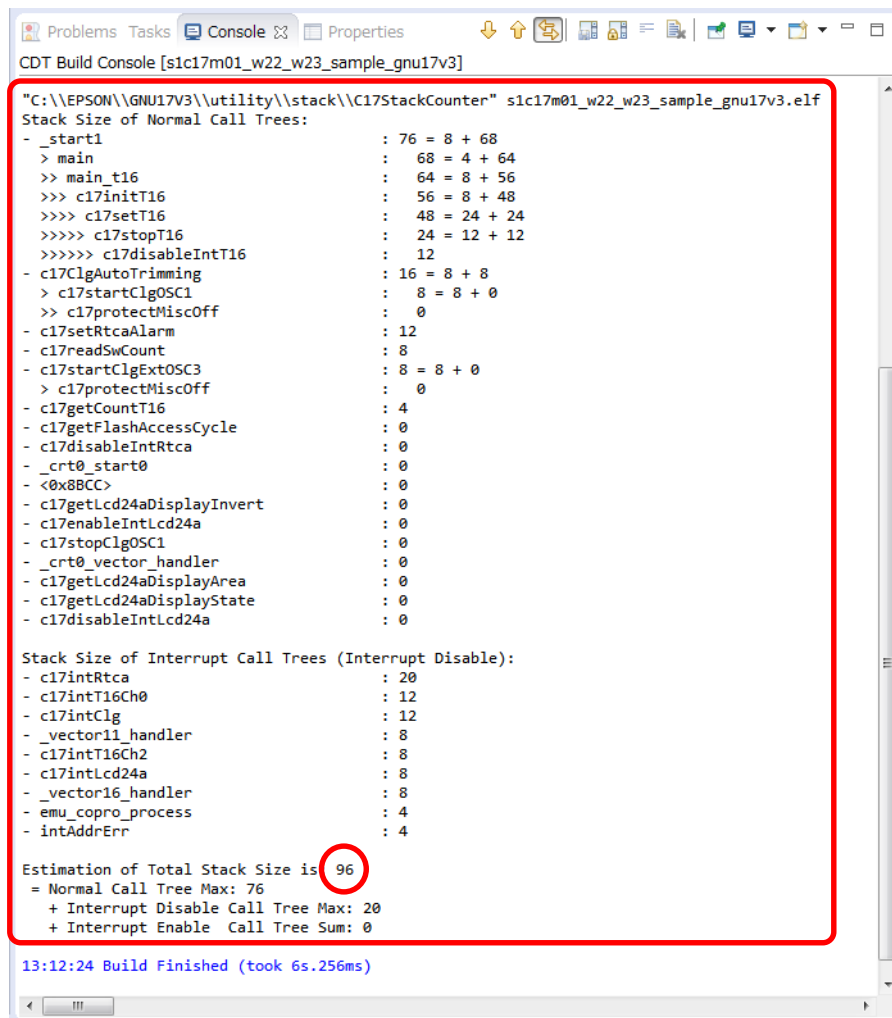


# スタック使用量静的解析ツールの使い方

“C17StackCounter” は、スタックが使用するメモリサイズを静的に解析するツールです。Build時に、[Console]ビューに結果を表示することが可能です。



```
CDT Build Console [s1c17m01_w22_w23_sample_gnu17v3]

"C:\EPSON\GNU17V3\utility\stack\C17StackCounter" s1c17m01_w22_w23_sample_gnu17v3.elf
Stack Size of Normal Call Trees:
- _start1 : 76 = 8 + 68
  > main : 68 = 4 + 64
    >> main_t16 : 64 = 8 + 56
      >>> c17initT16 : 56 = 8 + 48
        >>>> c17setT16 : 48 = 24 + 24
          >>>>> c17stopT16 : 24 = 12 + 12
            >>>>>> c17disableIntT16 : 12
          - c17ClgAutoTrimming : 16 = 8 + 8
            > c17startClgOSC1 : 8 = 8 + 0
              >> c17protectMiscOff : 0
            - c17setRtcaAlarm : 12
            - c17readSwCount : 8
            - c17startClgExtOSC3 : 8 = 8 + 0
              > c17protectMiscOff : 0
            - c17getCountT16 : 4
            - c17getFlashAccessCycle : 0
            - c17disableIntRtca : 0
            - _crt0_start0 : 0
            - <0x8BCC> : 0
            - c17getLcd24aDisplayInvert : 0
            - c17enableIntLcd24a : 0
            - c17stopClgOSC1 : 0
            - _crt0_vector_handler : 0
            - c17getLcd24aDisplayArea : 0
            - c17getLcd24aDisplayState : 0
            - c17disableIntLcd24a : 0

Stack Size of Interrupt Call Trees (Interrupt Disable):
- c17intRtca : 20
- c17intT16Ch0 : 12
- c17intClg : 12
- _vector11_handler : 8
- c17intT16Ch2 : 8
- c17intLcd24a : 8
- _vector16_handler : 8
- emu_copro_process : 4
- intAddrErr : 4

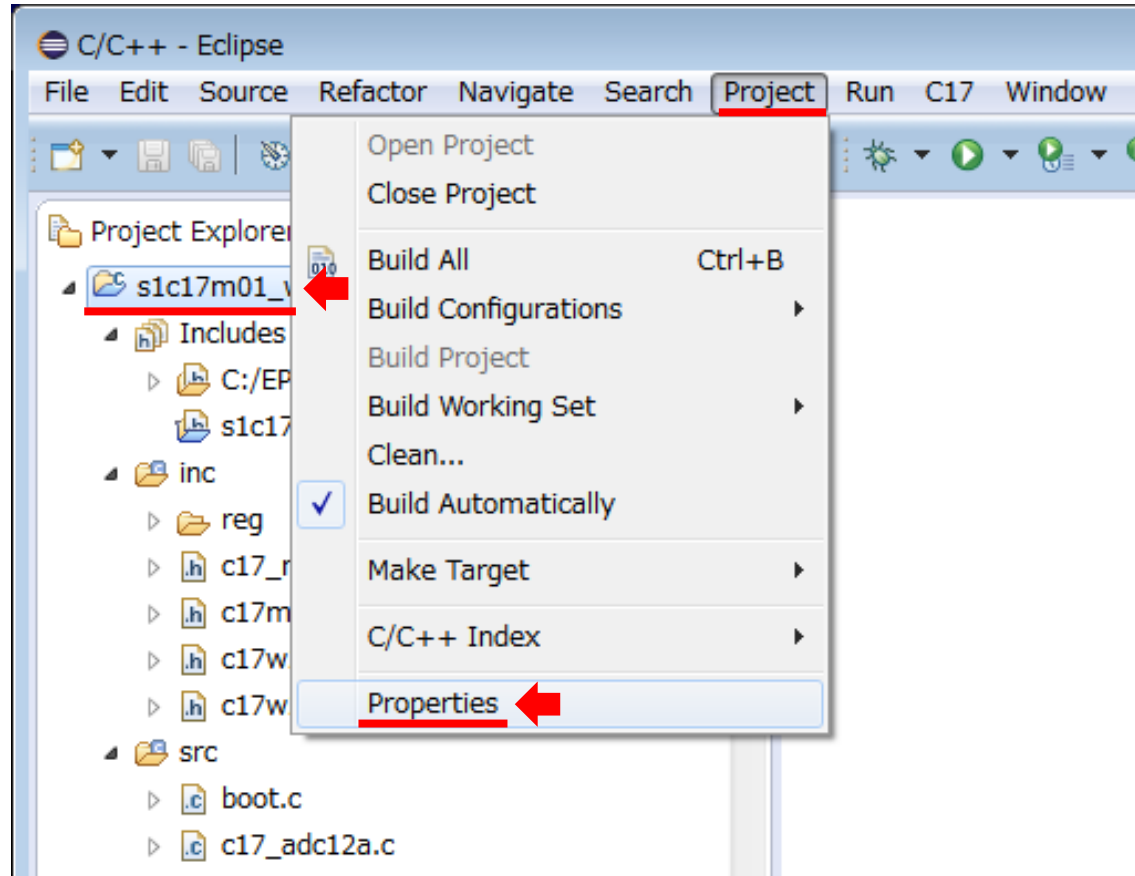
Estimation of Total Stack Size is 96
= Normal Call Tree Max: 76
+ Interrupt Disable Call Tree Max: 20
+ Interrupt Enable Call Tree Sum: 0

13:12:24 Build Finished (took 6s.256ms)
```

Estimation of Total Stack Size is : 96 → 見積サイズ 96byte

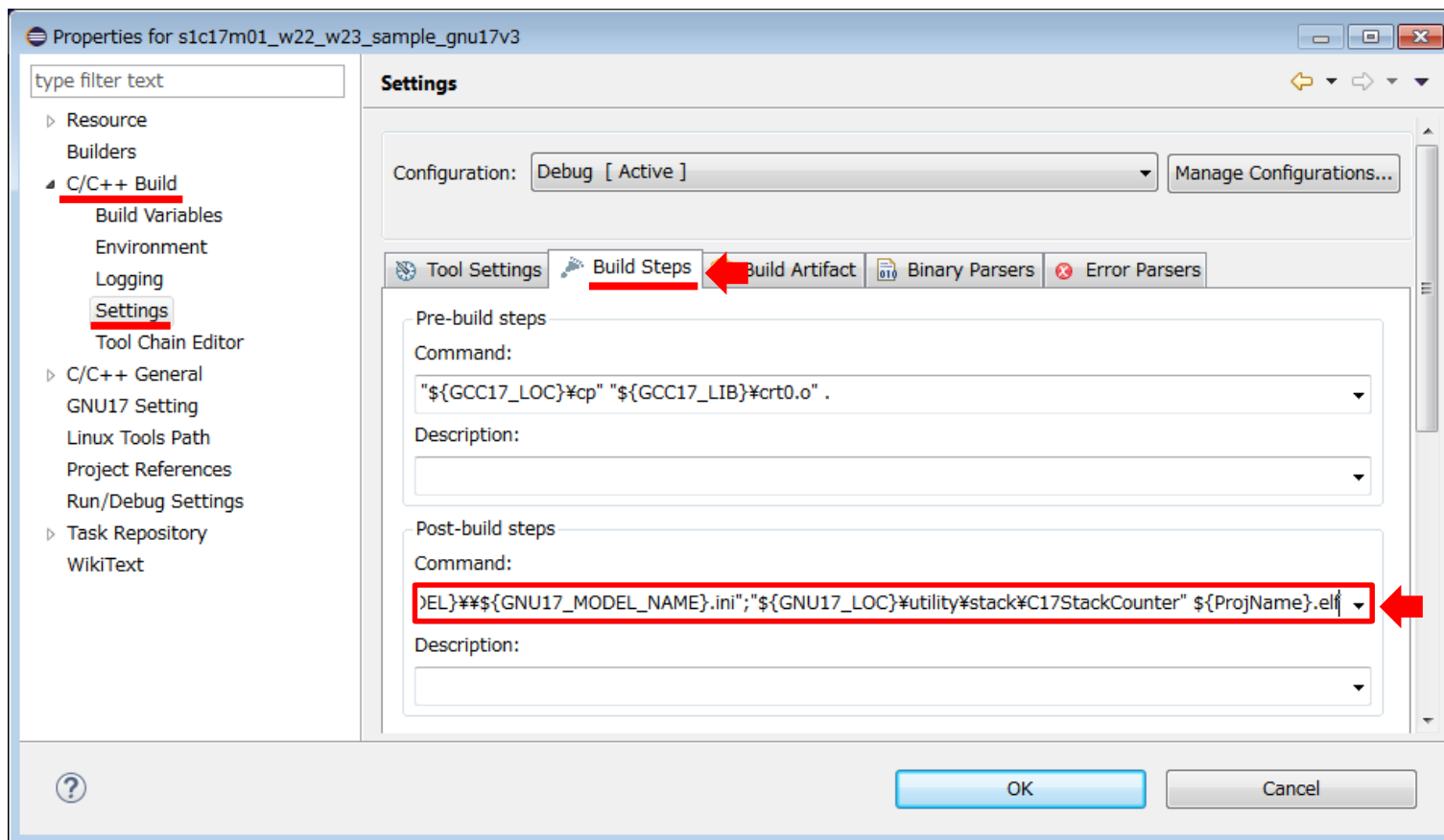
## 設定方法

1. プロジェクトを選択し、[Project]>[Propaties]を選択してください。



2. [C/C++ Build]>[Settings]>[Build Steps]タブを選択します。  
Post-build stepsの[Command:]ボックスに以下の内容を、**追記**してください。  
※元々の設定は消さずに、追記してください。

**;"\${GNU17\_LOC}%utility%stack%C17StackCounter" \${ProjName}.elf**



以下の場合、解析ができません。

- ・ アセンブラを用いて、gccと異なる関数出口処理を記述した場合
- ・ 飛び先アドレスをテーブルや変数で管理している場合  
(ベクターテーブルを除く)

解析ができなかった場合は、

Stack Size of Unsolved Functions and its Callees

に紐づけされなかった関数群が表示され、以下のお知らせが表示されます。

Note: This program contains some unsolved calls.