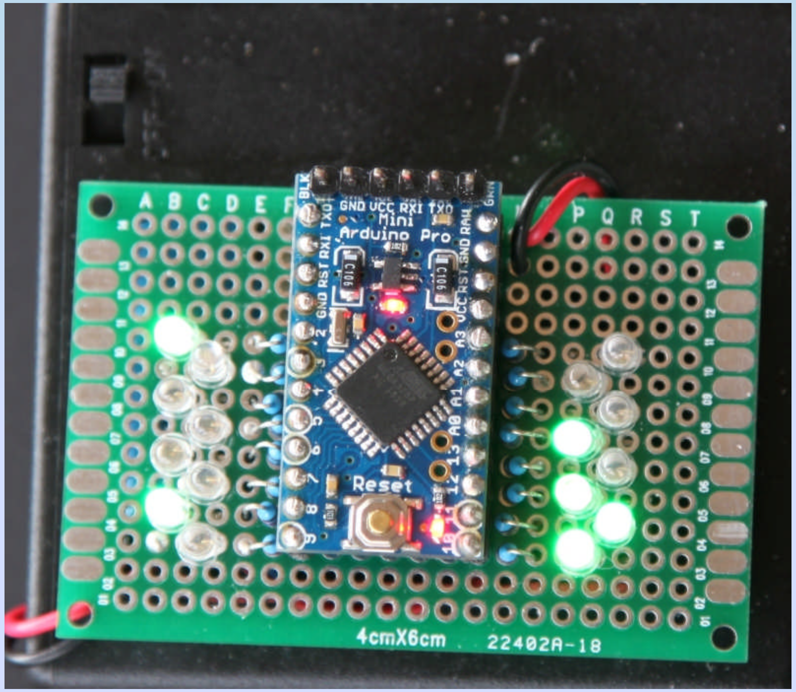
## *First*, Learn to *Read* and *Edit*

```
//  Blink Crossing Signal LEDS
void setup()  {                  //This setup section runs ONCE when you press reset or on power up
                    pinMode(12, INPUT);        // Set up Digital pin 11 as an input
                    pinMode(10, OUTPUT);   // Set up Digital pin 12 as an output
                    pinMode(11, OUTPUT);   // Set up Digital pin 13 as an output
                 }
void loop()                                  // This loop section runs OVER AND OVER again forever
  {  if (digitalRead(12)==LOW)        //Check if the Pushbutton is pressed (LOW=pressed)
       {
                                             // This sequence is run if the button IS  pressed
         digitalWrite(10, HIGH);        // turn one Flasher on (HIGH is the voltage level)
         digitalWrite(11, LOW);         // turn the other Flasher off (LOW is the voltage level)
         delay(1000);                   // wait for a second – 1000 milliseconds
         digitalWrite(10, LOW);         // turn one Flasher off (LOW is the voltage level)
         digitalWrite(11, HIGH);        // turn the other Flasher on(HIGH is the voltage level)
         delay(1000);                   // wait for a second – 1000 milliseconds
       }
     else  {                                 // This sequence is run if the button is NOT pressed
               digitalWrite(10, LOW);   // turn one Flasher off (LOW is the voltage level)
               digitalWrite(11, LOW);   // turn the other Flasher off (LOW is the voltage level)
             }
  }
```

# *Example: Triggered Crossbucks Flashers*

```
//  Blink Crossing Signal LEDS 9 times for each press of the button
void setup()  {                 //This setup section runs ONCE when you press reset or on power up
                  pinMode(12, INPUT);      // Set up digital pin 11 as an input
                  pinMode(10, OUTPUT);  // Set up digital pin 12 as an output
                  pinMode(11, OUTPUT);  // Set up digital pin 13 as an output
               }
void loop()                                    // This loop section runs OVER AND OVER again forever
  {  if (digitalRead(12)==LOW)        //Check if the Pushbutton is pressed (LOW=pressed)
       for (int i=1; i<10; i++)             // Repeat the following sequence 9 times
       {
                                               // This sequence is run if the button IS  pressed
         digitalWrite(10, HIGH);       // turn one Flasher on (HIGH is the voltage level)
         digitalWrite(11, LOW);        // turn the other Flasher off (LOW is the voltage level)
         delay(1000);                         // wait for a second – 1000 milliseconds
         digitalWrite(10, LOW);        // turn one Flasher off (LOW is the voltage level)
         digitalWrite(11, HIGH);        // turn the other Flasher on(HIGH is the voltage level)
         delay(1000);                         // wait for a second – 1000 milliseconds
       }
      else  {                                  // This sequence is run if the button is NOT pressed
              digitalWrite(10, LOW);  // turn one Flasher off (LOW is the voltage level)
              digitalWrite(11, LOW);  // turn the other Flasher off (LOW is the voltage level)
           }
  }
```

# *Example: Lighted Windows*

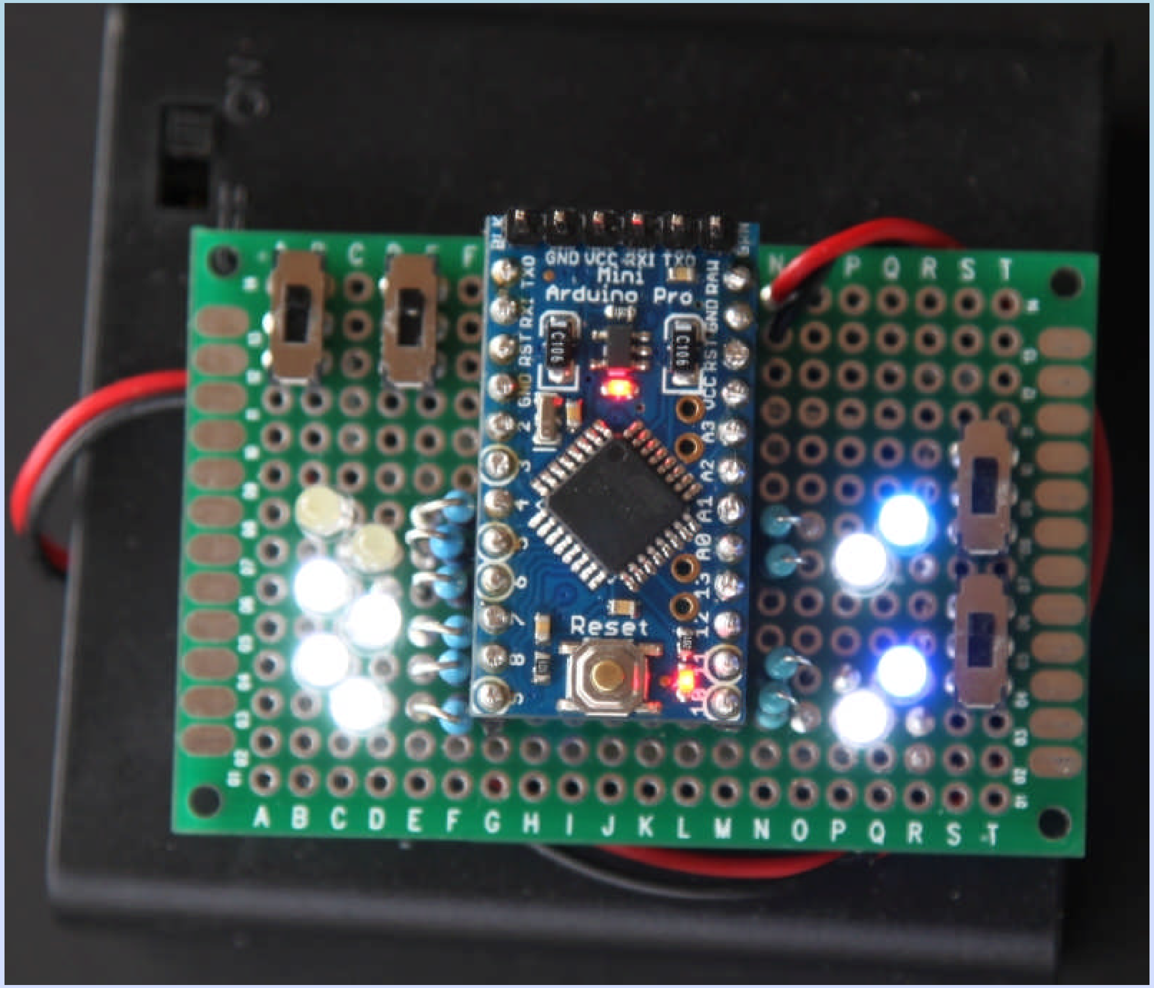


```
//  Blink_Everything
int tim_delay = 900;
#define numleds  16
byte ledpins [] = {0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,0};
void setup() {
    // initialize the digital pins as an outputs
    randomSeed(analogRead(2));
    for (int i=1; i<= numleds; i++) {
    pinMode(ledpins[i], OUTPUT);
    digitalWrite(ledpins[i], HIGH);
   }
}
void loop() {
    digitalWrite(ledpins [random (0,numleds+1)], lightsw() );
    delay(tim_delay);              // wait for a bit
}
boolean lightsw() {
  if (random(0,100)>41) return LOW;
   else return HIGH;
}
```

Program “Clinic_All_Building.ino” is in:

**http://home.comcast.net/~gbglacier/Clinics/AnimationControl_Clinic_adds.zip**

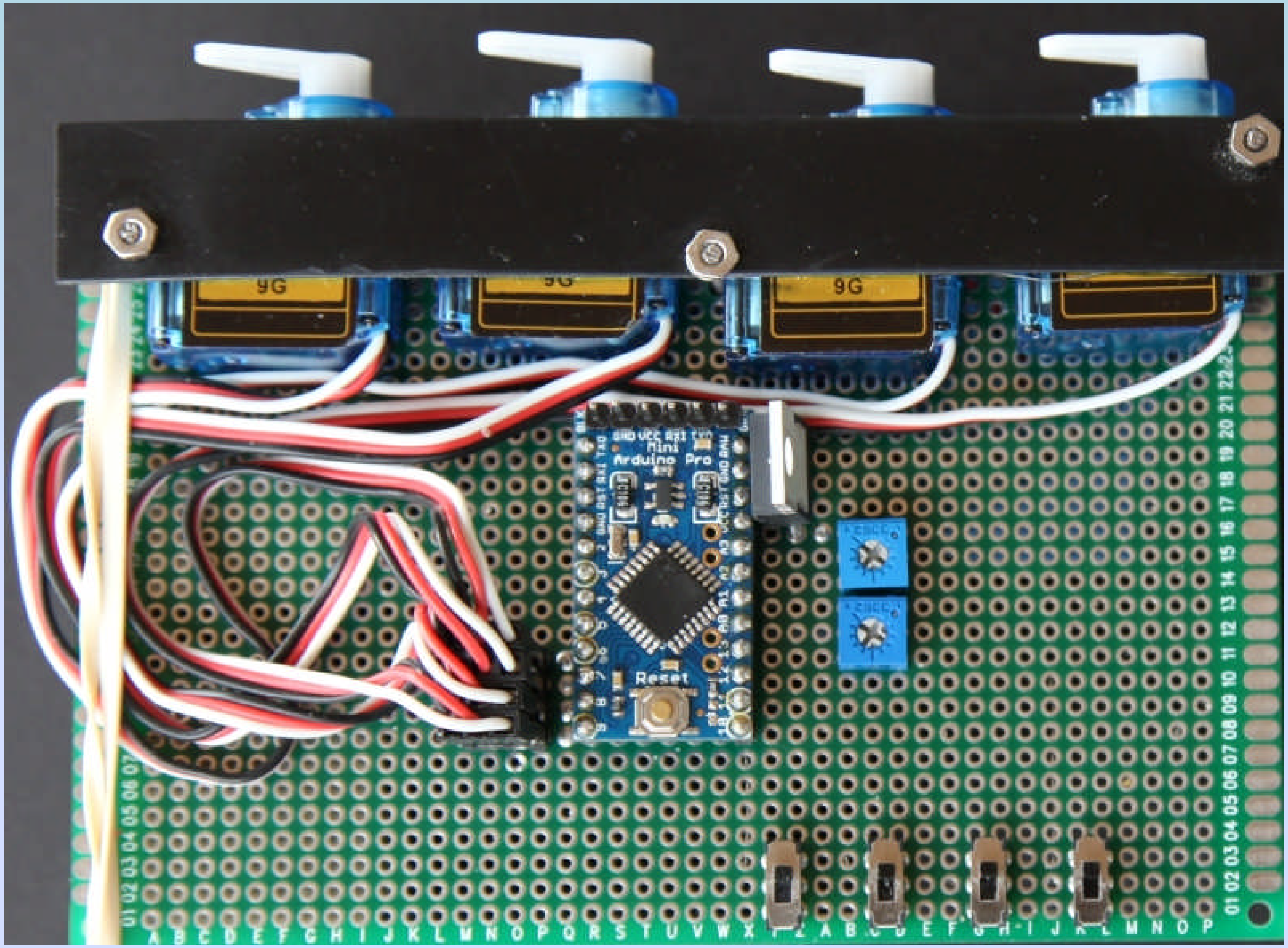
# *Example: Welders and Fluorescents*



Program “Clinic_Welder_FL3.ino” is in:

**http://home.comcast.net/~gbglacier/Clinics/AnimationControl_Clinic_adds.zip**

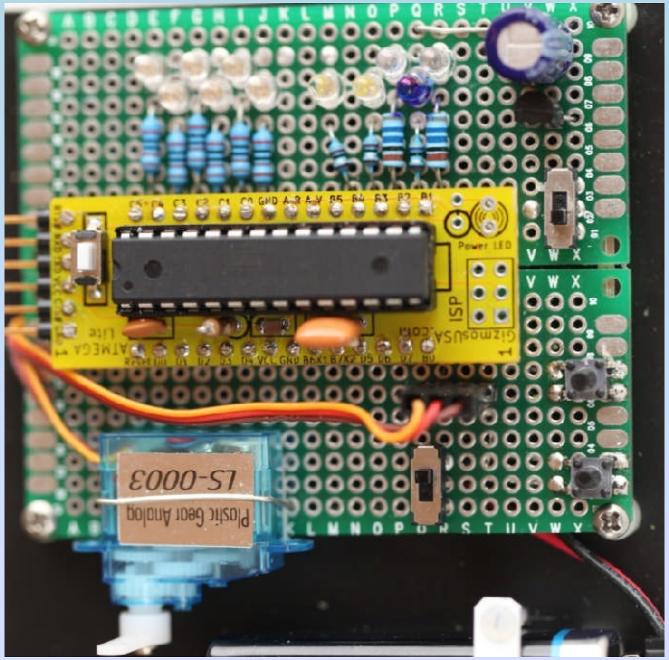
# *Example: Servo Drivers*



Program “Clinic_ServoTestSet_4_switched_slow.ino” is in:

**http://home.comcast.net/~gbglacier/Clinics/AnimationControl_Clinic_adds.zip**

# *Programmed Timing & Sequencing*



- **Different Sequences Occurring Simultaneously**
- **Pseudo Random Sequencing**
- **Multiple triggers**
- **Lights, Motors, Servos, Sensors**

Program “Clinic_Lighting_Demo_1b.ino” is in:
**http://home.comcast.net/~gbglacier/Clinics/AnimationControl_Clinic_adds.zip**

# *Building Lights, Sound, & Motion*

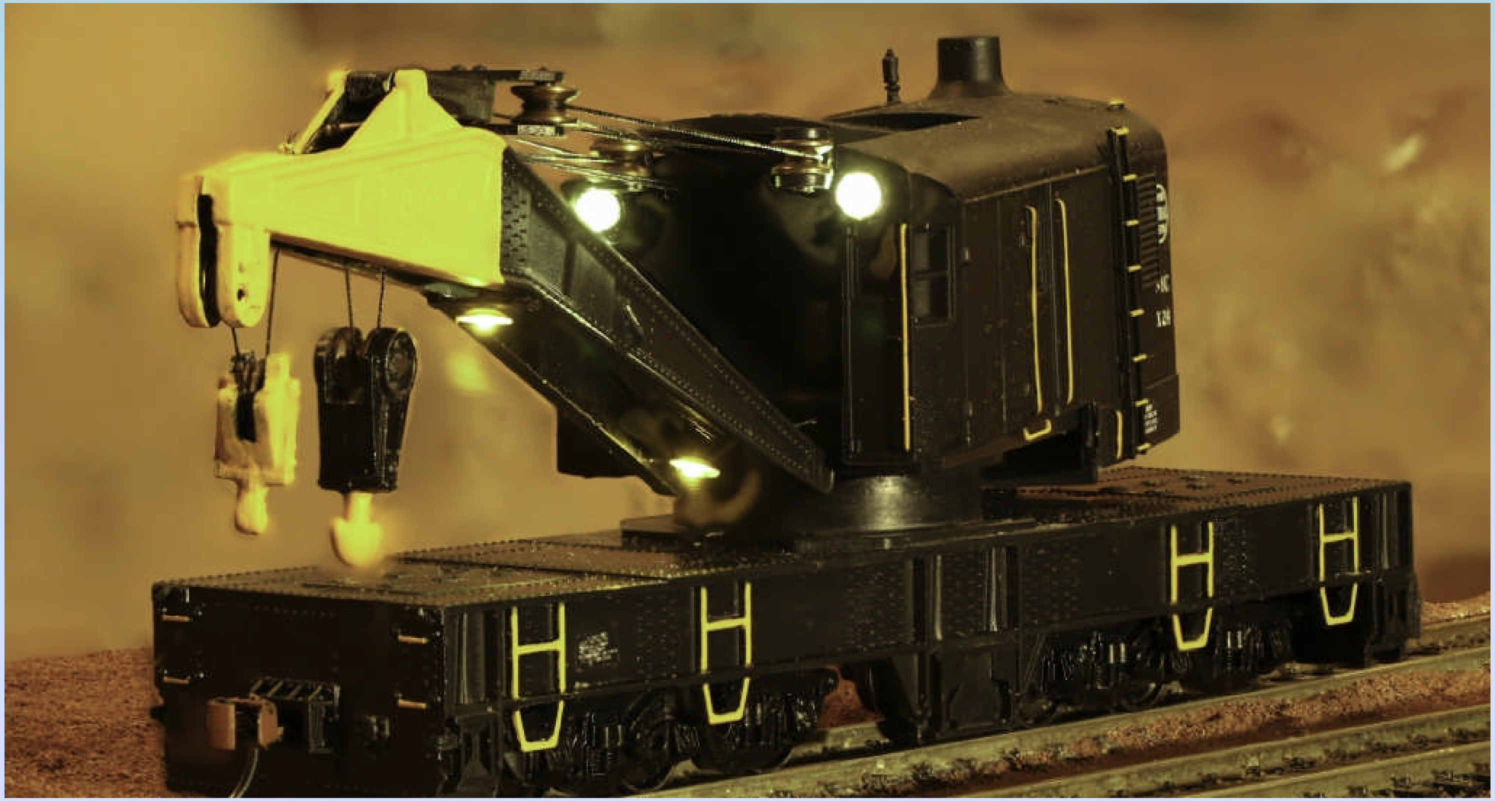
# *DCC Timing, Cueing, & Synchronization*

**Using JMRI Scripts one can get:**

- **Repeatable Sequencing**
- **Repeatable Timing**
- **Cueing Among Functions, Sound, and Movements**

*******On an "*Isolated*" DCC system*******

# *With DCC Cueing: Animate a DCC Crane*



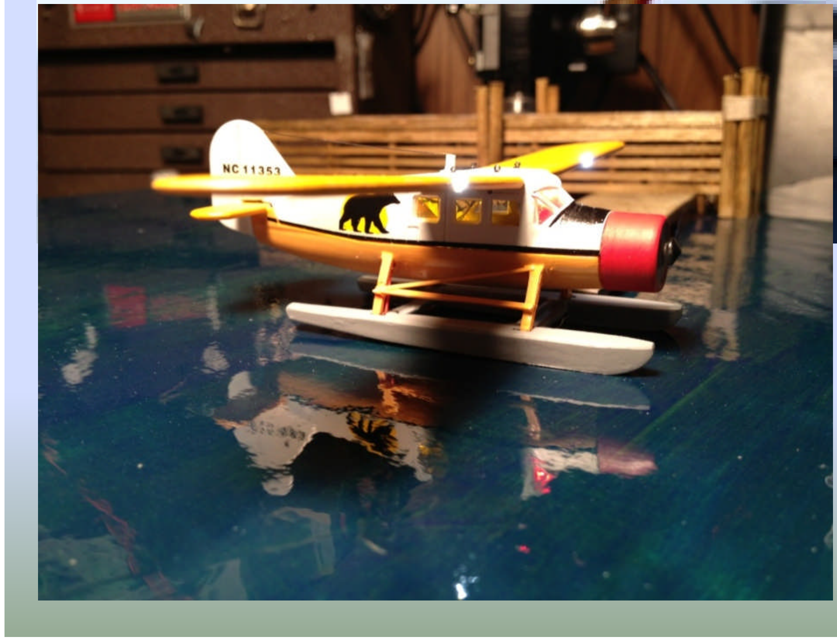
- **0402 LEDs Under Boom and 0603 Spotlight LEDs**
- **3 degrees of movement: CAB, Boom, Main Hook**
- **Prototypical Steam Boiler, Whistle, Bell, and Mechanical sounds**
- **We need to time and sequence a long set of DCC commands**

# Sample JMRI DCC Crane Script for Animation

```
##### Set Speed to Forward, stopped
self.throttle.speedSetting = 0.
self.throttle.setIsForward(True)
self.waitMsec(self.delay*400)

############## F8 on, forward"
self.throttle.setF8(True);
self.throttle.setIsForward(True)
self.waitMsec(self.delay*50)

############## F6 rear light on
self.throttle.setF6(True);
self.waitMsec(self.delay*500)

############## Mute (F8) off"
self.throttle.setF8(False);
self.throttle.setIsForward(True)
self.waitMsec(self.delay*1000)

######### Whistle
self.throttle.setF2(True);
self.waitMsec(self.delay*500)
self.throttle.setF2(False);
```

```
######### Crane Boom Up, 20%
self.throttle.speedSetting = 0.20
self.throttle.setIsForward(True)
self.throttle.setF3(True);
self.waitMsec(self.delay*4500)
self.throttle.setF3(False);
self.waitMsec(self.delay*1500)

########## Whistle
self.throttle.setF2(True);
self.waitMsec(self.delay*500)
self.throttle.setF2(False);
########## Turn Cab CW, 11%"
self.throttle.speedSetting = 0.11
self.throttle.setIsForward(True)
self.throttle.setF1(True);
self.waitMsec(self.delay*4000)
self.throttle.setF1(False);
self.waitMsec(self.delay*2000)

########## Whistle
self.throttle.setF2(True);
self.waitMsec(self.delay*500)
self.throttle.setF2(False);
```

# *Synchronized Light, Sound, & Motion*



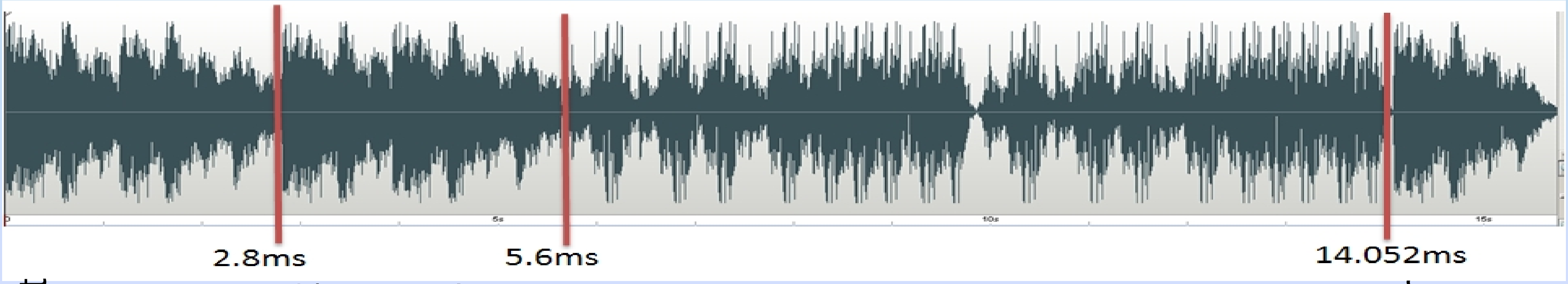
**DCC**

**&**

**Direct**

# *Tying Motion to Sound*

Floatplane Engine Sound Clip as shown in Sound Editor
(Audacity, Nero Wave Editor, WavePad Sound Editor, etc.



Engine Start

Engine Sputter

Engine Start

Engine Sputter

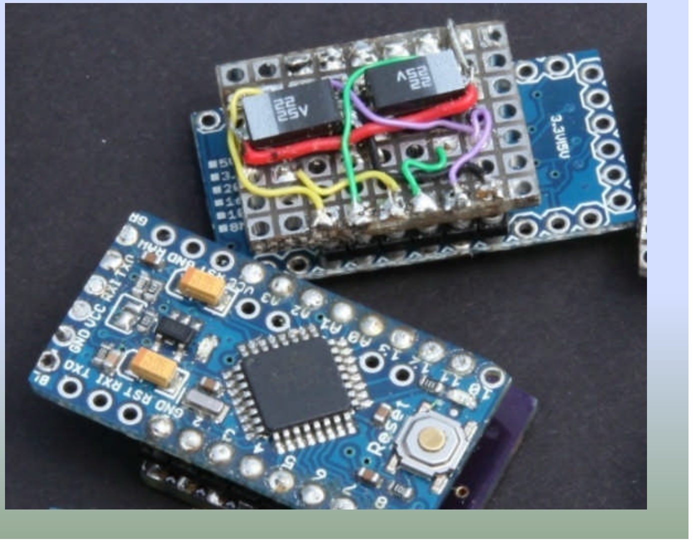
Rough Running

Engine Sputter

**These events match the delays in the script following**
**The script provides the necessary synchronization throughout the sound clip**

# *Programmed Control & DCC Control*

- **Adding the figure enhances and complicates the animation**
- **DCC Software Library (CmdrArduino) for Arduinos (Downloadable from: www.Railstars.com)**
  **Gives you *direct* DCC control**
- **Simultaneous Direct Control of All Pins**
- **Finer Control of Timing and Functions**
- **Using Arduino Pro Mini & TI SN754410 Driver**



# *Programmed Control & DCC Control*

## Sequences get more complicated, but so does the action

```
functs ^= funct4;
dps.setFunctions0to4(locoAdr,DCC_SHORT_ADDRESS,functs); // F4 rear cabin lights on
wait_dcc(4000);          //Wait 4 seconds
functs =functs|funct3;
dps.setFunctions0to4(locoAdr,DCC_SHORT_ADDRESS,functs); // F3 front cabin lights on
wait_dcc(4000);          //Wait 4 seconds
functs =functs|funct0;
dps.setFunctions0to4(locoAdr,DCC_SHORT_ADDRESS,functs); // F0 Landing Lights on
wait_dcc(4000);          //Wait 4 seconds
functs = functs|funct2;
dps.setFunctions0to4(locoAdr,DCC_SHORT_ADDRESS,functs); // F2 start motor sound
wait_dcc(1);          //Wait for the queue
dps.setSpeed128(locoAdr,DCC_SHORT_ADDRESS,high_rpm); // Speed high_rpm speed
wait_dcc(1425);       //Wait 1.425 seconds
dps.setSpeed128(locoAdr,DCC_SHORT_ADDRESS,low_rpm); // Speed low_rpm
wait_dcc(1400);         //Wait 1.4 seconds
dps.setSpeed128(locoAdr,DCC_SHORT_ADDRESS,high_rpm); // Speed high_rpm
wait_dcc(1425);         //Wait 1.425 seconds
dps.setSpeed128(locoAdr,DCC_SHORT_ADDRESS,low_rpm); // Speed low_rpm
wait_dcc(1400);         //Wait 1.4 seconds
dps.setFunctions0to4(locoAdr,DCC_SHORT_ADDRESS,functs); // F2 refresh motor sound
dps.setSpeed128(locoAdr,DCC_SHORT_ADDRESS,high_rpm); // Speed .high_rpm
wait_dcc(629);
digitalWrite(arm_pin1, 1);
wait_dcc (4940);
digitalWrite(arm_pin1, 0);
wait_dcc(2883);
functs =funct3|funct4|funct0; // Leave only the lights on
dps.setFunctions0to4(locoAdr,DCC_SHORT_ADDRESS,functs); // F2 motor sound off
```

# *You can carry this along quite a way!*

**Wire-Guided Mobile Crane**

# *Useful Links and Sources:*

***Electronics Parts sources:***

http://www.digikey.com -- Commercial first quality parts<br>
http://www.mouser.com -- Commercial first quality parts<br>
http://www.allelectronics.com -- Surplus Parts, LEDs, motors<br>
http://www.surplusgizmos.com – Surplus parts<br>
http://www.goldmine-elec-products.com -- Surplus parts<br>
http://stores.ebay.com/ledbaron -- Wired SMD micro LEDs<br>
http://www.ngineering.com -- Wire LEDs & tools<br>
http://stores.ebay.com/tech-fixx -- 38 Gauge Wire Source:

***Arduino parts of all sorts:***

http://sparkfun.com -- Stuff for sale & reference material<br>
http://www.adafruit.com -- Stuff for sale & reference material<br>
http://ebay.com/ -- Lowest cost Pro Mini boards

***Arduino reference material:***

http://www.arduino.cc -- Arduino website: libraries, tutorials & reference<br>
http://railstars.com/software/cmdrarduino -- Free DCC Software Library for Arduinos<br>
http://tinyurl.com/mvqmrlj -- ModelRailroadingWithArduino2.pdf -- John Plocher Clinic Slides from 2011 NMRA Convention in Sacramento, CA

***This Presentation:***

**http://home.comcast.net/~gbglacier/Clinics/AnimationControl_Clinic_1.4.pdf**

***Clinic Supplemental Materials:***

**http://home.comcast.net/~gbglacier/Clinics/AnimationControl_Clinic_adds.zip**

**Here are links to some of my animation articles which are all in free online Modeling Magazines:**

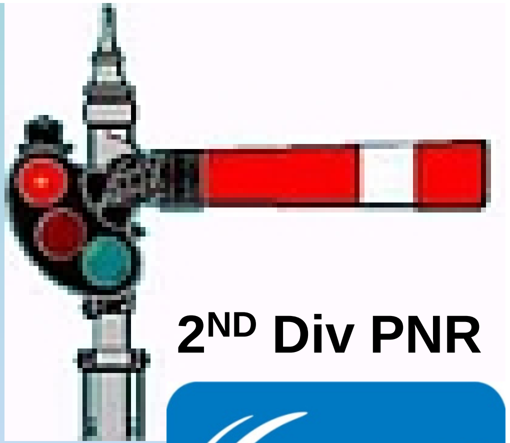
**Using Micro LEDs on Your Layout**
http://model-railroad-hobbyist.com/magazine/mrh-2012-02-feb (Page 74)
**Crossbucks and Crossing Gates**
http://www.nycshs.org/pdf/NYCM_2Q2012.pdf (Page 24)
**Bring a Wrecking Crane to Life**
http://tinyurl.com/mjtwdp4 (Cover Article, Page 53)
**Teach Your Engineers to Turn Heads**
http://publ.com/5nm2Wxy#/100
**Scale Flash Photography**
http://mrhpub.com/2013-07-jul/land/#/51

**You can read my Scale Model Animation blog here:**
http://model-railroad-hobbyist.com/blog/geoff-bunza

**and you can see additional videos on my YouTube channel on animation here:**
http://www.youtube.com/user/DrGeoffB

**I hope you enjoy them too!**

# Control for Scale Model Animation Including DCC



2$^{ND}$ Div PNR

Geoffrey Bunza gbglacier@comcast.net.net

1.4 http://home.comcast.net/~gbglacier/Clinics/AnimationControl_Clinic_1.4.pdf