

C 언어 기초

목차

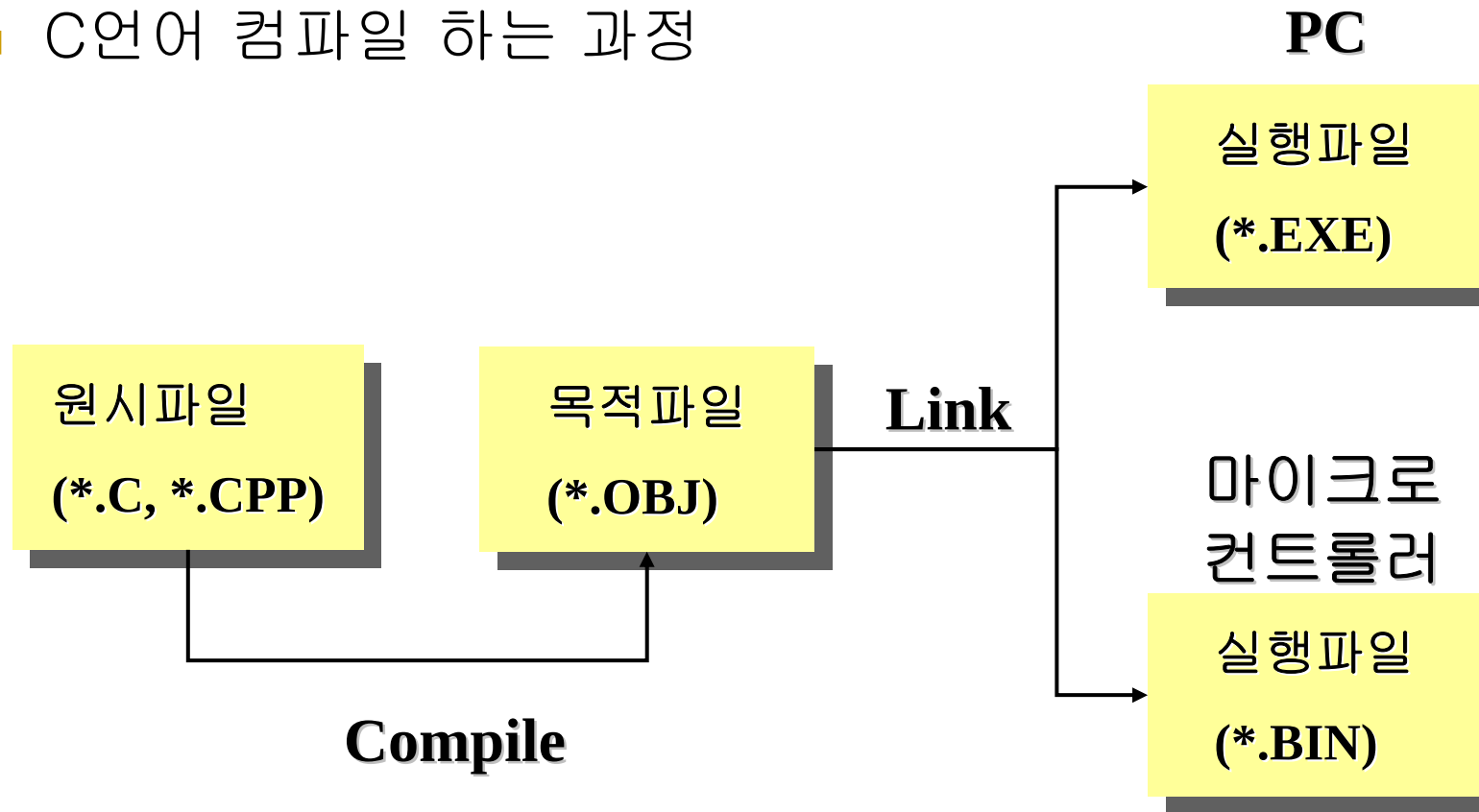
- C언어 개요
- ICG AVR 사용
- 마이크로 컨트롤러 프로그래밍
- 개발환경 구축하기
- Printf 사용하기
- 프로그래밍 시 주의 사항

C언어 개요

- C언어를 사용하는 이유?
 - 구조적 프로그래밍
 - 마이크로 컨트롤러를 위한 컴파일러가 제공 된다.
 - 포인터를 사용하여 메모리 접근이 가능하다.

C언어 개요

■ C언어 컴파일 하는 과정



C언어 개요

■ C소스 기본 구조

```
/* 파일 이름 : C1-1.C 프로그램 내용 : 화면에 메시지를 출력하는 프로그램. */
```

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    int a, b, c;
```

```
    a = 1; b = 2;
```

```
    c = addFunction(a, b);
```

```
    printf("%d = %d + %d", c, a, b);
```

```
}
```

```
Int addFunction (int ta, int tb)
```

```
{
```

```
    return (a+b);
```

```
}
```

주석

헤더파일 선언

메인 함수

변수 선언

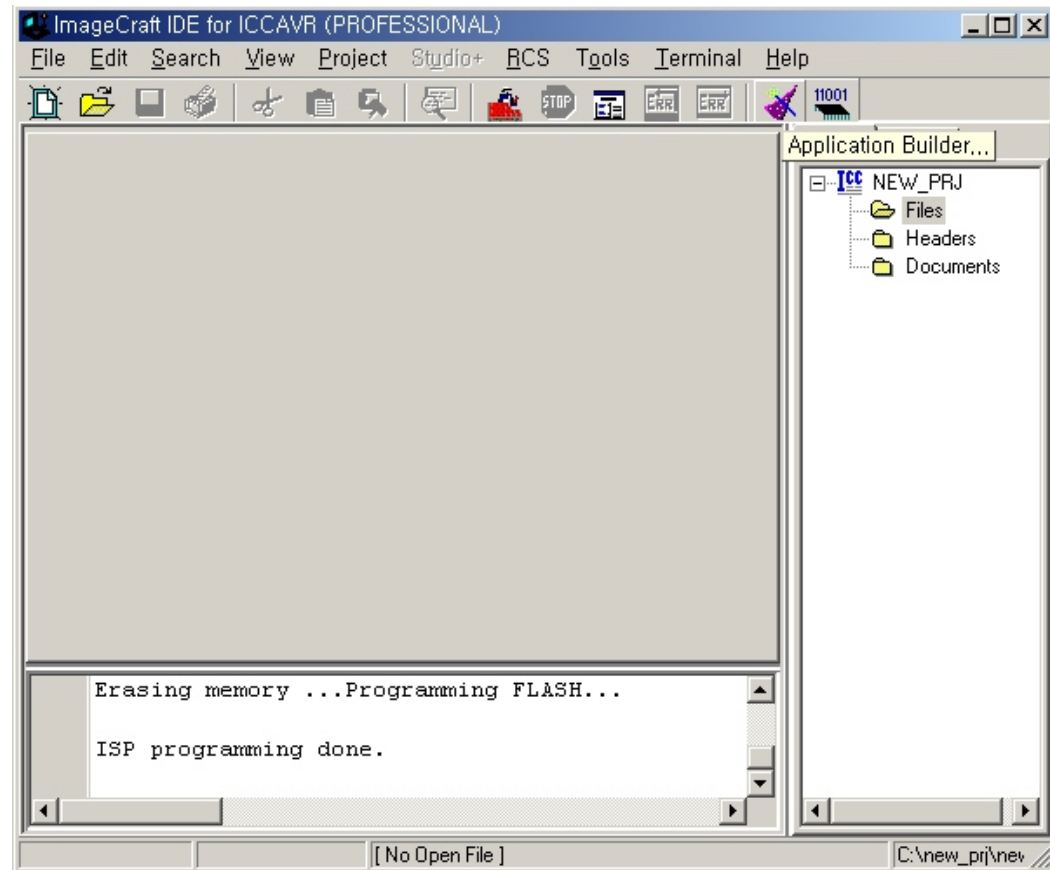
사용자 함수 호출

사용자 함수

ICC-AVR 사용

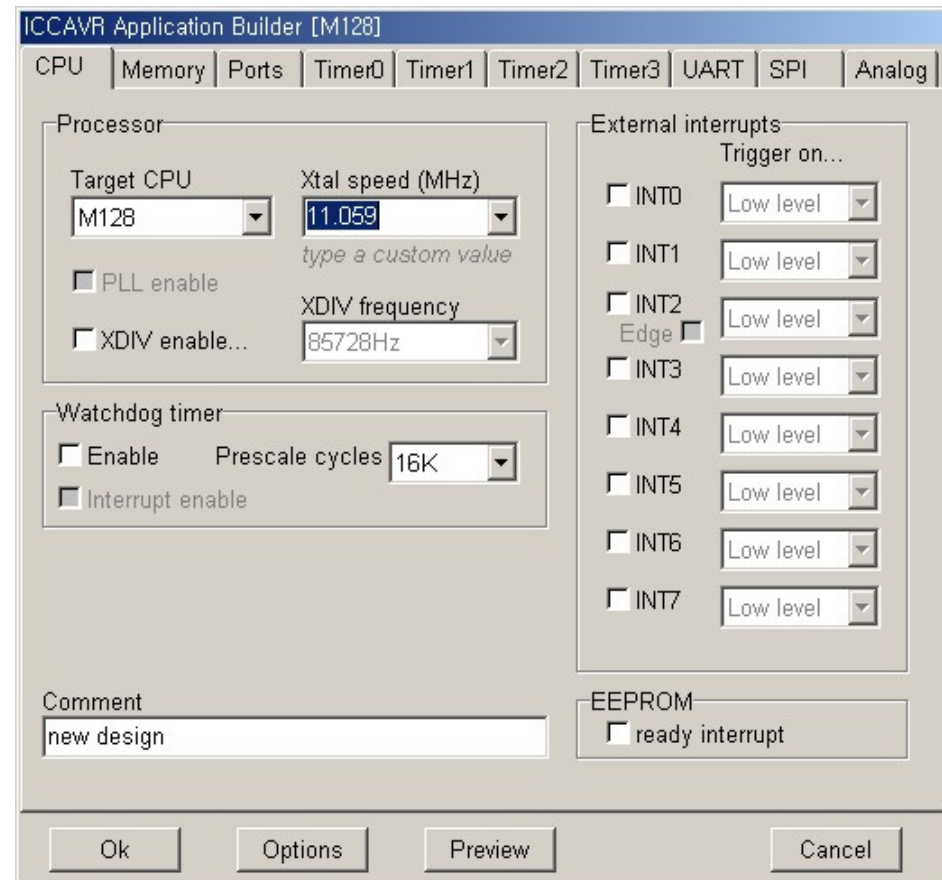
■ ICG AVR의 특징

- 통합개발환경 (IDE)
- 어플리케이션 빌더를 통한 레지스터 및 인터럽트 셋팅 기능
- 컴파일후 자동 ISP 다운로드
- 시리얼 포트를 통한 printf 제공

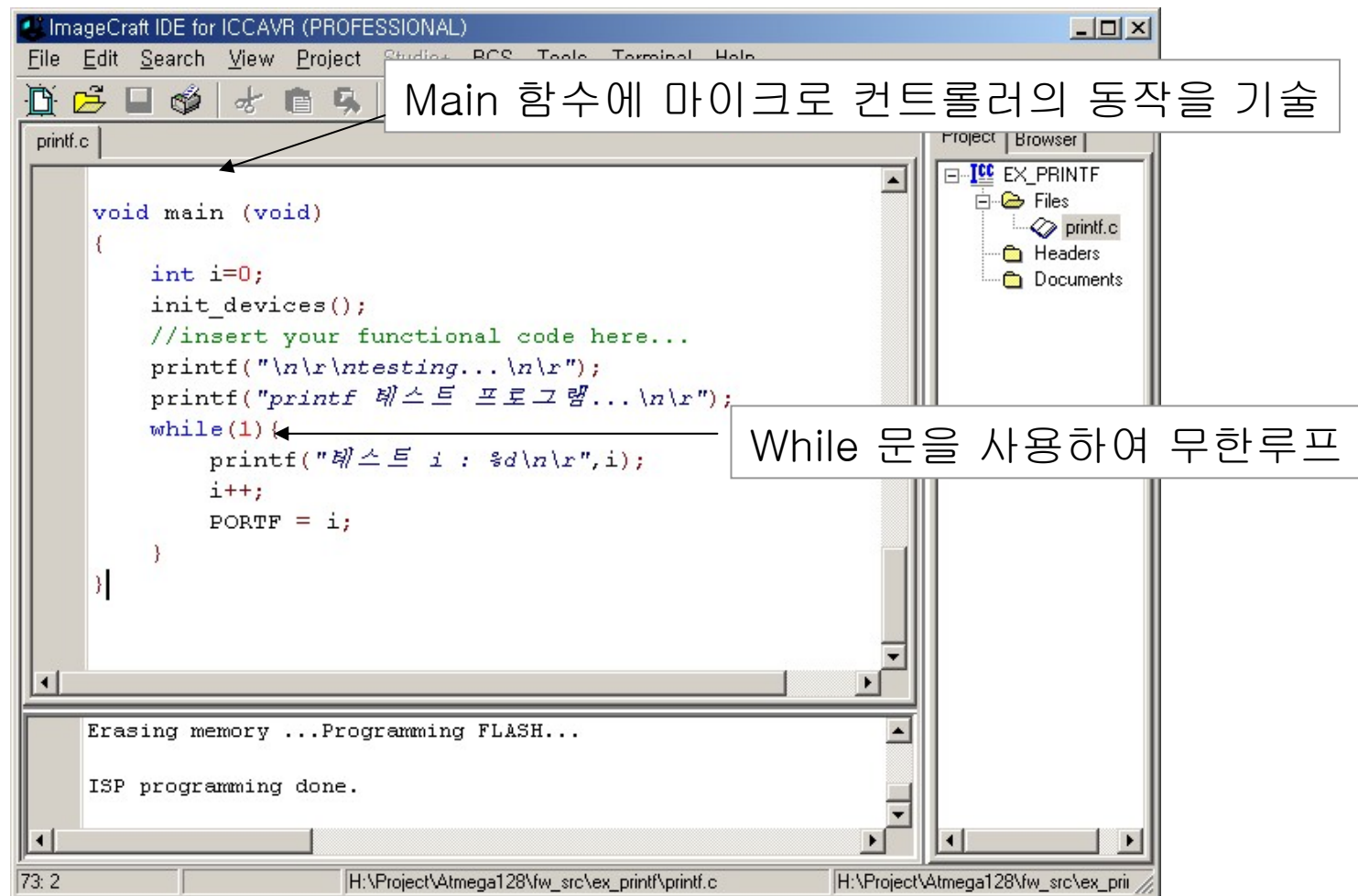


ICC-AVR 사용

- 어플리케이션 빌더
 - 각 레지스터, 인터럽트 등을 간단하게 셋팅 할 수 있다.
- “AVR 강좌- 2. Application Builder 이용하기” 참조



마이크로 컨트롤러 프로그래밍



마이크로 컨트롤러 프로그래밍

- ANSI C 기본 제어문
 - For 문
 - While 문
 - switch case 문
 - If else 문
- 포인터, 변수, 함수 등은 ANSI C 표준에 따름
- 레지스터에 값을 쓰거나 읽는 것은 8 bit 변수에 값을 쓰거나 읽는 것과 동일하다.

마이크로 컨트롤러 프로그래밍

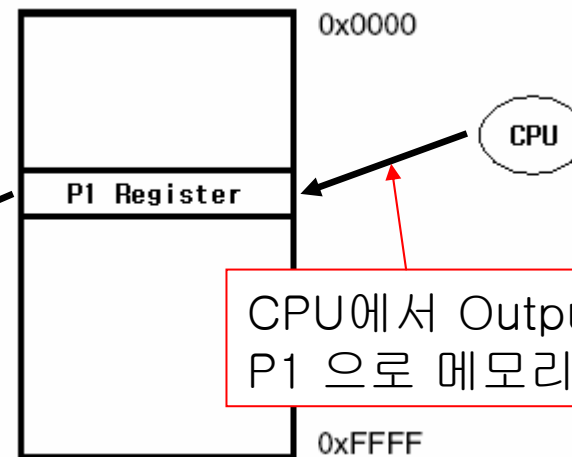
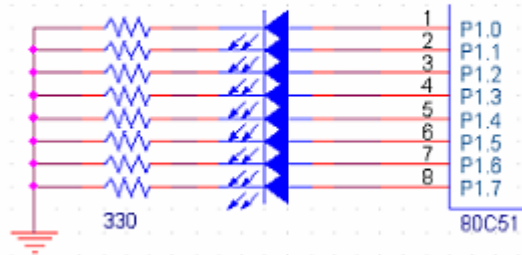
■ Example

- PORT1에 연결되어 있는 8개의 LED 를 켜기
- PORT1 에 5V 또는 0V 의 출력을 주면 연결되어 있는 LED 가 켜지거나 꺼지게 됩니다. SFR에 PORT1으로 데이터를 출력할 수 있게 하는 P1 이라는 레지스터가 있다.

C code

```
P1 = 0x55;
```

Output Register 에 쓰인 값이 하드웨어 출력으로 나간다.



CPU에서 Output Register P1 으로 메모리를 Write 한다

마이크로 컨트롤러 프로그래밍

■ 레지스터에 데이터 쓰기

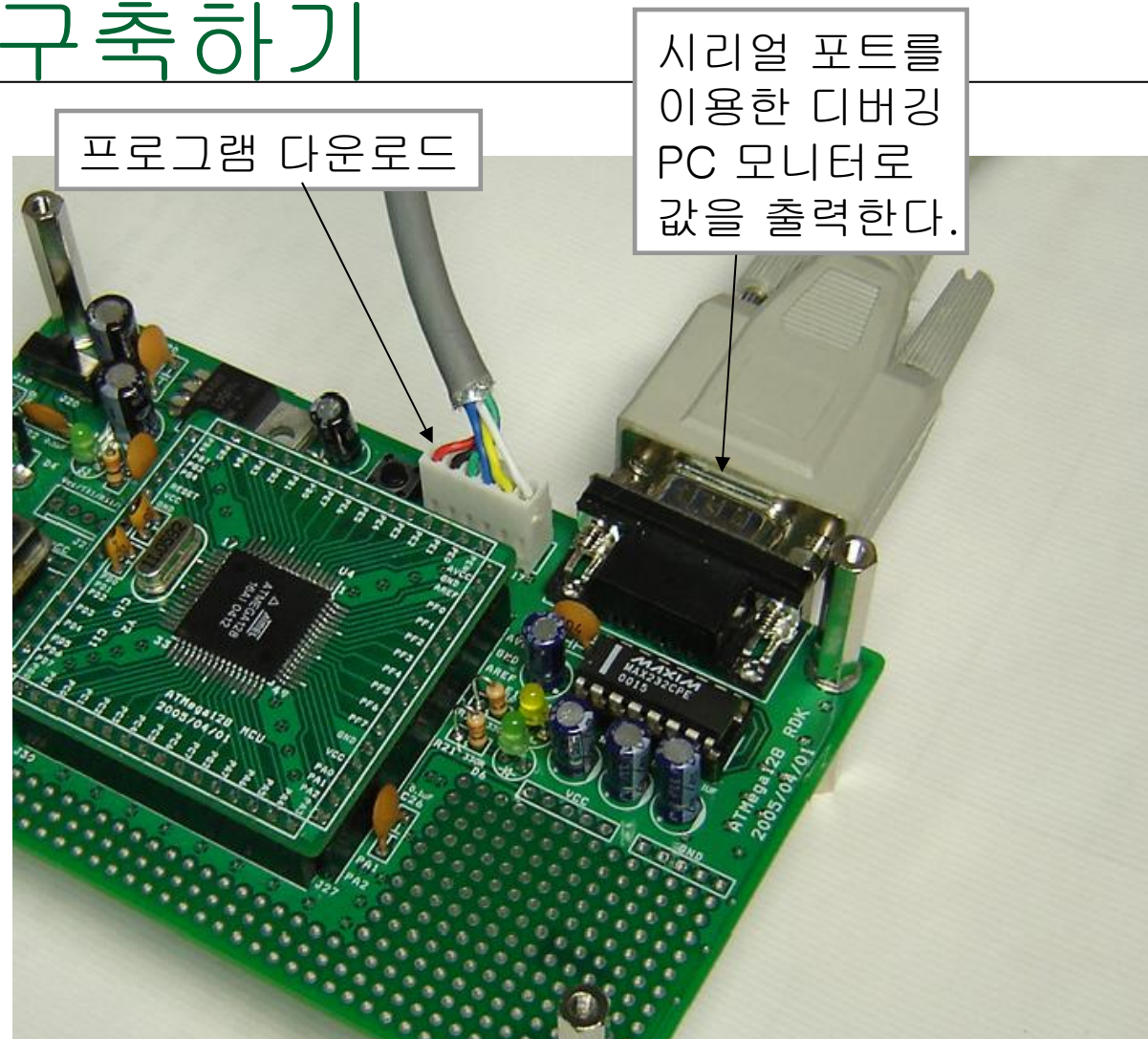
- 레지스터는 각 헤더파일에 정의 되어 있으며 8bit 변수와 같이 사용한다.
- EX) PORTA = 0x00; temp = PINA;

■ 인터럽트 핸들러 사용

```
#pragma interrupt_handler timer0_ovf_isr:17
void timer0_ovf_isr(void)
{
    TCNT0 = 0x06; //reload counter value
}
```

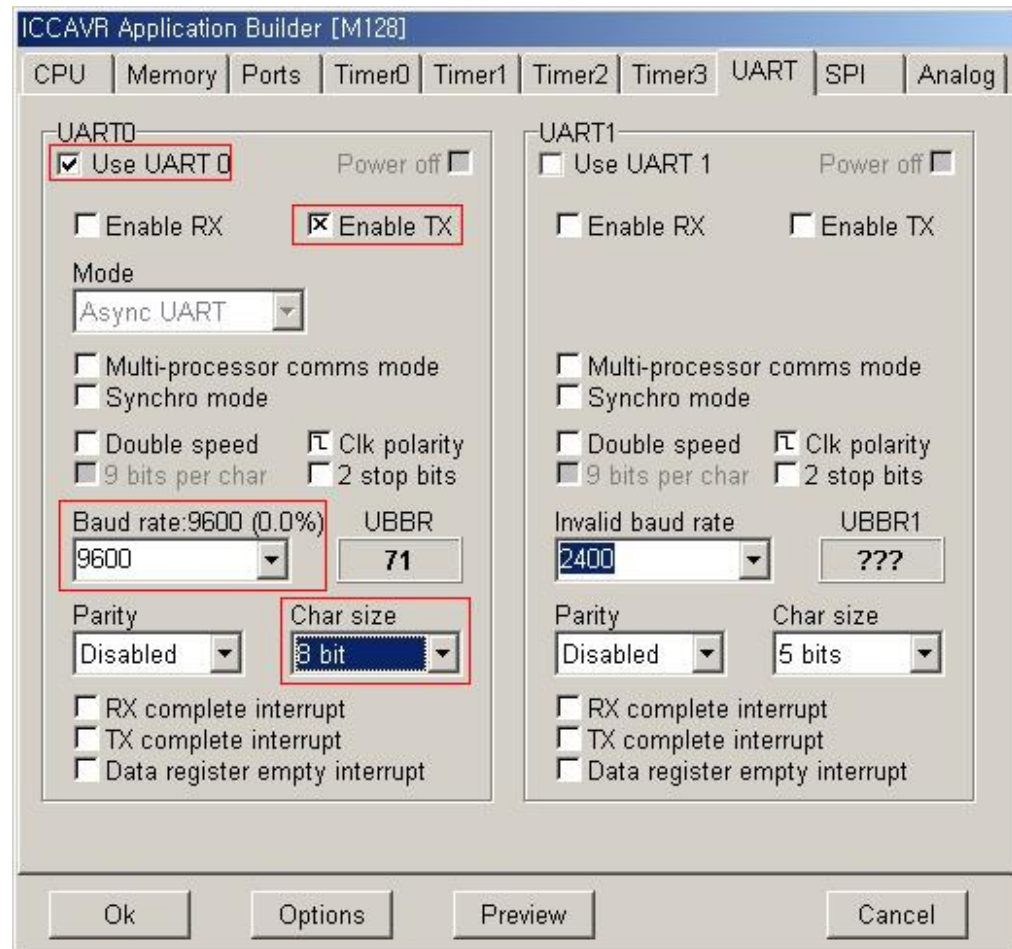
개발환경 구축하기

■ 개발환경

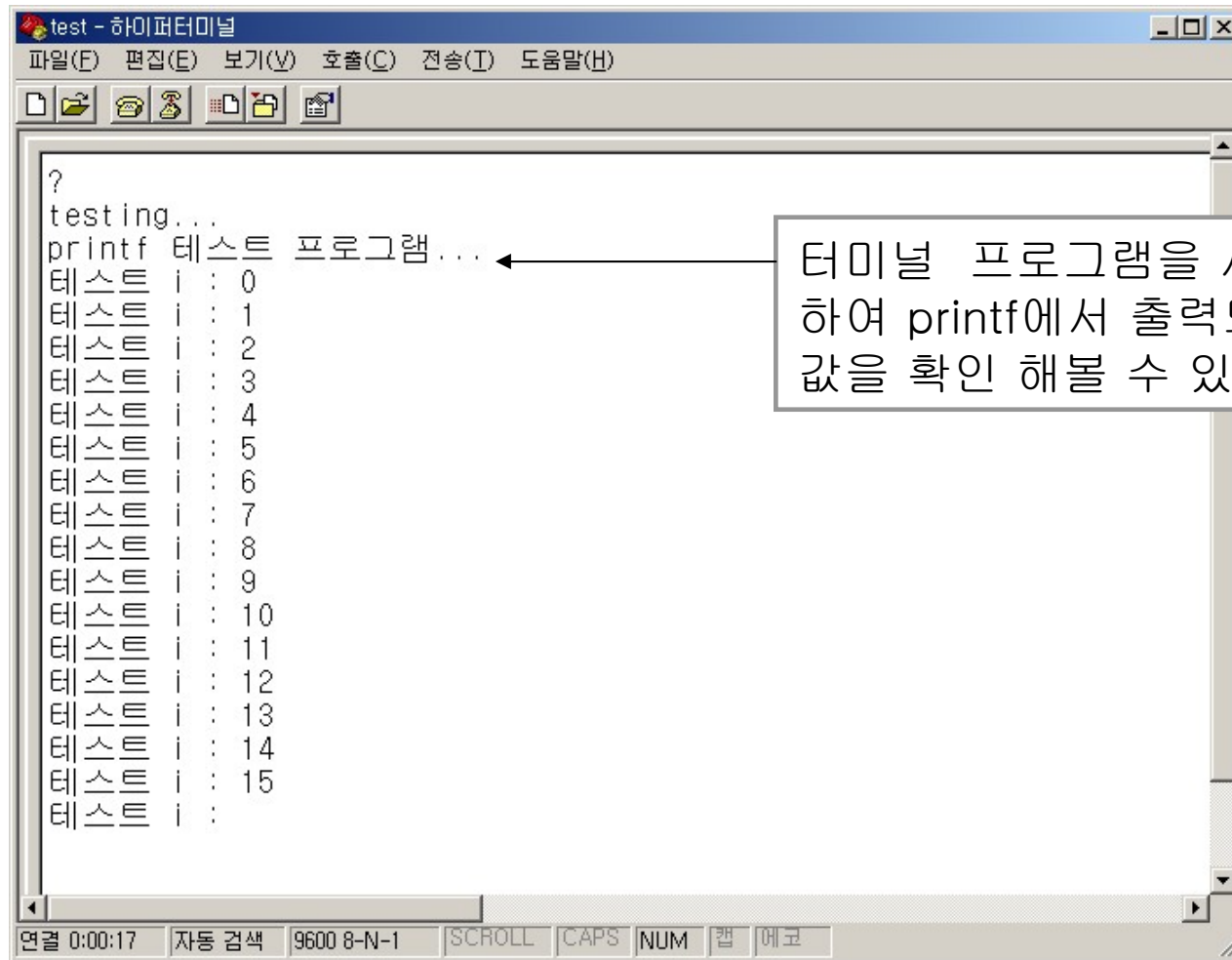


Printf 사용하기

- ICG AVR에서 기본으로 제공하는 printf를 사용하여 동작을 모니터링 하면서 개발할 수 있다.
- UART를 사용할 수 있도록 셋팅해야 한다.
- 관련된 내용은 “AVR 강좌 - 3. printf 사용하여 디버깅 하기”를 참고한다.



Printf 사용하기



```
?  
testing...  
printf 테스트 프로그램...  
테스트 i : 0  
테스트 i : 1  
테스트 i : 2  
테스트 i : 3  
테스트 i : 4  
테스트 i : 5  
테스트 i : 6  
테스트 i : 7  
테스트 i : 8  
테스트 i : 9  
테스트 i : 10  
테스트 i : 11  
테스트 i : 12  
테스트 i : 13  
테스트 i : 14  
테스트 i : 15  
테스트 i :
```

터미널 프로그램을 사용하여 printf에서 출력되는 값을 확인 해볼 수 있다.

프로그래밍 시 주의 사항

- PC 상에서 C언어를 먼저 공부하는 것이 좋다.
Visual C++ 보다는 Turbo C로 공부하는 것이 좋습니다.
- 프로그램이 시작하는 함수는 main 함수이며 소스상에 main 함수는 1개만 있을 수 있다.

프로그래밍 시 주의 사항

- 외부 메모리가 있는 경우가 아니라면 재귀호출 (Recursive Call)을 사용하지 않는 것이 좋습니다.
- 전역변수를 사용하여 인터럽트와 메인 함수가 값을 주고 받을 경우 volatile 로 선언한다.
 - EX) volatile int global_variable;
- 대용량의 값이 변하지 않는 배열(테이블)을 사용할 경우 const 키워드를 사용하여 코드 메모리에 저장한다.