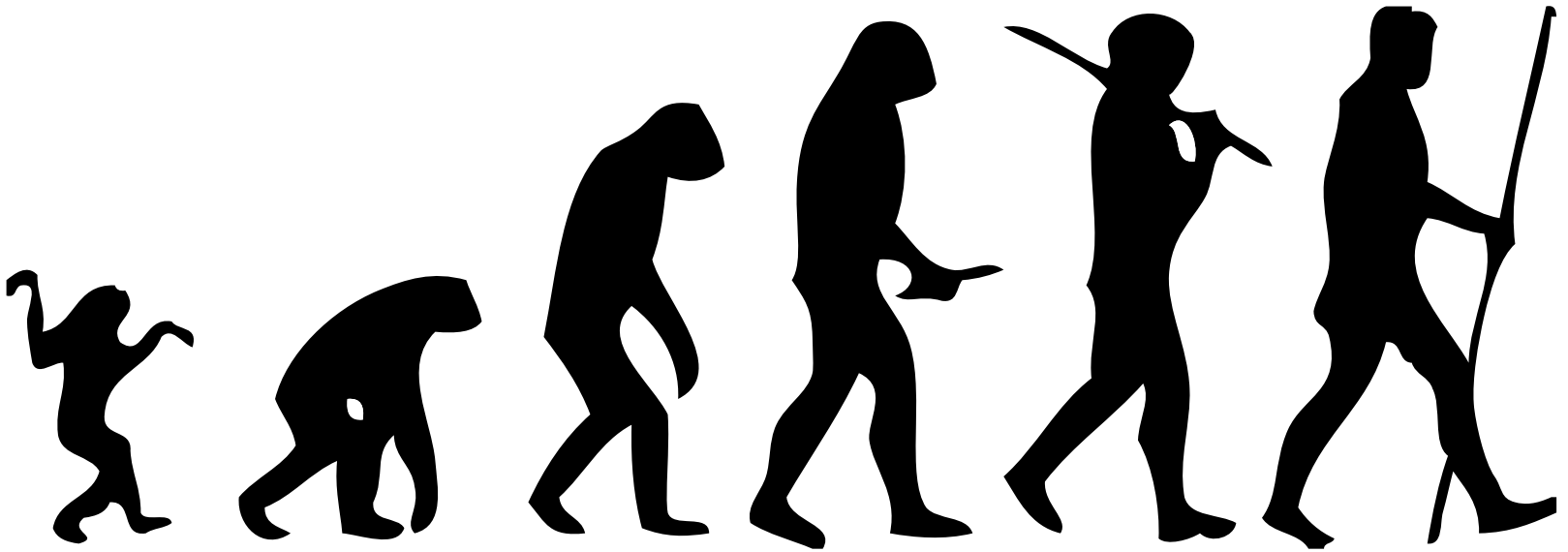
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DISCA

# Desarrollo en C/C++ para la RPi



2016/2/20

Àngel Perles

# Contenido

- Objetivo
- Introducción
- El desarrollo “cruzado”
- Herramientas GNU C compiler (GCC)
  - Instalar
  - Primer programa
  - Bibliotecas y wiringPi
  - “Activity”

# Objetivo

- Comprender el concepto de desarrollo cruzado
- Trasladar conocimientos de C a la plataforma RPi
- Aprender a usar bibliotecas externas

C/C++

# Introducción

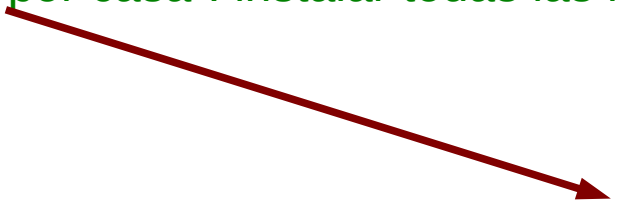
**while (1) {**

- C/C++ es “clásico”, nativo, portable, modular, bla, bla ... asumimos que lo sabéis
- “El lenguaje” universal del mundo embebido y no embebido
  - cualquier “cacharro” dispone siempre de ensamblador y de C/C++
  - de bajo/medio nivel --> ideal para la programación de periféricos
- Pegas
  - poco expresivo, retorcido, mal tipado, traga aberraciones, muy susceptible a errores de programación, ... durillo para novatos, ...
- ¡Puñeta!, ¿y por qué todo esto está hecho en C?
  - freno ABS de tu coche, bomba de insulina, marcapasos, ...
  - satélite artíficial, GPS, ...

**} // end while**

# Introducción

- Se asume la imposibilidad de escapar de C/C++
  - (Hay una notable falta mundial de ingenieros formados en este área)
- Soluciones/paliativos a los defectos de C
  - Restringir las construcciones C/C++ a usar, i.e. MISRA C
  - Usar generadores automáticos de código
  - ... pero escapa del ámbito del curso
- ¡Quiero usar C/C++ en la RPi!
  - “Profesional”: usar desarrollo cruzado
  - “De ir por casa”: instalar todas las herramientas en la propia SD

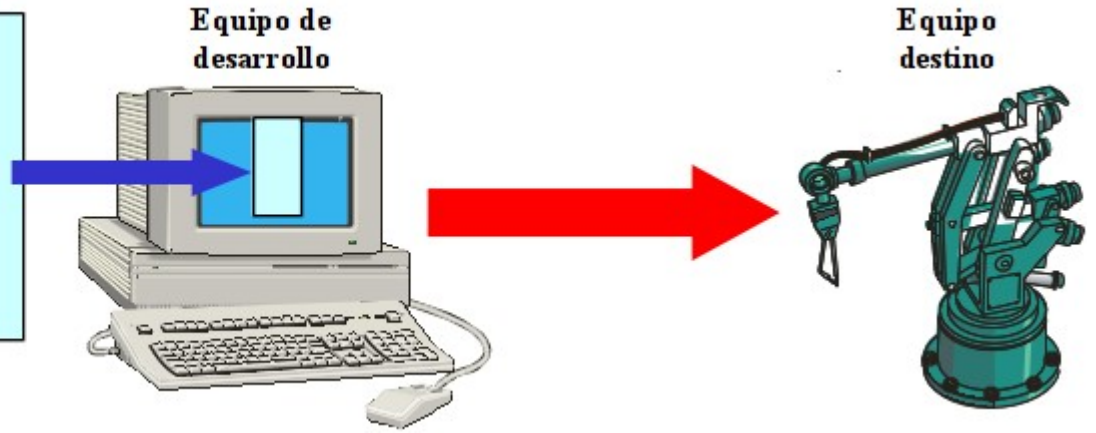


Este curso

# El desarrollo “cruzado”

- La plataforma de desarrollo (“host”) es distinta a la plataforma destino (“target”)
  - Facilita enormemente el desarrollo
  - Pero poner a punto el entorno suele ser complicado

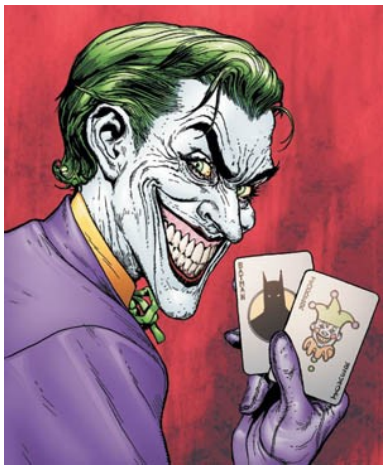
```
sfr P0=0x80;
sfr P1=0x90;
void main(void){
  while(1)
        P1=~P0;
}
```




- Descartada por que hay que explicar muuuchas cosas.
- No sirve para introducir RP

# GCC: Instalar

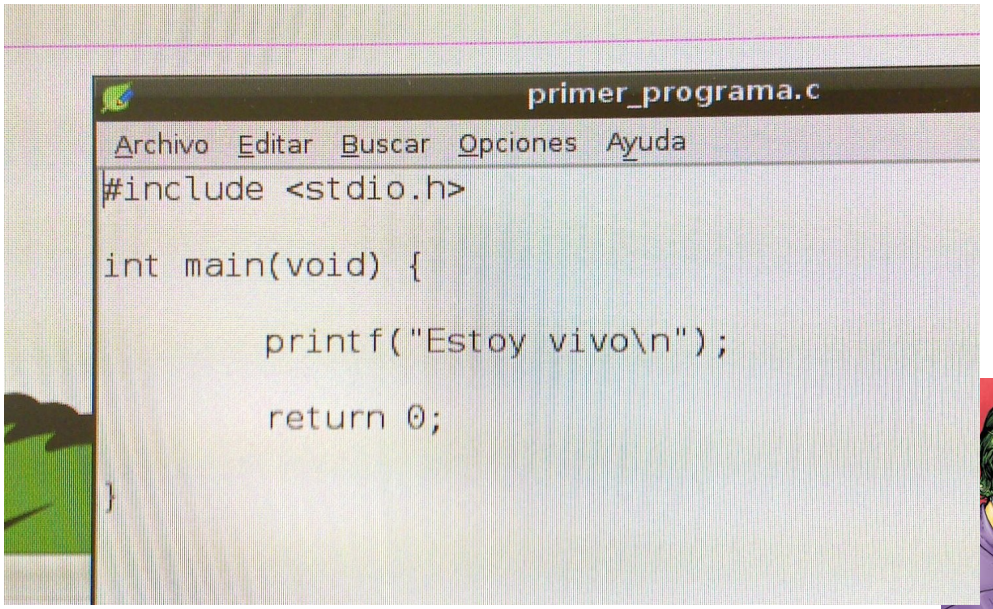
- GNU C Compiler
  - De código abierto, libre y disponible para infinidad de plataformas (de micros a supercomputadores)
  - Lo instalamos en la RPi y trabajamos en ella directamente
    - pi$ sudo apt-get install gcc
  - Usando el compilador “a pelo”
    - pi$ gcc -o hola main.c



NOTA: Parece que se ha instalado GCC, pero es una falacia.
En realidad, se ha instalado arm-linux-gnueabihf-gcc-4.6

# GCC: Primer programa

- Vamos a probar:
  - pi$ cd
  - pi$ mkdir test
  - pi$ cd test
  - pi$ leafpad primer_programa.c &  (suponemos terminal dentro X o entubado X)
  - pi$ gcc -o prueba primer_programa.c
  - pi$ ./prueba




# GCC: Bibliotecas y wiringPi

- Lo bueno de Linux es la cantidad de bibliotecas disponibles
  - Hemos instalado wiringPi. ¡¡¡¡¡ Podemos usarlo !!!!!

```
#include <wiringPi.h>
int main (void)
{
  wiringPiSetupSys () ;
  pinMode (17, OUTPUT) ;
  for (;;)
  {
    digitalWrite (17, HIGH) ;
    delay (500) ;
    digitalWrite (17, LOW) ;
    delay (500) ;
  }
  return 0 ;
}
```

- Construir
  - gcc -o blink blink.c -lwiringPi

# GCC: Bibliotecas y wiringPi

- Probando
  - $ ./blink
- Upsss! No va!!! … hay que “exportar el pin” para dejarlo accesible
  - $ gpio export 17 out
  - $ ./blink
- Hay otras opciones, pero implican “sudo”

# GCC: “Activity”

- Crea el módulo *valve.c* partiendo de la siguiente cabecera *valve.h*

```
/**
  @file valve.h
  @brief Header providing generic functions for handling a valve
*/

#ifndef VALVE_H
#define VALVE_H

typedef enum {VALVE_OPEN, VALVE_CLOSE} TValveState;

void valve_Init(void);
void valve_SetState(TValveState st);

#endif
```

- You are not alone in the world
- If you are an engineer, then you should use english for developing software

# GCC: “Activity”

- Ejemplo de programa principal que usa el módulo

```c
/**
  @file test_valve.c
  @brief For testing the "valve" module
*/

#include <unistd.h>
#include "valve.h"

int main(void)
{
  valve_Init();

  while(1) {
    valve_SetState(VALVE_OPEN);
    sleep(1);
    valve_SetState(VALVE_CLOSE);
    sleep(1);
  }

}
```

- Construir y probar el ejemplo
  - gcc -c valve.c
  - gcc -o test_valve test_valve.c valve.o -lwiringPi
  - ./test_valve


The End