

# Interrupts & Using ICs

Press Play: Interactive Device Design | May 4, 2010

# Before Integrated Circuits

## Mechanical Computing

1849



Wikimedia, by Joe D in  
January 2005.

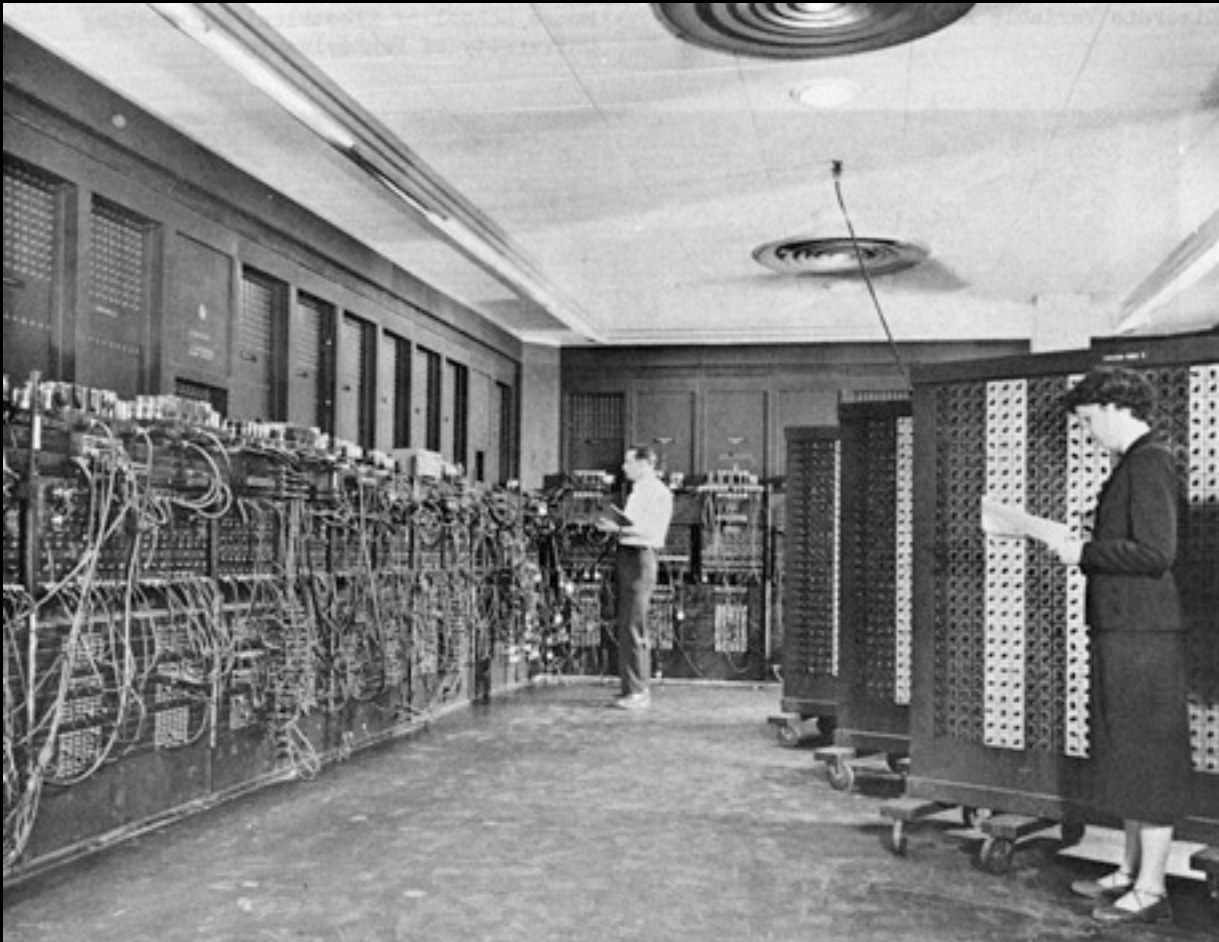


Wikimedia, Carsten Ullrich, 2005

# Before Integrated Circuits

Vacuum Tubes

1945

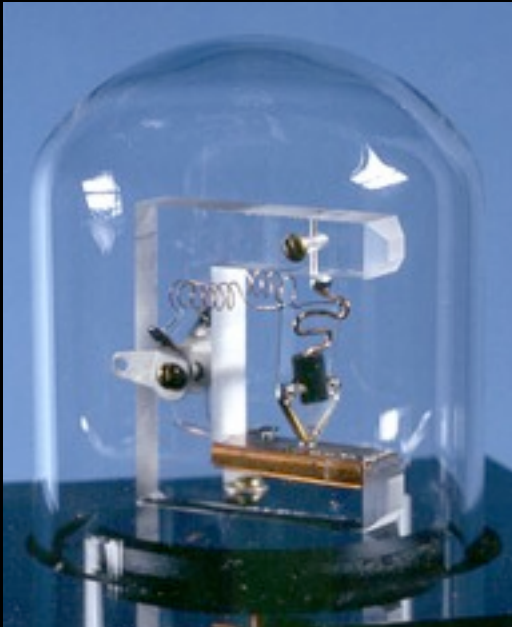


Wikimedia, US Army Photo



# Before Integrated Circuits

## Transistors



1947

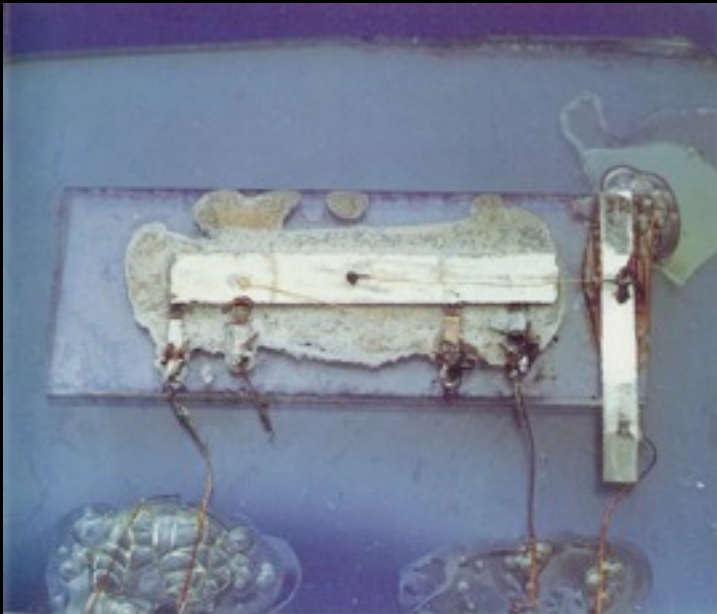
1955





# Integrated Circuits

## Multiple Components on a Single Substrate



STATE OF THE ART ©Copyright Stan Augarten

1958

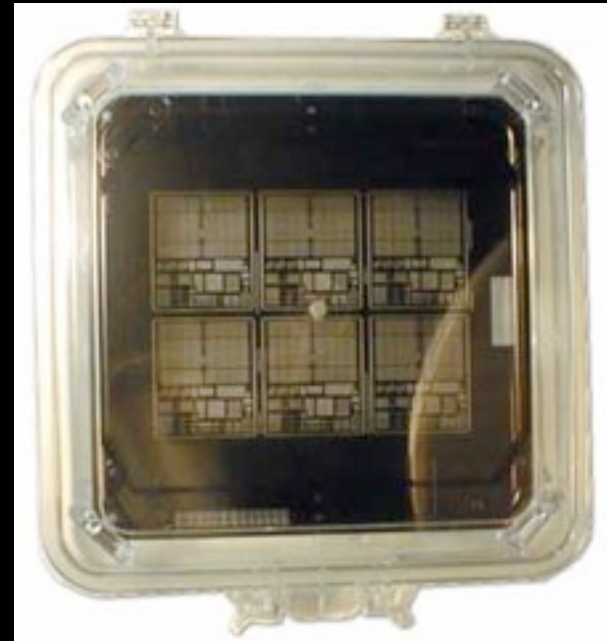


STATE OF THE ART ©Copyright Stan Augarten

1961

# Integrated Circuits Manufacturing

## Similar to Silkscreening



# Analog vs Digital

"Real World" vs "Computer World"

## Analog

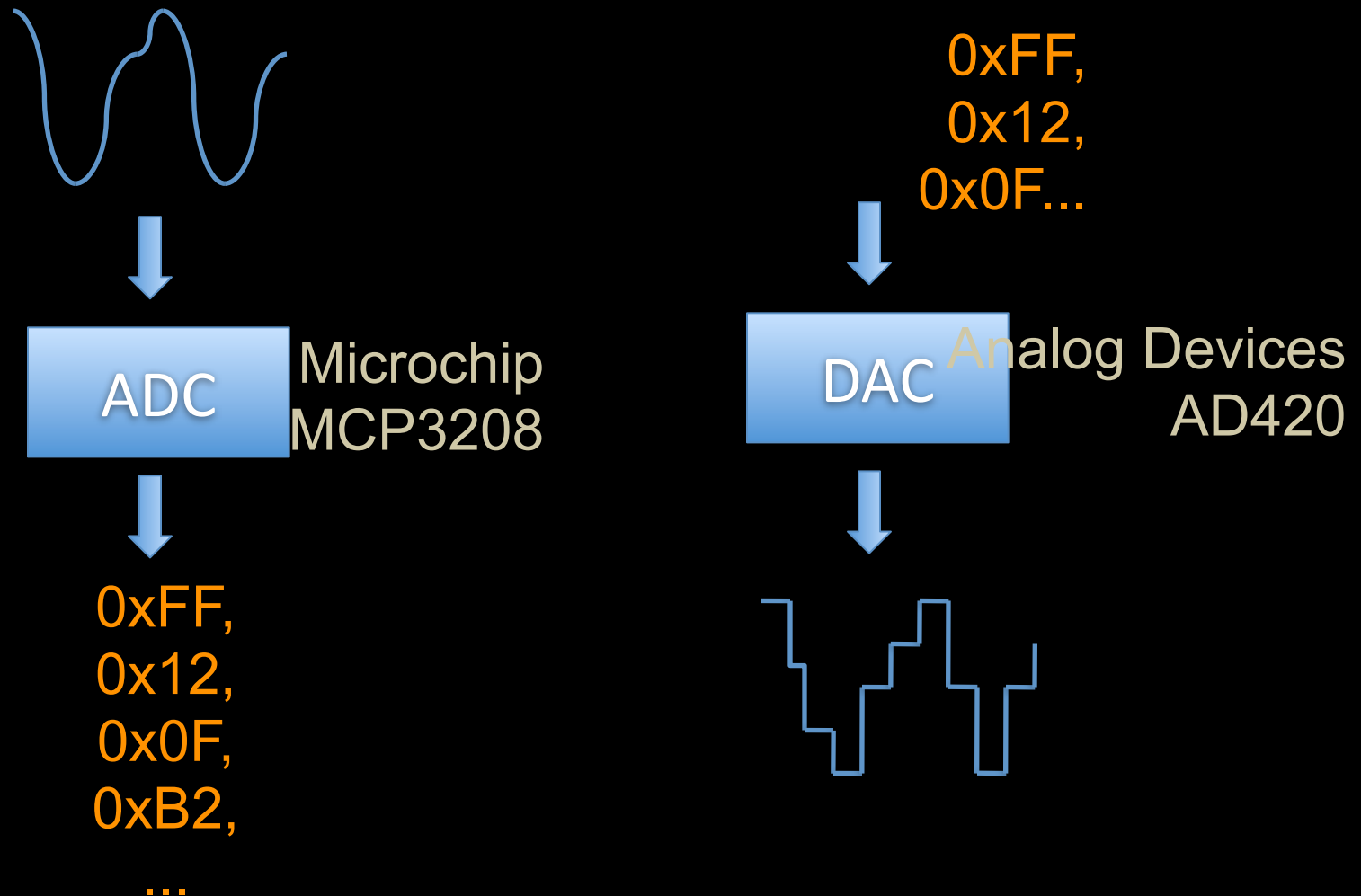


## Digital



# ADCs and DACs

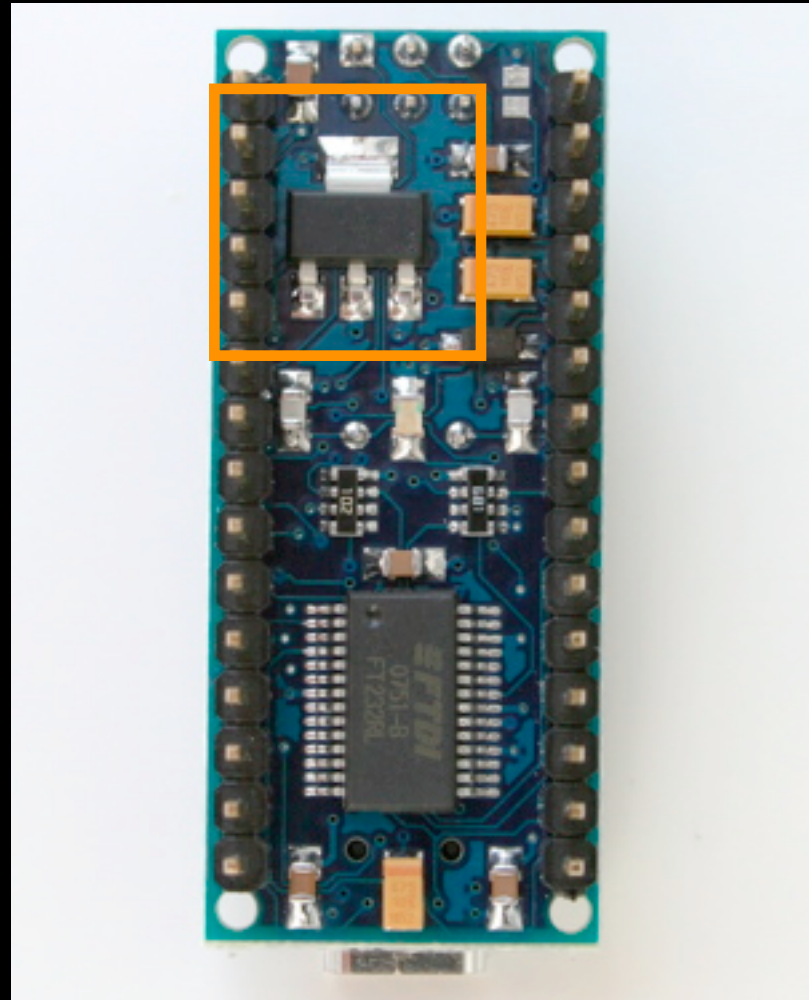
Converting from Analog to Digital, and Back again





# Analog ICs

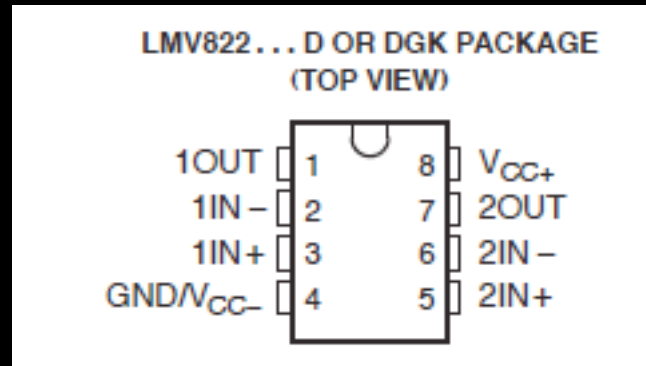
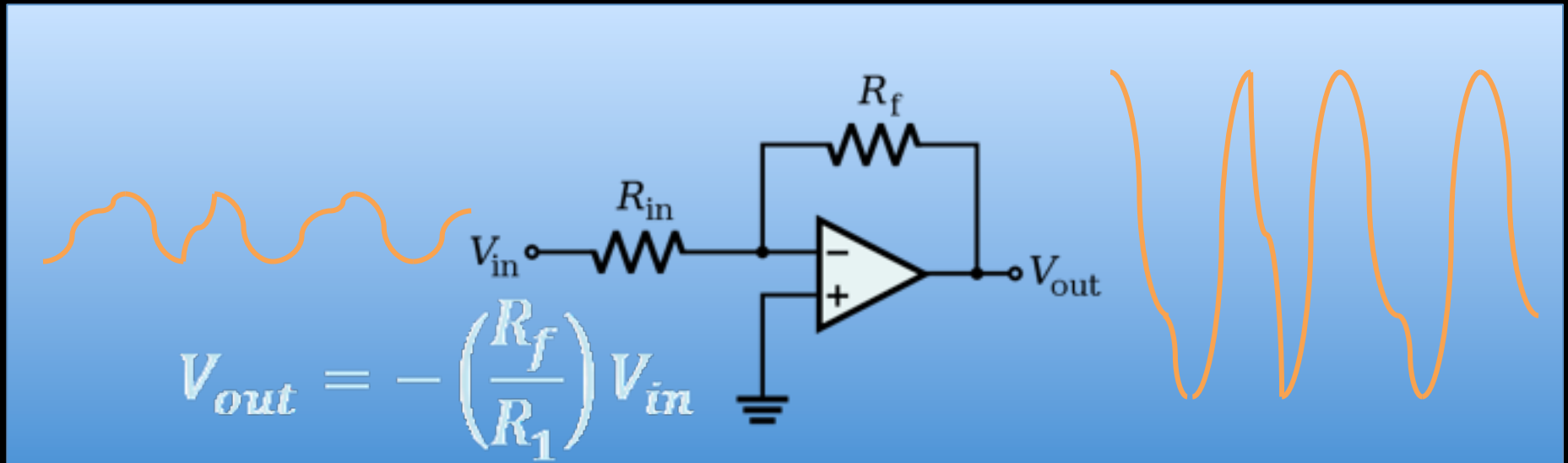
You have already used a few!



# Analog ICs

## Operational Amplifier





Texas Instruments  
LMV822

# Analog ICs

## Filter Chips



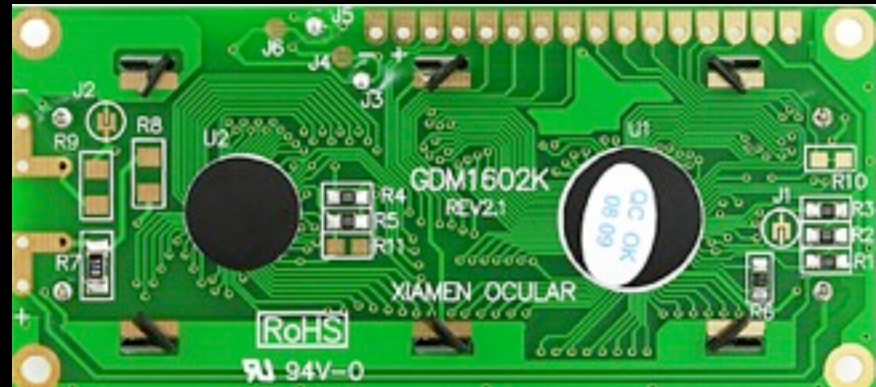
or



Can design your own with R's and C's

# Digital ICs

You've already used a ton!



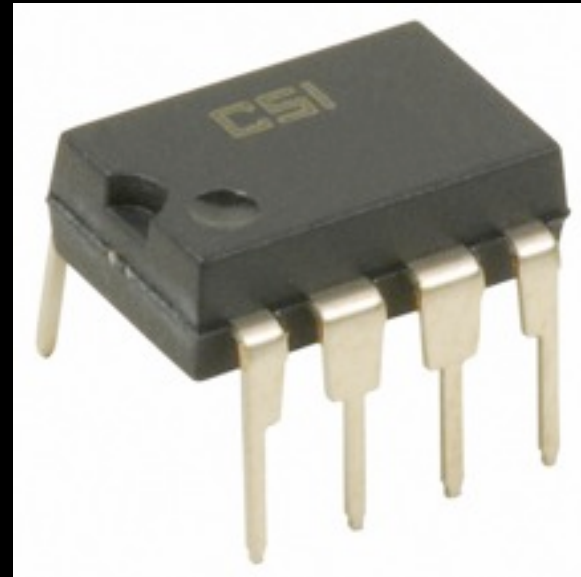


# Digital ICs

## Memory



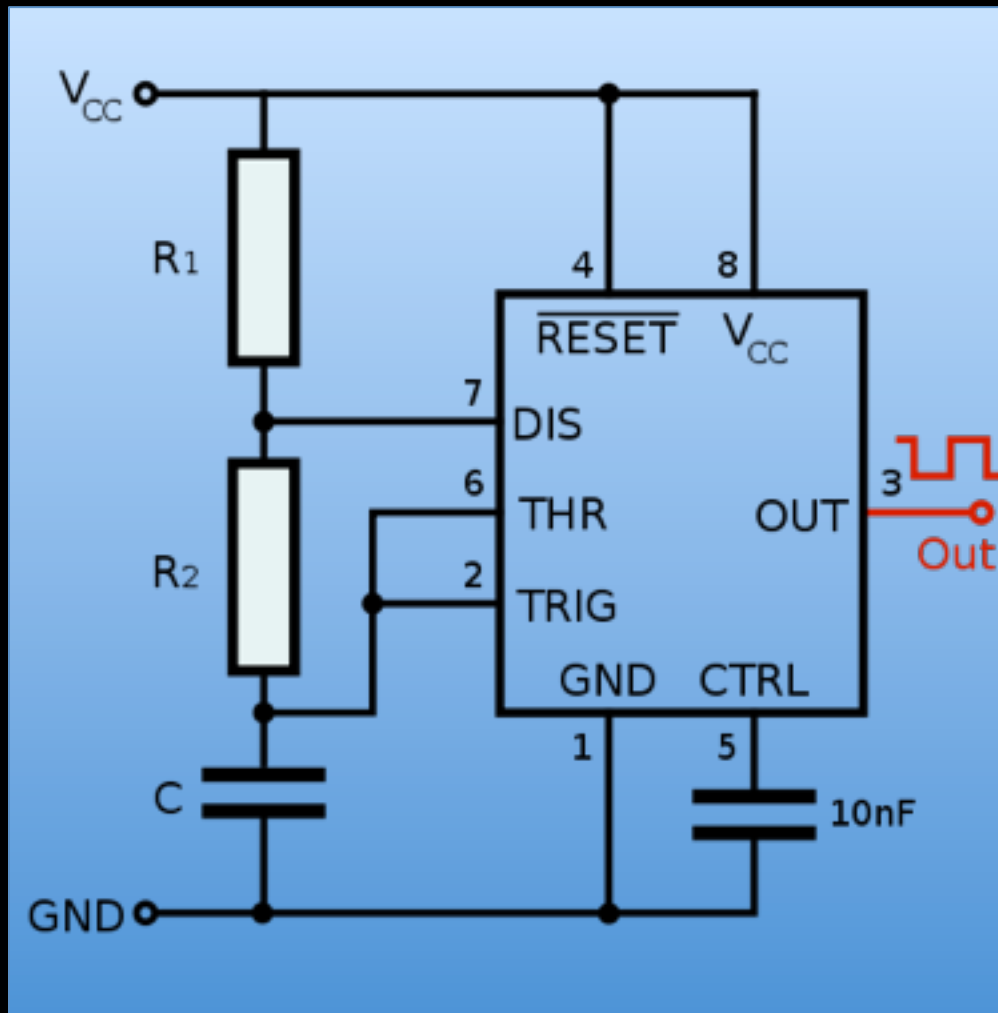
FLASH



EEPROM

# Digital ICs

## 555 Timer



$$f = \frac{1}{\ln(2) \cdot C \cdot (R_1 + 2R_2)}$$

# Digital ICs

## 555 Timer

### ASTABLE OPERATION

When the circuit is connected as shown in figure 4 (pin 2 and 6 connected) it triggers itself and free runs as a multivibrator. The external capacitor charges through  $R_A$  and  $R_B$  and discharges through  $R_B$  only. Thus the duty cycle may be precisely set by the ratio of these two resistors.

In the astable mode of operation, C charges and discharges between  $1/3 V_{CC}$  and  $2/3 V_{CC}$ . As in the triggered mode, the charge and discharge times and therefore frequency, are independent of the supply voltage.

Figure 5 shows actual waveforms generated in this

mode of operation.

The charge time (output HIGH) is given by :

$$t_1 = 0.693 (R_A + R_B) C$$

and the discharge time (output LOW) by :

$$t_2 = 0.693 (R_B) C$$

Thus the total period T is given by :

$$T = t_1 + t_2 = 0.693 (R_A + 2R_B) C$$

The frequency of oscillation is then :

$$f = \frac{1}{T} = \frac{1.44}{(R_A + 2R_B)C}$$

$$\text{The duty cycle is given by : } D = \frac{R_B}{R_A + 2R_B}$$

Figure 4

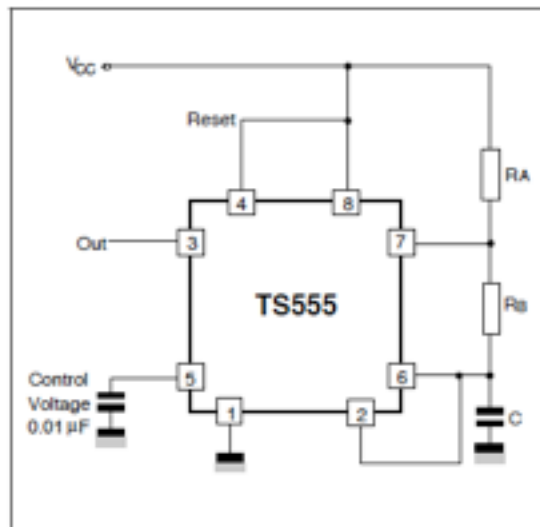
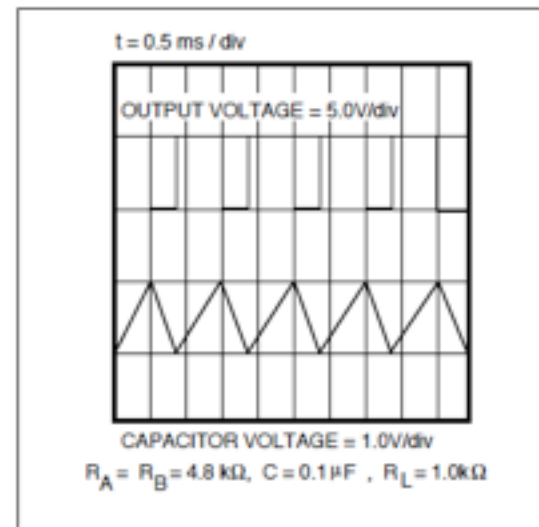


Figure 5





[Mini Flasher 2130 LED](#) 2130 Mini Flasher in Stock Now. Buy online and save today. [www.pelicanproducts.us](http://www.pelicanproducts.us)

[Analog IC Circuit Design](#) Learn Analog/RF, PLL, Mixed Signal. Convenient Silicon Valley Location! [www.ucsc.edu](http://www.ucsc.edu)

[ITT Tech - Official Site](#) Computer & Electronics Tech Degree 100+ Locations & Online Courses [www.itt-tech.edu](http://www.itt-tech.edu)



Ads by

## Learning

The 555 timer is a simple integrated circuit that can be used to make many different electronic circuits. With this information you will learn how the 555 works and will have the experience to build some of the circuits below.

- 1. An Overview
- 2. Pin Configuration
- 3. Inside The 555
- 4. Operating Modes
- 5. Using The Output
- 6. Calculator
- 7. Common Mistakes
- 8. 555 Datasheets

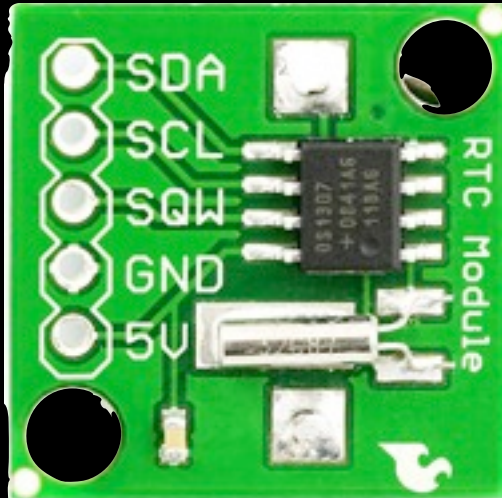
## Fun Circuits

The following are complete electronic circuits that you can build, they all utilize the 555 Timer circuit.

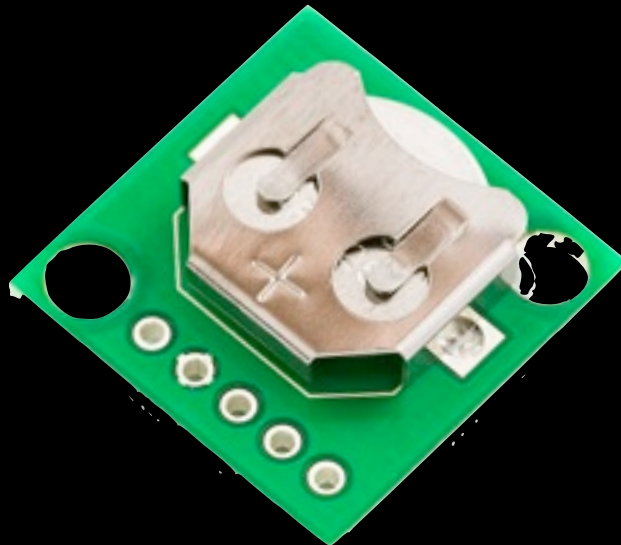
- 3x3x3 LED Cube
- 555 Amplifier
- Automatic Curtain Closer
- Bike Turning Signal
- Bi-Polar LED Driver
- Latch
- LED Dice
- LED Dimmer
- Light Detector
- Machine Gun
- Siren 100dB
- Stepper Motor Controller
- Stun Gun
- Ticking Bomb
- Tilt Switch

# Digital ICs

## Real-Time Clocks



- Seconds, minutes, hours, date, month, and year
- Battery lasts for 7-9 years
- Keeps track of leap years!



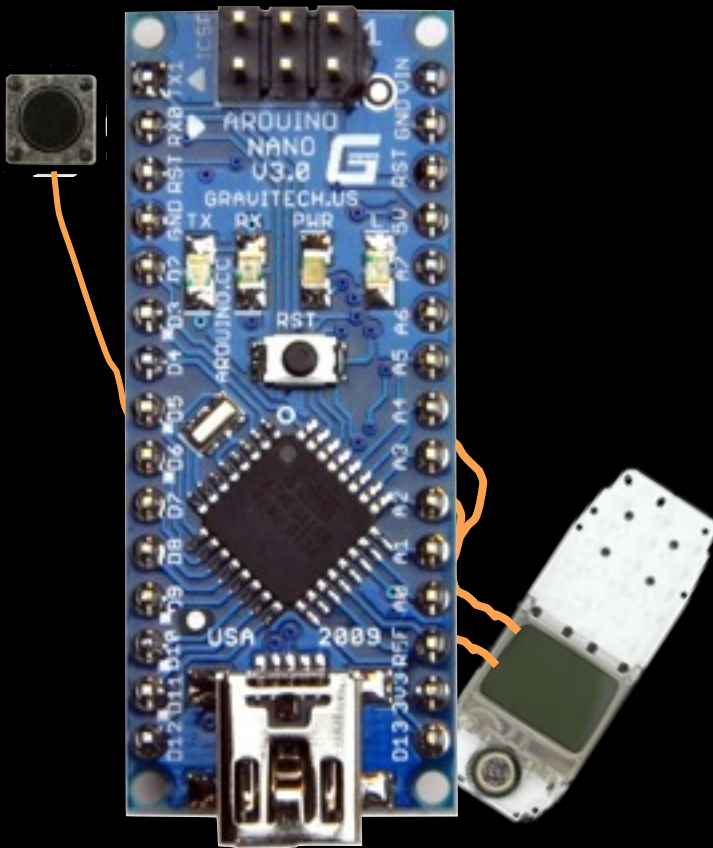
**DS1307**

<http://www.sparkfun.com/products/99>



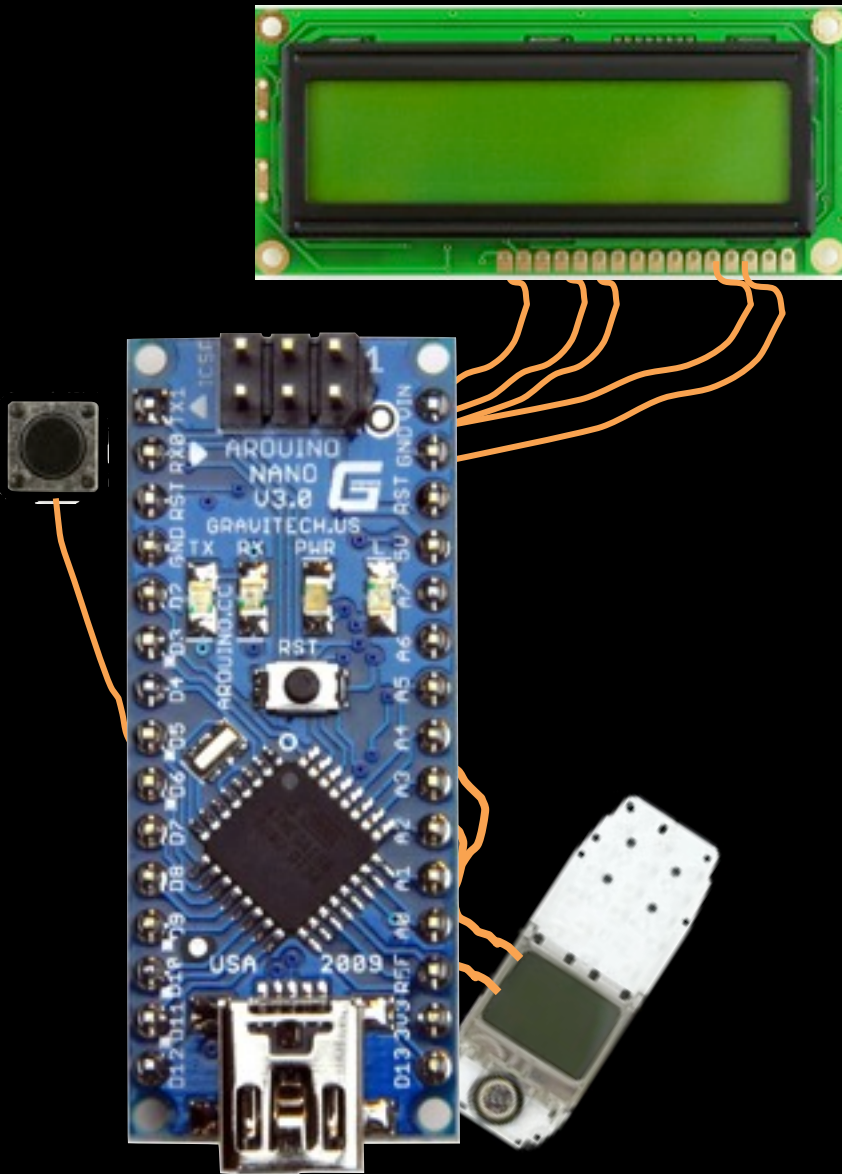
# Digital ICs

## Saving Pins



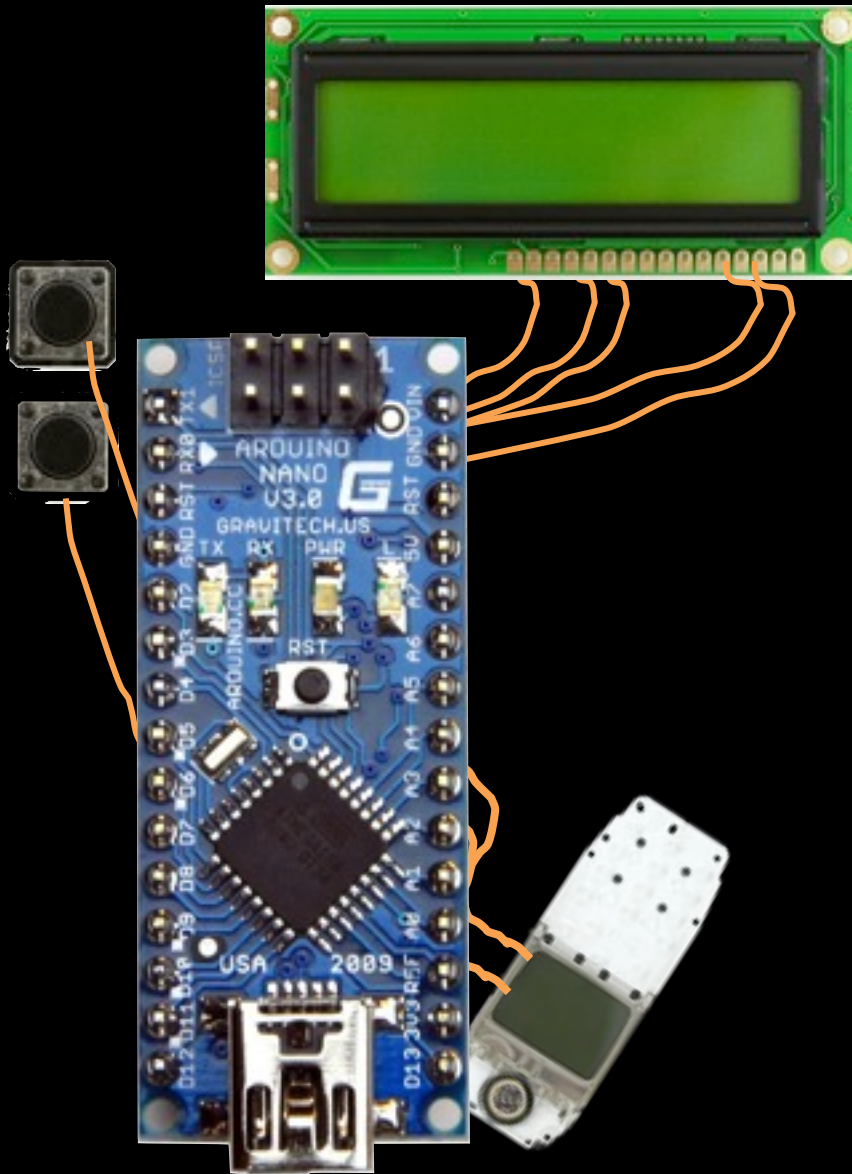
# Digital ICs

## Saving Pins



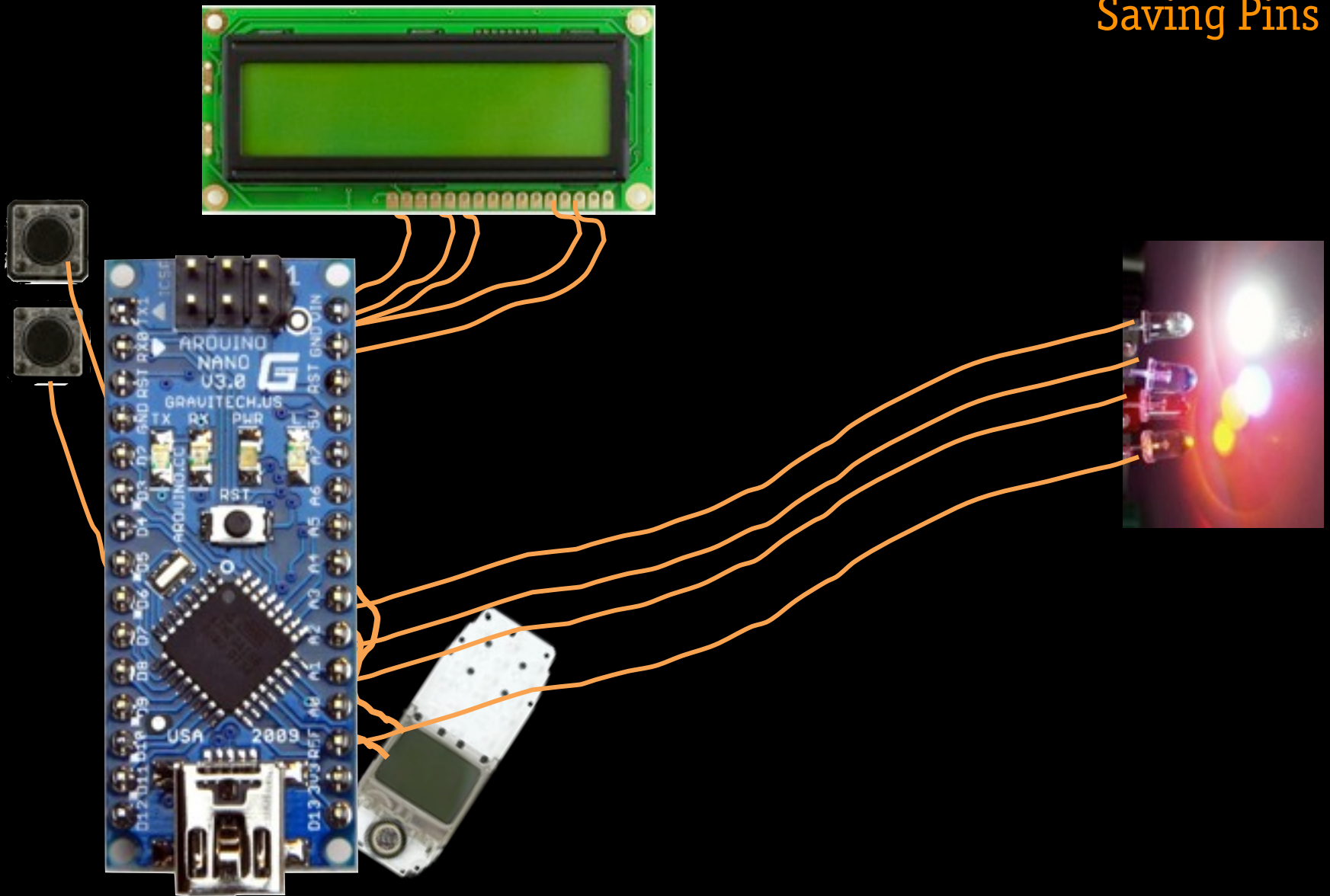
# Digital ICs

## Saving Pins



# Digital ICs

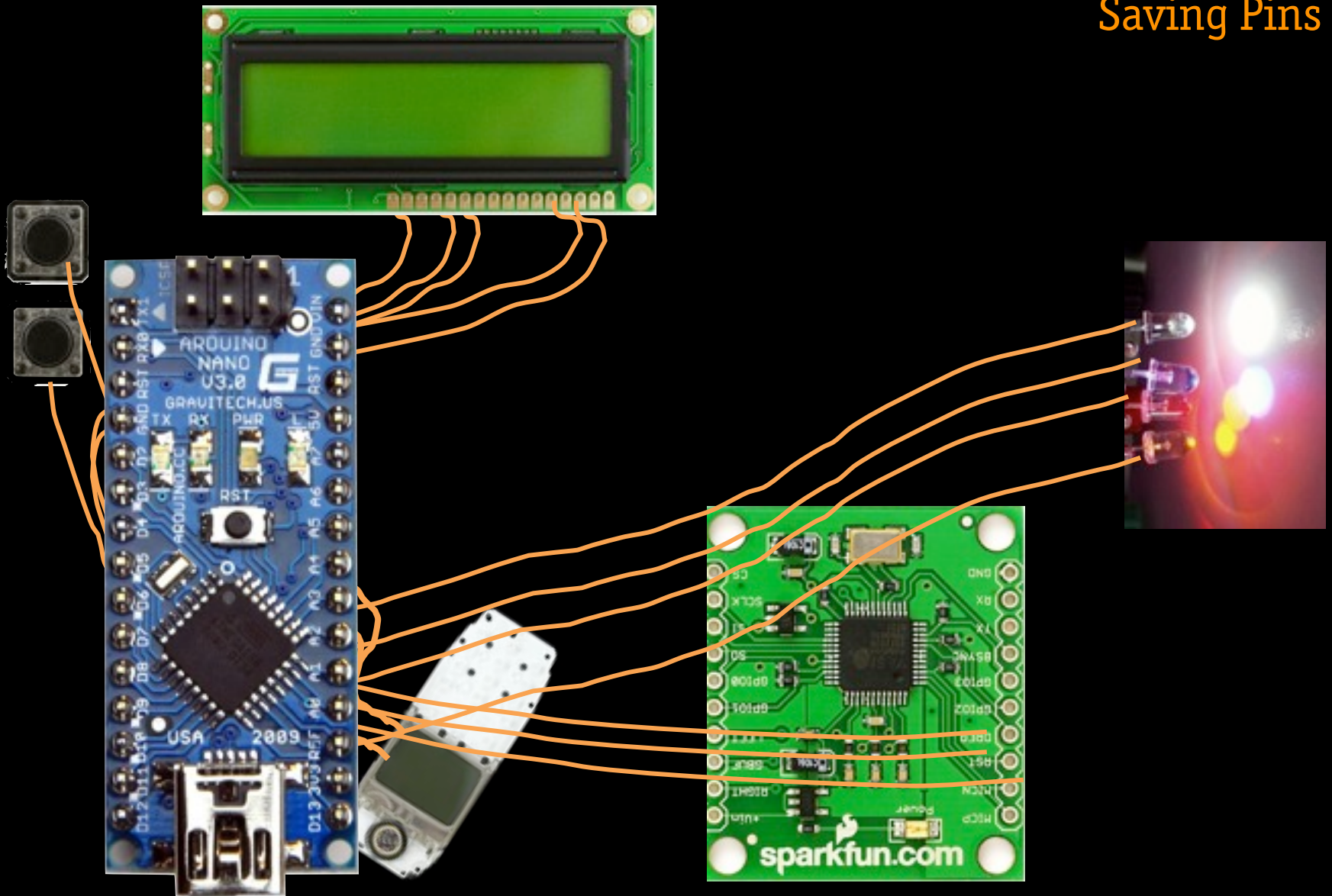
## Saving Pins





# Digital ICs

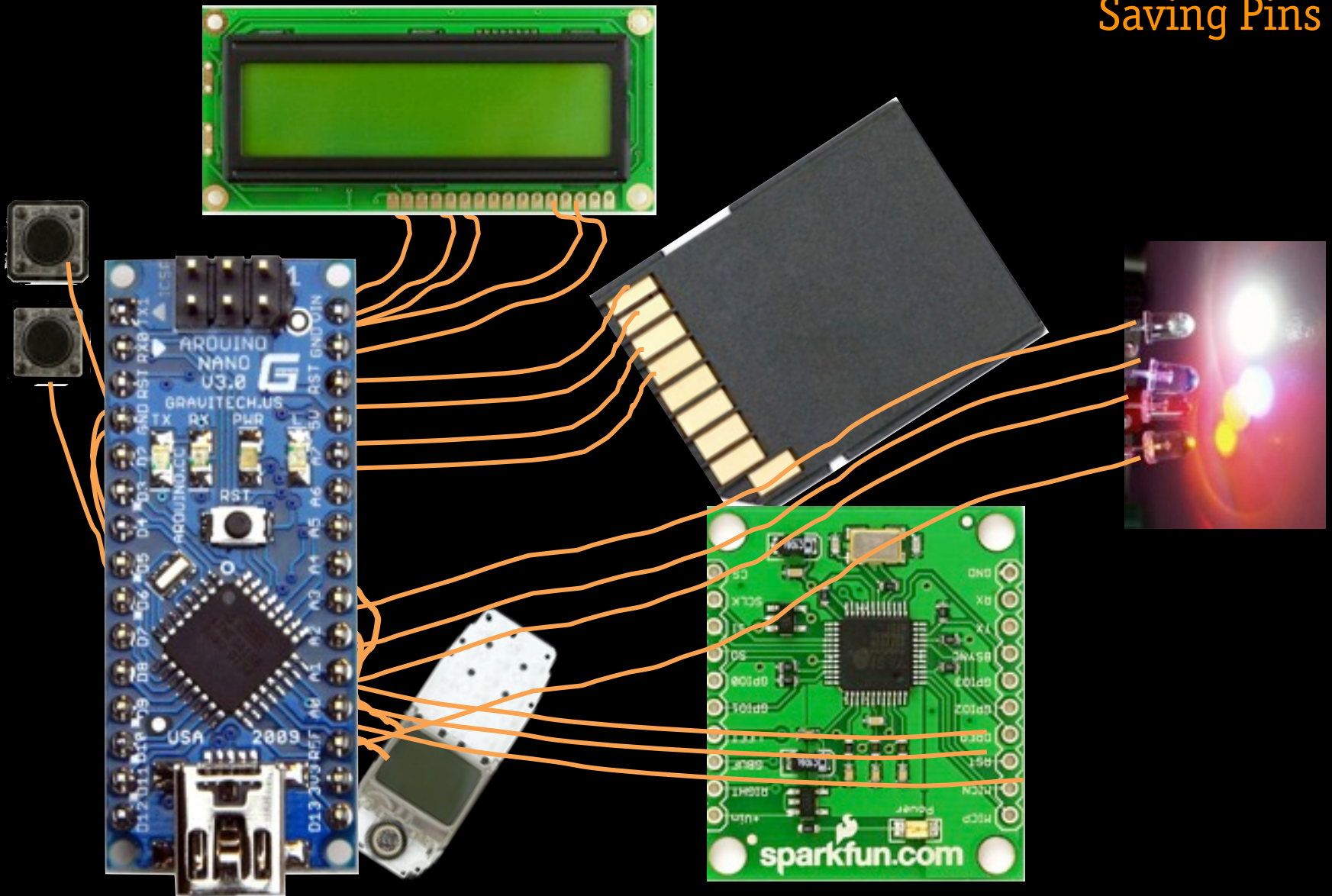
## Saving Pins





# Digital ICs

## Saving Pins



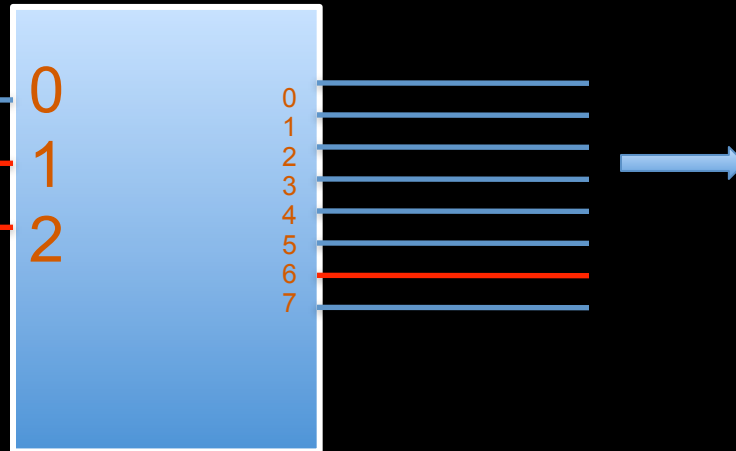
# Digital IC's

## Decoder

chip select lines

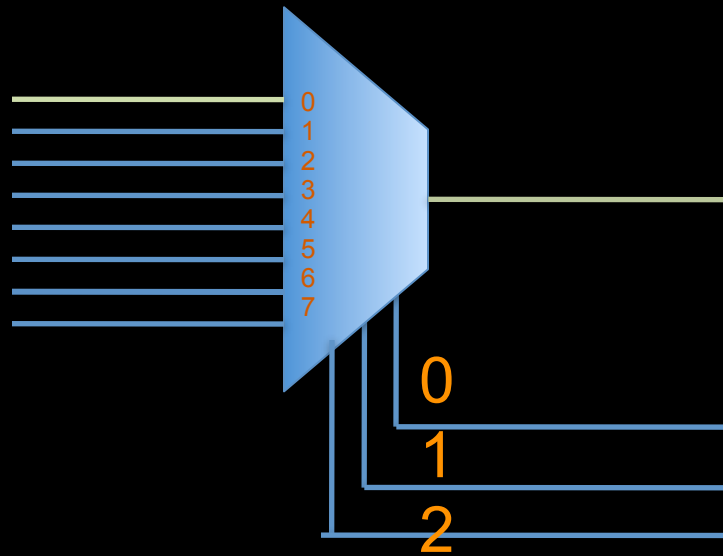


**74HC238**  
**3-8 Decoder**



# Digital IC's

## Multiplexer



CD74HC4067  
8-1 Mux  
(works for Analog too)

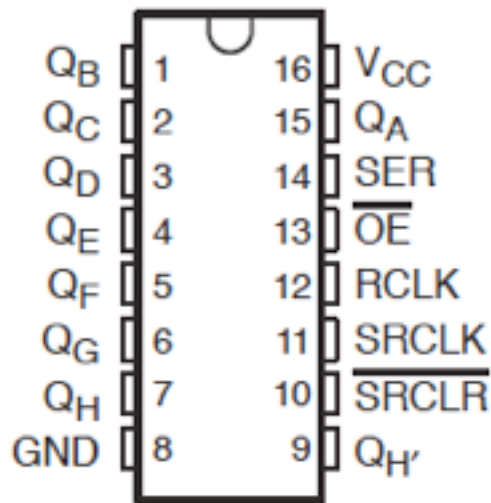


# Digital IC's

## Shift Register



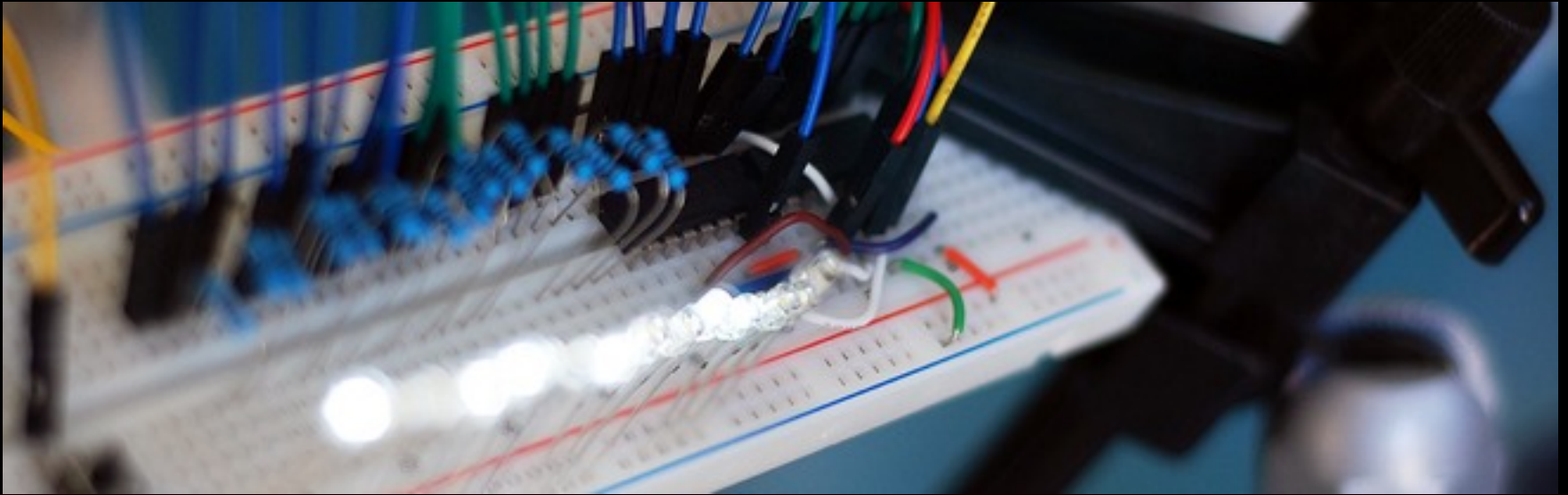
SN54HC595 . . . J OR W PACKAGE  
SN74HC595 . . . D, DB, DW, N, OR NS PACKAGE  
(TOP VIEW)



- Data is read in on the serial (SER) input line
- Data is shifted once with each clock cycle (SCLK).
- The “register clock” (RCLK) acts as a clutch, and holds current values when set LOW.

# Digital IC's

## Shift Register



<http://bildr.org/2011/02/74hc595/>

Say you want to control a huge number of LEDs with your Arduino

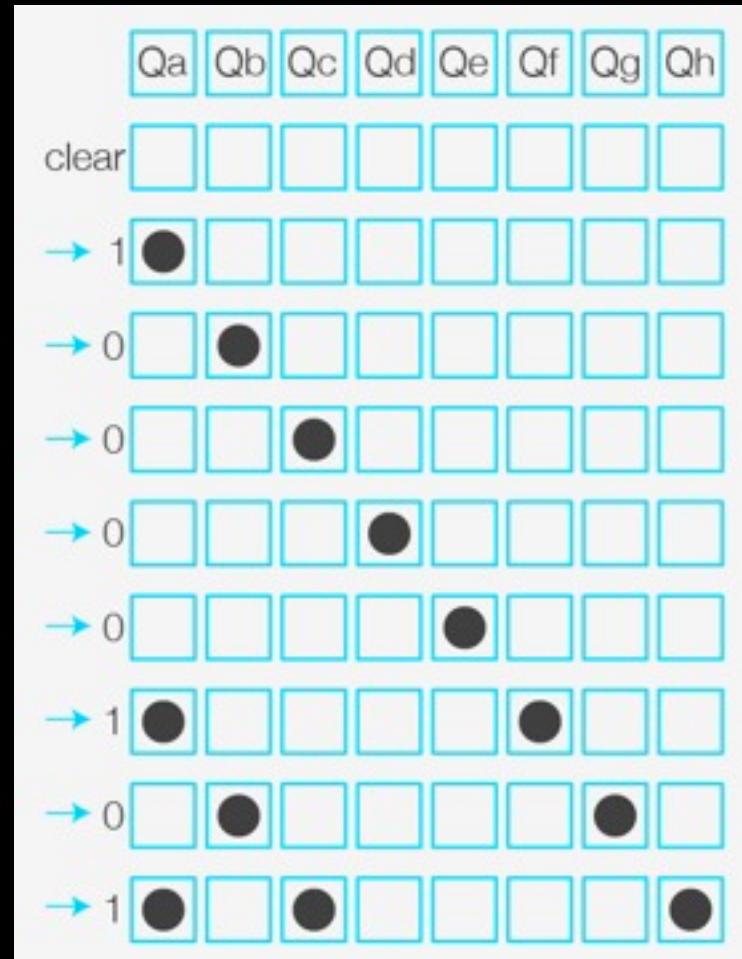


# Digital IC's

## Shift Register

To turn on the 1st, 3rd, and 8th LED:

- Pull SRCLR low to clear the register
- Pull RCLK low to clutch the output
- Pulse SER 1, 0, 0, 0, 0, 1, 0, 1 with each CLK pulse
- Pull RCLK high to turn on the LEDs



# Integrated Circuits...

So many more!

- Specialized Processors (like your Mp3 Decoder)
- Digital Signal Processors



- 7-Segment Display Driver (MC14489B)

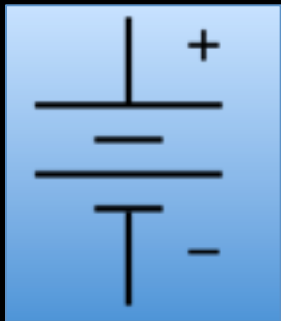


- Check out SparkFun's [General IC's](#) page for more!

# Choosing ICs

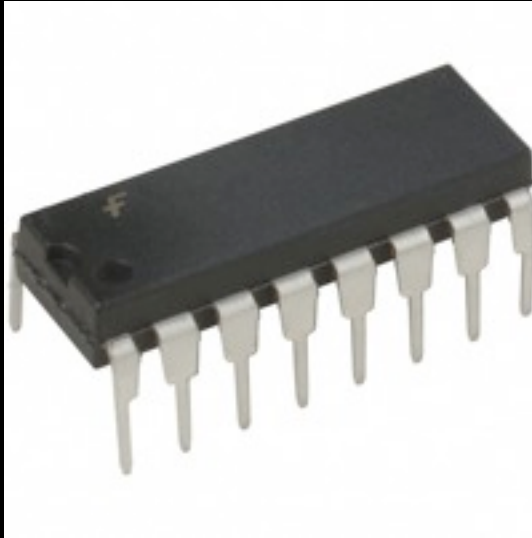
## Voltage & Current

Choosing parts with the same operating voltage will save major headaches and components!

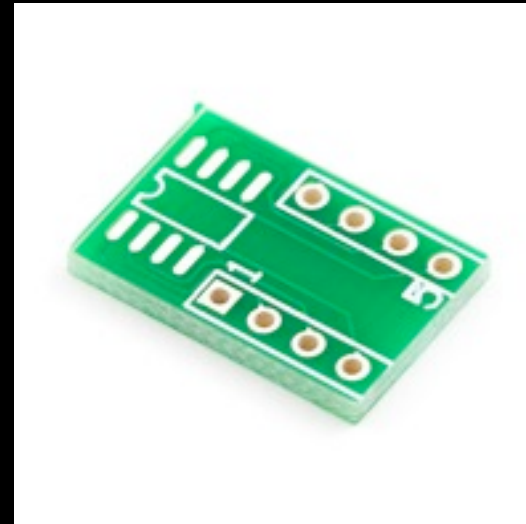


Keep your power source in mind..  
USB, batteries, and voltage regulators can only provide so much current.

# Choosing ICs Packaging



16-DIP



8-SOIC to DIP Adapter

# Choosing ICs

## Communication Interface

### Atmega168 Pin Mapping

Arduino function

reset

SPI?

VCC

GND

I2C?

digital pin 7

UART?

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT24/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

Arduino function

analog input 5

analog input 4

analog input 3

analog input 2

analog input 1

analog input 0

GND

analog reference

VCC

digital pin 13

digital pin 12

digital pin 11 (PWM)

digital pin 10 (PWM)

digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MISO, MOSI, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.



# Choosing ICs

## Community Support

Address <http://www.arduino.cc/playground/Main/InterfacingWithHardware>

## Arduino: Playground

*The playground is a publicly-editable wiki about Arduino.*

[view](#) [history](#) [edit](#) [print](#) [login](#) [register](#)

### :: Interfacing with Hardware ::

These topics cover the hardware and software setup required to connect an Arduino device with a variety of electronic parts, chips and devices. A related topic not covered under this section is the shield, boards that plug directly into an Arduino's pin layout. Information on the creation and use of specific shields belongs in that section. Information on shields in general and their creation belongs here.

---

## Navigation

### Output

- [Visual](#)
  - [LED Lights and Displays](#)
  - [LCDs](#)
  - [Video](#)
- [Audio](#)
  - [µC Hobby Arduino Sound Tutorial](#)
  - [Realtime audio processing](#)
  - [Tone Generation Libraries](#)
  - [Synthesizers and sound generation](#)
  - [MIDI](#)

[Manuals and Curriculum](#)

[Board Setup and Configuration](#)

[Development Tools](#)

[Interfacing With Hardware](#)

- [Output](#)
- [Input](#)
- [Storage](#)
- [Communication](#)

# Lab 6 Preview:

Using the MP3 Decoder Chip

Debouncing

Interrupt Handlers

# Lab 6 Preview

## Interrupts



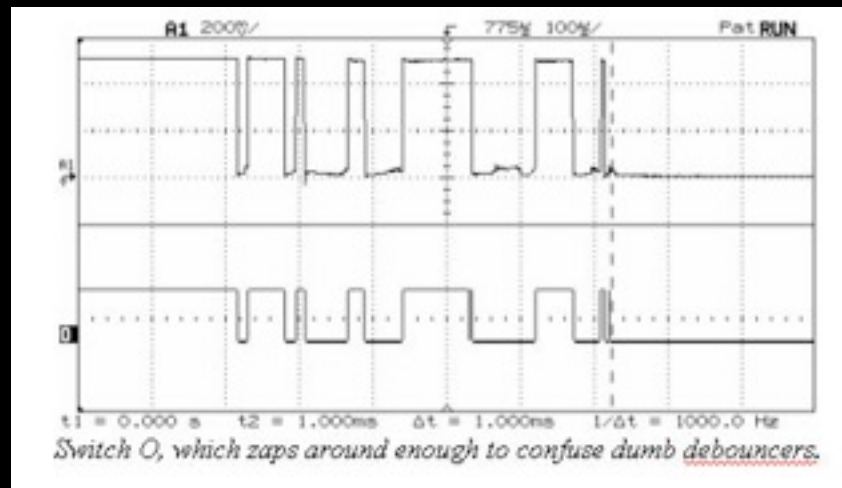
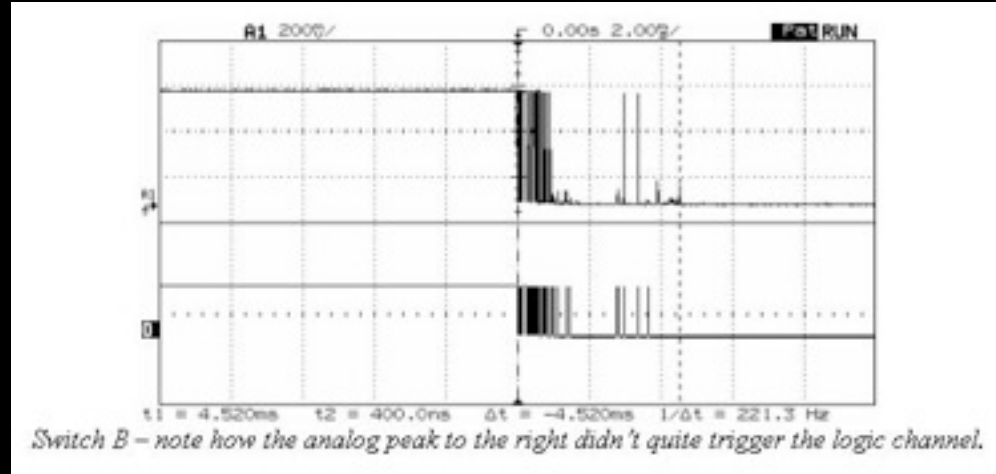
Write a special function  
("Interrupt Service Routine") that  
gets called exactly when the  
button is pressed.

# Lab 6 Preview

## Debouncing



What happens when you press the button?



©The Ganssle Group