PROT: SMPU_Switching

- **Description**:
  - This example assumes that the system assigns memory areas to two tasks - taskA, and taskB using SMPU and SMPU_STRUCTUREs 1 / 2. The system changes its core’s PC to “4” before entering taskA, and “5” before entering taskB.
  - Global variable array “taskA_Region0” and “taskA_Region1” are the RAM memory which can be accessed only by task A (“Routine_TaskA”) (or only by master who has a PC of “4”).
  - Global variable array “taskB_Region0” is the RAM memory which can be accessed only by taskB (“Routine_TaskB”) (or only by master who has a PC of “5”).
  - In normal operation, “taskB_Region0” is accessed in “Routine_TaskB”, “taskA_Region0” and “taskA_Region1” are accessed in “Routine_TaskA”.
  - User can infuse bugs intentionally by changing “INJECT_BUG_MODE”, e.g. if INJECT_BUG_MODE = INJECT_BUG_TO_TASK_A, “Routine_TaskA” will access memory of “taskB_Region0” and will cause bus fault error (hardfault).

- **Target Device**:
  - Traveo-II CYT2Bx devices

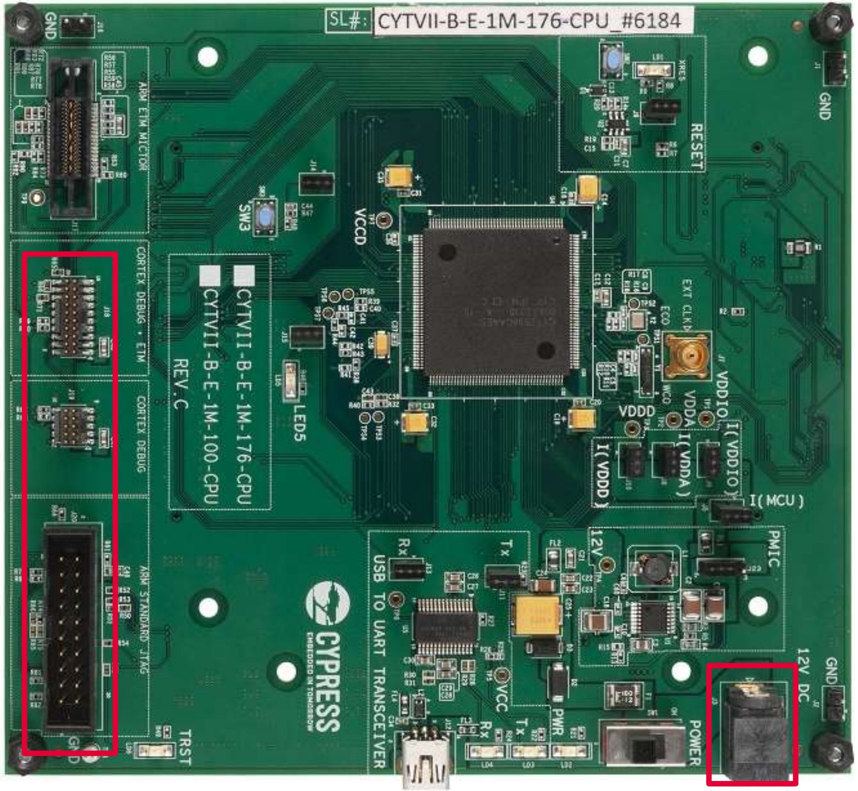
- **CPU Board**:
  - CYTVII-B-E-1M-176-CPU Rev. C Board

PROT: SMPU_Switching

- **Dependency:**
  - None

- **Expectation**:
  - INJECT_BUG_MODE = INJECT_BUG_TO_NONE
    - Nothing happens. Program will continue increasing value of variable “taskA_Region0”, “taskA_Region1”, “taskB_Region2”.
  - INJECT_BUG_MODE = INJECT_BUG_TO_TASK_A
    - Will cause bus fault in “Routine_TaskA” because of accessing taskB memory.
  - INJECT_BUG_MODE = INJECT_BUG_TO_TASK_B_0
    - Will cause bus fault in “Routine_TaskB” because of accessing taskA memory region 0.
  - INJECT_BUG_MODE = INJECT_BUG_TO_TASK_B_1
    - Will cause bus fault in “Routine_TaskB” because of accessing taskA memory region 1.

PROT: SMPU_Switching



Legend:

- Red block for power, debug (Mandatory)