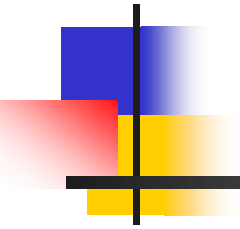
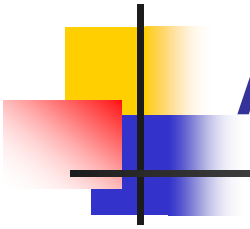


Harvard to Princeton at Stanford

Silicon Valley Forth Interest Group

November 19, 2011

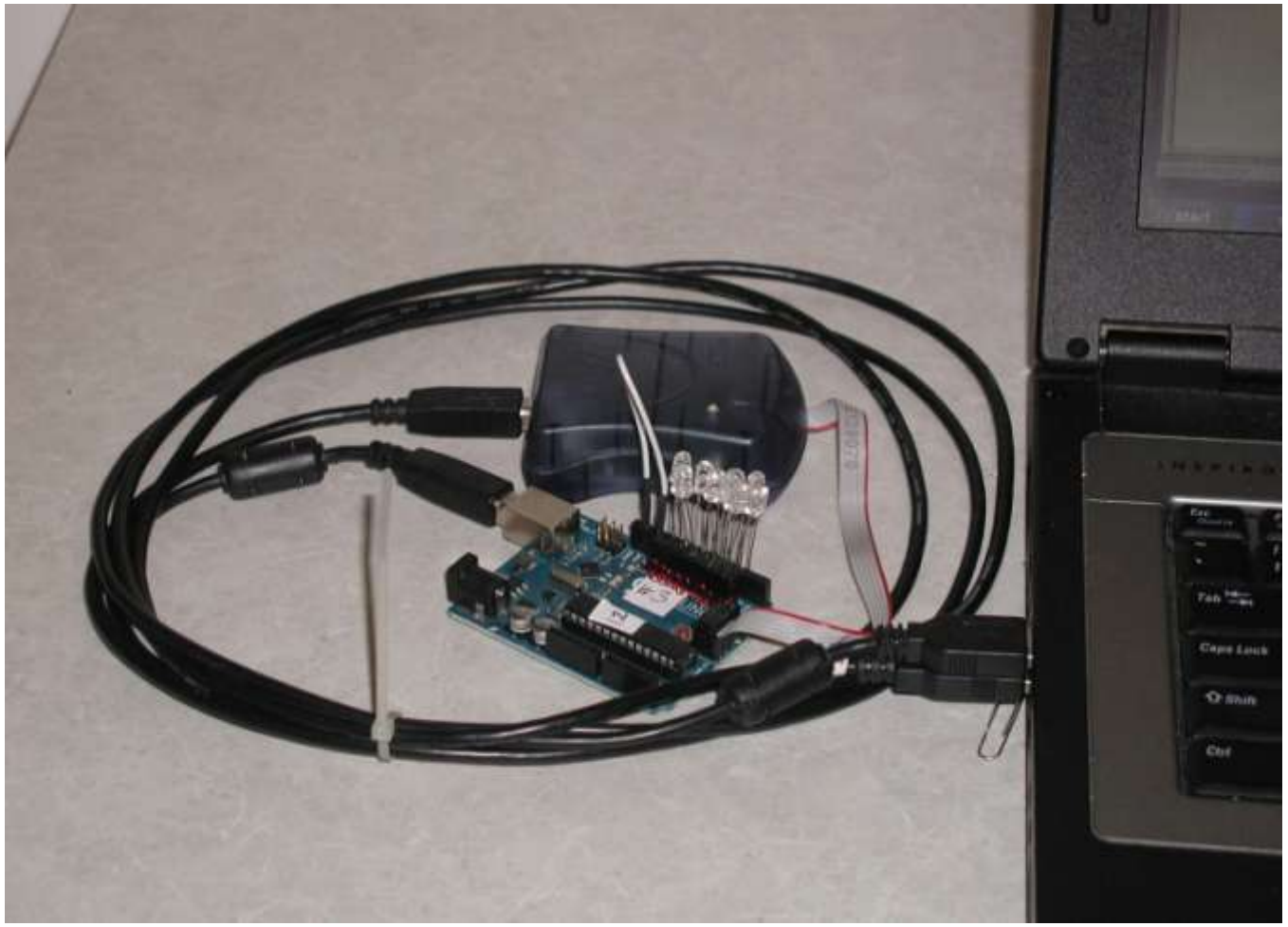
Chen-Hanson Ting



A Simple FORTH for Arduino

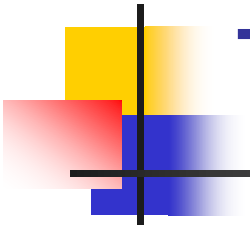
- A simple Forth system which can be loaded as a Arduino Sketch.
- Must be written in C (Arduino Process) and does not require separated AVR programmer.
- Attractive to beginning Arduino users.

Arduino Uno with AVRISP mkII

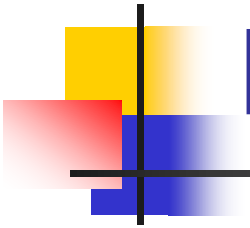


Arduino Uno Alone

The Fundamental Problem

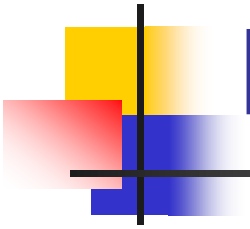


- ATmega328P is a microcontroller with Harvard Architecture. Program memory space and data memory space are separated.
- C is a programming language with Harvard Architecture. Program space is hidden. Data space must be declared.
- Forth is a programming language with Princeton Architecture, with an unified program and data memory space.



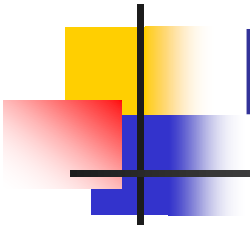
Forth Virtual Machine

- 33 Pseudo instructions as byte codes.
- Pseudo instructions are stored either in program or data memory.
- A Finite State Machine to execute pseudo instructions.



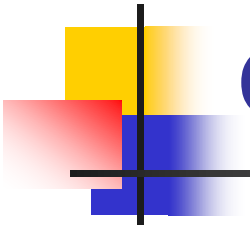
Forth Virtual Machine

- A unified memory model spanning both program and data memories.
- Program memory and data memory are mapped to different areas in the memory space
- Forth dictionary is initially in program memory, but can be extended into data memory.



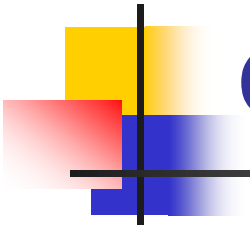
Forth Commands

- Forth commands are records linked into a dictionary.
- A command record has a link field, a name field and a code field.
- A primitive command has pseudo instructions in code field.
- A compound command has a token list in code field.



ceForth_328 Sketch

- Pseudo instructions coded in C.
- Finite State Machine coded in C.
- A dictionary imported as a data array.
- The dictionary is produced by cefMETA328 metacompiler.



ceForth_328 Memory Space

- 0000-00FF ATmega328 registers
- 0100-02FF Data space assigned by C
- 0300-031F Forth variables
- 0320-087F Free space for dictionary
- 0880-08FF TIB/ATmega328 stack
- 0900-1FFF Dictionary in flash memory

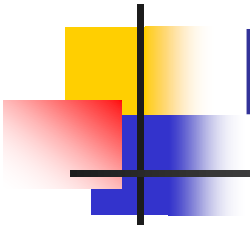
Finite State Machine

```
void setup()
{       clock = 0; phase = 0; P = 0; IP = 0;
        S = stack; R = rack; top = 0;}
void loop()
{       phase = clock & 3;
          switch(phase) {
                case 0: fetch_decode(); break;
                case 1: execute(I1); break;
                case 2: execute(I2); break;
                case 3: jump(); break;          }
          clock += 1;
}
```

cefMETA328 Metacompiler

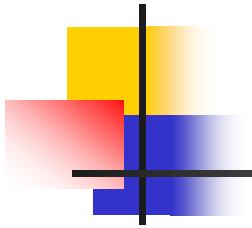
All files compiled by cefMETA328.fex under F#

- cefMETA328.f Metacompiler
- cefASM328.f Assembler
- cefKERN328.f Kernel commands
- cEF328.f Compound commands
- cefSIM328.f Simulator

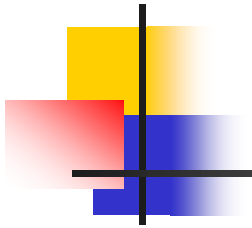


Demo

- Run ceForth328.pde under Arduino 0022 system.
- HyperTerminal interaction
- Compile applications:
  - Blink
  - Tone
  - Servo motors
  - Traffic controller



Questions?



Thank You.