

# DOG-1

*Danny's Obtuse Gadget, version 1.0*

(I'm writing this as if the thing was finished - far from it! - So this is all provisional design)

A very limited computer based on an Arduino Uno and a TM1638 I/O card. Runs as a virtual machine on top of the Arduino. Inspired by the single-board computers of the mid-1970's notably the [MK14](#) and [KIM-1](#). The architecture and instruction set is loosely derived from the processors used on these machines, the [SC/MP](#) and [6502](#) respectively.

Some novelty is offered by the hardware used is that it should be possible to interface with the built-in I/O of the Arduino. So eg. a digital thermometer could be made by attaching the appropriate sensor and programming the DOG-1 directly, without access to a PC.

## Basic Architecture

- 16-bit addressing
- 8-bit instructions
- 8-bit data

### Registers

Addressing will be handled primarily by a 16-bit Program Counter (PC).

Operations will be carried out using 8-bit registers:

- Accumulator - primary
- X and Y - secondary
- Status Register (SR) - system flags
- Stack Pointer (SP) - for remembering the origin the source of subroutine jumps

### Flags

Bit	Flag	Name	Description
0	N	Negative	Set if bit 7 of ACC is set
1	V	Overflow	-
2	Z	Zero	-
3	C	Carry	-

### Addressing Modes

nn = 2 hex digits nnnn = 4 hex digits

Mode	Assembler Format	Description
Immediate	#nn	Value is given immediately after opcode
Absolute	nnnn	Value is contained in the given address
Indirect Absolute	(nnnn)	-
Absolute Indexed, X	nnnn, X	-
Absolute Indexed, Y	nnnn, Y	-
Relative	nnnn	-

## I/O

**TM1638 Card**

	0	1	2	3	4	5	6	7
LEDs	*	*	*	*	*	*	*	*
7-Segs	8	8	8	8	8	8	8	8
Buttons	0	0	0	0	0	0	0	0

The LEDs will usually display the contents of the Status Register.

Generally, the 7-Segment displays 0-3 will display the current value (address) of the Program Counter. Displays 6 and 7 showing the contents at that address. All in hexadecimal.

The functionality of the I/O will depend on the system's mode : **Program** or **Run**. Display 4 shows the current mode, P or R. Display 5 is unused (TBD).

Push-button 4 switches between these modes. At any time, pressing buttons 4 and 5 **together** will reset the PC to 0000.

### Program Mode

Pressing the buttons 0-3, 6-7 will increment the value corresponding to that of the display above it. Programming is achieved by pressing button 3 to increment the PC (with overflow occurring, counting up on displays 0-2). Pressing button 7 will increment the value on display 7 (overflowing to display 6) providing the value at the given address.

Pressing button 4 again will switch to *Run* mode.

### Run Mode

Initially the system will be halted at the current address. Pressing button 3 will single-step through the program (pressing buttons 0-3 will cause the PC to skip to the corresponding address [running or skipping code in between? TBD]).

Alternately the program may be run in real time by pressing button 5. Pressing this button again will halt the program.

## Instruction Set

*note to self* - things like LDA will have a version for each of the addressing modes, ~ 6, so it's probably an idea to hop 8 values between base versions

Instruction	OpCode	Size	Operation	Description
NOP	00	1	PC++	No operation
LDA #nn	10	3	ACC <- nn	Load accumulator, immediate
HLT	FF	1	Halt	Stops program flow