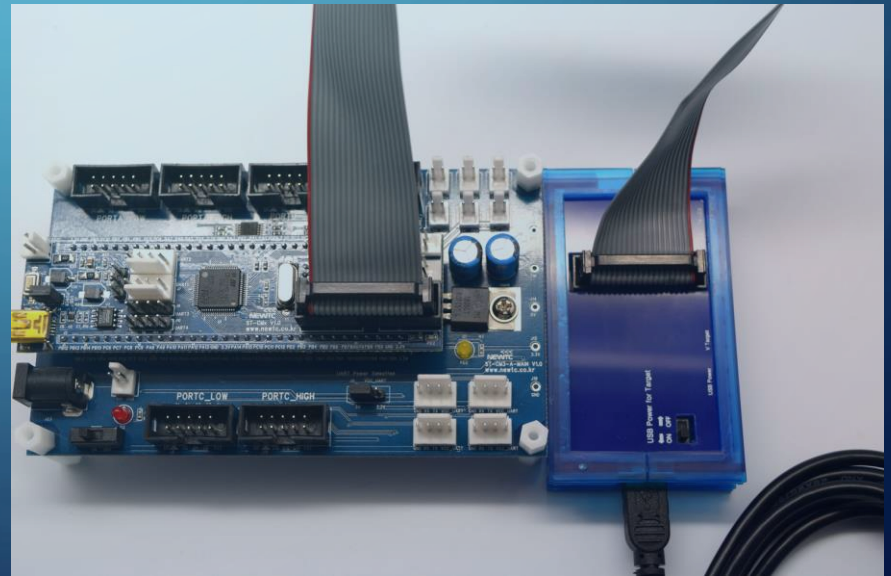


STM32 CORTEX-M3 예제 코드 불러오기

(주) 뉴티씨

대표이사 류대우





목차

1. 예제 프로젝트 불러오기

2. 컴파일

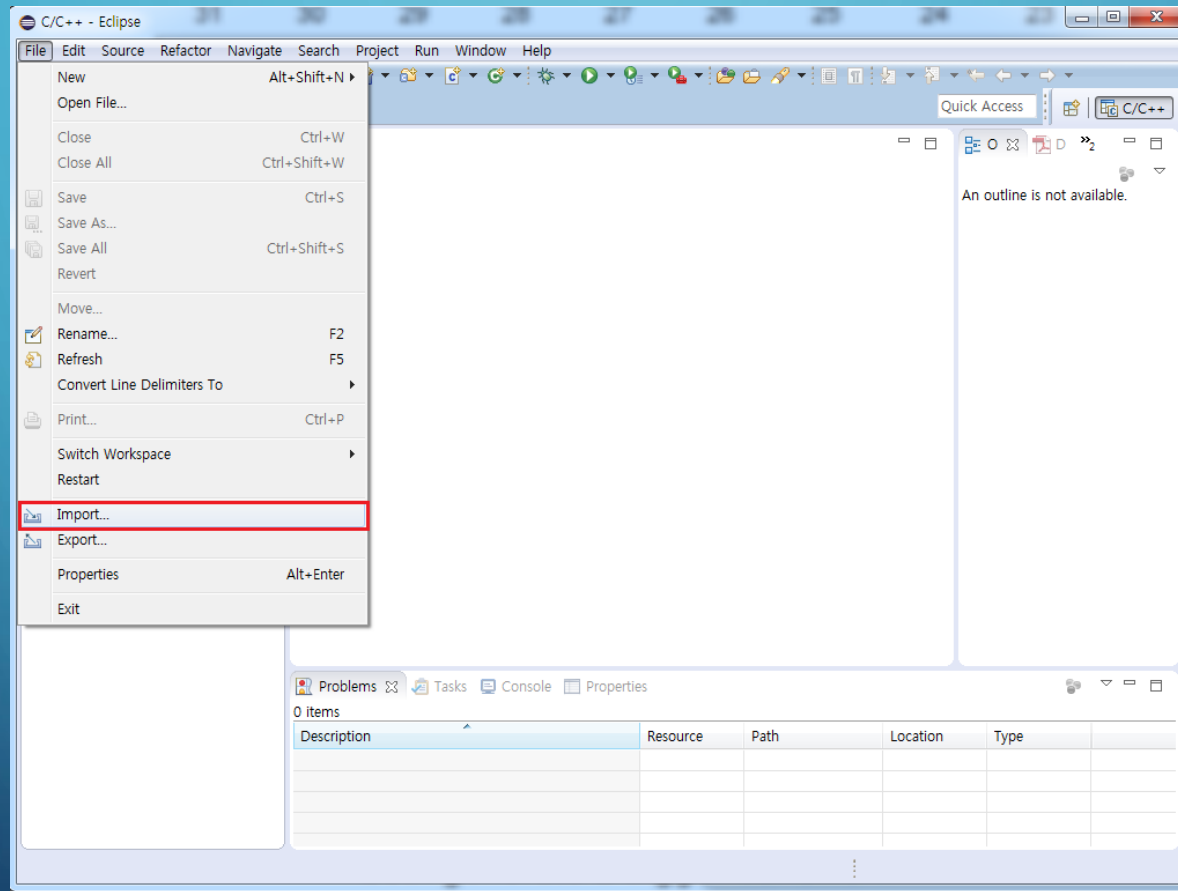


1. 예제 프로젝트 불러오기

1) Project Import 창을 연다.

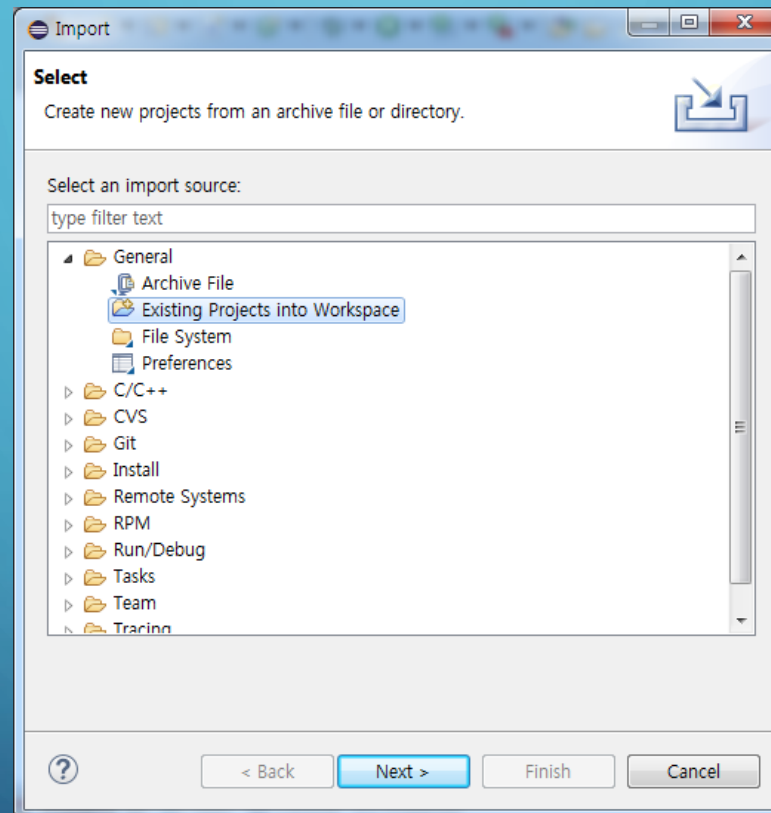
2) 자료실에서 내려받은 프로젝트 디렉터리를 불러온다.

1-(1)-A. PROJECT IMPORT



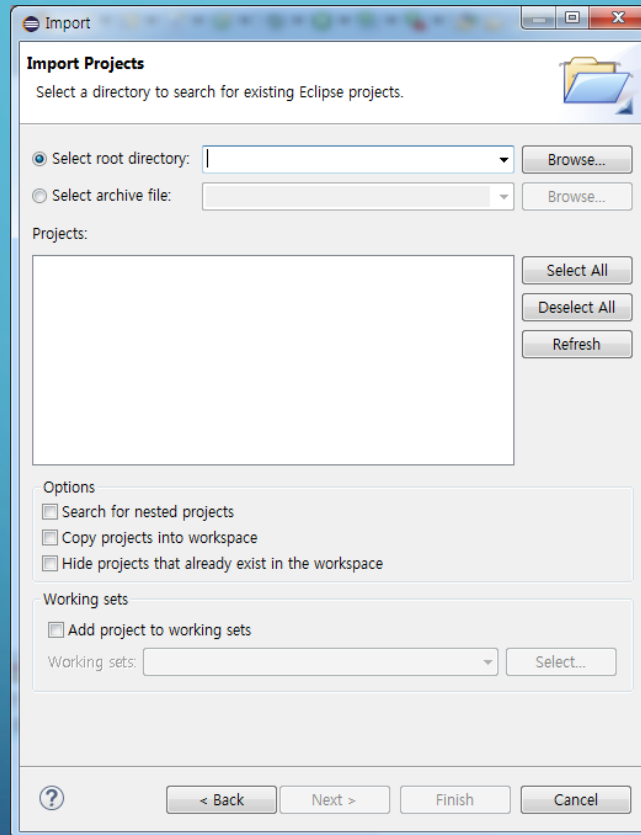
Eclipse 의 File -> Import... 를 클릭

1-(1)-B. PROJECT IMPORT



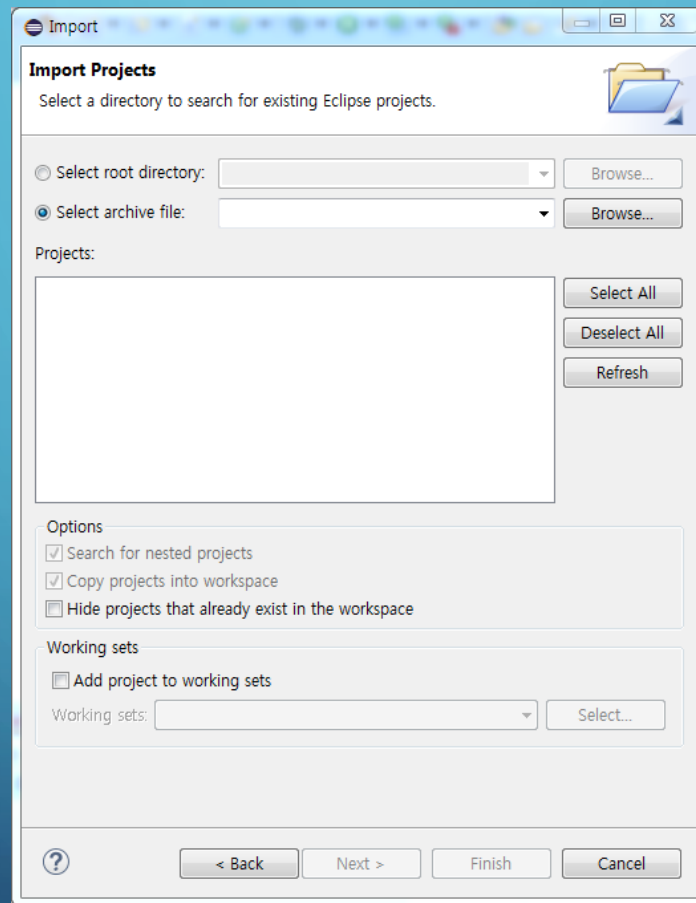
General -> Existing Projects into Workspace 선택
Next 버튼 클릭

1-(2)-A. PROJECT IMPORT



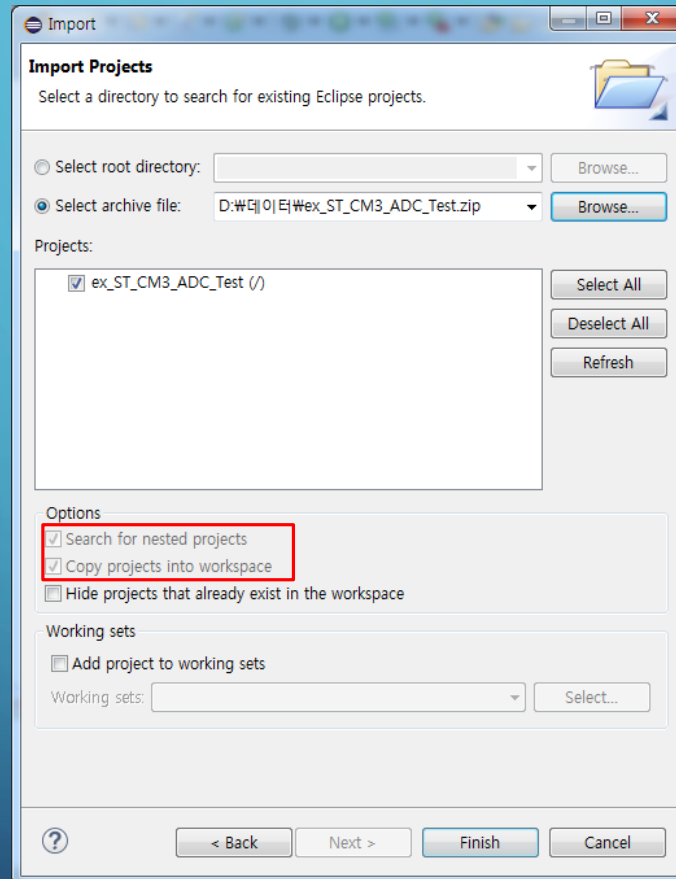
1. Select root directory 에서 예제의 압축을 푼 경로를 지정

1-(2)-B. PROJECT IMPORT



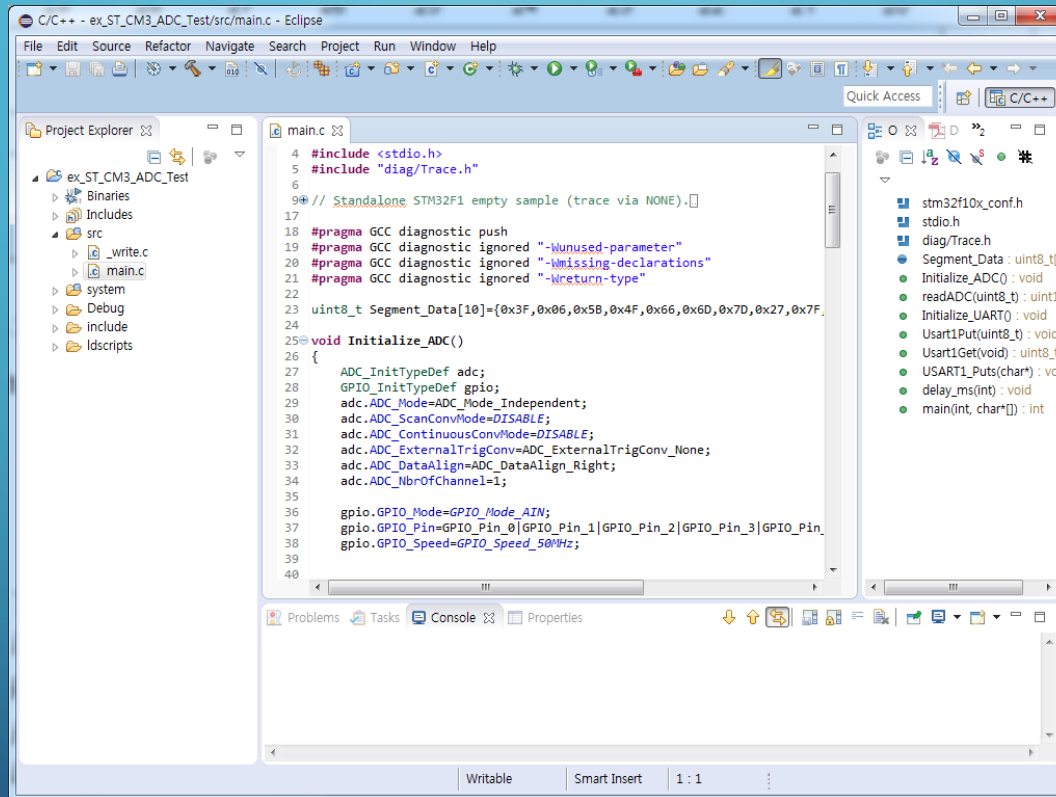
2. Select archive file 에서 예제 압축파일 경로를 지정

1-(2)-C. PROJECT IMPORT



Search for nested projects, Copy projects into workspace 체크
Finish 클릭

1-(2)-D. PROJECT IMPORT



```
1  #include <stdio.h>
2  #include "diag/Trace.h"
3
4  // Standalone STM32F1 empty sample (trace via NONE).
5
6
7
8
9
10
11
12
13
14
15
16
17
18 #pragma GCC diagnostic push
19 #pragma GCC diagnostic ignored "-Wunused-parameter"
20 #pragma GCC diagnostic ignored "-Wmissing-declarations"
21 #pragma GCC diagnostic ignored "-Wreturn-type"
22
23 uint8_t Segment_Data[10]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x27,0x7F,0x7F};
24
25 void Initialize_ADC()
26 {
27     ADC_InitTypeDef adc;
28     GPIO_InitTypeDef gpio;
29     adc.ADC_Mode=ADC_Mode_Independent;
30     adc.ADC_ScanConvMode=DISABLE;
31     adc.ADC_ContinuousConvMode=DISABLE;
32     adc.ADC_ExternalTrigConv=ADC_ExternalTrigConv_None;
33     adc.ADC_DataAlign=ADC_DataAlign_Right;
34     adc.ADC_NbrOfChannel=1;
35
36     gpio.GPIO_Mode=GPIO_Mode_AIN;
37     gpio.GPIO_Pin=GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6|GPIO_Pin_7;
38     gpio.GPIO_Speed=GPIO_Speed_50MHz;
39
40 }
```

Main.c 확인



2. 컴파일

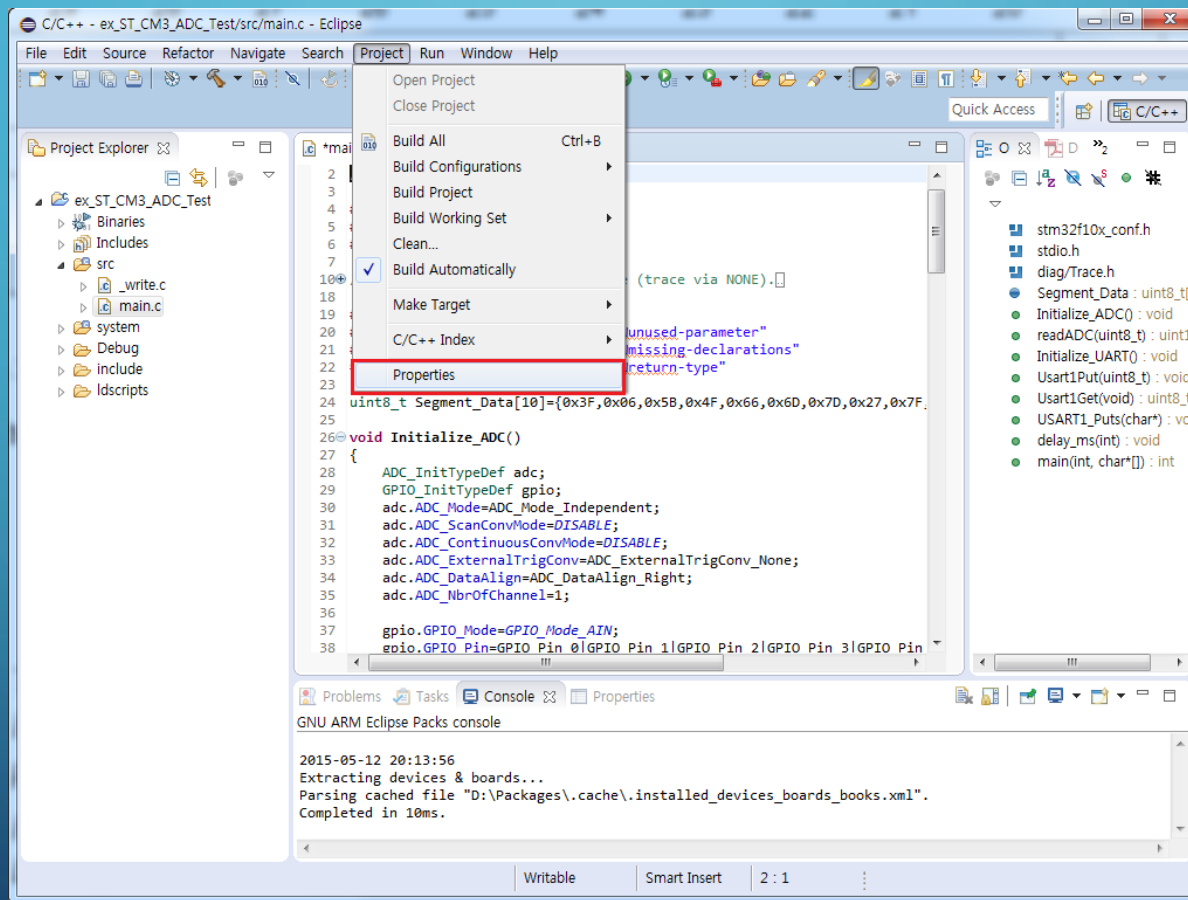
1) Project Settings 창을 연다.

2) Build tools 경로를 수정한다.

3) 컴파일.

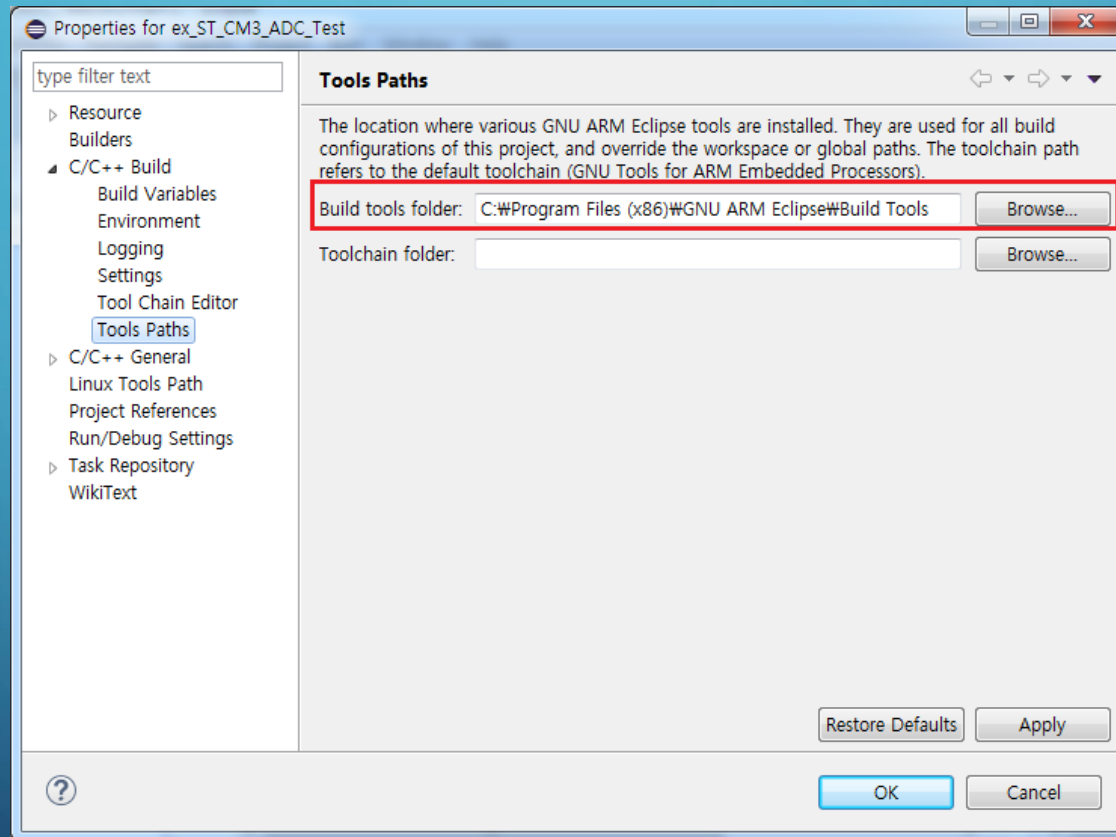


2- (1). 컴파일



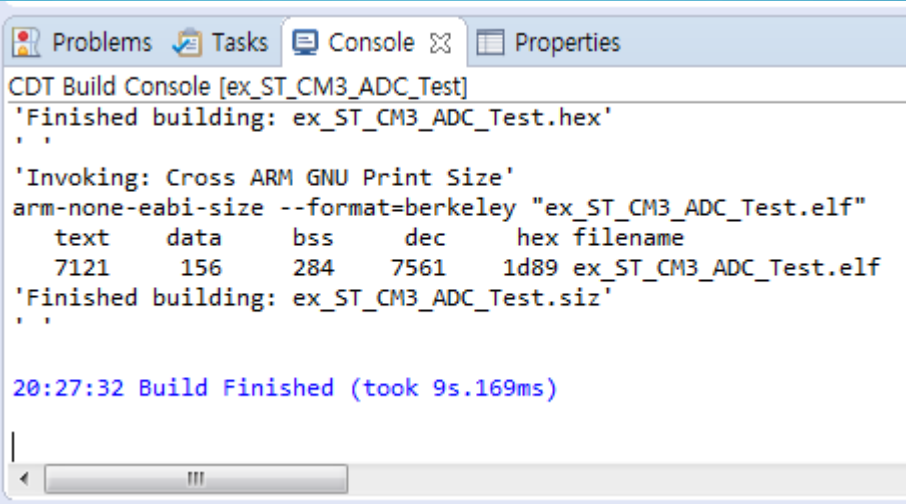
Projects -> Properties 클릭 (왼쪽쪽에서는 main.c 선택)

2- (2). 컴파일



C/C++ Build -> Tools Paths 에 접근
Build tools 경로 지정 (설치된 경로로), Linux 는 해당 칸을 공란으로 변경
OK 클릭

2- (3). 컴파일



The screenshot shows the CDT Build Console window for the project 'ex_ST_CM3_ADC_Test'. The console output indicates that the build is finished, and the size of the generated ELF file is being calculated using the 'arm-none-eabi-size' command. The output shows the size of the text, data, and bss sections in decimal and hexadecimal, along with the filename 'ex_ST_CM3_ADC_Test.elf'. The build finished at 20:27:32 and took 9s.169ms.

```
CDT Build Console [ex_ST_CM3_ADC_Test]
'Finished building: ex_ST_CM3_ADC_Test.hex'
', '
'Invoking: Cross ARM GNU Print Size'
arm-none-eabi-size --format=berkeley "ex_ST_CM3_ADC_Test.elf"
  text    data    bss    dec    hex filename
  7121    156    284    7561    1d89 ex_ST_CM3_ADC_Test.elf
'Finished building: ex_ST_CM3_ADC_Test.siz'
', '

20:27:32 Build Finished (took 9s.169ms)
```

Ctrl+B 키를 눌러 빌드