

# iTE Control Board

## 外部輸出入架構說明文件

修訂記錄

修訂日期	修訂說明	頁次
2016/11/17	初建版本	
2016/12/7	更新程式範例	

目錄

<b>1. 前言 .....</b>	<b>1</b>
1.1 編寫目的 .....	1
1.2 適用範圍 .....	1
1.3 適用人員 .....	1
<b>2. 外部輸出入架構介紹 .....</b>	<b>3</b>
2.1 ITE CONTROL BOARD 外部輸出入架構介紹 .....	3
2.2 外部輸入架構流程介紹 .....	3
2.3 外部輸出架構流程介紹 .....	6
2.4 訊息定義 .....	8

## 1. 前言

### 1.1 編寫目的

介紹iTE Control board 外部輸出入架構說明。

### 1.2 適用範圍

使用於iTE Control board相關產品。

### 1.3 適用人員

適用於使用iTE FAE or 客戶RD開發人員。

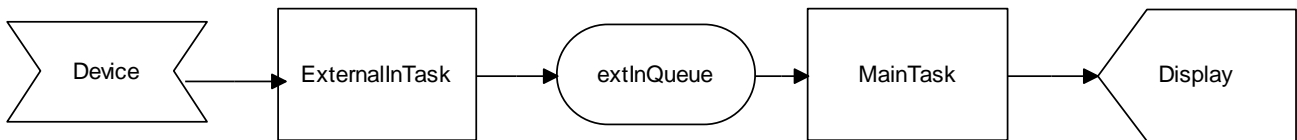


## 2. 外部輸出入架構介紹

### 2.1 iTE Control board 外部輸出入架構介紹

- 透過外部輸出入架構，從週邊讀取訊息，再傳遞到畫面，和透過畫面輸入，再傳遞到週邊，有統一的寫法，不至於讓畫面阻塞住。

### 2.2 外部輸入架構流程介紹



- 程式在ExternalInTask執行緒中不斷的讀取週邊的訊息，然後將訊息寫入extInQueue的訊息佇列中。程式碼檔案在external.c，範例代碼如下：

```

static void* ExternalInTask(void* arg)
{
    while (!extQuit)
    {
        ExternalEvent ev;
        char readbuf[4];

#ifdef CFG_UART0_ENABLE && !defined(CFG_DBG_UART0)
        if (read(ITP_DEVICE_UART0, readbuf, 4))
        {
            switch (readbuf[0])
            {
                case '0':
                    ev.type = EXTERNAL_TEST0;
                    break;

                case '1':
                    ev.type = EXTERNAL_TEST1;
                    ev.arg1 = readbuf[1];
                    break;

                case '2':
                    ev.type = EXTERNAL_TEST2;
                    ev.arg1 = readbuf[1];
                    ev.arg2 = readbuf[2];
                    break;

                case '3':
                    ev.type = EXTERNAL_TEST3;
                    ev.arg1 = readbuf[1];
                    ev.arg2 = readbuf[2];

```

```

        ev.arg3 = readbuf[3];
        break;

    case '4':
        ev.type = EXTERNAL_TEST4;
        ev.arg1 = readbuf[1];
        break;

    case '5':
        ev.type = EXTERNAL_TEST5;
        ev.arg1 = readbuf[1];
        break;

    default:
        ev.type = EXTERNAL_SHOW_MSG;
        memset(ev.buf1, 0 , EXTERNAL_BUFFER_SIZE);
        memcpy(ev.buf1, readbuf, 4);
        break;
    }
    mq_send(extInQueue, (const char*)&ev, sizeof (ExternalEvent), 0);
}
#endif
    usleep(10000);
}
mq_close(extInQueue);
extInQueue = -1;

return NULL;
}

```

新的讀取週邊訊息程式碼需要在此加入。

- 在主執行緒中不斷檢查extInQueue之中是不是有新的訊息。若有，則會讀出新訊息，交給ExternalProcessEvent() 處理，相關程式碼在scene.c，如下：

```

static void CheckExternal(void)
{
    ExternalEvent ev;
    int ret = ExternalReceive(&ev);

    if (ret)
    {
        ScreenSaverRefresh();
        ExternalProcessEvent(&ev);
    }
}

```

- 在ExternalProcessEvent()中依訊息型別做不同的處理，再更新畫面。程式碼在external\_process.c。範例如下：

```
static void ExternalProcessUart(ExternalEvent* ev)
{
    if (!ituWidgetIsVisible(buttonLayer))
        ituLayerGoto(buttonLayer); // goto button layer

    ituCoverFlowGoto(buttonCoverFlow, 2); // goto #3 page
    ituSceneSendEvent(&theScene, EVENT_CUSTOM_UART, ev->buf1); // send received UART data to UI
}

void ExternalProcessEvent(ExternalEvent* ev)
{
    char buf[64];
    assert(ev);

    switch (ev->type)
    {
    case EXTERNAL_SHOW_MSG:
        printf("EXTERNAL_SHOW_MSG\n");
        ExternalProcessUart(ev);
        break;

    case EXTERNAL_TEST0:
        printf("EXTERNAL_TEST0\n");
        ituSceneSendEvent(&theScene, EVENT_CUSTOM_KEY0, NULL);
        break;

    case EXTERNAL_TEST1:
        printf("EXTERNAL_TEST1: %d\n", ev->arg1);
        ituSceneSendEvent(&theScene, EVENT_CUSTOM_KEY1, NULL);
        break;

    case EXTERNAL_TEST2:
        printf("EXTERNAL_TEST2: %d %d\n", ev->arg1, ev->arg2);
        sprintf(buf, "%d %d", ev->arg1, ev->arg2);
        ituSceneSendEvent(&theScene, EVENT_CUSTOM_KEY2, buf);
        break;

    case EXTERNAL_TEST3:
        printf("EXTERNAL_TEST3: %d %d %d\n", ev->arg1, ev->arg2, ev->arg3);
        sprintf(buf, "%d %d %d", ev->arg1, ev->arg2, ev->arg3);
        ituSceneSendEvent(&theScene, EVENT_CUSTOM_KEY1, buf);
        break;

    case EXTERNAL_TEST4:
        printf("EXTERNAL_TEST4: %d\n", ev->arg1);
        break;
    }
```

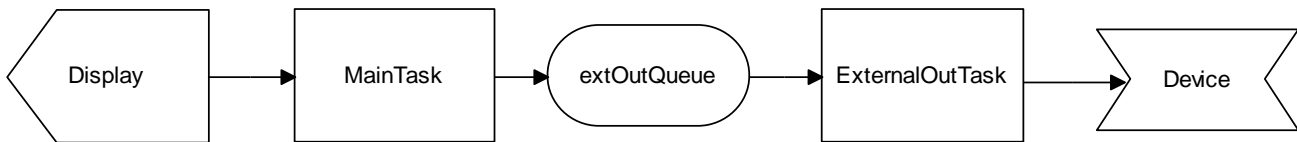


```

        case EXTERNAL_TEST5:
            printf("EXTERNAL_TEST5: %d\n", ev->arg1);
            break;
    }
}

```

## 2.3 外部輸出架構流程介紹



- 從畫面取得輸出訊息後，呼叫ExternalSend()將訊息寫入extOutQueue的訊息佇列中。範例程式碼如下：

```

// called when user press send button
bool ButtonUartSendButtonOnPress(ITUWidget* widget, char* param)
{
    ExternalEvent ev;

    // fill in data
    ev.type = EXTERNAL_SHOW_MSG;
    strcpy(ev.buf1, "test");

    // send to output message queue
    ExternalSend(&ev);

    return true;
}

```

- 程式在ExternalOutTask執行緒中不斷的讀取extOutQueue訊息佇列中的訊息，然後將訊息輸出到對應的週邊。 程式碼檔案在external.c，範例代碼如下：

```

static void* ExternalOutTask(void* arg)
{
    while (!extQuit)
    {
        ExternalEvent ev;

        // read message
        if (mq_receive(extOutQueue, (char*)&ev, sizeof(ExternalEvent), 0) > 0)
        {
#ifdef CFG_UART0_ENABLE && !defined(CFG_DBG_UART0)
            char writebuf[4];

            switch (ev.type)

```

```
{
case EXTERNAL_TEST0:
    writebuf[0] = '0';
    write(ITP_DEVICE_UART0, writebuf, 4);
    break;

case EXTERNAL_TEST1:
    writebuf[0] = '1';
    writebuf[1] = (char)ev.arg1;
    write(ITP_DEVICE_UART0, writebuf, 4);
    break;

case EXTERNAL_TEST2:
    writebuf[2] = '2';
    writebuf[1] = (char)ev.arg1;
    writebuf[2] = (char)ev.arg2;
    write(ITP_DEVICE_UART0, writebuf, 4);
    break;

case EXTERNAL_TEST3:
    writebuf[0] = '3';
    writebuf[1] = (char)ev.arg1;
    writebuf[2] = (char)ev.arg2;
    writebuf[3] = (char)ev.arg3;
    write(ITP_DEVICE_UART0, writebuf, 4);
    break;

case EXTERNAL_TEST4:
    writebuf[0] = '4';
    writebuf[1] = (char)ev.arg1;
    write(ITP_DEVICE_UART0, writebuf, 4);
    break;

case EXTERNAL_TEST5:
    writebuf[0] = '5';
    writebuf[1] = (char)ev.arg1;
    write(ITP_DEVICE_UART0, writebuf, 4);
    break;

case EXTERNAL_SHOW_MSG:
    write(ITP_DEVICE_UART0, ev.buf1, 4);
    break;
}
#endif
}
}
mq_close(extOutQueue);
extOutQueue = -1;

return NULL;
```

```
}
```

## 2.4 訊息定義

- 訊息型別定義在`ctrlboard.h`。範例程式碼如下：

```
typedef enum
{
    EXTERNAL_SHOW_MSG, ///< UART
    EXTERNAL_TEST0, ///< Test #0
    EXTERNAL_TEST1, ///< Test #1
    EXTERNAL_TEST2, ///< Test #2
    EXTERNAL_TEST3, ///< Test #3
    EXTERNAL_TEST4, ///< Test #4
    EXTERNAL_TEST5 ///< Test #5
} ExternalEventType;
```

使用者需要在此添加新的型別。

- 訊息結構也定義在`ctrlboard.h`。範例程式碼如下：

```
#define EXTERNAL_BUFFER_SIZE 64 ///< External buffer size

typedef struct
{
    ExternalEventType type;
    int arg1;
    int arg2;
    int arg3;
    uint8_t buf1[EXTERNAL_BUFFER_SIZE];
} ExternalEvent;
```

使用者可修改此結構以符合實際訊息的結構。