

TI DSP, MCU 및 Xilinx Zynq FPGA

프로그래밍 전문가 과정

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 - 하성용
accept0108@naver.com

51 일차

sudo arduino

```
#include<Servo.h> //대소문자 구분할것
```

```
#define DT 100.0 //DT 가 100, DT 는 0.1 초
```

```
//millisecond, ms 단위(1000 분의 1 초)기 때문에 100 으로 적어둔것
```

```
#define DTHETA 20.0 //디세타가 20, 각도의 변화량이 20 도씩 변한다는것
```

```
Servo myservo; //서버핑
```

```
int theta = 0; //값을 정하지 않아서 0 부터시작
```

```
double omega; //오메가=각속도
```

```
double alpha; //각가속도
```

```
double velocity; //속도
```

```
double acceleration; //가속도
```

```
double dt = DT / 1000.0; //dt 0.1
```

```
double time = 0.0; //초기값 0.0 세팅
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600); //속도지정
```

```
  myservo.attach(9); //9 번 pin 을 pwn 핀으로 쓰겠다
```

```
}
```

```
void loop()
```

```
{
```

```
  Serial.println((double)(DTHETA/ (500.0/1000.0)));
```

```
//DTHETA 를 2 분에 1 로나누고있음, 2 DTHETA 를 출력하는거, 2DTHETA= 40
```

```
//시리얼 모니터 보는방법
```



```
//우측 돋보기 클릭
```

```
for(theta = 0; theta < 180; theta += DTHETA)
```

```
//0 부터 180 도까지 보는데 theta 의 증가폭을 20 도로 바꿈
```

```
{
```

```
  myservo.write(theta); //myservo write theta 맨처음엔 0 들어감 0.1 초
```

```
  delay(DT);
```

```
  Serial.print("Theta = "); //현재 Theta 값 0
```

```
  Serial.println(theta);
```

```
  Serial.print("Dtheta="); //Dtheta= 20
```

```
  Serial.println(DTHETA);
```

```
  time += dt; // time 에 dt, 0.1 초가 지났음을 알수있음
```

```
  Serial.print("Total Time = "); //0.1 초
```

```
  Serial.println(time);
```

```
  Serial.print("dt=");
```

```
  Serial.println(dt);
```

```
  Serial.print("Omega="); // 오메가 계산, 오메가는 시간에 따른 각도의 변화량
```

```
// 샘플링타임을 0.1 초로 잡음, dt 0.1 초가 지난상태에서 dtheta 를 출력하게되면
```

```
//0.1 분에 20 이 나오게됨
```

```
  Serial.println((double)(DTHETA)/dt);
```

```

//여기에 상당한 오류가있음
//항상 각속도는 200 이 나올것, radiant second rpm 으로 바꾸면 60 을 곱하면 12000
//12000rpm 은 속도가 엄청빠른거지만 모터가 느리게 도는이유는?
//호도법 = 360 도가  $2\pi$ 
//1 도는 180 분에  $\pi$  가 되는거
//여기서 20 은 무엇으로바꿔줘야함 radian 으로
//여기서 도퍼세컨드로 잘못계산한것
Serial.println();
//delay(1500);
}
for(theta=180; theta >= 1; theta-=DTHETA)
{
  myservo.write(theta);
  delay(DT);
}
}

//alpha 는 오메가의 변화량
//9 분에 1 파이에 대한 변화량
//그러면 알파값은 있는가? X
//alpha 값은 정말 없는가? X
//오메가값을 변하게하면 각가속도를 구할수있는것

```

sketch_may10a 5

```
#include <Servo.h>

#define DT 100.0
#define DTHETA 20.0

Servo myservo;

int theta = 0;
double omega;
double alpha;
double velocity;
double acceleration;

double dt = DT / 1000.0;
double time = 0.0;

void setup()
{
  Serial.begin(9600);
  myservo.attach(9);
}

void loop()
{
  Serial.println((double){DTHETA/ (500.0/1000.0)});
  for(theta = 0; theta < 180; theta += DTHETA)
  {
    myservo.write(theta);
    delay(DT);
    Serial.print("Theta = ");
    Serial.println(theta);
    Serial.print("DTheta=");
    Serial.println(DTHETA);
    time += dt;
    Serial.print("Total Time = ");
    Serial.println(time);
    Serial.print("dt=");
    Serial.println(dt);
    Serial.print("Omega=");
    Serial.println((double){DTHETA}/dt);
    Serial.println();
    //delay(1500);
  }
  for(theta=180; theta >= 1; theta-=DTHETA)
  {
    myservo.write(theta);
    delay(DT);
  }
}
```

Done compiling.

Binary sketch size: 6,124 bytes (of a 32,256 byte maximum)

- Random Number Control with Arduino

```
int randNumber;

void setup()
{
  Serial.begin(9600);
  Serial.println("Print Random Numbers 0~9");

  for(int i = 0; i < 20; i++)
  {
    randNumber = random(10);
    Serial.print(randNumber);
    Serial.print(" "); //동일한 숫자 출력
  }

  Serial.println();
  Serial.println("Print Random Numbers 2~9");

  for(int i=0; i<20; i++)
  {
    randNumber = random(2,10);
    Serial.print(randNumber);
    Serial.print(" "); //동일한 숫자 출력
  }

  randomSeed(analogRead(0));
  Serial.println();
  Serial.println("Print Random Numbers 0~9");

  for(int i=0; i<20; i++)
  {
    randNumber=random(10);
    Serial.print(randNumber);
    Serial.print(" "); //srand 로 다른숫자출력
  }
  Serial.println();
  Serial.println();
}

void loop()
{
}
```

```

Print Random Numbers 0~9
7 9 3 8 0 2 4 8 3 9 0 5 2 2 7 3 7 9 0 2
Print Random Numbers 2~9
9 3 7 7 2 7 5 8 2 9 3 4 2 5 4 3 5 7 5 7
Print Random Numbers 0~9
1 9 5 1 0 6 9 4 9 5 4 7 5 8 7 3 2 6 4 7

Print Random Numbers 0~9
7 9 3 8 0 2 4 8 3 9 0 5 2 2 7 3 7 9 0 2
Print Random Numbers 2~9
9 3 7 7 2 7 5 8 2 9 3 4 2 5 4 3 5 7 5 7
Print Random Numbers 0~9
5 7 4 0 3 0 7 3 8 6 7 0 2 2 1 2 9 4 7 1

Print Random Numbers 0~9
7 9 3 8 0 2 4 8 3 9 0 5 2 2 7 3 7 9 0 2
Print Random Numbers 2~9
9 3 7 7 2 7 5 8 2 9 3 4 2 5 4 3 5 7 5 7
Print Random Numbers 0~9
2 3 6 6 8 6 1 3 3 1 5 0 3 4 5 3 8 3 5 3

Print Random Numbers 0~9
7 9 3 8 0 2 4 8 3 9 0 5 2 2 7 3 7 9 0 2
Print Random Numbers 2~9
9 3 7 7 2 7 5 8 2 9 3 4 2 5 4 3 5 7 5 7
Print Random Numbers 0~9
7 1 3 2 4 9 1 5 0 1 1 4 5 4 0 4 2 8 1 3

```

```
#include <Servo.h>
```

```
#define DT 50.0
```

```
Servo myservo;
```

```
double pi = 3.1415926535897932384643383279502884197163993751058209;
//무리수 → 길게해야 정밀도가 높음 →  $\pi$ (파이) 할때 이거 복사하면 좋음
```

```
int theta = 0;
double d_theta = 0.0; //각도의 변화량
```

```
double radian = 0.0; //180 분에  $\pi$ (파이)를 곱해주는거
double omega = 0.0;
double d_omega = 0.0; //현재 측정한 오메가-이전오메가
```

```
double alpha = 0.0;
double velocity;
double acceleration; //r 오메가 제공
```

```
double dt = DT/1000.0; //DT 50 분에 1000 이니까 20 분에 1 , 100 분에 5 , 0.05 초,
//샘플타임을 0.05 초로 만듦
double time = 0.0;
```

```
void setup()
{
  Serial.begin(9600);
  randomSeed(analogRead(0));
  myservo.attach(9);
}
```

```

void loop(){
while(theta < 180)
{
myservo.write(theta);
delay(DT);

Serial.print("Theta = "); //현재세타값 0
Serial.println(theta); //초기값은 0

Serial.print("DTheta = ");
Serial.println(d_theta);

Serial.print("Radian =");
radian = (d_theta/360)*2*pi;
Serial.println(radian);

time += dt; //0.05 초

Serial.print("Total Time = "); //0.05
Serial.println(time);

Serial.print("dt = ");
Serial.println(dt);

Serial.print("Omega = "); // 초기값은 0
d_omega = (radian/dt)-omega;
omega=radian/dt;
Serial.println(omega);

Serial.print("DOmega = "); // 초기값은 0
Serial.println(d_omega);

Serial.print("Velocity = "); // 초기값은 0
velocity = 0.01815 * omega;
Serial.println(velocity);

Serial.print("Acceleration = ");
acceleration = 0.01815 * omega * omega;
Serial.println(acceleration, 10); //10 은 10 진수로 출력하라
// 1 부터 10 까지 각도가 변한단소리
//가장 많이 변하는건 9

Serial.print("Alpha = ");
alpha = d_omega/dt;
Serial.println(alpha);

Serial.println(); //개행문자

d_theta = random(1, 11);
theta += d_theta;
}
for(theta = 180; theta >= 1; theta -= random(1, 11))
{
myservo.write(theta);
delay(DT);
}
}

```

//디오메가값은 음수가 나올수있음

- PWM

CCS, HALCogen 구동

hetPWM

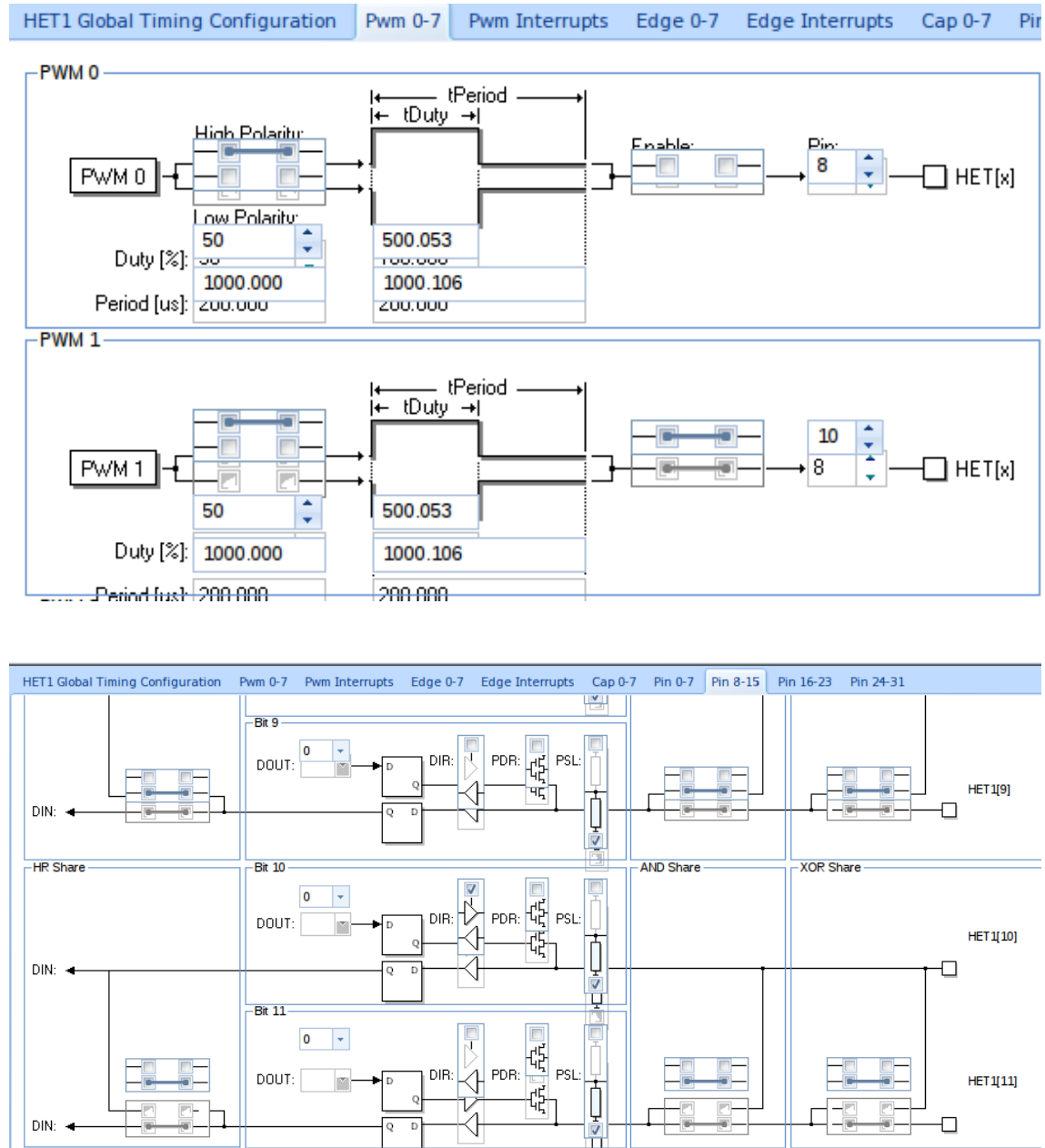
location : /home/yong/workspace_v8/hetPWM

HALCogen - driver enable - Het1

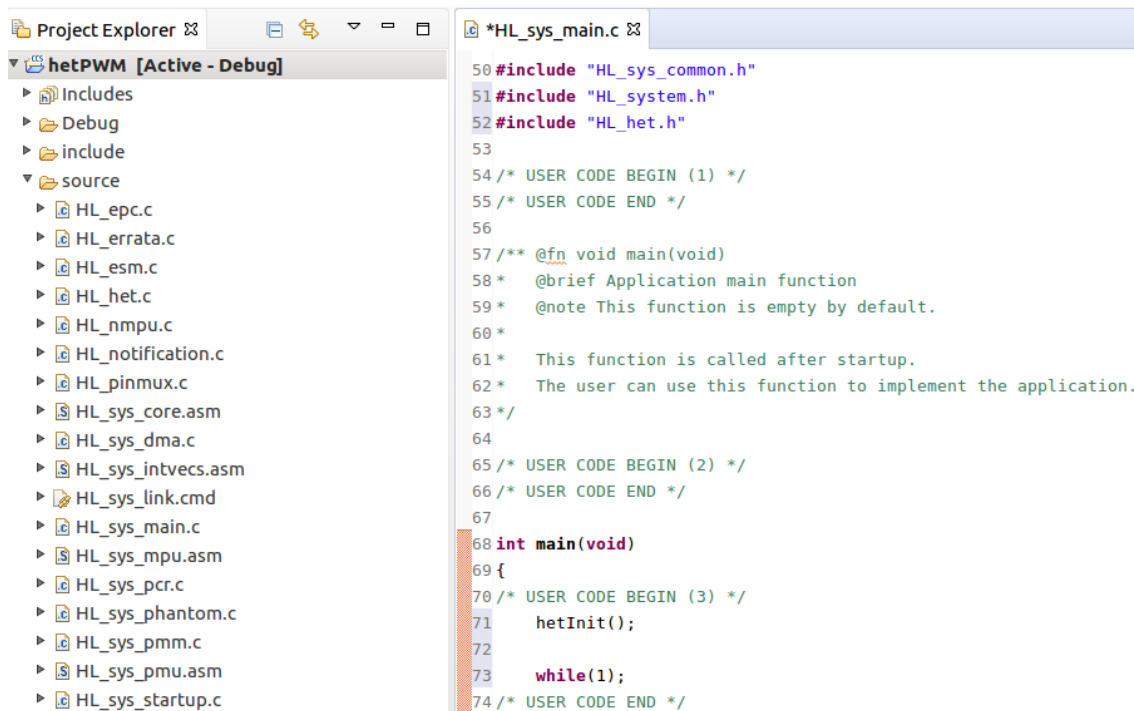
het1 에서

VCLK2 가 75 인걸 확인

PWM 0-7



CCS - HL_het.h



HETDIR

0 이면 검색 1 이면 출력
출력인게 10 번비트밖에없음

HET110

10 번비트가 출력비트로 사용됨

HETPDR

0 번이면 pin 이 pushple 모드
1 번이면 open 모드

HETPULDIS

0 번이면 HET 에 풀업을 inable 하겠다

PSL

pull 할거니까 up 을할건지 dawn 을 할건지 선택
0 이니까 pulldown 을 하겠다

HETHRSH

파일 리솔루션 관련된부분
1 번이면 서로 쉐어하고 0 번이면 쉐어안함
쉐어란? 서로 신호선이 공유가된다

HETAND

0 이니까 앤드게이트를통해 쉐어되진않는다
XOR 도 마찬가지

HETPFR

0110 → 8
9 번하고 10 번비트가 설정된거
6th 보면 64 분주한다는걸 볼수있음

HETPCR

에러에대한 검색안함

hetRAM 안을 보면

각종 instruction 들이 있음

het1PROGRAM

리눅스커널에서 구조체초기화할때 본것들

46000 부터 46200 까지는 프로그램과제어가 들어있음

두번째는 오류가 났을때 어떻게처리할것인지

96 비트들을 가지고있음

12 바이트라는거

16 바이트로 언라인된다는거

정렬을 4 곱하기 2 에 n 승으로 해주는게 좋음

인스트럭션은 4 바이트씩 가짐

제어(컨트롤), 프로그램, 데이터, reserved?

네번째 구현되지않은 주소에 의해서

128 비트로 강제로 정렬이 됨

N2HET RAM

Program

Control

Data

명령어가 16 바이트로 동작

굉장히 특수한 cpu

하이엔드타이머는 별도의 조작을 할수가있음

instructionset 을 본다

n2het instructionhet 이라는게 나오고있음

숫자값을 보고 디코딩하라는거

즉 기계어를 이해하고 기계어를 디코딩

프로그램이 cnt

N2HET 전용 어셈블리어

이게 해석이 안되면 하이엔드타이머를 조작할 때가왔을때 조작을 하지못함

FPGA 로 타이머를 만든과정

13,11,10,7 비트들로 세팅

P : 0x2C80

C : 0x01FFFFFF

D : 0xFFFFFFFF80

오코드에 해당하는거 [mov] r1, r2

16/32 검색

563a//

프로그램필드

CNT Program

13 번의미, 11 번의미(0110 → opcode)

8 번비트 값없음(off)

결과적으로 하이엔드타이머를 사용하기위해서는 데이터필드를 다봐야한다는걸 볼수있음