



ESCUELA UNIVERSITARIA
POLITÉCNICA
DE SEVILLA



PROYECTO FIN DE CARRERA

**CONTROL DE ILUMINACIÓN CON FT 3120 SMART
TRANSCEIVER**

DOCUMENTO 2:

ANEXOS

ALUMNO AUTOR: JOAQUÍN RAMOS DOMÍNGUEZ
TITULACIÓN: I.T.I. ELECTRÓNICA INDUSTRIAL
TUTOR: D. FRANCISCO SIMÓN MUÑIZ
CONVOCATORIA: SEPTIEMBRE 2009



ÍNDICE DE ANEXOS

1. DOCUMENTACIÓN DE PARTIDA	3
2. CARACTERIZACIÓN DE LA FUENTE DE ALIMENTACIÓN	4
3. PROGRAMA DEL PROYECTO	13



1. DOCUMENTACIÓN DE PARTIDA.

1.1 Hojas de datos de los distintos componentes utilizados.

Se presentan adjuntas las hojas de datos de todos los componentes seleccionados para el diseño de las dos PCBs:

- Transformador encapsulado 12VA 1 x 9V / 1 x 1.330^a. Relación de transformación 230/9.
- Puente rectificador de diodos ADB18PS.
- Regulador de tensión LM317.
- Diodo 1N4148.
- Transistor BC557.
- Transistor BC547.
- Potenciómetro Bourns Multivuelta 20k. Montaje sobre circuito impreso.
- Pulsador SW073.
- Relé Finder 36.11-0300.
- Cristal de cuarzo X-TAL 20,00000 \pm 3ppm.
- 74HC14
- FT 3120 Smart Transceiver y transformador FT-X2.
- Caja de pared cerrada 235x185x119mm.



2. CARACTERIZACIÓN DE LA FUENTE DE ALIMENTACIÓN.

La caracterizamos y para ello tomamos el siguiente conjunto de medidas:

2.1 Regulamos la tensión a 5V.

-Carga en vacío.

V entrada positiva (V)	V salida (V)
1	0.293
2	1.199
3	2.107
4	3.02
5	3.79
6	4.48
7	5.00
8	5.00
9	5.00
10	5.00
11	5.00
12	5.00
13	5.00
14	5.00
15	5.00
25	5.00



-Carga 47R.

V entrada positiva (V)	V salida (V)
1	0.061
2	0.492
3	1.433
4	2.405
5	3.37
6	4.34
7	5.01
8	5.01
9	5.00
10	5.00
11	5.00
12	5.00
13	5.00
15	5.00

-Carga 10R.

V entrada positiva (V)	V salida (V)
1	0.015
2	0.407
3	1.318
4	2.251
5	3.194
6	4.12
7	4.93
8	4.92
9	4.92
10	4.94
11	4.94
12	4.94



Para el resto de medidas la fuente no puede suministrar la corriente necesaria por ser ésta mayor de 1 A.

Tensión máxima de regulación: 23.45V

2.2 Regulamos la tensión a 12.06 V (problemas con la resolución del potenciómetro).

-Carga en vacío.

V entrada positiva (V)	V salida (V)
1	0.297
2	1.196
3	2.117
4	3.023
5	3.94
6	4.88
7	5.78
8	6.71
9	7.63
10	8.58
11	9.55
12	10.53
13	11.47
14	12.03
15	12.04
16	12.04
17	12.05
18	12.05
19	12.05
20	12.05
21	12.05
22	12.05
23	12.05
25	12.05



-Carga 47R.

V entrada positiva (V)	V salida (V)
1	0.061
2	0.494
3	1.436
4	2.407
5	3.38
6	4.35
7	5.34
8	6.32
9	7.31
10	8.29
11	9.28
12	10.26
13	11.25
14	12.12
15	12.05
16	12.01
17	12.00
18	11.98
19	11.98
20	11.98
21	11.98

Nota: baja la tensión regulada por el calentamiento de los componentes.



-Carga 10R.

V entrada positiva (V)	V salida (V)
1	0.015
2	0.405
3	1.323
4	2.262
5	3.208
6	4.15
7	5.10
8	6.05
9	7.00
10	7.96
11	8.92
12	9.87

Para el resto de medidas la fuente no puede suministrar la corriente necesaria.



2.3 Datos medidos con deficiencias en el circuito (transistores al revés).

-Regulamos la tensión a 10V.

-Carga en vacío.

V entrada positiva (V)	V salida (V)
1	0.296
2	1.199
3	2.059
4	2.865
5	3.66
6	4.55
7	5.52
8	6.50
9	7.48
10	8.47
11	9.46
12	10.00
13	10.00
14	10.00
15	10.00
25	10.00



-Carga 1k

V entrada positiva (V)	V salida (V)
1	0.238
2	1.019
3	1.779
4	2.560
5	3.55
6	4.48
7	5.47
8	6.46
9	7.45
10	8.44
11	9.44
12	9.99
13	10.00
25	10.00



-Carga 100k

V entrada positiva (V)	V salida (V)
1	0.289
2	1.192
3	2.054
4	2.859
5	3.65
6	4.55
7	5.51
8	6.49
9	7.48
10	8.46
11	9.45
12	10.00
13	10.00
14	10.00
15	10.00
16	10.00
17	10.00
18	10.00
19	10.00
20	10.00
21	10.00
25	10.00



Descripción del circuito.

- Condensadores C1 y C2: estabilizan la señal que llega a la entrada del LM317 y LM337 respectivamente.
- P1 y P2: girando su eje ajustamos la tensión que queremos tener fija a la salida del LM317 y LM337 respectivamente, variando la corriente de emisor de los transistores.
- R3, R4, Q1 y Q2: adaptan los dos potenciómetros de 20k para que puedan suplir a los de 2k que son difíciles de conseguir.
- Condensadores C5 y C6: estabilizan la tensión de salida regulada.

Conclusiones tras las toma de medidas:

- **Tensión máxima de regulación: 23.45V**
- **Los reguladores LM317 regulan a la tensión fijada por el potenciómetro siempre que caigan sobre él una tensión mayor a 2 voltios.**



3. PROGRAMA DEL PROYECTO.

3.1 Programa placa actuador.

```
#include <isi.h>
#include <mem.h>
#include <stdlib.h>
#include <control.h>
#include <snvt_cfg.h>

// El numero de entradas en la tabla de alias es 16 debido a que se recomiendan 4
// como minimo para trabajar con 2 variables de red.
#pragma num_alias_table_entries 16

// NumAddr = min(15, NumAlias + NvCount)
#pragma num_addr_table_entries 15

// directivas de compilador varias:
#pragma enable_io_pullups
#pragma codegen optimization_on
#pragma codegen cp_family_space_optimization

// Variables de red
network output SNVT_switch nvoLampValueFb1;
network input SNVT_switch nviLampValue1;

network output SNVT_switch nvoLampValueFb2;
network input SNVT_switch nviLampValue2;
```



```
network output SNVT_switch nvoLampValueFb3;  
network input SNVT_switch nviLampValue3;
```

```
network output SNVT_switch nvoLampValueFb4;  
network input SNVT_switch nviLampValue4;
```

```
// Bloque funcional actuador 1.  
fblock SFPTlampActuator {  
    nvoLampValueFb1 implements nvoLampValueFb;  
    nviLampValue1 implements nviLampValue;  
} fbactuador1;
```

```
// Bloque funcional actuador 2.  
fblock SFPTlampActuator {  
    nvoLampValueFb2 implements nvoLampValueFb;  
    nviLampValue2 implements nviLampValue;  
} fbactuador2;
```

```
// Bloque funcional actuador 3.  
fblock SFPTlampActuator {  
    nvoLampValueFb3 implements nvoLampValueFb;  
    nviLampValue3 implements nviLampValue;  
} fbactuador3;
```

```
// Bloque funcional actuador 4.  
fblock SFPTlampActuator {  
    nvoLampValueFb4 implements nvoLampValueFb;  
    nviLampValue4 implements nviLampValue;  
} fbactuador4;
```

```
// Declaración de las salidas del FT 3120 SMART TRANSCEIVER
```

```
IO_4 output bit Luz1;  
IO_5 output bit Luz2;  
IO_6 output bit Luz3;  
IO_7 output bit Luz4;
```

```
// Encendido/apagado de los puntos de luz.
```



```
when(nv_update_occurs(nviLampValue1)){
    // Se encendera/apagara el punto de luz dependiendo del valor que venga
    // de la placa switch, contenido en la entrada nviLampValueX.
    if(nviLampValue1){
        io_out(Luz1, TRUE);
    }
    else{
        io_out(Luz1, FALSE);
    }

    nvoLampValueFb1 = nviLampValue1;
}

when(nv_update_occurs(nviLampValue2)){

    if(nviLampValue2){
        io_out(Luz2, TRUE);
    }
    else{
        io_out(Luz2, FALSE);
    }

    nvoLampValueFb2 = nviLampValue2;
}

when(nv_update_occurs(nviLampValue3)){

    if(nviLampValue3){
        io_out(Luz3, TRUE);
    }
    else{
        io_out(Luz3, FALSE);
    }

    nvoLampValueFb3 = nviLampValue3;
}
```



```
when(nv_update_occurs(nviLampValue4)){  
  
    if(nviLampValue4){  
        io_out(Luz4, TRUE);  
    }  
    else{  
        io_out(Luz4, FALSE);  
    }  
  
    nvoLampValueFb4 = nviLampValue4;  
}
```

3.2 Programa placa interruptor.

```
#include <isi.h>  
#include <mem.h>  
#include <stdlib.h>  
#include <control.h>  
#include <snvt_cfg.h>  
  
// El numero de entradas en la tabla de alias es 16 debido a que se recomiendan 4  
// como minimo para trabajar con 2 variables de red.  
#pragma num_alias_table_entries 16  
  
// NumAddr = min(15, NumAlias + NvCount)  
#pragma num_addr_table_entries 15  
  
// directivas de compilador varias:  
#pragma enable_io_pullups  
#pragma codegen optimization_on  
#pragma codegen cp_family_space_optimization  
  
// Variables de red  
network output bind_info(unackd_rpt) SNVT_switch nvoSwitch1;  
network input SNVT_switch nviSwitchFb1;  
network output bind_info(unackd_rpt) SNVT_switch nvoSwitch2;  
network input SNVT_switch nviSwitchFb2;
```




```
network output bind_info(unackd_rpt) SNVT_switch nvoSwitch3;  
network input  SNVT_switch nviSwitchFb3;  
network output bind_info(unackd_rpt) SNVT_switch nvoSwitch4;  
network input  SNVT_switch nviSwitchFb4;
```

```
// Bloque funcional pulsador 1.
```

```
fblock SFPTswitch {  
    nvoSwitch1 implements nvoSwitch;  
    nviSwitchFb1 implements nviSwitchFb;  
} fbswitch1;
```

```
// Bloque funcional pulsador 2.
```

```
fblock SFPTswitch {  
    nvoSwitch2 implements nvoSwitch;  
    nviSwitchFb2 implements nviSwitchFb;  
} fbswitch2;
```

```
// Bloque funcional pulsador 3.
```

```
fblock SFPTswitch {  
    nvoSwitch3 implements nvoSwitch;  
    nviSwitchFb3 implements nviSwitchFb;  
} fbswitch3;
```

```
// Bloque funcional pulsador 4.
```

```
fblock SFPTswitch {  
    nvoSwitch4 implements nvoSwitch;  
    nviSwitchFb4 implements nviSwitchFb;  
} fbswitch4;
```

```
// Declaración de las entradas del FT 3120 SMART TRANSCEIVER
```

```
IO_4 input bit Pulsador1;  
IO_5 input bit Pulsador2;  
IO_6 input bit Pulsador3;  
IO_7 input bit Pulsador4;
```



// Captura de la señal del pulsador y correccion del rebote.

```
#define APP_MS_TICK 50
mtimer repeating kbTick = APP_MS_TICK;
```

```
when(timer_expires(kbTick)) {
```

```
    unsigned Boton1, Boton2, Boton3, Boton4, i;
```

```
    Boton1 = 1;
```

```
    Boton2 = 1;
```

```
    Boton3 = 1;
```

```
    Boton4 = 1;
```

```
    for( i = 0; i < MG_BUTTONS_DEBOUNCE; ++i){
```

```
        Boton1 &= io_in(Pulsador1);
```

```
        Boton2 &= io_in(Pulsador2);
```

```
        Boton3 &= io_in(Pulsador3);
```

```
        Boton4 &= io_in(Pulsador4);
```

```
        // Se obtiene para cada pulsador si ha sido pulsado o no.
```

```
    }
```

```
    // Asignamos las ordenes de los pulsadores a las variables de red.
```

```
    if(!Boton1){
```

```
        // El pulsador se activa en nivel bajo.
```

```
        // Se conmuta la luz.
```

```
        nvoSwitch1 = ~nviSwitchFb1;
```

```
        Boton1 = 1;
```

```
    }
```

```
    if(!Boton2){
```

```
        // El pulsador se activa en nivel bajo.
```



```
// Se conmuta la luz.
    nvoSwitch2 = ~nviSwitchFb2;
    Boton2 = 1;
}

if(!Boton3){
// El pulsador se activa en nivel bajo.

// Se conmuta la luz.
    nvoSwitch3 = ~nviSwitchFb3;
    Boton3 = 1;
}

if(!Boton4){
// El pulsador se activa en nivel bajo.

// Se conmuta la luz.
    nvoSwitch4 = ~nviSwitchFb4;
    Boton4 = 1;
}
}
```



Sevilla a 14 de Setiembre de 2009

El Ingeniero Técnico Industrial

Joaquín Ramos Domínguez