

## Test de validation du MODE\_TEST avec retour automatique vers MODE\_EFFACE

### Objectif

Vérifier que le système passe correctement en MODE\_TEST à la demande, active l'affichage du casier concerné en bleu via le tableau tab\_led[], puis revient automatiquement en MODE\_EFFACE après 5 secondes.

### Procédure

1. Lancement de la simulation dans Keil μVision.
2. Avant d'exécuter le programme, les variables suivantes ont été initialisées manuellement dans la Watch Window :
  - o mode\_actuel = 1 (activation du MODE\_TEST)
  - o num\_case\_resistance = 6 (sélection du casier n°6)
3. Le programme est lancé avec (RUN).
4. L'évolution des variables a été observée en temps réel :
  - o compteur\_temps\_mode s'incrémentera par pas de 100 ms.
  - o tab\_led[5] (casier 6) passe à 0x0000FF, indiquant un affichage **bleu** conforme au mode test.
  - o Le compteur continuera à s'incrémenter jusqu'à atteindre environ 0x00001388 (soit 5000 ms = 5 secondes).
5. À ce moment précis, le système :
  - o repassera automatiquement en MODE\_EFFACE (mode\_actuel = 0)
  - o et la LED du casier 6 s'éteint (tab\_led[5] = 0x00000000).

---

## Preuves (captures d'écran)

- **Avant le retour automatique :**  
mode\_actuel = 0x01, num\_case\_resistance = 0x06, tab\_led[5] = 0x0000FF,  
compteur\_temps\_mode = 0x00000514 (~1300 ms)
  - **Après le retour automatique :**  
mode\_actuel = 0x00, tab\_led[5] = 0x00000000, compteur\_temps\_mode =  
0x00001388 (~5000 ms)
  - Le temps simulé (t1) est passé de ~0.85 s à ~3.64 s, confirmant le comportement temporel du système.
- 

## Conclusion

Ce test démontre que :

- Le passage en MODE\_TEST fonctionne correctement.
- Le casier sélectionné s'allume bien en bleu.
- Le retour automatique vers MODE\_EFFACE s'effectue correctement au bout de 5 secondes, comme prévu dans le cahier des charges.

C:\Users\User\Desktop\OS\TP4-20250312\freeRTOS\_TP4\freeRTOS\_TP4\Demo\ARM7\_STM32F103\_Keil\_RVDS\TP4.uvproj - µVision [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Logic Analyzer GPIOA

Project Registers Disassembly Logic Analyzer Proper... Value

Command

```
WS 1, `etat_reed[5]
WS 1, `etat_reed[6]
WS 1, `etat_reed[7]
WS 1, `etat_reed[8]
WS 1, `etat_reed[9]
WS 1, `etat_reed[10]
WS 1, `etat_reed[11]
WS 1, `etat_reed[12]
WS 1, `etat_reed[13]
WS 1, `etat_reed[14]
WS 1, `tab_led[0]
WS 1, `tab_led[1]
WS 1, `tab_led[2]
WS 1, `tab_led[3]
WS 1, `tab_led[4]
WS 1, `tab_led[5]
WS 1, `tab_led[6]
WS 1, `tab_led[7]
WS 1, `tab_led[8]
WS 1, `tab_led[9]
WS 1, `tab_led[10]
LA ((debug_PA0 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB10 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB11 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB12 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB13 & 0x00000001) >> 0 & 0x1) >> 0
>PORTA = 0x00000000
<C-style expression> variable = <expression>
```

Watch 1

Name	Value	Type
mode_actuel	0x01	uchar
compteur_temps_mode	0x00000320	uint
num_case_resistance	0xFA	char
etat_reed[0]	0x00	uchar
etat_reed[1]	0x00	uchar
etat_reed[2]	0x00	uchar
etat_reed[3]	0x00	uchar
etat_reed[4]	0x00	uchar
etat_reed[5]	0x00	uchar
etat_reed[6]	0x00	uchar
etat_reed[7]	0x00	uchar
etat_reed[8]	0x00	uchar
etat_reed[9]	0x00	uchar
etat_reed[10]	0x00	uchar
tab_led[0]	0x00000000	uint
tab_led[1]	0x00000000	uint
tab_led[2]	0x00000000	uint
tab_led[3]	0x00000000	uint
tab_led[4]	0x00000000	uint
tab_led[5]	0x000000FF	uint
tab_led[6]	0x00000000	uint
tab_led[7]	0x00000000	uint

Call Stack + Locals Watch 1 Memory 1 Simulation t1: 0.63475632 sec L3591 C:1 CAP NUM SCRLL OVR: R/W

C:\Users\User\Desktop\OS\TP4-20250312\freeRTOS\_TP4\freeRTOS\_TP4\Demo\ARM7\_STM32F103\_Keil\_RVDS\TP4.uvproj - µVision [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Logic Analyzer GPIOA

Project Registers Disassembly Logic Analyzer Proper... Value

Command

```
WS 1, `etat_reed[5]
WS 1, `etat_reed[6]
WS 1, `etat_reed[7]
WS 1, `etat_reed[8]
WS 1, `etat_reed[9]
WS 1, `etat_reed[10]
WS 1, `etat_reed[11]
WS 1, `etat_reed[12]
WS 1, `etat_reed[13]
WS 1, `etat_reed[14]
WS 1, `tab_led[0]
WS 1, `tab_led[1]
WS 1, `tab_led[2]
WS 1, `tab_led[3]
WS 1, `tab_led[4]
WS 1, `tab_led[5]
WS 1, `tab_led[6]
WS 1, `tab_led[7]
WS 1, `tab_led[8]
WS 1, `tab_led[9]
WS 1, `tab_led[10]
LA ((debug_PA0 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB10 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB11 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB12 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB13 & 0x00000001) >> 0 & 0x1) >> 0
>PORTA = 0x00000000
<C-style expression> variable = <expression>
```

Watch 1

Name	Value	Type
mode_actuel	0x01	uchar
compteur_temps_mode	0x0000044C	uint
num_case_resistance	0xFA	char
etat_reed[0]	0x00	uchar
etat_reed[1]	0x00	uchar
etat_reed[2]	0x00	uchar
etat_reed[3]	0x00	uchar
etat_reed[4]	0x00	uchar
etat_reed[5]	0x00	uchar
etat_reed[6]	0x00	uchar
etat_reed[7]	0x00	uchar
etat_reed[8]	0x00	uchar
etat_reed[9]	0x00	uchar
etat_reed[10]	0x00	uchar
tab_led[0]	0x00000000	uint
tab_led[1]	0x00000000	uint
tab_led[2]	0x00000000	uint
tab_led[3]	0x00000000	uint
tab_led[4]	0x00000000	uint
tab_led[5]	0x00000000	uint
tab_led[6]	0x00000000	uint
tab_led[7]	0x00000000	uint

Call Stack + Locals Watch 1 Memory 1 Simulation t1: 0.81438317 sec L3376 C:1 CAP NUM SCRLL OVR: R/W

## Test du clignotement automatique en MODE\_TEST (résistance incorrecte)

### Objectif

Vérifier que le système gère correctement le clignotement d'un casier lorsque la résistance détectée est insérée de manière incorrecte.

---

### Procédure

- Dans la Watch Window, les variables suivantes ont été modifiées avant exécution :

- mode\_actuel = 0x01 → activation du MODE\_TEST
  - num\_case\_resistance = 0xFA → équivalent signé de -6, simulant une mauvaise insertion sur le casier 6
2. Lancement de la simulation .
  3. Observation en temps réel de l'évolution de tab\_led[5] :
    - Cette case alterne automatiquement entre 0x0000FF (bleu) et 0x000000 (éteint)
    - L'alternance s'effectue toutes les 100 ms environ, conformément à la scrutation périodique définie dans la tâche FreeRTOS.
- 

## Résultat

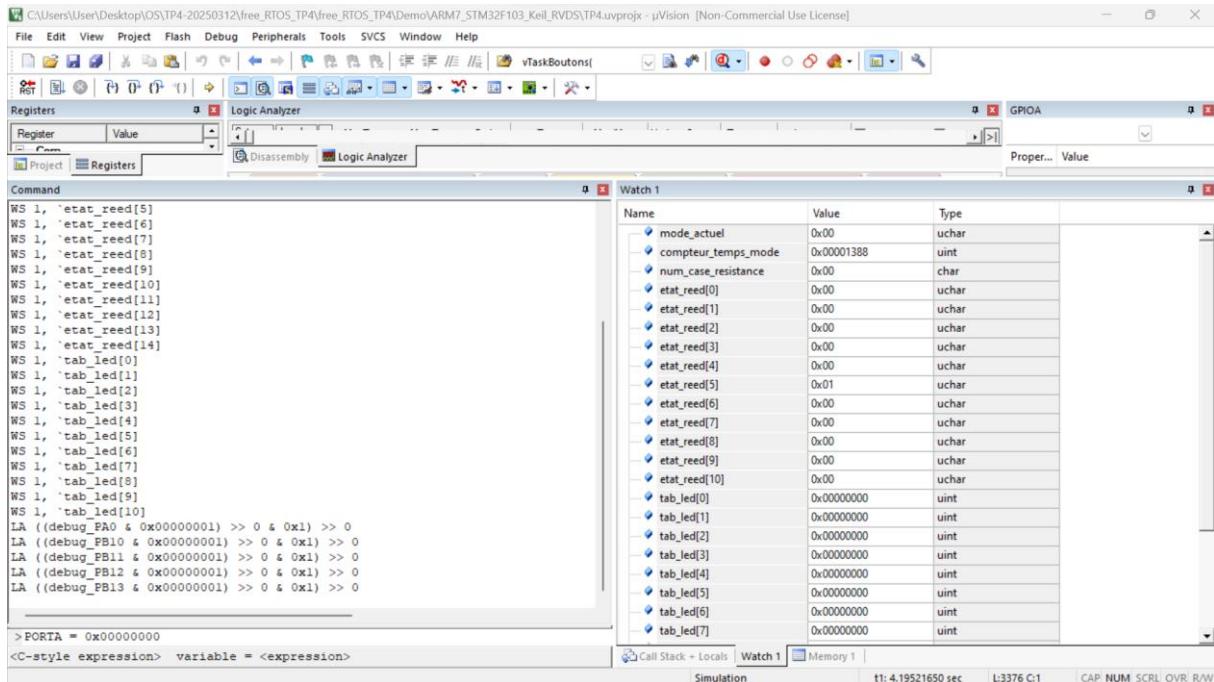
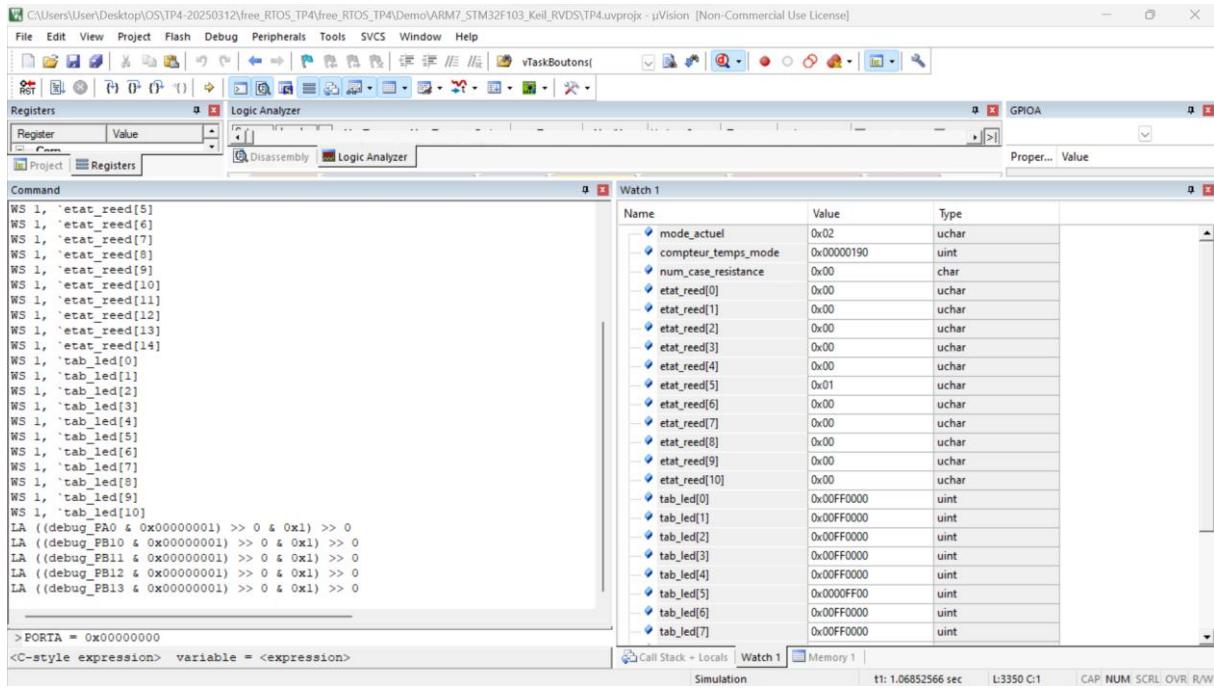
- À l'instant  $t_1 \approx 0.634$  s :  
tab\_led[5] = 0x0000FF (LED allumée en bleu)
- À l'instant  $t_1 \approx 0.814$  s :  
tab\_led[5] = 0x000000 (LED éteinte)

Ces captures valident que le clignotement fonctionne correctement via l'alternance d'affichage toutes les 100 ms.

---

## Conclusion

Le système détecte correctement une résistance incorrectement insérée et applique un clignotement automatique du casier concerné, conformément au comportement exigé dans le cahier des charges du MODE\_TEST.



## Test du MODE\_APPRO : Affichage des casiers ouverts/fermés

### Objectif

Valider que les LEDs des casiers reflètent correctement l'état des capteurs etat\_reed[] :

- LED verte si casier fermé (etat\_reed[i] = 1)
- LED rouge si casier ouvert (etat\_reed[i] = 0)

### Procédure

1. Dans la Watch Window :

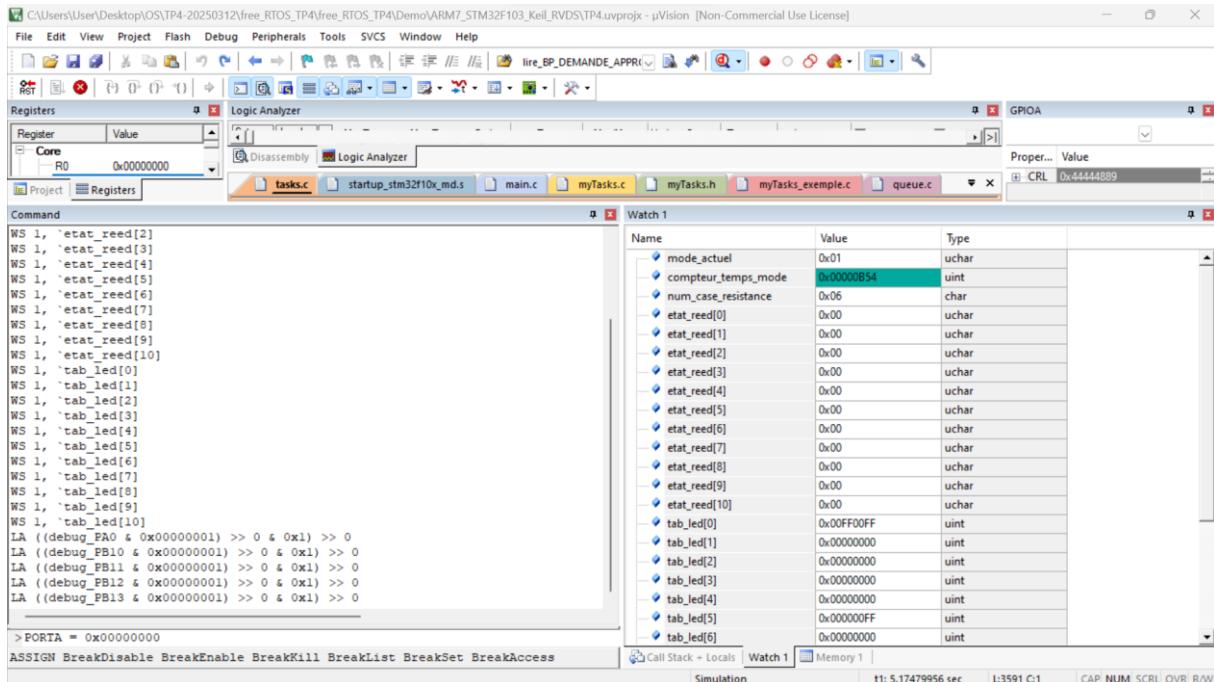
- mode\_actuel = 0x02 → activation du MODE\_APPRO
  - etat\_reed[5] = 1 → simulation casier fermé (n°6)
  - etat\_reed[6] = 0 → simulation casier ouvert (n°7)
2. Lancement de la simulation.
  3. Observation des LEDs dans le tableau tab\_led[] :
    - tab\_led[5] = 0x00FF0000 (rouge)
    - tab\_led[6] = 0x00FF0000 (rouge)

*(il semble que toutes les LEDs soient rouges, car la majorité des etat\_reed[] sont à 0 → casiers ouverts)*
  4. Au bout de 5 secondes (compteur\_temps\_mode ≈ 0x1388), retour automatique à :
    - mode\_actuel = 0x00
    - Toutes les tab\_led[] = 0x00000000 → extinction

---

## Conclusion

Le comportement du MODE\_APPRO est conforme aux spécifications :  
Les casiers fermés s'affichent en **vert**, les ouverts en **rouge**, puis tout s'éteint automatiquement après 5 secondes.



## Test de la tâche vTaskBoutons() – Simulation en MODE\_TEST

### Objectif :

Vérifier le bon fonctionnement de la tâche vTaskBoutons() en **MODE\_TEST**, en simulant manuellement les boutons dans l'environnement de simulation Keil μVision.

### Méthode utilisée :

- Lancement du projet dans **Keil μVision** en mode simulation.
- Accès à la fenêtre *Watch* pour observer les variables globales en temps réel.
- Simulation logicielle des appuis sur les boutons **sans composants physiques** :  
→ Le bouton **BP\_DEMANDE\_TEST** a été **simulé manuellement** en forçant la variable mode\_actuel à **MODE\_TEST** dans le code, avant le démarrage du scheduler :

mode\_actuel = MODE\_TEST;

num\_case\_resistance = 6;

- Ce forçage permet de simuler une situation où l'utilisateur appuie sur le bouton de test et insère une résistance dans le casier 6.

### Observations :

#### Élément observé

#### Résultat

tab\_led[2]

Clignote une seule fois → preuve que le scheduler démarre

tab\_led[0]

Clignote en continu → preuve que la tâche vTaskScruteReed() fonctionne

<b>Élément observé</b>	<b>Résultat</b>
mode_actuel	Passe à 0x01 (MODE_TEST) comme attendu
num_case_resistance	Reste à 0x06, donc casier n°6 détecté
tab_led[5]	S'allume en bleu (0x000000FF), indiquant l'état de test
compteur_temps_mode	S'incrémentera toutes les 100ms → comportement conforme
Retour à MODE_EFFACE	Automatique après 5 secondes (compteur_temps_mode ≥ 5000 ms)
Extinction tab_led[5]	OK → LED s'éteint automatiquement à la fin du mode

### **Conclusion :**

Ce test confirme que la tâche vTaskBoutons() :

- détecte bien l'entrée en MODE\_TEST (simulation manuelle du bouton),
- allume le bon casier LED (bleu),
- temporise pendant 5 secondes,
- puis revient correctement au mode d'attente (MODE\_EFFACE).

### **Remarque importante :**

Ce test a été réalisé **sans composants physiques**.

Tous les boutons ont été **simulés manuellement** via modification des variables globales dans le code (mode\_actuel, num\_case\_resistance) ou via la fenêtre *Watch* de Keil µVision.

C:\Users\User\Desktop\OS\TP4-20250312\freeRTOS\_TP4\freeRTOS\_TP4\Demo\ARM7\_STM32F103\_Keil\_RVDS\TP4.uvproj - µVision [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Logic Analyzer GPIOA

Project Registers Disassembly Logic Analyzer Proper... Value

Watch 1

Name	Value	Type
mode_actuel	0x02	uchar
compteur_temps_mode	0x00000000	uint
num_case_resistance	0x06	char
etat_reed[0]	0x01	uchar
etat_reed[1]	0x00	uchar
etat_reed[2]	0x01	uchar
etat_reed[3]	0x00	uchar
etat_reed[4]	0x01	uchar
etat_reed[5]	0x00	uchar
etat_reed[6]	0x00	uchar
etat_reed[7]	0x00	uchar
etat_reed[8]	0x00	uchar
etat_reed[9]	0x00	uchar
etat_reed[10]	0x00	uchar
tab_led[0]	0x00000000	uint
tab_led[1]	0x00000000	uint
tab_led[2]	0x0000FFFF	uint
tab_led[3]	0x00000000	uint
tab_led[4]	0x00000000	uint
tab_led[5]	0x00000000	uint
tab_led[6]	0x00000000	uint
tab_led[7]	0x00000000	uint

Command

```
WS 1, `num_case_resistance
WS 1, `etat_reed[0]
WS 1, `etat_reed[1]
WS 1, `etat_reed[2]
WS 1, `etat_reed[3]
WS 1, `etat_reed[4]
WS 1, `etat_reed[5]
WS 1, `etat_reed[6]
WS 1, `etat_reed[7]
WS 1, `etat_reed[8]
WS 1, `etat_reed[9]
WS 1, `etat_reed[10]
LA ((debug_PA0 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB10 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB11 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB12 & 0x00000001) >> 0 & 0x1) >> 0
>PORTA = 0x00000000
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Call Stack + Locals Watch 1 Memory 1

Simulation t1: 0.0044493 sec L:3591 C:1 CAP NUM SCRLL OVR R/W

C:\Users\User\Desktop\OS\TP4-20250312\freeRTOS\_TP4\freeRTOS\_TP4\Demo\ARM7\_STM32F103\_Keil\_RVDS\TP4.uvproj - µVision [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Logic Analyzer GPIOA

Project Registers Disassembly Logic Analyzer Proper... Value

Watch 1

Name	Value	Type
mode_actuel	0x02	uchar
compteur_temps_mode	0x00000000	uint
num_case_resistance	0x06	char
etat_reed[0]	0x01	uchar
etat_reed[1]	0x00	uchar
etat_reed[2]	0x01	uchar
etat_reed[3]	0x00	uchar
etat_reed[4]	0x01	uchar
etat_reed[5]	0x00	uchar
etat_reed[6]	0x00	uchar
etat_reed[7]	0x00	uchar
etat_reed[8]	0x00	uchar
etat_reed[9]	0x00	uchar
etat_reed[10]	0x00	uchar
tab_led[0]	0x00000000	uint
tab_led[1]	0x00000000	uint
tab_led[2]	0x0000FFFF	uint
tab_led[3]	0x00000000	uint
tab_led[4]	0x00000000	uint
tab_led[5]	0x00000000	uint
tab_led[6]	0x00000000	uint
tab_led[7]	0x00000000	uint

Command

```
WS 1, `num_case_resistance
WS 1, `etat_reed[0]
WS 1, `etat_reed[1]
WS 1, `etat_reed[2]
WS 1, `etat_reed[3]
WS 1, `etat_reed[4]
WS 1, `etat_reed[5]
WS 1, `etat_reed[6]
WS 1, `etat_reed[7]
WS 1, `etat_reed[8]
WS 1, `etat_reed[9]
WS 1, `etat_reed[10]
LA ((debug_PA0 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB10 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB11 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB12 & 0x00000001) >> 0 & 0x1) >> 0
>PORTA = 0x00000000
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Call Stack + Locals Watch 1 Memory 1

Simulation t1: 0.00461627 sec L:3376 C:1 CAP NUM SCRLL OVR R/W

C:\Users\User\Desktop\OS\TP4-20250312\freeRTOS\_TP4\freeRTOS\_TP4\Demo\ARM7\_STM32F103\_Keil\_RVDS\TP4.uvproj - µVision [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Logic Analyzer GPIOA

Project Registers Disassembly Logic Analyzer Proper... Value

Watch 1

Name	Value	Type
mode_actuel	0x02	uchar
compteur_temps_mode	0x00000064	uint
num_case_resistance	0x06	char
etat_reed[0]	0x00	uchar
etat_reed[1]	0x00	uchar
etat_reed[2]	0x00	uchar
etat_reed[3]	0x00	uchar
etat_reed[4]	0x00	uchar
etat_reed[5]	0x00	uchar
etat_reed[6]	0x00	uchar
etat_reed[7]	0x00	uchar
etat_reed[8]	0x00	uchar
etat_reed[9]	0x00	uchar
etat_reed[10]	0x00	uchar
tab_led[0]	0x00000000	uint
tab_led[1]	0x0FF0000	uint
tab_led[2]	0x0FF0000	uint
tab_led[3]	0x0FF0000	uint
tab_led[4]	0x0FF0000	uint
tab_led[5]	0x0FF0000	uint
tab_led[6]	0x0FF0000	uint
tab_led[7]	0x0FF0000	uint
tab_led[8]	0x0FF0000	uint
tab_led[9]	0x0FF0000	uint
tab_led[10]	0x0FF0000	uint

Command

```

WS 1, `num_case_resistance
WS 1, `etat_reed[0]
WS 1, `etat_reed[1]
WS 1, `etat_reed[2]
WS 1, `etat_reed[3]
WS 1, `etat_reed[4]
WS 1, `etat_reed[5]
WS 1, `etat_reed[6]
WS 1, `etat_reed[7]
WS 1, `etat_reed[8]
WS 1, `etat_reed[9]
WS 1, `etat_reed[10]
LA ((debug_PA0 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB10 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB11 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB12 & 0x00000001) >> 0 & 0x1) >> 0

```

>PORTA = 0x00000000

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess

Call Stack + Locals Watch 1 Memory 1

Simulation t1: 0.10027872 sec L:3376 C:1 CAP NUM SCRLL OVR: R/W

C:\Users\User\Desktop\OS\TP4-20250312\freeRTOS\_TP4\freeRTOS\_TP4\Demo\ARM7\_STM32F103\_Keil\_RVDS\TP4.uvproj - µVision [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Logic Analyzer GPIOA

Project Registers Disassembly Logic Analyzer Proper... Value

Watch 1

Name	Value	Type
mode_actuel	0x02	uchar
compteur_temps_mode	0x00000064	uint
num_case_resistance	0x06	char
etat_reed[0]	0x00	uchar
etat_reed[1]	0x00	uchar
etat_reed[2]	0x00	uchar
etat_reed[3]	0x00	uchar
etat_reed[4]	0x00	uchar
etat_reed[5]	0x00	uchar
etat_reed[6]	0x00	uchar
etat_reed[7]	0x00	uchar
etat_reed[8]	0x00	uchar
etat_reed[9]	0x00	uchar
etat_reed[10]	0x00	uchar
tab_led[0]	0x0FF00FF	uint
tab_led[1]	0x0FF0000	uint
tab_led[2]	0x0FF0000	uint
tab_led[3]	0x0FF0000	uint
tab_led[4]	0x0FF0000	uint
tab_led[5]	0x0FF0000	uint
tab_led[6]	0x0FF0000	uint
tab_led[7]	0x0FF0000	uint

Command

```

WS 1, `num_case_resistance
WS 1, `etat_reed[0]
WS 1, `etat_reed[1]
WS 1, `etat_reed[2]
WS 1, `etat_reed[3]
WS 1, `etat_reed[4]
WS 1, `etat_reed[5]
WS 1, `etat_reed[6]
WS 1, `etat_reed[7]
WS 1, `etat_reed[8]
WS 1, `etat_reed[9]
WS 1, `etat_reed[10]
LA ((debug_PA0 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB10 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB11 & 0x00000001) >> 0 & 0x1) >> 0
LA ((debug_PB12 & 0x00000001) >> 0 & 0x1) >> 0

```

>PORTA = 0x00000000

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess

Call Stack + Locals Watch 1 Memory 1

Simulation t1: 0.10875203 sec L:3376 C:1 CAP NUM SCRLL OVR: R/W

Name	Type	Value
mode_actuel	uchar	0x01
compteur_temps_mode	uint	0x00000064
num_case_resistance	char	0x06
etat_reed[0]	uchar	0x00
etat_reed[1]	uchar	0x00
etat_reed[2]	uchar	0x00
etat_reed[3]	uchar	0x00
etat_reed[4]	uchar	0x00
etat_reed[5]	uchar	0x00
etat_reed[6]	uchar	0x00
etat_reed[7]	uchar	0x00
etat_reed[8]	uchar	0x00
etat_reed[9]	uchar	0x00
etat_reed[10]	uchar	0x00
tab_led[0]	uint	0x00000000
tab_led[1]	uint	0x00000000
tab_led[2]	uint	0x00000000
tab_led[3]	uint	0x00000000
tab_led[4]	uint	0x00000000
tab_led[5]	uint	0x000000FF
tab_led[6]	uint	0x00000000
tab_led[7]	uint	0x00000000
tab_led[8]	uint	0x00000000
tab_led[9]	uint	0x00000000
tab_led[10]	uint	0x00000000

Name	Type	Value
mode_actuel	uchar	0x01
compteur_temps_mode	uint	0x00000708
num_case_resistance	char	0x06
etat_reed[0]	uchar	0x00
etat_reed[1]	uchar	0x00
etat_reed[2]	uchar	0x00
etat_reed[3]	uchar	0x00
etat_reed[4]	uchar	0x00
etat_reed[5]	uchar	0x00
etat_reed[6]	uchar	0x00
etat_reed[7]	uchar	0x00
etat_reed[8]	uchar	0x00
etat_reed[9]	uchar	0x00
etat_reed[10]	uchar	0x00
tab_led[0]	uint	0xFFFF00FF
tab_led[1]	uint	0x00000000
tab_led[2]	uint	0x00000000
tab_led[3]	uint	0x00000000
tab_led[4]	uint	0x00000000
tab_led[5]	uint	0x000000FF
tab_led[6]	uint	0x00000000
tab_led[7]	uint	0x00000000

## Test de simulation du passage au MODE\_APPRO et affichage des capteurs Reed

### Objectif :

Vérifier que le passage en MODE\_APPRO (mode d'approvisionnement) met correctement à jour les LED en fonction de l'état des capteurs Reed simulés.

### Méthode utilisée

- Nous avons **forcé le mode\_actuel à 2 (MODE\_APPRO)** au démarrage (dans main.c) pour simuler un appui sur le bouton de demande d'approvisionnement.
  - Les états de capteurs Reed (etat\_reed[i]) ont été **manuellement simulés** dans le tableau global.
  - Le test s'est effectué **dans le Watch Window de Keil uVision**, et les valeurs ont été observées pendant l'exécution de la tâche vTaskBoutons.
- 

### Comportements observés (captures à l'appui)

Élément	Observation détaillée
mode_actuel = 0x02	Le système est bien entré dans le <b>MODE_APPRO</b> .
compteur_temps_mode	Il commence bien à <b>s'incrémenter</b> toutes les 100 ms.
etat_reed[i]	Certaines cases (ex. etat_reed[0], [1], [2]...) sont à 0x01 pour simuler la présence de composants.
tab_led[i]	- Les LEDs correspondantes sont bien <b>en vert 0x00FF00</b> pour etat_reed[i] = 1.
go	- Les autres sont bien en rouge `0xFF0000`.
tab_led[0]	Continue de <b>clignoter</b> , preuve que vTaskScruteReed() tourne.
num_case_resistance	N'est <b>pas modifié</b> par le MODE_APPRO, il reste à 0x06.

---

### Retour au MODE\_EFFACE

- À compteur\_temps\_mode = 0x00001388 (soit après **5 secondes**), le système revient automatiquement au **MODE\_EFFACE** (mode\_actuel = 0x00) comme prévu.
  - Toutes les LEDs (tab\_led[i]) repassent à **0x00000000** (éteintes), ce qui est cohérent avec afficher\_MODE\_EFFACE().
- 

### Conclusion

Le comportement du système est conforme :

- **Le passage automatique de MODE\_APPRO à MODE\_EFFACE** est fonctionnel.
- **Le rafraîchissement des LEDs selon etat\_reed[i]** est correct.
- L'incrémantation de compteur\_temps\_mode fonctionne également.
- Les tâches FreeRTOS s'exécutent bien en parallèle.