TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

2018-08-08 (중간발표 5회차)

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 - 정유경
ucong@naver.com

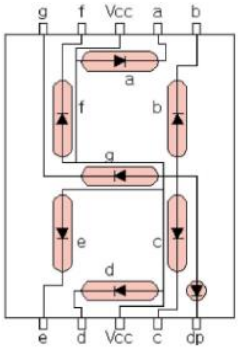
[1] 진행상황 및 목표: 7-Segment를 이용하여 현재 차량의 속도 Display

Sensorless BLDC모터를 구매하였으므로, 차량의 속도를 구하기 위하여 ESC를 이용한다.
ESC에 내장된 엔코더와 MCU의 eQEP모듈을 사용하여 차량의 속도를 측정한다.
측정된 현재 속도 값을 7-Segment를 통해 표현한다.

[2] 회로 구성

사용한 부품

: YDSR-1056AM, 74HC138, 160옴 저항7개, 점퍼선, 브레드보드, USB to TTL Serial Cable



step1

- 7-Segmen를 제어하기 위한 MCU핀을 최소화하기 위해 디코더를 사용하였다(74HC138 3-to-8 line decoder/demultiplexer; inverting)
- 디코더에서 출력되는 부논리(Active Low)신호를 이용하기 위해 Common Anode 방식의 7-Segment를 사용한다

step2

- YDSR-1056AM은 Common 단자에 VCC를 연결하고 각각 8개의 Cathod핀에 Low 신호를 입력하면 동작한다.
- Typical한 동작전압과 전류는 1.8V 20mA이므로 전류제한용 저항으로 (5 - 1.8) /20mA = 160옴보다 큰 저항을 각 핀에 연결한다.

step3

동작전압이 최대 7V임을 확인하고 디코더에 VCC, GND를 인가한다.

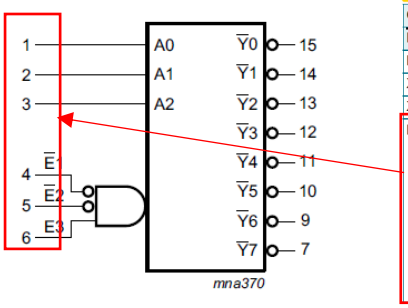


Table 3. Function table[1]

Control			Input			Output							
$\overline{E1}$	$\overline{E2}$	E3	A2	A1	A0	$\overline{Y7}$	$\overline{Y6}$	$\overline{Y5}$	$\overline{Y4}$	$\overline{Y3}$	$\overline{Y2}$	$\overline{Y1}$	$\overline{Y0}$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X											
X	X	L											
L	L	H	L	L	L	H	H	H	H	H	H	H	L
			L	L	H	H	H	H	H	H	H	L	H
			L	H	L	H	H	H	H	H	L	H	H
			L	H	H	H	H	H	H	L	H	H	H
			H	L	L	H	H	H	L	H	H	H	H
			H	L	H	H	H	L	H	H	H	H	H
			H	H	L	H	L	H	H	H	H	H	H
			H	H	H	L	H	H	H	H	H	H	H

- 디코더의 진리표를 보면 Enable핀에 LLH 신호를 주었을 때 동작함을 알 수 있다.

Step4

- A0, A1, A2핀은 각각 GIOA[2:0]핀에 연결하고, /Y[0:7]은 각각 7 Segment에 연결한다.
- 전체 동작을 분석해보면 다음과 같다.

GIOA[7:0]	A[0:2]	/Y[0:7]	Segment
0x00	000	/Y[0]	A
0x04	100	/Y[1]	B
0x02	010	/Y[2]	C
0x06	110	/Y[3]	D
0x01	001	/Y[4]	E
0x05	101	/Y[5]	F
0x03	011	/Y[6]	G
0x07	111	/Y[7]	DP

- 하나의 숫자를 나타내기 위해서는 7-Segment의 각 핀이 병렬로 연결되어 있으므로, Static이 아닌 Dynamic 방식으로 구동시켜야 한다.
즉, 7-Segment를 순차적으로 하나씩 빠르게 점등시켜서 잔상효과를 이용한다.

회로구성이 끝났다면 다음과 같이 코드를 작성한다.

구현목표
eQEP에서 받아오는 입력값을 RTI 인터럽트마다 갱신하여 표현하기 위하여,
우선 count를 증가시키면서 RTI 인터럽트가 들어올때마다 count값을 Segment에 표시하는 기능을 구현한다.

HalCoGen설정은 다음과 같다.
GIO, RTI Driver Enable하고 GIOA Pinmux를 설정한다.
RTI Compare0 Period를 1000으로 변경하여 RTI 인터럽트 주기를 Actual Period 1000ms로 한다
VIM Channel에서 RTI Compare0의 IRQ인터럽트를 허용한다.
GIOA Port의 Direction을 출력으로 설정한다.
Generate Code.

CCS에서 다음과 같이 코드를 작성한다.

```
#include "HL_gio.h"
#include "HL_rti.h"
#include "HL_sys_core.h"
#include <stdio.h>

uint32 count=0;
```

```
void delay(uint32 time);
void rtiNotification(rtiBASE_t *rtiREG,uint32 notification)
{
    count = (count+1) % 10;
}

int main(void)
{
    gioInit();
    rtiInit();
    gioSetPort(gioPORTA,0);
    _enable_IRQ_interrupt_();       //_enable_interrupt_();는  FIQ, IRQ 둘다 포함함
    rtiStartCounter(rtiREG1, rtiCOUNTER_BLOCK0);
    rtiEnableNotification(rtiREG1, rtiNOTIFICATION_COMPARE0);

    while(1){
        switch(count){
        case 0:
            while(1)
            {
                gioPORTA->DOUT = 0x00; delay(2500); // a
                gioPORTA->DOUT = 0x04; delay(2500); // b
                gioPORTA->DOUT = 0x02; delay(2500); // c
                gioPORTA->DOUT = 0x06; delay(2500); // d
                gioPORTA->DOUT = 0x01; delay(2500); // e
                gioPORTA->DOUT = 0x05; delay(2500); // f
               // printf("count: %d₩n", count);
                if(count!=0)
                    break;
            }
            break;

        case 1:
            while(1)
            {
                gioPORTA->DOUT = 0x04; delay(2500); // b
```

```
            gioPORTA->DOUT = 0x02; delay(2500); // c
            if(count!=1)
                break;
        }
        break;

case 2:
        while(1)
        {
            gioPORTA->DOUT = 0x00; delay(2500); //a
            gioPORTA->DOUT = 0x04; delay(2500); //b
            gioPORTA->DOUT = 0x03; delay(2500); //g
            gioPORTA->DOUT = 0x01; delay(2500); //e
            gioPORTA->DOUT = 0x06; delay(2500); //d
            if(count!=2)
                break;
        }
        break;

case 3:
        while(1)
        {
            gioPORTA->DOUT = 0x00; delay(2500); //a
            gioPORTA->DOUT = 0x04; delay(2500); //b
            gioPORTA->DOUT = 0x03; delay(2500); //g
            gioPORTA->DOUT = 0x02; delay(2500); //e
            gioPORTA->DOUT = 0x06; delay(2500); //d
            if(count!=3)
                break;
        }
        break;

case 4:
        while(1)
        {
            gioPORTA->DOUT = 0x05; delay(2500); //f
```

```
            gioPORTA->DOUT = 0x03; delay(2500); //g
            gioPORTA->DOUT = 0x04; delay(2500); //b
            gioPORTA->DOUT = 0x02; delay(2500); //c
            if(count!=4)
                break;
        }
        break;

case 5:
        while(1)
        {
            gioPORTA->DOUT = 0x00; delay(2500); //a
            gioPORTA->DOUT = 0x05; delay(2500); //f
            gioPORTA->DOUT = 0x03; delay(2500); //g
            gioPORTA->DOUT = 0x02; delay(2500); //c
            gioPORTA->DOUT = 0x06; delay(2500); //d
            if(count!=5)
                break;
        }
        break;

case 6:
        while(1)
        {
            gioPORTA->DOUT = 0x00; delay(2500); //a
            gioPORTA->DOUT = 0x05; delay(2500); //f
            gioPORTA->DOUT = 0x03; delay(2500); //g
            gioPORTA->DOUT = 0x01; delay(2500); //e
            gioPORTA->DOUT = 0x02; delay(2500); //c
            gioPORTA->DOUT = 0x06; delay(2500); //d
            if(count!=6)
                break;
        }
        break;

case 7:
```

```
        while(1)
        {
            gioPORTA->DOUT = 0x05; delay(2500); //f
            gioPORTA->DOUT = 0x00; delay(2500); //a
            gioPORTA->DOUT = 0x04; delay(2500); //b
            gioPORTA->DOUT = 0x02; delay(2500); //c
            if(count!=7)
                break;
        }
        break;

    case 8:
        while(1)
        {
            gioPORTA->DOUT = 0x00; delay(2500); //a
            gioPORTA->DOUT = 0x04; delay(2500); //b
            gioPORTA->DOUT = 0x02; delay(2500); //c
            gioPORTA->DOUT = 0x06; delay(2500); //d
            gioPORTA->DOUT = 0x01; delay(2500); //e
            gioPORTA->DOUT = 0x05; delay(2500); //f
            gioPORTA->DOUT = 0x03; delay(2500); //g
            if(count!=8)
                break;
        }
        break;

    case 9:
        while(1)
        {
            gioPORTA->DOUT = 0x00; delay(2500); //a
            gioPORTA->DOUT = 0x05; delay(2500); //f
            gioPORTA->DOUT = 0x03; delay(2500); //g
            gioPORTA->DOUT = 0x04; delay(2500); //b
            gioPORTA->DOUT = 0x02; delay(2500); //c
            gioPORTA->DOUT = 0x06; delay(2500); //d
            if(count!=9)
```

```
                        break;
                }
                break;
            }
        }
}
void delay(uint32 time)
{
    int i;
    for(i=0;i<time;i++)
        ;
}
```

아래와 같은 결과를 확인할 수 있다.