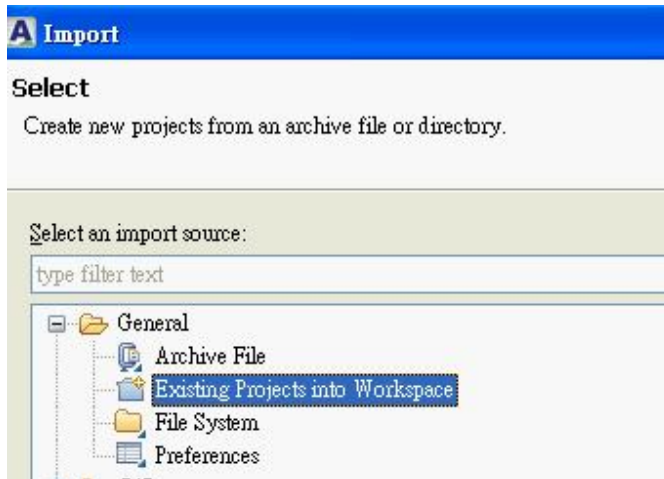


附件是一個程式例子。

將附件解壓縮後，選擇 `import existing projects into workspace` 可匯入至 AndeSight 裡。



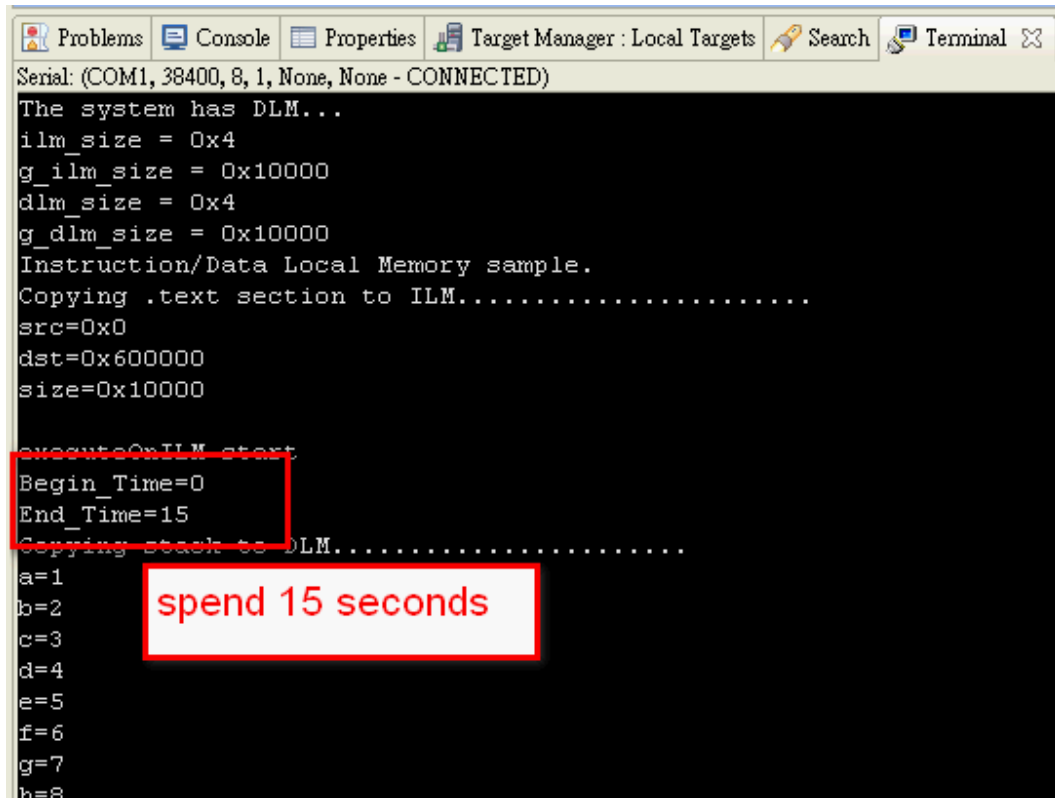
這個程式是從 AndeSight 下的 `demo/startup` 裡的 `demo-int` 及 `demo-lm` 改寫的。
要評估的程式放在下面這個區塊裡。

`time` 是一個 `global` 變數，`timer` 每 1 秒會觸發 1 次 `interrupt`，把 `time` 這個變數加 1，
所以 `time` 變數會每 1 秒就加 1。



```
428     return;
429
430 }
431 void executeOnILM()
432 {
433     //cindy
434     puts("\nexecuteOnILM start\n");
435     int Begin_Time, End_Time;
436     Begin_Time=time;
437     //Execute program
438
439     //execute start
440     int i,j,k;
441
442     for(j=0;j<1000;j++){
443         for(i=0;i<10000;i++){
444             k=i*3;
445         }
446         *(unsigned int *)0x902FFFFC=j;
447     }
448     //end execute
449
450     End_Time=time;
451     printS("Begin_Time=", Begin_Time, 10);
452     printS("End_Time=", End_Time, 10);
453 }
```

下面是執行結果，這一段程式花掉 15 秒。



```
Serial: (COM1, 38400, 8, 1, None, None - CONNECTED)
The system has DLM...
ilm_size = 0x4
g_ilm_size = 0x10000
dln_size = 0x4
g_dln_size = 0x10000
Instruction/Data Local Memory sample.
Copying .text section to ILM.....
src=0x0
dst=0x600000
size=0x10000
executeOnILM_start
Begin_Time=0
End_Time=15
Copying stack to DLM.....
a=1
b=2
c=3
d=4
e=5
f=6
g=7
h=8
```

這個程式 run 在 ILM 裡面，所以速度比較快一些，

但是 data 和 stack 仍在 DRAM 裡。

這一塊 N8 上的板子有 64KB 的 ILM。另外也有 64KB 的 DLM，

不過這個程式裡沒有把 .data 區 copy 到 DLM。

makefile 裡的 optimization 可以自己改，我在裡面寫的是 -O0，沒有經過最佳化。

您可以下 flags 做最佳化。