

# USO DE RIOT-OS PARA PROGRAMACIÓN EN EL ÁMBITO DE IOT

María Ariza Gamero  
Alejandro Hernán Luque  
Alberto Vázquez Baeza

# Índice

- 1.- Presentación
- 2.- Git
- 3.- Hardware
- 4.- Docker
- 5.- RIOT-OS
- 6.- Demo
- 7.- Comunicaciones
- 8.- Conclusiones y futuras mejoras

# 1. – Presentación

# RIOT



Instalar sistema operativo RIOT  
en placas de Texas Instruments

Objetivos

- LEDS
- Botones
- Comunicaciones

## 2. - Git

Control de versiones  
Trabajo en paralelo  
Trabajo desde terminal

```
Alejandro@DESKTOP-MBMJ0E0 MINGW64 ~/Desktop/pruebas git/rito/4.-Memoria proyecto
/ing (master)
$ ls
cc2538.png cc2650.png git.png logoriot.png

Alejandro@DESKTOP-MBMJ0E0 MINGW64 ~/Desktop/pruebas git/rito/4.-Memoria proyecto
/ing (master)
$ cd ..

Alejandro@DESKTOP-MBMJ0E0 MINGW64 ~/Desktop/pruebas git/rito/4.-Memoria proyecto
(master)
$ cd ..

Alejandro@DESKTOP-MBMJ0E0 MINGW64 ~/Desktop/pruebas git/rito (master)
$ git pull origin master
From https://github.com/blalebla/IARS
* branch      master      -> FETCH_HEAD
Already up to date.

Alejandro@DESKTOP-MBMJ0E0 MINGW64 ~/Desktop/pruebas git/rito (master)
$
```

rito: All files - gitk

File Edit View Help

master remotes/origin/master otro

- mejora archivo .md
- Update README.md
- readme.md y reestructuración carpeta
- corregida una falta ortográfica en una guía y añadidos algunos archivos que se habían generado al c
- Merge branch 'master' of https://github.com/blalebla/IARS
- cambio estilo: carpeta proyecto, eliminac archivo sin uso
- más cambios de estilo
- Corrección faltas
- Actualización de la presentación del proyecto
- Avance de la memoria
- Merge branch 'master' of https://github.com/blalebla/IARS
- Avance de la memoria
- actualizados los readme y añadidos comentarios a los códigos que aún no los tenían. La parte
- versión original de RIOT, clonada en nuestro git
- versión original de RIOT, clonada en nuestro git
- versión original de RIOT, clonada en nuestro git
- versión original de RIOT, clonada en nuestro git
- v1Memoria
- prueba de subida maria
- actualización guia git
- v1 guia inicio git
- cambiado de hubicación los archivos
- actulización de carpeta con readme.md
- archivos test leds
- Create guia GIT
- añadir el archivo main
- archivos bin, hex y elf placa cc2538
- Create Tutorials.md
- Update README.md
- Update README.md
- Update README.md
- Update README.md

## Comandos Básicos:

- Git init
- Git add
- Git commit
- Git push
- Git pull



### 3. - Hardware

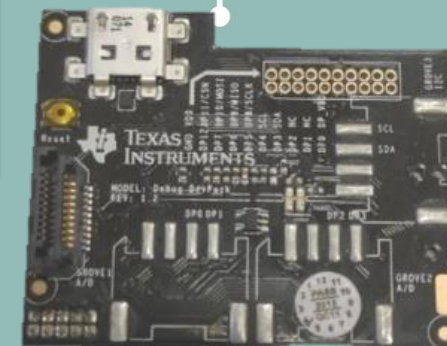
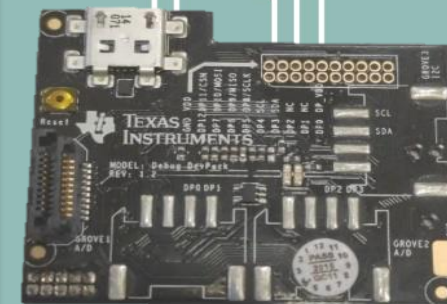
#### cc2538:

- ARM Cortex-M3
- 2,4 GHz IEEE 802.15.4
- 4 LEDs
- 5 Botones
- Sensor temperatura
- Placa de evaluación Smart RF06
- Programador integrado gracias a la placa de expansión

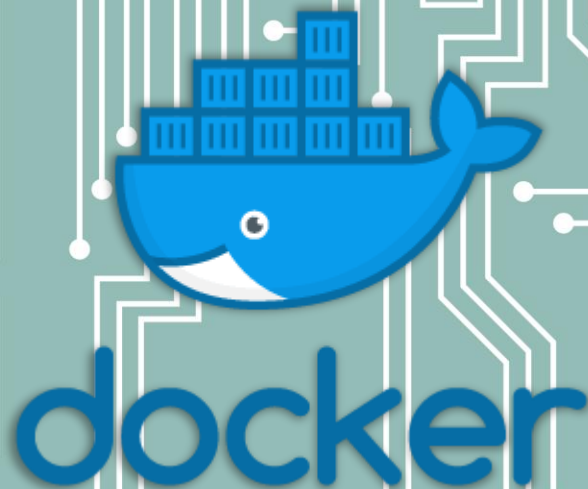


#### cc2650:

- ARM Cortex-M3
- BLE
- IEEE 802.15.4
- 2 LEDs
- 2 Botones
- Sensor de temperatura
- Acelerómetro
- Necesario programador XDS1100v3



## 4. – Docker



Comando más utilizado:

- `docker run -ti --rm --name RIOT -v $PWD/CarpetaMiPc:/CarpetaRiot riot/riotbuild`

- Virtualización a nivel de Sistema Operativo
- Aislamiento de recursos a nivel de kernel
- Flexibilidad y portabilidad
- Enfocado a sistemas altamente distribuidos

### Docker Pull Command

```
docker pull riot/riotbuild
```



## 5. - RIOT-OS

Sistema operativo friendly enfocado en IOT

- Programación estándar en C/C++
- Puerto nativo si se desarrolla sobre linux



Comparison of Current Operating Systems

OS	Min RAM	Min ROM	C Support	C++ Support	Multi-Threading	MCU w/o MMU	Modularity	Real-Time
Contiki	< 2kB	< 30kB	●	×	●	✓	●	●
Tiny OS	< 1kB	< 4KB	×	×	●	✓	×	×
Linux	~ 1MB	~ 1MB	✓	✓	✓	×	●	●
RIOT	~ 1.5kB	~ 5kB	✓	✓	✓	✓	✓	✓

Full support ✓  
Partial support ●  
No support ×

## 6. – Demo

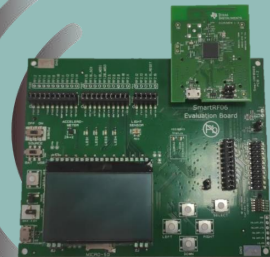
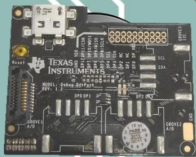




## 7.- Comunicaciones



Enciende led



## 8.- Conclusiones y futuras mejoras



docker

R.IOT



TEXAS  
INSTRUMENTS

SmartRF™Flash Programmer 2 ver. 1.5.0



Futuras mejoras:

- Automatizar encendido del led en función de una lectura del sensor de temperatura o luz
- Encender climatizadora en función de la temperatura