◇ Master & Slave 개발 과정

DATE : 2018.08.30

학과	전자전기공학부
학번	32153675
이름	이진호

1. 개발 과정

처음 PID제어기를 설정하고 마스터앤 슬레이브를 만들어봤는데 생각보다 어려웠던것 같다.

[그림 1] 마스터 앤 슬레이브 테스트

일단 총 6개의 센서 중에 가운데 2개로 앞뒤 제어를 하고 센서값은 127등분 기준 50을 유지하도록 하였다. 그러고 나서 6개의 센서에 각각의 무게값을 대입한 뒤, 트레이서 할 때와 똑같이 포지션 값을 받고 또 다른 쪽에서 값이 들어와도 기존에 보던 것을 유지하도록 포지션 이네이블을 적용시켰다.

기존에 16개로 쓰던 것을 6개로 새로 지정해서 포지션 값을 받고 이네이블을 짜는데 살짝 혼동이 왔었지만 혜원이의 도움으로 기존의 것보다 연산 량을 줄여서 짤 수 있어서 만족스러웠다.

[그림 2] 마스터 앤 슬레이브

가운데 센서로 전후방 제어를 해서 센서값 50을 유지하도록 했고 50의 값에 핸들값을 적용시켜서 무게값을 적용시켰다. 확인해본 바로는 핸들이 가장 오른쪽이나 왼쪽에 있을 때 0.5 1.6(1.6 0.5)정도로 약 3배정도 차이가 났었던것 같다. 하지만 포지션이 그렇게 안정적이지는 않아서 다음에 할 때는 센서를 여러개 써서 더 안정적인 포지션을 받고싶다. 형들은 센서를 여러개 쓸 필요 없다고는 하지만 보다 정밀한 제어를 위해서는 여러 개 써야 될 것 같다. 포지션이 너무 급격하게 올라가거나 내려가는 바람에 모터가 들그득 거리는 경우가 좀 있었다.

그리고 PID제어를 하는데 내가 하는 게 PID제어가 애매하게 적용되어 힘들었다. 분명히 P제어는 먹히고 있는 것같은데 D제어가 적용되는지 안 되는지 구별이 안되어 아쉬웠다. 더한 값을 계속해서 누적해서 더하는 게 트레이서 소스에 있었는데 그렇게 하니까 값이 너무 빨리 뛰어서 그냥 대입만 시켰는데, 아무리 생각해도 아닌 것 같아서 내년에 할 때는 일단 그걸 먼저 해결하려고 한다.

또한 이번에 마슬을 제어할 때는 PWM에다가 직접 값을 넣어서 제어를 했는데 이런 식으로 하면 트레이서의 2차주행같은 급격한 변화에 잘 대응하지 못했다. 실제로 1700 2차를 굴렸을 때 슬레이브와 트레이서가 부딪혀서 파손될 뻔했다.
멘토의 말에 의하면 PWM이 아닌 속도로 제어를 하면 트레이서의 2차주행시에도 거리유지를 할 수 있다고 하니 참고해서 내년엔 하도록 해야겠다. 올해에는 하려다가 안되었지만 소스는 남아있으니까 내년에 다시 한 번 도전해서 2500 2차로 대회를 나가보려고 한다.

또 하나 놓쳤던 것은 PD 상수 값을 맞출 때 ROM에다가 저장을 시켜서 즉각적으로 바꿔서 테스트했으면 더 빠르고 다양하게 했을 텐데 멍청하게 바꾸고 넣고 바꾸고 넣고 하면서 해서 괜한 시간을 낭비했던 것 같다. 다음부터는 그런 것도 롬에다 저장해서 효율적으로 코딩해야겠다.

2. 소스코드

소스 전문을 첨부하는 것은 어렵기 때문에 모터 제어와 센서 부분만 발췌했다.

```
g_cont.iq17cur_RSTresult = _IQdiv((g_sen[2].iq17data + g_sen[3].iq17data),_IQ(2.0));
g_cont.iq17RSTvalue[3]=g_cont.iq17RSTvalue[2];
g_cont.iq17RSTvalue[2]=g_cont.iq17RSTvalue[1];
g_cont.iq17RSTvalue[1]=g_cont.iq17RSTvalue[0];
g_cont.iq17RSTvalue[0]=g_cont.iq17cur_RSTresult;

g_cont.iq17err_RSTresult=_IQmpy(_IQ(50),g_q17right_handle) -
g_cont.iq17cur_RSTresult;
g_cont.iq17RSTPresult = _IQmpy(q17STKP,g_cont.iq17err_RSTresult);
g_cont.iq17RSTDresult = _IQmpy(q17STKD,(g_cont.iq17RSTvalue[0]-
g_cont.iq17RSTvalue[3])+_IQmpy((g_cont.iq17RSTvalue[1]-g_cont.iq17RSTvalue[2]),_
IQ(3.0)));
g_cont.iq17RSTresult = g_cont.iq17RSTDresult + g_cont.iq17RSTPresult; // Right




g_cont.iq17cur_LSTresult = _IQdiv((g_sen[2].iq17data + g_sen[3].iq17data),_IQ(2.0));
g_cont.iq17LSTvalue[3]=g_cont.iq17LSTvalue[2];
g_cont.iq17LSTvalue[2]=g_cont.iq17LSTvalue[1];
g_cont.iq17LSTvalue[1]=g_cont.iq17LSTvalue[0];
g_cont.iq17LSTvalue[0]=g_cont.iq17cur_LSTresult;

g_cont.iq17err_LSTresult=_IQmpy(_IQ(50),g_q17left_handle) -
g_cont.iq17cur_LSTresult;
g_cont.iq17LSTPresult = _IQmpy(q17STKP,g_cont.iq17err_LSTresult);
g_cont.iq17LSTDresult = _IQmpy(q17STKD,(g_cont.iq17LSTvalue[0]-
g_cont.iq17LSTvalue[3])+_IQmpy((g_cont.iq17LSTvalue[1]-g_cont.iq17LSTvalue[2]),_I
Q(3.0)));

g_cont.iq17LSTresult = g_cont.iq17LSTDresult + g_cont.iq17LSTPresult; // Left
} // Straight_PID 함수



void sen_vari_init(void)
{
```

```
memset( ( void * )g_sen , 0x00 , sizeof( sen_t) * 16 );
memset( ( void * )&g_pos, 0x00 , sizeof( position_t ) );


g_u16pos_cnt = S_THREE;
g_u16sen_enable = 0xffff;

g_sen[ 5 ].iq7weight = _IQ7(6000);
g_sen[ 4 ].iq7weight = _IQ7(3000);
g_sen[ 3 ].iq7weight = _IQ7(500);
g_sen[ 2 ].iq7weight = _IQ7(-500);
g_sen[ 1 ].iq7weight = _IQ7(-3000);
g_sen[ 0 ].iq7weight = _IQ7(-6000);
}  // 센서 초기설정 및 무게값 대입

void make_position(void)
{
g_pos.iq17sum = _IQ(0);

switch(g_u16pos_cnt)
{
case S_ONE:
g_pos.iq17sum += g_sen[S_TWO].iq17data;

case NONE:
g_pos.iq17sum += g_sen[S_ONE].iq17data;
g_pos.iq17sum += g_sen[NONE].iq17data;

break;

case S_TWO:
g_pos.iq17sum += g_sen[S_ONE].iq17data;
g_pos.iq17sum += g_sen[S_TWO].iq17data;
g_pos.iq17sum += g_sen[S_THREE].iq17data;

break;

case S_THREE:
```

```
//TxPrintf("Rolling");
g_pos.iq17sum += g_sen[S_ONE].iq17data;
g_pos.iq17sum += g_sen[S_TWO].iq17data;
g_pos.iq17sum += g_sen[S_THREE].iq17data;
g_pos.iq17sum += g_sen[S_FOUR].iq17data;

break;

case S_FOUR:
g_pos.iq17sum += g_sen[S_TWO].iq17data;
g_pos.iq17sum += g_sen[S_THREE].iq17data;
g_pos.iq17sum += g_sen[S_FOUR].iq17data;

break;

case S_FIVE:
g_pos.iq17sum += g_sen[S_THREE].iq17data;

case S_SIX:
g_pos.iq17sum += g_sen[S_FOUR].iq17data;
g_pos.iq17sum += g_sen[S_FIVE].iq17data;

break;

}
/*
switch(g_u16pos_cnt)
{
case NONE:
case S_SIX:
g_pos.iq17avr = _IQdiv(g_pos.iq17sum , _IQ(2.0) );

case S_ONE:
case S_TWO:
case S_FOUR:
case S_FIVE:
g_pos.iq17avr = _IQdiv(g_pos.iq17sum , _IQ(3.0) );

case S_THREE:
g_pos.iq17avr = _IQdiv(g_pos.iq17sum , _IQ(4.0) );
```

```
}
*/
if( g_pos.iq17sum )
{
//TxPrintf("%d\t",g_pos.iq17sum);
g_pos.iq7sum_of_sec = _IQ7(0);
switch(g_u16pos_cnt)
{
case S_ONE:
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_TWO ].iq7weight, 7, g_sen[ S_TWO ].iq17data, 17 );
case NONE:
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_ONE ].iq7weight, 7, g_sen[ S_ONE ].iq17data, 17 );
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ NONE ].iq7weight, 7, g_sen[ NONE ].iq17data, 17 );

break;

case S_TWO:
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_ONE ].iq7weight, 7, g_sen[ S_ONE ].iq17data, 17 );
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_TWO ].iq7weight, 7, g_sen[ S_TWO ].iq17data, 17 );
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_THREE ].iq7weight, 7, g_sen[ S_THREE ].iq17data, 17 );

break;

case S_THREE:
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_ONE ].iq7weight, 7, g_sen[ S_ONE ].iq17data, 17 );
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_TWO ].iq7weight, 7, g_sen[ S_TWO ].iq17data, 17 );
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_THREE ].iq7weight, 7, g_sen[ S_THREE ].iq17data, 17 );
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_FOUR ].iq7weight, 7, g_sen[ S_FOUR ].iq17data, 17 );

break;
```

```
case S_FOUR:
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_TWO ].iq7weight, 7, g_sen[ S_TWO ].iq17data, 17 );
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_THREE ].iq7weight, 7, g_sen[ S_THREE ].iq17data, 17 );
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_FOUR ].iq7weight, 7, g_sen[ S_FOUR ].iq17data, 17 );

break;

case S_FIVE:
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_THREE ].iq7weight, 7, g_sen[ S_THREE ].iq17data, 17 );
case S_SIX:
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_FOUR ].iq7weight, 7, g_sen[ S_FOUR ].iq17data, 17 );
g_pos.iq7sum_of_sec += _IQ7mpyIQX( g_sen[ S_FIVE ].iq7weight, 7, g_sen[ S_FIVE ].iq17data, 17 );

break;

}

g_pos.iq7sum = g_pos.iq17sum >> 10;

g_pos.iq7temp_pos = _IQ7div( g_pos.iq7sum_of_sec, g_pos.iq7sum );

if( g_pos.iq7temp_pos > POS_END ) g_pos.iq7temp_pos = POS_END;
else if( g_pos.iq7temp_pos < -POS_END ) g_pos.iq7temp_pos = -POS_END;
else;

//if( g_line_info.u16turn_Info[g_line_info.u16turnmark_total_cnt] == STRAIGHT )
g_pos.iq7temp_pos = _IQmpy(g_pos.iq7temp_pos,_IQ(1.5));


position_enable();
}
```

```
else
{
g_int32lineout_pre_cnt++;


if( g_int32lineout_pre_cnt > 400 )
{
g_int32lineout_pre_cnt = 0;


g_Flag.lineout_flag = ON;
}
}
}

static void position_enable (void)
{
if( g_pos.iq7temp_pos >= g_sen[ 5 ].iq7weight )
{
g_u16pos_cnt = S_SIX;
g_u16sen_state = THREE_SHIFT;
g_u16sen_enable = 0x0060; //0000 0000 0110 0000
}
else if( g_pos.iq7temp_pos <= g_sen[ 0 ].iq7weight )
{
g_u16pos_cnt = NONE;
g_u16sen_state = THREE_SHIFT;
g_u16sen_enable = 0x0600; //0000 0110 0000 0000
}



else if( g_pos.iq7temp_pos > g_sen[ 4 ].iq7weight )
{
g_u16pos_cnt = S_FIVE;
g_u16sen_state = TWO_SHIFT;
g_u16sen_enable = 0x0070; //0000 0000 1110 0000
}
else if( g_pos.iq7temp_pos < g_sen[ 1 ].iq7weight )
{
```

```
g_u16pos_cnt = S_ONE;
g_u16sen_state = TWO_SHIFT;
g_u16sen_enable = 0x070; //0000 0111 0000 0000
}



else if( g_pos.iq7temp_pos > 0 )
{
g_u16pos_cnt = S_FOUR;
g_u16sen_state = ONE_SHIFT;
g_u16sen_enable = 0x01C0; //0000 0001 1100 0000
}
else if( g_pos.iq7temp_pos < 0 )
{
g_u16pos_cnt = S_TWO;
g_u16sen_state = ONE_SHIFT;
g_u16sen_enable = 0x0380; //0000 0011 1000 0000
}


else //if( g_pos.iq7temp_pos >= g_sen[ 2 ].iq7weight &&  g_pos.iq7temp_pos <=
g_sen[ 3 ].iq7weight )
{
g_u16pos_cnt =  S_THREE;
g_u16sen_state = NON_SHIFT;
g_u16sen_enable = 0x03c0; //0000 0011 1100 0000
}
// TxPrintf("%d\n",g_u16pos_cnt);

}

void Handle(void)
{
volatile _iq16 _iq16left_handle = _IQ16(0.0);
volatile _iq16 _iq16right_handle = _IQ16(0.0);

if( g_pos.iq7temp_pos < _IQ7(0.0) ) //left
{
_iq16left_handle = _IQ16mpy(g_q16han_decstep, ( HANDLE_CENTER + (
```

```
g_pos.iq7temp_pos << 9 )) ) + g_q16han_decmax;
_iq16right_handle = _IQ16mpy( g_q16han_accstep , ( HANDLE_CENTER - (
g_pos.iq7temp_pos << 9 )) ) + g_q16han_accmax;

if( _iq16left_handle < _IQ16(0.0) ) _iq16left_handle = _IQ16(0.0);

}
else
{
_iq16left_handle = _IQ16mpy( g_q16han_accstep , HANDLE_CENTER + (
g_pos.iq7temp_pos << 9 ) ) + g_q16han_accmax;
_iq16right_handle = _IQ16mpy( g_q16han_decstep , HANDLE_CENTER - (
g_pos.iq7temp_pos << 9 ) ) + g_q16han_decmax;

if( _iq16right_handle < _IQ16(0.0) ) _iq16right_handle = _IQ16(0.0);
}



g_q17left_handle = _iq16left_handle << 1;
g_q17right_handle = _iq16right_handle << 1;

}
```