



**Xilinx Zynq FPGA, TI DSP,
MCU 기반의
프로그래밍 전문가 과정**

날 짜 : 2018 . 5 . 9

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

학생 – 정한별
hanbulkr@gmail.com

리눅스에 MCU CCS 와 Driver 와 HALCoGen 설치하기 !

1. 의존성 패키지 설치 -- 오래걸린다. 참고.

```
> sudo apt-get update
> sudo apt-get install libc6:i386 libx11-6:i386 libasound2:i386 libatk1.0-0:i386 libcairo2:i386
libcups2:i386 libdbus-glib-1-2:i386 libgconf-2-4:i386 libgdk-pixbuf2.0-0:i386 libgtk-3-0:i386
libice6:i386 libncurses5:i386 libsm6:i386 liborbit2:i386 libudev1:i386 libusb-0.1-4:i386 libstdc++
+6:i386 libxt6:i386 libxtst6:i386 libgnomeui-0:i386 libusb-1.0-0-dev:i386 libcanberra-gtk-
module:i386 gtk2-engines-murrine:i386 unzip
```

2. CCS 설치파일(.bin) 다운로드 -- Chrome 환경에서 받는다

http://processors.wiki.ti.com/index.php/Download_CCS

3. 압축을 푼다

4. 압축해제한 폴더로 들어간 후 bin 파일을 실행한다 -- ccs 치고 tab 키를 누르면 자동으로 파일이름 다 뜬 --
여기 xxxxx 부분은 자신이 받은 파일에 맞게 셋팅될겁니다. tab 누르면 자동으로요

```
> ./ccs_setup_6.x.x.xxxxxx.bin
```

5. accept => next => Hercules Safety MCUs 선택 => select debug probes 부분에서는 3 항목을 다 체크
한다 => 까는중... 역시 오래걸린다.. 에뮬레이터를 돌리기 때문이라고 한다

6. 드라이버 설치하기

Driver 설치를 위한 스크립트는 {설치 경로}/ccsv6/install_scripts/ 디렉토리에 있다. 설치 과정에서 설치 경로를 바꾸지 않고 그대로 설치했다면 /home/{UserName}/ti/ccsv6/install_scripts/ 디렉토리에 있을 것이다. 그 경로로 이동해서 install_driver.sh 를 Root 권한으로 실행한다.

```
> cd /home/{UserName}/ti/ccsv7/install_scripts
```

```
> sudo ./install_drivers.sh
```

리스타트 ㅋㅋ 하기

7. wine 설치 -- windows 환경에서만 실행가능한 파일을, 리눅스에서 설치 및 사용하기 위한 툴

<http://ko.wikihow.com/%EC%9A%B0%EB%B6%84%ED%88%AC%EC%97%90-Wine%EC%9D%84-%EC%84%A4%EC%B9%98%ED%95%98%EB%8A%94-%EB%B2%95>

wine 실행시 버전설정 -- applications 탭의 맨 아래의 window 버전을 win7 로 변경해준다

drives – Autodetect 클릭해준다

```
> winecfg
```

playonlinux 설치 -- windows 환경에서만 실행가능한 파일을, 리눅스에서 설치 및 사용하기 위한 가상의 공간 설치하는것

```
> wget -q "http://deb.playonlinux.com/public.gpg" -O- | sudo apt-key add -
```

```
> sudo wget http://deb.playonlinux.com/playonlinux_precise.list -O
```

```
/etc/apt/sources.list.d/playonlinux.list
```

```
> sudo apt-get update
```

```
> sudo apt-get install playonlinux
```

다음의 사이트에서 Automatic Code Generator 인 HALCoGen 을 다운받기 -- 폴더를 잘 기억하자
<http://www.ti.com/tool/halcogen>

halcogen 은 **wine3.0** 버전을 다운로드 했으면 바로 설치가 가능해진다. (압축을 굳이 풀지 않고 열어서 안에 들어있는 설치파일을 우클릭 하여 **open with other application** 을 눌러 **wine** 으로 실행 하면 설치파일이 실행이 된다.

밑에는 **playon linux** 로 받는법인가 보다.

8. 맨왼쪽위의 [보라색 아이콘]에서 play on linux 를 검색하여 찾는다. => 프로그램 설치 클릭 => 맨 밑에 install a non-listed program 클릭 => 다음..다음..다음.. => install a program in a new virtual drive 클릭 하고, 다음 => 가상공간 네이밍 하고 => 체크 안하고 다음 => 32bits windows installation 클릭 => [찾아보기 클릭] 하여, HALCoGen.exe 받은폴더 잘 기억하고 계시죠? 그 exe 파일 클릭 하고 다음~ => 그럼 설치됩니다. 두번째 인스톨되는 것 옵션체크 할것 - play on linux 에 shortcut 생성할 것
설치하실때 shortcut 설치하지 마세요. 왜냐하면.. play on linux 에 만든 가상공간에 깔린 것을 실행해야 하니까요.. 마치 usb 꼽은것처럼.. play on linux 를 켜야 HALCoGen 실행되는거 거든요
play on linux 를 Lock to 런처 해두는 것이 사용에 더 편리할겁니다

9. 자 이제 쌤이 올려주신 <http://cafe.naver.com/hestit/2038> 글의 마지막 셋팅 해주시면 끝나요오..

wine3.0 에 대해 url

<https://steemit.com/kr/@chicpro/16-04-wine-3-0>

아두이노 스케치 설치 하기.

<http://findnew.tistory.com/7>

<http://webnautes.tistory.com/736>

이 두사이트를 참고 하면 빠르다.

```
sudo apt-add-repository ppa:ubuntu-desktop/ubuntu-make
```

```
sudo apt-get update
```

```
sudo apt-get install ubuntu-make
```

```
sudo umake ide arduino
```

```
sudo usermod -aG dialout {사용자 이름} -
```

현재 로그인된 사용자를 dialog 그룹에 추가해준 후, 설정을 적용하기 위해 로그 아웃했다가 다시 로그인합니다.

실행시에는 `sudo arduino` 이런 식으로 하면 된다.

Coretex -R5 프로그램 하기.

Coretex-R5 의 LED 를 다루어 보자.

```
#include "HL_sys_common.h"
#include "HL_system.h"

#include "HL_sys_core.h"
#include "HL_mibspi.h"
#include "HL_esm.h"
#include "HL_rti.h"
#include "HL_gio.h"
#include "HL_het.h"

int main(void)
{
    /* USER CODE BEGIN (3) */

    rtiInit();      // 리얼 타임 인터럽트의 초기값을 지정해준다.
    gioSetDirection(hetPORT1, 0xFFFFFFFF); // 타이머 포트 하나를 출력으로 설정.
    rtiEnableNotification(rtiREG1, rtiNOTIFICATION_COMPARE0);
    _enable_IRQ_interrupt_(); // 인터럽트를 사용가능케 한다.
    rtiStartCounter(rtiREG1, rtiCOUNTER_BLOCK0);
    // 선택된 RTI 카운터블럭 모듈을 사용 시작.

    while(1)

    return 0;
}

// 이함수는 설정된 rti 가 플레그가 셋이 되면 time 에 맞게 동작하는 녀석이다.
void rtiNotification(rtiBASE_t *rtiREG, uint32 notification)
{
    int i=0;
    uint32 x[9]={0x00020000, 0x80000000,0x00000001,
                0x00000020,0x02000000,0x00040000,0x20000000,
0x08000000,0xAA060021};

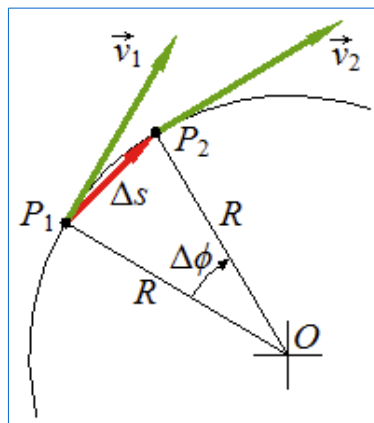
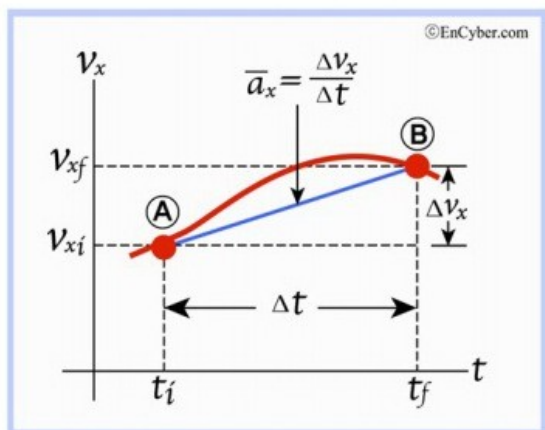
    // 인터럽트 timer 포트 1 에 gpio 포트의 값을 이용해 LED 를 켜다.
    gioSetPort(hetPORT1, gioGetPort(hetPORT1) ^ x[i]);
    i++;
    if(i==9) i=0;
}

Coretex-R5 의 LED 모양.
17          31          00
0x00020000  0x80000000  0x00000001

27          5
0x02000000  0x00000020

29          18          25
0x20000000  0x00040000  0x02000000
```

Uniform circular Motion(등속 원운동)



	각운동 (angular motion)	선운동 (linear motion)
속력(speed)	$\frac{\text{각거리}(\phi)}{\text{시간}(\Delta t)} = \text{각속력}(\delta)$	$\frac{\text{이동거리}}{\text{시간}(\Delta t)} = \text{속력}$
속도(velocity)	$\frac{\text{각변위}(\Delta\theta)}{\text{시간}(\Delta t)} = \text{각속도}(\omega)$	$\frac{\text{이동변위}(\Delta d)}{\text{시간}(\Delta t)} = \text{속도}(v)$
가속도(acceleration)	$\frac{\text{각속도}(\Delta\omega)}{\text{시간}(\Delta t)} = \text{각가속도}(\alpha)$	$\frac{\text{속도}(\Delta v)}{\text{시간}(\Delta t)} = \text{가속도}(a)$

Relation Between Angular Acceleration and Linear Acceleration

The acceleration of a particle is given by

$$a = \frac{dv}{dt} \dots\dots\dots (1)$$

But $v = r\omega$

$$\therefore a = \frac{d(r\omega)}{dt}$$

$$\therefore a = r \frac{d\omega}{dt} + \omega \frac{dr}{dt} \dots\dots (2)$$

$\therefore r$ is a constant radius,

$$\frac{dr}{dt} = 0$$

$$\therefore a = r \frac{d\omega}{dt}$$

$$\text{But } \frac{d\omega}{dt} = \alpha$$

α is the angular acceleration

$$\therefore \boxed{a = r\alpha} \dots\dots\dots (3)$$

Quantity	Linear	Angular
distance	$s = r\theta$	θ
velocity	$v = r\omega$	ω
acceleration	$a = r\alpha$	α
force / torque	F	$\tau = r \otimes F$
mass / moment of Inertia	m	$I = \sum mr^2$
Newton's Law	$F = ma$	$\tau = I\alpha$
Momentum	$p = mv$	$L = I\omega$