

# RTEMS su Raspberry Pi per applicazioni real-time

RELATORE: Prof. Domenico Giorgio Sorrenti

CO-RELATORE: Ing. Fabrizio Bernardini, FBIS

TUTOR AZIENDALE: Prof. Pietro Braione

Relazione della prova finale di:  
Clark Ezpeleta  
Matr. 832972

# Obiettivi e finalità

- Porting di RTEMS su Raspberry Pi.
- Creazione degli eseguibili RTEMS di validazione per le interfacce GPIO, UART, SPI e I2C.

# RTEMS

RTEMS sta per Real-Time Executive Multiprocessor System



# Raspberry Pi



# Porting di RTEMS su Raspberry Pi

1. Installazione dalla tool-suite sul computer host
2. Verifica del corretto funzionamento della tool-suite
3. Configurazione del IDE Eclipse C
4. Compilazione ed esecuzione dei programmi

# Interfacce I/O da validare

Le interfacce che ci interessano validare sono: GPIO, UART, I2C e SPI.

# Struttura programma di validazione RTEMS

# UART

La validazione viene effettuata creando un programma 'hello world'.

```
RTEMS RPi 3B+ 1.3 (1GB) [00a020d3]

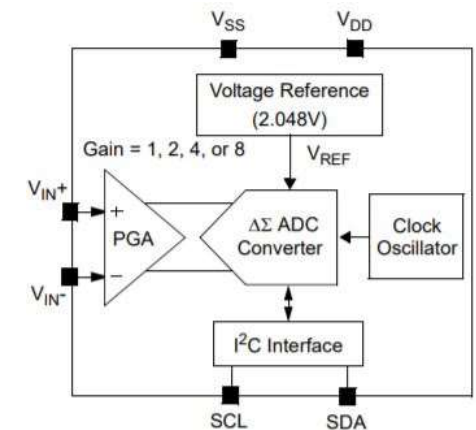
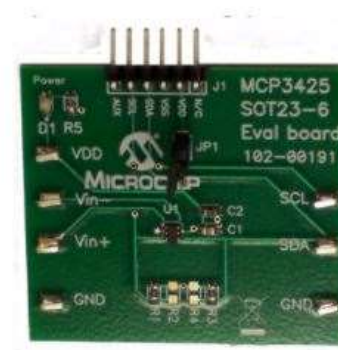
*** HELLO WORLD TEST ***
Hello World
*** END OF HELLO WORLD TEST ***

*** FATAL ***
fatal source: 5 (RTEMS_FATAL_SOURCE_EXIT)
fatal code: 0 (0x00000000)
RTEMS version: 5.0.0.61ccb9c05dcd695114541960aa6bfc1315f30514
RTEMS tools: 7.5.0 20191114 (RTEMS 5, RSB 5 (46e9d4911f09 modified), Newlib 7947581)
executing thread ID: 0x08a010001
executing thread nam
```



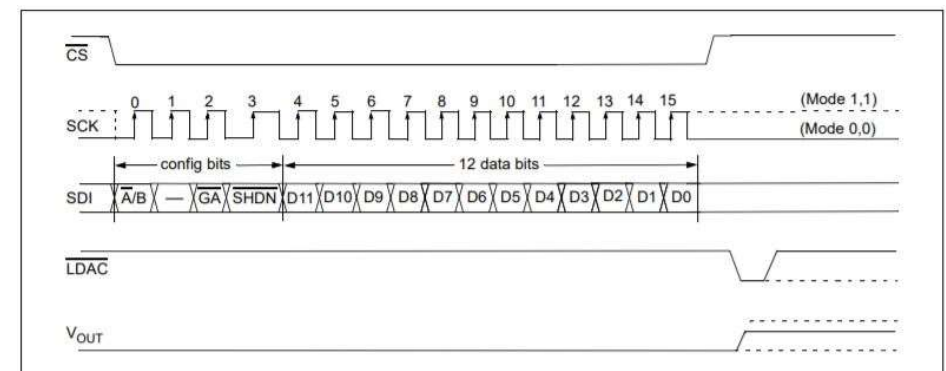
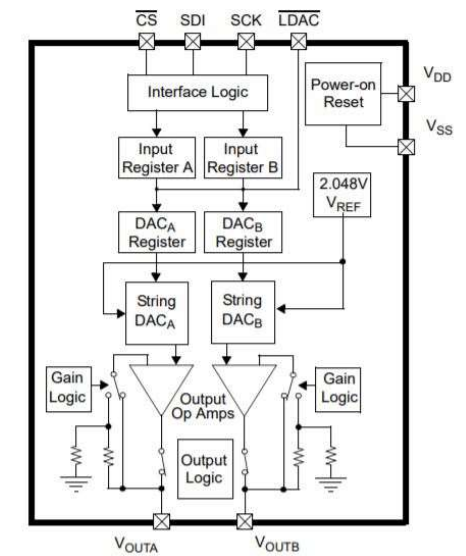
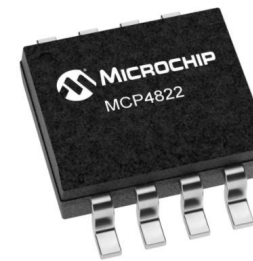
# I2C

Per l'interfaccia I2C viene utilizzato il componente aggiuntivo MCP3425.



Conversion Option	Digital Output Codes
16-bits	D15 ~ D8 (1st data byte) - D7 ~ D0 (2nd data byte) - Configuration byte. <b>(Note 1)</b>
14-bits	MMD13 ~ D8 (1st data byte) - D7 ~ D0 (2nd data byte) - Configuration byte. <b>(Note 2)</b>
12-bits	MMMMD11 ~ D8 (1st data byte) - D7 ~ D0 (2nd data byte) - Configuration byte. <b>(Note 3)</b>

# SPI



# Compilazione, caricamento, esecuzione e verifica

- Compilazione: viene effettuata utilizzando RTEMS gcc cross compiler
- Caricamento: bisogna copiare l'eseguibile sulla microSD della Raspberry Pi, e modificare il nome del kernel nel file 'config.txt'
- Esecuzione: all'accensione della Raspberry Pi il bootloader carica il kernel file indicato sul config.txt
- Verifica: viene effettuata controllando i log sul terminale tramite interfaccia UART ed utilizzando il multimetro e l'oscilloscopio.

# Risultati

- Porting effettuato con successo ed è stata creata una guida con descritto il procedimento in dettaglio
- Creati applicativi RTEMS di validazione delle interfacce
- Validate le interfacce prefissate tranne SPI

# Grazie per l'attenzione