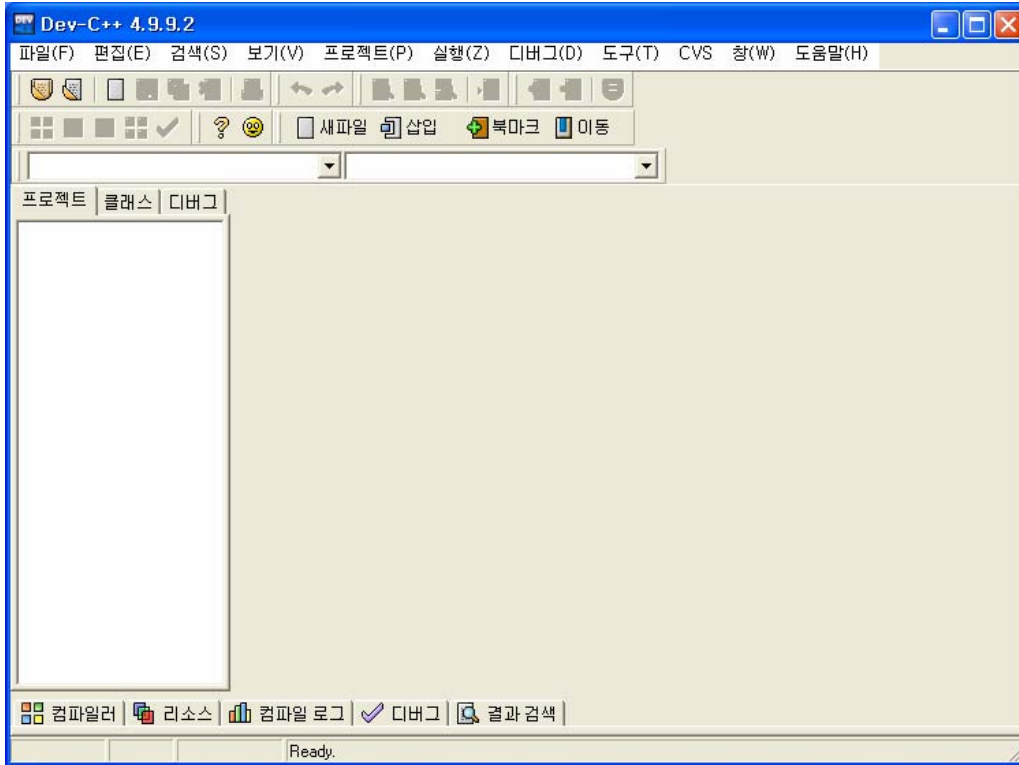
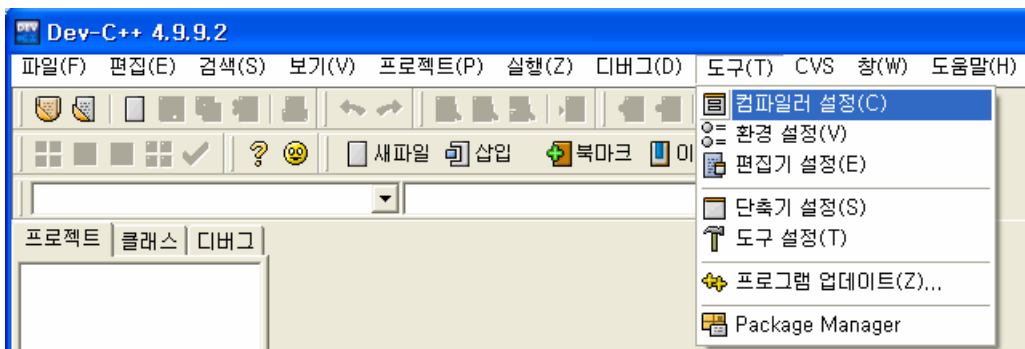


* mspgcc로 Project 환경 만들기

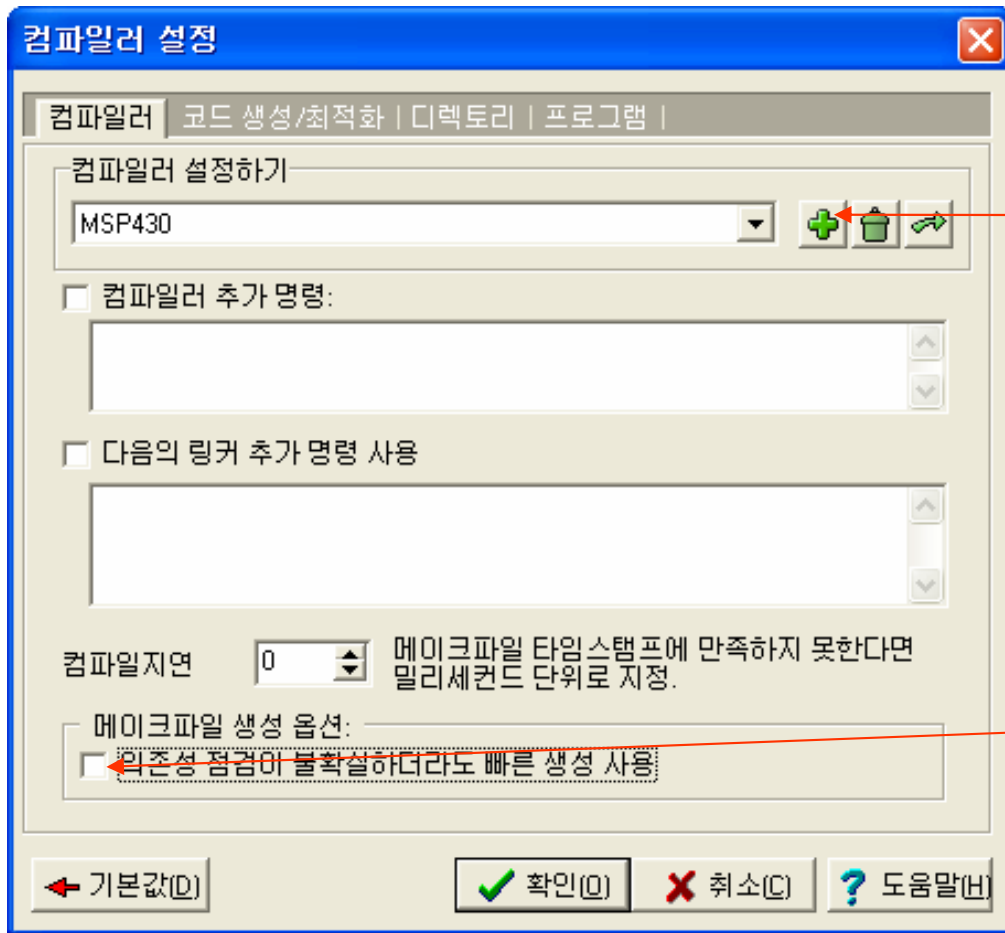
1. Dev-C++를 실행 합니다.



* 컴파일러 설정창 을 선택 합니다.



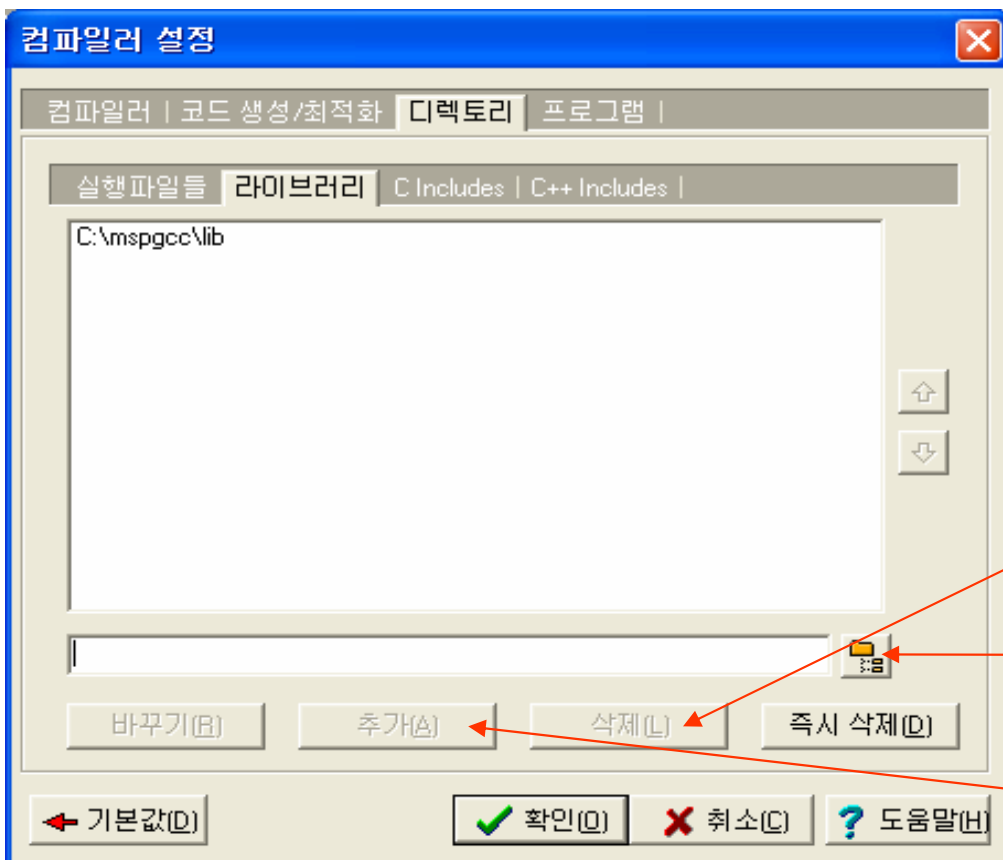
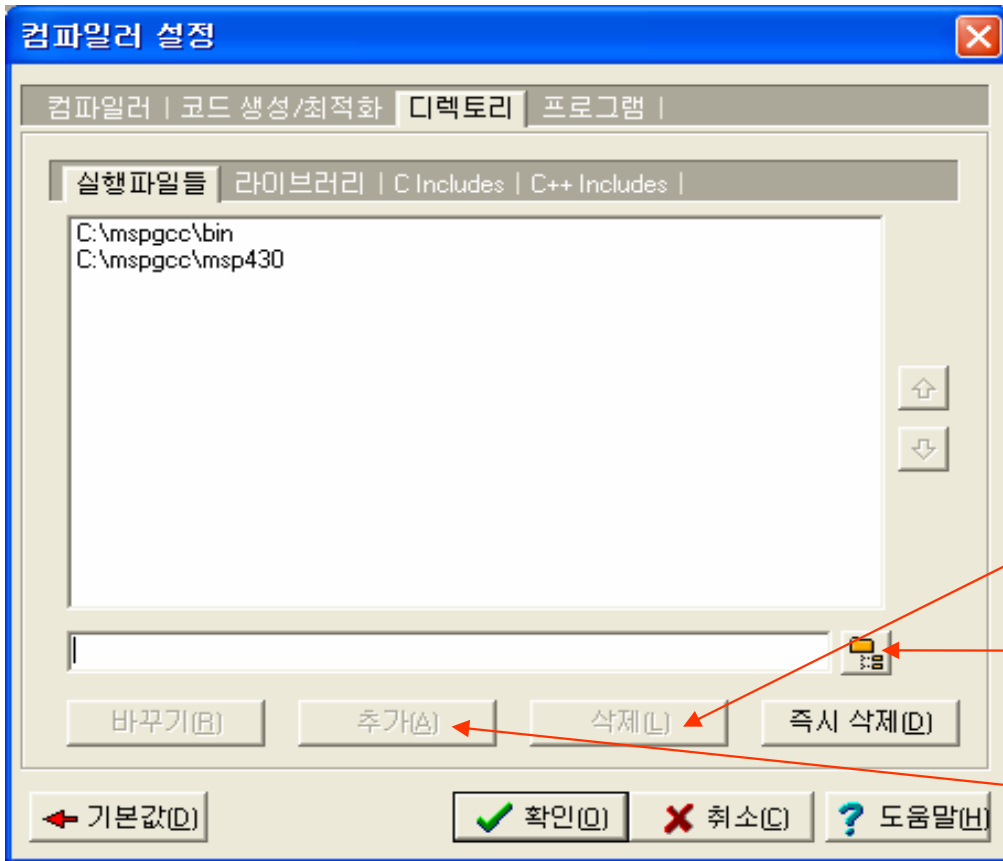
* 컴파일러 설정

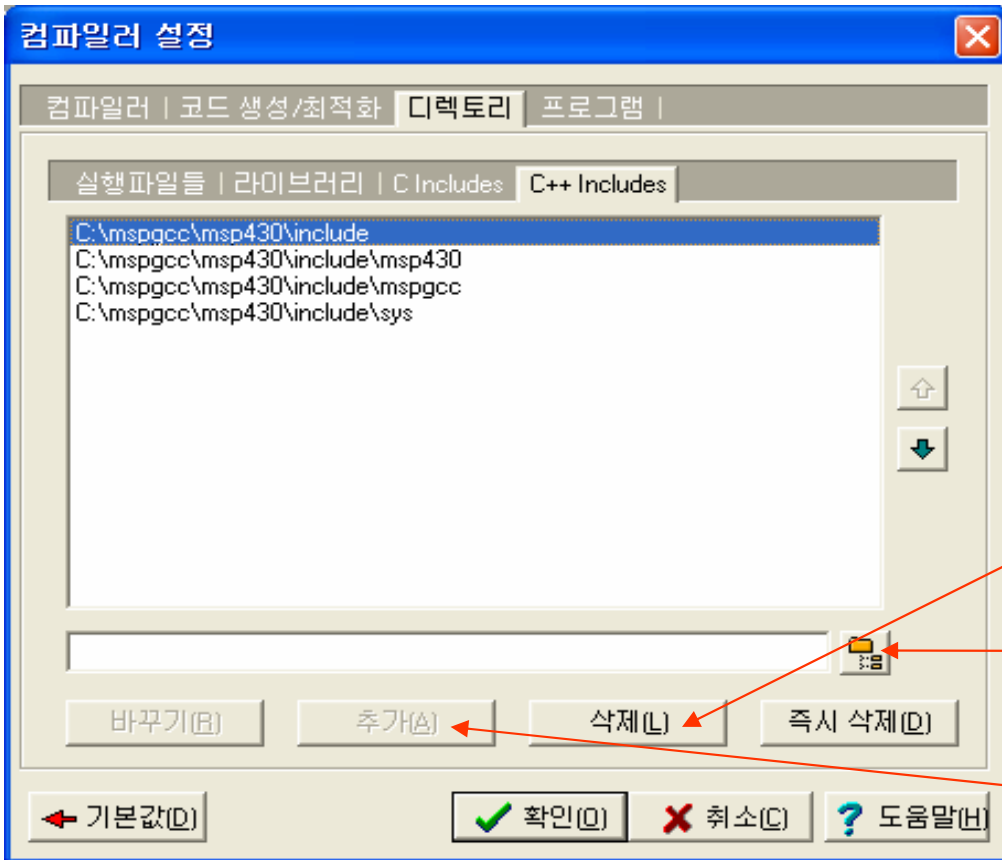
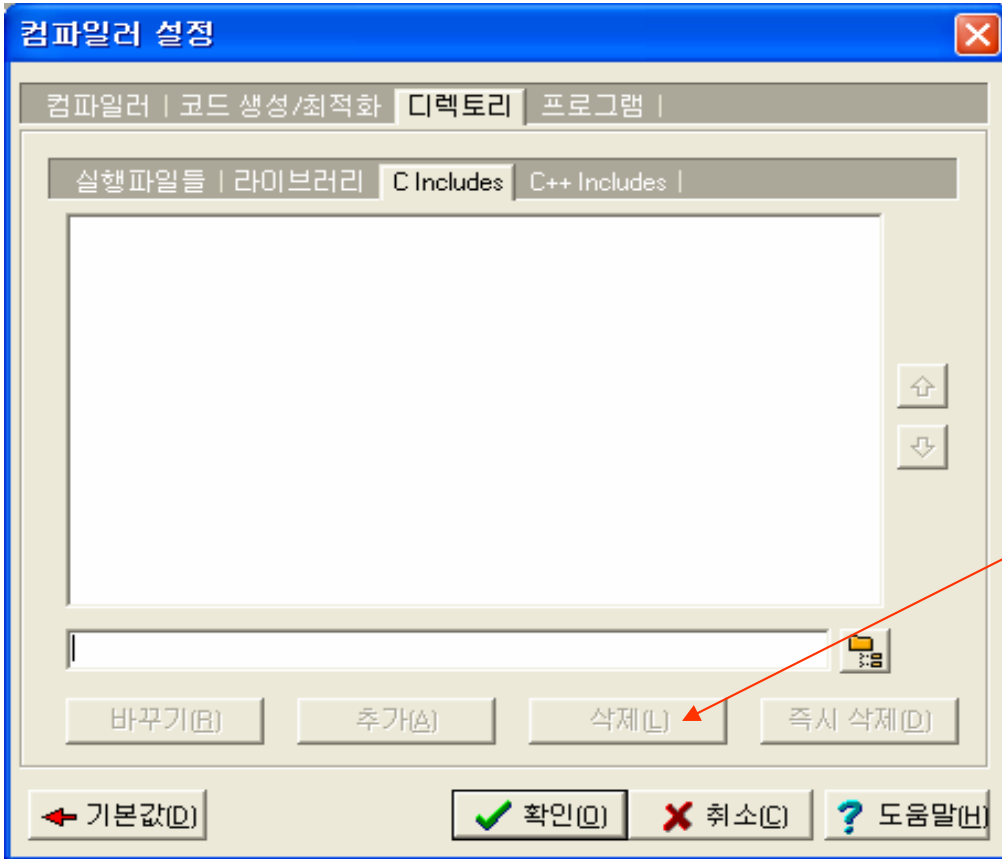


① 컴파일러 명 생성

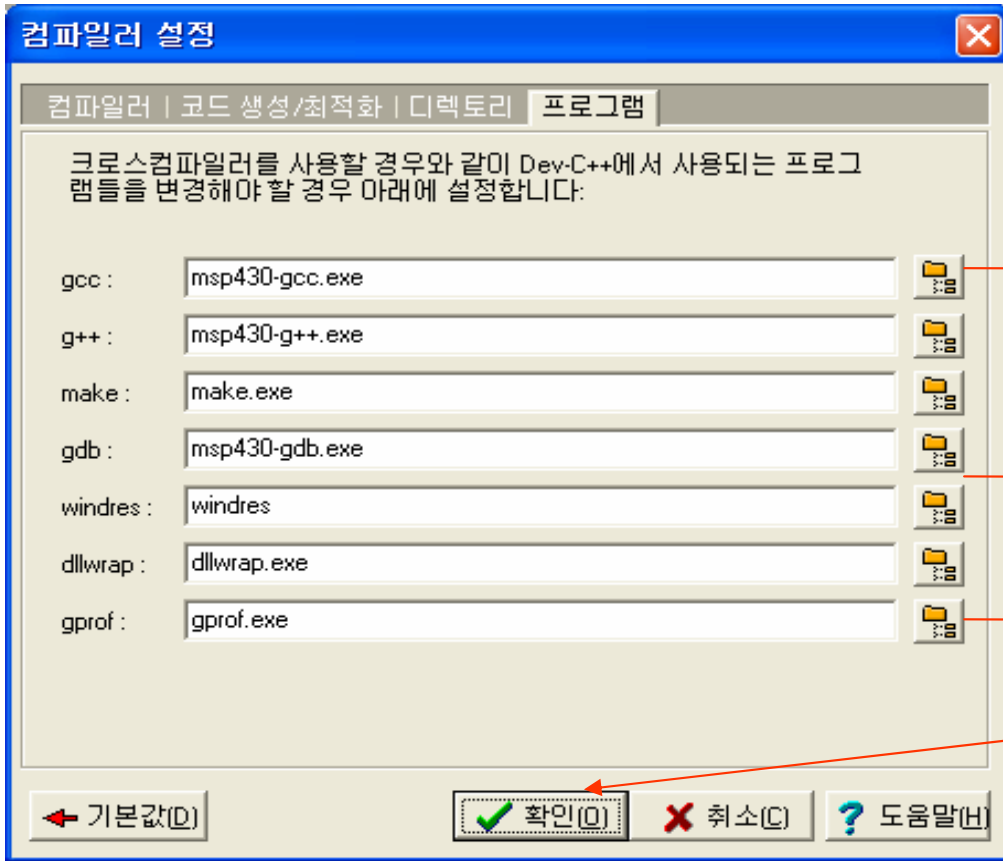
② 체크 삭제

* 컴파일러 디렉토리 설정





* 컴파일러 프로그램 설정



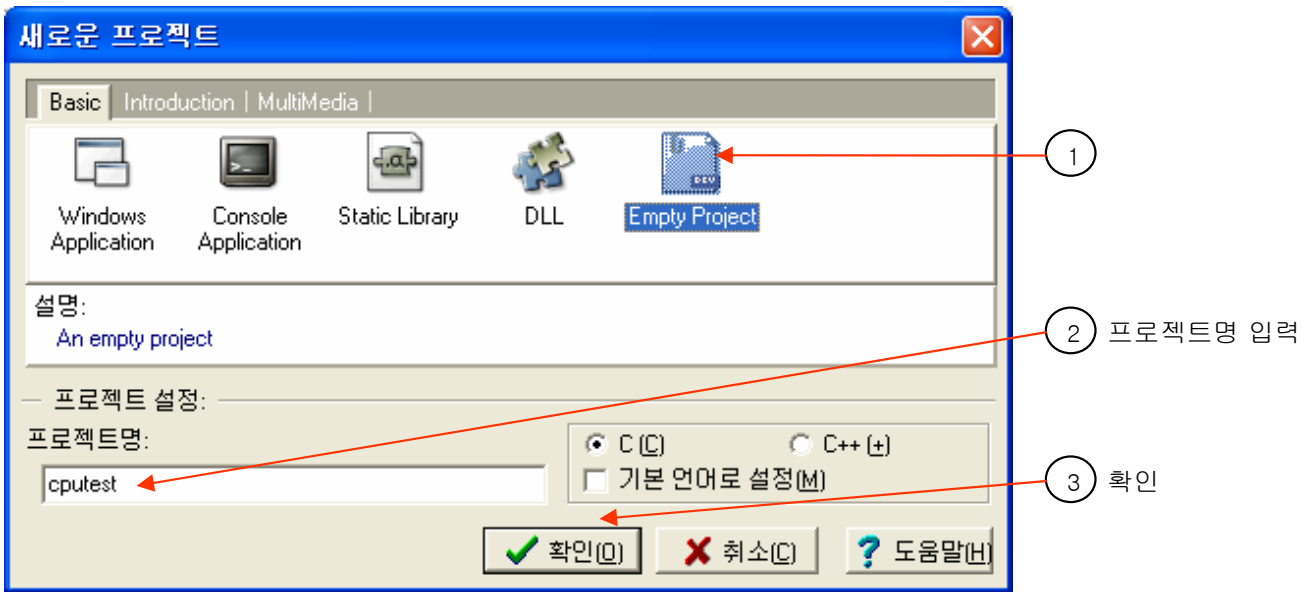
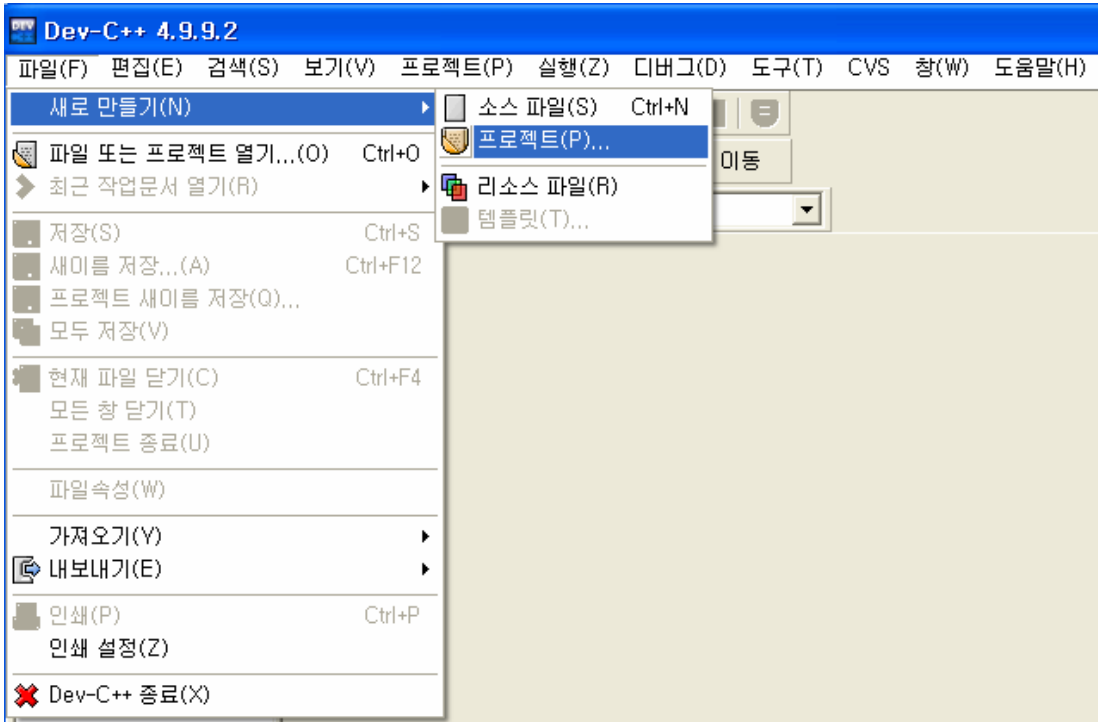
① MspgccWbin
에서 각 항목 설정
make는 직접 입력

② 기존 항목

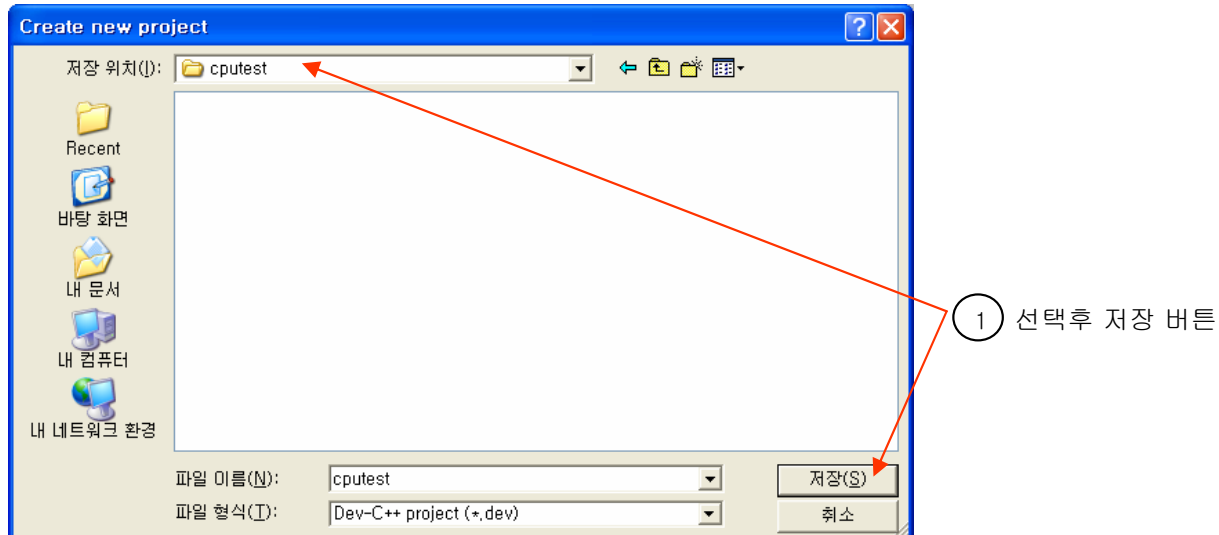
③ 확인 버튼

* mspgcc로 Project 만들기

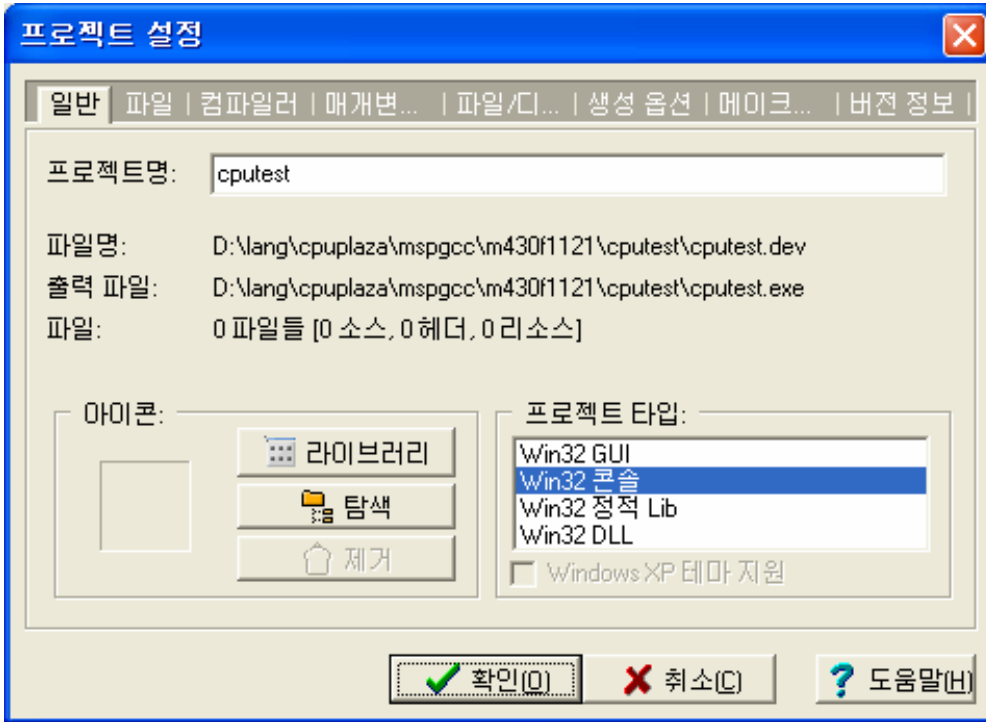
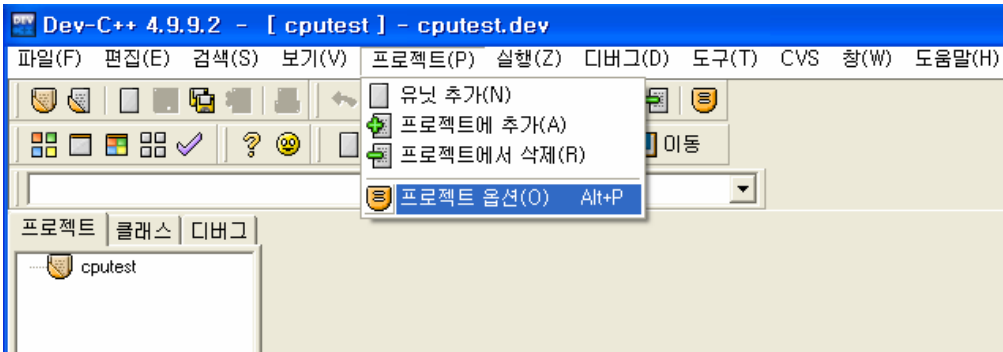
1. 프로젝트 생성(파일 -> 새로만들기 -> 프로젝트)



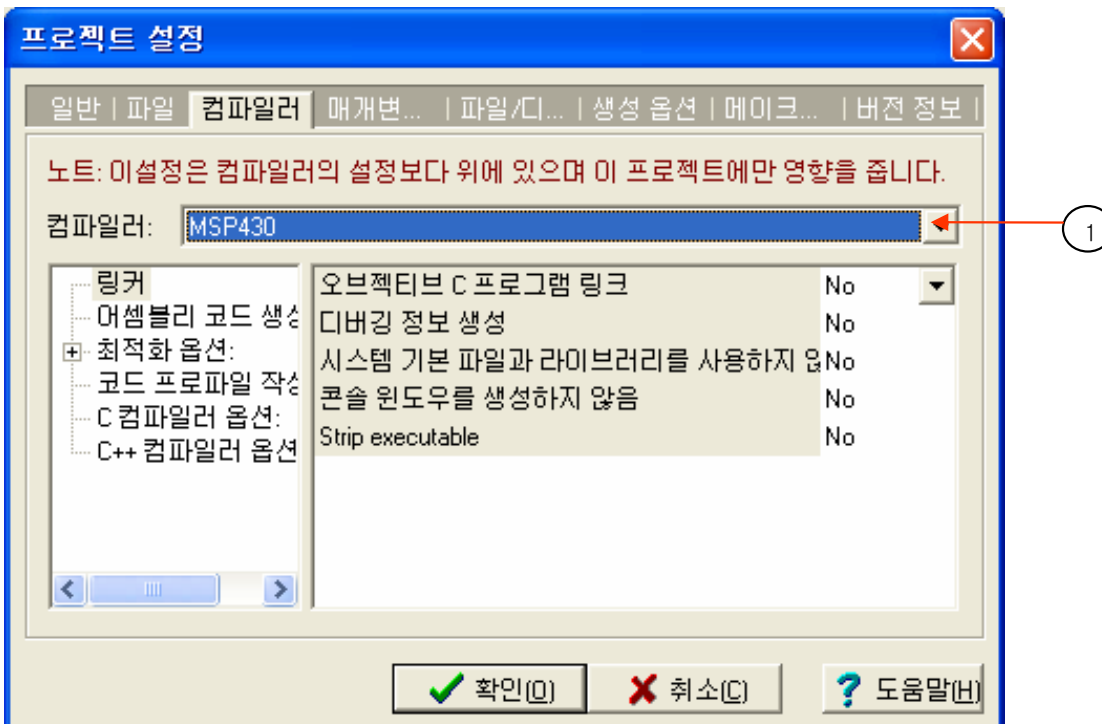
2. 프로젝트를 생성할 위치에 폴더를 만든후 저장 버튼을 클릭 한다.



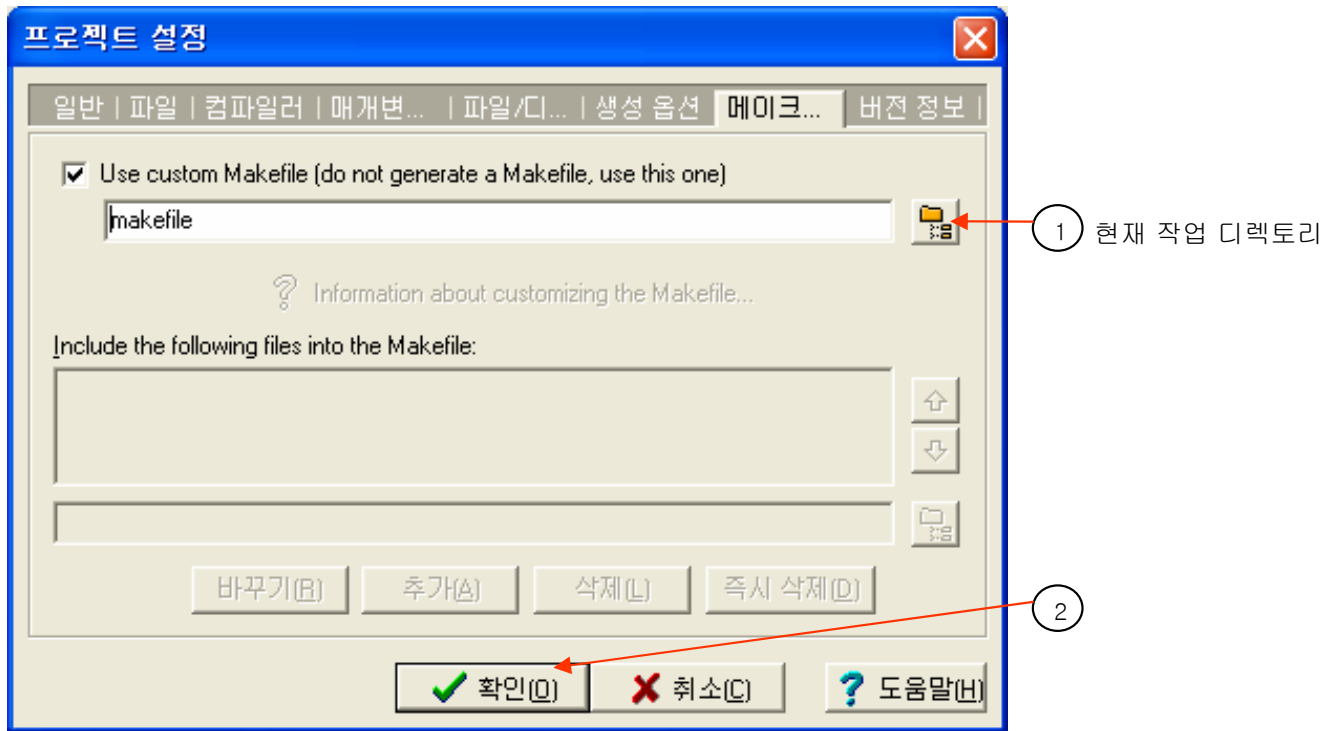
3. 프로젝트 옵션 설정(프로젝트 -> 프로젝트 옵션)



* 도구 -> 컴파일러 설정에서 지정한 명칭을 선택 한다.



* 현재 작업 디렉토리에 있는 makefile을 등록 한다.



* Makefile에서 mmcpu 옵션 설정시 CPU 선언 종류

mcp1	mcp2			
mcp430x110	mcp430x112			
mcp430x1101				
mcp430x1111	mcp430x1121			
mcp430x122	mcp430x123			
mcp430x1222	mcp430x1232			
mcp430x133	mcp430x135			
mcp430x1331	mcp430x1351			
mcp430x147	mcp430x148	mcp430x149		
mcp430x1471	mcp430x1481	mcp430x1491		
mcp430x155	mcp430x156	mcp430x157		
mcp430x167	mcp430x168	mcp430x169	mcp430x1610	mcp430x1611
mcp430x311	mcp430x312	mcp430x313	mcp430x314	mcp430x315
mcp430x323	mcp430x325	mcp430x336	mcp430x337	

* Makefile 설명 및 프로젝트에 적용시 사용자 수정 위치

```
# makfile configuration
NAME      = test
OBJECTS   = test.o
CPU       = msp430x1121

CFLAGS    = -mmcu=${CPU} -O2 -Wall -g

#switch the compiler (for the internal make rules)
CC        = msp430-gcc

.PHONY: all FORCE clean download download-jtag download-bsl dist

#all should be the first target. it's built when make is run without args
all: ${NAME}.elf ${NAME}.a43 ${NAME}.lst

#configure the next line if you want to use the serial download
download: download-jtag
#download: download-bsl

#additional rules for files
${NAME}.elf: ${OBJECTS}
            ${CC} -mmcu=${CPU} -o $@ ${OBJECTS}

${NAME}.a43: ${NAME}.elf
            msp430-objcopy -O ihex $^ $@

${NAME}.lst: ${NAME}.elf
            msp430-objdump -dSt $^ >$@

download-jtag: all
            msp430-jtag -e ${NAME}.elf

download-bsl: all
            msp430-bsl -e ${NAME}.elf

clean:
            rm -f ${NAME}.elf ${NAME}.a43 ${NAME}.lst ${OBJECTS}

#backup archive
dist:
            tar czf dist.tgz *.c *.h *.txt makefile

#dummy target as dependency if something has to be build everytime
FORCE:

#project dependencies
test.o: test.c
```

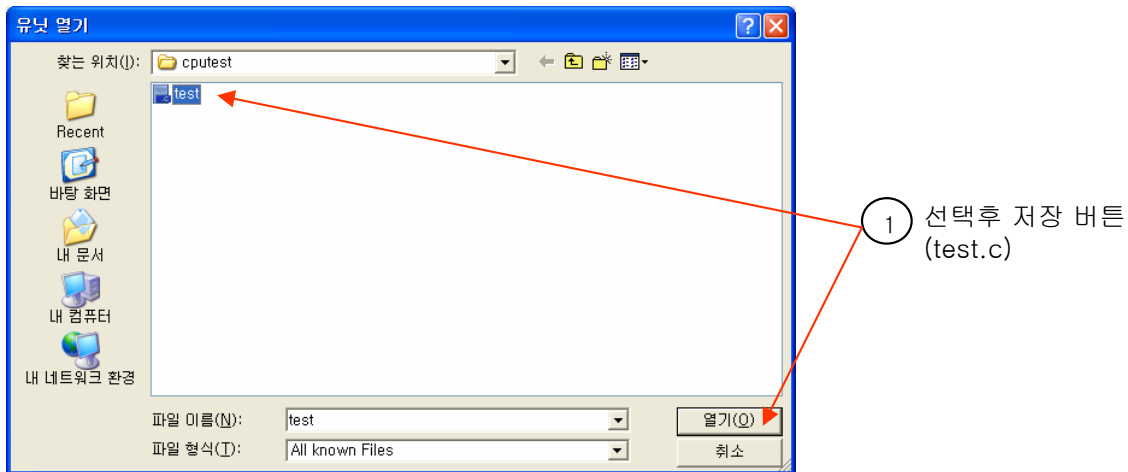
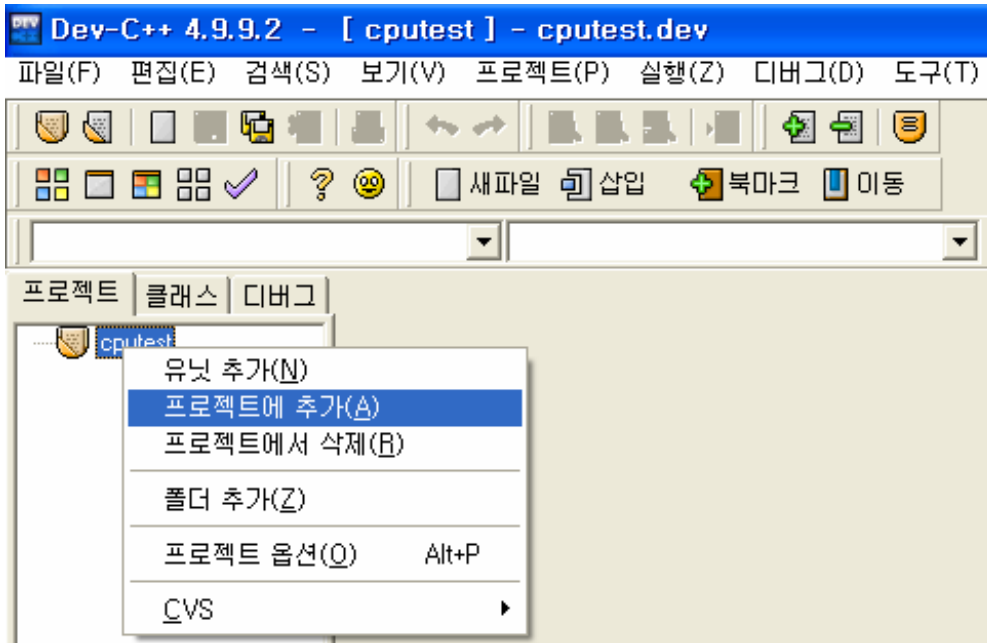
① Main() 있는 C 파일명으로

② 현재 사용된 CPU종류

컴파일 옵션
CPU 종류/최적화 레벨
/모든 에러표시/디버그

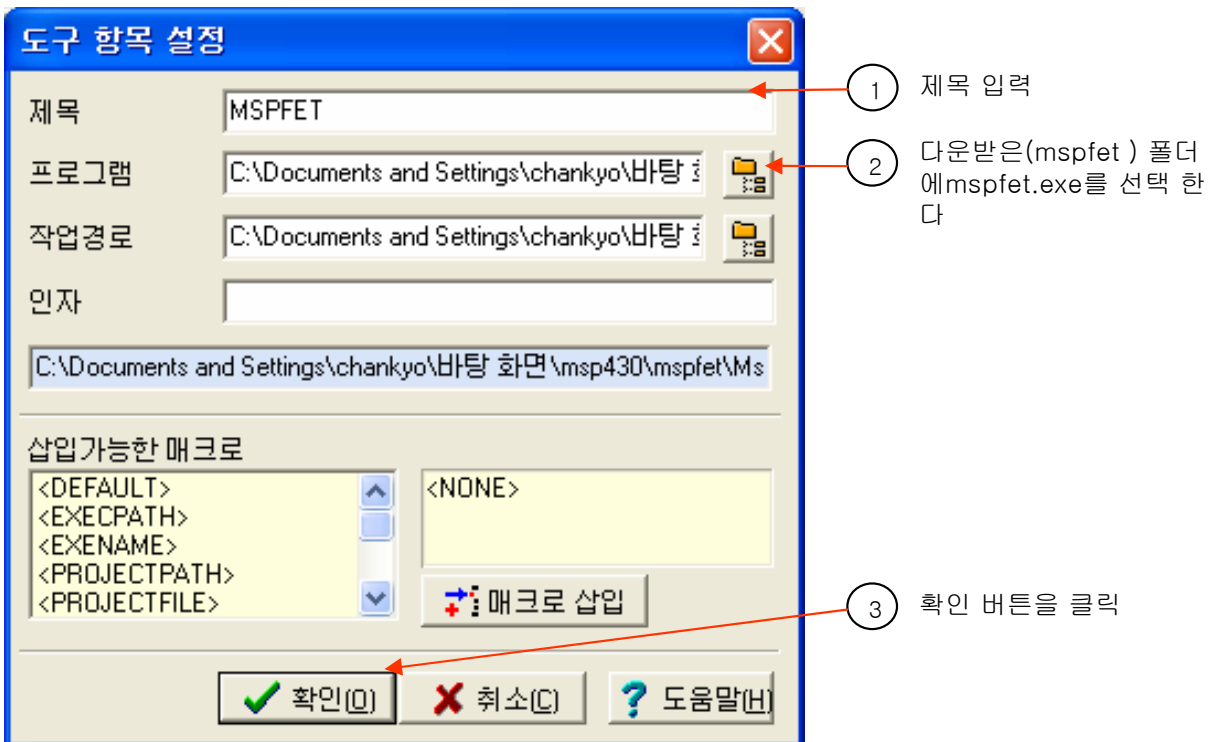
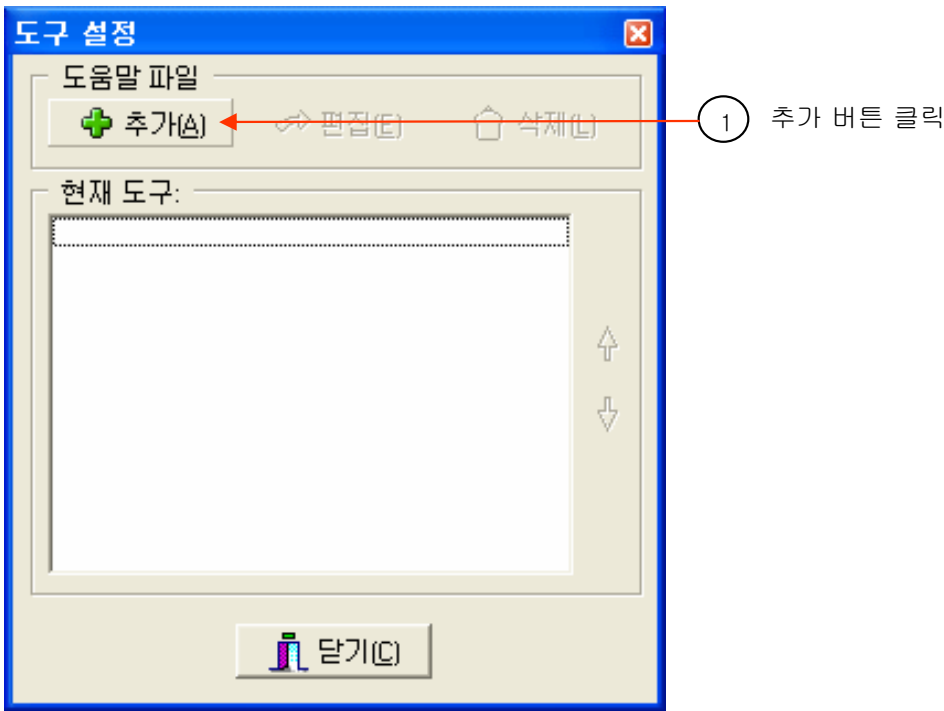
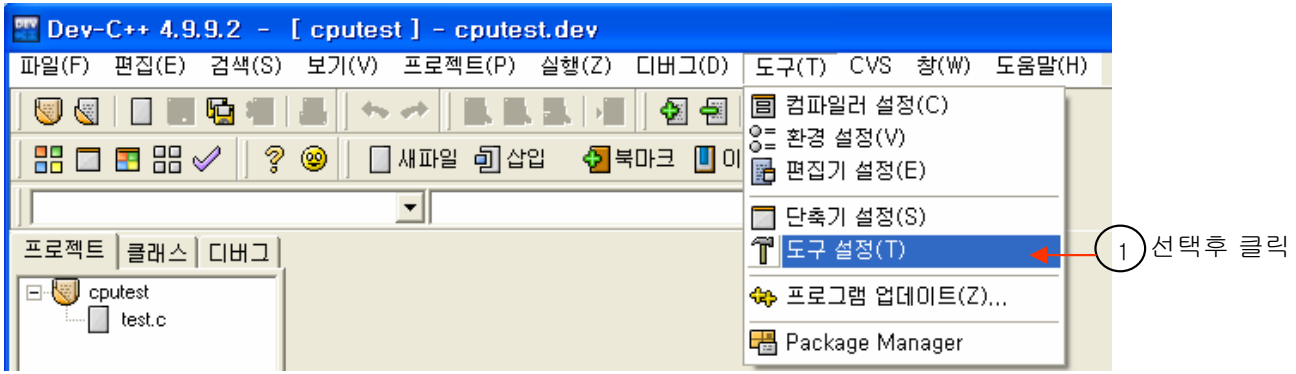
③ Main() 있는 C 파일명으로

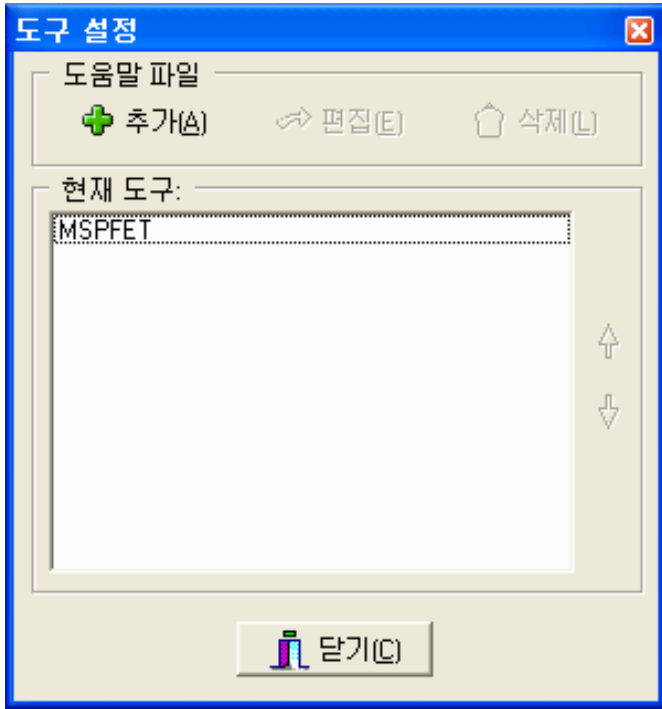
* 현재 작업 디렉토리에 있는 소스(*.c) 파일을 등록 한다.



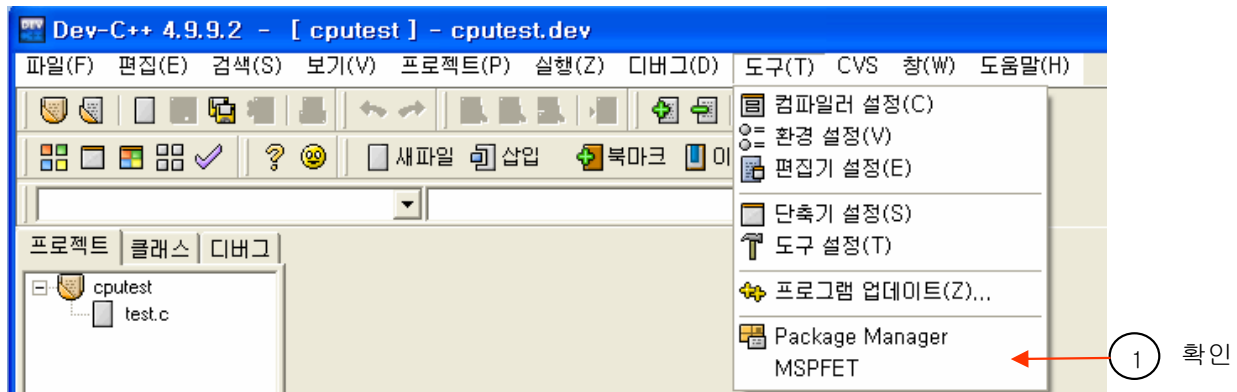
* Flash Programming Utility 연결 하기

1. Flash Writer 연결(도구 -> 도구설정)



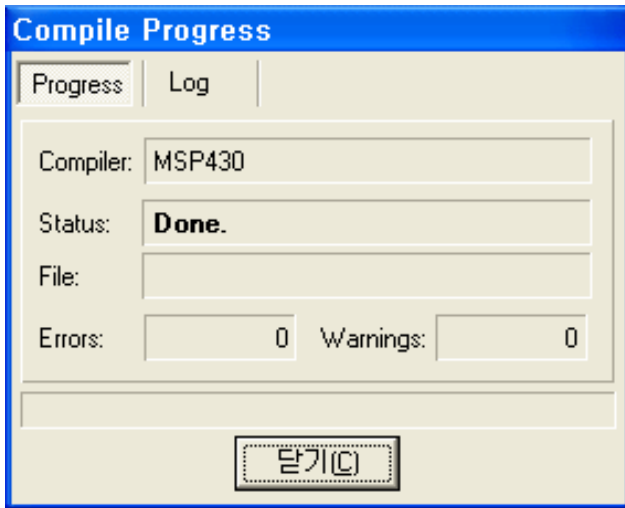
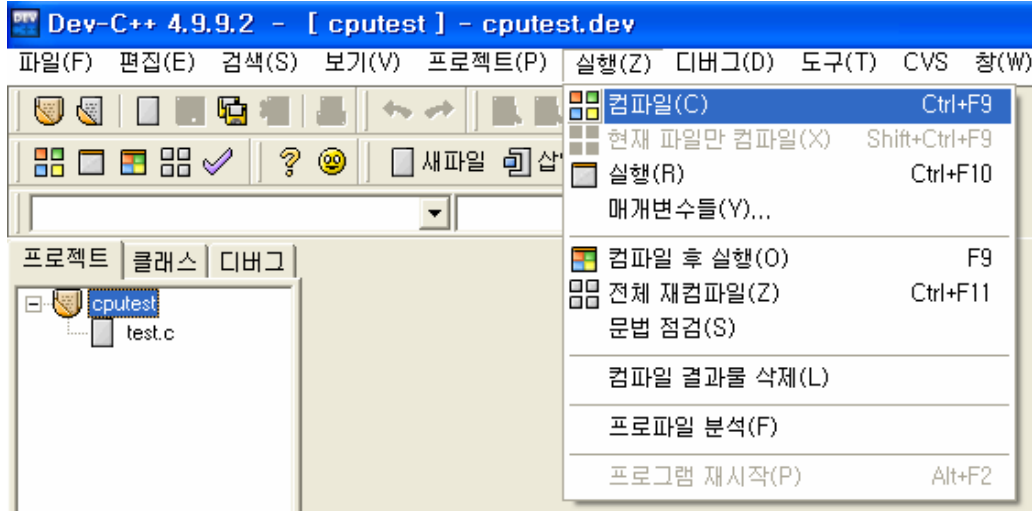


* MSPFET 도구 생성 완료 확인(도구 -> MSPFET)

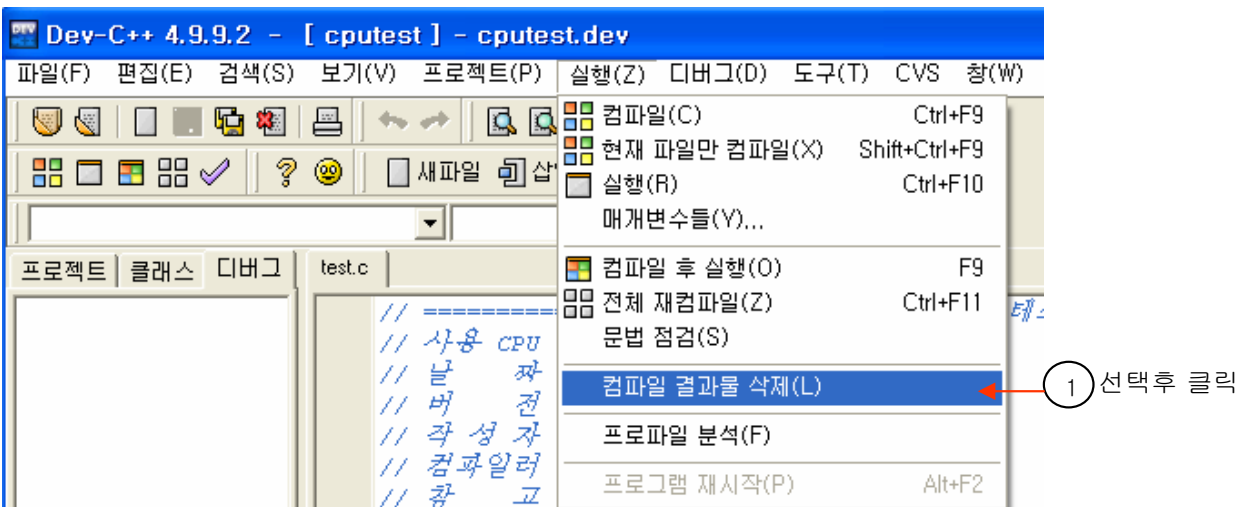


* 컴파일 및 Flash Programming

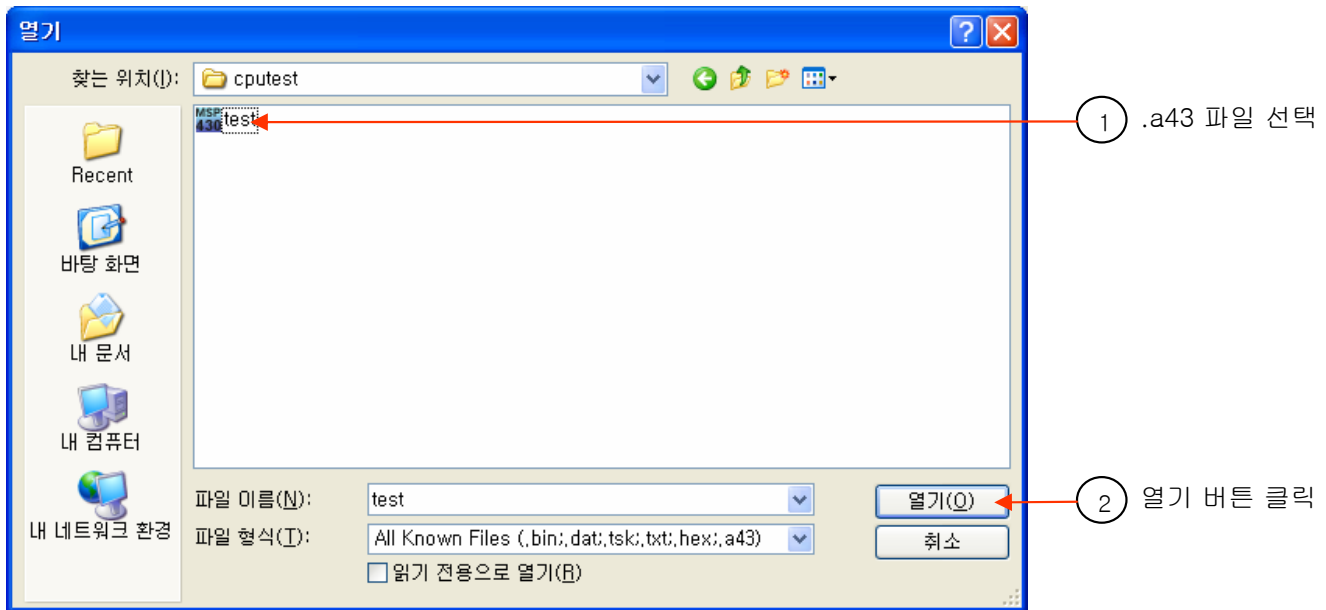
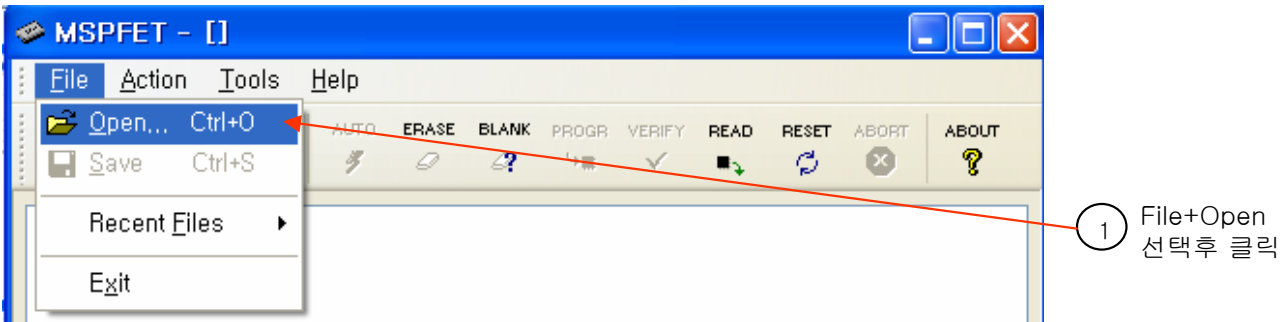
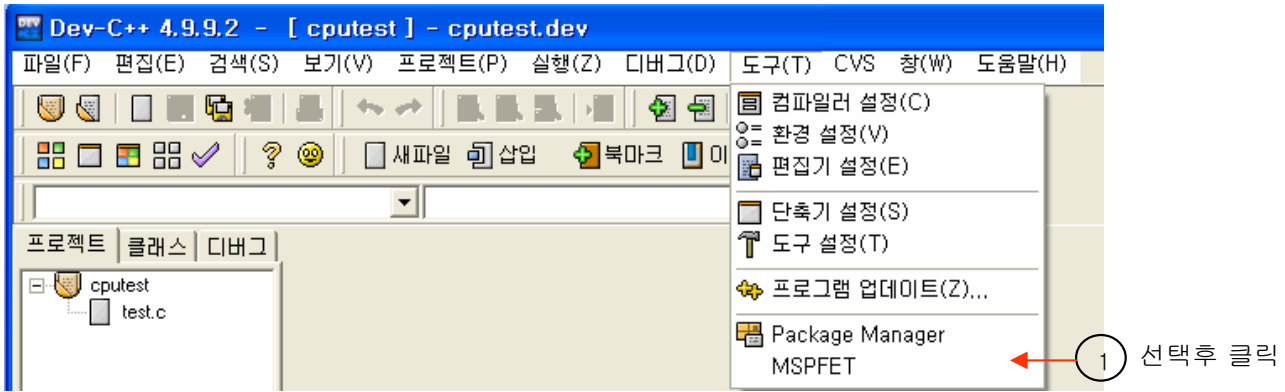
* 실행파일 생성(실행 -> 컴파일)



* 결과물 삭제시(실행 -> 컴파일 결과물 삭제)



* Flash Program(도구 -> MSPFET)



* Flash Program(도구 -> MSPFET)

The screenshot shows the MSPFET software interface. The 'Action' menu is open, highlighting the 'Program' option. The main window displays a memory dump with hexadecimal and ASCII values. The bottom status bar shows the device as 'MSP430F11x1' with '4KB+256B Flash, 256B RAM'. A log window at the bottom shows the following messages:

```
19:47:58 Blank check device memory...
19:47:59 Blank check complete.
19:47:59 Programming the device memory...
19:47:59 Program complete.
19:47:59 Verifying device memory programming...
19:47:59 Verify complete.
```

Annotations with red arrows and circles:

- ② Program: Points to the 'Program' option in the 'Action' menu.
- ① CPU 설정 및 확인: Points to the device selection dropdown menu.
- ② 메시지 확인: Points to the log window.