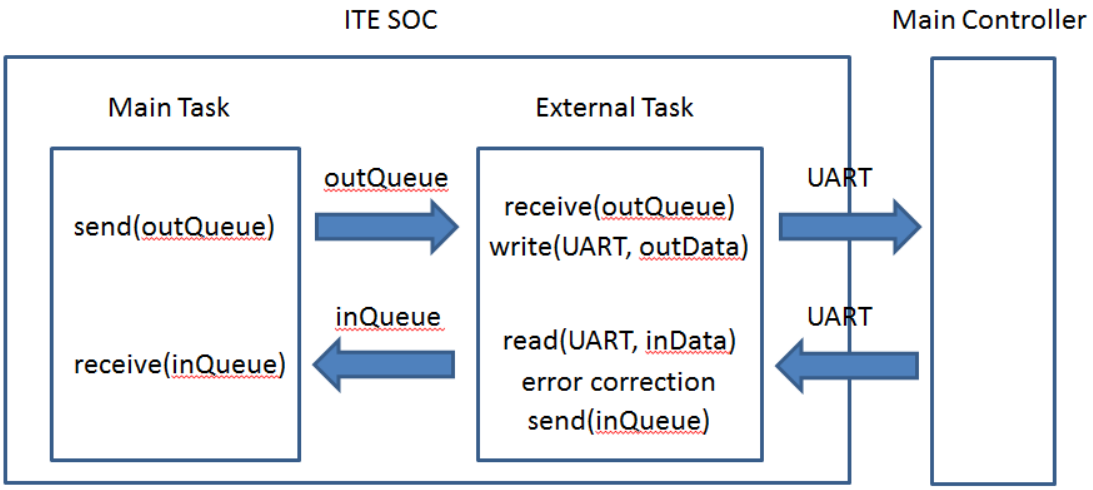
external 使用說明



用途
做為主控與 ITE SOC 之間資料溝通。

說明
SDK 提供的 external.c 以 UART 當作參考範例，包含了一份簡單的 error correction 動作，避免從 UART 收到的資料偏移，這部份需依自定義 payload 與實際使用情況做修改。

當 ITE SOC 有資料要往主控傳送時，從 Main Task 將資料透過 outQueue 傳遞到 External Task，External Task 接收 outQueue 裡面的資料，再呼叫 write(UART, outData) 經由 UART 傳送至主控端。

相反的，當主控端有資料經由 UART 傳送至 ITE SOC 時，先在 External Task 呼叫 read(UART, inData)接收，並且經過 error correction 的處理，確定收集到完整的資料再進行 parsing 解析，最後透過 inQueue 傳送至 Main Task 做出相對應的動作。

queue 解釋
這裡使用的 queue 是 openrtos 提供，原始定義在 openrtos\openrtos\queue.c，為了讓上層使用，另外做了一層包裝在 sdk\driver\itp\itp_posix_openrtos.c，提供如 mq_open、mq_close、mq_send、mq_receive 等 API。

呼叫 mq_open 可建立 queue，這裡必須決定 queue 大小與 block 或 non-block 行為。queue 大小由底下兩個參數值指定
qattr.mq_msgsize: 每筆資料大小
qattr.mq_maxmsg: 單位是個數，表示 queue 最大可容納多少筆資料，需依 Main

Task 與 External Task 實際處理速度來決定該值大小
以 external.c 為例
qattr.mq_msgsize = sizeof(ExternalEvent): 代表一筆資料就是一個 ExternalEvent 結構
qattr.mq_maxmsg = EXT_MAX_QUEUE_SIZE: 最大可容納筆數，設定為 8

block 與 non-block 的行為，由 O_NONBLOCK 參數決定
block: 當 queue 裡的資料已經放滿，下一筆資料再進來就會被 block 卡住，等待資料被取走。另一種情況，當 queue 裡已經空掉無資料，也會 block 卡住，等待新資料送來。
non-block: 當 queue 裡的資料已經放滿，下一筆資料再進來會被丟棄。另一種情況，當 queue 裡已經空掉無資料，則不會卡住直接離開。