Introducción a Psychtoolbox

Int. a la Neurociencia Cognitiva y Computacional

1 de septiembre de 2014

Registrando

Nos interesa saber qué algoritmo corre el cerebro para resolver una tarea.

¿Qué podemos medir?

Fisiología:

- Técnicas invasivas (*pinchar* moscas, ratas, monos, etc...)
- No invasivas (EEG, NIRS, fMRI, etc...)

Psicofísica:

- Tipo de respuesta
- Tiempo de respuesta

Vamos a inferir la algoritmia del cómputo humano a partir de experimentos psicofísicos.

PsychToolBox

Software para psicofísica que usaremos: Psychophysics Toolbox, versión 3. `http://psychtoolbox.org/`

> *Psychophysics Toolbox Version 3 (PTB-3) is a free set of Matlab and GNU/Octave functions for vision research. It makes it easy to synthesize and show accurately controlled visual and auditory stimuli and interact with the observer.*

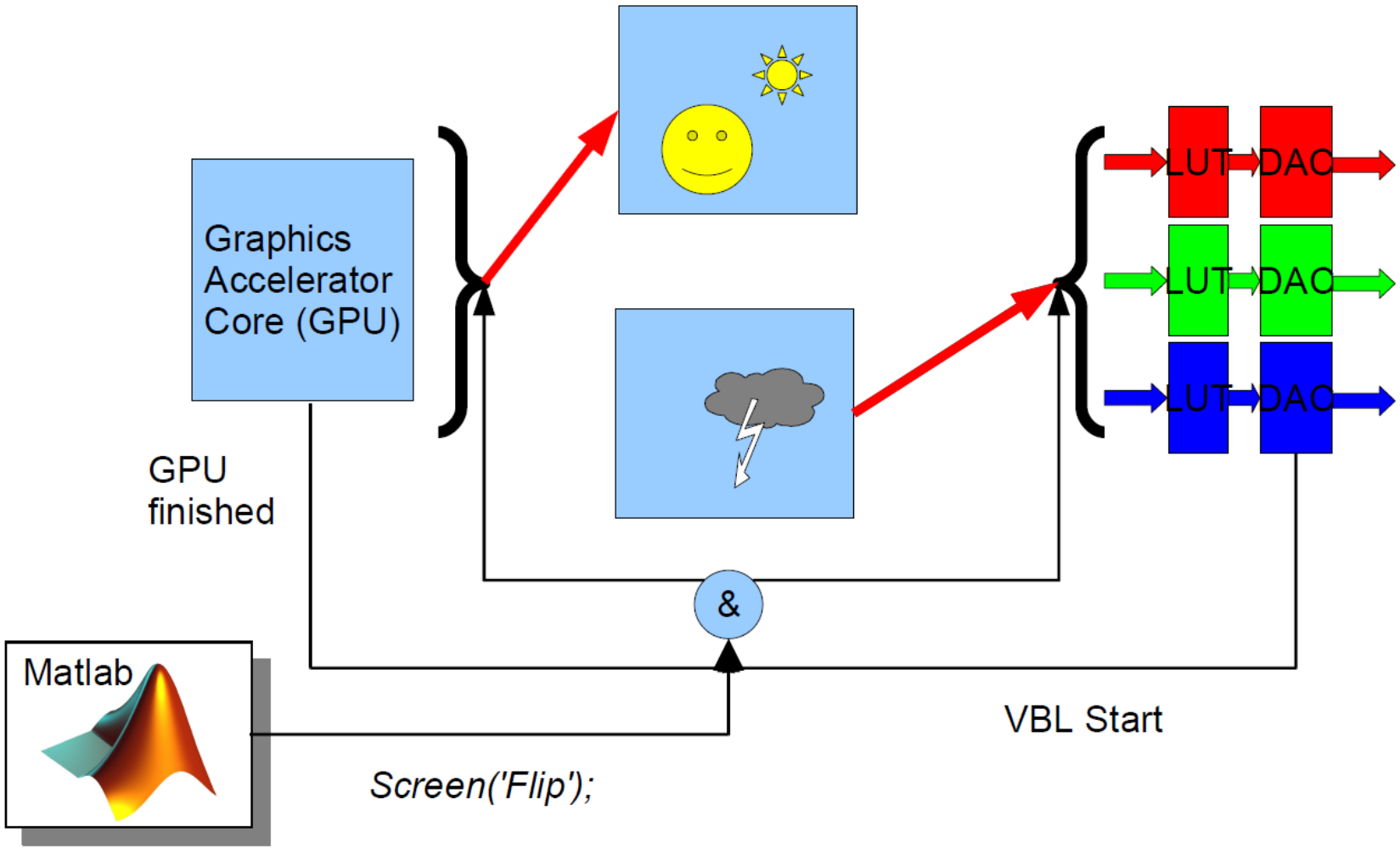
PsychToolBox

Instalación:

1. Instalar Subversion (SVN)
   - Windows: `http://tortoisesvn.tigris.org/`
   - Mac: `http://downloads.open.collab.net/binaries.html`
   - Linux: Debian/Ubuntu: `apt-get install subversion`
2. Bajar el instalador `http://svn.berlios.de/viewvc/*checkout*/osxptb/trunk/Psychtoolbox/DownloadPsychtoolbox.m`
3. En Matlab, ejecutar: `DownloadPsychtoolbox('<directorio donde quieren instalar>')`
4. Testearlo: >>`ScreenTest`

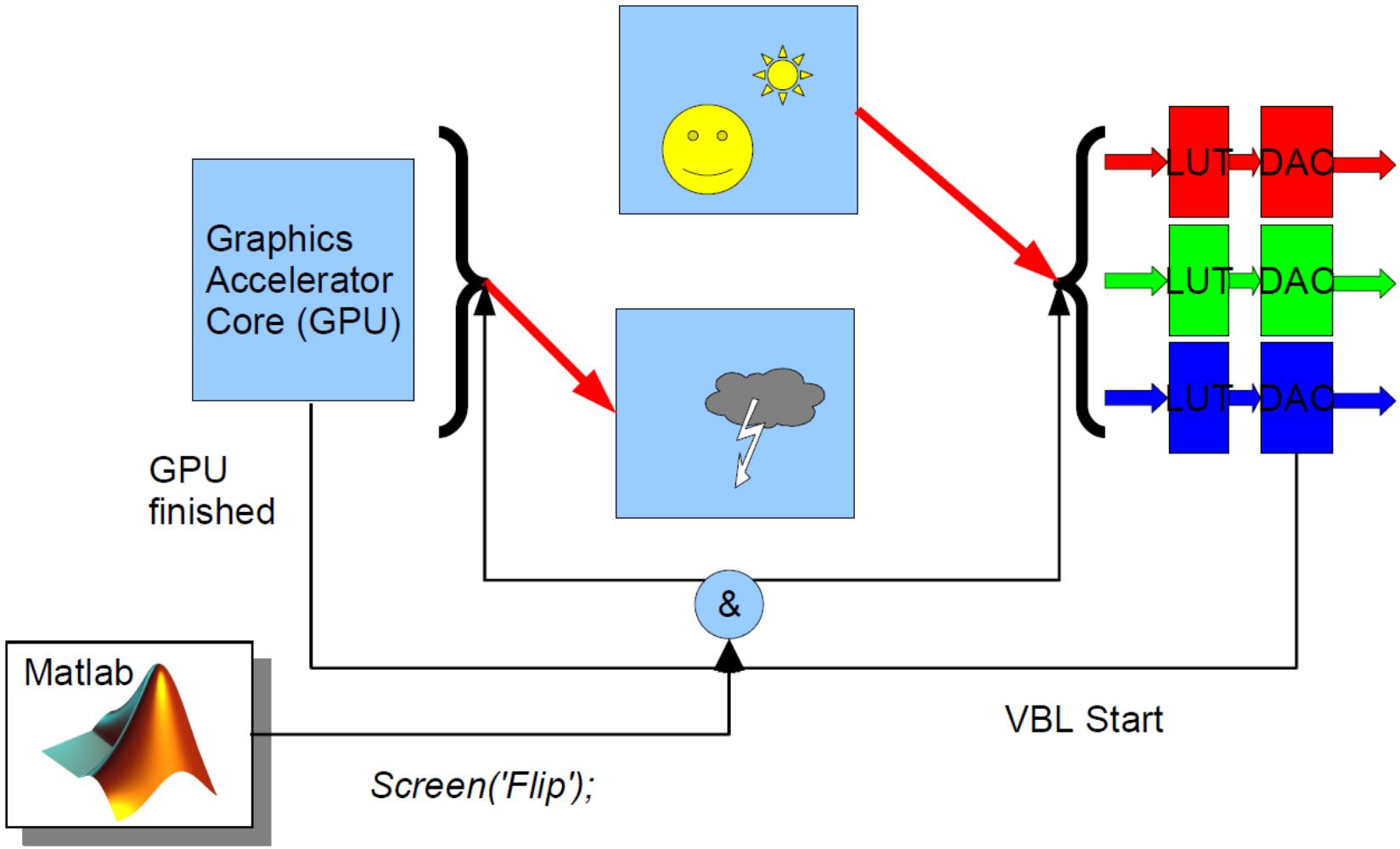
PsychToolBox

Double buffered drawing model - Implementation:



PsychToolBox

Double buffered drawing model - Implementation:



PsychToolBox

"Todo" se hace con `Screen(...)`. Primer "experimento": pantalla negra, pantalla blanca.

```
screenNum=0;
res=[1280 1024];
clrdepth=32;
[wPtr,rect]=Screen('OpenWindow',screenNum,0,[0 0 res(1) res(2)], clrdepth);
black=BlackIndex(wPtr);
white=WhiteIndex(wPtr);
Screen('FillRect',wPtr,black);
Screen(wPtr, 'Flip');

HideCursor;
tic
while toc<3
  ;
end
Screen('FillRect',wPtr,white);
Screen(wPtr, 'Flip');
HideCursor;
tic
while toc<3
  ;
end
Screen('CloseAll');
ShowCursor;
```

PsychToolBox

- `[windowPtr, rect] = Screen('OpenWindow', windowPtrOrScreenNumber [,color] [,rect] [,pixelSize] [,numberOfBuffers] [,stereomode] [,multisample] [,imagingmode] [,specialFlags]);`
- `[VBLTimestamp StimulusOnsetTime FlipTimestamp Missed Beampos] = Screen('Flip', windowPtr [,when] [,dontclear] [,dontsync] [,multiflip]);`
- `[ monitorFlipInterval nrValidSamples stddev ] = Screen('GetFlipInterval', windowPtr [,nrSamples] [,stddev] [,timeout]);`

PsychToolBox

Segundo "experimento": pantalla negra, pantalla blanca.

```
screenNum=0;
res=[1280 1024];
clrdepth=32;
[win,rect]=Screen('OpenWindow',screenNum,0,[0 0 res(1) res(2)], clrdepth);
black=BlackIndex(win);
white=WhiteIndex(win);
Screen('FillRect',win,black);

refresh = Screen('GetFlipInterval', win)
%Synchronize to retrace at start of trial/animation loop:
vbl = Screen('Flip', win);
%Loop: Cycle through 300 images:
tic
for i=1:30
    %Draw i'th image to backbuffer:
    Screen('DrawTexture', win, myImage(i));
    %Show images exactly 2 refresh cycles apart of each other:
    if mod(i,2)==0
        Screen('FillRect',win,black);
    else
        Screen('FillRect',win,white);
    end
    vbl = Screen('Flip', win, vbl + (2-0.1) * refresh);
    %Keyboard checks, whatever... Next loop iteration.
end;
%End of animation loop, blank screen, record offset time:
toffset = Screen('Flip', win, vbl + (2 - 0.5) * refresh);
toc
Screen('CloseAll');
ShowCursor;
```

PsychToolBox

Dibujando con PsychToolBox:

- (Filled) Circles and ellipses:
  - Screen('FrameOval', window, color, boundingrect [, penWidth]);
  - Screen('FillOval', window, color, boundingrect);
- (Filled) Rectangles:
  - `Screen('FrameRect', window, color, boundingrect [, penWidth]);`
  - Screen('FillRect', window, color, boundingrect);
- Lines of different thickness and stipple patterns:
  - Screen('DrawLine', window, color, fromH, fromV, toH, toV[,penWidth]);
- (Filled) Arcs:
  - `Screen('DrawArc', window, color, boundingrect, startAngle, arcAngle);`
  - `Screen('FrameArc', window, color, boundingrect, startAngle, arcAngle);`
  - `Screen('FillArc', window, color, boundingrect, startAngle, arcAngle);`
- (Filled) convex and concave Polygons:
  - Screen('FillPoly', window, color, xy);

PsychToolBox

Procesando Batch:

- En lugar de:

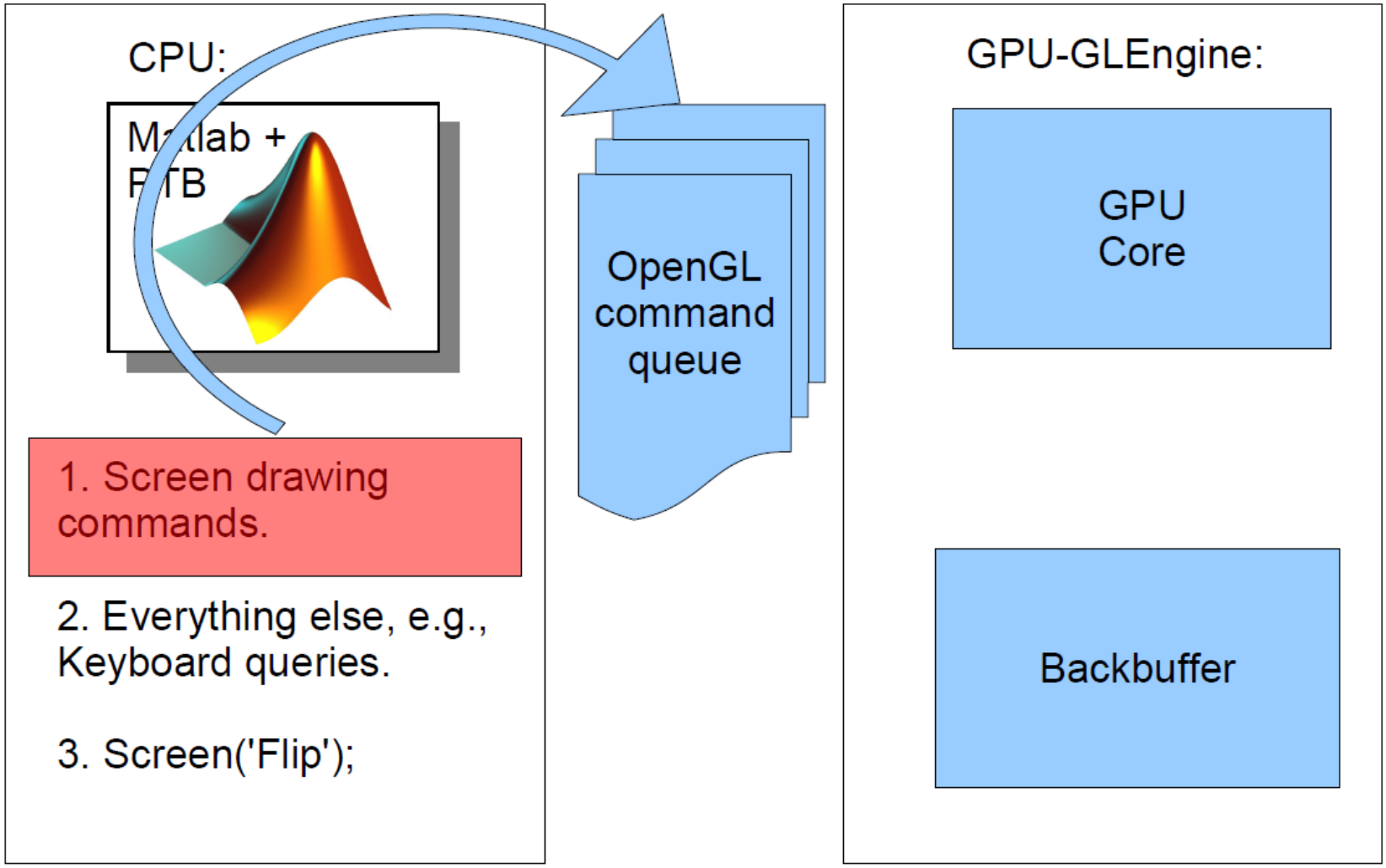
```
Screen('FillRect', win, [red1 green1 blue1], [left1 top1 right1 bot1]);
Screen('FillRect', win, [red2 green2 blue2], [left2 top2 right2 bot2]);
...
Screen('FillRect', win, [redn greenn bluen], [leftn topn rightn botn]);
```

- Usar:

```
mycolors = [red1 green1 blue1; red2 green2 blue2; ... ; redn greenn bluen];
myrects = [left1 top1 right1 bot1; left2 top2 right2 bot2; ... ; leftn topn rightn botn];
Screen('FillRect', win, mycolors, myrects);
```

PsychToolBox

Procesando Batch:



PsychToolBox

Dibujando con PsychToolBox:

```
image2D=255*rand(100, 100);
textureIndex=Screen('MakeTexture', wPtr, image2D);
Screen('DrawTexture', wPtr, textureIndex);
```

PsychToolBox

System control and timing:

- `T = GetSecs`
  - Query time with microsecond resolution.
  - Uses highest resolution system clock for measurement of time.
- `WaitSecs(duration)`
  - Wait for a specified amount of time 'duration'.
  - On MacOS/X: Accurate to 0.3 milliseconds on average.
  - On MS-Windows: Accurate to 2 milliseconds on average on a modern machine.
- `Priority()` - Switch Matlab process to realtime-scheduling mode.

PsychToolBox

Obteniendo respuestas de teclado y mouse:

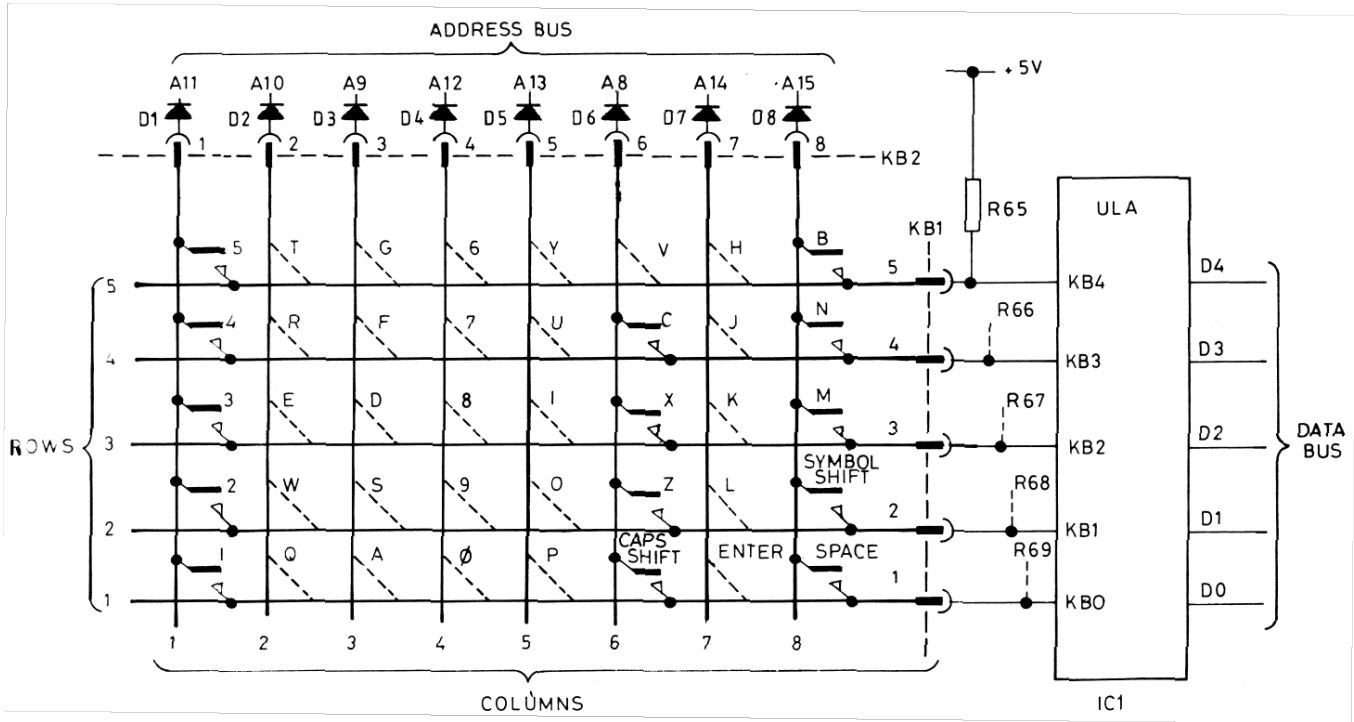
- `[x,y,buttons]=GetMouse(window);`: Query current mouse position and mouse button state.
- `[down, secs, keycode]=KbCheck;` Query current state of all keys on a keyboard
- `secs = KbWait;` Just like Kb, but it waits until a key on the keyboard is pressed down and simply returns the time that the key was pressed
  - Can detect and report multiple simultaneous keypresses.
  - Can query multiple keyboards on Linux and OS/X, one on Windows.
  - Can mask out dead or stuck keys, often a problem on Laptops. `DisableKeysForKbCheck()`

Importante:

- Queries are fast, bypassing OS queues and application event queues.
- Despite that: Standard keyboards problematic for RT measurements!

PsychToolBox

¿El teclado es lento?



Kb-Encoder: 5-20 ms	Debouncer: Up to20 ms	USB: Up to 10 ms

PsychToolBox

A LABURAR!