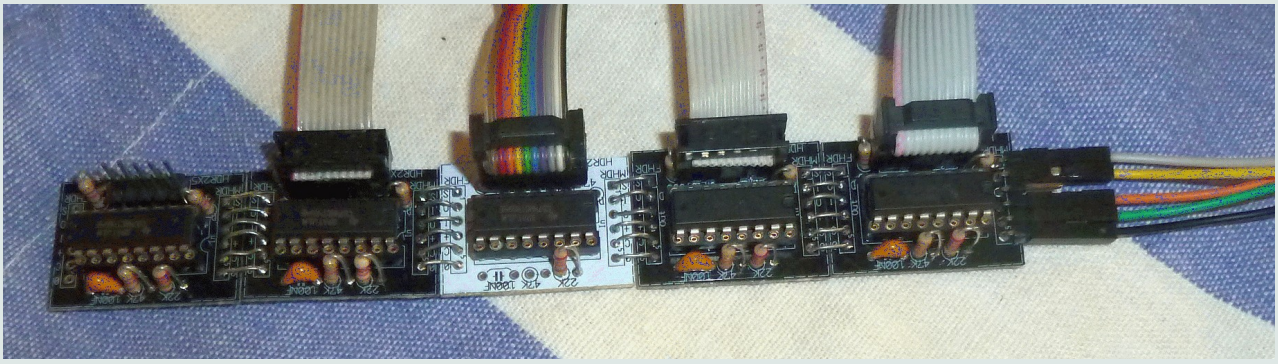
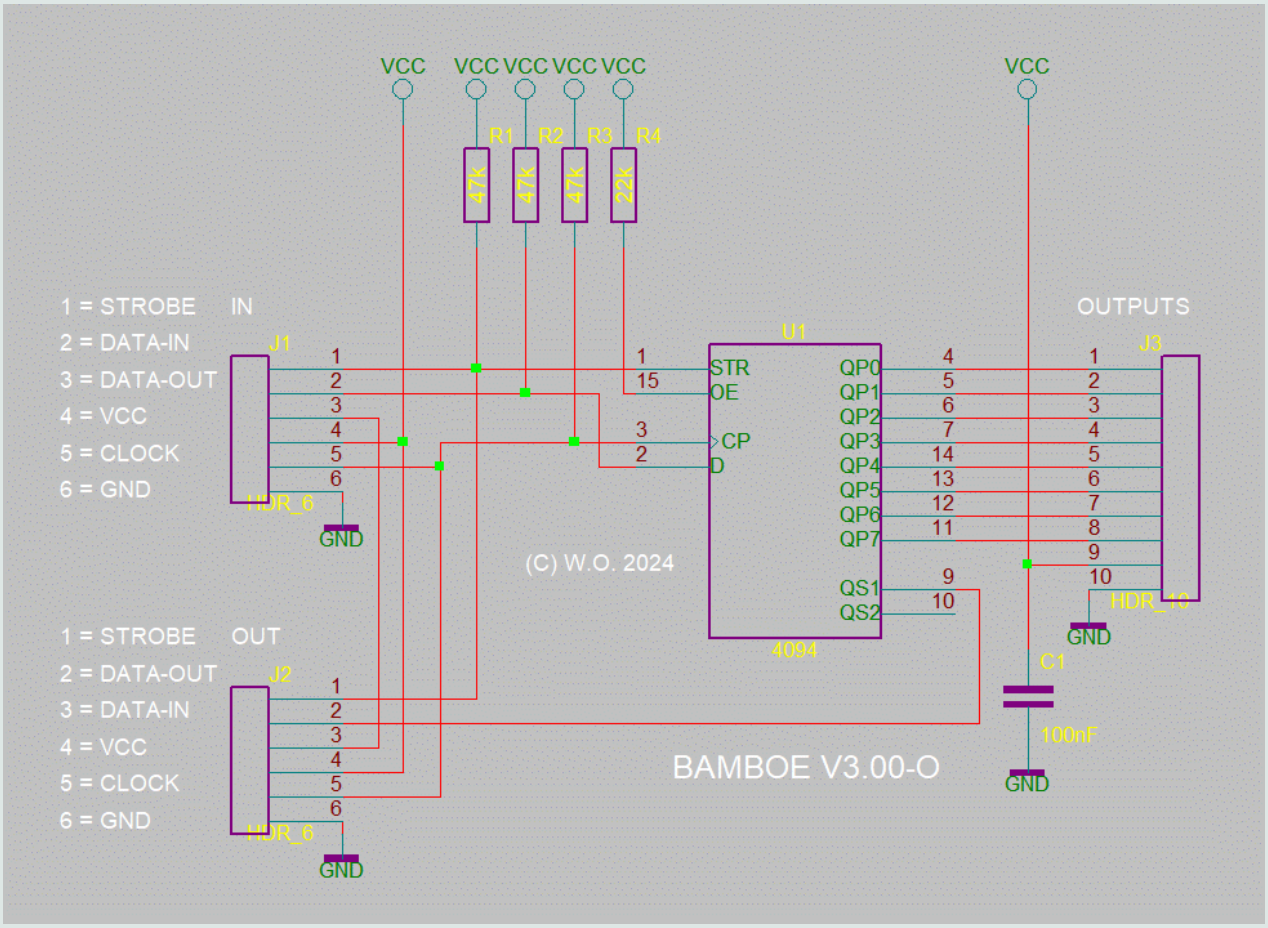
Bamboo-3.00o

Bamboo version-3 is an 8-bit parallel output block.
The data is pushed in serially, when all the data is left it is copied to the outputs.
Bamboo is expandable to any length.

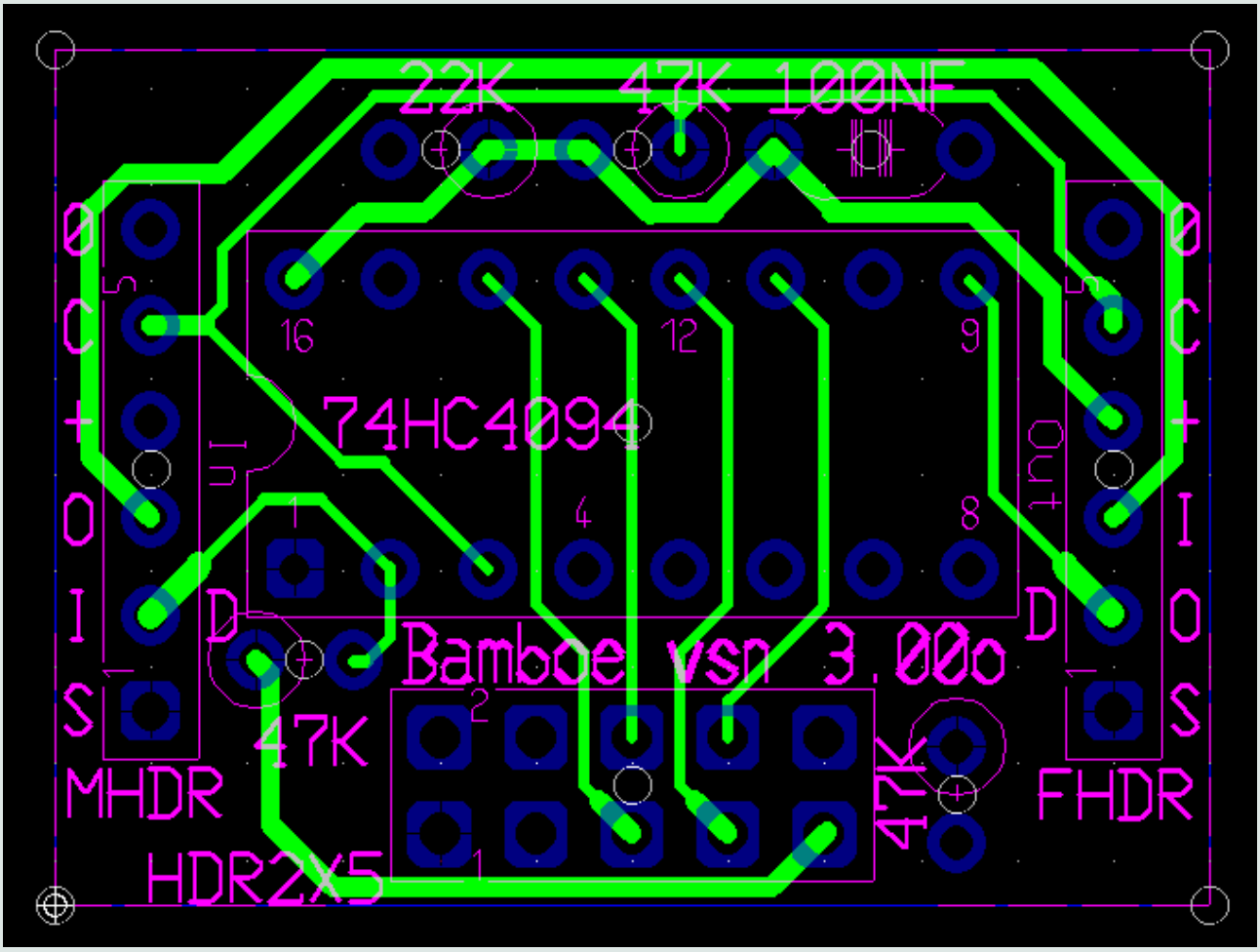
The 40-bits bamboo as used in the Tingel-Tangel:



Bamboo block schematics:



Bamboo PCB example:



```
\ Bamboo code example for noForth t on the RP2040
hex
D0000020    constant GPIO-OE        \ GPIO output enable
D0000010    constant GPIO-OUT       \ GPIO output value
6 bitmask   constant STR            \ Bamboe I/O bits
7 bitmask   constant OUT
8 bitmask   constant CLK

: INIT      ( -- )
    5A dm 6 pads!                  \ Enable output on pin 6
    5A dm 7 pads!                  \ Enable output on pin 7
    5A dm 8 pads!                  \ Enable output on pin 8
    [ str out clk or or ] literal
    gpio-oe **bis ;   init

: READY     ( -- )      str gpio-out **bis  1 us  str gpio-out **bic ;

: >BAMBOO   ( b -- )       \ Output data for one bamboo block
    dm 24 lshift               \ Data to high byte
    8 for                      \ 8 bits
        dup 0< if              \ Highest bit set?
            out gpio-out **bis  \ Data high
        else
            out gpio-out **bic  \ data low
        then
        clk gpio-out **bic      \ Clock low
        clk gpio-out **bis      \ Clock high
        2*                      \ Next bit
    next  drop ;

dm 05 constant #B     \ Number of used bamboo blocks
create LATCH  #B 8 * allot  align
: >BAMBOOS  ( -- )
    #B for
        latch i + c@ >bamboo
    next  ready ;

: LOC       ( +n a1 -- bit a2 ) \ Bit location in byte-address a2
    over 3 rshift +  >r         \ Convert to byte addresses
    07 and bitmask  r> ;        \ Convert low 3-bits to bit mask

: ZERO      ( -- )          latch #B 0 fill ;
: SET       ( +n a -- )     loc *bis ;
: CLR       ( +n a -- )     loc *bic ;
: >BB       ( +n -- )       zero  latch set  >bamboos ;

: RUNNER        ( -- )     \ Running light over all bamboo outputs used
    init  0  begin
        dup >bb  1+  20 ms
        dup dm 40 = if  dup -  then ( Hard coded end! )
    key? until  drop ;
```