



SENSORS &
FIELD TRANSMITTERS

HUMAN MACHINE
INTERFACE (HMI)

INDUSTRIAL
MOTOR DRIVE

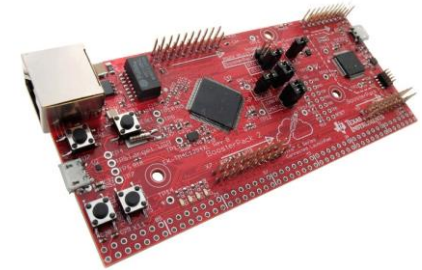
INDUSTRIAL
COMMUNICATION

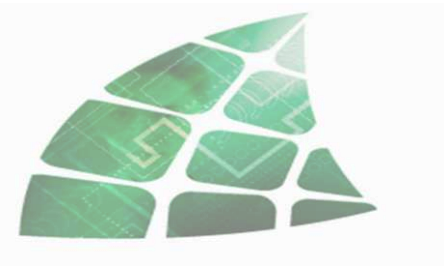
PROGRAMMABLE
LOGIC CONTROL (PLC)

Tema 3. Sistemas procesadores para automatización

4º Grado de Ingeniería en Electrónica, Robótica y Mecatrónica
Andalucía Tech

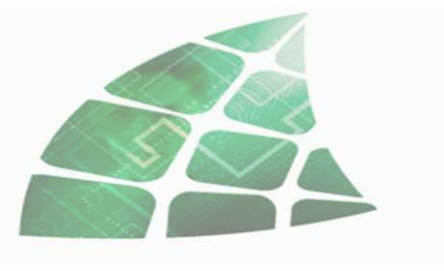
- Introducción
- Familia ARM Cortex M4F
- Microcontrolador TM4C1294
- Tarjeta *Connected Launchpad*
- Programación del TM4C1294





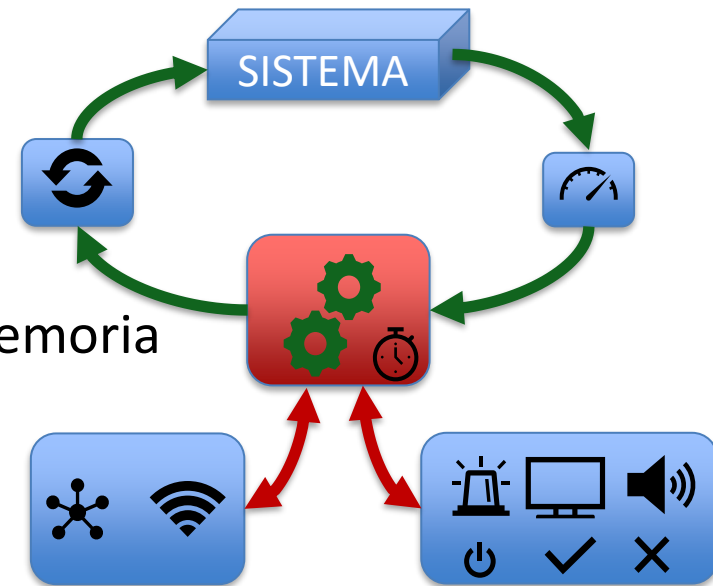
Objetivos

- Fijar las necesidades de un sistema microcontrolador para SEPA
- Presentar la familia TIVA y en concreto el TM4C1294
- Sentar las bases de cómo realizar un proyecto con este micro en CCS.



Introducción

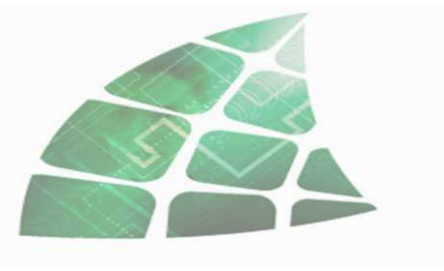
- Sistema de control
- Necesidades básicas:
 - Multitud de pines de E/S
 - Periféricos de alta resolución
 - Velocidad de proceso y cantidad de memoria adecuada
 - Programación en tiempo real
 - Comunicaciones
- Lógica programable
- Sistema microprocesado
 - Microcontrolador
 - Microprocesador





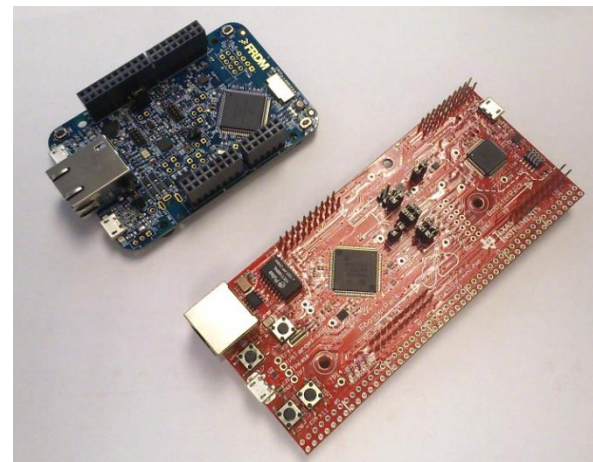
Introducción

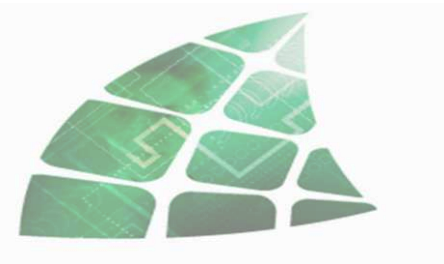
- Microcontroladores básicos:
 - Pocas capacidades de cálculo
 - Limitación de memoria
 - HMI pobres (botones, led's, displays de texto)
- Microprocesadores:
 - Poco robustos
 - Sistemas operativos (poco fiables)
 - Necesidad de gran cantidad de periféricos



Solución *ideal*

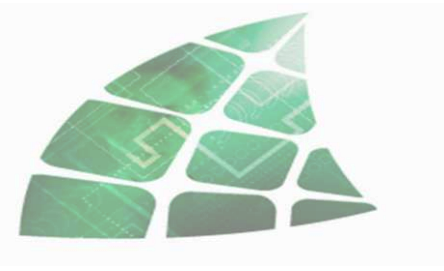
- Microcontrolador más potente
 - Más memoria
 - Periféricos de alto nivel
 - USB, Ethernet, CAN...
 - Herramientas de desarrollo baratas
 - Tiva TM4C1294XL (TI)
 - Freedom K64F (NXP)
 - MSP432 (TI)
 - NUCLEO-F446RE (ST)
 - Hercules RM46 (TI)





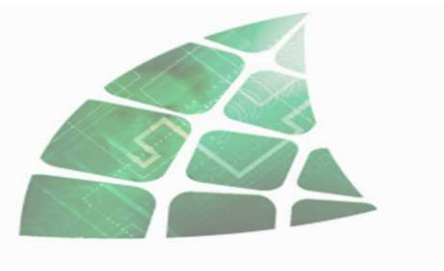
Comparación (@2015)

	TIVA-Connected	FRDM-K64F
Microcontrolador	TM4C1294NCPDT @120MHz	MK64FN1M0VLL12 @120MHz
Núcleo	ARM M4F, 32 Bit-P.F.	ARM M4F, 32 Bit-P.F.
Memoria (FLASH/RAM)	1M/256k	1M/256k
Pines	128 pines	100 pines
Extras	4 led, 3 botones, 80 pines, compatibilidad Launchpad, 2 usb, 1 ethernet	Acelerómetro+magnetómetro, rgb led, 2 botones, 54 pines, Compatibilidad arduino, 2 usb, 1 ethernet, 1 micro SD
Precio	19,98€	25€



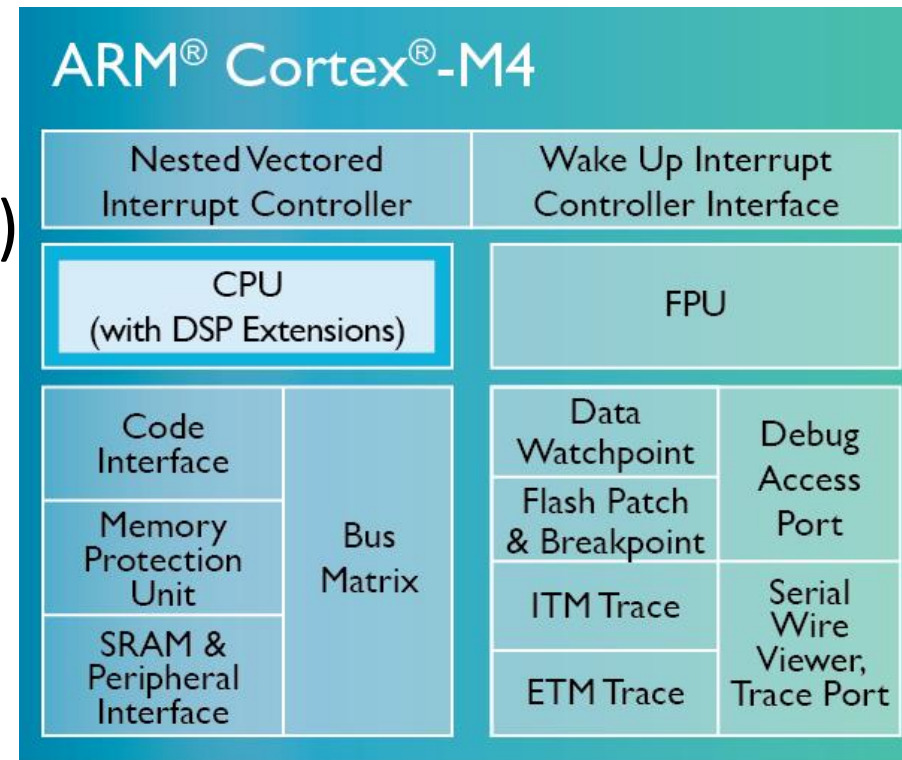
Microcontroladores ARM

- Núcleo de instrucciones RISC, de 32 bits
- Primer desarrollo: Acorn Archimedes (1987)
- Presentes en muchos dispositivos móviles
- Actuales: Núcleo ARM CORTEX:
 - Cortex-A: procesadores potentes (aplicaciones)
 - Cortex-R: aplicaciones en tiempo real (seguridad)
 - Cortex-M: perfil microcontrolador (periféricos)



ARM Cortex M4F

- Núcleo de altas prestaciones
 - Instrucciones DSP (MAC)
 - FPU: unidad de punto flotante
 - Gran cantidad de herramientas de desarrollo



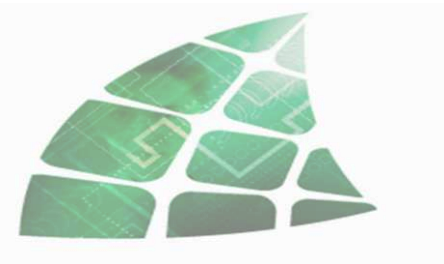


Familia TM4C (Tiva)

- Parte de una gran familia...

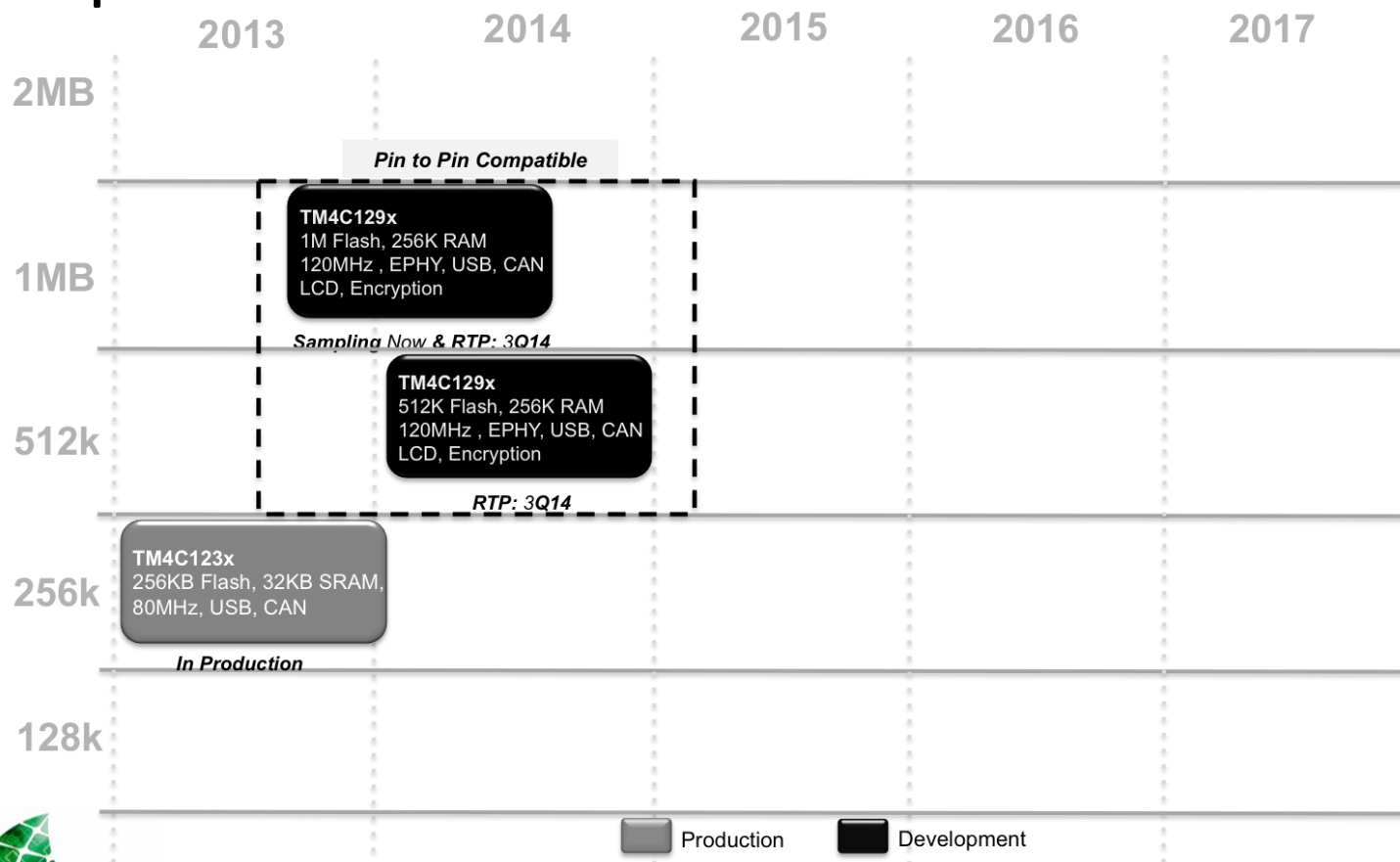
Microcontroller (MCU) Portfolio at a Glance			ARM®-Based Processor Portfolio at a Glance		Digital Signal Processor (DSP) Portfolio at a Glance	
MCU			Software, Tools, Kits & Boards			DSP & ARM® MPU
16-bit ultra-low power MCUs	32-bit real-time MCUs	32-bit ARM® MCUs	32-bit ARM® safety MCUs	32-bit ARM® processors	Singlecore DSPs	Multicore processors
MSP430™	C2000™	Tiva™ C Series ARM Cortex™-M4F	Hercules™ ARM Cortex-R4F	Sitara™ ARM Cortex-A8 ARM9™	C5000™ C6000™	C6000™ DSP and ARM Cortex-A15
Overview	Overview	Overview	Overview	Overview	Overview	Overview
Device Table	Device Table	Device Table	Device Table	Device Table	Device Table	Device Table
SW & Kits	SW & Kits	SW & Kits	SW & Kits	SW & Kits	SW & Kits	SW & Kits
Up to 25 MHz Flash 0.5 KB to 512 KB Analog I/O, ADC, LCD, USB, FRAM Measurement, sensing, general purpose \$0.25 to \$9.00	40 MHz to 300 MHz Flash, RAM 16 KB to 512 KB PWM, ADC, CAN, SPI, I²C Motor control, digital power, lighting, ren. energy \$1.85 to \$20.00	Up to 120 MHz Flash 32 KB to 1 MB ENET + PHY, LCD, USB, CAN, ADC, EMIF, ADC, Crypto, Tamper Industrial Communication, MicroPLC, IOT, Network Controller, Applications Processor \$2.15 to \$10.00	Fixed/floating up to 220 MHz Flash 256 KB to 3 MB USB, ENET, FlexRay™, Timer/PWM, ADC, CAN, LIN, SPI, I²C, EMIF Safety, transportation, industrial & medical \$5.00 to \$30.00	Up to 1.35 GHz Up to 32 KB I/D cache 256 KB L2, LPDDR, DDR2/3 support GEMAC, PCIe+PHY, SATA+PHY, CAN, USB+PHY, PR-ICSS Consumer, industrial, connected home, POS smart grid, medical \$5.00 - \$25.00	Up to 800 MHz DSPs SDRAM, DDR2 uPP, I²C, I²S, UHPI, McASP/McBSP, LCDC, integrated connectivity options: USB 2.0, EMAC Patient monitoring, biometric security, smart e-meter, industrial drives \$2.00 to \$25.00	Up to 10 GHz multicore, fixed/floating + accelerators Up to 4 MB SL2, 32 KB L1, 1 MB L2 RapidIO®, PCIe, McBSP, 10/100 MAC, uPP, UART, Hyperlink, DDR2/3 Telecom, medical, mission critical, base stations \$30 to \$225.00

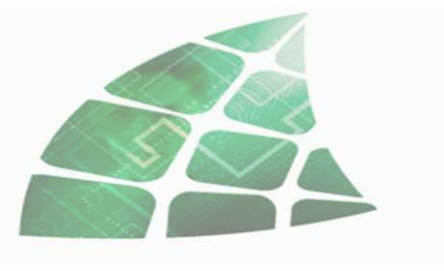




Familia TM4C (Tiva)

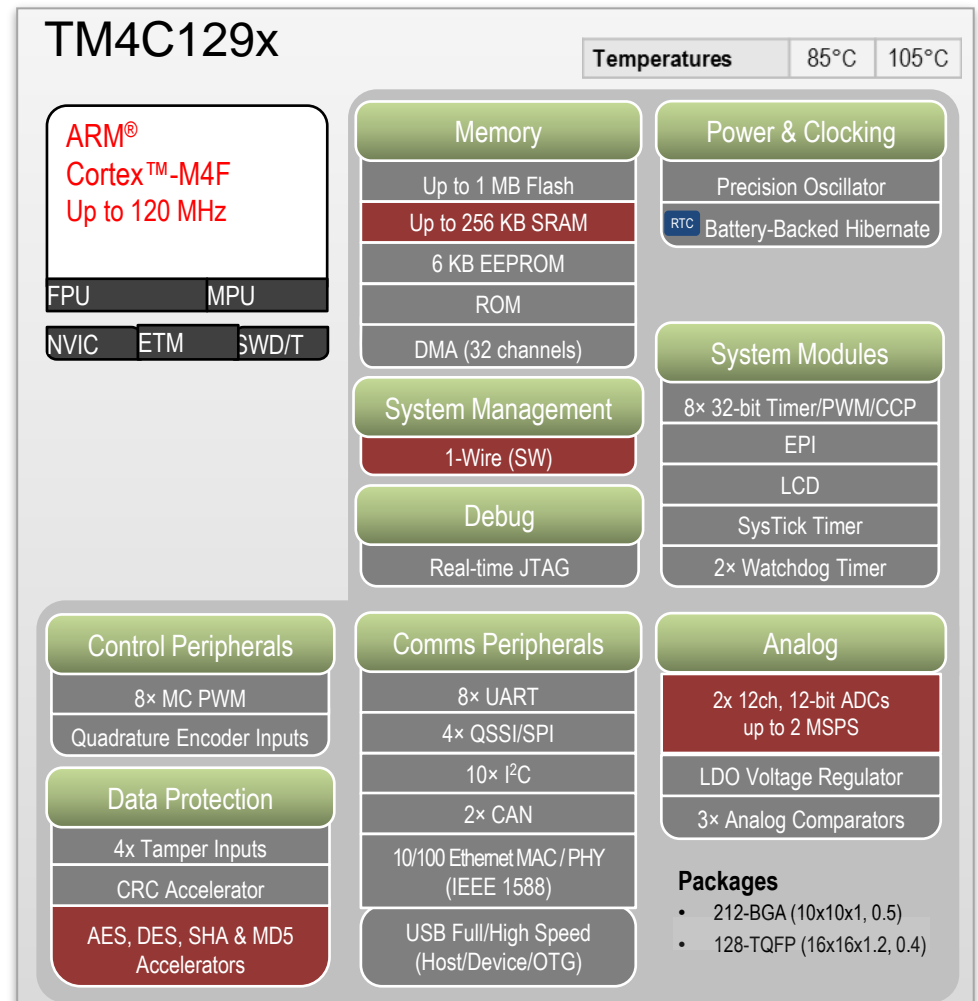
- ...en proceso de desarrollo

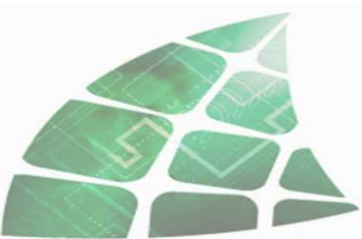




Familia TM4C (Tiva)

- Familia TM4C129x:
 - Core M4F
 - Flash, Ram y ROM
 - DriverLib
 - Bootloader
 - I/F serie y paralelo
 - Timers 16/32 bit
 - Control LCD
 - ADC 12 bits

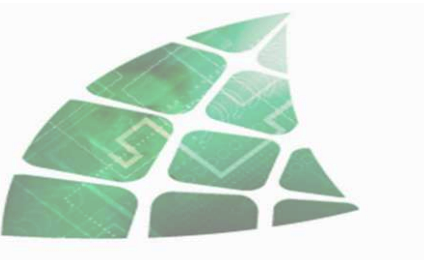




El procesador TM4C1294NCPDT

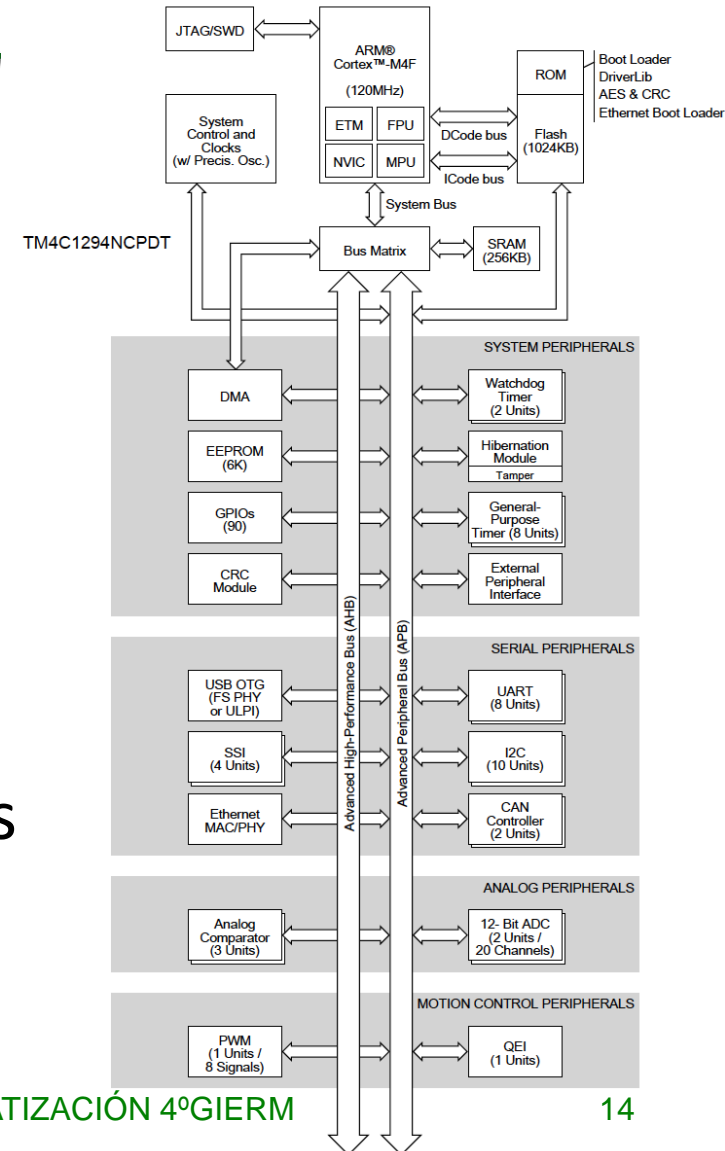
- 32-bit ARM® CortexM4F 120MHz / 150DMIPS CPU
- 1 MB Flash / 256 kB RAM / 6 kB EEPROM / ROM con librería TivaWare
- NVIC: Nested Vectored Interrupt Controller
- Interfaz Paralela externa de 8/16/32-bit
- 2 ADC de 12-bit / 2MSPS SAR ADCs con 16 comparadores digitales
- 3 comparadores analógicos
- 8 timers de 16/32-bit / 2 watchdog timers / 24-bit SysTick timer
- Módulo PWM con 4 parejas de salidas
- 32-Channel DMA
- 2 controladores CAN 2.0 A/B
- 4 QSSI / 8 UARTs / 10 I2C
- Full- & Low-speed USB 2.0
- 10/100 Ethernet MAC + PHY

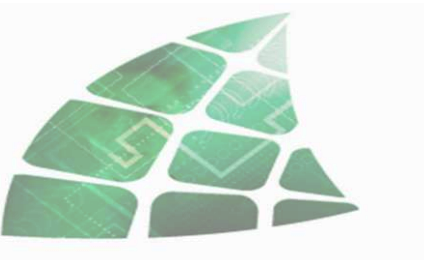




Arquitectura interna

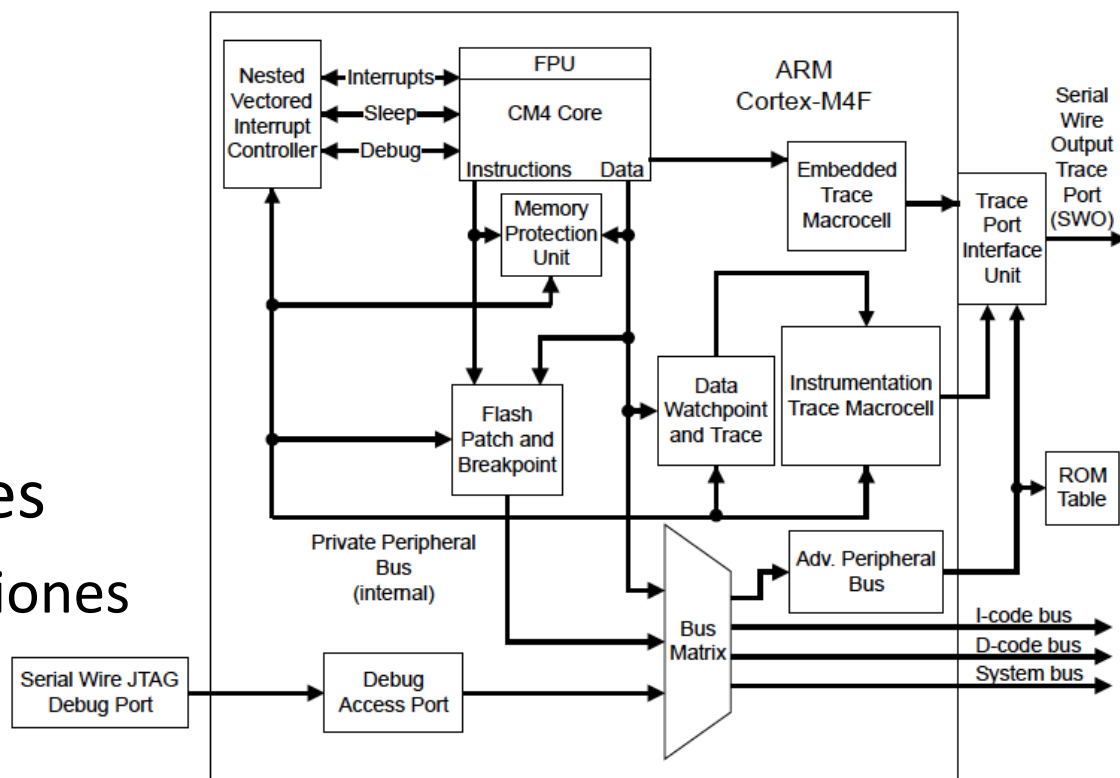
- Core
 - Bus específico para programa
 - JTAG
- Dos buses internos
 - AHPB
 - APB
 - Periféricos a alguno de los dos

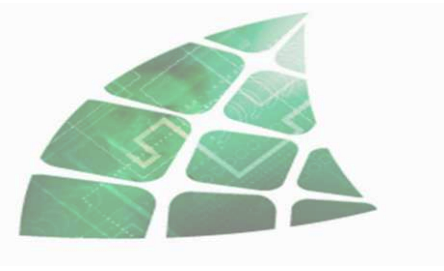




Cortex M4F

- Dentro del core:
 - NVIC
 - MPU
 - FPU
 - Selector de buses
 - Bus de instrucciones
 - Bus de datos
 - Bus de sistema

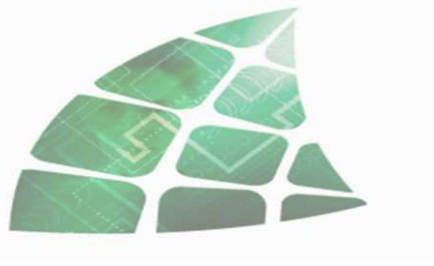




Mapa de memoria

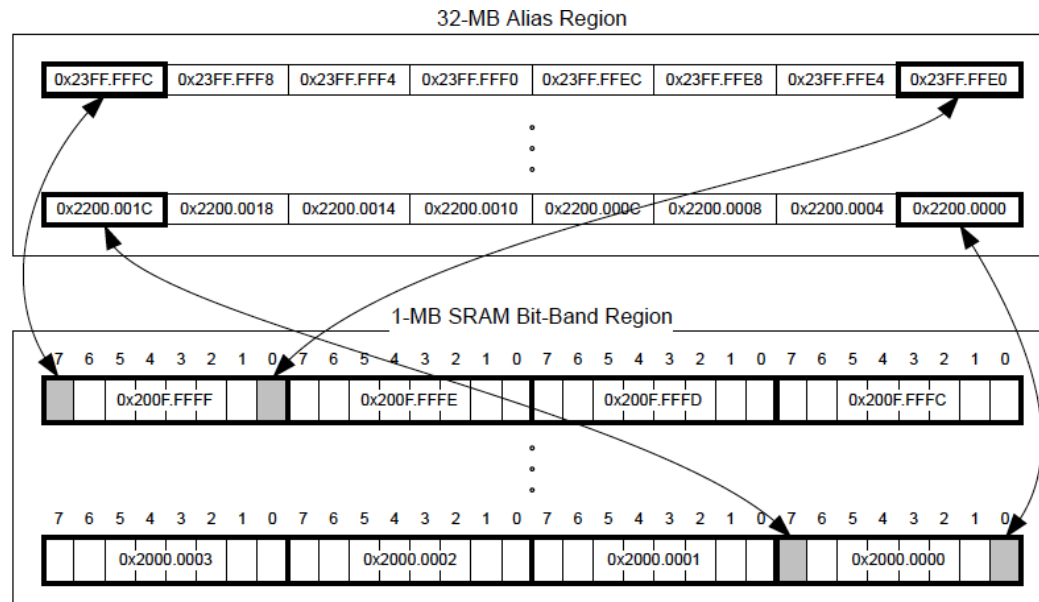
- 4GB de espacio
- Zona de Alias (bit-banding)
- Rom con:
 - Bootloader
 - Vectores iniciales
 - Librería DriverLib
 - Tablas para AES
- Zona de memoria *externa* (¿MC?)

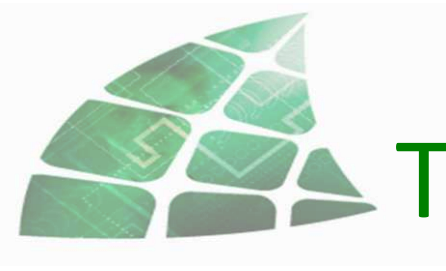
0x0000 0000	Flash
0x0010 0000	Reserved
0x0200 0000	ROM
0x2000 0000	SRAM
0x2200 0000	Bit-band alias of SRAM
0x4000 0000	Peripherals
0x4200 0000	Bit-band alias of Peripherals
0x6000 0000	External Peripheral Interface
0xE000 0000	Private Peripheral Bus



Bit-banding

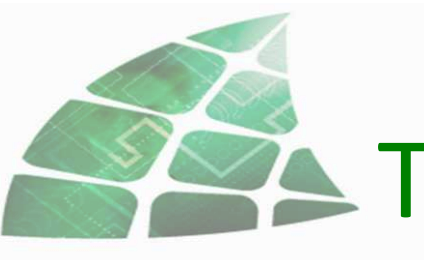
- Cada posición de memoria tiene 8 *alias* para cambiar un solo bit:
 - Escribir un 1 en la 0x2200-0018 es lo mismo que escribir un 0x40 en la posición 0x2000-0000



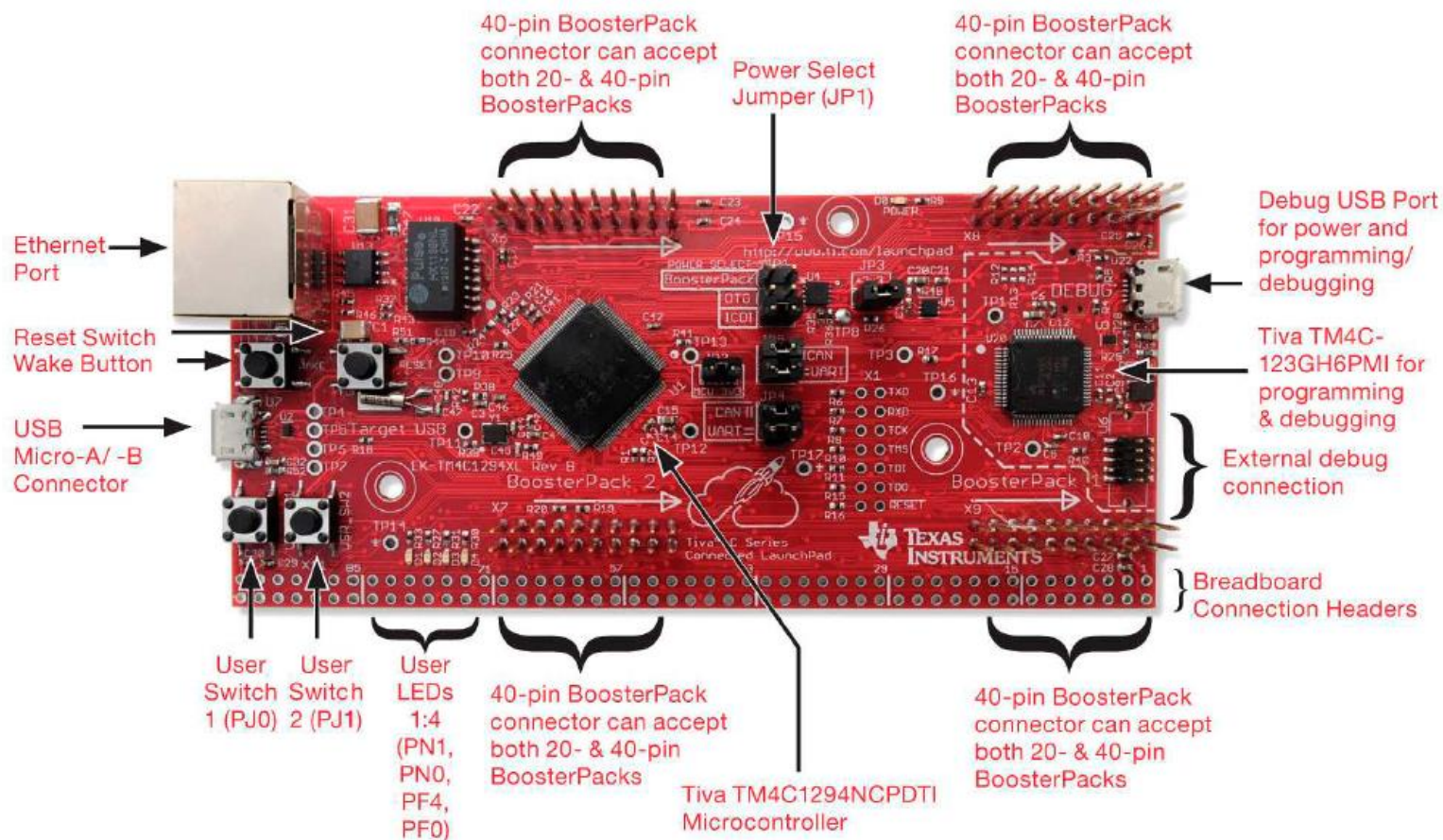


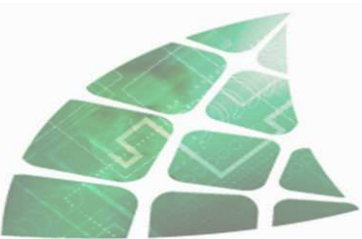
Tarjeta Connected LaunchPad

- Tarjeta de desarrollo rápido para el micro TM4C1294
- 2 conectores Boosterpack XL (40 pines)
- Conector Ethernet
- 2 puertos USB (depuración y periférico)
- Pines accesibles en el borde
- 2 botones y 4 leds programables



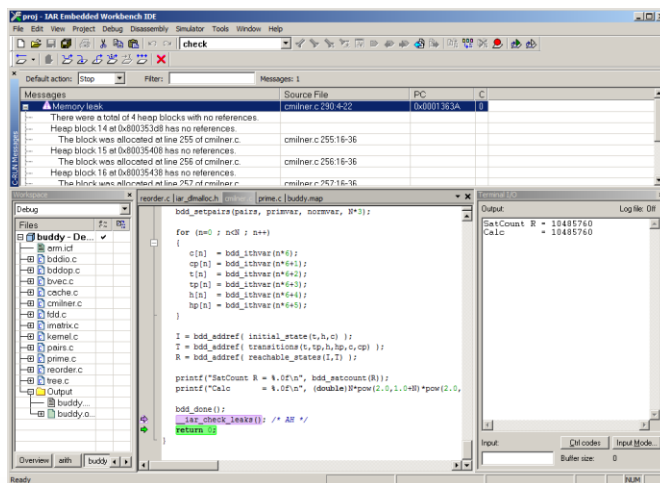
Tarjeta Connected LaunchPad





Programación del TM4C1294

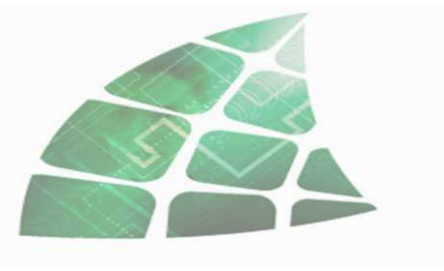
- Múltiples herramientas para la programación:
 - Code Composer Studio
 - IAR Systems, Keil
 - Energia





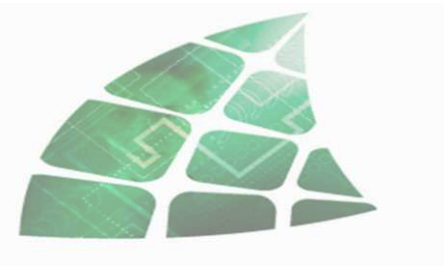
Code Composer Studio

- Actualmente, v 12.0.0 (Jul. 2022)
 - OJO: a partir de la 9, sólo versiones 64bit
- Herramienta *nativa* y ya usada
- Plug-ins (Mejor Online)
 - TivaWARE
 - Ejemplos y librerías de programación (driverlib, sensorlib, grlib...)
 - PinMux
 - Configuración guiada de los pines (offline y Cloud)
 - GUI Composer
 - Generación de interfaces gráficas en el PC
 - TI-RTOS
 - Sistema operativo en Tiempo Real

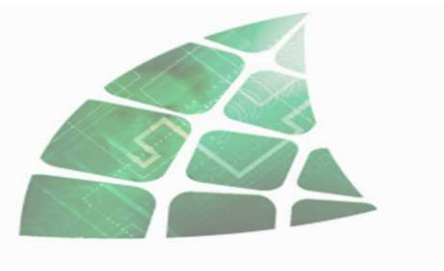


Cambio de paradigma

- Para programar microcontroladores simples:
 - Pocos recursos
 - Necesidad de optimización
 - Programación sobre registros
- Para microcontroladores potentes:
 - Mucha complejidad
 - Muchos recursos
 - Capas de abstracción(HAL, drivers, BSL...)



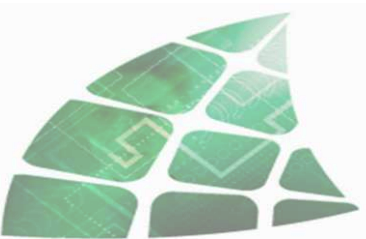
- Librería de funciones para configuración y manejo de los periféricos
- Grabada en la ROM del dispositivo
 - No ocupan memoria flash
 - También en código fuente modificable
- API: Application Programming Interface
- Capa de abstracción: no se escribe ya en registros



- Multitud de funciones predefinidas.
- Organizadas por periféricos
- Máscaras de bit definidas en los *.h
- Ejemplo:

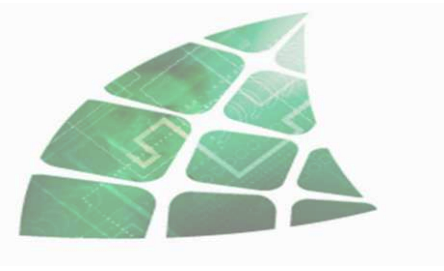
```
uint32_t reloj;  
reloj=SysCtlClockFreqSet((SYSCTL_XTAL_25MHZ | SYSCTL_OSC_MAIN | SYSCTL_USE_PLL  
|SYSCTL_CFG_VCO_480), 120000000);  
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPION);  
GPIOPinTypeGPIOOutput(GPIO_PORTN_BASE, GPIO_PIN_0|GPIO_PIN_1);  
GPIOPinWrite(GPIO_PORTN_BASE, GPIO_PIN_0|GPIO_PIN_1, 0x00);
```

- Más fácil, pero sintaxis un poco compleja



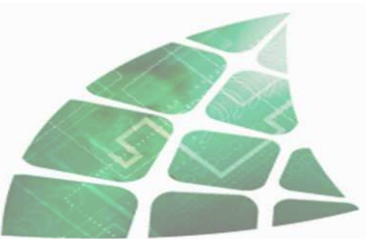
Instalación mínima y recomendada

- Instalación mínima:
 - Code Composer Studio
 - Versión 6.1 o mayor. Recomendada a partir de la 10.0
 - Win, Linux, Mac...
 - http://processors.wiki.ti.com/index.php/Download_CCS
 - TivaWare
 - Archivo comprimido .EXE (necesita PC!!)
 - <http://www.ti.com/tool/sw-tm4c>
- Instalación adicional
 - Utilidades en línea: PinMux, GUI composer
 - <https://dev.ti.com/>
 - Necesidad de darse de alta en TI.



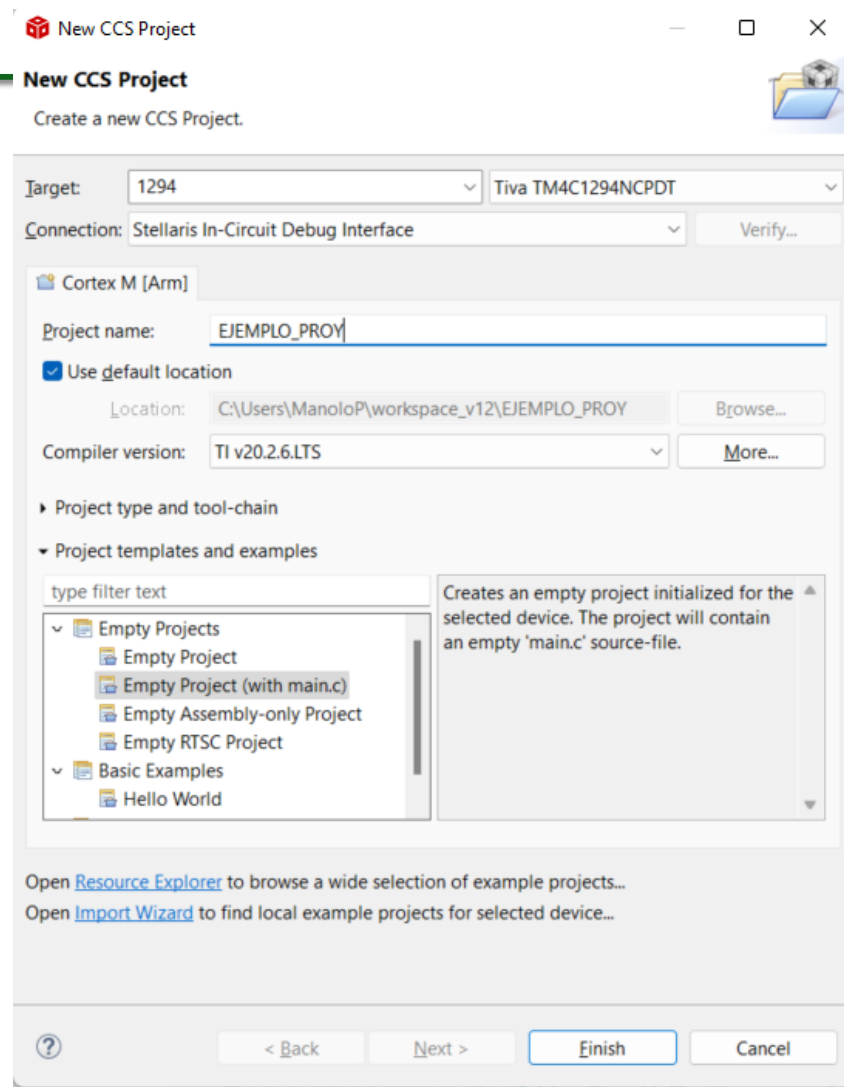
Creación de proyectos con CCS

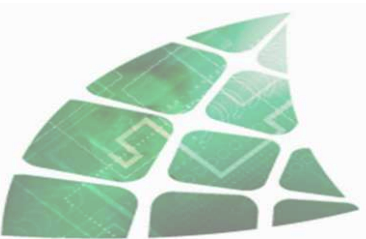
- A diferencia de los proyectos con MSP430, múltiples ficheros de gran tamaño
- Recursos *enlazados*, no copiados
 - Librerías de TivaWare
 - Librería DriverLib
- Problema para la *portabilidad*
 - Variables de entorno
 - Proyecto tipo y copia de ficheros (main.c, etc)



Creación de proyectos con CCS

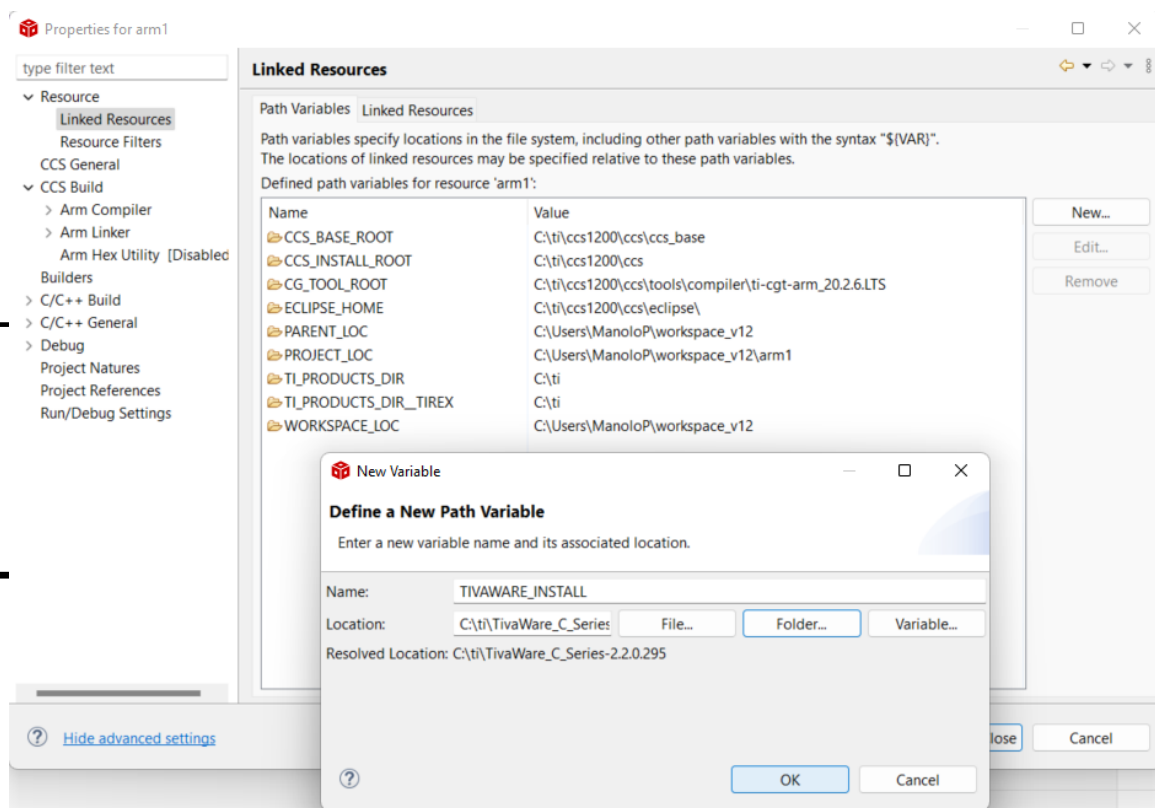
- 1. Creación de un proyecto vacío:
 - Familia: ARM
 - Variant: TIVA
TM4C1294NCPDT
 - Connection: Stellaris In-Circuit Debug Interface

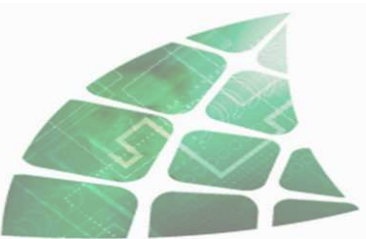




Creación de proyectos con CCS

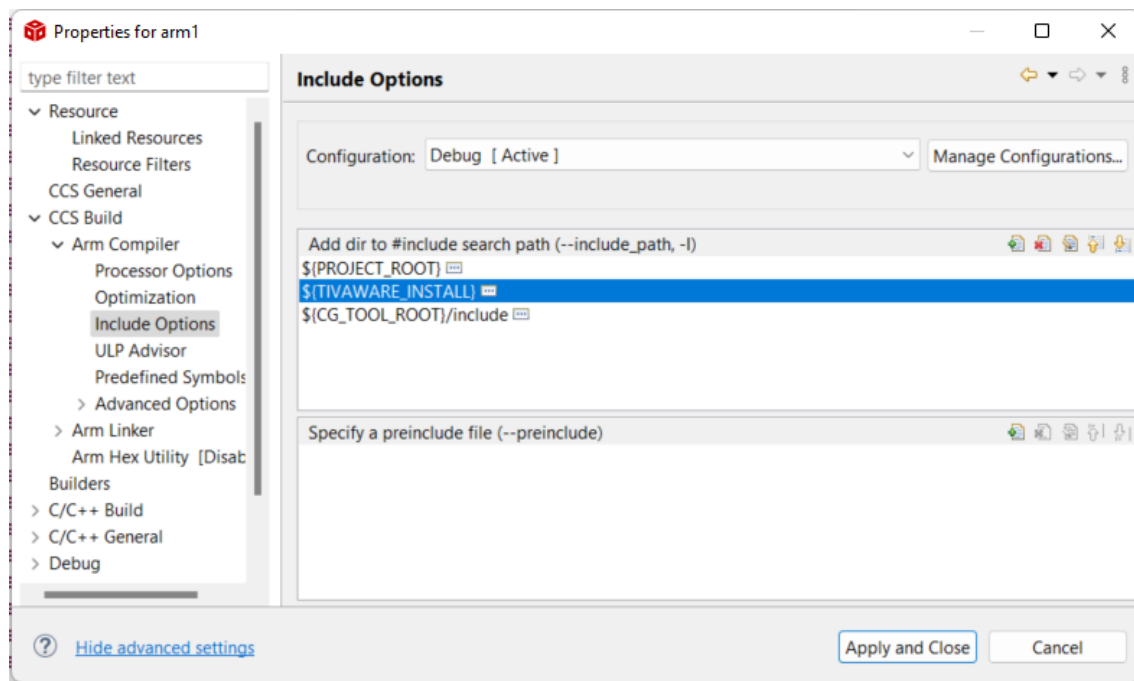
- 2. Variable de entorno:
 - En Resource-> Linked Resources -> Path Variables
 - Añadir **TIVAWARE_INSTALL**
 - Hacer que *apunte* al directorio de instalación de TivaWare

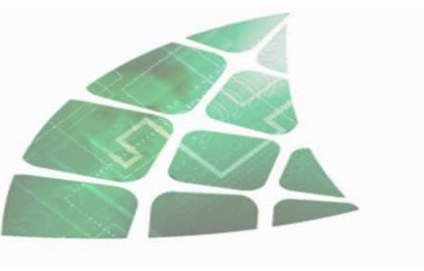




Creación de proyectos con CCS

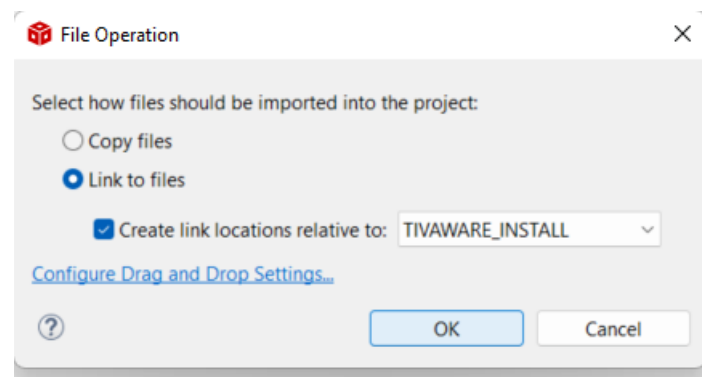
- 3. Añadir librerías al path:
 - En CCS Build
 - >ARM Compiler
 - >Include Options-
 - >Add dir to include search path
 - Añadir “variable”
`$(TIVAWARE_INSTALL)`

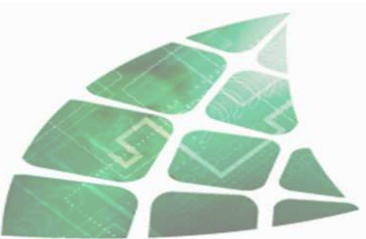




Creación de proyectos con CCS

- 4. Añadir DRIVERLIB:
 - Para facilitar la portabilidad, enlazarla a partir de la variable TIVAWARE_INSTALL:
 - Add files (buscar driverlib.lib)
 - C:\ti\TivaWare_C_Series-2.2.0.295\driverlib\ccs\Debug

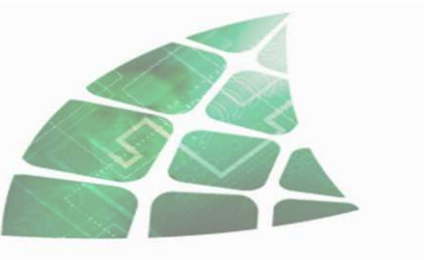




Creación de proyectos con CCS

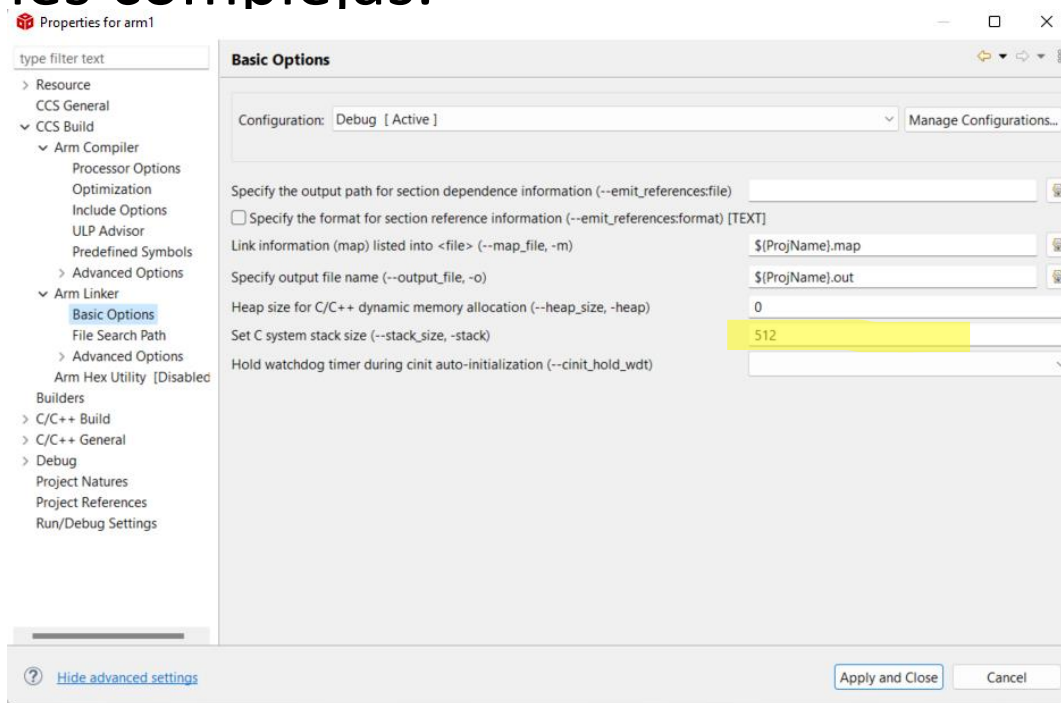
- 4'. (OPCIONAL) Añadir driverlib2.h:
 - Fichero de cabecera *propio* que enlaza todos los ficheros .h necesarios o no.
 - Descargar de página de ejemplos.
 - Copiar al proyecto (no enlazar)

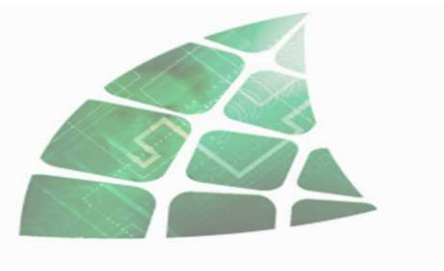
```
1
2//INCLUIR TODAS LAS LIBRERÍAS de DRIVERLIB
3#ifndef DRIVERLIB_H_
4#define DRIVERLIB_H_
5
6#include "driverlib/adc.h"
7#include "driverlib/aes.h"
8#include "driverlib/can.h"
9#include "driverlib/comp.h"
10#include "driverlib/cpu.h"
11#include "driverlib/crc.h"
12#include "driverlib/debug.h"
13#include "driverlib/des.h"
14#include "driverlib/eeeprom.h"
15#include "driverlib/emacs.h"
16#include "driverlib/epi.h"
17#include "driverlib/flash.h"
18#include "driverlib/fpu.h"
19#include "driverlib/gpio.h"
20//#include "driverlib/hibernate.h"
21#include "driverlib/i2c.h"
22#include "driverlib/interrupt.h"
23#include "driverlib/lcd.h"
24#include "driverlib/mpu.h"
25#include "driverlib/onewire.h"
26#include "driverlib/pin_map.h"
27#include "driverlib/pwm.h"
28#include "driverlib/qei.h"
29#include "driverlib/rtc.h"
```



Creación de proyectos con CCS

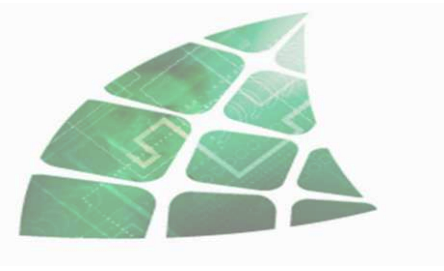
- 5. Aumentar el tamaño de la pila
 - Por defecto, 512 bytes. Muy pequeña para manejar funciones complejas.
 - Subir a 4096





Recursos y bibliografía

- WorkShop de Texas Instruments:
 - http://processors.wiki.ti.com/index.php/Creating_IoT_Solutions_with_the_TM4C1294XL_Connected_LaunchPad_Workshop (Videos, explicaciones, ejemplos...)
- Manual del Microcontrolador (1890 páginas):
 - <http://www.ti.com/lit/ds/symlink/tm4c1294ncpdt.pdf>
 - Descripción detallada de cada periférico y características
- Manual de la DriverLib (704 páginas):
 - <http://www.ti.com/lit/ug/spmu298d/spmu298d.pdf>
 - Definición de las funciones de la API.



Ejemplo 1

- Realizar un programa que encienda y apague alternativamente los led situados en los pines 0 y 1 del puerto N, cada 500ms
- Configurar el reloj a 120MHz (con un cristal externo de 25MHz)
- [Cargar el ejemplo 0 y modificarlo]
 - Cambiar retraso de 100ms a 500ms
 - Añadir la configuración del Pin 1 (por analogía)
 - Encender y apagar D1 y D0