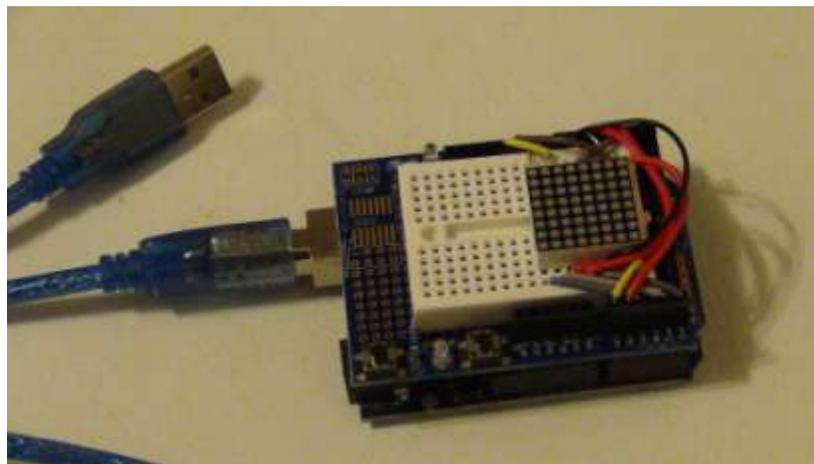


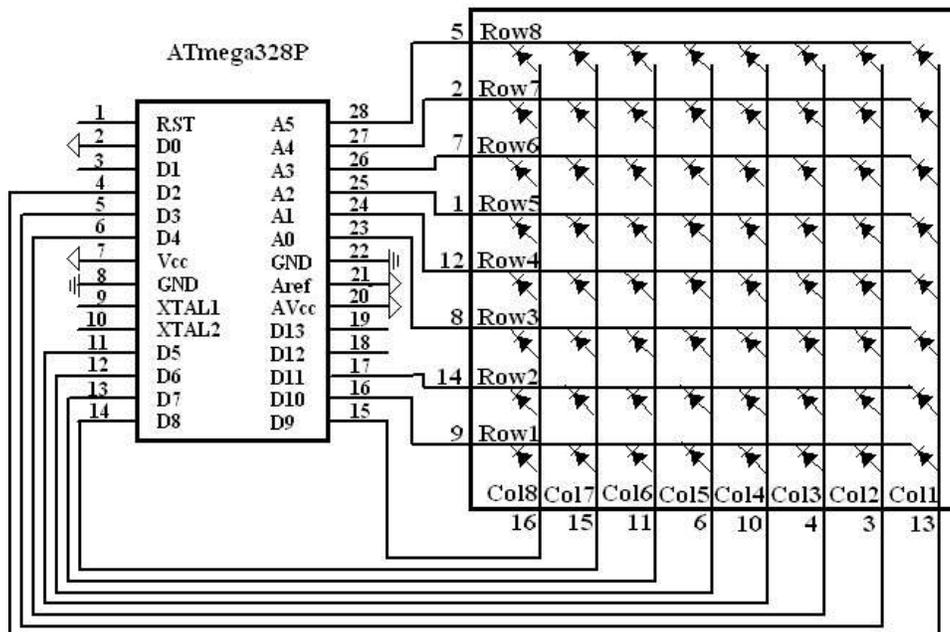
# 2015 Forth Day Programming Challenge

Chen-Hanson Ting

The goal of this programming challenge is to generate the best images on an 8x8 LED array. The LED array is from Jameco, ValuePro LD788BS. It is mounted on a shield that is driven by an Arduino Uno or compatible board. I built four sets of this array for Programming Challenge contestants. However, you are welcome to use your own Arduino board with which you may be more familiar. On these Arduino boards, I have loaded 328eforth in the ATmega328P chip. You can use it to write your Forth code. However, Forth is not required in the contest - you can use C or another language of your preference,



8x8 Dot Matrix Display  
Jameco/ValuePro LD788BS



## Programming in 328eForth

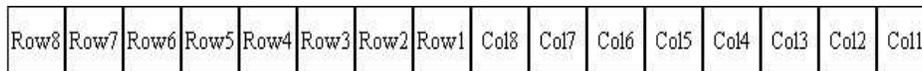
When you plug the Arduino Uno board into a USB port into your PC the first time, the PC will ask you to install an USB terminal driver, which is available in the `..\arduino-1.0\drivers` folder. Use the Hardware Device Manager to ensure the drivers are properly installed and note the COM port number. You may have to first uninstall the existing COM port first, scan new devices, and reinstall the driver. If the drivers work, you can open the HyperTerminal, select the proper COM port and set the communication protocols as: 19200 baud, 8 data bits, 1 stop bit, and no flow control.

In the ASCII Setup Panel, set the Line Delay to 300 ms, to allow Forth to compile lines of text you will download to the ATmega328A chip.

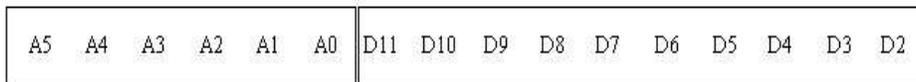
If everything is set up correctly, 328eForth will boot up and show the sign-on message:  
328eForth v2.20

## Mapping Data Bits to Columns and Rows of LED Array

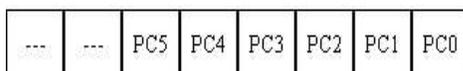
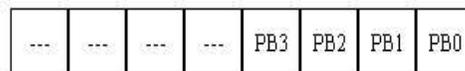
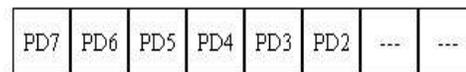
The following figure shows the mapping of data bits in the ATmega328A IO registers to the IO ports on the Arduino Uno Board, to the drive lines and sink lines of the LED arrays.



Data Bits in DRIVE to drive the LED Array



Data Bits on the Arduino Uno Board



Data Bits in the ATmega328P I/O Registers

## Core of Source Code

The following source code has been loaded on the top of 328eForth, and you can use them to construct your images. They could also serve as reference code if you chose to program in other languages.

```
hex
variable ticks
: init-ports
  f 24 c! 3f 27 c! fc 2a c!      \ output ports
  ;
: ms ( n -- ) for aft 100 for next then next ;
: delay ticks @ ms
  ?key abort" done" ;
: rshift ( n n -- )
  for aft 2/ then next ;
: drive ( n -- )
  0 25 c! 0 28 c! 0 2b c!
  ff00 xor
  dup 2* 2* 2b c!              \ PD outputs
  6 rshift dup f and 25 c!     \ PB outputs
  4 rshift 3f and 28 c!       \ PC outputs
  delay ;
```

TICKS determine the lengths of LED on-time in the command DELAY.

MS delays that many milliseconds.

DELAY delays TICKS milliseconds. It monitors the UART input line with ?KEY. It aborts when the user types a key on the keyboard, to break the display infinite loop.

INIT-PORTS initializes all the output pins used in the three IO ports.

RSHIFT aligns a 16 bit word so that its lowest 8 bits can be written to an 8 bit output data register.

DRIVE takes a 16 bit words and drives 8 columns of the LED array with the lower 8 bits. It inverts the upper 8 bits and uses them to sink 8 rows of the LED array.