

# Journey to the Center of the HP28C/S



by  
Travis Goodspeed  
<travis at utk.edu>

regarding the work of  
Paul Courbis  
Sébastien Lalande

# Disclaimer

I am just the translator.  
My French sucks.

# HP 28S



Photo by  
Kurt Moerman

# Part 1: Software

- Objects: List and Structure
- I/O Zone
- RAM
- Accessing Machine Language
- Further Discoveries
- Useful Routines

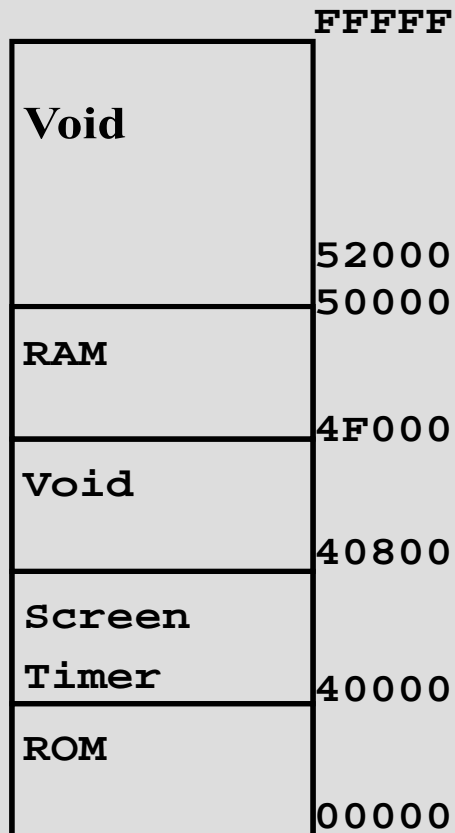
# Introduction

- Three Models
  - HP28C-1BB
  - HP28C-1CC
  - HP28S-2BB
- HP SATURN architecture, same as HP 71b
- Nibble-wise Little Endian

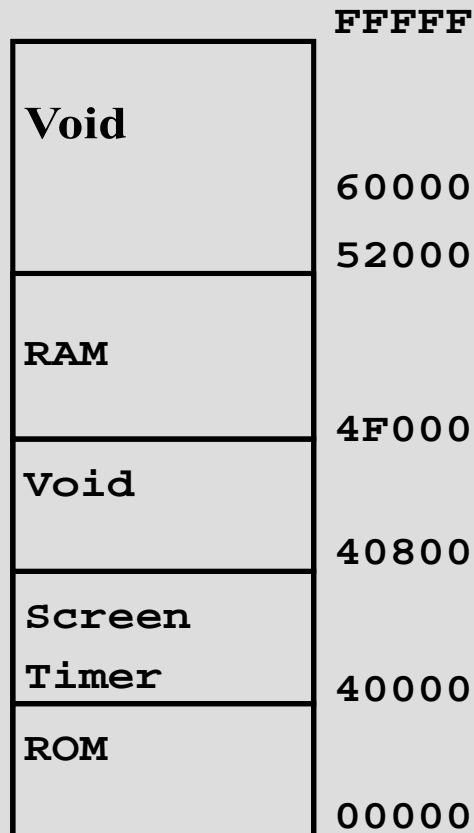
# Memory Layout

## HP 28C

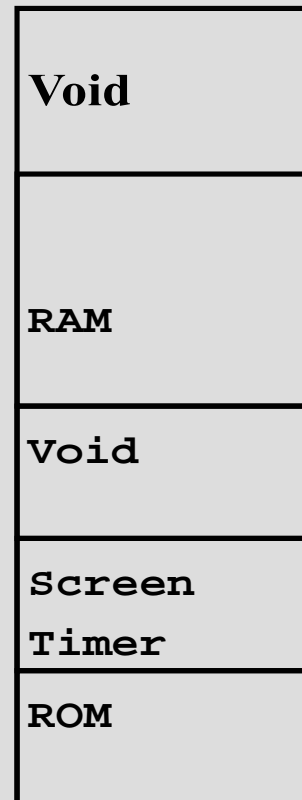
### Standard



### 6 Ko

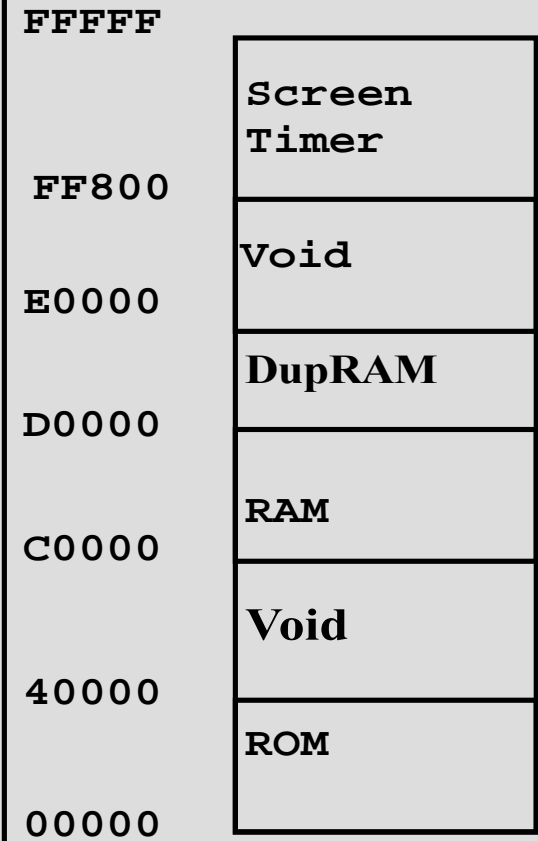


### 34 Ko



## HP 28S

### Standard



# Registers

I/O

OUTPUT

12

INPUT

16

Status

CARRY

1

HW Status

4

Status

16

Pointers

P

20

D0,D1

20 each

PC

20

Safeguard

RSTK

20 by 8

R0 to R4

64 each

Working

A to D

64 each

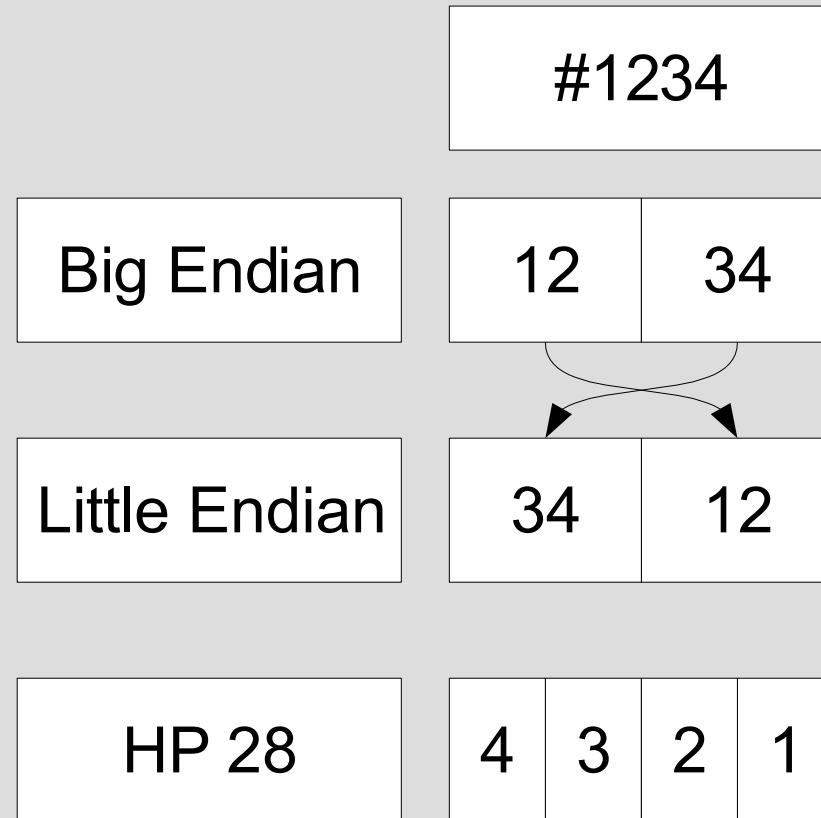
# Nybbles (Quartets)

00	01	...
----	----	-----

0	1	2	4	...
---	---	---	---	-----

- Byte
  - 0x00 to 0xFF
  - 0 to 255
- Nibble
  - 0 to F
  - BCD
    - 0 to 9

# Nybble Little Endian



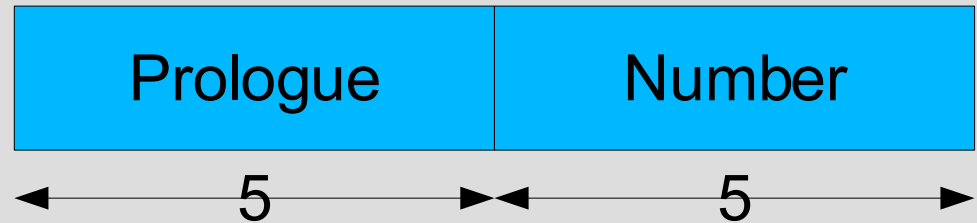
# Objects: List and Structure

- Prologue
  - 5 nibbles
  - Specifies Type
- Type
  - System or User
  - Implies Length

Object	Prologue	Page
Short Integer <sup>1</sup>	#02991	10
Real	#02933	11
Extended Real <sup>1</sup>	#02955	11
Complex	#02977	11
Extended Complex <sup>1</sup>	#0299D	12
Byte <sup>1</sup>	#029BF	12
First Unknown Type <sup>1</sup>	#029E1	12
Array	#02A0A	13
Second Unknown Type <sup>1</sup>	#02A2C	13
String	#02A4E	14
Binary Integer	#02A70	14
List	#02A96	15
ROM/RAM Pair <sup>1</sup>	#02AB8	15
Algebraic	#02ADA	19
Program	#02C67	21
Assembly Code <sup>1</sup>	#02C96	22
Global Name	#02D12	22
Local Name <sup>1</sup>	#02D37	22
ROM Pointer <sup>1</sup>	#02D5C	23

Table 2.1: Object Types and their Prologues

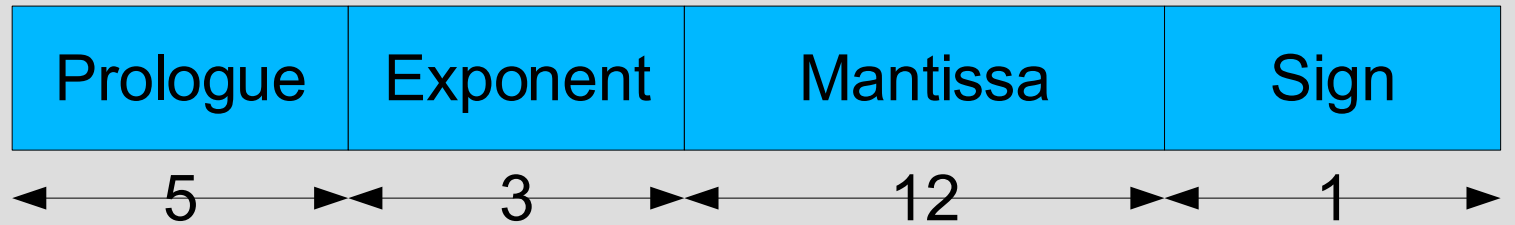
# Short Integer #02911



00000
12345

11920	00000
11920	54321

# Real #02933

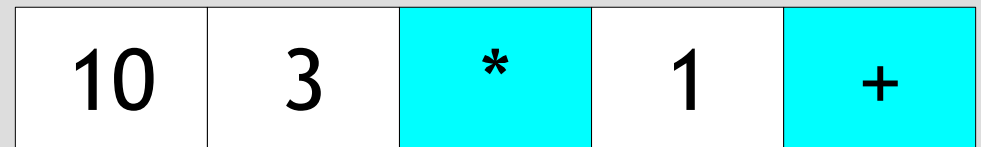
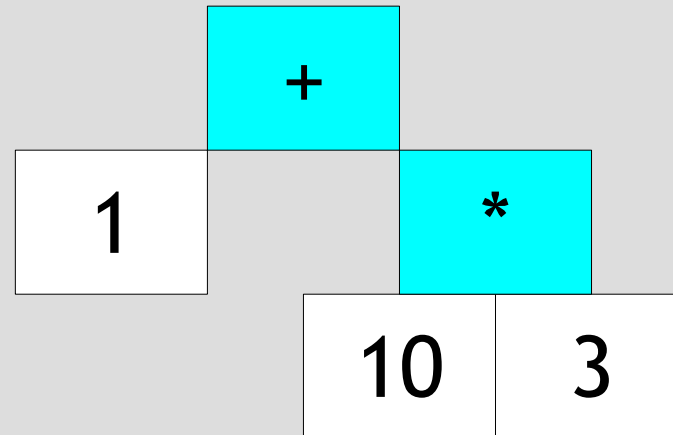


#02933	5	3.14159...	+
33290	500	953562951413	0

$\pi * 10^5$

# Reverse Polish Notation

- Stack Metaphor
- Left to right.
  - If value
    - PUSH onto stack.
  - If operator,
    - POP parameters.
    - PUSH result.
- No grouping symbols.
- Easy to interpret.



# RPN Interpretation

- Input

10	3	*	1	+
----	---	---	---	---

10	3	*	1	+
----	---	---	---	---

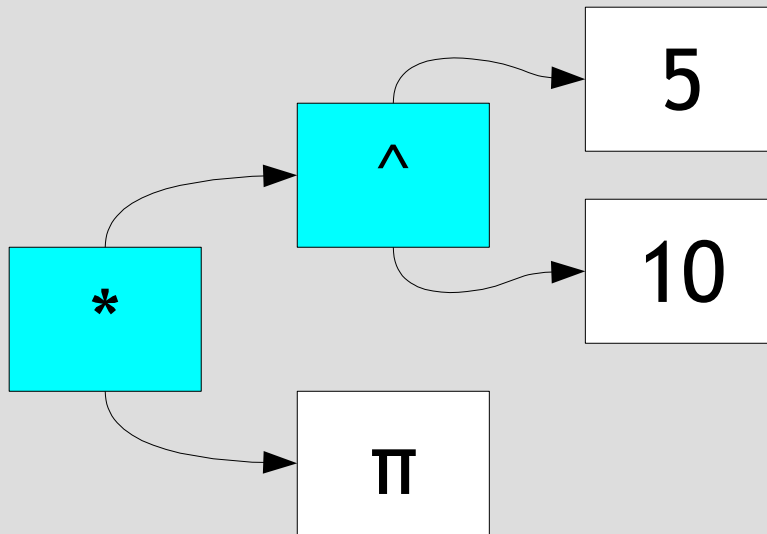
- Stack

1
31

# Algebraic #02ADA



$\pi * 10^5$

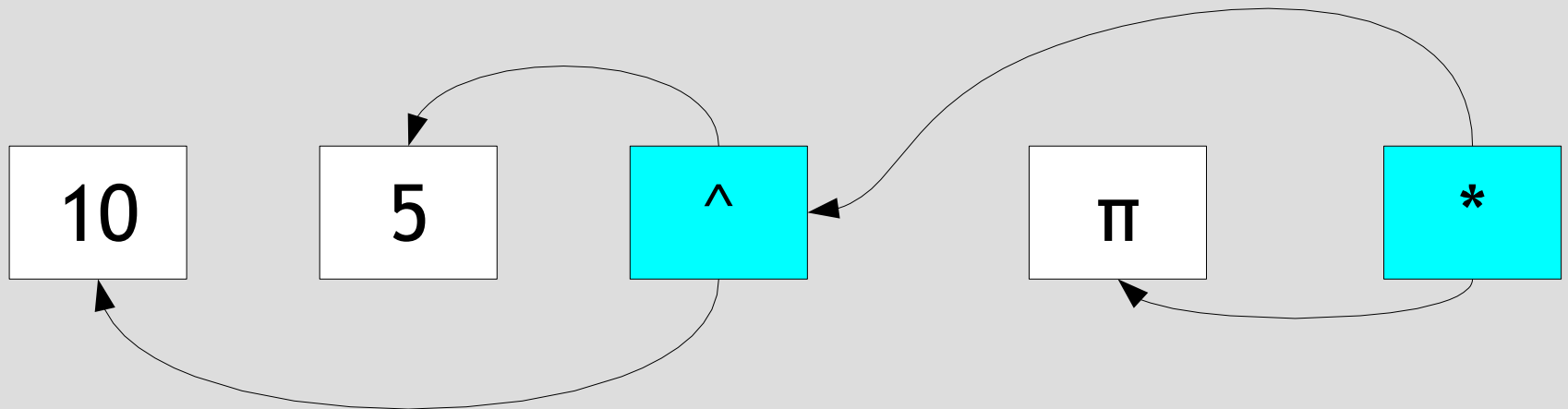


# Algebraic #02ADA



$\pi * 10^5$

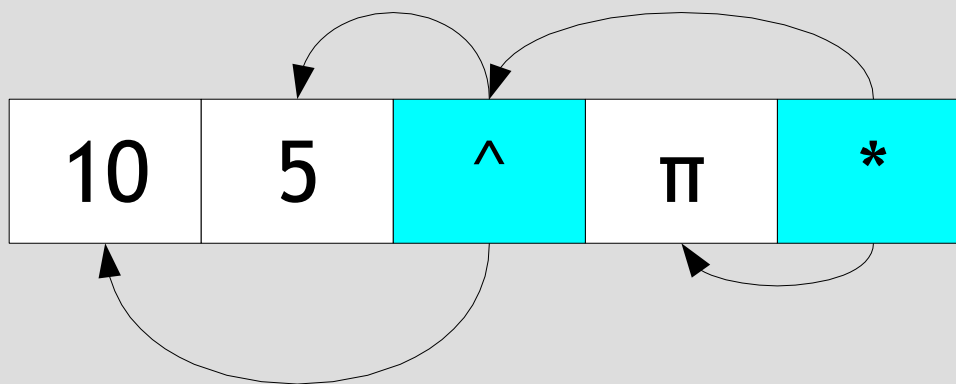
$\pi$  \* 10 ^ 5



# Algebraic #02ADA



$\pi * 10^5$



# Algebraic #02ADA



$\pi * 10^5$

$\pi$  \* 10 ^ 5

Prologue 10 5 ^  $\pi$  \* End. Alg.

33290 000 953562951413 0

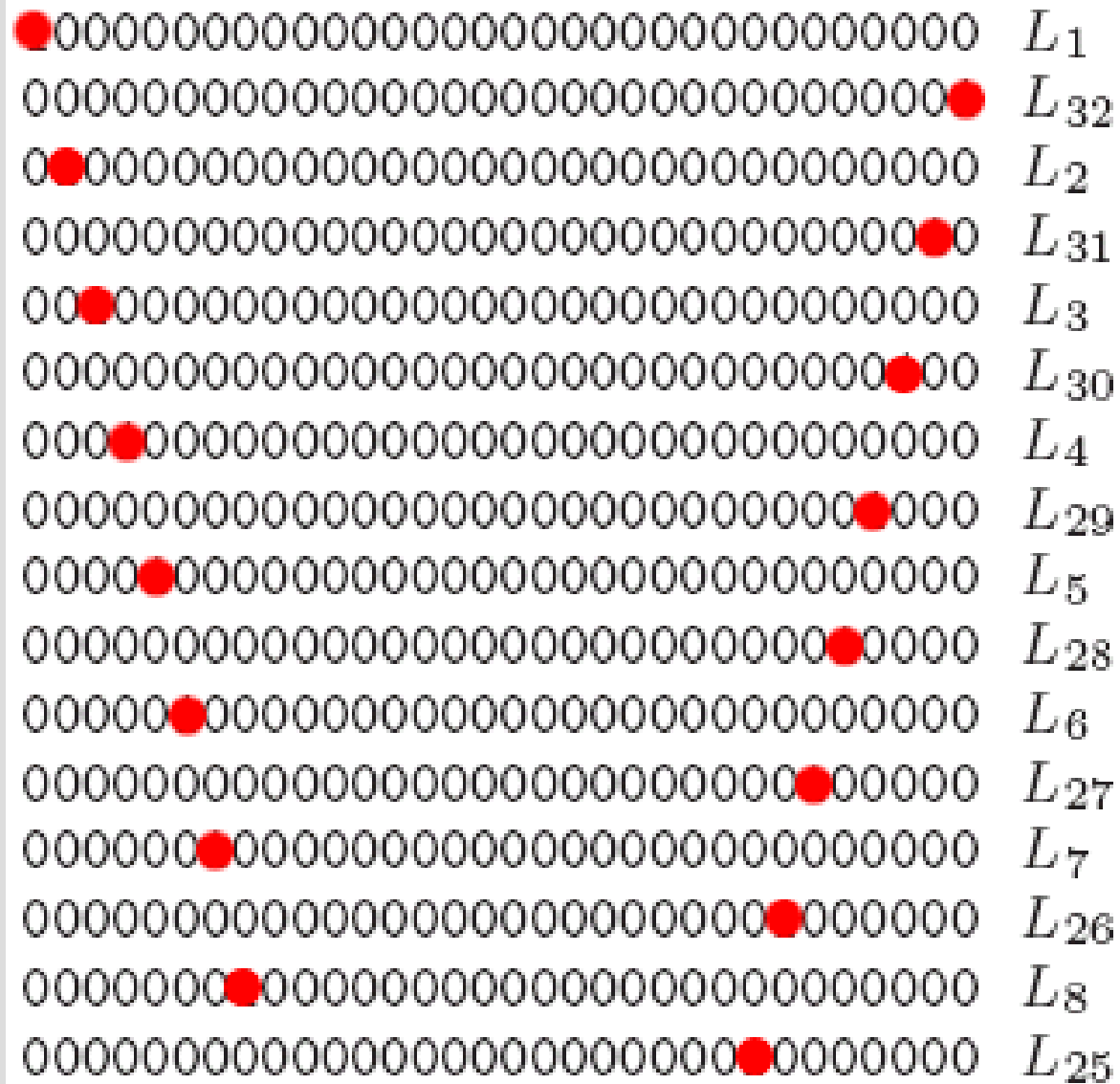
# I/O Zone

- IR in/out
- Timers
- Contrast
- Screen/Indicators
- Speed
  - 28S Only
- Row Driver Waveform
  - 28C Only

# Row Driver Waveform

- Maps logical lines to physical lines.
- Scrolling
  - Swap line-mappings, not data.
- Mirroring
  - Map the same logic line to many physical lines.
- Initialized at boot.
- Interpreted by software video driver.

# Row Driver Waveform

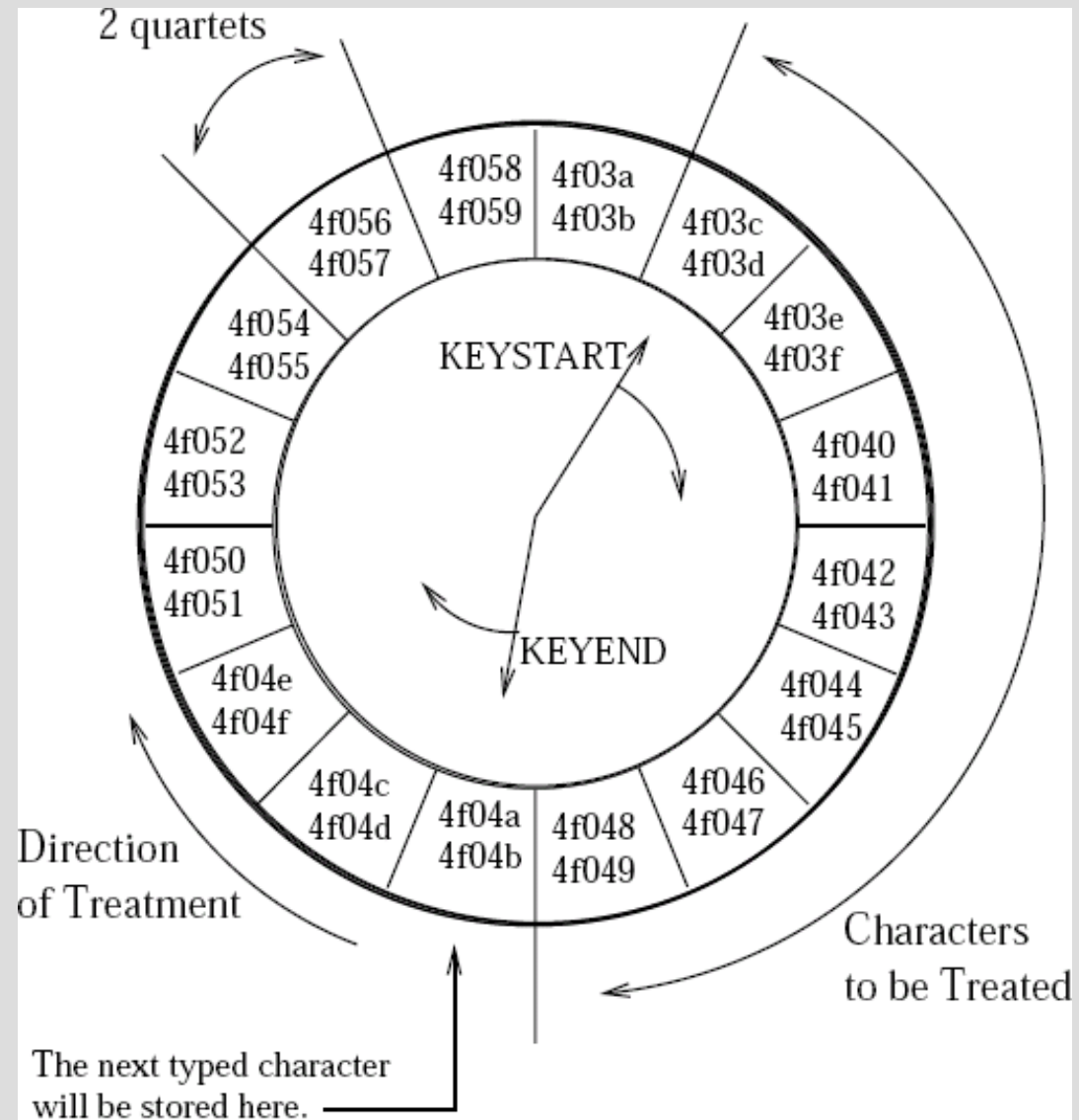


# RAM

- Polling
  - Background Processes Watch RAM
  - I/O-like Behavior
- Keyboard Buffer
- Flags
- Command-Line
- Stacks
- Temp Environment

# Keyboard Buffer

- Samples Keys
  - Not ASCII
- Clock
- KEYSTART
  - next to be handled
- KEYEND
  - next to be sampled



# Accessing Machine Language

- ML Program
  - is an Object Type (#02C96)
  - may not be created by the user.
- Bootstrapping
  - ASS
  - LASS

# LASS

- Takes as input an ML string.
- One character, one nybble.
- Assembles
  - ML String to ML Object
  - not Assembly String to ML Object
- Usage
  - Push string to stack.
  - LASS
  - Store resultant object to variable.

**INV.VID**

# INV.VID

76C20	91C70	69C20	E5000	13310	334FD	2041F
87004	72203	4FD50	41F00	40470	10113	13114
21648	08C27	1531B	9C151	11771	33131	8BA9E
20010	9F20					

- “76C2091C7...
- LASS
- `INV.VID' STO

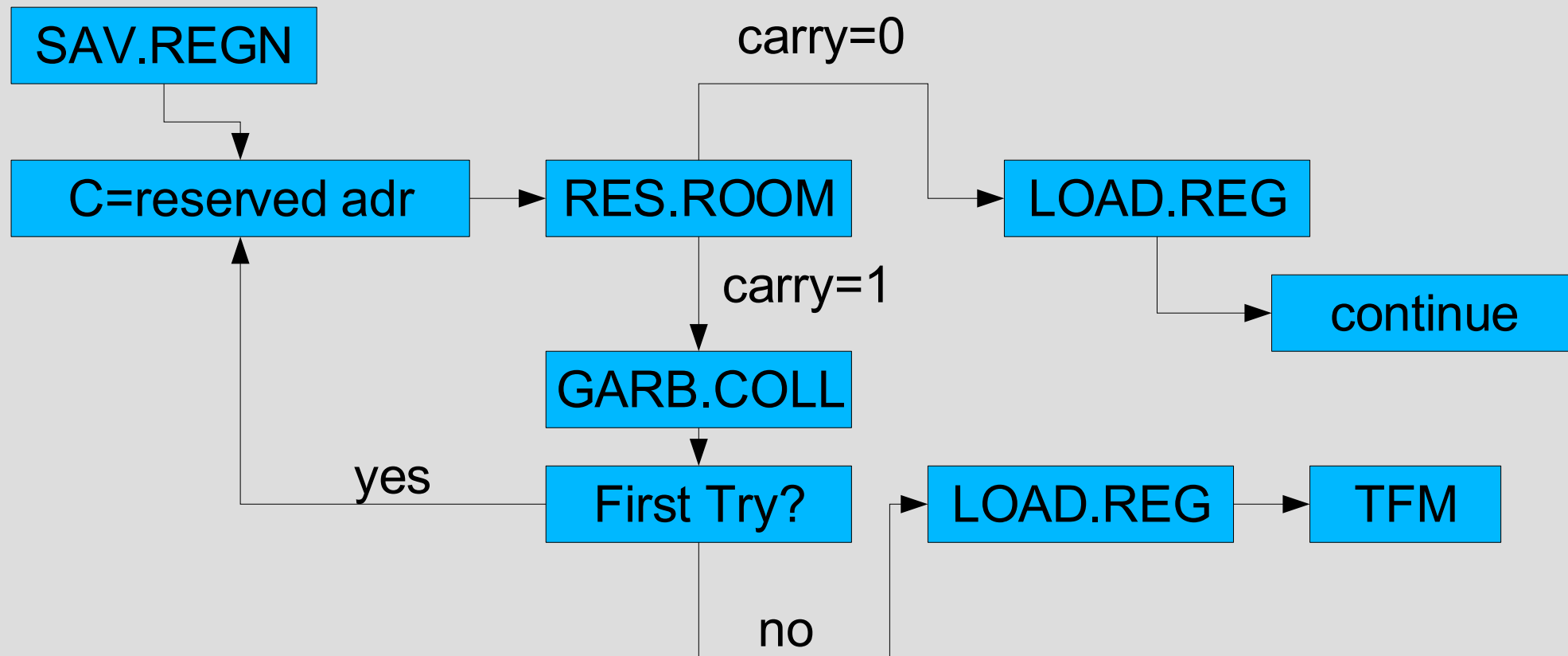
# Further Discoveries

- Search ROM
  - Strings Tables in Appendix
  - I/O or RAM addresses
- Systematic Disassembly
  - Disassembler Program
  - Dump to a Workstation

# Useful Routines

- SAV.REG
- LOAD.REG
- RES.ROOM
- GARB.COLL
  - Force Garbage Collection
- TFM
  - Too Few Memory
- ERROR
- BEEP

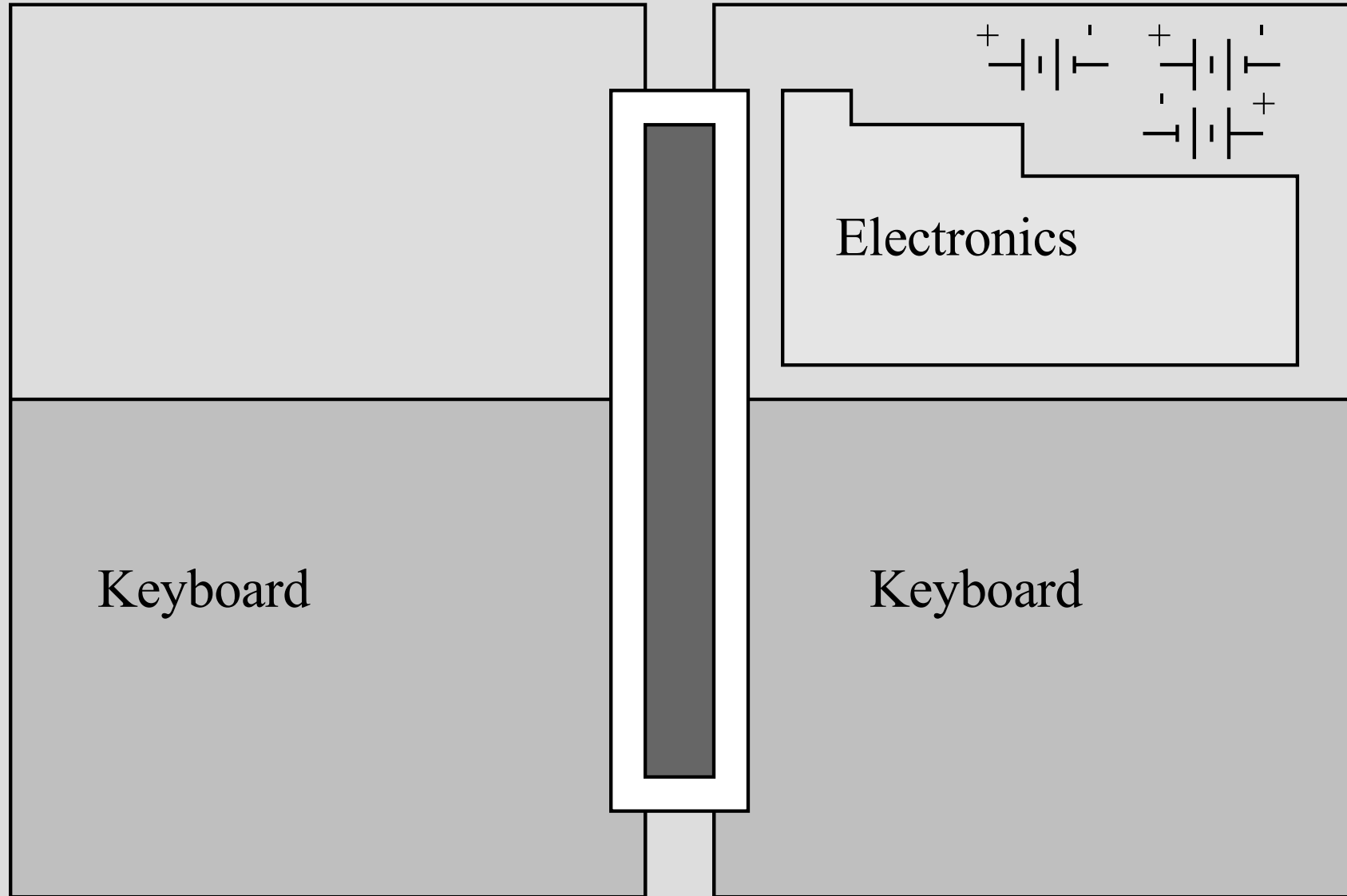
# Useful Routines



# Part 2: Hardware

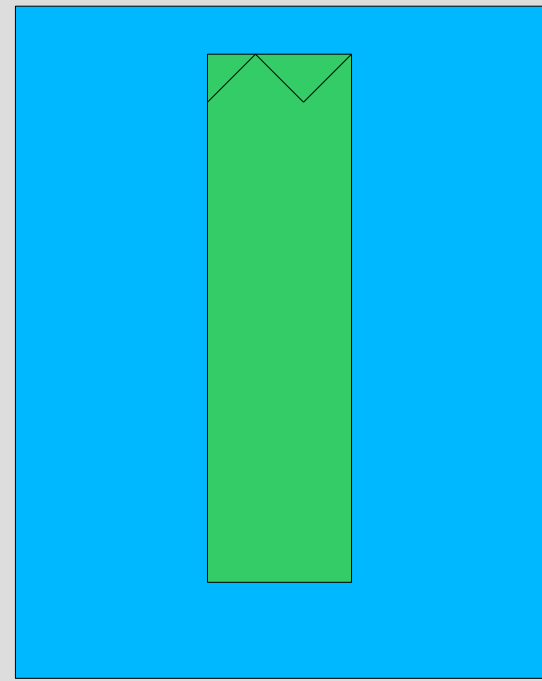
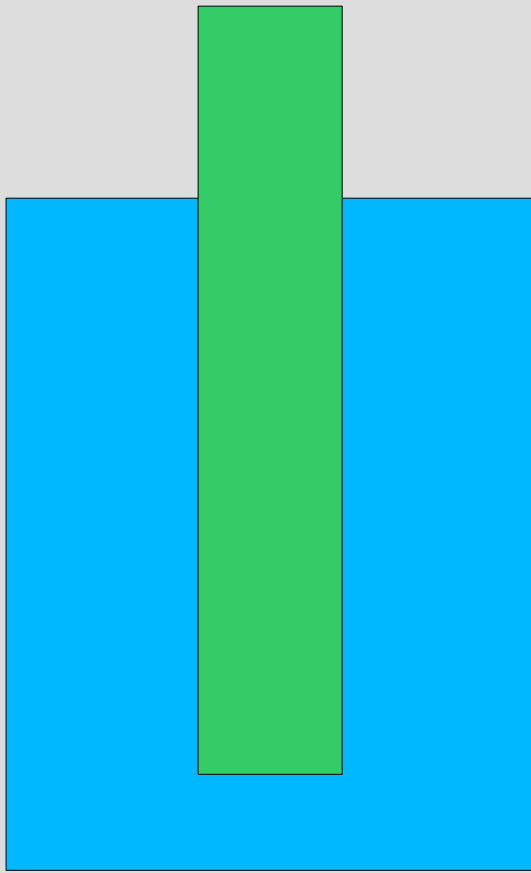
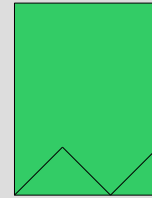
- Exterior Description
- Opening the Machine
- Interior Description
- Transformations
  - External Power Feed
  - Overclocking
  - Extended Memory
- The I/O Port
- Closing the Machine
- I/O Interfaces

# Exterior Description

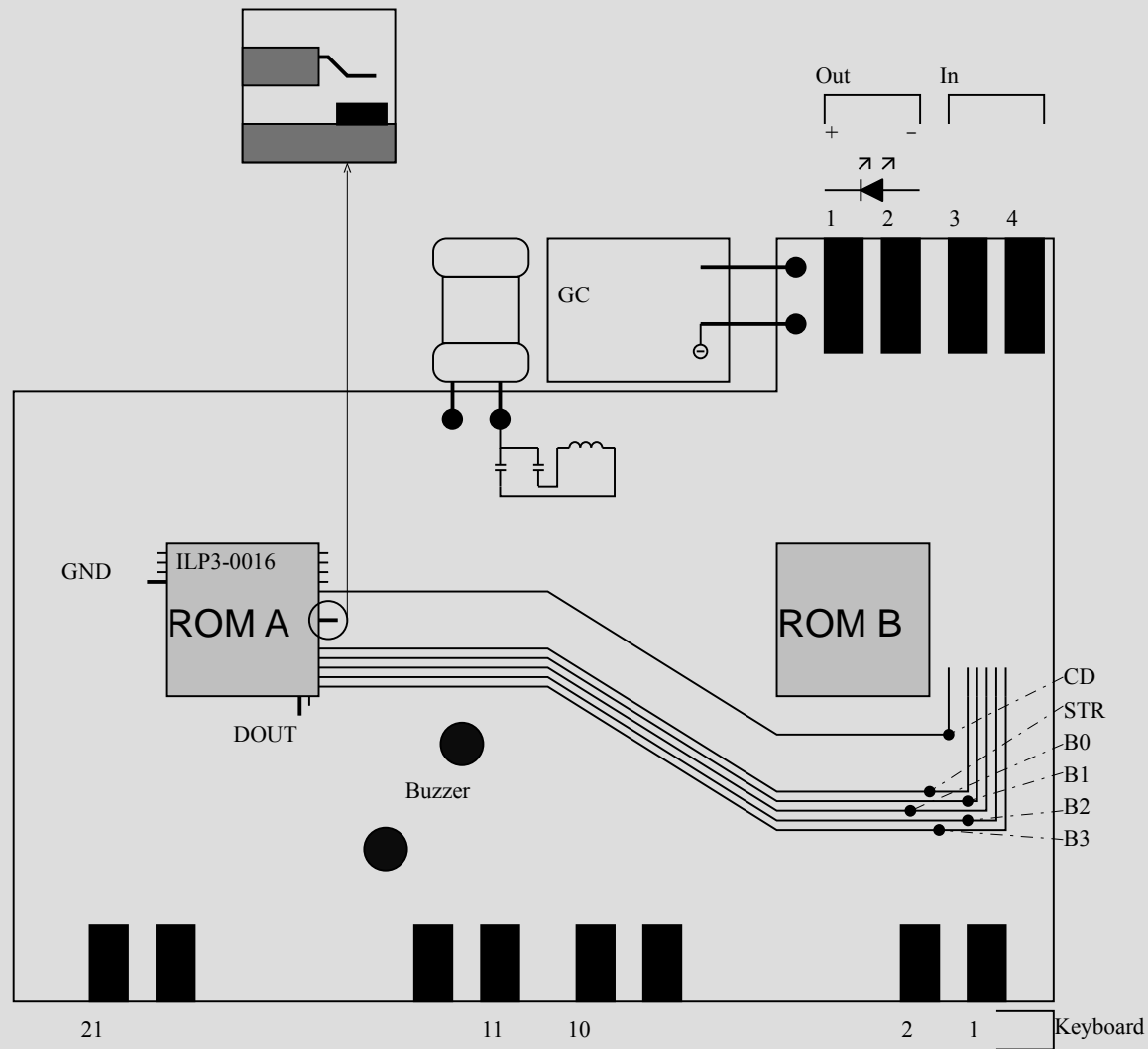


# Opening

- No Screws!
- Gotta break some pins.



# Interior Description



# Transformations

- External Power Feed
- Overclocking
- Memory Upgrade

# Overclocking

- HP28C
  - Replace or Remove a Capacitor
  - Replace an Inductor
- HP28S
  - Overlocking in Software
  - SPEED Program

# Memory Upgrade

- HP71B Memory Module
- Installation
  - Cut Traces of Memory Bus
  - Patch in Upgrade Module

Educalc:

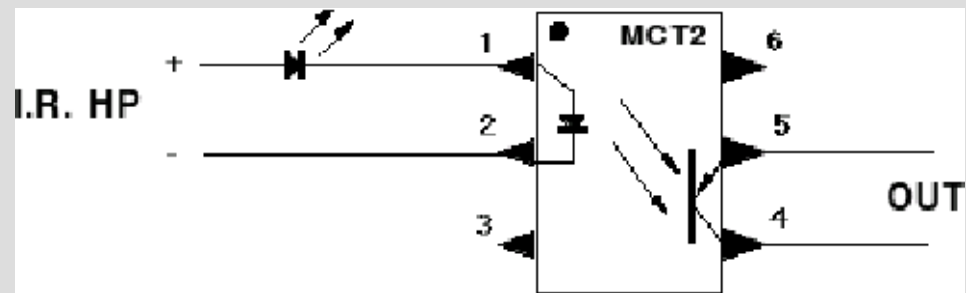
27953 Cabot Road  
Laguna Niguel  
CA 92677  
U.S.A  
Tel: 19-1-714-582-26-37

Module:

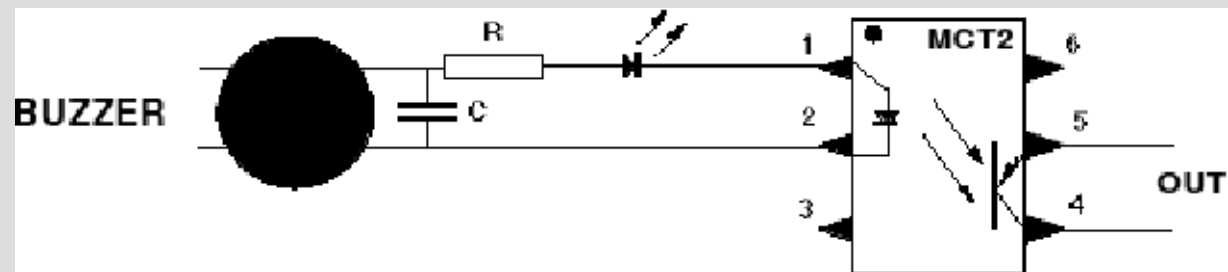
64K RAM	#71-656B	\$294.95
64K RAM	#71-656	\$149.95
4K RAM	#82-420	\$ 65.95

# I/O Ports

- Hack of the IR Led

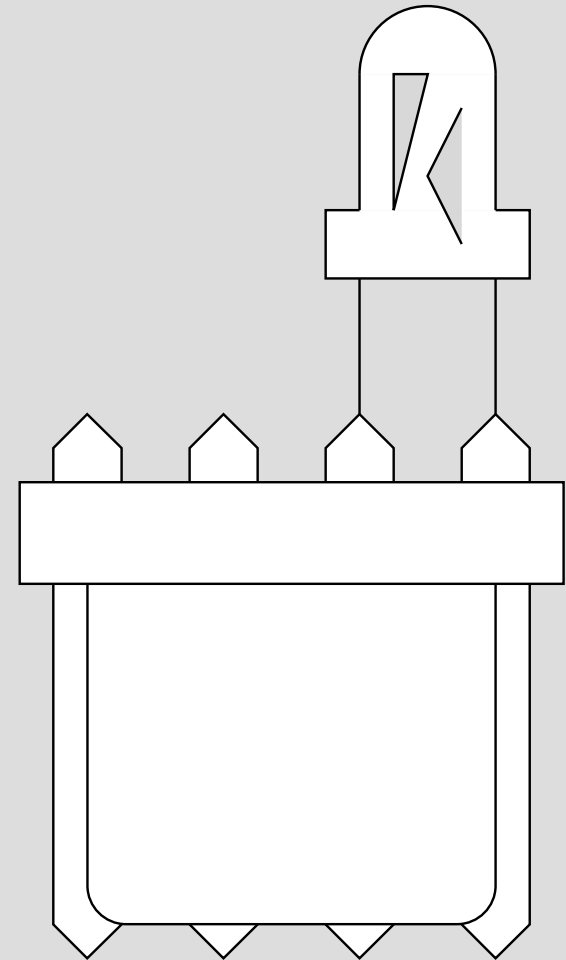
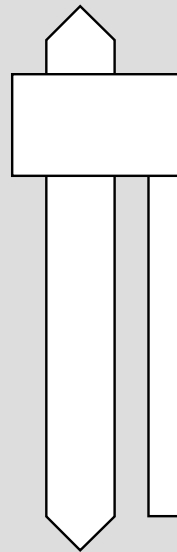
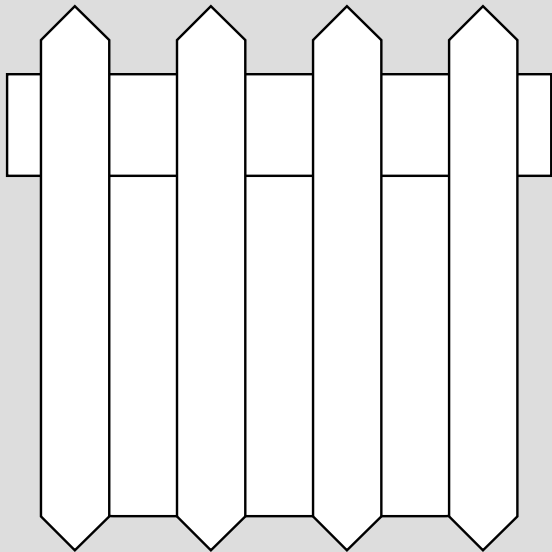


- Hack of the Buzzer



# I/O Ports

- IR Cartridge
- Replacement Adapters
  - Apple ][e Joystick
  - HP to HP
  - RS232



# HP28 Printer

- HP82240B



# Ideas!

- Add a Joystick
- Telephone Composer
  - Photocoupler and Relays
  - Provides click of a rotary phone.
  - Dial by counting digits, then waiting.
- Motor Control
- Robot
- Plotter

# Part 3: Appendices

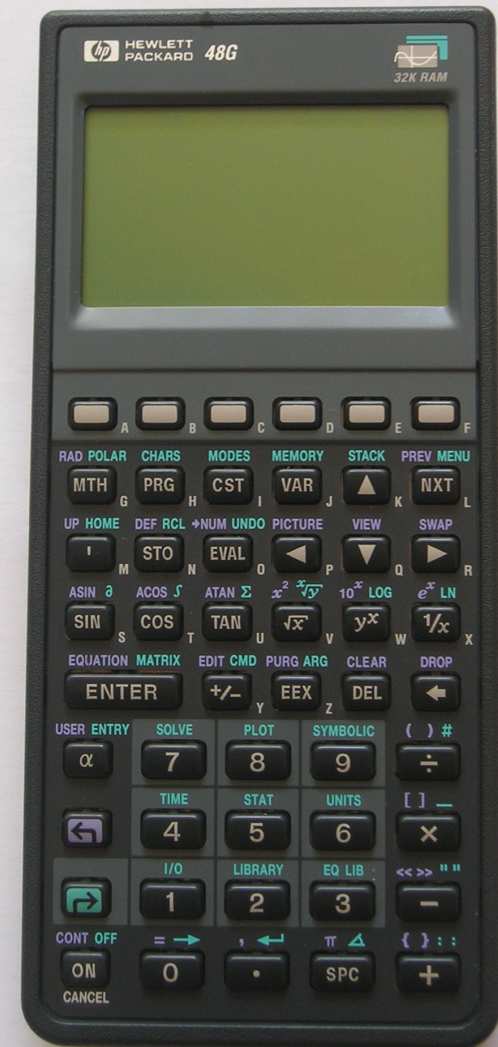
- Machine Language
- SATURN Microprocessor
- SATURN Instructions
- Exhaustive List of Error Codes
- List of Objects and RPL Entry Points
- Library of Programs

# Epilogue



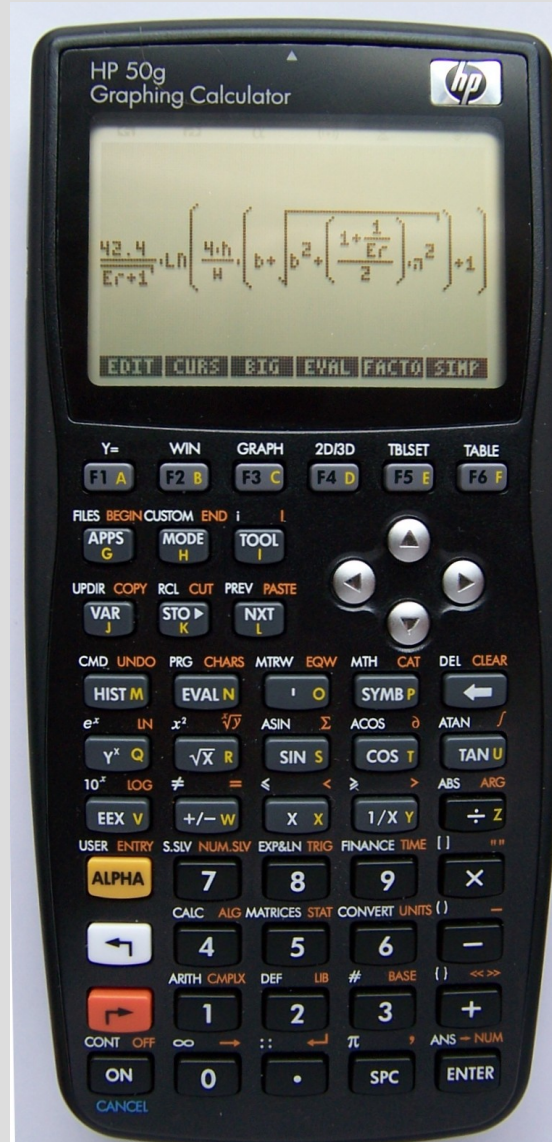
# HP48 Series

- Memory Expansion
- IR I/O
- Serial Port
- Meta Kernel
  - Replacement OS

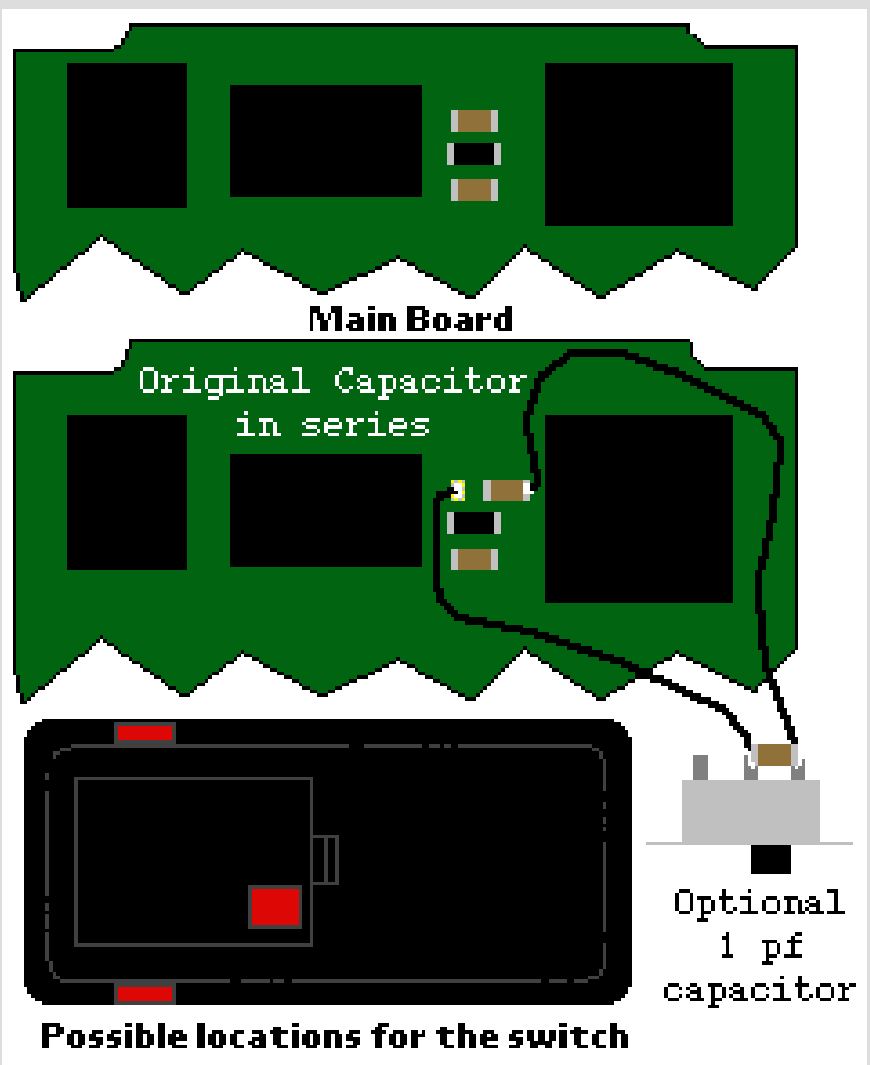


# HP49 Series

- SD Card Slot
- USB, IR, Serial
- ARM CPU
  - Emulates SATURN
- HPGCC
- Meta Kernel
  - Official OS

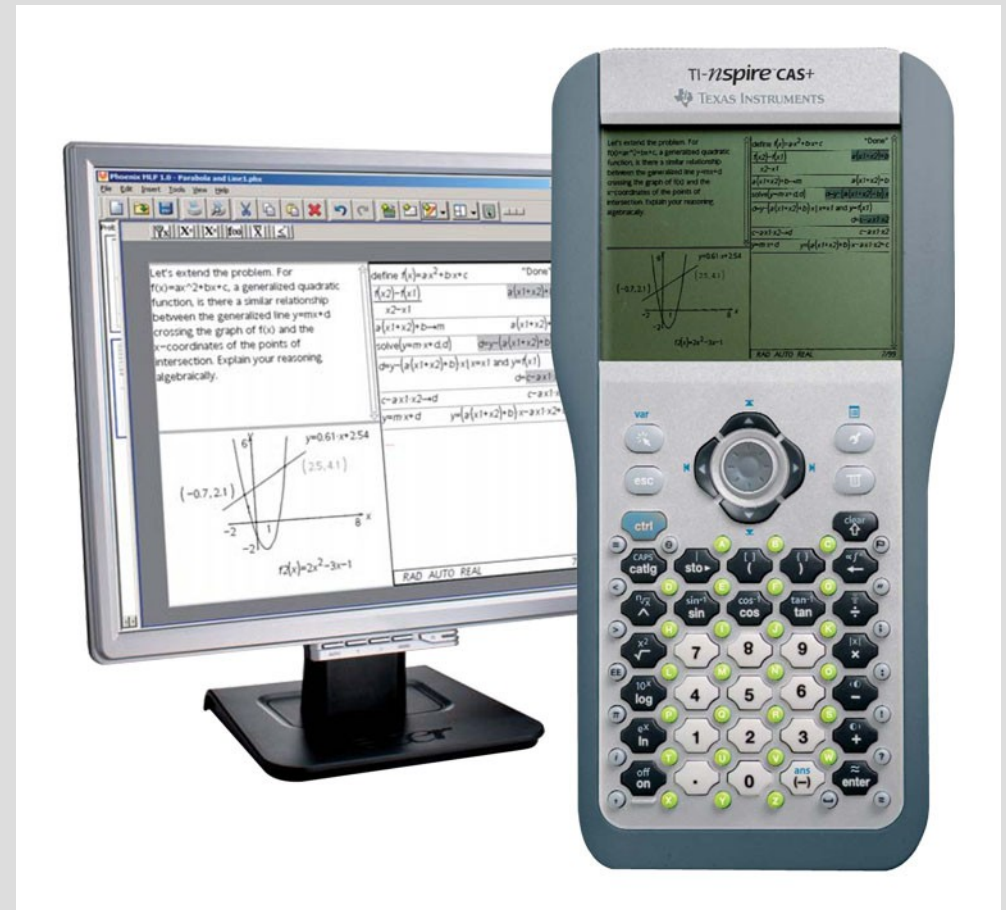


# TI Calculators



# Next Step: TI Nspire

- No ML Programming
- No Games
- No ROM Dump



# Questions?

Travis Goodspeed  
travis@utk.edu  
865.300.0094