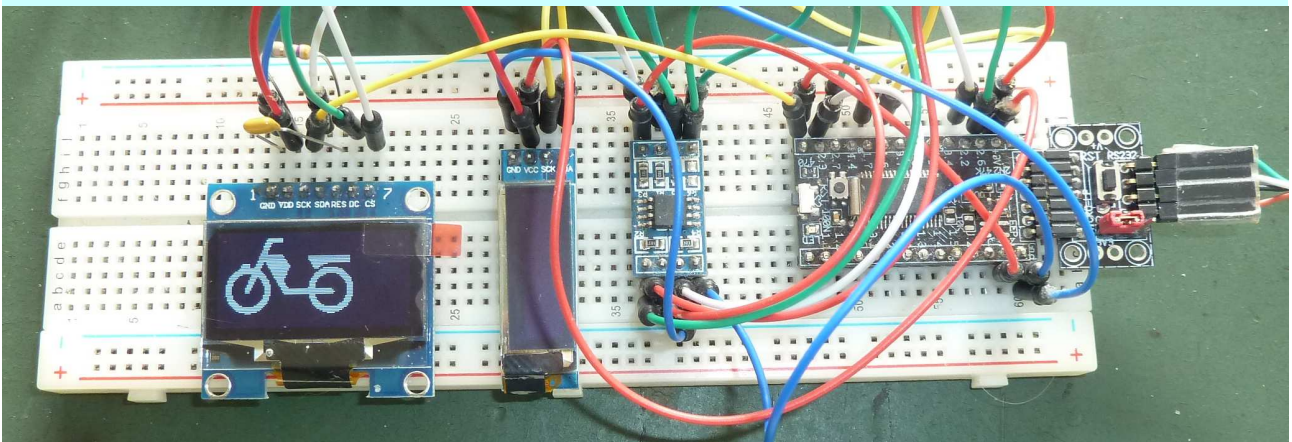


# Bitmaps maken en toepassen

Omdat zelf bitmaps maken een behoorlijk creatieve en arbeidsintensieve klus is, heb ik naar een methode gezocht om plaatjes te hergebruiken.



## Wat hebben we nodig:

- 1) Browser voor het zoeken naar plaatjes en voor het conversie programma "IMAGETOCPP"
- 2) Beeld bewerkings programma om plaatjes te knippen en plakken b.v. IRFANVIEW
- 3) Teken programma om plaatjes te kunnen te kunnen aanpassen b.v. PAINT
- 4) Programma editor als CRIMSON
- 5) noForth versie 190101 met workshop software
- 6) Een Egel-kit, MLP, Launchpad, of FRAM bordje en een OLED scherm met I2C of SPI interface.

## Wat moet je doen:

- 1) Plaatje maken of zoeken met de browser
- 2) Mogelijk bewerken met IRFANVIEW en/of PAINT
- 3) In browser IMAGETOCPP starten en het plaatje importeren  
<https://javl.github.io/image2cpp>

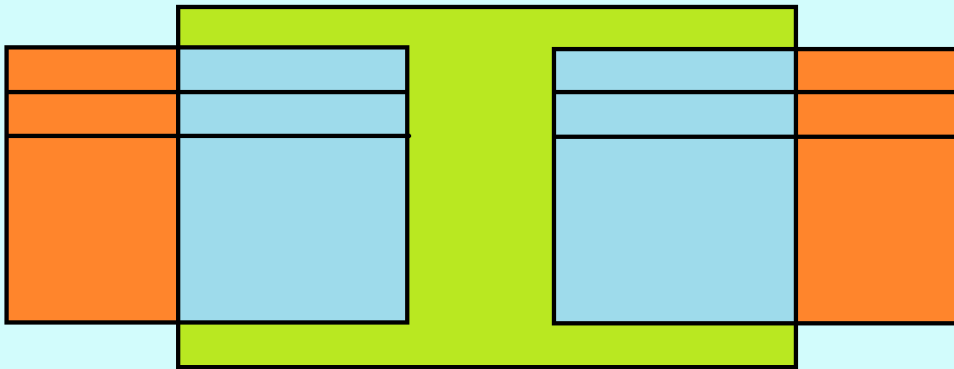
## In IMAGETOCPP:

- 1) Gewenste maat opgeven
- 2) Instellingen aanpassen
- 3) Bitmap genereren
- 4) Bitmap converteren voor gebruik in noForth: CREATE "naam"  
dm xx dm yy PIC:
- 5) Bitmap testen: "naam" .DEMO

## Extra notities:

In IRFANVIEW een uitsnede verslepen kan, door de cursor binnen het vak te zetten en de rechter muisknop in te drukken. Daarmee houdt je de maat van de uitsnede hetzelfde voor elk plaatje.

## OLED Beeldscherm



## Het afbeelden van delen van plaatjes

### De code voor het afbeelden:

```
value LINES  value ROWS
value 'PIC   value DELAY
Value XS      \ XS = Step Size
: SKIP?      ( -- +n a ff | tf )
  rows 'pic   \ +n a
  begin onscr? 0= while \ +n a Skip offscreen part
    1+ incr x   \ +n a+1
  zero? until drop \ -
  rows +to 'pic incr y
  x rows - to x true exit then false ;

: .BBITROW    ( -- ) \ Output one line of a graphic symbol
  skip? ?exit x y xy 40 {ol
  begin
    onscr? if \ Bitrow outside valid screen area?
      count i2out \ Output every second bit row if on screen
    else 1+ \ To next bitrow
    then incr x
  zero? until drop i2stop}
  x rows - y 1+ xy \ Next line
  rows +to 'pic ; \ To next picture row

: GET-PICT ( a -- ) @+ to lines @+ to rows to 'pic ;
: .PIC ( a x y -- ) xy get-pict lines 0 ?do .bbitrow loop ;

: .DEMO ( a -- )
  dup cell+ @ 2/ negate 40 + \ Calc. center
  &page 0 .pic ;
```

## Een geconverteerd plaatje:

```
\ 'thermometer 2' 48x48px
create THERMO2 30 30 pic:
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 f0 fc fe 0e 07 07 07 07 0e fe fc f0 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 ff ff ff 00 00 f8 f8 00 00 ff ff ff 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 ff ff ff 00 00 ff ff 00 00 ff ff ff 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 ff ff ff 00 00 ff ff 00 00 ff ff ff 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 f0
fc fe 1f 07 f3 f8 f8 ff ff f8 f8 f3 07 1f fe fc
f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03
0f 1f 3e 78 73 f7 e7 e7 e7 e7 f7 73 78 3e 1f 0f
03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

## De code voor het inlezen:

```
\ Compile pictures and picture frames for animation
: (PIC)      ( xp yp -- +n )
    8 / dup , over , * ;

: PIC:      ( xp yp ccc -- ) \ Read a bitmap
    (pic) 0 ?do              \ Read +n bytes bitmap
    0. bl-word count \ Get next data byte
    2dup upper          \ Make string uppercase
    >number             \ Convert to number
    2drop drop          \ Save lowest cell only
    c,                  \ Store in memory
  loop ;

: PIC;      ( -- )      0 , ;
```