

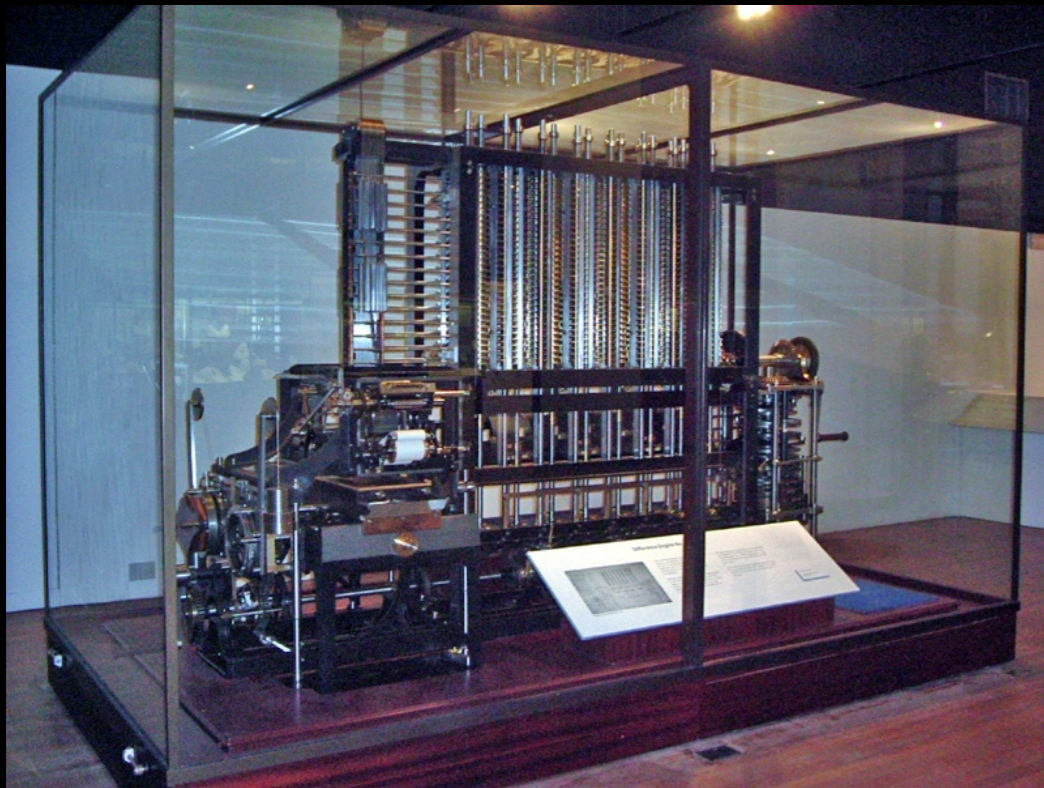
# Interrupts & Using ICs

Press Play: Interactive Device Design | Aug 08, 2012

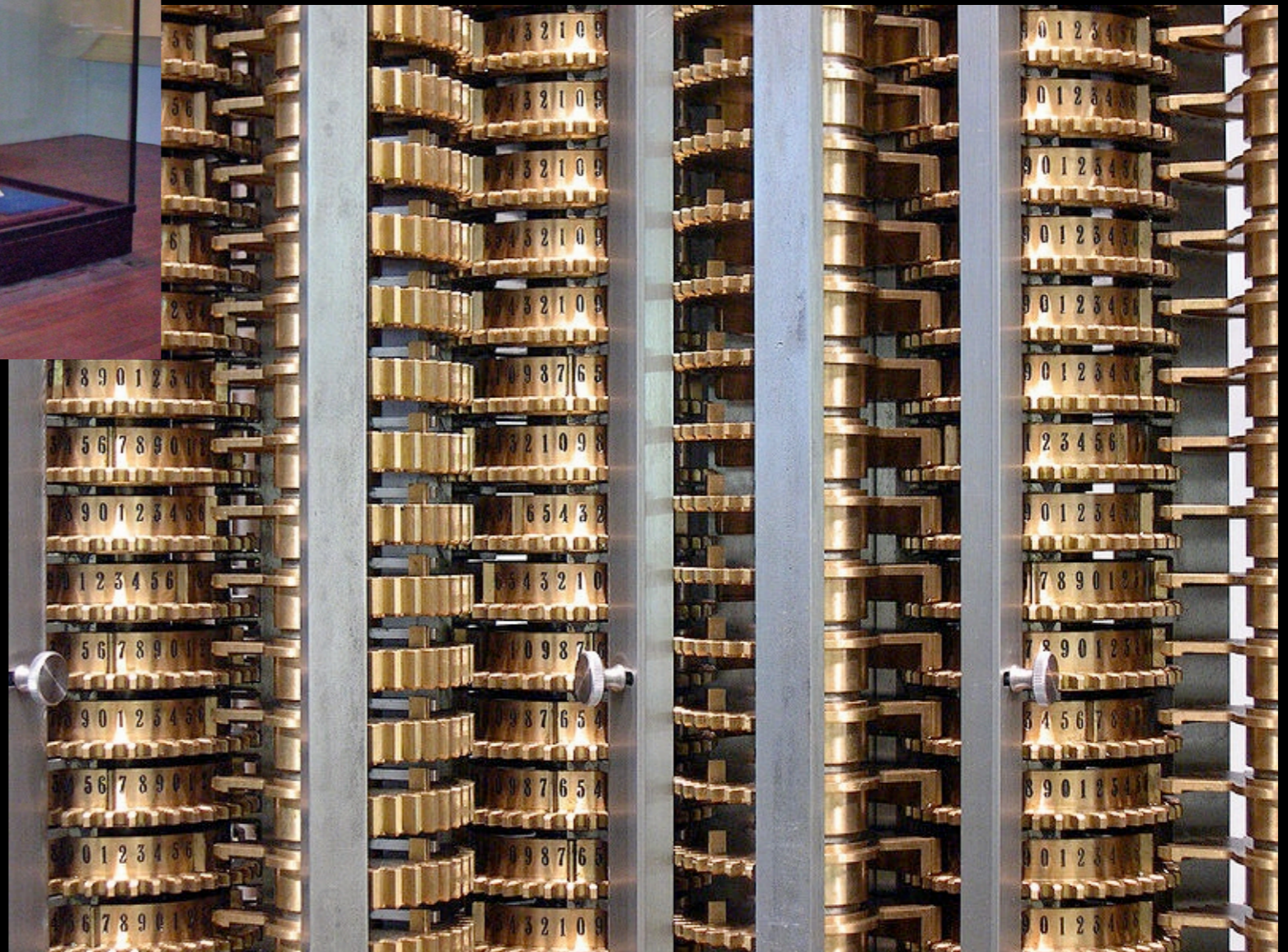
# Before Integrated Circuits

## Mechanical Computing

1849



*Wikimedia, by Joe D in January 2005.*

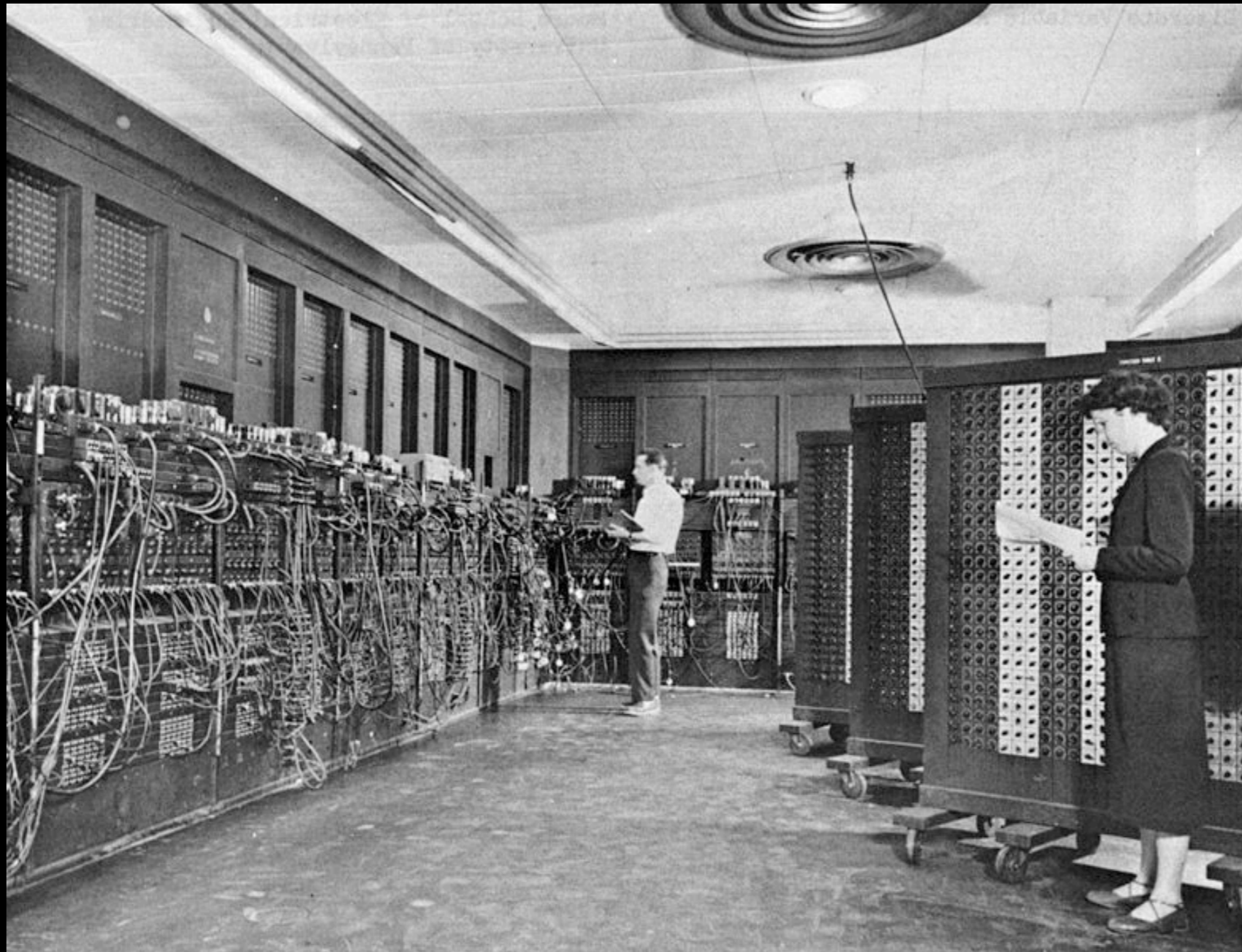


*Wikimedia, Carsten Ullrich, 2005*

# Before Integrated Circuits

## Vacuum Tubes

1945

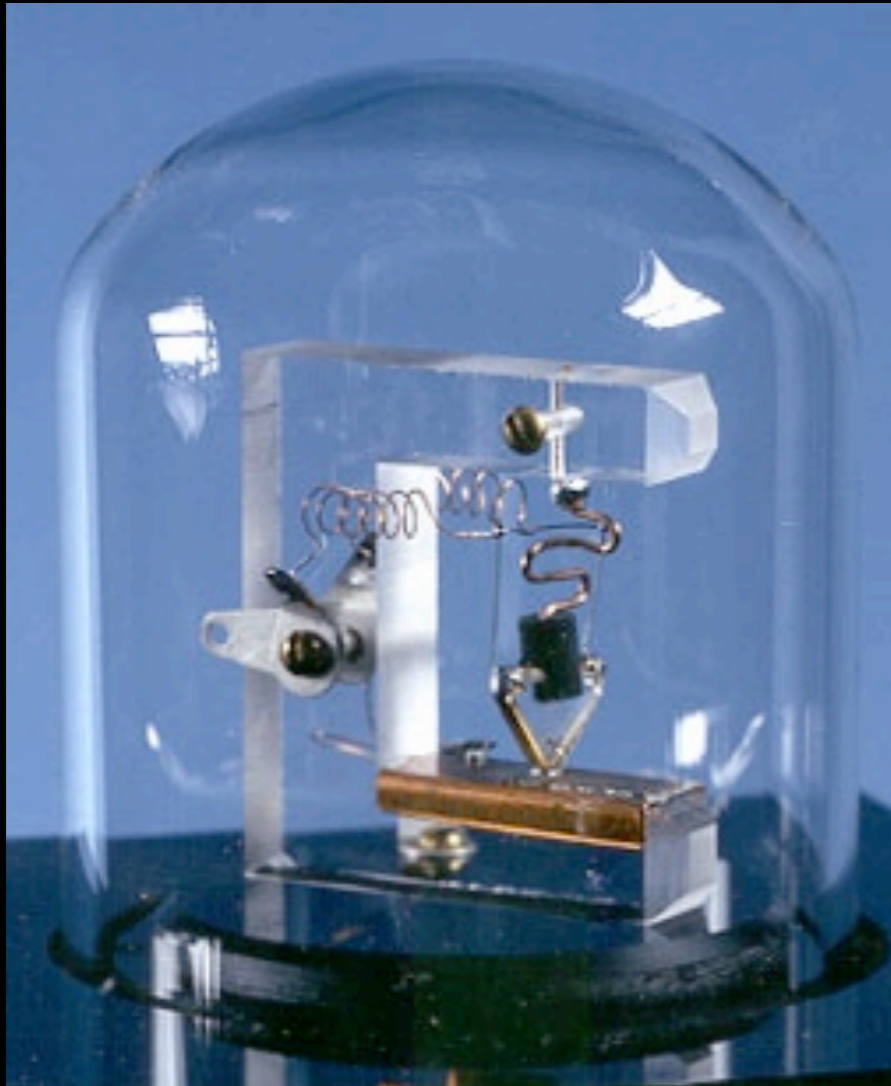


*Wikimedia, US Army Photo*

# Before Integrated Circuits

## Transistors

1955

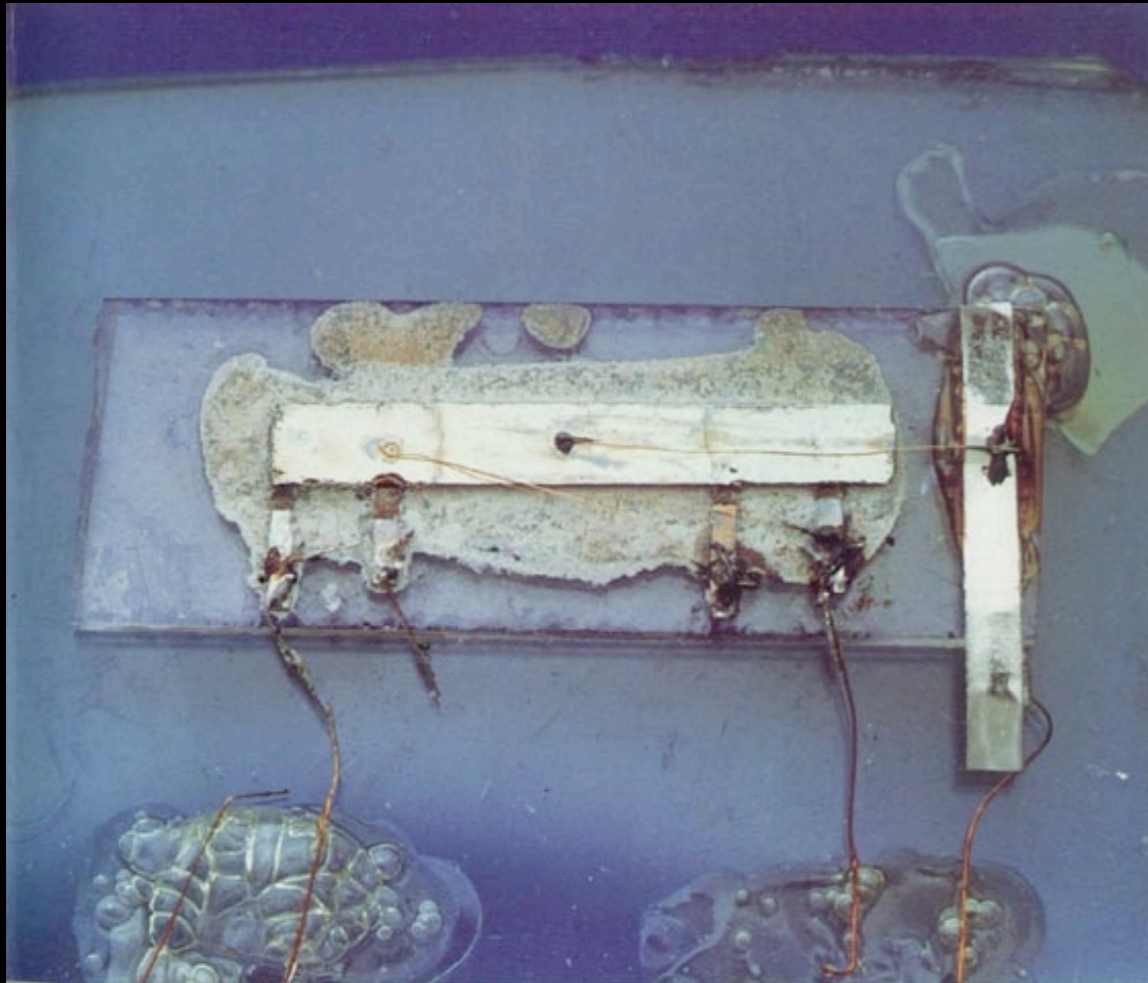


1947



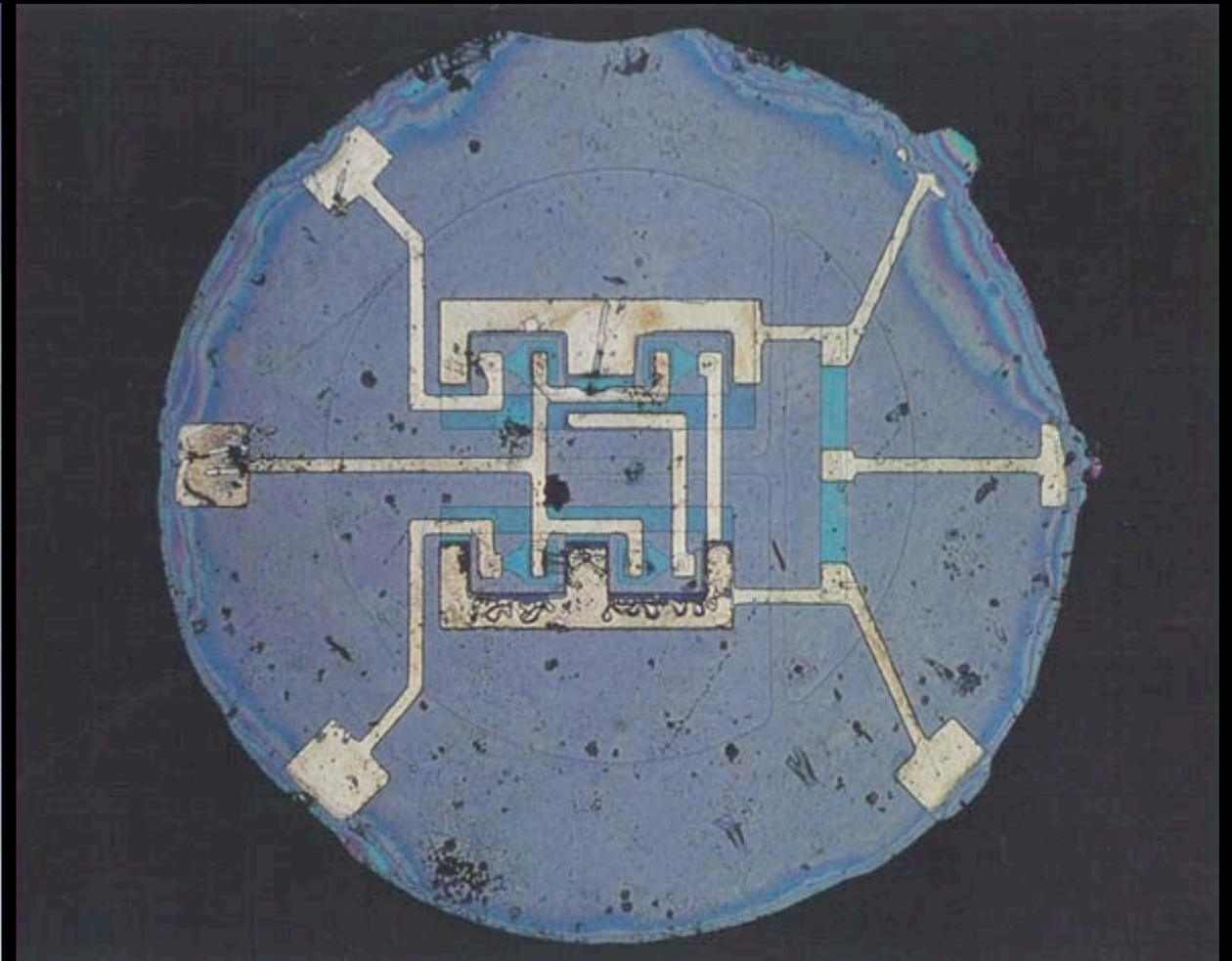
# Integrated Circuits

## Multiple Components on a Single Substrate



STATE OF THE ART ©Copyright Stan Augarten

1958

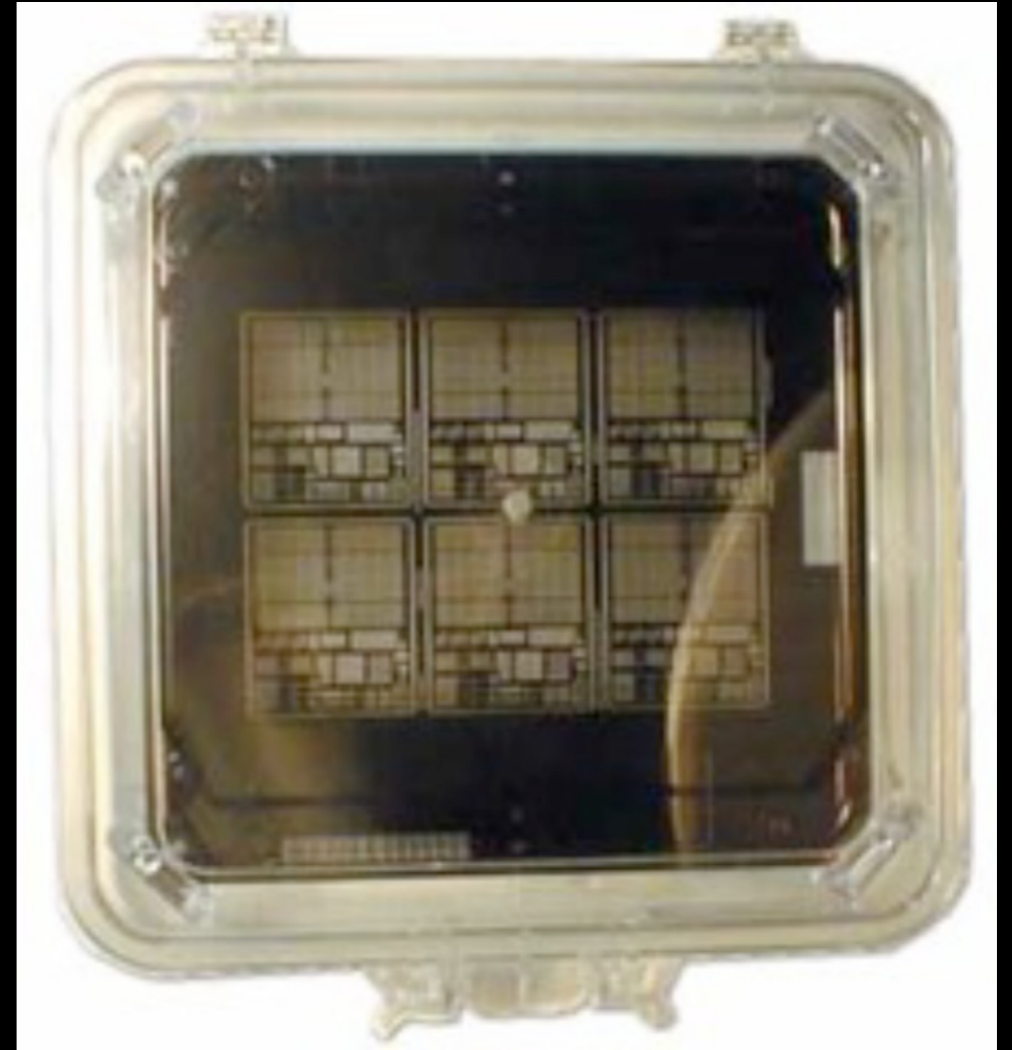


STATE OF THE ART ©Copyright Stan Augarten

1961

# Integrated Circuits Manufacturing

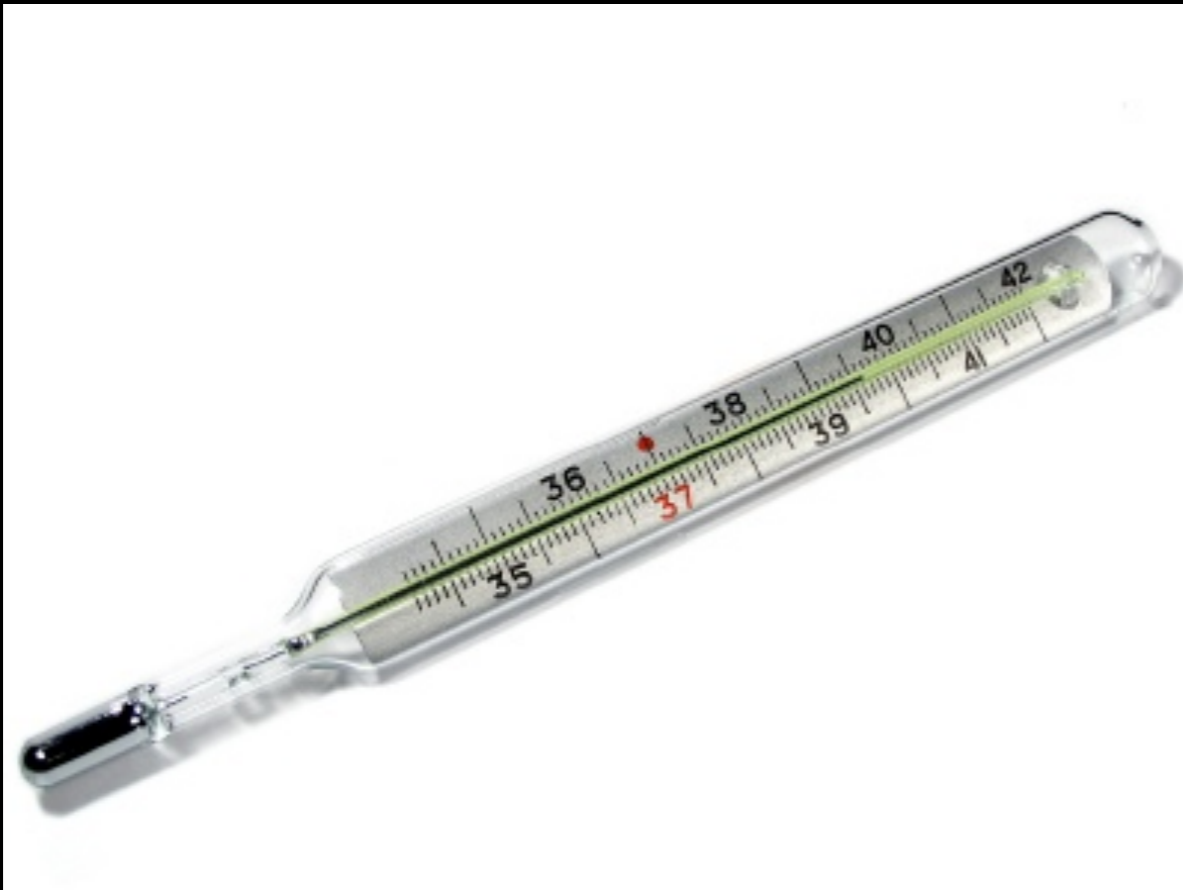
Similar to Silkscreening



# Analog vs Digital

'Real World' vs 'Computer World'

Analog

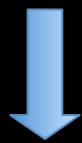
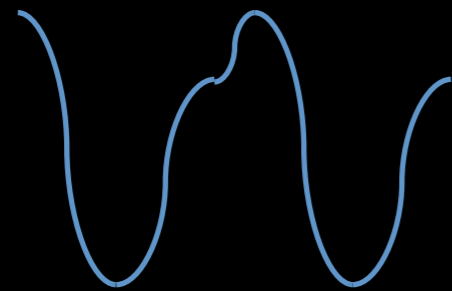


Digital

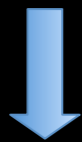


# ADCs and DACs

Convert from Analog to Digital and Back Again

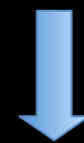


Microchip  
MCP 3208

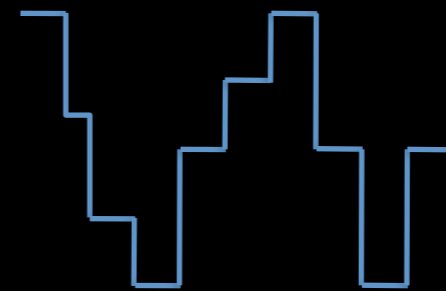


0xFF, 0x12, 0x0F, ...

0xFF, 0x12, 0x0F, ...



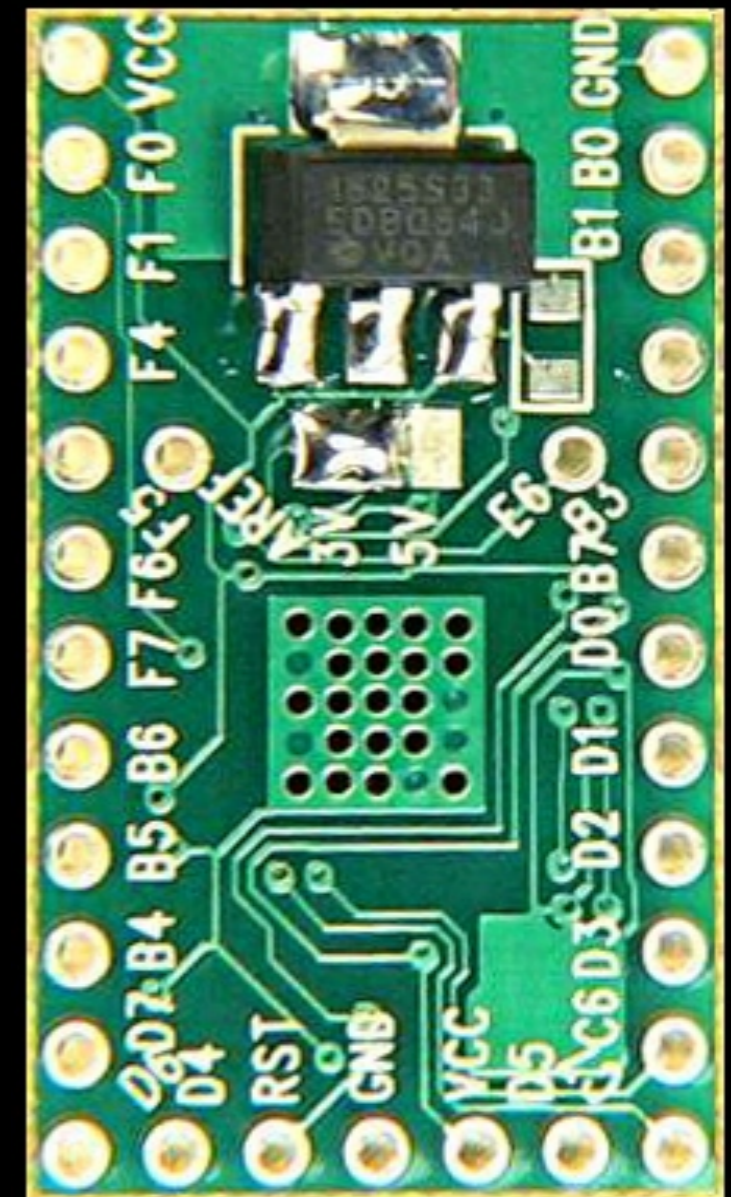
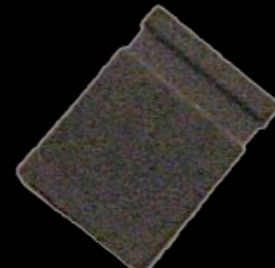
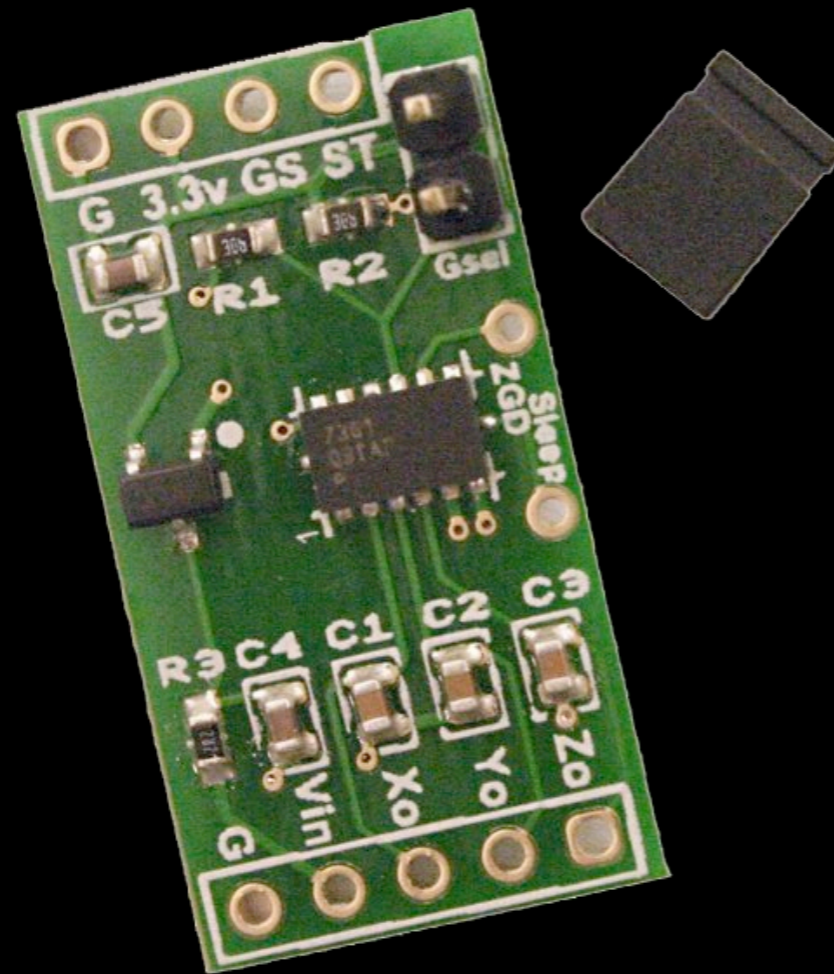
Analog Devices  
AD 420





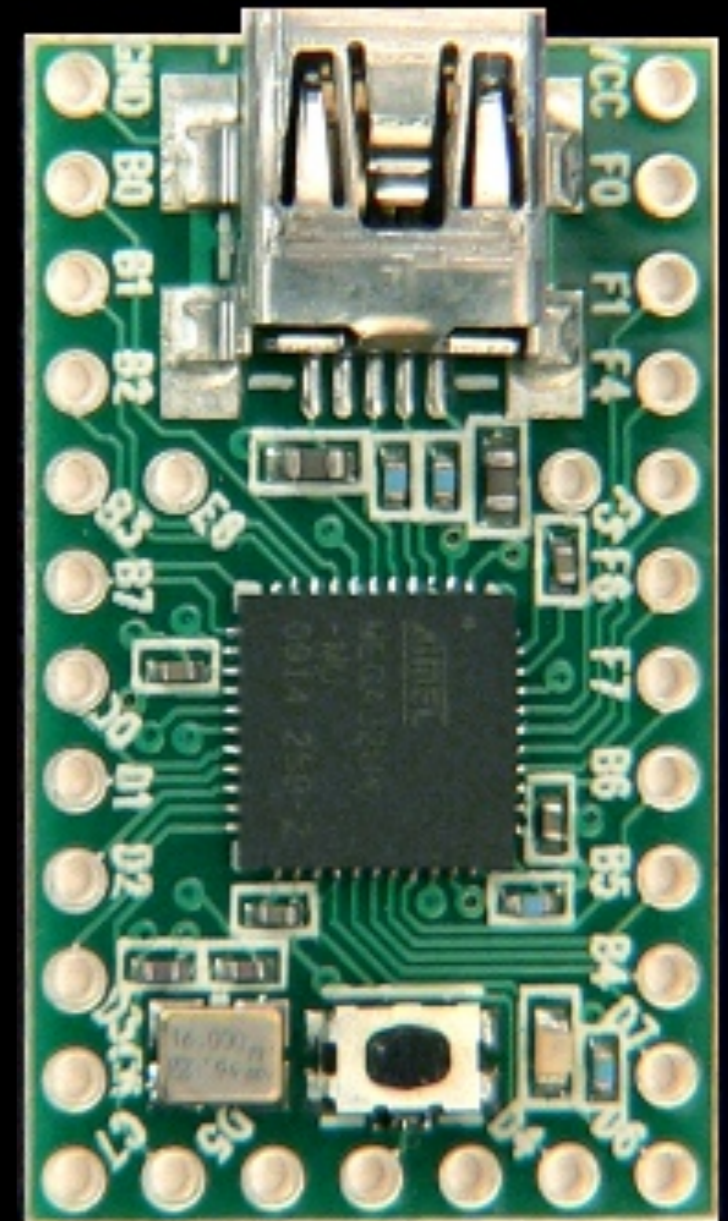
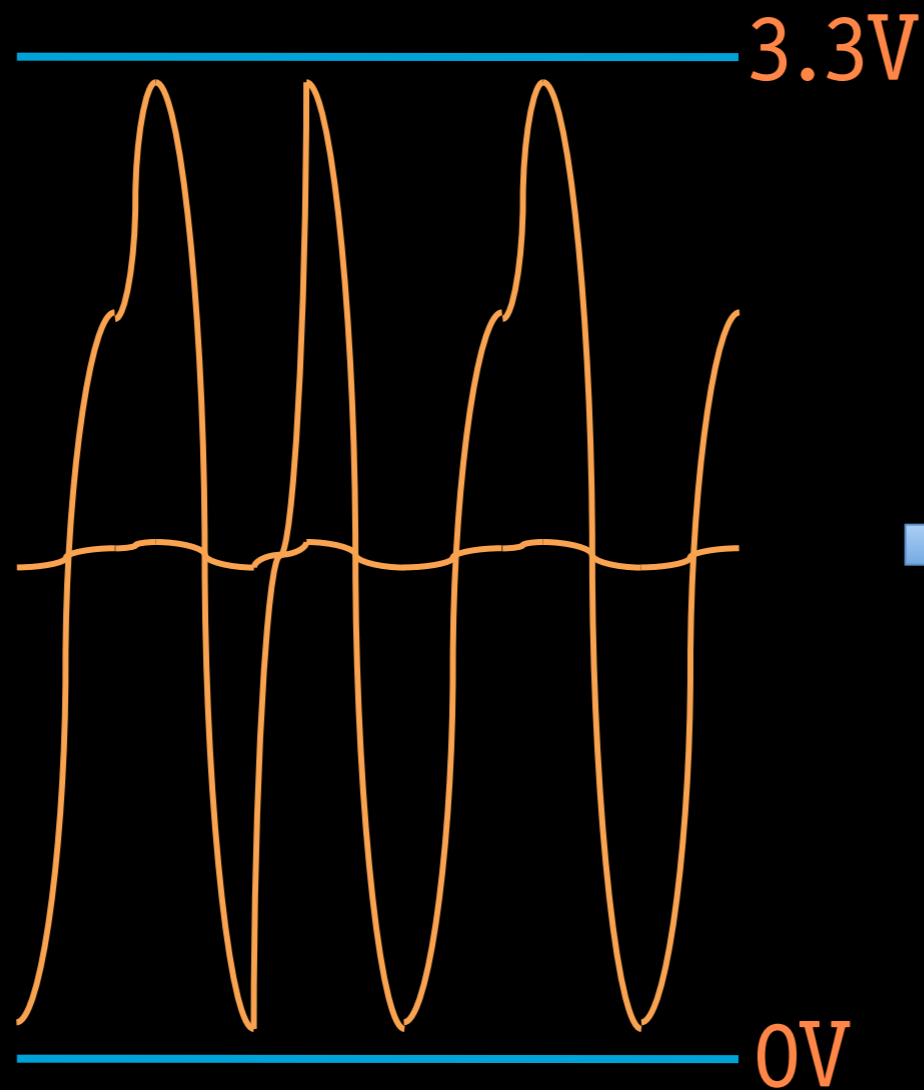
# Analog ICs

You Have Already Used A Few!



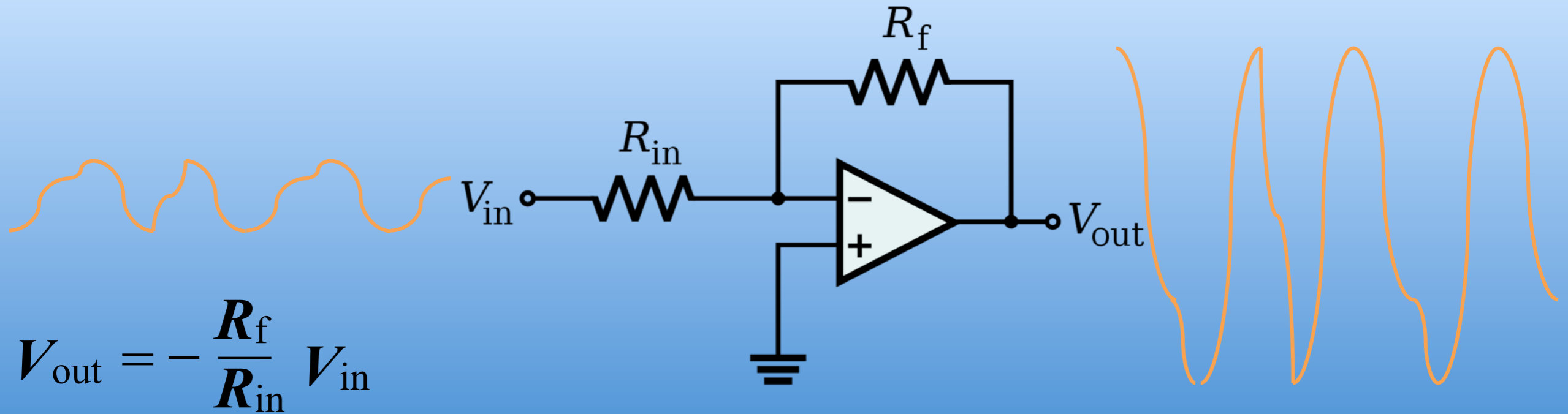
# Analog ICs

## Operational Amplifier

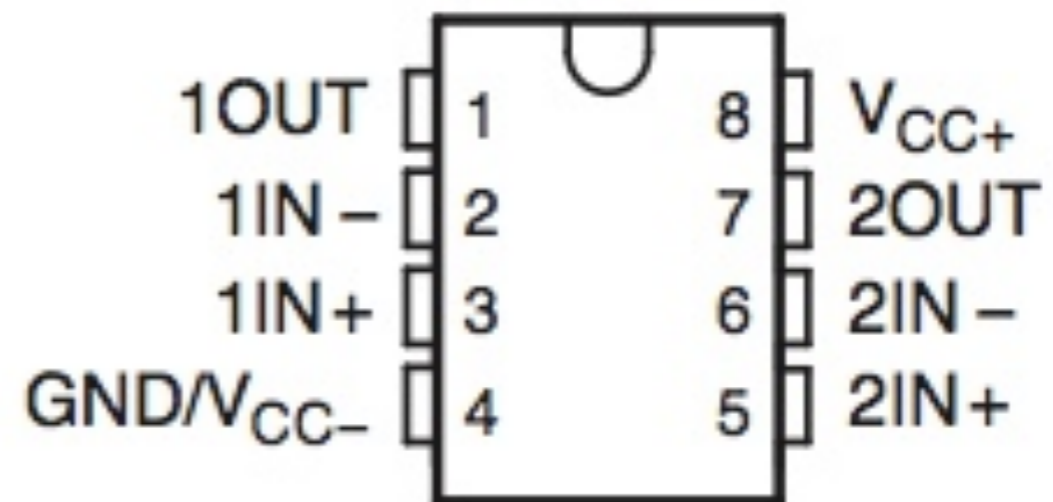


# Analog ICs

## Operational Amplifier (Inverting Circuit)

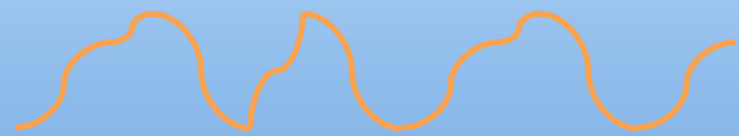


LMV822... D OR DGK PACKAGE  
(TOP VIEW)

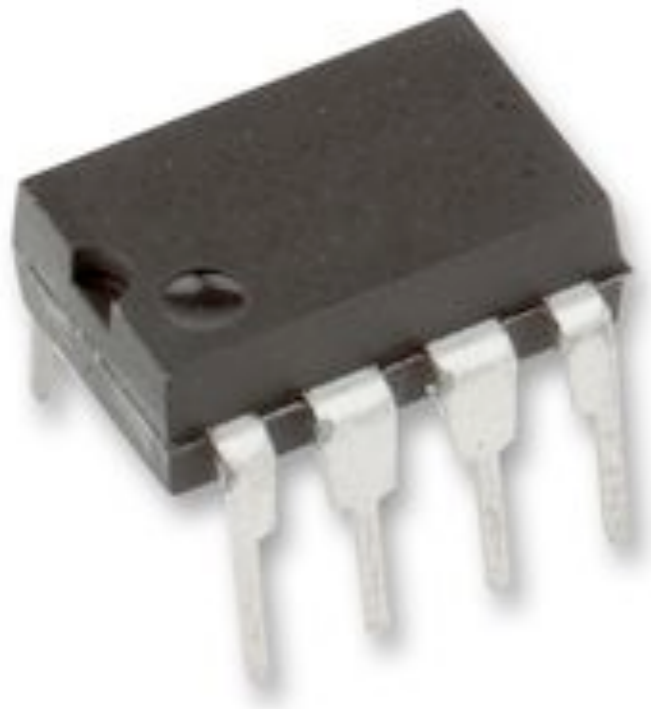
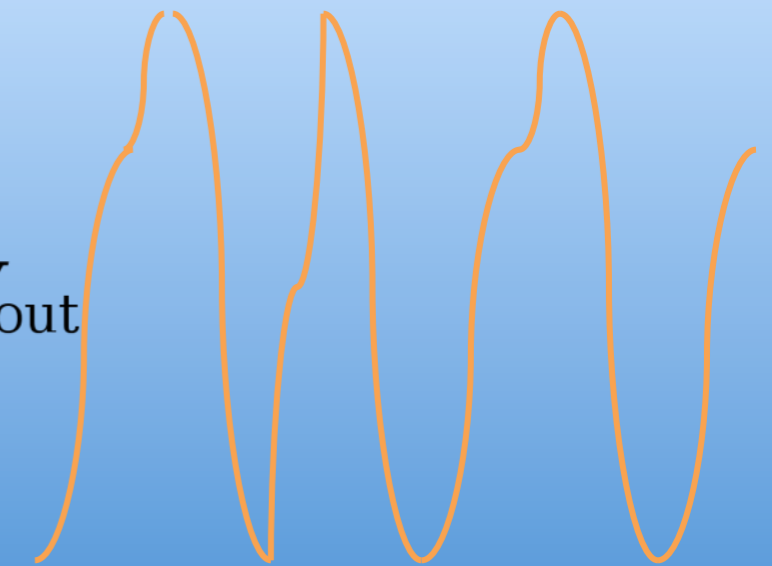
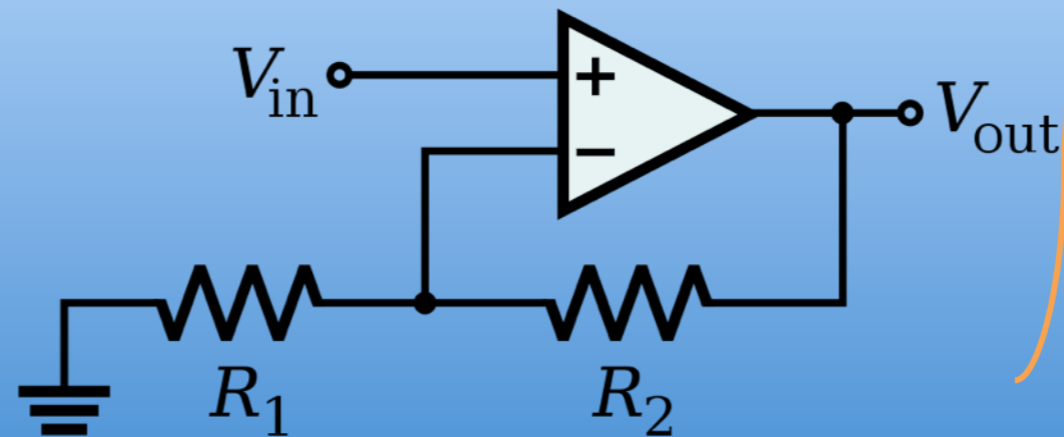


# Analog ICs

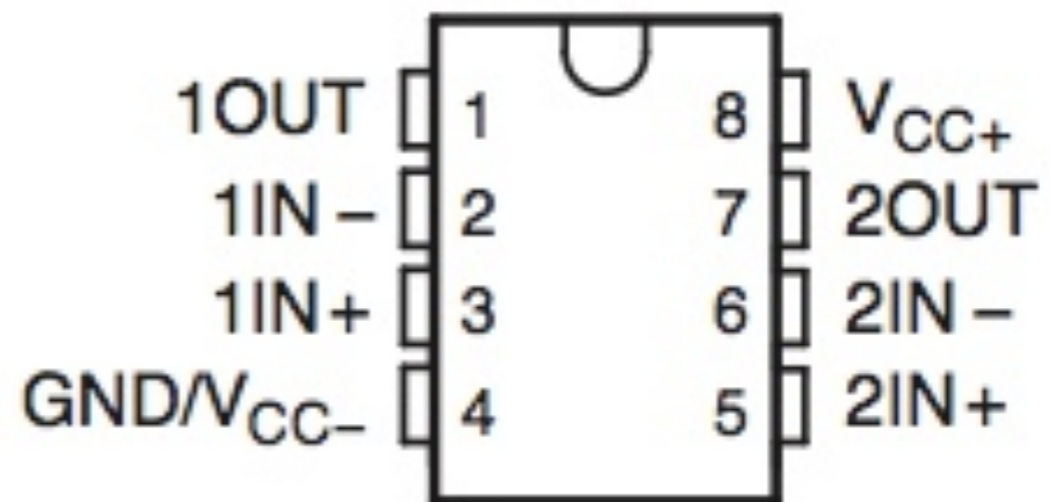
## Operational Amplifier (Non-Inverting Circuit)



$$V_{\text{out}} = \left(1 + \frac{R_2}{R_1}\right) V_{\text{in}}$$



LMV822... D OR DGK PACKAGE  
(TOP VIEW)



# Analog ICs

## Filter Chips



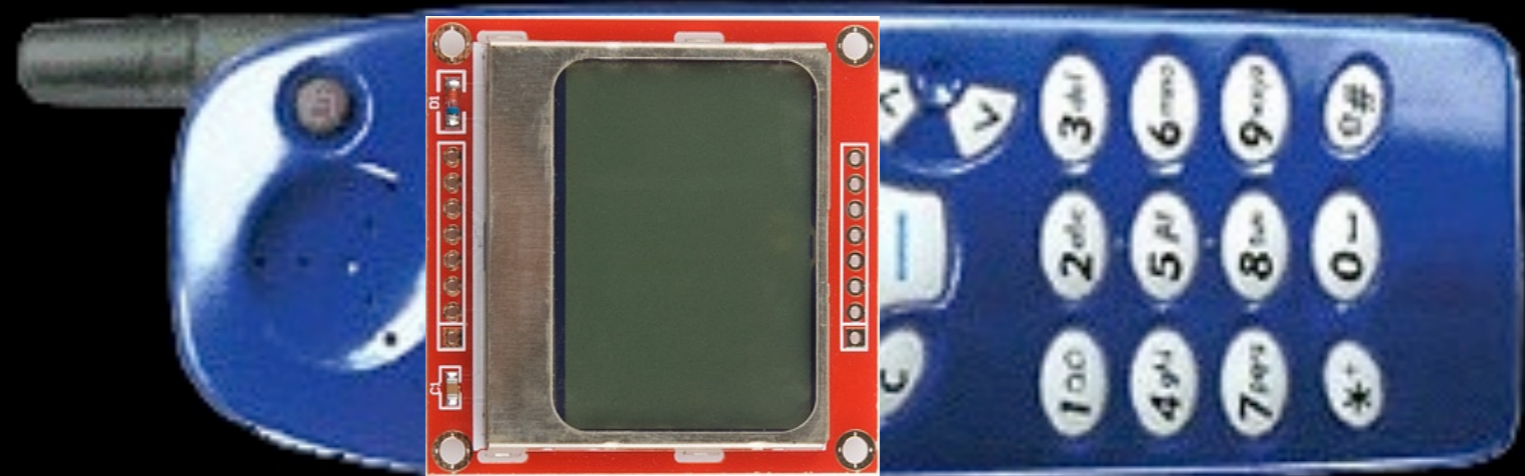
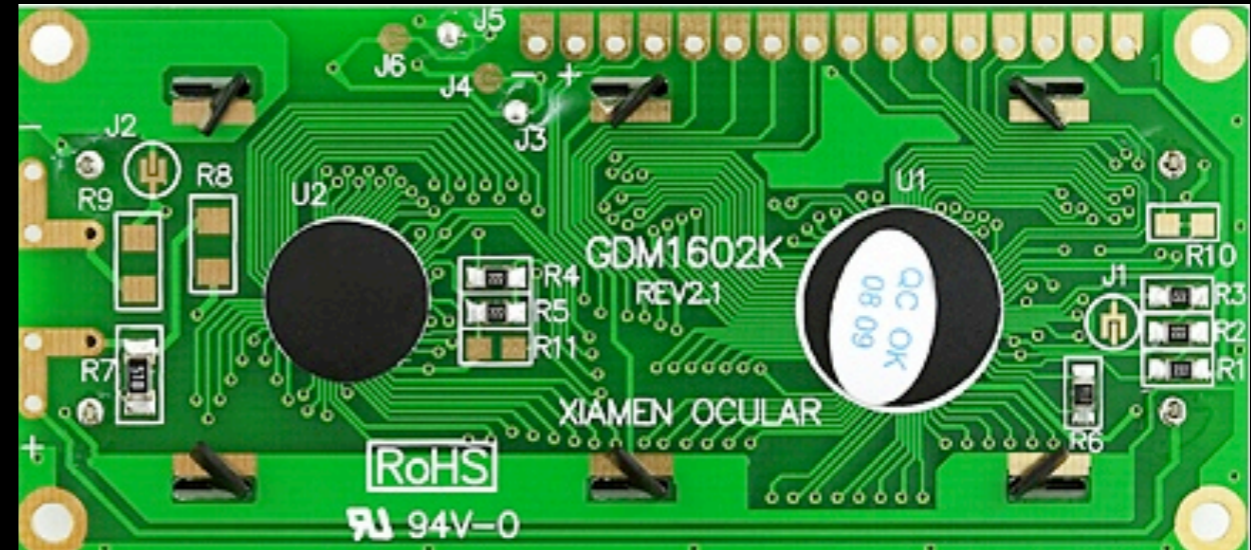
or



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

# Digital ICs

You Have Already Used a Ton!

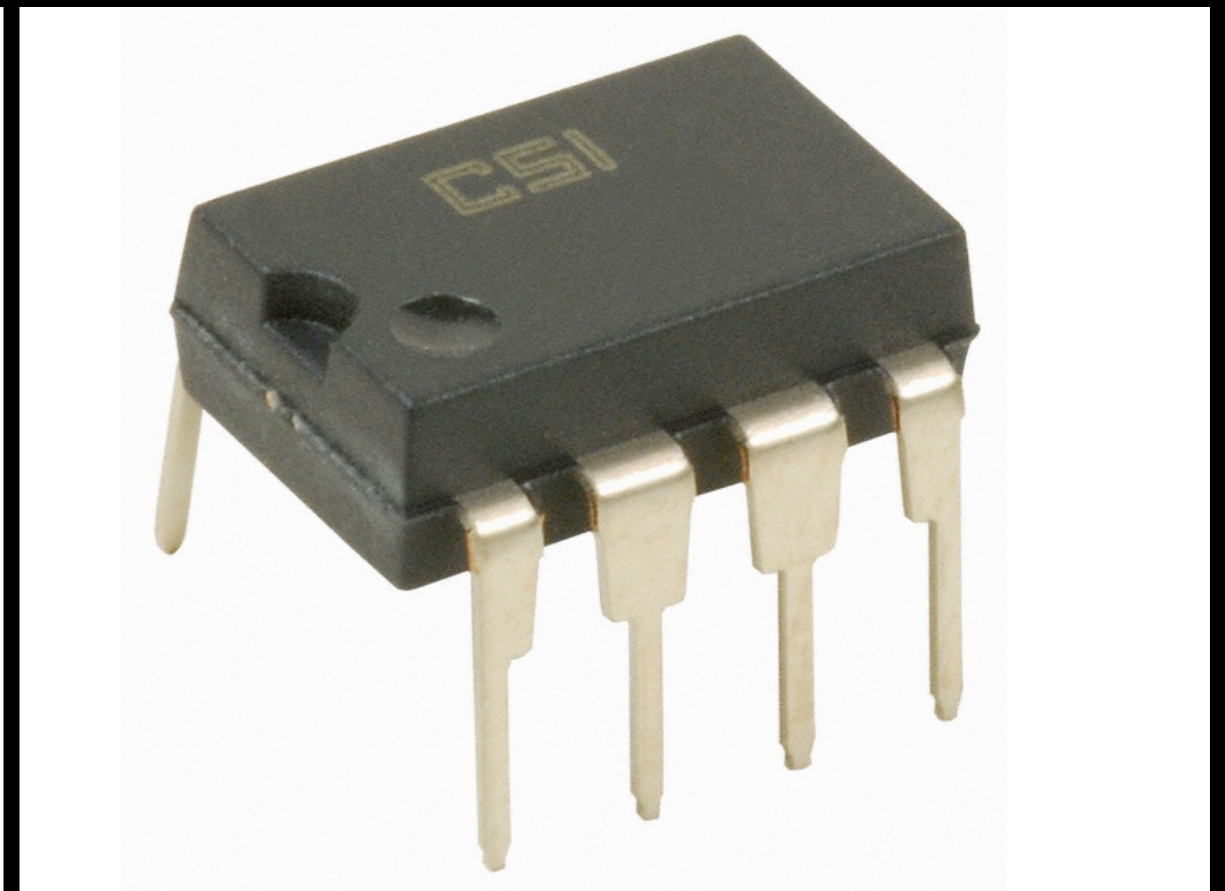


# Digital ICs

## Memory

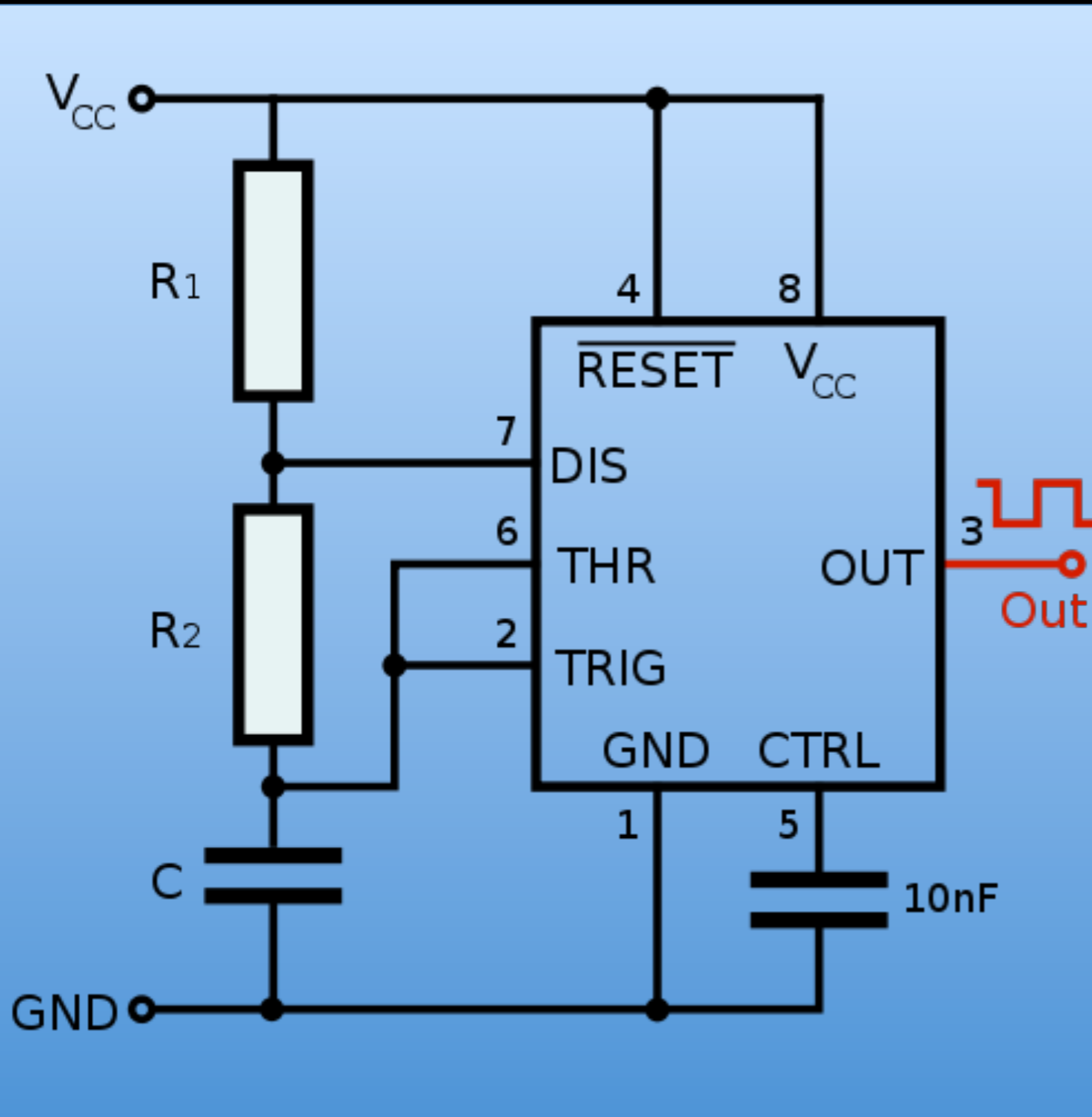
### FLASH

### 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 :

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

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

Figure 4

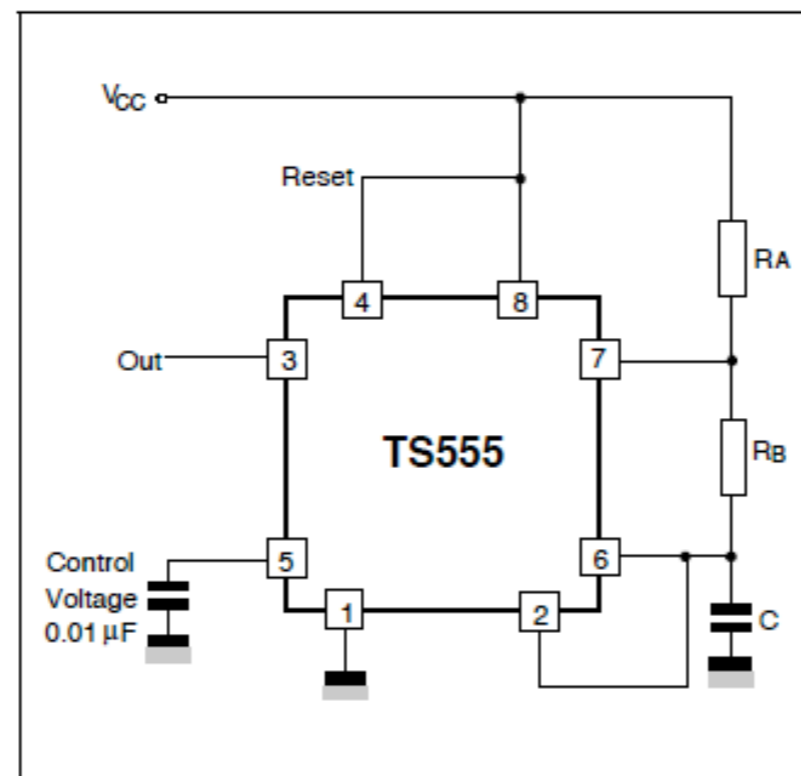
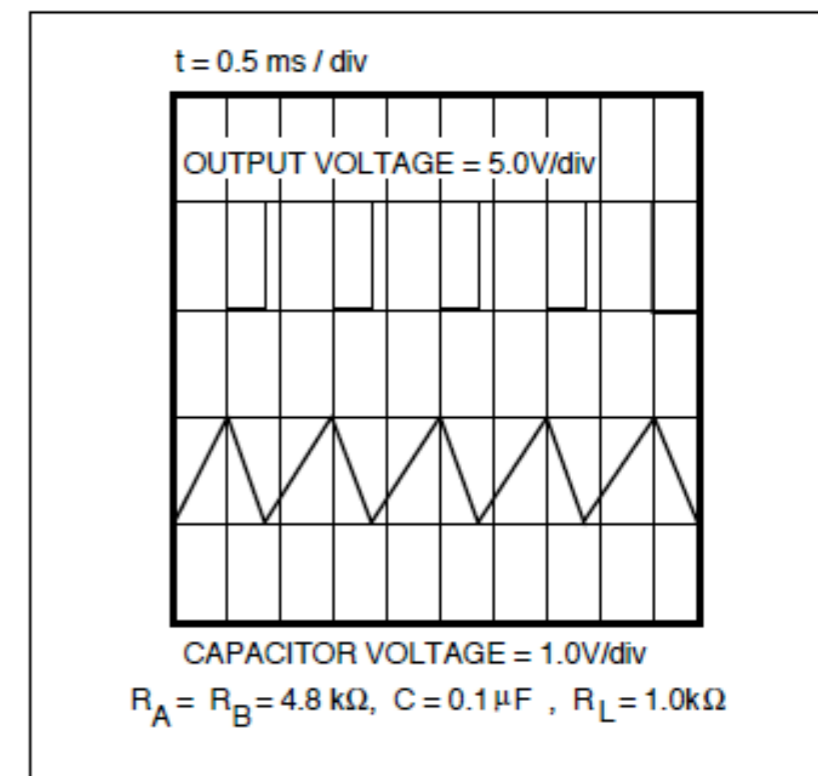


Figure 5



# Digital ICs

## 555 Timer



[HOME](#) | [LEARNING](#) | [CIRCUITS](#) | [LINKS](#) | [EMAIL US](#)

**555 Timer Circuits**

555 Timers are fun and a great way to start learning electronics

[TI Li-Ion Battery Charger](#) Power-Path Management IC, Vout 5.5V Smallest, USB Compliant Solution. [TI.com/bq24075np/](http://TI.com/bq24075np/)

[MDrive Integrated Motion](#) IMS Integrated Step Motor & Driver All-in-One MotionControl Technology [www.imshome.com](http://www.imshome.com)

[Electric Circuits Help](#) Electric Circuits 9th Edition Solutions. View Free! [Cramster.com/Nilsson](http://Cramster.com/Nilsson)



AdChoices

## 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 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

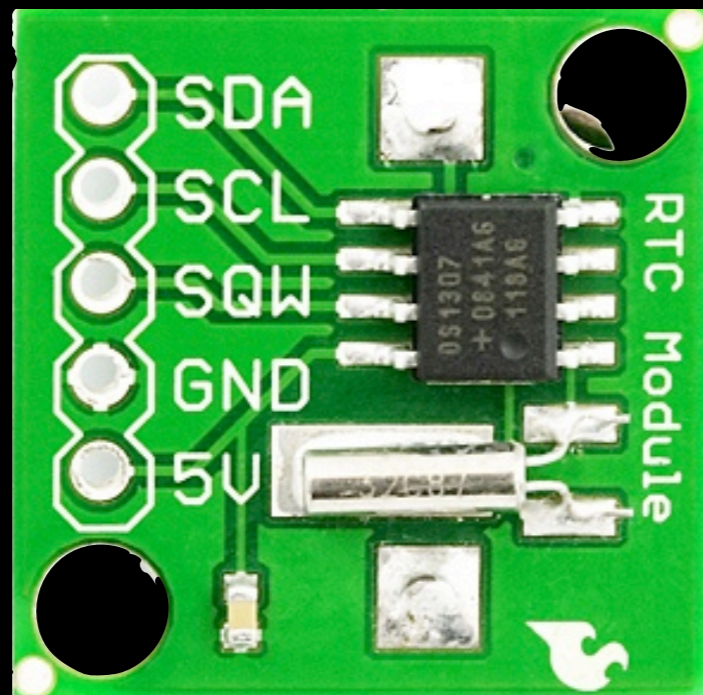
The following are complete electronic circuits that you can build, they all utilize the 555 Timer circuit.

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

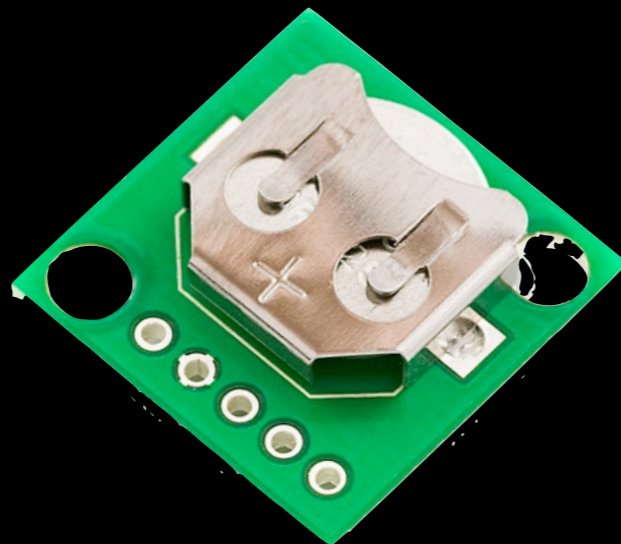
# Digital ICs

## Real-Time Clocks

DS1307



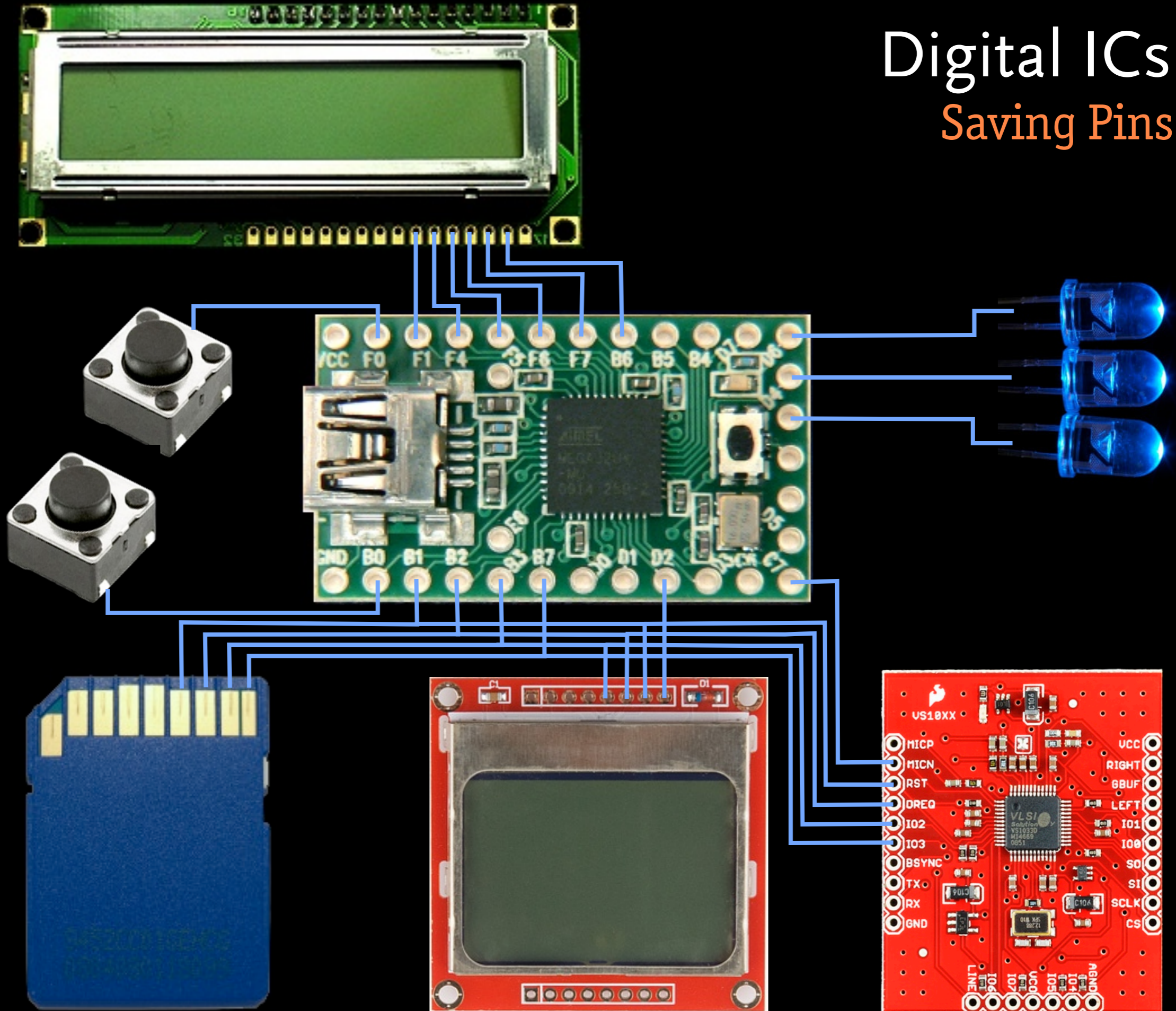
- ❑ Seconds, minutes, hours, date, month and year
- ❑ Battery lasts for 7-9 years
- ❑ Keeps track of leap years!



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

# Digital ICs

## Saving Pins

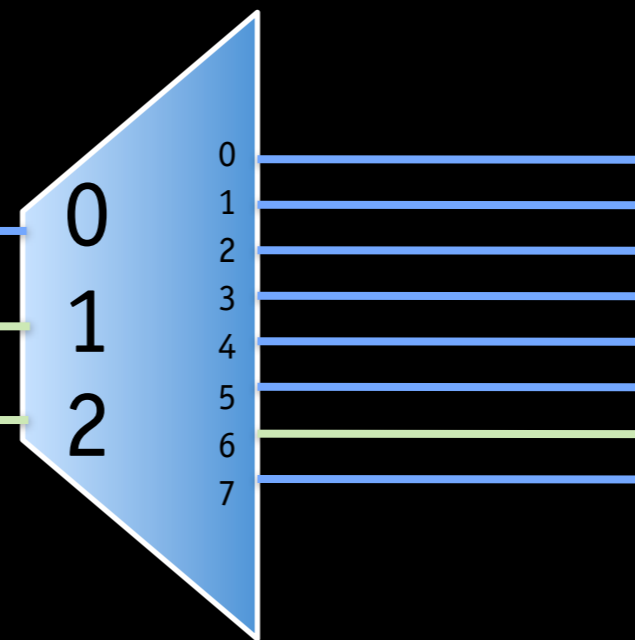


# Digital ICs

## Decoder



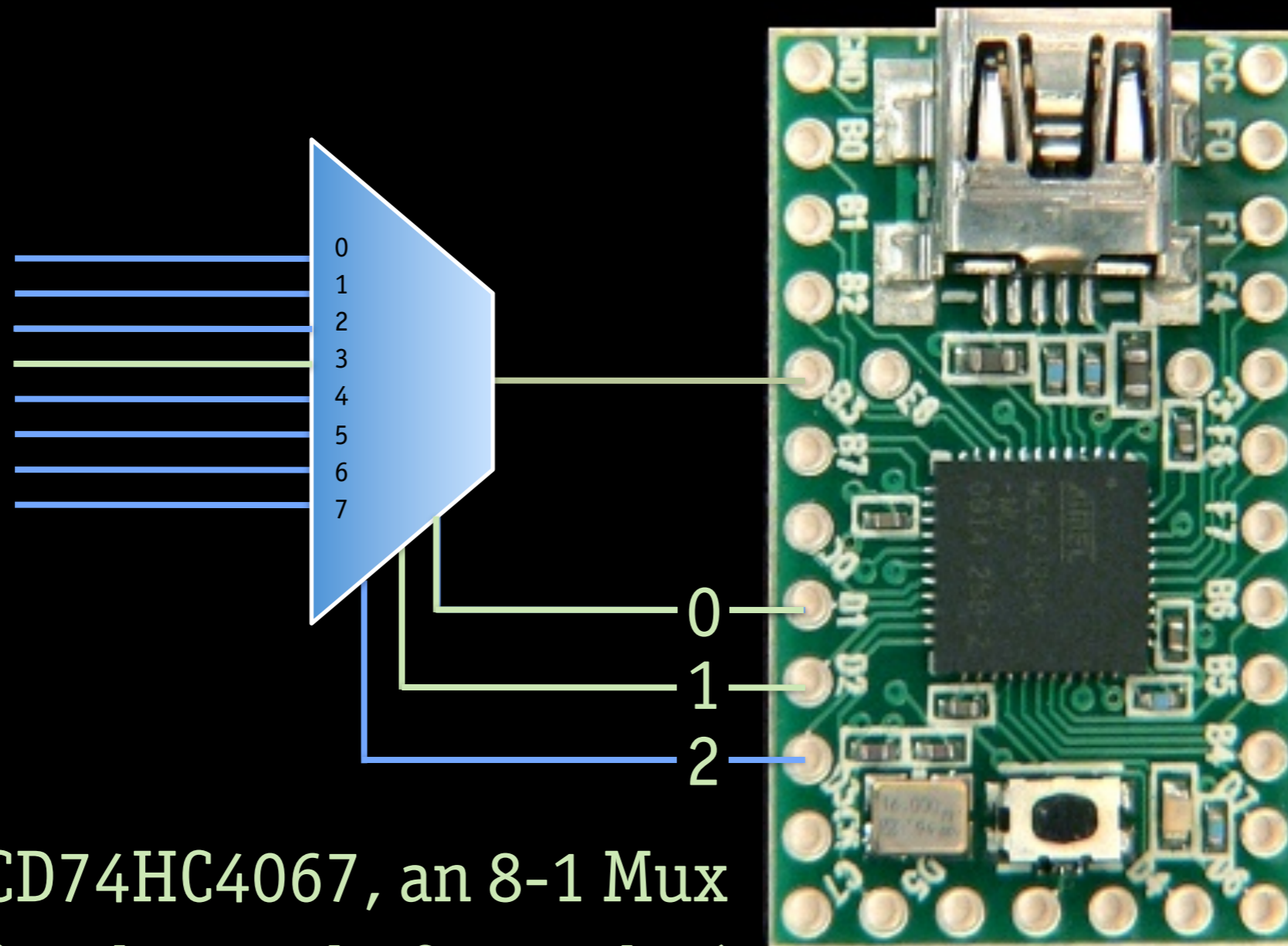
74HC238, a  
3-8 Decoder



Chip  
Select Lines

# Digital ICs

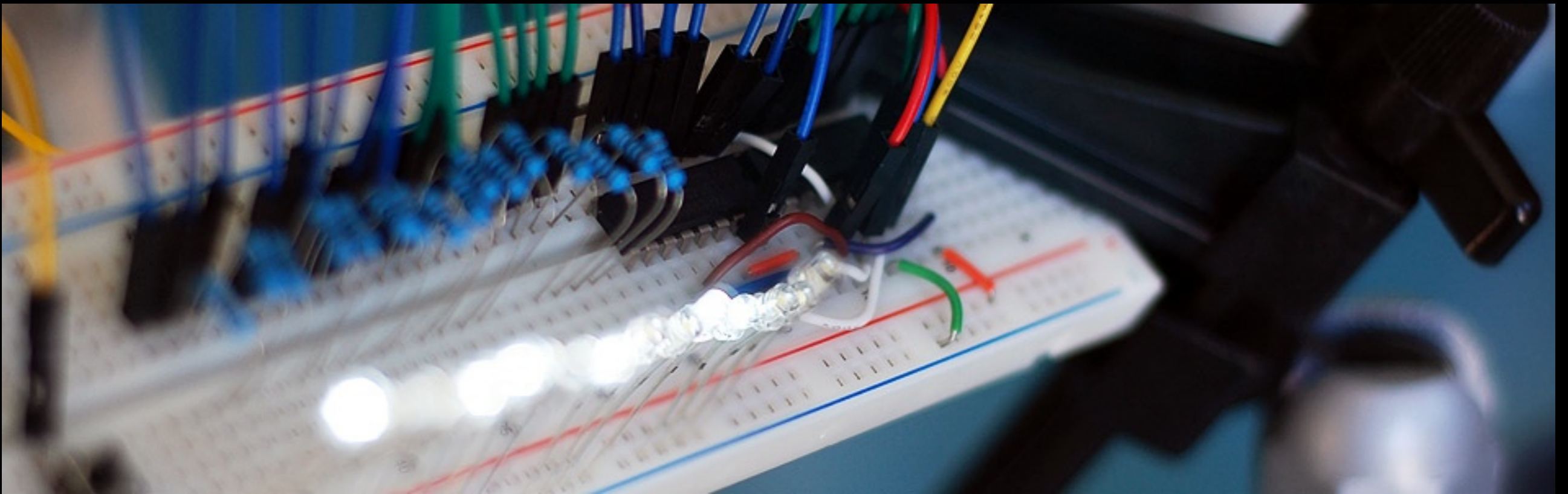
## Multiplexer



CD74HC4067, an 8-1 Mux  
(It Also Works for Analog)

# Digital ICs

## Shift Register

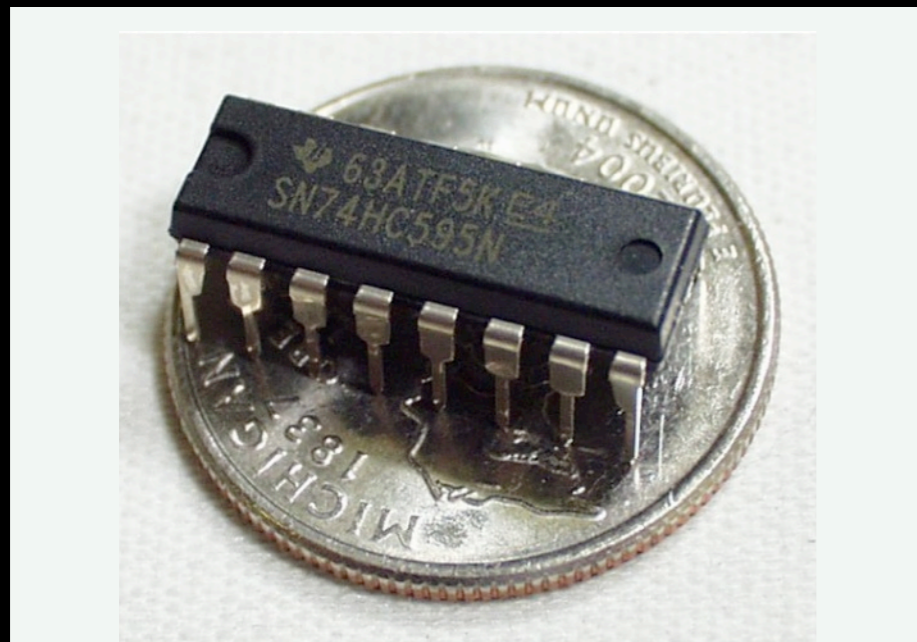


Say you want to control a huge number of LEDs with your Teensy...

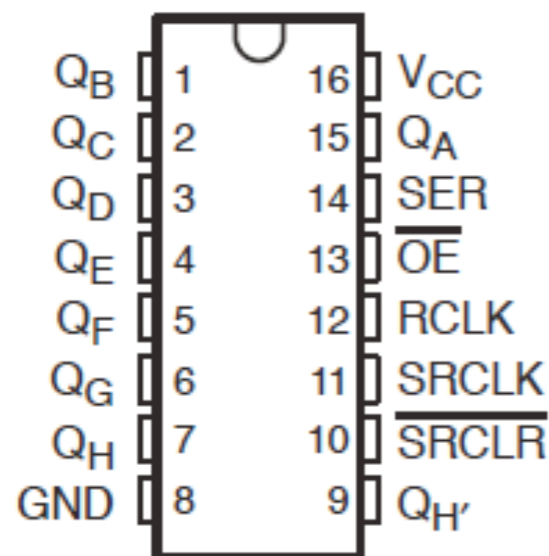
Visit <http://bildr.org/2011/02/74hc595/>

# Digital ICs

## Shift Register



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



What does a shift register really do?

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

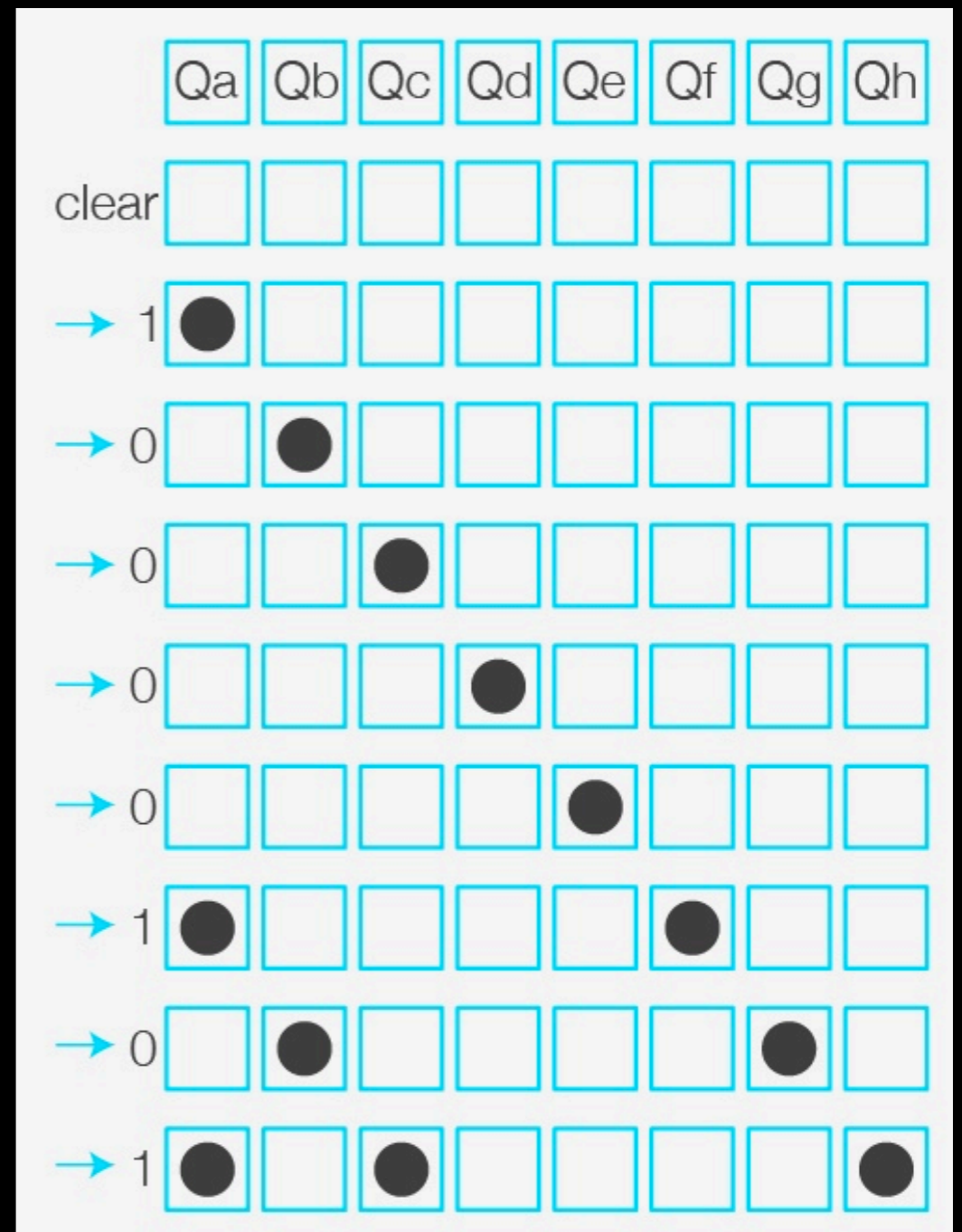


# Digital ICs

## Shift Register

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

- ❑ 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

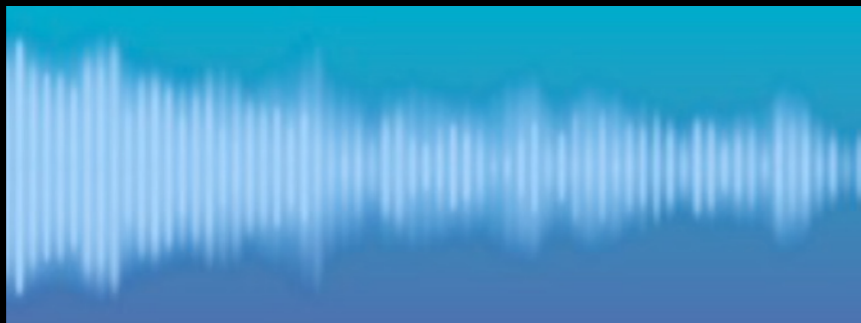


# Integrated Circuits

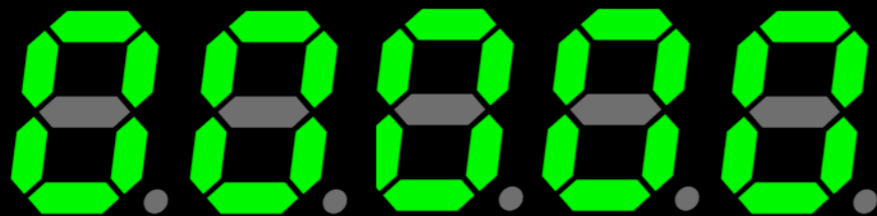
There Are So Many More!

Specialized Processors (Like Your Mp3 Decoder)

- Digital Signal Processors



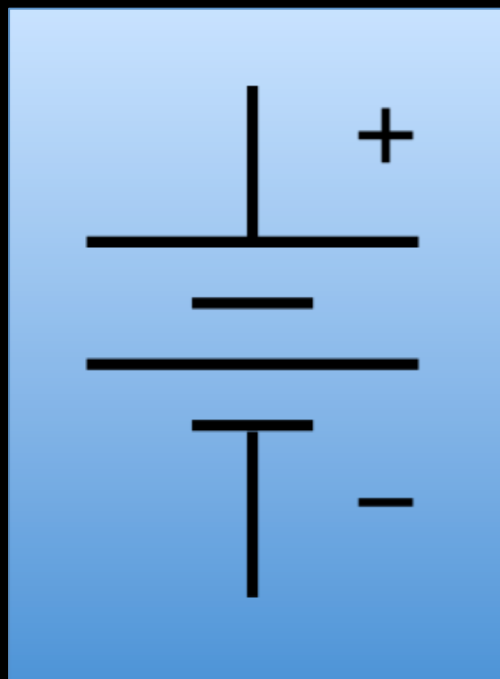
- 7-Segment Display Driver (MC14489B)



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

# Choosing ICs

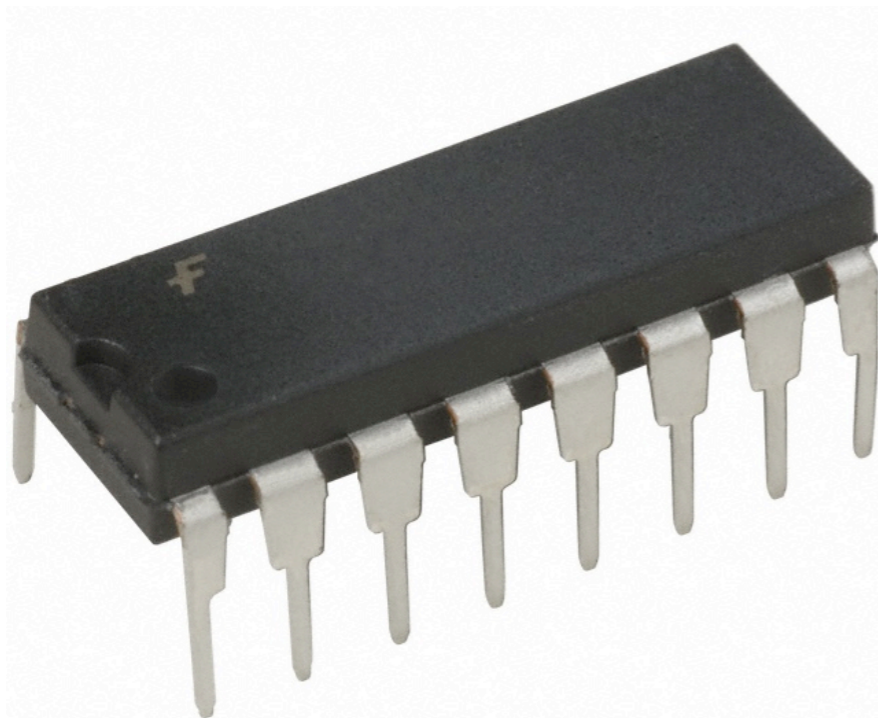
## Voltage & Current



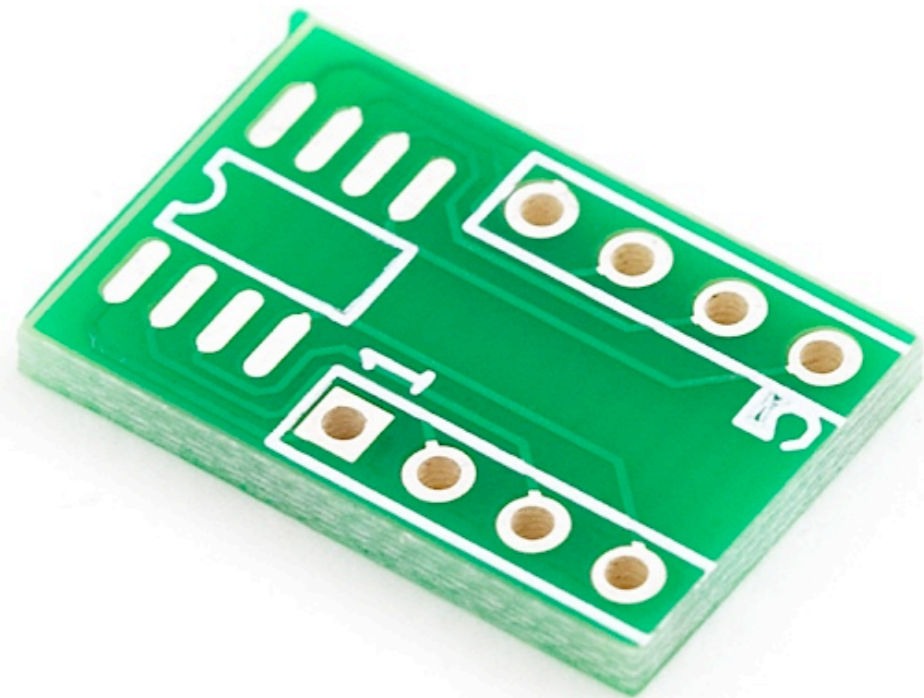
- ❑ Choosing parts with the same operating voltage will save major headaches and components!
- ❑ Keep your power source in mind. USB, batteries and voltage regulators can only provide so much current.

# Choosing ICs Packaging

16-DIP



8-SOIC to DIP Adapter



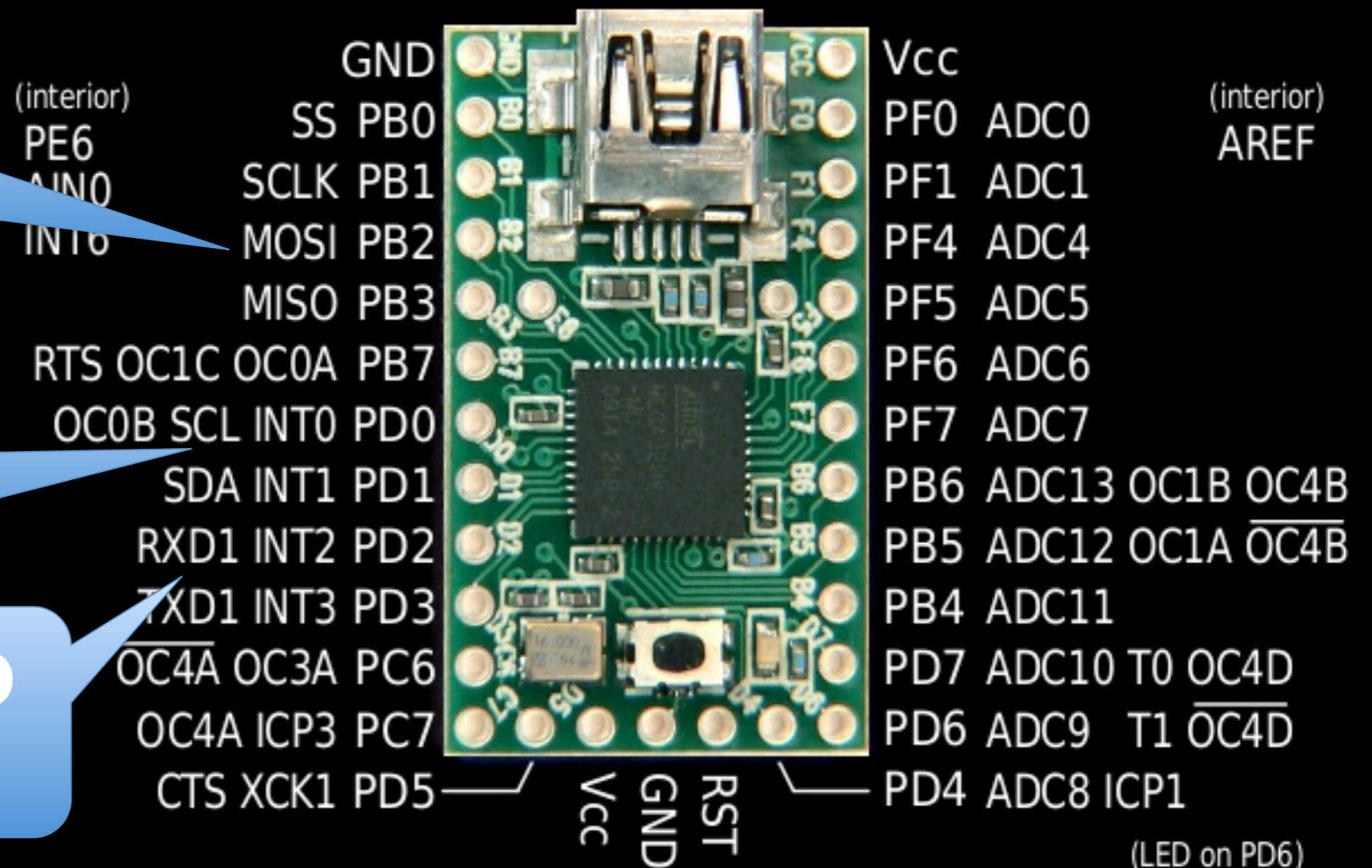
# Choosing ICs

## Communication Interface

SPI?

I<sup>2</sup>C?

UART?



# Choosing ICs

## Community Support



ARDUINO PLAYGROUND

The playground is a publicly-editable wiki about [Arduino](#).

[Manuals and Curriculum](#)

[Board Setup and Configuration](#)

[Development Tools](#)

[Interfacing With Hardware](#)

- [Output](#)
- [Input](#)
- [User Interface](#)
- [Storage](#)
- [Communication](#)
- [Power supplies](#)
- [General](#)

[Interfacing with Software](#)

[Code Snippets and Sketches](#)

- [Libraries](#)
- [Tutorials](#)

[Suggestions & Bugs](#)

[Electronics Technique](#)

[Sources for Electronic Parts](#)

[Related Hardware and](#)

[Initiatives](#)

[Arduino People/Groups & Sites](#)

[Exhibition](#)

[High Resolution Photo of an Arduino Board](#)

## :: The Arduino Playground ::

Welcome to the Arduino Playground, a wiki where all the users of [Arduino](#) can contribute and benefit from their collective research.

This is the place to post and share your own code, circuit diagrams, tutorials, DIY instructions, tips and tricks, and after all the hard work, to show off your projects! Anyone can edit and add to the pages here.

Arduino Playground is a **work in progress**. We can use all the help you can give, so please read the [Participate](#) section and get your fingers typing!

NOTE (20081002): Solved the problem for adding content in non-western Latin languages, now it is possible to start typing in any language on the internet [test here](#)

## Playground Content Tree

### [Manuals and Curriculum](#)

- More Good starting places - Cohesive documentation that will step you through a variety of topics.

# Lab 6 Review

Using Your MP3 Decoder (Finally!)

Interrupt Handlers (ISRs)

Debouncing Buttons

# Lab 6 Preview

## Interrupts



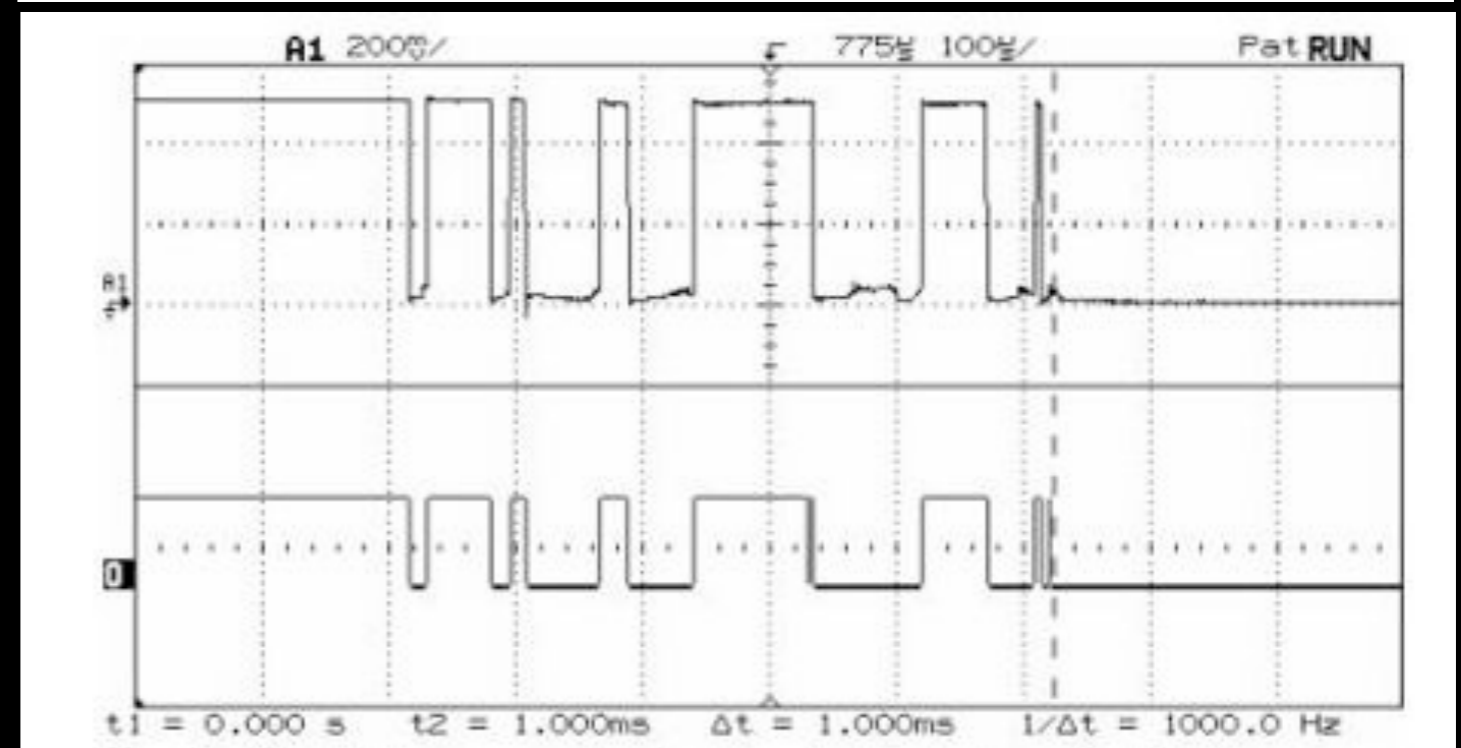
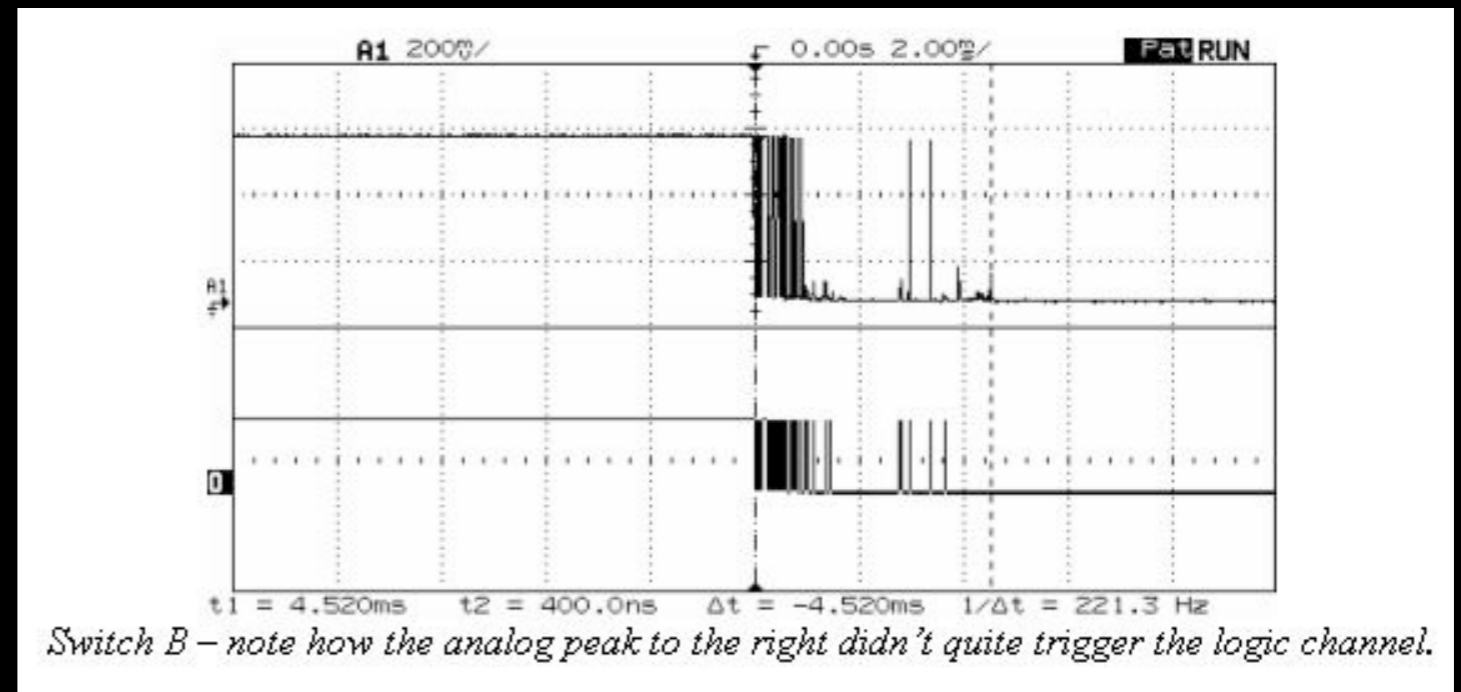
Write a special function (an 'Interrupt Service Routine' or ISR) that gets called exactly when the button is pressed.



# Lab 6 Preview

## Debouncing

What happens when you press a button?



# Lab 4 Data Loggers

*Vote for a Favorite Project, Win Swag!*