

# **TI DSP, MCU 및 Xilinx Zynq FPGA**

## **프로그래밍 전문가 과정**

**강사 - Innova Lee(이상훈)**  
**[gcccompil3r@gmail.com](mailto:gcccompil3r@gmail.com)**  
**학생 - 하성용**  
**[accept0108@naver.com](mailto:accept0108@naver.com)**

70 일차

```
#include <iostream>
using namespace std;

class Counter
{
    int val;
public:
    Counter(void)
    {
        val=0;
    }
    void Print(void)
    {
        cout<<val<<endl;
    }
    friend void SetVal(Counter& c, int val);
};

void SetVal(Counter& c, int val)
{
    c.val=val;
}

int main(void)
{
    Counter cnt;
    cnt.Print();
    SetVal(cnt, 2002);
    cnt.Print();
    return 0;
}
```

```
yong@yong-Z20NH-A551B5U:~/test/70$ ./a.out
0
2002
```

---

```
#include<iostream>
using namespace std;

class A
{
    private:
        int data;
    friend class B;
};

class B
{
    public:
        void SetData(A&a,int data)
        {
            a.data=data;
        }
};

int main(void)
{
    A a;
```

```

        B b;
        b.SetData(a, 10);
        return 0;
}

```

---

```

#include<iostream>
using namespace std;

```

```

class A
{
    public:
        A(void)
        {
            cout<<"A() Call"<<endl;
        }
        A(int i)
        {
            cout<<"A(int i)Call"<<endl;
        }
        A(const A& a)
        {
            cout<<"A(const(A& a)Call"<<endl;
        }
};

```

```

int main(void)
{
    A obj1;
    A obj2(10);
    A obj3(obj2);

    return 0;
}

```

```

yong@yong-Z20NH-AS51B5U:~/test/70$ ./a.out
A() Call
A(int i)Call
A(const(A& a)Call

```

---

//복사생성자  
 p1 에 10 이랑 20 들어감  
 p2 에 p1 을 집어넣음  
 p2 에는 10 20 이 들어있어야함  
 나오는거 확인

```

#include<iostream>
using namespace std;

```

```

class Point:
{
    int x, y;
public:
    Point(int_x, int_y)
    {
        x=_x;
    }
}

```

```

        y=_y;
    }
    void ShowData(void)
    {
        cout<<x<<"<<y<<endl;
    }
};

int main(void)
{
    Point p1(10,20);
    Point p2(p1);

    p1.ShowData();
    p2.ShowData();
    return 0;
}

```

-----

(const Person& p)  
이게 정의가 안되어있음

디스트럭터?  
디스트럭터는 return 0 과 함께 동작

```

#include<iostream>
using namespace std;

```

```

class Person
{
    char *name;
    char *phone;
public:
    Person(char *_name,char *_phone);
    Person(const Person& p);
    ~Person();
    void ShowData();
};

```

```

Person::Person(char *_name,char *_phone)
{
    name = new char[strlen(_name) + 1];
    strcpy(name, _name);

    phone=new char[strlen(_phone) + 1];
    strcpy(phone, _phone);
}

```

```

Person::~~Person(void)
{
    delete []name;
    delete []phone;
}

```

```

Person::Person(const Person& p)
{
    name=new char[strlen(p.name) + 1];

```

```

        strcpy(name, p.name);
        phone=new char[strlen(p.phone) + 1];
        strcpy(phone, p.phone);
    }

void Person::ShowData(void)
{
    cout<<"name:"<<name<<endl;
    cout<<"phone:"<<phone<<endl;
}
int main(void)
{
    Person p1("Jo","011-9272-6523");
    Person p2=p1;

    return 0;
}

```

---

```

#include<iostream>
using namespace std;

class A
{
    int val;
public:
    A(int i)
    {
        cout<<"A(int i)Call"<<endl;
        val=i;
    }
    A(const A& a)
    {
        cout<<"A(const A& a)Call"<<endl;
        val=a.val;
    }
    void ShowData(void)
    {
        cout<<"val:"<<val<<endl;
    }
};

void function(A a)
{
    a.ShowData();
}

int main(void)
{
    A obj(30);
    function(obj);
    return 0;
}

```

평선 동작시키면 Call~들 나오고  
구동시켜보면 쇼데이타에 의해 데이타가 뿌려짐

```
yong@yong-Z20NH-AS51B5U:~/test/70$ ./a.out
A(int i)Call
A(const A& a)Call
val:30
```

-----  
3 번째케이스

A a(10) 해서 디폴트생성자

맨처음 동작시키면 endl 나오고 endl

//디폴트생성자랑 복사생성자 두개나올것임

```
#include<iostream>
using namespace std;
```

```
class A
{
    int val;
public:
    A(int i)
    {
        cout<<"A(int i)Call"<<endl;
        val=i;
    }
    A(const A&a)
    {
        cout<<"A(const A&a)Call"<<endl;
        val=a.val;
    }
    void ShowData(void)
    {
        cout<<"val:"<<val<<endl;
    }
};
```

```
A function(void)
{
    A a(10);
    return a;
}
```

```
int main(void)
{
    function();
    return 0;
}
```

```
yong@yong-Z20NH-AS51B5U:~/test/70$ ./a.out
A(int i)Call
```

```
#include<iostream>
using namespace std;
```

```
class A
{
    int val;
public:
    A(int i)
    {
```

```

        cout<<"A(int i)Call"<<endl;
        val=i;
    }
    A(const A&a)
    {
        cout<<"A(const A&a)Call"<<endl;
        val=a.val;
    }
    void ShowData(void)
    {
        cout<<"val:"<<val<<endl;
    }
};

A function(void)
{
    A a(10);
    return a;
}

int main(void)
{
    function();
    return 0;
}

```

-----  
상속을 사용하는이유?

1. 우선

학생  
군인  
교수  
경찰

사람이라는 범주에 속함  
이름, 나이, 직업

이름나이이름나이 적기 번거로우니  
상속을 함으로써 번거로움을 최소화

이게 바로 재활용성

/..  
사람을 학생으로 묶기

함수뒤에 const 붙이는거  
함수 오버로딩할때 써먹을수있음  
그것이 안붙으면 오버로딩이 된다 안된다 같이

메인코드보면 스튜던트 팩이라는게보임  
디폴트생성자가 동작해야함  
퍼블릭에 스튜던트가있음  
메인제어에 컴퓨터사이언스등록하고  
그리고 초대이타하면 에이지 메이지 어쩌구 뿌리는데  
그런정보 정하지않아서 없음  
디폴트 파라미터 들어간애들 나오니까  
컴퓨터 사이언스 어쩌구 나오는거

그리고 상속이라는건 이런식으로하느것  
클래스 필슨이라고되어있는거 모든내용을 스튜던트가 사용할수있다는것

```
include<iostream>
using namespace std;

class Person
{
    int age;
    char name[20];
public:
    int GetAge(void) const
    {
        return age;
    }
    const char *GetName(void) const
    {
        return name;
    }
    Person(int _age = 1, char *_name = "noname")
    {
        age= _age;
        strcpy(name,_name);
    }
};

class Student:: public Person
{
    char major[20];
public:
    Student(char *_major)
    {
        strcpy(major, _major);
    }
    const char *GetMajor(void) const
    {
        return major;
    }
    void ShowData(void) const
    {
        cout<<"name:"<<GetName()<<endl;
        cout<<"age:"<<GetAge()<<endl;
        cout<<"major:"<<GetMajor()<<endl;
    }
};

int main(void)
{
    Student Park("Computer Science");
    Park.ShowData();
    return 0;
}
```

---



dsp

설치

cd ti-processor-sdk-linux-am57xx-evm-04.03.00.05/

sd 카드를 먼저 꽂고

setup.sh

cd ti-processor-sdk-linux-am57xx-evm-04.03.00.05/bin/

sudo ./create-sdcard.sh

device number 는 # 에 숫자에 sd 카드에 해당하는 용량을 선택

그답

2 번누르기

y

choose file path to install

1 번

((1-2-y-1 → 순서는 바뀔수있음)

boot 폴더 생겼을거

sd 카드 이그젝시키고 dsp 에 꽂을것

-----

4 랑 9 가 나옴

오퍼레이터

-----

++p

wja 의 데이터를 한칸씩 증가시키고싶을대

#include <iostream>

using namespace std;

class Point

{

private:

int x, y;

public:

Point(int \_x=0,int \_y=0):x(\_x),y(\_y){}

void ShowPosition(void);

Point&operator++(void);

friend Point& operator--(Point& p);

};

void Point::ShowPosition(void)

{

cout<<x<<" "<<y<<endl;

}

Point& Point::operator++(void)

{

x++;

```

        y++;
        return *this;
    }
    Point& operator--(Point& p)
    {
        p.x--;
        p.y--;
        return p;
    }

    int main(void)
    {
        Point p(3,7);
        ++p;
        p.ShowPosition();

        --p;
        p.ShowPosition();

        ++(++p);
        p.ShowPosition();

        --(--p);
        p.ShowPosition();

        return 0;
    }

```

```

yong@yong-Z20NH-AS51B5U:~/test/70$ ./a.out
48
37
59
37

```

++p 전위연산은 인자를 void 를 주자  
후위는 int 를 주자  
하고 약속을 하게됨

friend 붙어있는건 예외

```

Point& Point::operator++(void)
{
    x++;
    y++;
    return *this;
}

```

\*this 는 자기자신

p 자기 자신 그래서 값이 4 랑 8 이나옴

클래스외부에서 야 재 내친구야 하고 외부에서 동작하는거

friend 니까 --operator  
인자로 p 받고  
3 7  
4 8 다시받고 ++되니까  
5 9 가됨

3 7

---

```

#include<iostream>

```

```

using namespace std;

class Point
{
    private:
        int x, y;
    public:
        Point(int _x=0,int _y=0):x(_x),y(_y){}
        void ShowPosition(void);
        Point& operator++(void);
        Point operator++(int);
};

void Point::ShowPosition(void)
{
    cout<<x<<" "<<y<<endl;
}

Point& Point::operator++(void)
{
    x++;
    y++;
    return *this;
}

Point Point::operator++(int)
{
    Point temp(x,y);
    // ++(*this);
    x++;
    y++;
    return temp;
}

int main(void)
{
    Point p1(3,7);
    (p1++).ShowPosition();
    p1.ShowPosition();

    Point p2(33,77);
    (++p2).ShowPosition();
    p2.ShowPosition();

    return 0;
}

```

```

yong@yong-Z20NH-ASS1BSU:~/test/70$ ./a.out
37
48
3478
3478

```

---

동작설명

p1 에 37 들어가있음  
p1++ 이건 후위연산

주석쳐진건 x++, y++, return temp 가 결국 ++(\*this)니까  
풀어쓴거뿐

33, 77  
void 로감

프랜드가 필요한이유는

3+operator+p 이런연산이 가능하게됨

-----  
Commutative Law  
Point p1(3,7);

//x 에들어온거 더하고있고, y 에 들어온거 더하고있음

```
#include<iostream>
using namespace std;
```

```
class Point
{
    private:
        int x, y;
    public :
        Point(int _x=0, int _y =0):x(_x),y(_y) {}
        void ShowPosition(void);
        Point operator+(int val);
};
```

```
void Point::ShowPosition(void)
{
    cout<<x<<" "<<y<<endl;
}
```

```
Point Point::operator+(int val)
{
    Point temp(x+val, y+val);
    return temp;
}
```

```
int main(void)
{
    Point p1(3,7);
    Point p2=p1+3;
    p2.ShowPosition();
    return 0;
}
```

```
yong@yong-Z20NH-AS51B5U:~/test/70$ ./a.out
6 10
```

p1.operator+(3)이런얘기

-----  
3+p1 을 표현할방법이있는지  
3 은 클래스가아님  
그래서 프랜드를 써야함  
opertor+(숫자랑, 이객체) // operator 안에 숫자랑 이객체

3.opertor(p) → 안됨  
클래스가 아닌 함수로 접근하면  
p.opertor+3 값이 return p+val

```
#include<iostream>
using namespace std;

class Point
{
    private:
        int x, y;
    public:
        Point(int _x=0, int _y=0):x(_x),y(_y){}
        void ShowPosition(void);
        Point operator+(int val);
        friend Point operator+(int val, Point& p);
};

void Point::ShowPosition(void)
{
    cout<<x<<" "<<y<<endl;
}

Point Point::operator+(int val)
{
    Point temp(x + val, y+val);
    return temp;
}

Point operator+(int val, Point& p)
{
    return p+val;
}

int main(void)
{
    Point p1(3,7);
    Point p2=p1+3;
    p2.ShowPosition();

    Point p3=7+p2;
    p3.ShowPosition();
    return 0;
}
```

```
yong@yong-Z20NH-A551B5U:~/test/70$ ./a.out
6 10
13 17
```

판단해서 7+p2 는 int val, point& p 로 들어가게하고 대입하고 어찌꾸

앞에 클래스가 오게하기위해  
7 은 클래스가아님  
프렌드는 가능하게함

-----  
템플릿코드

```
#include<iostream>
using namespace std;

template <typename T>
T Add(T a, T b)
{
    return a+b;
}

int main(void)
{
    cout<<Add(10,20)<<endl;
    cout<<Add(1.1,2.2)<<endl;
    return 0;
}
```

```
yong@yong-Z20NH-AS51B5U:~/test/70$ ./a.out
30
3.3
```

타입을 알아서 지정해주는것

타입이름이 T 라는것  
Add 에 10 이랑 20 이음  
타입둘다 int  
float 이나 double  
float float  
Add template

-----

카페 AM5728 로 검색해보라

카페 메뉴중 AM5728 - 교재를 클릭  
AM5728 Face Detection

9 번 10 번 11 번 해야할것  
<http://cafe.naver.com/hestit/2055> //9 번  
<http://cafe.naver.com/hestit/2056> //10 번  
<http://cafe.naver.com/hestit/2057> //11 번

ccs 만드는방법임  
네트워크라 dsp 개발하는사람들이 모두 접근해서 쓸수있다는것  
컴퓨터에서 만든 작업들을 dsp 로 이동해서 여러가지 작업들을 할수있게됨

할때 dsp 에 랜선이 연결되어있어야함  
컴퓨터 랜선을 뽑는한이있어도 dsp 에 랜선이 연결되어있어야함

-----

9 번//  
create CCS Project

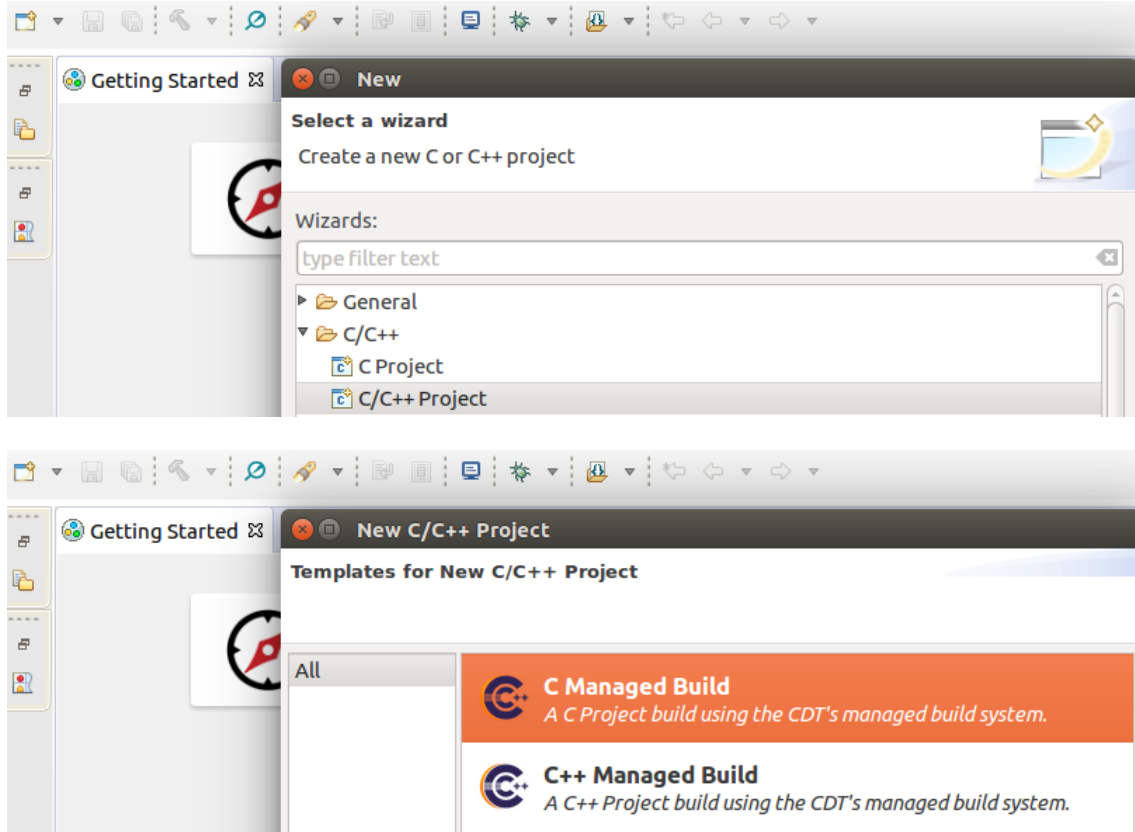
이 섹션에서는 CCS 에서 새 프로젝트를 만들고 ARM 장치 용으로 교차 컴파일하는 방법에 대해 설명합니다. 자세한 내용은 Code Composer Studio v5 사용자 설명서를 참조하십시오.

\* 새로운 크로스 컴파일 프로젝트를 만드는 방법  
\* 프로젝트에 소스를 추가하고 빌드하는 방법

## 실험 단계

1. 주 CCSv8 창에서 파일 -> 새로 작성 -> 프로젝트 ... 메뉴 항목을 선택하십시오.
2. 마법사 선택 창에서 C / C ++ -> C 프로젝트 마법사를 선택합니다.

좌측 최상단 +new icon 클릭

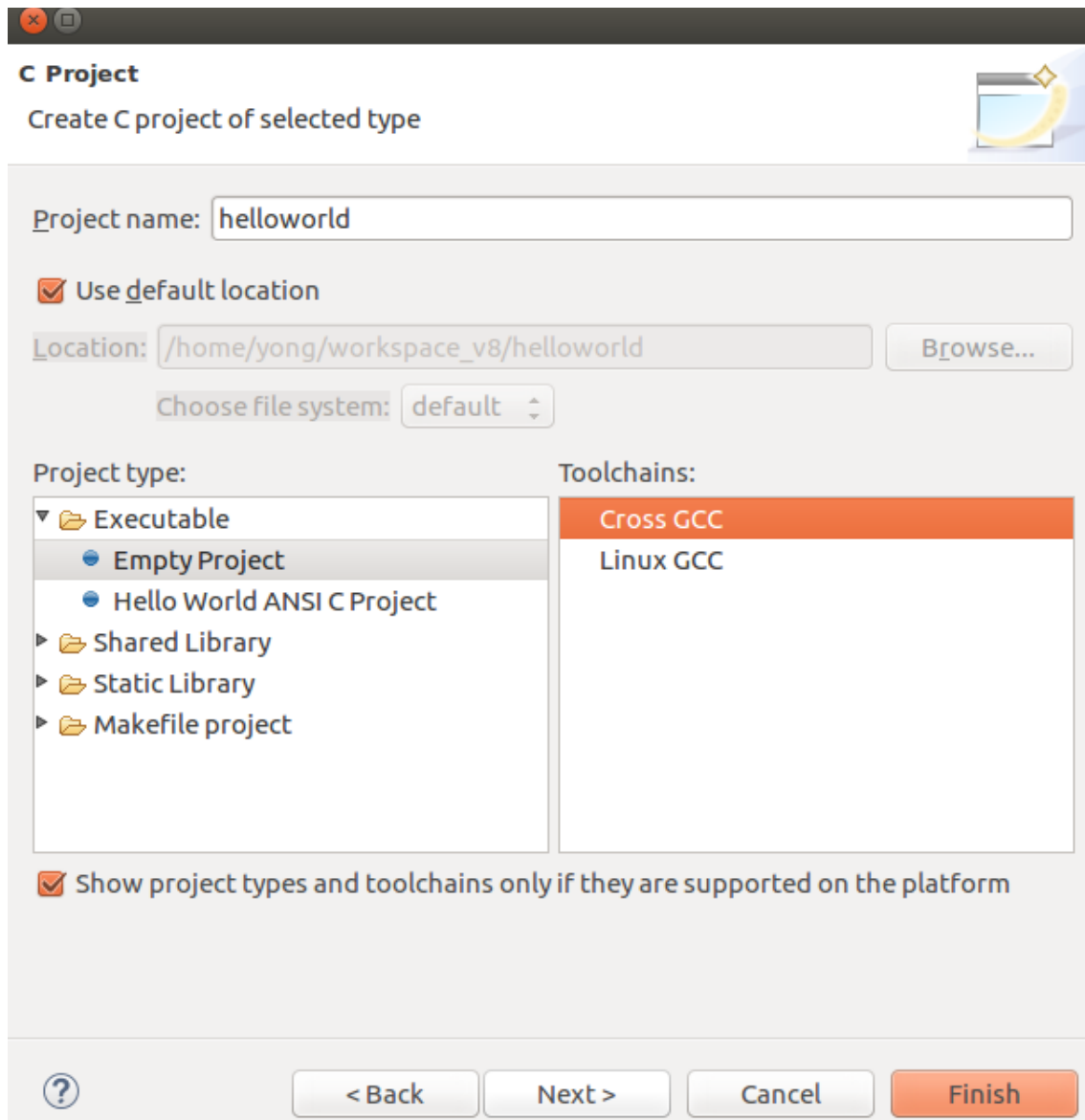


3. Click Next.
4. In the C Project dialog set the following values:  
//C 프로젝트 대화 상자에서 다음 값을 설정합니다.

Project Name: helloworld  
//프로젝트 이름 : helloworld

Project type: Executable -> Empty Project  
Toolchains: Cross GCC

3. 다음을 클릭한다.
4. C Project 대화 상자에서 아래 값을 설정한다.  
Project Name: helloworld  
Project type: Executable -> Empty Project  
Toolchains: Cross GCC



주의! Finish 는 무시하고 Next

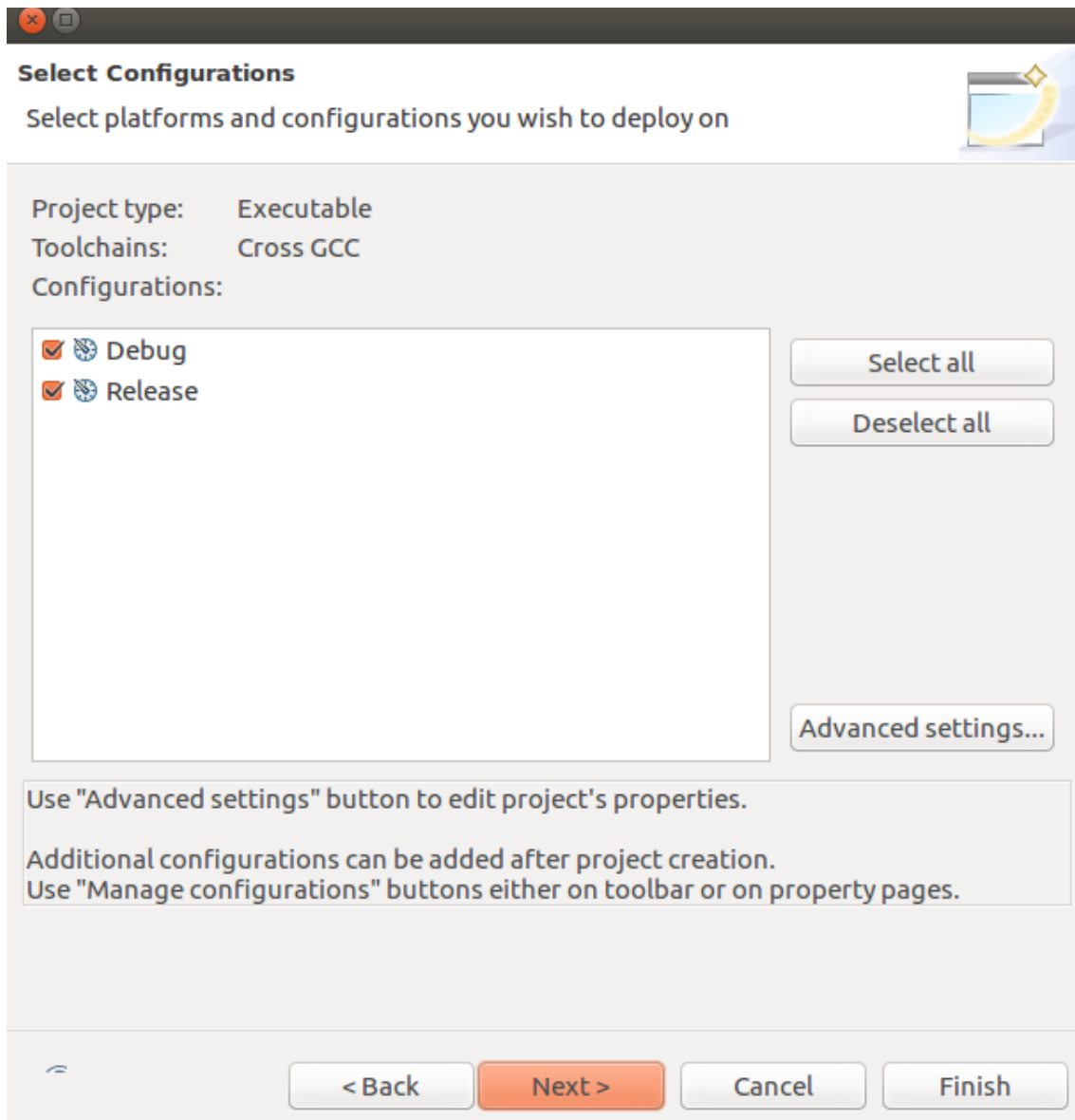
5. Click Next.

6. In the Select Configurations dialog, you can take the default Debug and Release configurations or add/remove more if you want.

5. 다음을 클릭한다.

6. Configuration 대화 상자에서 기본 Debug 및 Release 구성을 사용하거나 원하는 경우 추가/제거 할 수 있다.





7. Click Next.

8. In the Command dialog set the following values:

7. 다음을 클릭한다.

8. Command 대화 상자에서 아래 값을 설정한다:

Tool command prefix: arm-linux-gnueabihf-.

NOTE

Note the the prefix ends with a "-". This is the prefix of the cross-compiler tools as will be seen when setting the Tool command path.

접두사는 "-" 로 끝난다.

이것은 Tool 명령 경로를 설정할 때 볼 수 있는 Cross-Compiler Tool 의 접두사이다.

Tool command path: /home/sitara/ti-processor-sdk-linux-<machine>-<version>/linux-

devkit/sysroots/<Arago Linux>/usr/bin

Use the Browse.. button to browse to the Sitara Linux SDK installation directory and then to the linux-devkit/sysroots/<Arago Linux>/usr/bin directory. You should see a list of tools such as gcc with the prefix you entered above.

Tool command path: /home/계정/ti-processor-sdk-linux-<machine>-<version>/linux-devkit/sysroots/<Arago Linux>/usr/bin

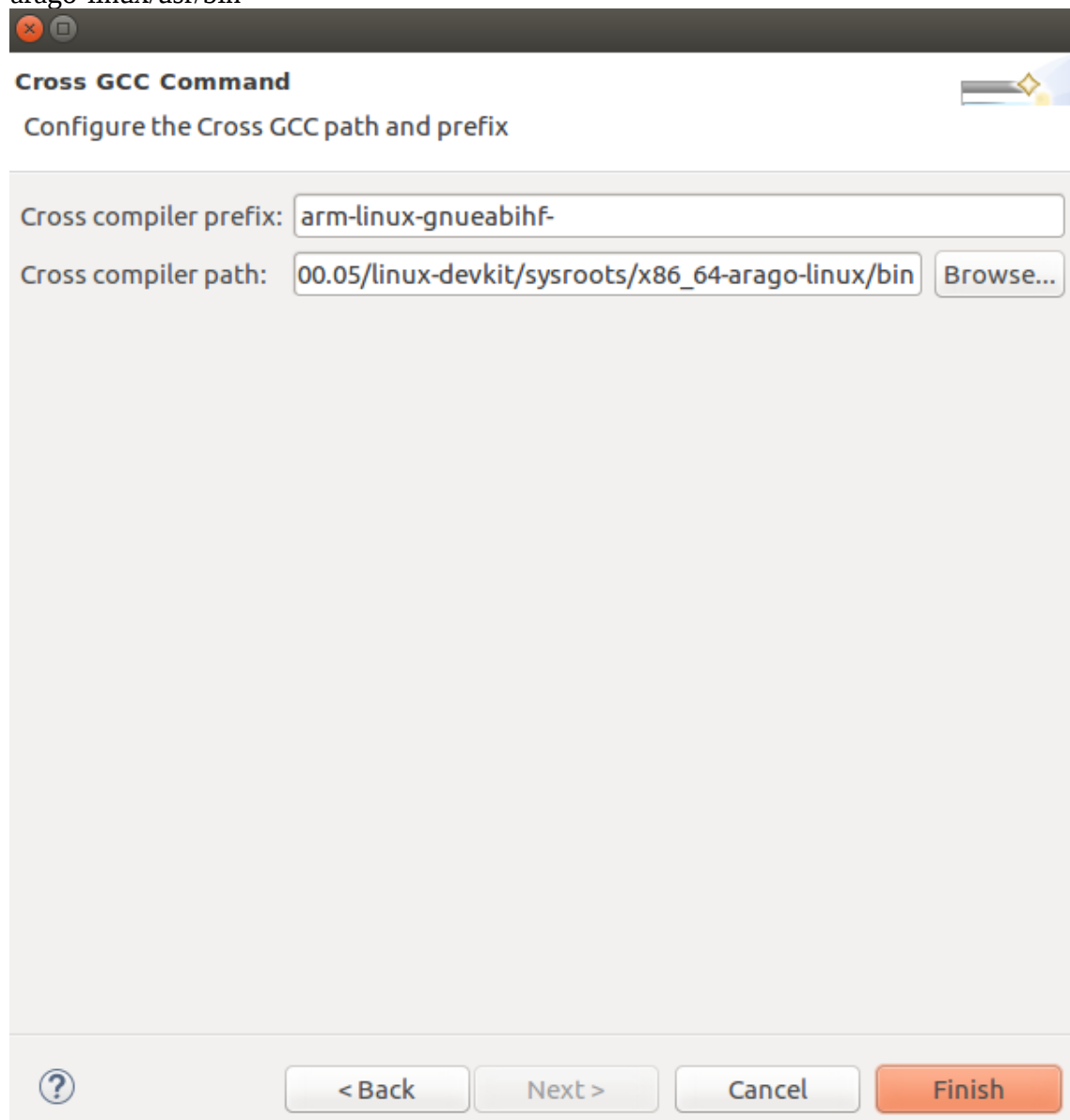
Browse 버튼을 사용하여 Sitara Linux SDK 설치 디렉토리를 찾은 다음 linux-devkit/sysroots/<Arago Linux>/usr/bin 디렉토리로 이동한다. 위에 입력한 접두어가 있는 gcc 와 같은 도구 목록이 나타난다.

//sitara 가 아니라 ti-

browse - >

/home/yong/ti-processor-sdk-linux-am57xx-evm-04.03.00.05/linux-devkit/sysroots/x86\_64-arago-linux/bin

cd ti-processor-sdk-linux-am57xx-evm-04.03.00.05/linux-devkit/sysroots/x86\_64-arago-linux/usr/bin



9. Click Finish.

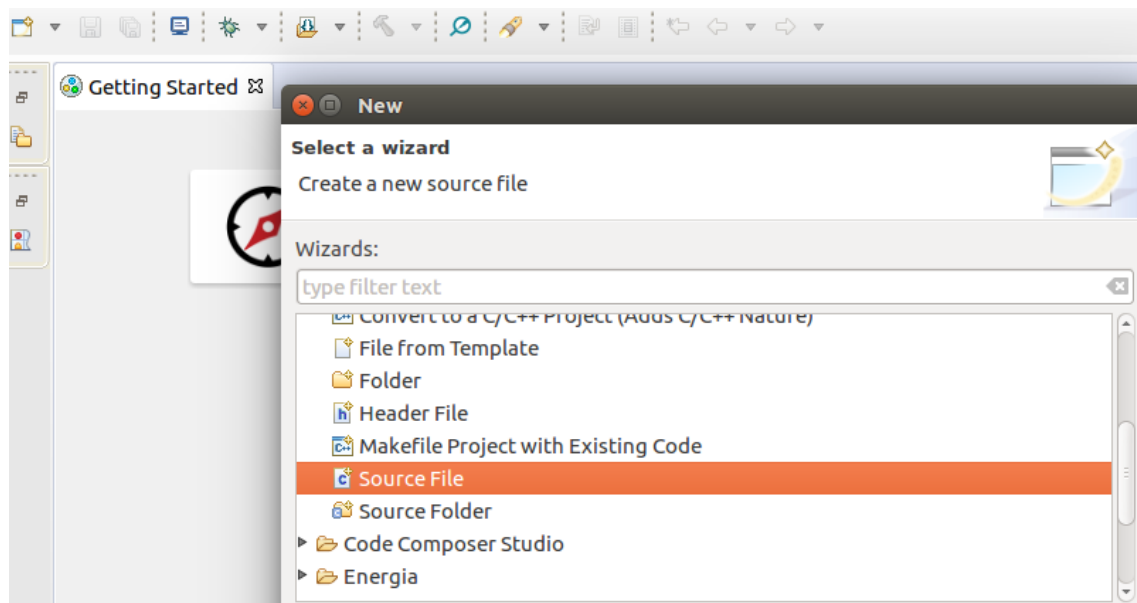
10. After completing the steps above you should now have a helloworld project in your CCS Project Explorer window, but the project has no sources.

//위의 단계를 완료한 후 CCS Project Explorer 창에 helloworld Project 가 있지만 Project 에는 소스가 없다.

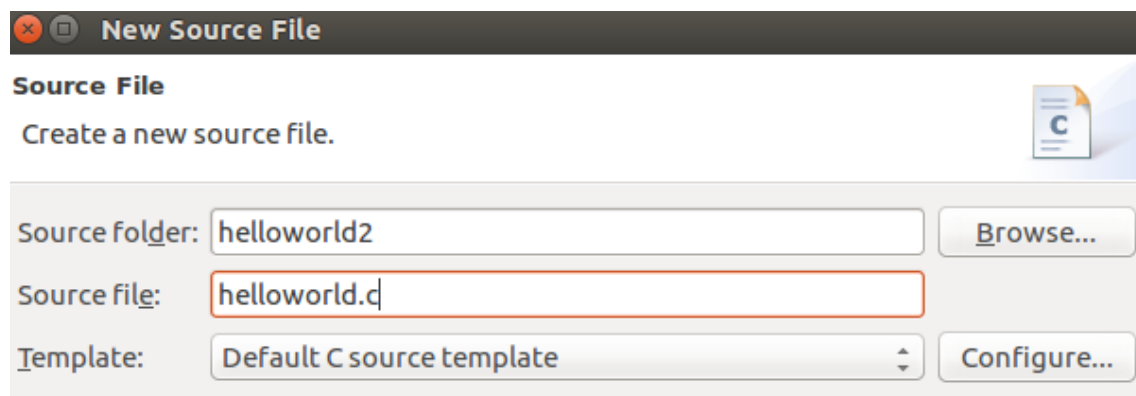
11. From the main CCSv8 window select File -> New> Source File menu item.

12. In the Source File dialog set the Source file: setting to helloworld.c

11. CCSv8 window 창에서 File -> New -> Source File 메뉴 항목을 선택한다.



12. Source File 대화 상자에서 Source file 을 설정: helloworld.c 로 설정한다.




13. Click Finish.

14. After completing the steps above you will have a template helloworld.c file. Add your code to this file like the image below:

//위의 단계를 완료하면 템플릿 helloworld.c 파일이 생긴다.

아래 Image 와 같이 이 파일에 코드를 추가한다.

Setting Started

 \*helloworld.c

```
/*  
 * helloworld.c  
 *  
 * Created on: Jun 8, 2018  
 * Author: yong  
 */  
  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello World from TI AM5728 Heterogeneous Architecture\n");  
    return 0;  
}
```

15. Compile the helloworld project by selecting Project -> Build Project  
//프로젝트 -> 프로젝트 빌드를 선택하여 helloworld 프로젝트를 컴파일하십시오.

16. The resulting executable can be found in the Debug directory.  
//결과로 나온 실행 파일은 Debug 디렉토리에서 찾을 수 있다.