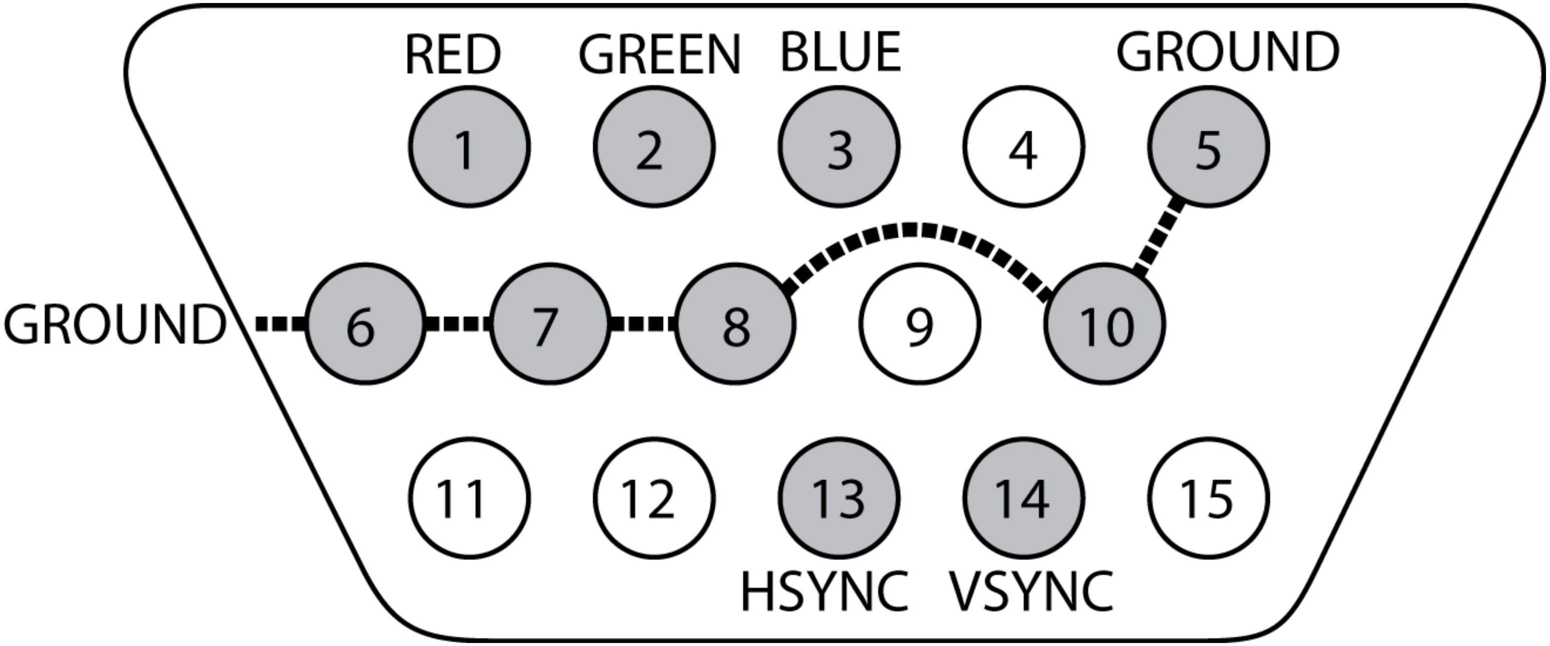
Workshop generating VGA signals

Piksel'09, Bergen

Arjan Scherpenisse

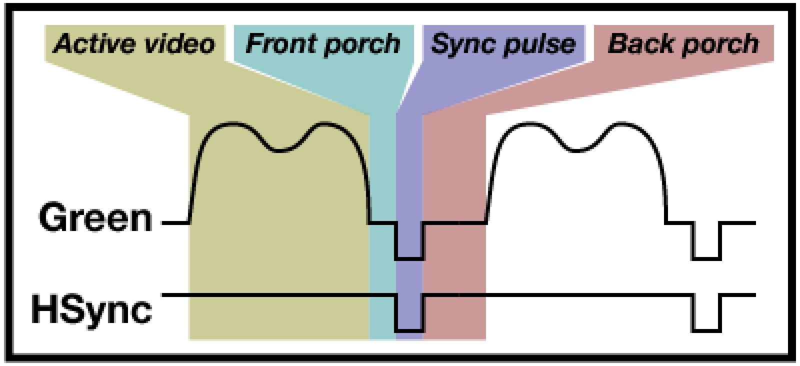
arjan@scherpenisse.net
http://scherpenisse.net

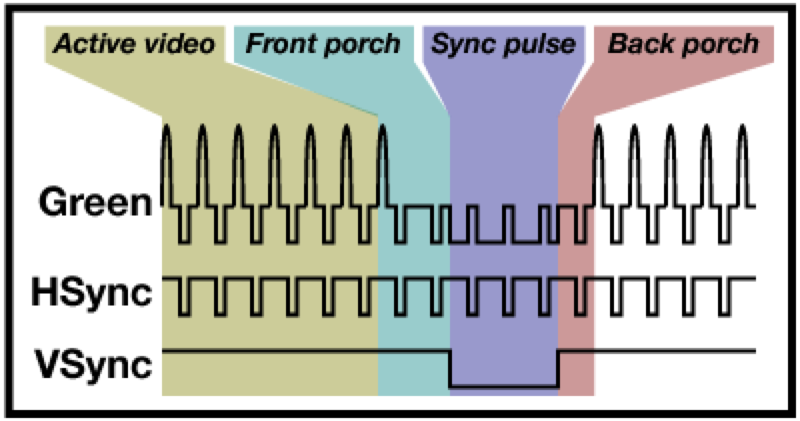
VGA connector



VGA – Theory of operation

- Data is sent over R, G and B pins.
- Synchronization is done horizontally and vertically

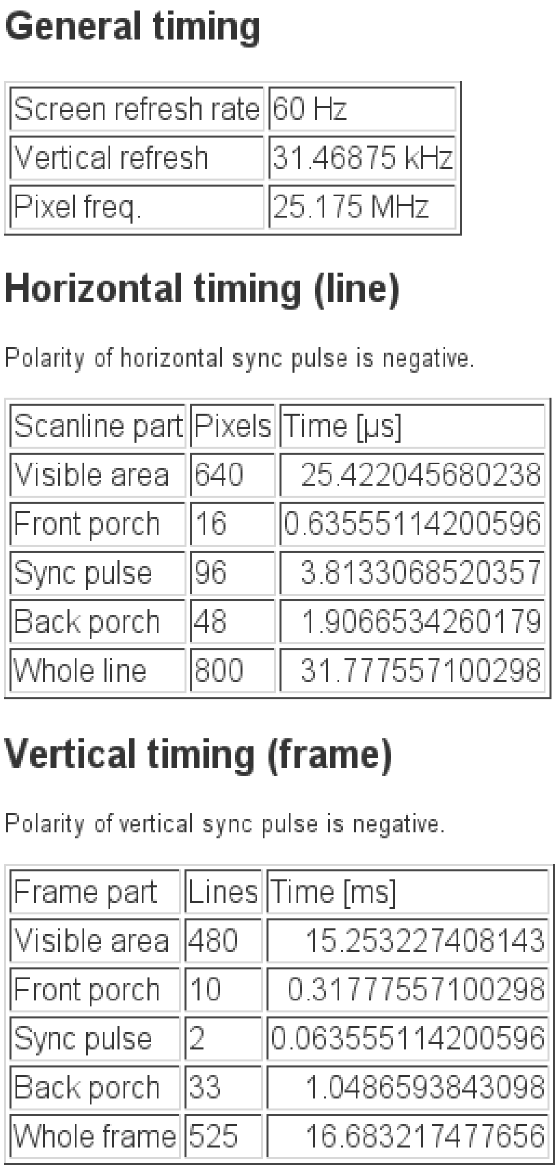




http://web.mit.edu/6.111/www/s2004/NEWKIT/vga.shtml

VGA – Theory of operation

- Timing is of the essence!



General timing

Screen refresh rate	60 Hz
Vertical refresh	31.46875 kHz
Pixel freq.	25.175 MHz

Horizontal timing (line)

Polarity of horizontal sync pulse is negative.

Scanline part	Pixels	Time [µs]
Visible area	640	25.422045680238
Front porch	16	0.63555114200596
Sync pulse	96	3.8133068520357
Back porch	48	1.9066534260179
Whole line	800	31.777557100298

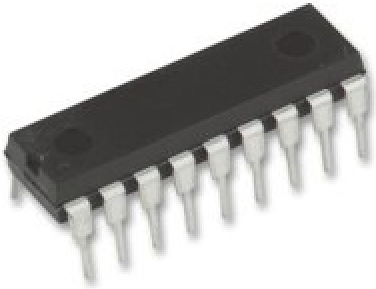
Vertical timing (frame)

Polarity of vertical sync pulse is negative.

Frame part	Lines	Time [ms]
Visible area	480	15.253227408143
Front porch	10	0.31777557100298
Sync pulse	2	0.063555114200596
Back porch	33	1.0486593843098
Whole frame	525	16.683217477656

http://tinyvga.com/vga-timing/640x480@60Hz

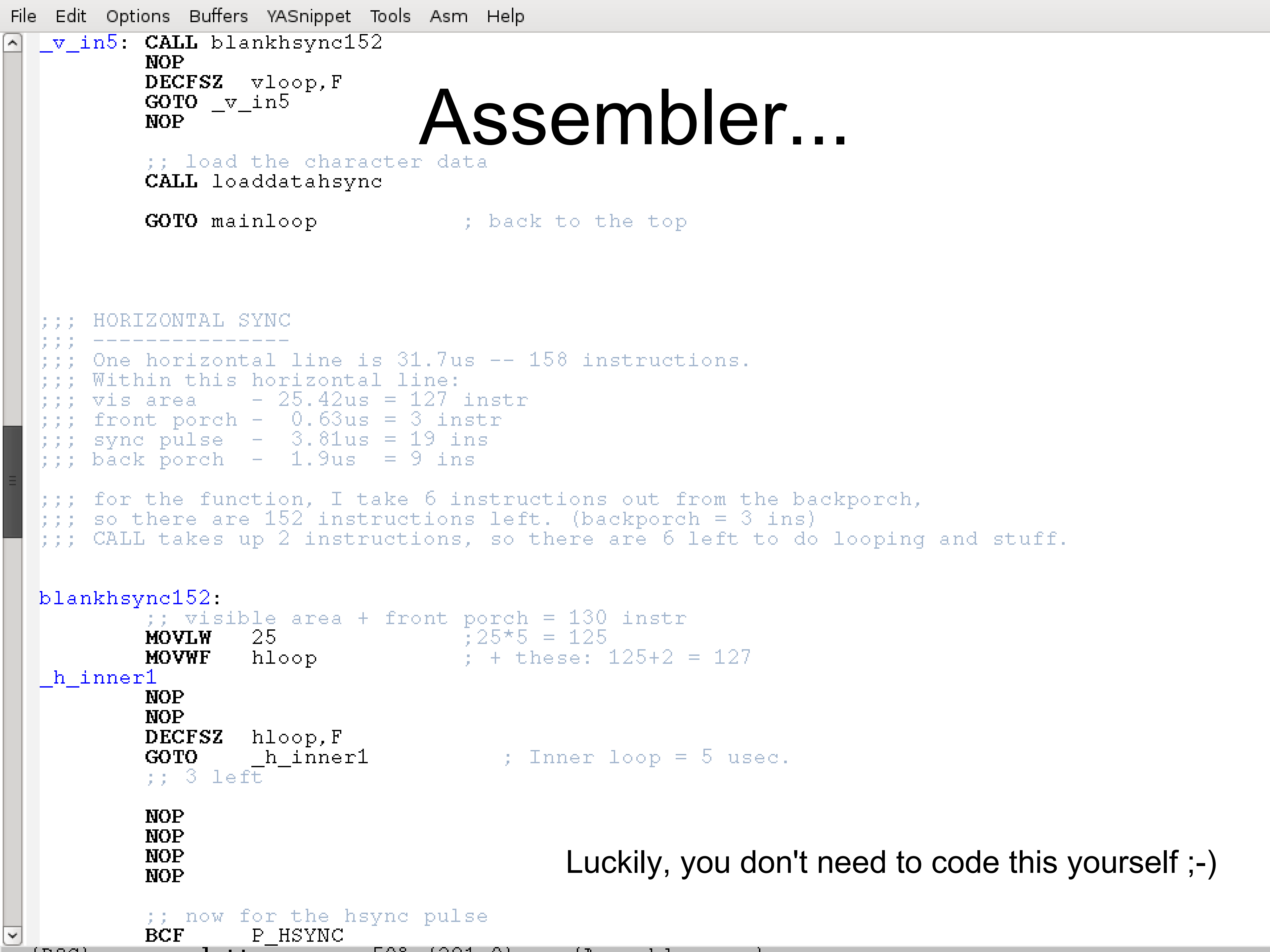
Hello, microcontroller!



Microcontrollers

- Most famous example: Arduino
  - We just need a much simpler one.
- I chose PIC16F62x, due to low costs
- More complex PICs / atmels allow for higher resolution, more data, interactivity, …
- Need to program the chip in assembler to get the timing right.

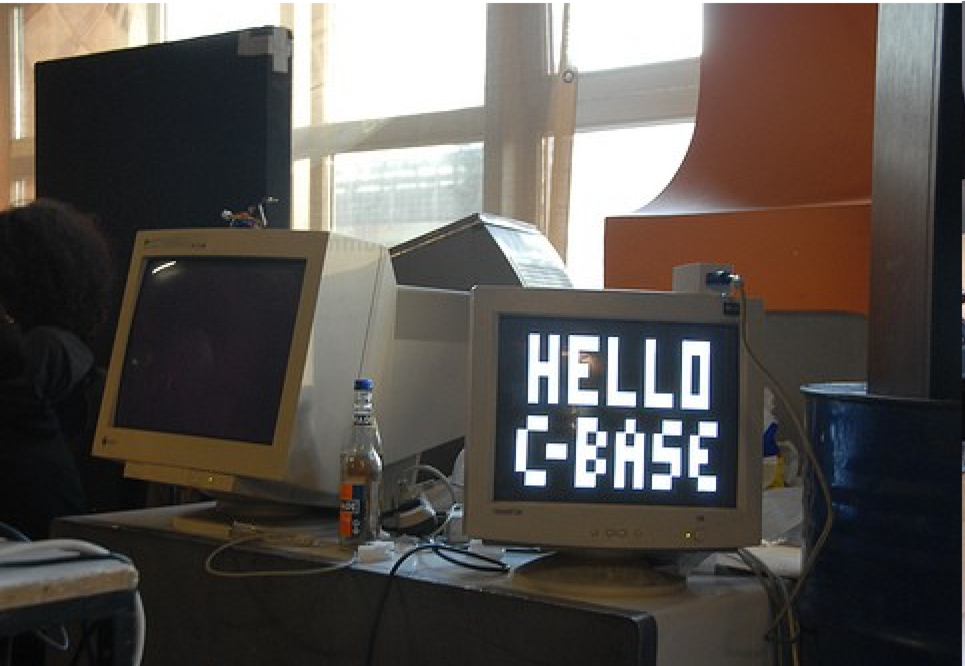
Assembler...

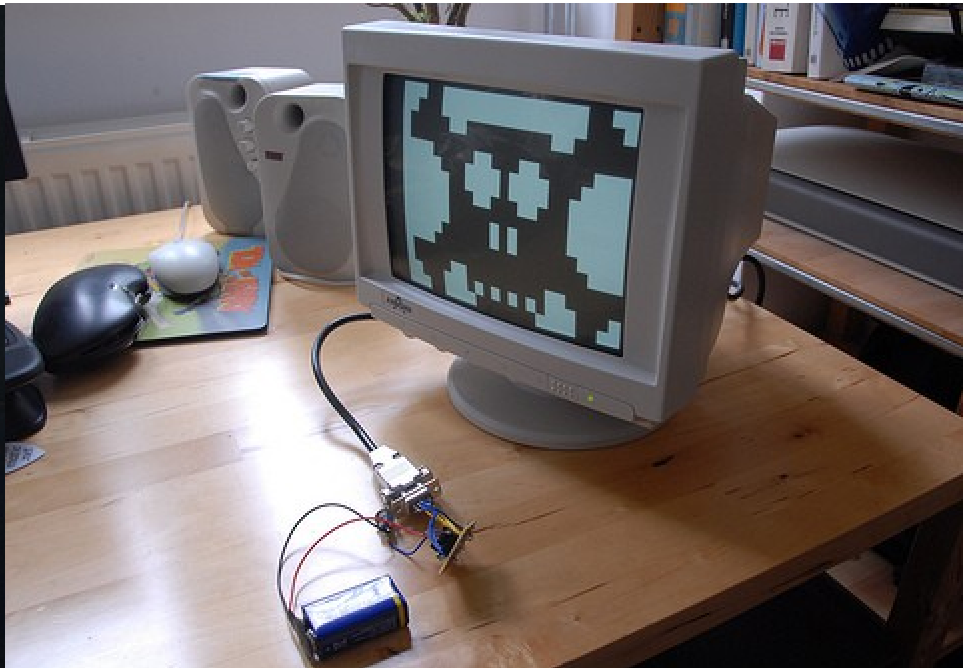


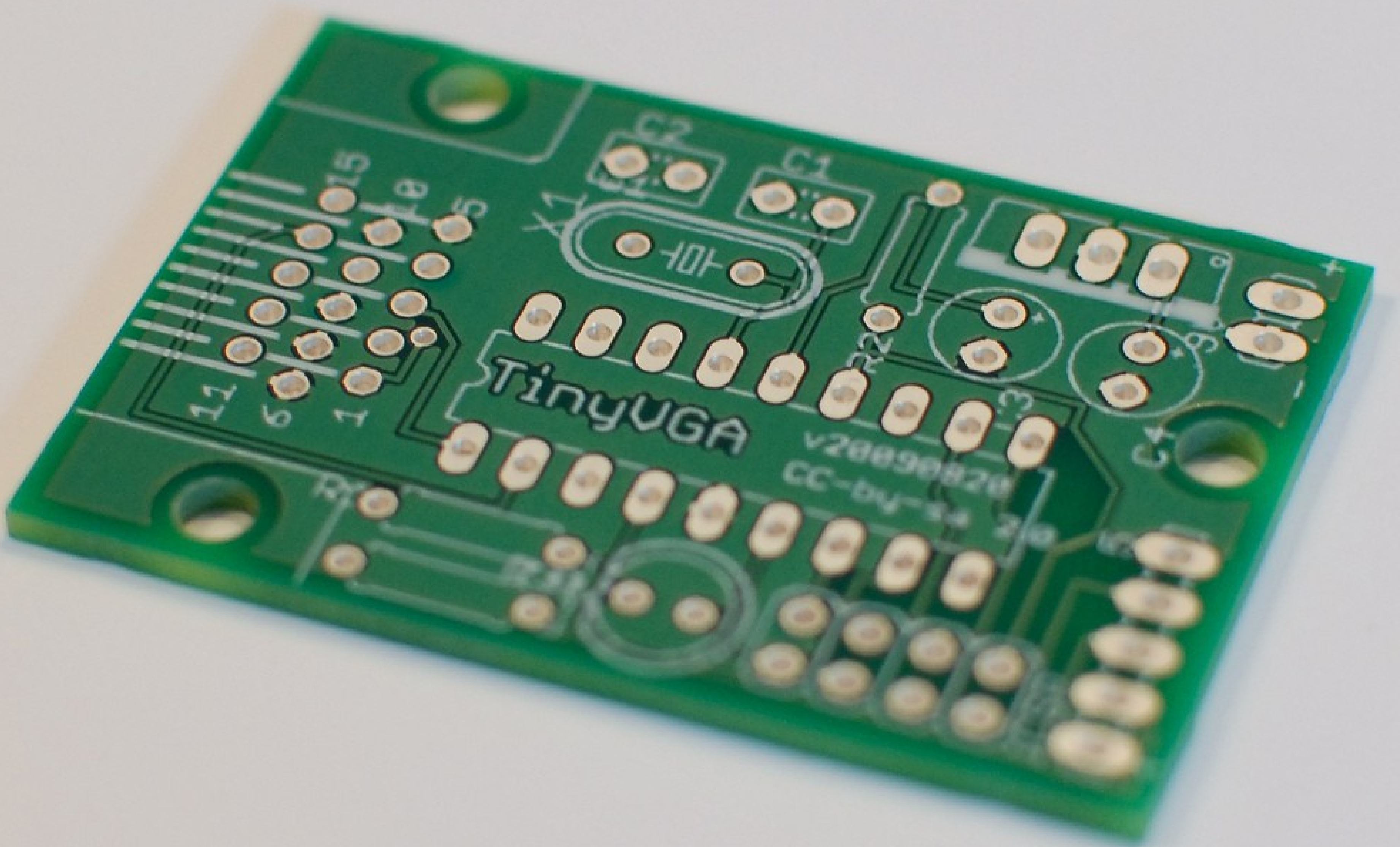
```
_v_in5: CALL blankhsync152
        NOP
        DECFSZ  vloop,F
        GOTO _v_in5
        NOP

        ;; load the character data
        CALL loaddatahsync

        GOTO mainloop           ; back to the top



;;; HORIZONTAL SYNC
;;; ---------------
;;; One horizontal line is 31.7us -- 158 instructions.
;;; Within this horizontal line:
;;; vis area    - 25.42us = 127 instr
;;; front porch -  0.63us = 3 instr
;;; sync pulse  -  3.81us = 19 ins
;;; back porch  -  1.9us  = 9 ins

;;; for the function, I take 6 instructions out from the backporch,
;;; so there are 152 instructions left. (backporch = 3 ins)
;;; CALL takes up 2 instructions, so there are 6 left to do looping and stuff.


blankhsync152:
        ;; visible area + front porch = 130 instr
        MOVLW   25              ;25*5 = 125
        MOVWF   hloop           ; + these: 125+2 = 127
_h_inner1
        NOP
        NOP
        DECFSZ  hloop,F
        GOTO    _h_inner1          ; Inner loop = 5 usec.
        ;; 3 left

        NOP
        NOP
        NOP
        NOP

        ;; now for the hsync pulse
        BCF     P_HSYNC
```

Luckily, you don't need to code this yourself ;-)

What we're gonna do






C2
C1
X1
15
10
5
11
6
1
TinyVGA
v20090820
CC-by-sa 2.0
R2
C3
C4
R1
R3

Code

- To “burn” firmware you'll need a PIC programmer.
- Currently, 3 different firmwares for the TinyVGA board:
  - Color Image 11x8 pixels
  - Mono image 32x20 pixels
  - Externally controllable “giant pixel”
- I'll put your image/logo on your chip for you.
- All code is available for download:
  - http://scherpenisse.net/tinyvga