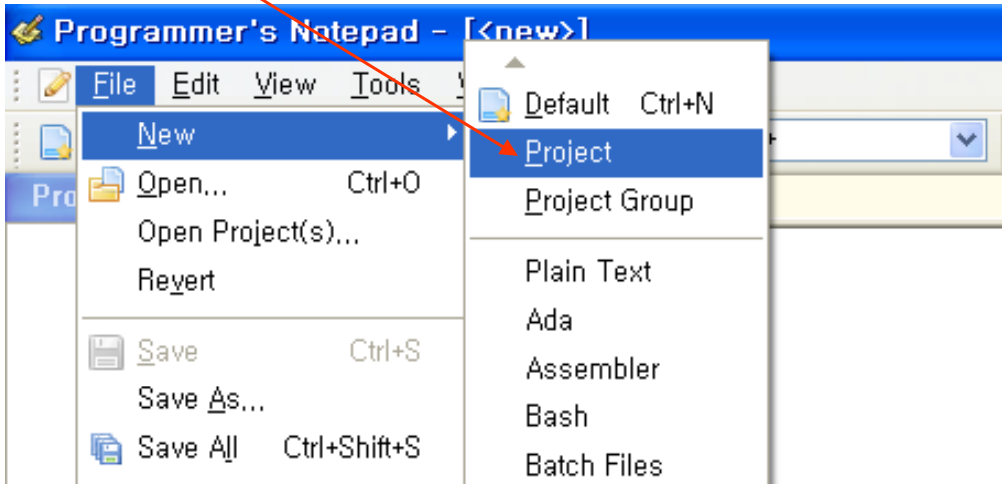


* WinARM Project 환경 만들기

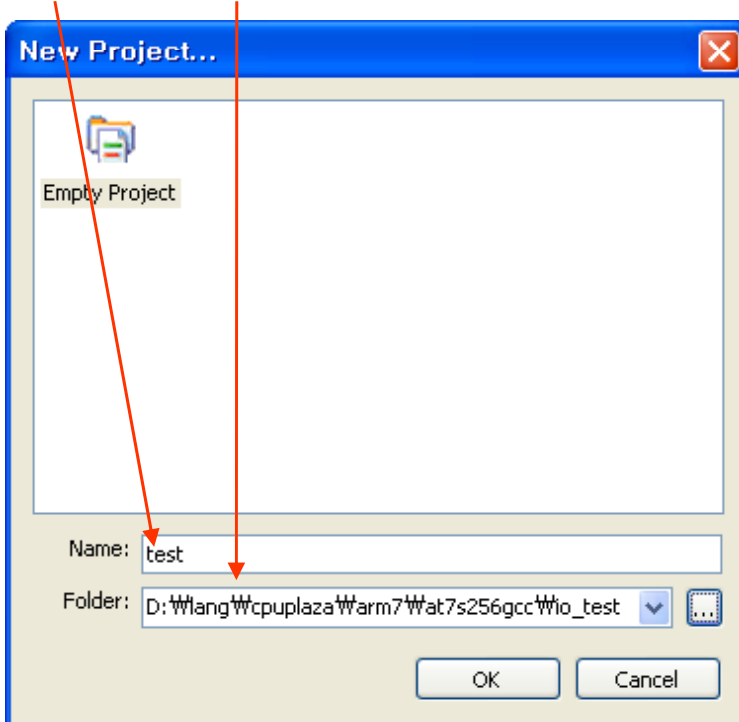
1. 설치된 Programmer's Notepad를 실행 합니다.



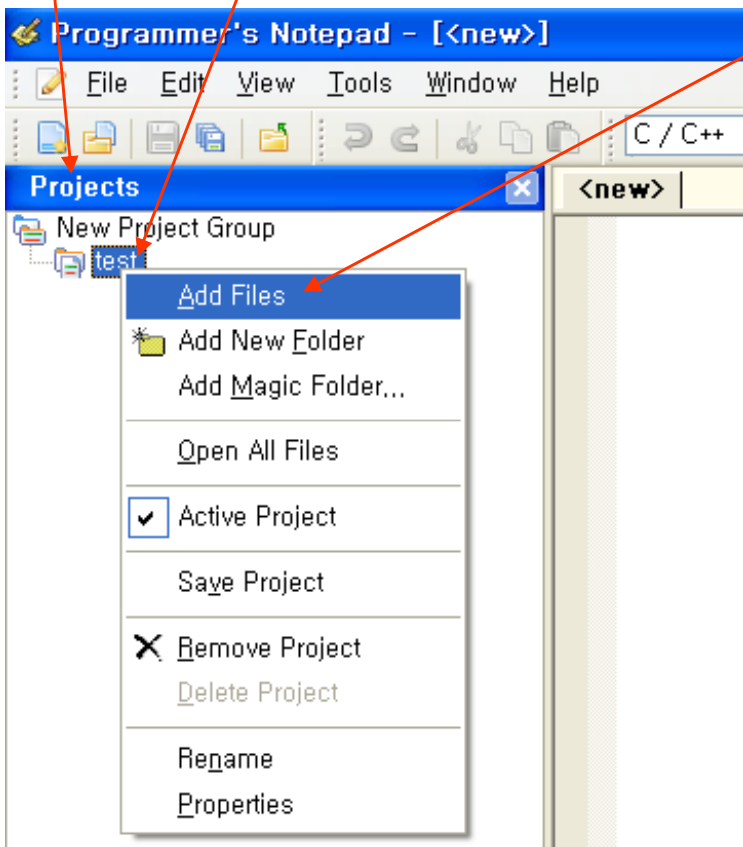
2. File->New->Project 를 이용하여 Project를 생성 합니다.



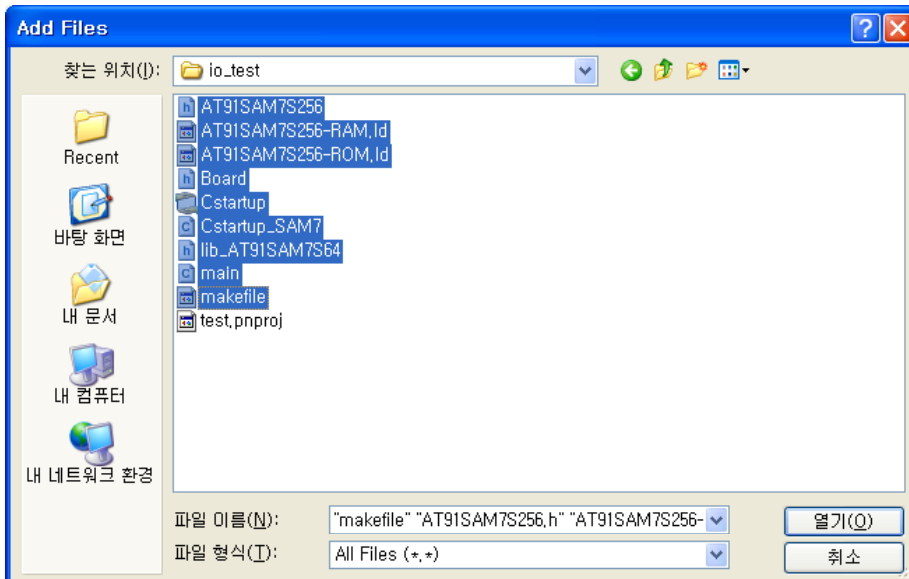
3. Project Name과 Folder를 지정후 OK버튼을 클릭 합니다.



4. Projects 창에서 Project명 을 선택후 오른쪽 마우스를 클릭 하여 Add Files로 파일을 등록 합니다.

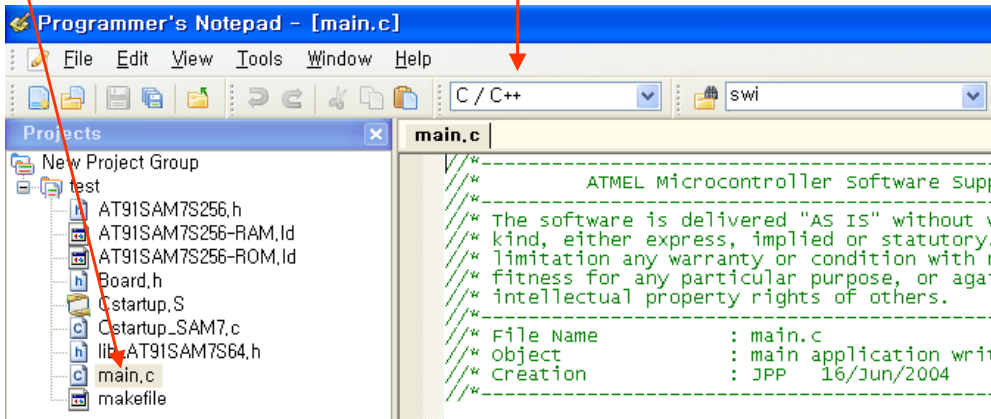


5. 등록 파일을 선택후 열기 버튼을 클릭 합니다.

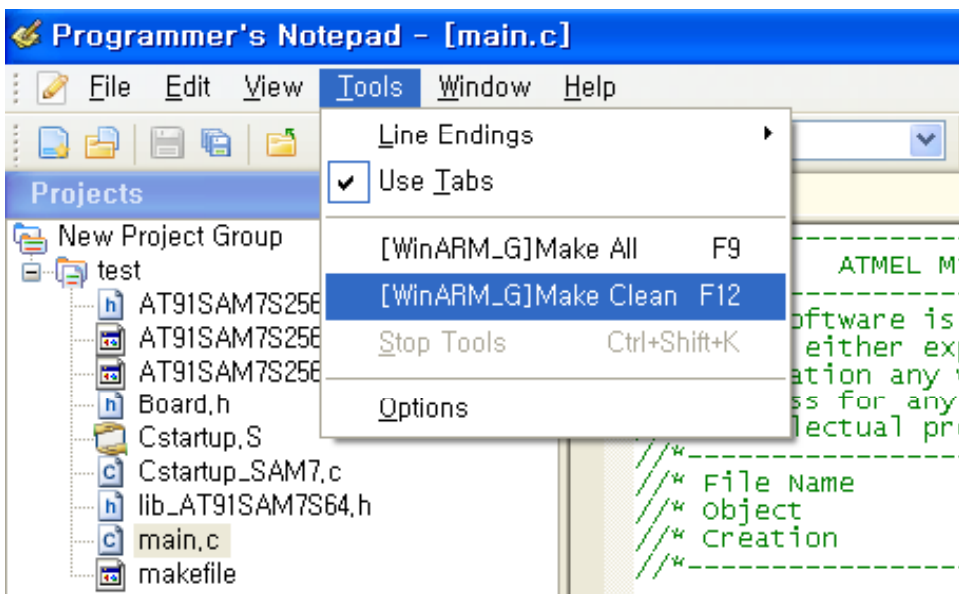


6. WinARM 인스톨 매뉴얼의 **Programmers Notepad (PN) 환경 설정**을 참조 하여 환경 설정을 확인 합니다.
제어판의 환경변수 등록(PATH항목)도 확인.

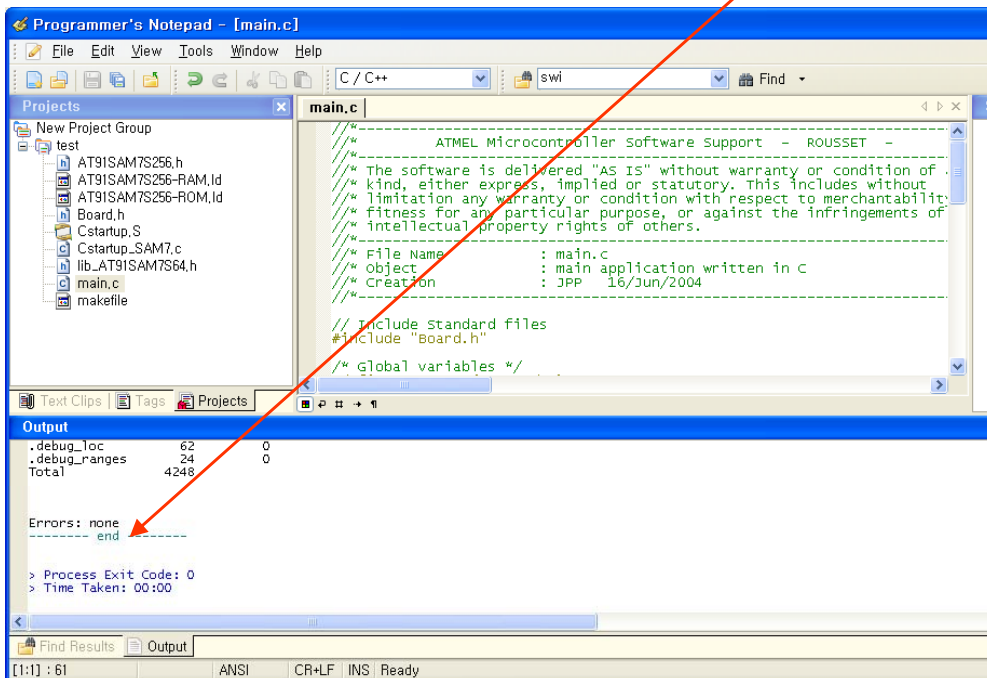
7. *.C화일을 더블클릭 하여 활성화 한후 C/C++창으로 활성화 되어 있는지 확인 합니다.



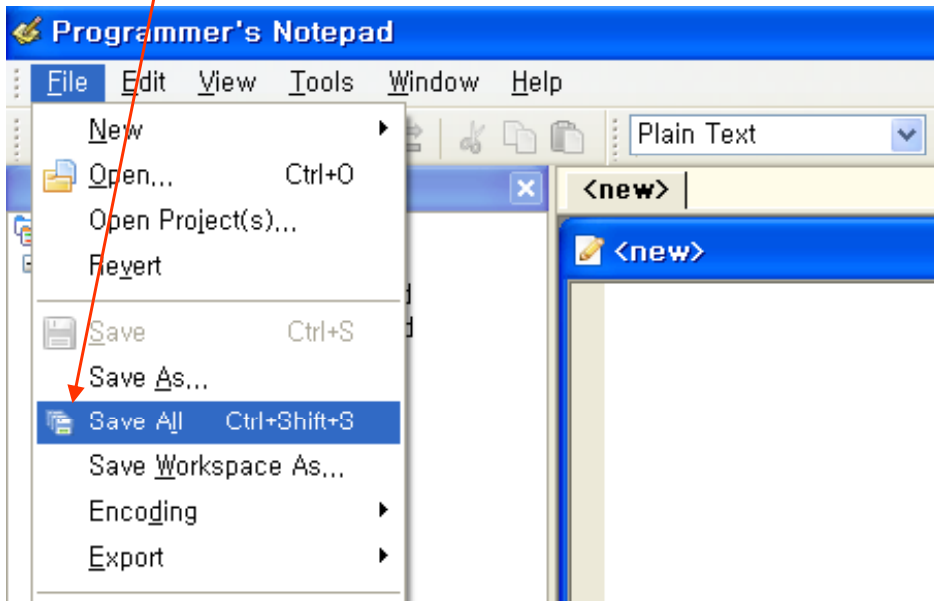
8. Tools Make Clean을 클릭 하여 .OBJ 등 기존 컴파일 생성물을 삭제 합니다.



9. Tools Make All을 클릭 하여 프로젝트를 컴파일 한후 에러 상태를 확인 합니다.



10. Project를 저장 합니다.



11. SAM-BA(ISP)나 HI-JTAG를 사용하여(매뉴얼참조) 프로그램을 로딩한후 전원을 Off->On 합니다.

12. 결과물을 *.HEX나 *.bin 으로 변경하는 방법

1. Text Edit를 이용 하여 Makefile을 Open한후 아래 부분을 변경후 저장 합니다.

```
# MCU name and submodel
MCU = arm7tdmi
SUBMDL = AT91SAM7S256 ← CPU 종류 설정

#USE_THUMB_MODE = YES
USE_THUMB_MODE = NO

## Create ROM-Image (final)
RUN_MODE=ROM_RUN
## Create RAM-Image (debugging)
##( not used: example does not fit in AT91SAM7S256 RAM )
#RUN_MODE=RAM_RUN

# not used in this example!
## Exception-Vector placement only supported for "ROM_RUN"
## (placement settings ignored when using "RAM_RUN")
## - Exception vectors in ROM:
#VECTOR_LOCATION=VECTORS_IN_ROM
## - Exception vectors in RAM:
#VECTOR_LOCATION=VECTORS_IN_RAM

# Target file name (without extension).
TARGET = main

# List C source files here. (C dependencies are automatically generated.)
# use file-extension c for "c-only"-files
SRC = $(TARGET).c

# List C source files here which must be compiled in ARM-Mode.
# use file-extension c for "c-only"-files
SRCARM = Cstartup_SAM7.c

# List C++ source files here.
# use file-extension cpp for C++-files (use extension .cpp)
CPPSRC =

# List C++ source files here which must be compiled in ARM-Mode.
# use file-extension cpp for C++-files (use extension .cpp)
#CPPSRCARM = $(TARGET).cpp
CPPSRCARM =

# List Assembler source files here.
# Make them always end in a capital .S. Files ending in a lowercase .s
# will not be considered source files but generated files (assembler
# output from the compiler), and will be deleted upon "make clean"!
# Even though the DOS/Win* filesystem matches both .s and .S the same,
# it will preserve the spelling of the filenames, and gcc itself does
# care about how the name is spelled on its command-line.
ASRC =

# List Assembler source files here which must be assembled in ARM-Mode..
ASRCARM = Cstartup.S

## Output format. (can be ihex or binary)
## (binary i.e. for openocd and SAM-BA, hex i.e. for lpc21isp and uVision)
#FORMAT = ihex ← HEX파일 생성시
FORMAT = binary ← BIN파일 생성시(현재)
```