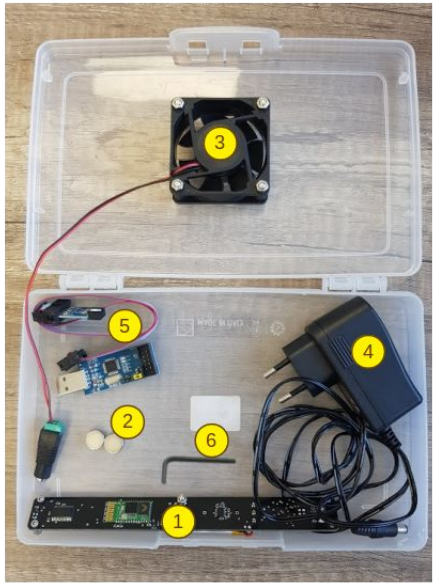
Embedded Systems

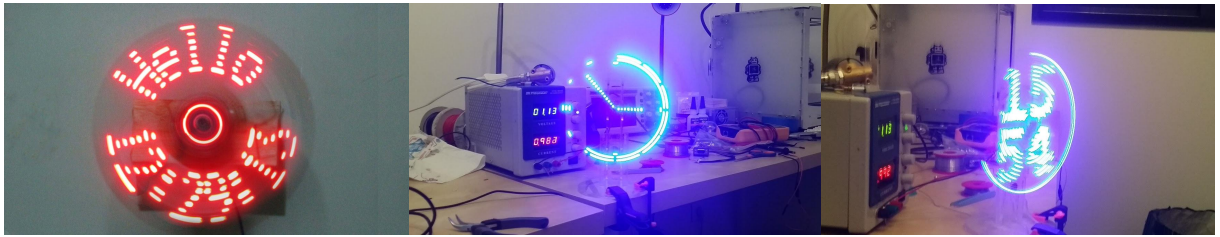
Clock project

Jean-Baptiste Cognet
Marc Duclusaud
Amine Karbab
Marion Othéguy

Project's presentation



Goals :



Main loop :

$Resolution = 240$

$global\_time \leftarrow 0$
$cycle\_duration \leftarrow 2500$

while 1 **do**
    $leds \leftarrow compute\_leds(global\_time)$
    $pos \leftarrow \frac{Resolution \times TCNT1}{cycle\_duration}$
    $display\_leds()$
end while

$//Hall\ sensor\ interrupt$
if $HALL$ **then**
    $cycle\_duration \leftarrow TCNT1$
    $TCNT1 \leftarrow 0$
    $global\_time \leftarrow global\_time + cycle\_duration$
end if

Needle Clock

Hours / Seconds / Minutes: computed from *global_time*

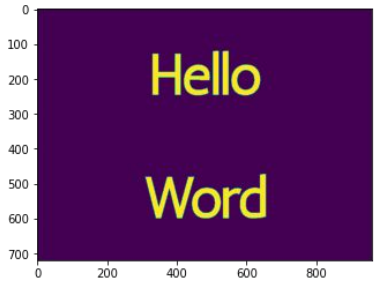
Display:

- each second
- seconds: extern border of the clock → change of the last bit of uint16 in *leds*
- minutes: update of the position of the big needle (0b0011111111111111) in *leds*
- hours: update of the position of the litlle needle (0b0000000011111111) in *leds*

Rounded digits

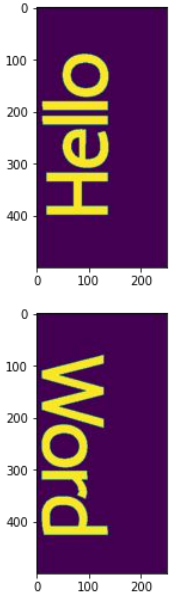
image → 16 bits x clock resolution

→ image processing to generate array of strings light



Binary image

Slicing & rotation

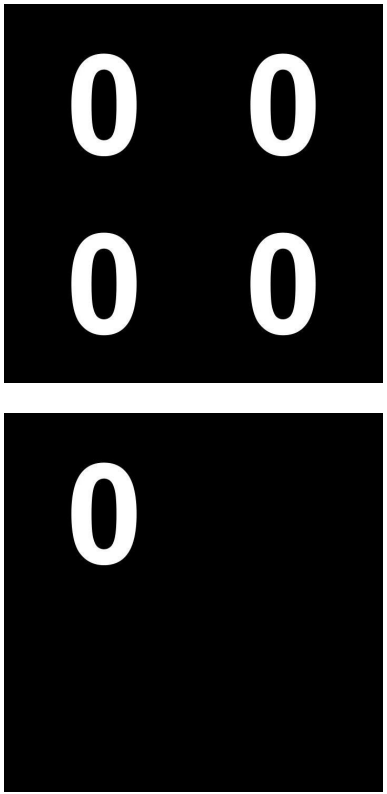


Mean

{..., 0b0000100001000000, 0b0000100001100000,
0b0000100001100000, 0b0000100001100000,
0b0000100001100000, 0b0000100001100000,
0b0000100001100000, 0b0000100001000000,
0b0000110001000000, 0b0000000000000000,
0b0000000011000000,...};

uint16_t leds

Straight digits



Cartesian coordinates → polar coordinates

1 image = 16 x *Resolution* / 8

For a resolution of 240 : 1 image = 480 octets

29 images :

0; 1; 2	0; 1; 2; 3; 4; 5; 6; 7; 8; 9
0; 1; 2; 3; 4; 5	0; 1; 2; 3; 4; 5; 6; 7; 8; 9

Benchmark

Revolution duration: estimated at 50ms

Minimal display frequency: Resolution x number of displays x rotation speed

240 x 2 x 20 = ~10kHz

Timer1 (16 bits) prescaled at 256, adds cycle duration to time every cycle duration calculation (ie Hall interruption trigger) at 50781Hz.

global_time being uint32_t => Overflows at 23.49 Hours.

An estimation of the slowest revolution 100ms=> doesn’t overflow the timer