

Material para a formación profesional inicial

Familia profesional	ELE	Electricidade e Electrónica
Ciclo formativo	CSELE04	Mantemento Electrónico
Grao		Superior
Módulo profesional	MP1052	Equipamentos Microprogramables
Unidade didáctica	UD03	Circuitos auxiliares, periféricos e de comunicacións
Actividade	A01	Dispositivos auxiliares
Autores		Javier Diz Bugarín, Jorge Domínguez Balado, Fernando J. García González, Antón Infante Taboada, José Manuel Ceide López, Manuel Martínez Méndez, M. Carmen Currás Francos, Fernando Rodríguez Rodríguez
Nome do arquivo		CSELE04_MP1052_V000301_UD03_A01_Dis_Aux
© 2014 Xunta de Galicia. Consellería de Cultura, Educación e Ordenación Universitaria. Este traballo foi realizado durante unha licenza de formación retribuída pola Consellería de Cultura, Educación e Ordenación Universitaria e ten licenza Creative Commons BY-NC-SA (recoñecemento - non comercial - compartir igual). Para ver unha copia desta licenza, visitar a ligazón http://creativecommons.org/licenses/by-nc-sa/3.0/es/.		

Índice

1.Ficha técnica.....	3
Contexto da actividade.....	3
Título da actividade.....	3
Resultados de aprendizaxe do currículo.....	3
Obxectivos didácticos e título e descrición da actividade.....	4
Criterios de avaliación.....	4
Contidos	4
Actividades de ensino e aprendizaxe e de avaliación, métodos, recursos e instrumentos de avaliación.....	6
2.A1. Dispositivos auxiliares.....	7
2.1.Introdución.....	7
2.2.Tarefas.....	8
2.2.1.Tarefa 1. Instalación do contorno de programación ECLIPSE-SDCC.....	8
2.2.2.Tarefa 2. Construción dos módulos hardware do sistema SISDECMi51.....	13
2.2.3.Tarefa 3. Módulo con sensor de ruído e convertedor analóxico a dixital.....	17
2.2.4.Tarefa 4. Módulo con sensor analóxico de temperatura (resistencia NTC) e convertedor analóxico a dixital.....	22
2.2.5.Tarefa 5. Módulo con sensor acelerómetro dixital FXOS8700CQR1.....	28
2.2.6.Tarefa 6. Módulo con sensor acelerómetro dixital LSM303.....	29
2.2.7.Tarefa 7. Medidor de velocidade con barreira luminosa.....	30
2.2.8.Tarefa 8. Sensores e actuadores en instalacións domóticas.....	37
3.Avaliación.....	42



1. Ficha técnica

Contexto da actividade

Módulo	Duración	Unidade didáctica.	Sesións 50'	Actividades	Sesións 50'
MP1052. Equipamentos Microprogramables	133	UD01. Fundamentos de loxica dixital e circuitos combinacionais.	62	A01. Fundamentos de Álgebra de Boole.	30
				A02. Circuitos integrados SSI.	6
				A03. Circuitos combinacionais.	26
		UD02. Circuitos secuenciais	40	A01. Circuitos secuenciais básicos.	18
				A02. Manexo do Osciloscopio.	4
				A03. Circuitos con biestables	18
		UD03. Circuitos auxiliares, periféricos e de comunicacións.	80	A01. Dispositivos auxiliares.	30
				A02. Circuitos mixtos.	30
				A03. Memorias.	20
		UD04. Unidades de control microprogramables.	137	A01. Sistemas microprogramables.	20
				A02. Programación de microcontroladores MCS51.	82
				A03. Proxectos con microcontroladores MCS51.	35

Título da actividade

Nº	Título	Descrición	Duración
A01	Dispositivos auxiliares	Dispositivos auxiliares de entrada e saída: visualizadores led/LCD, teclados, sensores , interfaces de potencia e motores paso a paso.	30

Resultados de aprendizaxe do currículo

Resultados de aprendizaxe do currículo	Completo
▪ RA1 - Configura dispositivos periféricos e auxiliares en sistemas microprocesados, comproba o seu funcionamento e verifica as súas prestacións.	Sí
▪ RA3 - Configura equipamentos dixitais microprogramables, programando funcións segundo a súa aplicación.	Non
▪ RA4 - Desenvolve pequenas aplicacións electrónicas con circuitos microprogramables, elaborando os programas de control e utilizando a documentación técnica e as solucións estándar dispoñibles.	Non
▪ RA5 - Mantén equipamentos electrónicos dixitais e microprogramables, arranzando avarías e disfuncións	Non

Obxectivos didácticos e título e descrición da actividade

Obxectivos específicos		Actividade		Descrición básica	Duración
O1.1	Que o alumno saiba conectar elementos de entrada e saída nos circuitos dixitais así como interpretar as características técnicas de ditos elementos	A1	Dispositivos auxiliares	Dispositivos auxiliares de entrada e saída: visualizadores led/LCD, teclados, sensores , interfaces de potencia e motores paso a paso.	30

Critérios de avaliación

Criterios de avaliación
▪ CA1.1 Interpretáronse esquemas e bloques funcionais. ▪ CA1.2 Identificáronse tipos de memoria. RAM. Estáticas. Dinámicas. ROM, PROM, EPROM, EEPROM e FLASH. ▪ CA1.3 Montáronse circuitos multivibradores, osciladores e circuitos PLL. ▪ CA1.4 Comprobouse o funcionamento dos conversores DAC/ADC. ▪ CA1.5 Comprobouse o funcionamento de teclados, visualizadores, etc. ▪ CA1.6 Configuráronse controladores de portos de entrada e saídas dixitais. ▪ CA1.7 Configuráronse parámetros de funcionamento de periféricos e sistemas auxiliares. ▪ CA3.7 Verificouse o funcionamento do circuito microprogramable e os seus elementos auxiliares. ▪ CA4.2 Elaboráronse os diagramas de bloques necesarios para resolver a nivel funcional a aplicación. ▪ CA4.3 Elaboráronse os esquemas eléctricos de principio, realizando ou adaptando, a partir de circuitos similares, os circuitos correspondentes a cada bloque funcional da aplicación. ▪ CA4.4 Realizáronse os cálculos dos circuitos aplicando as regras e as fórmulas adecuadas. ▪ CA4.5 Seleccionouse a tecnoloxía e os compoñentes do circuito a partir dos manuais de compoñentes, asegurando a súa dispoñibilidade. ▪ CA4.8 Construíuse a maqueta, realizando a montaxe do circuito, aplicando os procedementos adecuados. ▪ CA5.1 Resolvéronse disfuncións en circuitos combinacionais e secuenciais. ▪ CA5.2 Identificáronse os síntomas da disfunción ou avaría (fallos de comunicación, bloqueos de programa, ausencia de sinais de saída, etc.). ▪ CA5.3 Diagnosticouse a avaría de acordo coa disfunción atopada (control de portos, alimentación, fallo de programa, instrucións erróneas, etc.). ▪ CA5.4 Realizáronse medidas (oscilador de reloxo, transmisión de datos, valores de entrada e saída, etc.). ▪ CA5.5 Determinouse a avaría segundo os valores dos parámetros obtidos. ▪ CA5.6 Substituíuse o compoñente ou circuito dixital responsable da avaría.

Contidos

Contidos
▪ Bloques funcionais de dispositivos periféricos e auxiliares en sistemas microprocesados. ▪ Memorias: tipos. Sinais de control. Programación de memorias. ▪ Multivibradores. Temporizadores. Circuitos PLL. Tipos. Características. Parámetros de funcionamento. ▪ Convertedores de datos (DAC-ADC). Análise de entradas e saídas en conversores DAC-ADC. Parámetros de funcionamento. ▪ Dispositivos de entrada e saída. Teclados. Visualizadores. Pantallas LCD. Parámetros de funcionamento. ▪ Portos de comunicacións. Controladores de bus. Buses. Tipos. Características. ▪ Esquemas, planos e instrucións de montaxe e posta a punto.

Contidos

- Probas funcionais, de calidade e de fiabilidade.
- Simbología normalizada.
- Programas de simulación electrónica, instrucións e funcionamento.
- Tipoloxía de avarías en circuitos electrónicos dixitais e microprogramables.
- Localización de avarías en circuitos electrónicos combinacionais e secuenciais.
- Instrumentación de laboratorio utilizada na reparación de avarías en circuitos dixitais e microprogramables.
- Prevención de danos por descargas electrostáticas.
- Ferramentas software para a elaboración de informes.

Actividades de ensino e aprendizaxe e de avaliación, métodos, recursos e instrumentos de avaliación

Qué e para qué	Cómo			Con qué	Cómo e con qué se valora	Duración (sesións)
Actividade (título e descrición)	Profesorado (en termos de tarefas)	Alumnado (tarefas)	Resultados ou produtos	Recursos	Instrumentos e procedementos de avaliación	
A1. Dispositivos auxiliares Dispositivos auxiliares de entrada e saída: visualizadores led/LCD, teclados, sensores, interfaces de potencia e motores paso a paso.	Tp1.1 Explicación das características do contorno de programación ECLIPSE-SDCC. Tp1.2 Explicación dos elementos do sistema SISDECM151 e as características dos sensores dixitais e analóxicos empregados. Tp1.3 Explicación de circuíto con sensor analóxico de ruído ambiente e convertedor analóxico a dixital. Tp1.4 Explicación de circuíto con sensor analóxico de temperatura (resistencia NTC) e convertedor analóxico a dixital. Tp1.5 Explicación de circuíto con sensor acelerómetro dixital FXOS8700CQR1. Tp1.6 Explicación de circuíto con sensor acelerómetro dixital LSM303 Tp1.7 Explicación do principio de medidade velocidade con láser diodo e barreira luminosa. Tp1.8 Explicación das formas de utilización de sensores e actuadores en instalacións domóticas.	Ta1.1 Instalación do contorno de programación ECLIPSE-SDCC. Ta1.2 Construción e montaxe das placas de circuíto impreso do microcontrolador e periféricos do sistema SISDECM151. Ta1.3 Montaxe e programación de circuíto con sensor analóxico de ruído ambiente e convertedor analóxico a dixital. Ta1.4 Montaxe e programación de circuíto con sensor analóxico de temperatura (resistencia NTC) e convertedor analóxico a dixital. Ta1.5 Montaxe e programación de circuíto con sensor acelerómetro dixital FXOS8700CQR1. Ta1.6 Montaxe e programación de circuíto con sensor acelerómetro dixital LSM303 Ta1.7 Montaxe dun medidor de velocidade con láser diodo e barreira luminosa. Ta1.8 Montaxes con diferentes sensores e actuadores para instalacións domóticas.	- Circuítos e montaxes realizadas. - Medidas e verificacións efectuadas sobre os circuítos. - Programas elaborados.	- Instrumentación e ferramenta do laboratorio de electrónica. - Ordenadores dos alumnos con sistemas operativos Linux e Windows. - Información técnica (datasheets).	- Cuestionario sobre os contidos e tarefas da actividade. - Proba teórico-práctica sobre a actividade realizada.	30

2. A1. Dispositivos auxiliares

2.1. Introducción

Esta actividade consiste na realización de prácticas de lectura de diferentes tipos de sensores analóxicos e dixitais mediante un sistema de desenvolvemento de aplicacións con microcontroladores da familia MCS51 (SISDECMI51) e a realización dos programas necesarios tanto en ensamblador como en linguaxes de alto nivel (C). A actividade está dividida en diferentes tarefas que consisten na montaxe paso a paso dos elementos hardware e software necesarios para a realización das prácticas:

- TAREFA 1: Instalación do contorno de programación ECLIPSE-SDCC. Este contorno permite a realización posterior dos programas de control que se transferirán ó módulos SISDECMI.
- TAREFA 2: Construción dos módulos hardware do sistema SISDECMI. Este sistema está formado por unha unidade principal que leva o microcontrolador, un módulo de interfaz serie-usb para comunicación co ordenador e un conxunto de periféricos.
- TAREFA 3: Montaxe de módulo con sensor de ruído, circuito analóxico de interfaz e convertedor analóxico a dixital.
- TAREFA 4: Montaxe con sensor analóxico de temperatura NTC.
- TAREFA 5: Montaxe con sensor acelerómetro dixital FXOS8700CQR1.
- TAREFA 6: Montaxe con sensor acelerómetro dixital LSM303.
- TAREFA 7: Medidor de velocidade con láser diodo e barreira luminosa.
- TAREFA 8: Sensores e actuadores en instalacións domóticas.



2.2. Tarefas



2.2.1. Tarefa 1. Instalación do contorno de programación ECLIPSE-SDCC

Esta tarefa consiste na localización e instalación do conxunto de programas que constitúen o contorno de programación de microcontroladores eclipse-sdcc para mcs51.

OBTENCIÓN DO CÓDIGO FONTE

O código fonte pode obterse da páxina do proxecto eclipse e está formado polos seguintes elementos:

- Eclipse 3.2.1
- CDT 3.1.2. Este complemento cdt tamén se pode instalar desde eclipse na opción "help" "software updates"
- Compilador eclipse-sdcc 1.0.0
- Contorno J2SE Runtime Environment (JRE) 1.4.2-08

CONFIGURACIÓN DE ECLIPSE

Unha vez descomprimidos todos os arquivos de instalación hai que engadir o contidos das carpetas "features" e "plugins" de cdt ás carpetas do mesmo nome de eclipse. Facer o mesmo coas carpetas "features" e "plugins" de eclipse-sdcc copiando o contido ás carpetas do mesmo nome de eclipse. TODAS as versións deben coincidir ou pode que non funcione o conxunto.

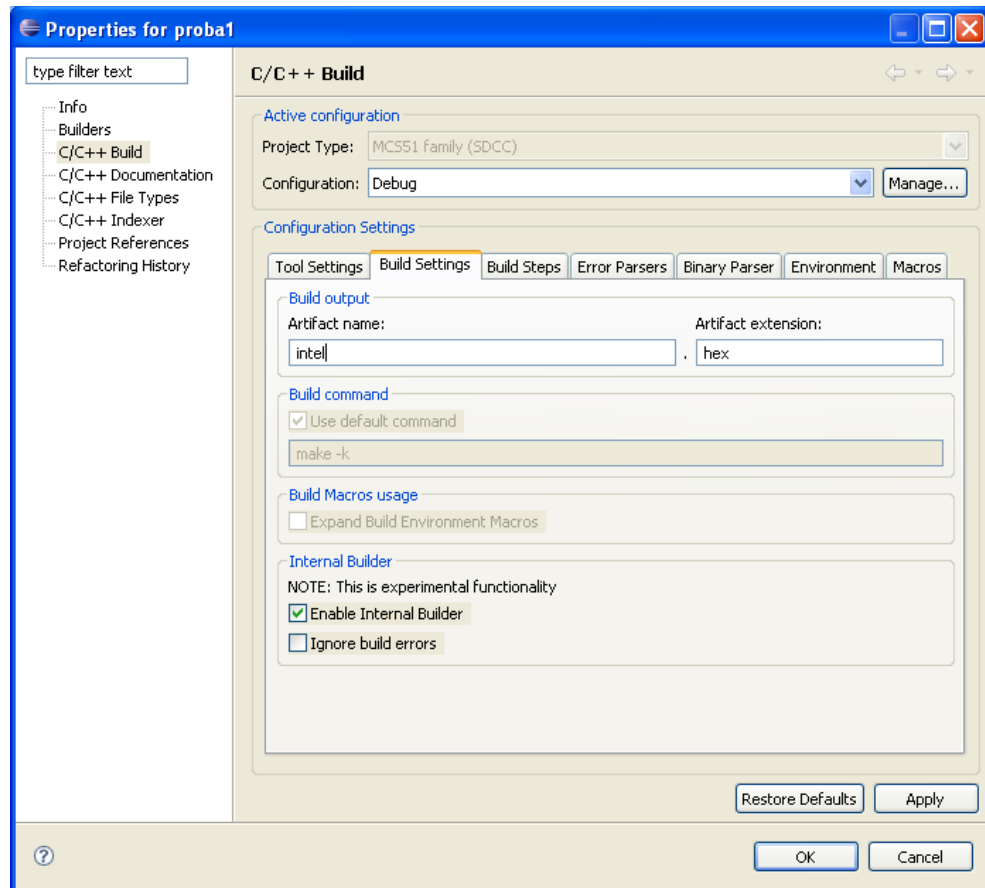
OUTROS PROGRAMAS

- Unha vez instalado eclipse, copiar da web o compilador sdcc 2.6.0 e instalalo. O mesmo para o compilador C++ MinGW 1.1 e descomprimir nunha carpeta (MinGW = Minimalist GNU for Windows, tamén vale o Cygwin).
- Engadir á variable PATH as rutas de "()/sdcc/bin" e "()/mingw/bin" (tendo en conta a ubicación exacta na que estean). Pode facerse en Panel de Control->Sistema->Opcións avanzadas->Variables de contorno. Mellor colocalas ó inicio de PATH para evitar que se poida chamar a outros arquivos co mesmo nome existentes no sistema (p. ex. doutros compiladores).
- En lugar de engadir as rutas ó path pode escribirse a ruta completa nas opcións do compilador, por exemplo: "d:\eclipse\sdcc\bin\sdcc --debug -c" ou similar. Así non é necesario instalar nada, pero hai que facelo para cada proxecto. Se se elimina o nome de unidade pode copiarse a carpeta a diferentes unidades sen cambiar nada, por exemplo: "\\eclipse\sdcc\bin\sdcc --debug -c".
- Preferiblemente debe instalarse todo na unidade C:, evitando rutas longas e con espazos, signos, etc.
- O o contorno eclipse non ten instalación e pode cambiarse de unidade ou carpeta. Os compiladores tamén son relocizables, poden cambiarse de unidade e carpeta modificando as rutas para que se localicen na compilación.

- O compilador sdcc ten que estar na mesma unidade que os arquivos que se compilen.

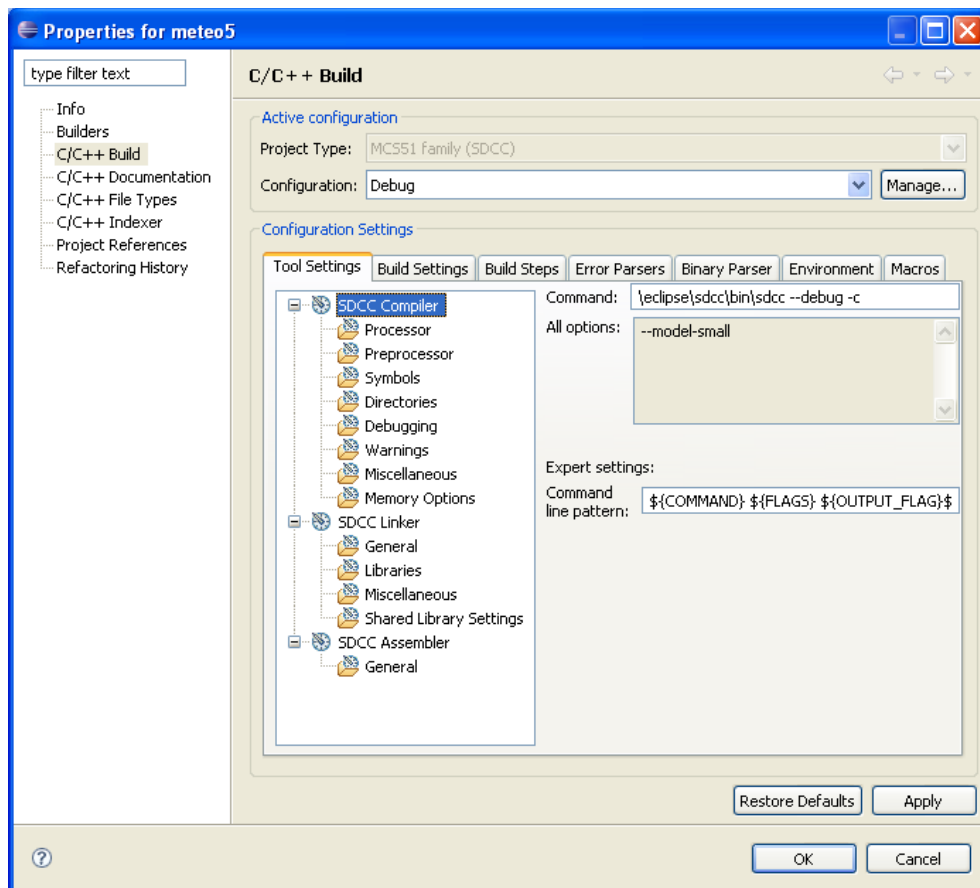
CONFIGURACIÓN DE PROXECTOS

- Iniciar eclipse e crear un proxecto co compilador sdcc (ver tutorial). Para compilar correctamente hai que habilitar o builder interno, en "Project" "Properties" "C/C++ Build" "Build Settings" marcar a casinha "Enable Internal Builder" e tamén quitar a casinha "Ignore Build Errors".
- Pódese modificar a extensión dos resultantes poñendo "hex" ou "ihx"
- Para evitar problemas cos nomes de arquivos é mellor seleccionar sempre como resultado do proxecto o nome intel.hex
- Se usamos os modos de compilación debug e release hai que cambiar a ruta en cada caso, mellor usar só un deles. Para iso cando se crea o proxecto seleccionar só "debug" ou "release".



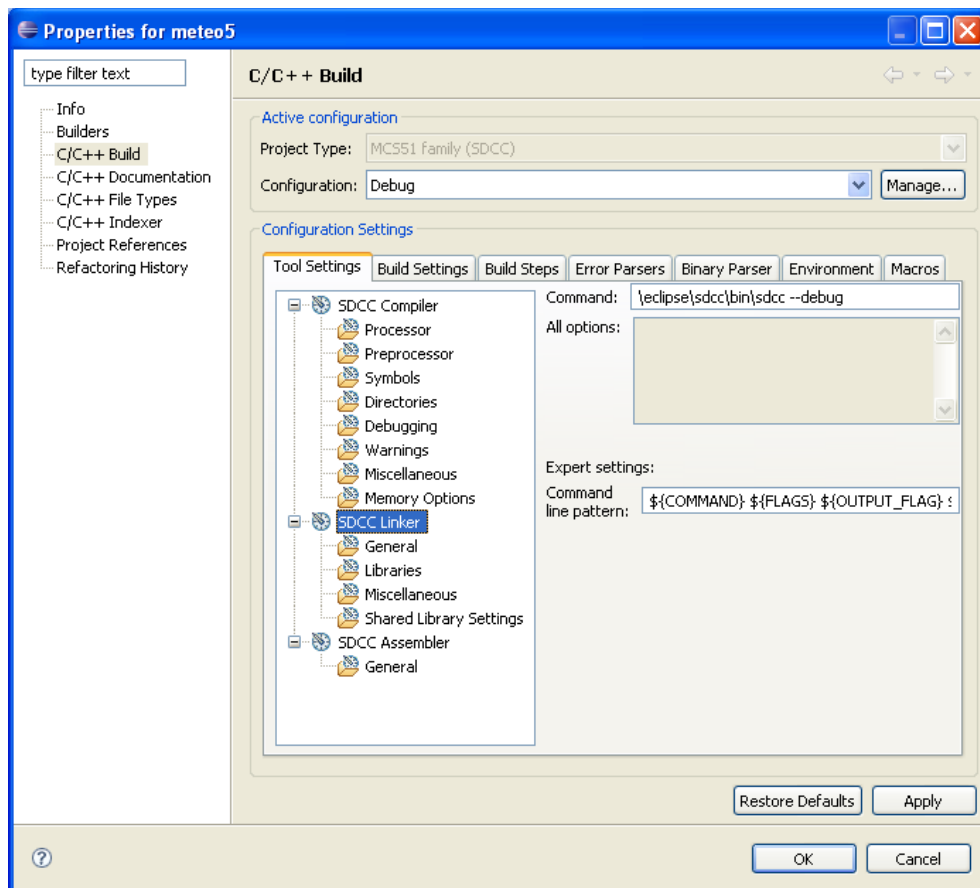
COMPILADOR

- Liña de comando: `"\eclipse\sdcc\bin\sdcc --debug -c"`
- Opcións: `"--model-small"`
- Formato de liña de comando: `"${COMMAND} ${FLAGS} ${OUTPUT_FLAG}${OUTPUT_PREFIX}${OUTPUT} ${INPUTS}"`



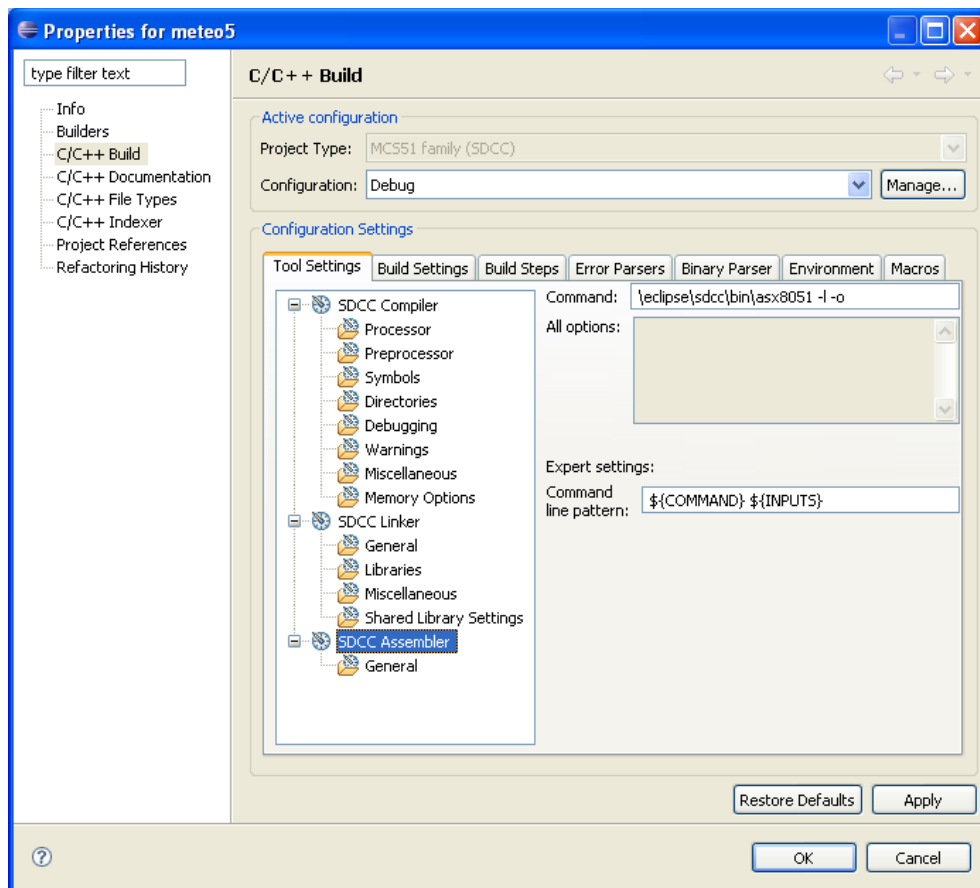
ENLAZADOR (LINKER)

- Liña de comando: “\\eclipse\\sdcc\\bin\\sdcc --debug”
- Opcións expertas: “\${COMMAND} \${FLAGS} \${OUTPUT_FLAG} \${OUTPUT_PREFIX} \${OUTPUT} .. \${INPUTS}”
- O ensamblador crea o arquivo de saída na mesma carpeta que o de entrada, mentres que sdcc espera os arquivos na subcarpeta "debug" ou "release", para evitar isto no linker hai que antepoñer "..\\" ós arquivos de entrada.



ENSAMBLADOR (ASSEMBLER)

- O que está incluído en sdcc é o ASX8051.exe, non o as51
- Hai que modificar a liña de comando do ensamblador a "<code>\\eclipse\\sdcc\\bin\\asx8051 -l -o</code>"
- Opcións: "<code>\${COMMAND} \${INPUTS}</code>"
- A opción -o crea o arquivo .rel, a opción -l crea o arquivo .lst
- Non hai que dar nome de arquivo de saída, só o de entrada, como: "<code>asx8051 -l -o meteo.asm</code>"



UTILIDADES

- utilidades para conversión de arquivos: HEX2BIN.EXE, BIN2HEX.EXE de Atmel:

http://www.atmel.com/dyn/resources/prod_documents/HEX2BIN.EXE

- Pódese executar Avrdude desde eclipse: ir a "Run" "External Tools" "External Tools" e engadir o nome de programa,

a ruta do executable, o directorio de traballo (pódese seleccionar con "Browse workspace") e as opcións. Por exemplo,

as opcións co 4051 e usbprog serían -p s4051 -c avrisp2 -P usb -B 10 -e -U eeprom:w:intel.hex

- na ruta do executable é mellor evitar o nome de unidade para permitir a relocación, como:

"\\eclipse\\isp\\avrdude\\avrdude.exe"

2.2.2. Tarefa 2. Construción dos módulos hardware do sistema SISDECMi51

Esta tarefa consiste na realización das placas de circuíto impreso do sistema de microcontrolador, a montaxe dos compoñentes e probas de funcionamento. Nesta tarefa tomaremos como exemplo a placa de circuíto impreso da versión reducida do proxecto, que incorpora o microcontrolador at89s4051 e un convertedor analóxico a dixital.

A construción das placas ten que facerse co material de uso común no taller de electrónica: placa fotosensible positiva, insoladora, sistema de revelado e atacado químico para pcb.

A continuación aparecen os esquemas de montaxe necesarios para a realización da placa.

PROCESO DE REALIZACIÓN DO CIRCUÍTO IMPRESO

O circuíto impreso pode realizarse por calquera dos métodos coñecidos, recomendándose o uso da insoladora. Pasos a seguir:

Insolado da placa (tempo aproximado: 2,5 minutos, recomendándose a realización de probas previas)

Revelado con líquido revelador ou sosa

Atacado con atacador ou combinación de ácido clorhídrico (augaforte) + auga osixenada

Eliminación da capa de barniz fotosensible

Taladrado da placa con broca de 1mm. Se hai compoñentes que non entran, ampliar os furados cunha broca de 2mm ou máis se é preciso.

Unha vez rematados os pasos anteriores é recomendable revisar tódalas pistas cuidadosamente para comprobar que non hai unións entre pistas, o que produciría cortocircuitos unha vez montada a placa.

MONTAXE DAS COMPOÑENTES NA PLACA

Hai que ter especial cuidado coa posición e polaridade dos condensadores e diodos. Na páxina seguinte aparece un plano coa situación e posición de tódalas compoñentes.

Na soldadura utiliza-la cantidade de estaño necesaria para conseguir unha boa suxección das compoñentes. Se son grandes ou pesados a soldadura debe ser forte, ou pode romper. Evita-la soldadura fría.

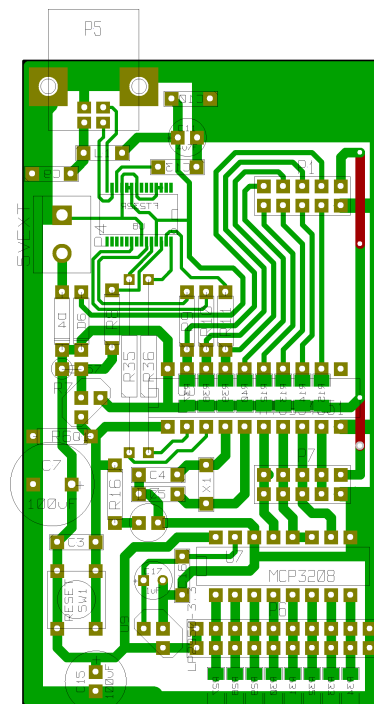
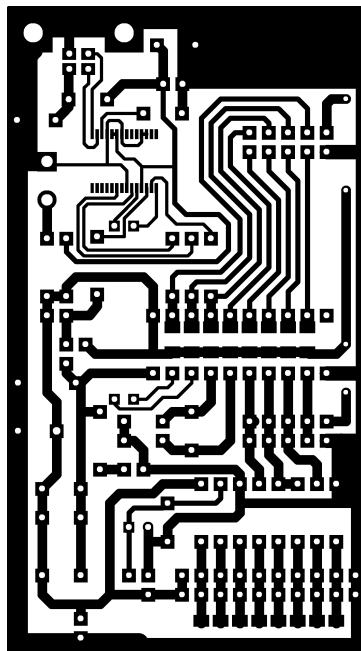
Os reguladores de tensión poden chegar a quentarse moito durante o funcionamento normal, polo que é recomendable que teñan un dissipador. Pode servir un perfil de aluminio calquera, ó que se lle farán os furados necesarios para a suxección dos reguladores con parafusos (M3).

ELEMENTOS ACCESORIOS

- **TRANSFORMADOR:** conéctase á placa con cables grosos, xa que teñen que soportar unha corrente bastante alta. Por un lado sóldanse ós terminais de baixa tensión do transformador e polo outro a terminais tipo faston. Na saída do transformador tamén se conecta a resistencia de $680\ \Omega$ e o diodo led indicador. Os terminais de 220V do transformador conéctanse a un cable de alimentación terminado nunha clavixa de enchufe. Neste cable tamén irá o fusible de entrada e un interruptor de 220V (tipo palanca)
- **SOPORTES:** os colocaremos nas catro esquinas da placa de circuito impreso.
- **CAIXA:** Se imos coloca-lo circuito nunha caixa hai que face-los furados axeitados para a montaxe de tódolos elementos (bananas, led, interruptor, saída do cable de rede e a placa de circuito impreso se vai con parafusos).

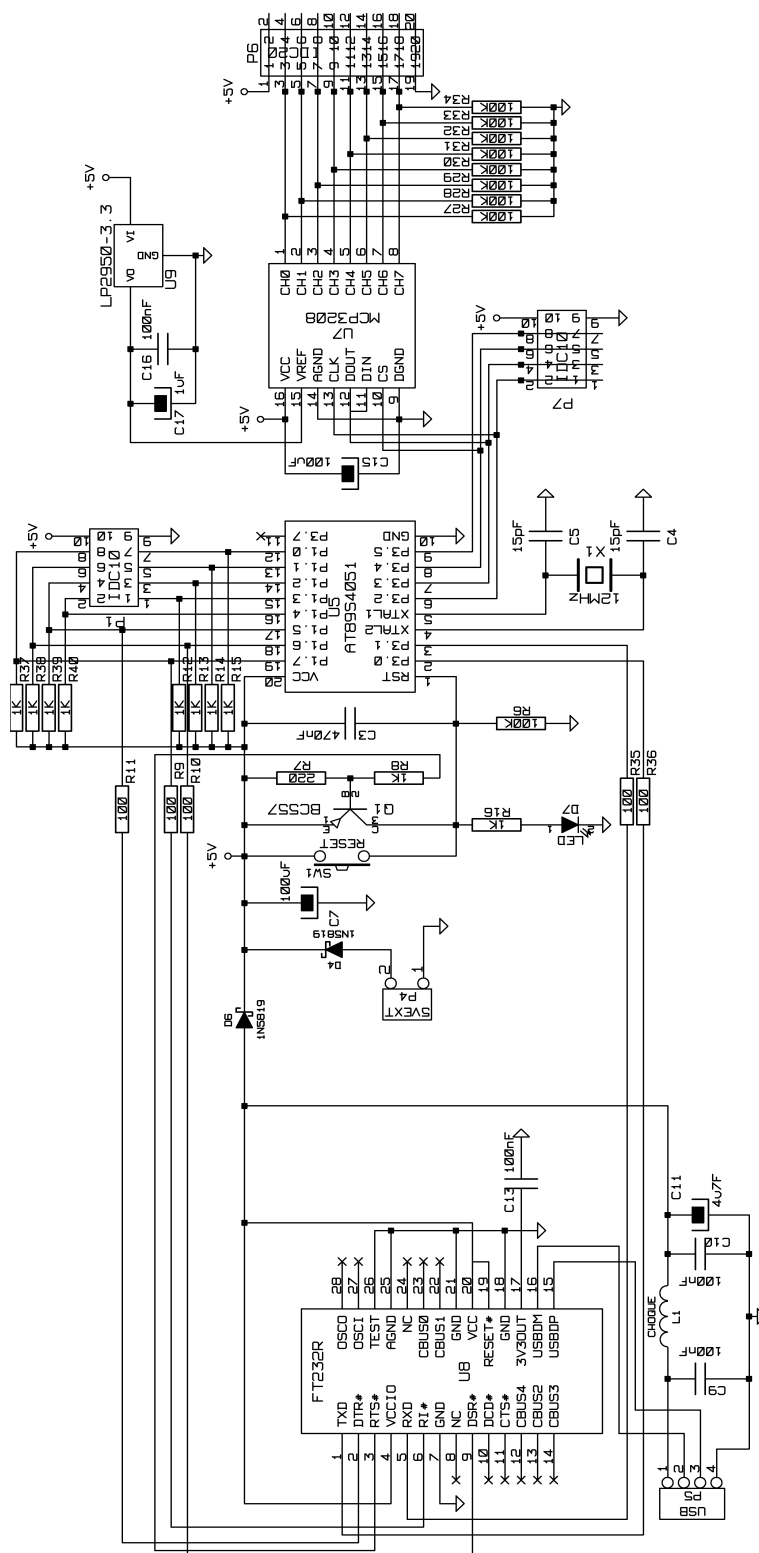
PLANO DE PISTAS E MONTAXE DE COMPOÑENTES

Nas figuras seguintes se representa o esquema de pistas (fotolito) e o plano de montaxe coa ubicación de tódolos compoñentes.



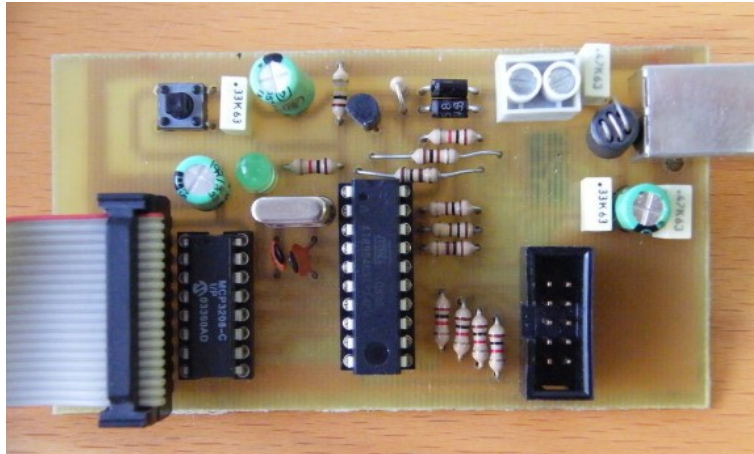
ESQUEMA ELECTRÓNICO

Na figura seguinte se representa o esquema electrónico da placa do microcontrolador e as conexións de entrada e saída:



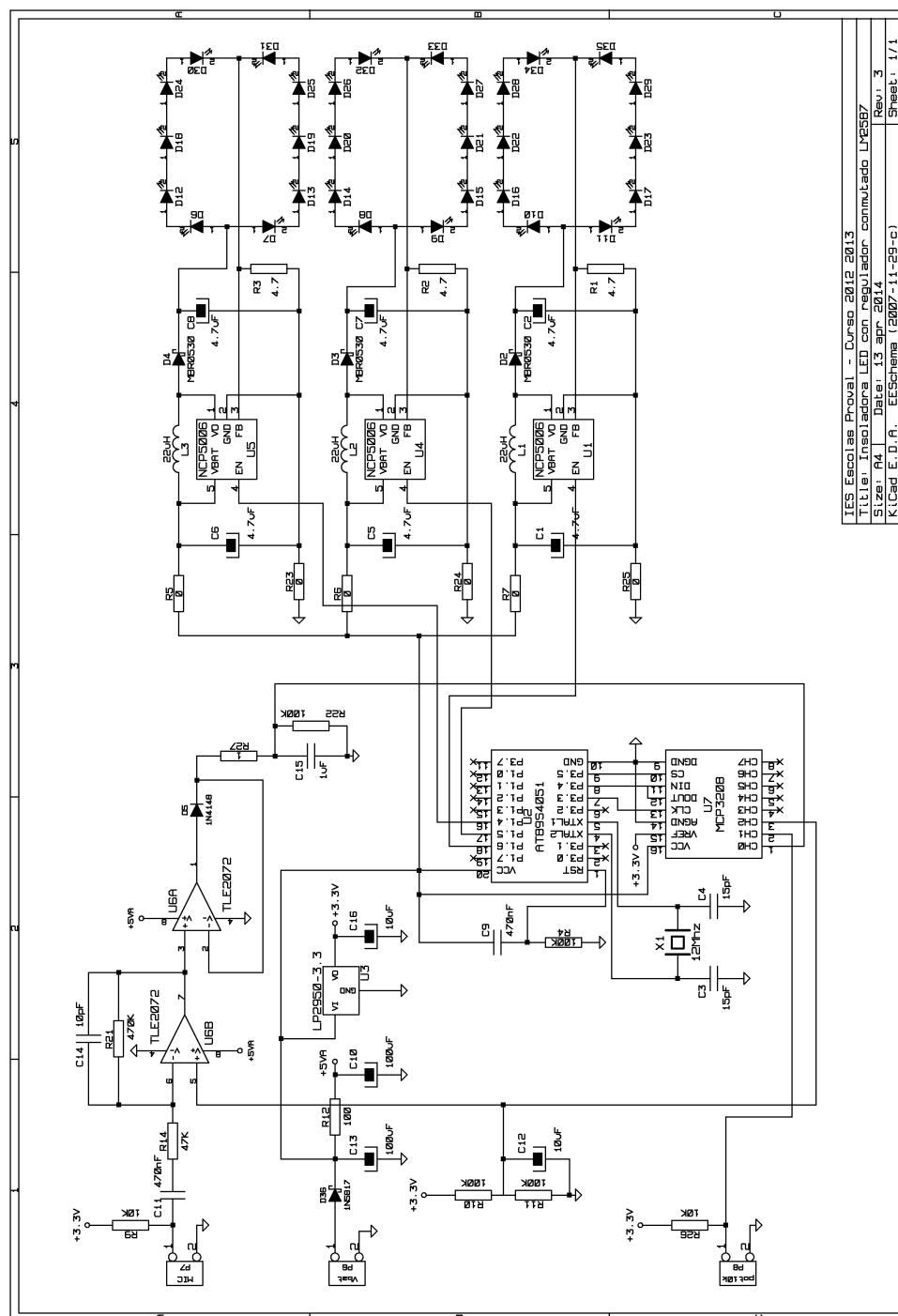
CONEXIONADO E VERIFICACIÓN

Unha vez completamente ensamblada a placa hai que comprobar que non hai avarías, empezando polas tensións de alimentación dos compoñentes. A continuación hai que conectar todos os cables e verificar que a placa funciona correctamente. Na figura seguinte pode verse o aspecto final do módulo completamente rematado:





ESQUEMA ELECTRÓNICO



PROGRAMA DE EXEMPLO

A continuación se presenta o código dun programa de exemplo (semaforo.c), que pode modificarse para adaptarse a distintos usos:

```
#define AD_CLK P3_3
#define AD_DAT P3_4 //nota: coa etiqueta ADDAT falla!!
#define AD_CS P3_5

//etiquetas para números de canles ad
#define NIVEL 0 //saida do detector de nivel de son
#define POT 1 //valor de comparacion (potenciómetro)
#define REF 2 //tensión de referencia analóxica

#define OFFSET 100 //valor de saida do detector de nivel con entrada nula

#define VERDE P1_4 //bit de saida que activa leds verdes
#define AMARELO P1_5 //bit de saida que activa leds amarelos
#define VERMELLO P1_6 //bit de saida que activa leds vermellos

#include <mcs51reg.h>
#include <8051.h>
#include <stdio.h>

////////////////////////////////////
// funciones de lectura mcp3208 por i2c
////////////////////////////////////
void ad_delay()
{
    _asm
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    _endasm;
}

////////////////////////////////////
void ad_pulso_clk()
{
    ad_delay();
    AD_CLK=1;
    ad_delay();
    AD_CLK=0;
    ad_delay();
}

////////////////////////////////////
unsigned int ad_lectura(unsigned char canle)
{
    bit b;
    unsigned int res;

    //inicio ad
```

```

AD_CS=1;  ad_delay();
AD_CLK=0; ad_delay();
//activa conversor
AD_CS=0;  ad_delay(); ad_delay();

AD_DAT=1; ad_delay(); ad_pulso_clk(); //bit start
AD_DAT=1; ad_delay(); ad_pulso_clk(); //modo single
//envia 3 bits co número de canle
if(canle&0x04) AD_DAT=1; else AD_DAT=0; ad_pulso_clk();
if(canle&0x02) AD_DAT=1; else AD_DAT=0; ad_pulso_clk();
if(canle&0x01) AD_DAT=1; else AD_DAT=0; ad_pulso_clk();
ad_delay(); //tempo de mostraxe ata o seguinte flanco de baixada
AD_DAT=1; ad_pulso_clk(); //pon dato=1 para permitir lectura
ad_pulso_clk(); //dato=0?
ad_delay();
res=0;
b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B12
b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B11
b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B10
b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B9
b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B8
b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B7
b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B6
b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B5
b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B4
b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B3
b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B2
b=AD_DAT; res=res|b; ad_pulso_clk(); //bit B1

//fin ad
AD_CS=1; //desactiva conversor
ad_delay();
AD_CLK=0;

return res;
}

////////////////////////////////////
///
void nop()
{
    _asm
    NOP
    _endasm;
}
////////////////////////////////////
void nop10()
{
    _asm
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    _endasm;
}
////////////////////////////////////
void nop100(void)

```

```

{
    nop10();nop10();nop10();nop10();nop10();
    nop10();nop10();nop10();nop10();nop10();
}
////////////////////////////////////
void main(void)
{
//variables para conversion ad
    unsigned int nivel,pot,ref,nivel_amarelo,nivel_vermello;
    unsigned int persistencia;

    persistencia=0;
    while(1)
    {

        nivel = ad_lectura(NIVEL);
        pot    = ad_lectura(POT);
        ref    = ad_lectura(REF);

        //inicia semáforo
        VERDE=1;
        AMARELO=0;
        VERMELLO=0;
        //se hubo un pico de ruido mantén o vermello acendido
        //e decrementa o contador
        if(persistencia>0)
        {
            persistencia--;
            if(persistencia&0x010)//cambia cada 16 contas
                VERMELLO=1;
            else
                VERMELLO=0;

        }

        //calcula referencias de comparación para acender amarelo e
vermello
        nivel_amarelo=ref + pot + OFFSET;
        nivel_vermello=ref + pot + pot + OFFSET;

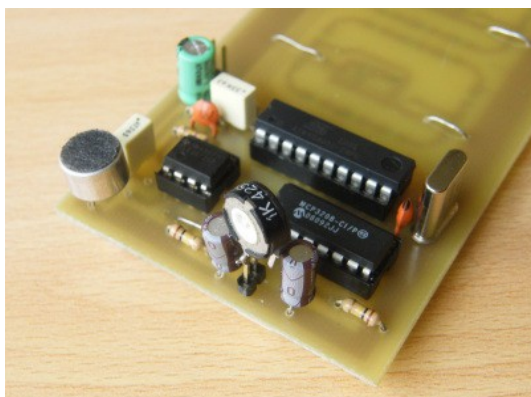
        //compara nivel de entrada con referencias e activa lámpadas
        if(nivel>nivel_amarelo) AMARELO=1;
        if(nivel>nivel_vermello) { VERMELLO=1; persistencia=1000; }

    } //fin de while
}

////////////////////////////////////
/

```

ASPECTO DA MONTAXE

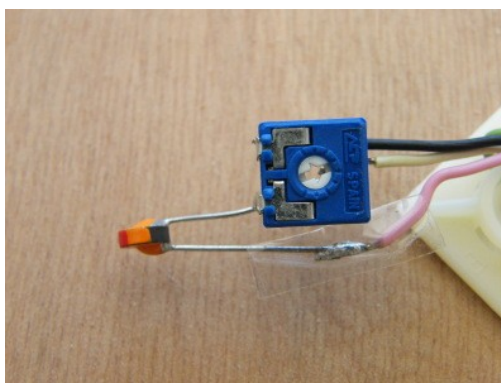


2.2.4. Tarefa 4. Módulo con sensor analóxico de temperatura (resistencia NTC) e convertedor analóxico a dixital

Esta tarefa consiste na conexión e programación dun sensor de temperatura analóxico con resistencia NTC. Este tipo de resistencias presentan unha resposta exponencial inversa coa temperatura, pero para evitar a complexidade de calibración pode facerse un sinxelo circuito divisor de tensión feito cun potenciómetro e a propia resistencia NTC. O potenciómetro permite axustar un nivel de umbral, o que pode empregarse para activar sistemas de calefacción, aire acondicionado, etc.

MONTAXE DO SENSOR

- A montaxe do sensor pode facerse aérea como na figura seguinte, co potenciómetro en serie configurando un divisor de tensión conectado á tensión de alimentación do circuito (+5V).
- A saída do divisor conéctase a unha das entradas do convertedor analóxico a dixital da placa do microcontrolador.



PROGRAMA DE EXEMPLO

A continuación se presenta o código dun programa de exemplo (domo.c), que pode modificarse para adaptarse a distintos usos:

```
// programa de control domótico simple

#include <mcs51reg.h>
#include <8051.h>
#include <stdio.h>

#define TEMPO1_ON 25 //número de retardos de activación do monoestable
#define TEMPO2_ON 5 //tempo do intermitente
#define UMBRAL_UP 2100 //umbral de comparación para sensor
#define UMBRAL_DN 2000 //umbral de comparación para sensor

// definicións de portos
#define PORTO_TEC1 P1_0
#define PORTO_TEC2 P1_1
#define PORTO_TEC3 P1_2
#define PORTO_TEC4 P1_3
#define PORTO_LED1 P1_7
#define PORTO_LED2 P1_6
#define PORTO_LED3 P1_5
#define PORTO_LED4 P1_4
#define PORTO_RELE1 P2_4
#define PORTO_RELE2 P2_5
#define PORTO_RELE3 P2_6
#define PORTO_RELE4 P2_7

// pins do convertedor mcp3208
#define AD_CLK P3_2
#define AD_DAT P3_3 //nota: coa etiqueta ADDAT falla!!
#define AD_CS P3_4

/////////////////////////////////////////////////////////////////
// funcións de lectura mcp3208 por i2c
/////////////////////////////////////////////////////////////////
void ad_delay()
{
    _asm
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
    _endasm;
}

/////////////////////////////////////////////////////////////////
void ad_pulso_clk()
{
    ad_delay();
    AD_CLK=1;
}
```

```

    ad_delay();
    AD_CLK=0;
    ad_delay();
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
unsigned int ad_lectura(unsigned char canle)
{
    bit b;
    unsigned int res;

    //inicio ad
    AD_CS=1;  ad_delay();
    AD_CLK=0; ad_delay();
    //activa conversor
    AD_CS=0;  ad_delay(); ad_delay();

    AD_DAT=1; ad_delay(); ad_pulso_clk(); //bit start
    AD_DAT=1; ad_delay(); ad_pulso_clk(); //modo single
    //envía 3 bits co número de canle
    if(canle&0x04) AD_DAT=1; else AD_DAT=0; ad_pulso_clk();
    if(canle&0x02) AD_DAT=1; else AD_DAT=0; ad_pulso_clk();
    if(canle&0x01) AD_DAT=1; else AD_DAT=0; ad_pulso_clk();
    ad_delay(); //tempo de mostraxe ata o seguinte flanco de baixada
    AD_DAT=1; ad_pulso_clk(); //pon dato=1 para permitir lectura
    ad_pulso_clk(); //dato=0?
    ad_delay();
    res=0;
    b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B12
    b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B11
    b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B10
    b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B9
    b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B8
    b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B7
    b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B6
    b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B5
    b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B4
    b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B3
    b=AD_DAT; res=res|b; res=res<<1; ad_pulso_clk(); //bit B2
    b=AD_DAT; res=res|b; ad_pulso_clk(); //bit B1

    //fin ad
    AD_CS=1; //desactiva conversor
    ad_delay();
    AD_CLK=0;

    return res;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
///
void nop10()
{
    _asm
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
}

```

```

        _endasm;
    }
    //////////////////////////////////////
void nop100(void)
{
    nop10();nop10();nop10();nop10();nop10();
    nop10();nop10();nop10();nop10();nop10();
}
    //////////////////////////////////////
// fai 100 nops por cada valor do parámetro
void retardo(unsigned int max)
{
    unsigned int r;
    for (r=0;r<max;r++)
    {
        nop10();nop10();nop10();nop10();nop10();
        nop10();nop10();nop10();nop10();nop10();
    }
}
    //////////////////////////////////////
// porto P1: pulsadores/leds
// (numeración dos leds ten que coincidir coas teclas)
// TECLAS 1-2-3-4 -> idc 7-5-3-1 -> P1.0-1-2-3
// LEDS 1-2-3-4 -> idc 8-6-4-2 -> P1.7-6-5-4
// porto P2: reles/optoacopladores
// (numeración dos relés ten que coincidir cos leds de esquerda a dereita
// coas fichas de saída cara arriba)
// RELES 1-2-3-4 -> idc 2-4-6-8 -> P2.4-5-6-7

// o sistema ten entradas mediante pulsadores e saídas mediante
// parellas led-relé (o led sinaliza o estado do relé asociado)
// usamos unha variable de estado para cada saída de xeito que
// se poida modificar o fluxo segundo a actuación dos pulsadores

void main(void)
{
    unsigned char estado1,estado2,estado3,estado4;
    bit tec1,tec2,tec3,tec4;
    unsigned int tempo1,tempo2;
    unsigned int sensor;

    //valores iniciais para os estados
    estado1=0; estado2=0; estado3=0; estado4=0;

    tempo1=0; tempo2=0;

    while(1)
    {
        //lectura teclas

        tec1=PORTO_TEC1;tec2=PORTO_TEC2;tec3=PORTO_TEC3;tec4=PORTO_TEC4;

        // lectura sensor analóxico (divisor con resistencia ntc + pot)
        sensor = ad_lectura(0);

        //TRATAMENTO DA SAIDA 1 (SEN USO)

        //TRATAMENTO DA SAIDA 1 (LÁMPADA 1)
        // esta saída funciona como un conmutador, a primeira pulsación
        // acende e a segunda apaga. O ciclo ten 4 estados:
        // 0: tecla off, led off -> 1: tecla on, led on ->
        // 2: tecla off, led on -> 3: tecla on, led on
        switch(estado1)

```

```

{
    case 0: //estado 0: tecla=1 (off), led=rele=1 (off)
        PORTO_LED1=1; PORTO_RELE1=1;
        if(tec1==0) estado1=1; //cambia estado
        break;
    case 1: //estado 1: tecla=0 (on), led=rele=0 (on)
        PORTO_LED1=0; PORTO_RELE1=0;
        if(tec1==1) estado1=2; //cambia estado
        break;
    case 2: //estado 2: tecla=1 (off), led=rele=0 (on)
        PORTO_LED1=0; PORTO_RELE1=0;
        if(tec1==0) estado1=3; //cambia estado
        break;
    case 3: //estado 3: tecla=0 (on), led=rele=1 (off)
        PORTO_LED1=1; PORTO_RELE1=1;
        if(tec1==1) estado1=0; //cambia estado
        break;
}

//TRATAMENTO DA SAIDA 2 (LÁMPADA 2)
// esta saida funciona como un automático de escaleira ou
// monoestable redispensible, a pulsación acende a lámpada
// e inicia unha temporización, cando remata apaga a luz
// o monestable é redispensible
// O ciclo ten 4 estados:
// tecla off, led off, temp=0 -> tecla on, led on, temp>0,
temp--
// -> tecla off, led on, temp>0, temp-- -> tecla on, temp=0
switch(estado2)
{
    case 0: //estado 0: tecla=1 (off), led=rele=1 (off)
        PORTO_LED2=1; PORTO_RELE2=1;
        tempol=0;
        //se tecla on cambia estado e inicia
temporizador
        if(tec2==0) { estado2=1; tempol=TEMPO1_ON; }
        break;
    case 1: //estado 1: tecla on, led=rele on, decrementa
tempo
        PORTO_LED2=0; PORTO_RELE2=0;
        // se tecla off cambia estado
        if(tec2==1) estado2=2;
        // se fin de tempo cambia estado
        tempol--; if(tempol==0) estado2=3;
        break;
    case 2: //estado 2: tecla off, led=rele on, decrementa
tempo
        PORTO_LED2=0; PORTO_RELE2=0;
        // se tecla on reinicia temporizador
        if(tec2==0) estado2=0;
        // se fin de tempo cambia estado
        tempol--; if(tempol==0) estado2=0;
        break;
    case 3: //estado 3: tecla on, led=rele off, espera tecla
off
        PORTO_LED2=1; PORTO_RELE2=1;
        // se tecla off cambia estado
        if(tec2==1) estado2=0;
        break;
}

//TRATAMENTO DA SAIDA 3 (LÁMPADAS 1 E 2 INTERMITENTES)
// Estados: 0=off, 1=intermitente on, 2=intermiente off

```

```

// o intermitente mantense mentres se pulsa a tecla
switch(estado3)
{
    case 0: //estado 0: tecla=1 (off), inactivo
        //se tecla on cambia estado e inicia
temporizador
        if(tec3==0) { estado3=1; tempo2=TEMPO2_ON; }
        break;
    case 1: //estado 1: led1=rele1 on, 2 off
        PORTO_LED1=0;PORTO_RELE1=0;PORTO_LED2=1;PORTO_RELE2=1;
        // se fin de tempo cambia estado
        tempo2--; if(tempo2==0) { estado3=2;
tempo2=TEMPO2_ON; }
        // tecla off, fin de intermitente
        if(tec3==1) estado3=3;
        break;
    case 2: //estado 1: led1=rele1 off, 2 on
        PORTO_LED1=1;PORTO_RELE1=1;PORTO_LED2=0;PORTO_RELE2=0;
        // se fin de tempo cambia estado
        tempo2--; if(tempo2==0) { estado3=1;
tempo2=TEMPO2_ON; }
        // tecla off, fin de intermitente
        if(tec3==1) estado3=3;
        break;
    case 3: //estado 3: leds=reles=1 (off)
        PORTO_LED1=1;PORTO_RELE1=1;PORTO_LED2=1;PORTO_RELE2=1;
        estado3=0; //vuelve a estado inactivo
        break;
}

// TRATAMENTO DA SAIDA 4 (CONTROL CALEFACCIÓN)
// esta aplicación compara o nivel de saída do convertedor ad
(ntc)
// cun valor de referencia, se o supera conecta a refrixeración
// estados: 0=saida inactiva, 1=saida activa
// poden definirse diferentes umbrais para activación e
desactivación
// (histérese)
switch(estado4)
{
    case 0: //estado 0: saída=1 (off), inactiva
        PORTO_LED3=1; PORTO_RELE3=1;
        //se tecla on cambia estado e inicia
temporizador
        if(sensor>UMBRAL_UP) estado4=1;
        break;
    case 1: //estado 1: saída=0 (on), activa
        PORTO_LED3=0; PORTO_RELE3=0;
        //se tecla on cambia estado e inicia
temporizador
        if(sensor<UMBRAL_DN) estado4=0;
        break;
}

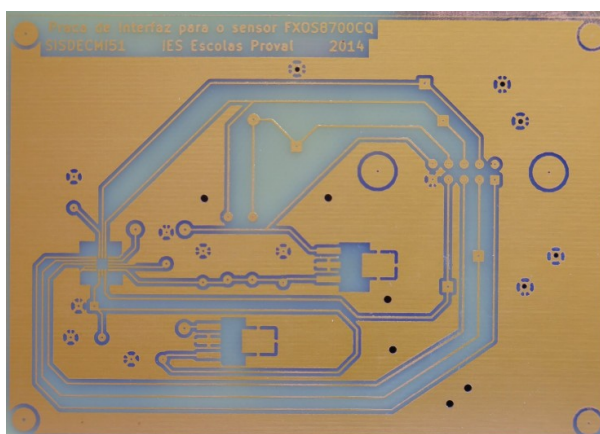
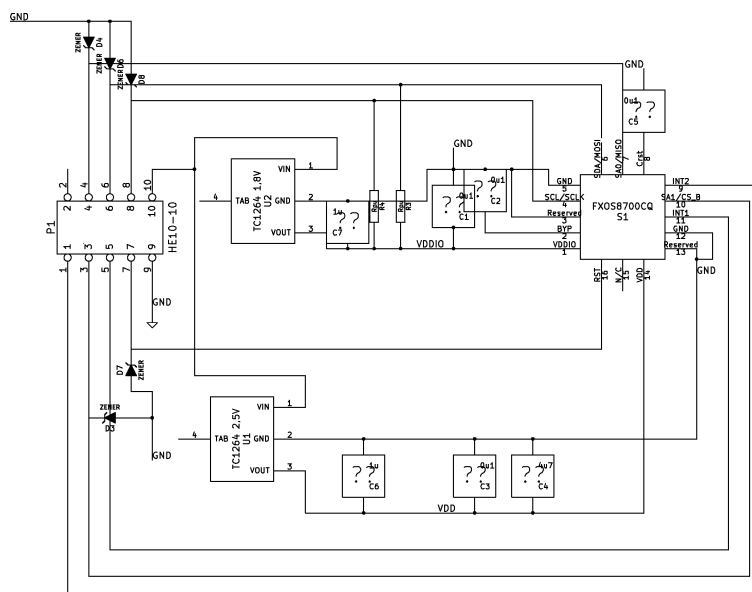
//retardo fixo en cada paso do bucle de control
retardo(2000);
}
}
////////////////////

```

2.2.5. Tarefa 5. Módulo con sensor acelerómetro dixital FXOS8700CQR1

Esta tarefa consiste na realización dun módulo de interfaz para un sensor acelerómetro combinado con magnetómetro. Como aplicación práctica está prevista a realización futura dun sistema de orientación de antenas parabólicas para uso docente. O sensor empregado é o modelo FXOS8700CQR1 de Freescale. O uso deste sensor presenta unha certa dificultade práctica polo seu reducido tamaño, polo que require o uso de material especializado (estación de soldadura smd, microscopio,...).

ESQUEMA E PLACA DE CIRCUÍTO IMPRESO



2.2.6. Tarefa 6. Módulo con sensor acelerómetro dixital LSM303

Esta tarefa consiste na realización doutro módulo de interfaz para un sensor acelerómetro combinado con magnetómetro (LSM303). O uso deste sensor presenta unha certa dificultade práctica polo seu reducido tamaño, polo que require o uso de material especializado (estación de soldadura smd, microscopio,...).

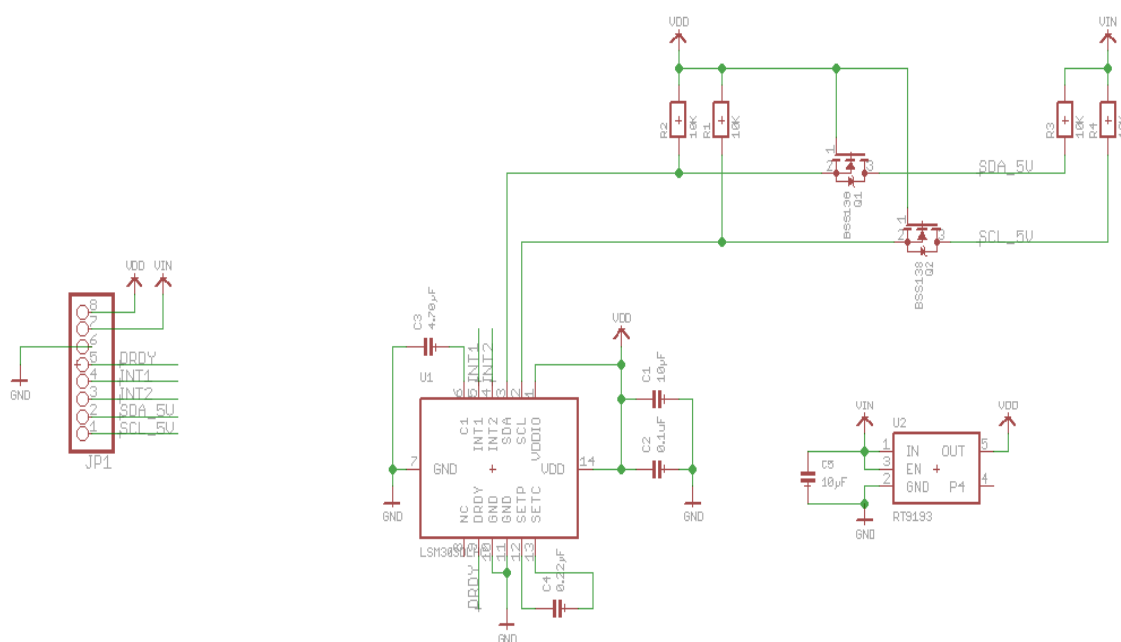
ESQUEMA E PLACA DE CIRCUÍTO IMPRESO

A información sobre este circuíto pode obterse nas páxinas:

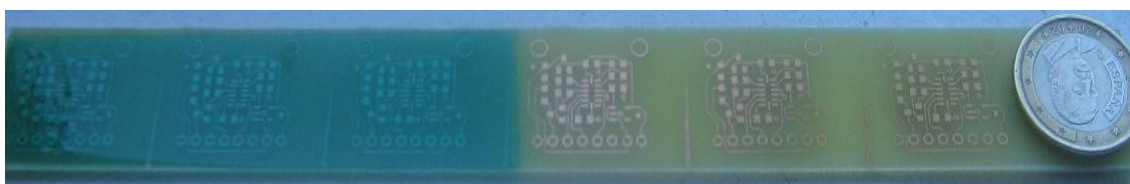
<https://github.com/adafruit/Adafruit-LSM303-PCB>

https://github.com/adafruit/Adafruit_LSM303DLHC

<http://www.st.com/web/en/resource/technical/document/datasheet/DM00027543.pdf>



Nesta imaxe pode verse a placa de circuíto impreso preparada para montaxe dos compoñentes:

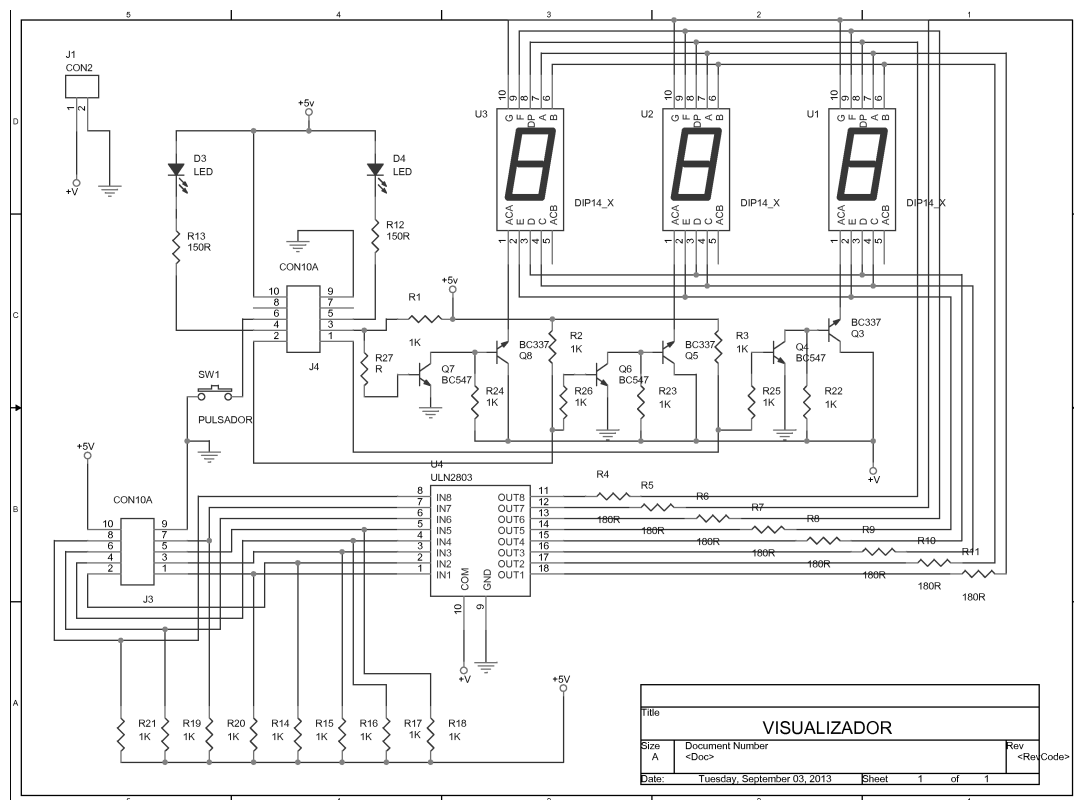


2.2.7. Tarefa 7. Medidor de velocidade con láser diodo e barreira óptica

Esta tarefa consiste na medición da velocidade dun obxecto polo tempo que permanece interrompida unha barreira luminosa atravesada polo obxecto. O resultado convértese a km/h e se presenta nun visualizador de 3 díxitos de gran tamaño, polo que pode servir para actividades deportivas. Como exemplo de aplicación se usa para a medida da velocidade dun balón de fútbol (“xutómetro”).

ESQUEMA ELECTRÓNICO

A continuación se representa o esquema electrónico do visualizador do medidor:



PROGRAMA DE EXEMPLO

```
; Usa os portos 1, 2 e 3
;P2.0 a P2.7 segmentos a....g,dp
;P3.2 Entrada de pulsos da barreira
;P1.5 Solicit. xute
;P1.4 Led de "preparado para xute"
;P1.3 Led de "record"
;P1.2 activa o díxito de cent
;P1.1 activa o díxito de dec
```

```

;P1.0 activa o díxito de unid
;Detecta o tempo de corte da barreira por un balón de 22cm
;Acende con uns nos segmentos
;acende un led en:
; P1.6 cando entra na rutina de desbordamento
; P3.0 cando vel - maior de 110Km/h
; P3.1 cando desbordou e recoñeceu que a entrada pasou a 1
;Modificado para activar os visualizadores mediante "0" en lugar de "1"
;nas bases dos transistores.
;Engadido un diodo led en P1.7 que se acenda cando recibe
;pulso da barreira e se apaga cando se solicita xute.
;Feito para unha barreira única que - o tempo que un balón de 22cm
;corta o feixe luminoso.
ORG 00
AJMP INICIO
ORG 1BH
AJMP DESBORD ;Int. T1 desbordamento por tiro lento
INICIO:
MOV R0,#0
MOV R1,#0
MOV R2,#0
MOV R3,#0
MOV R4,#0
MOV R5,#0
MOV R6,#0
MOV R7,#0
MOV P1,#0FFH
MOV P2,#0FFH
MOV P3,#0FFH
MOV 21H,#7EH
MOV 20H,#0FFH
MOV 25H,#0
MOV P1,#0FFH ;apagar leds e visualizadores
MOV P2,#00H ;apaga segmentos
MOV TMOD,#11H ;T0 e T1 en modo 1 como temp.
MOV IE,#08H ;Preparo para que admita interr.T1
;***** Bucle de espera a P1.5. *****
L1:
CLR P1.4 ;acender led "solicitud de tiro"
JNB P1.5,L2
ACALL VISUAL
AJMP L1

```

```

L2: SETB P1.7 ;apaga led indicador de paso pola barreira
SETB P1.4
SETB P1.3 ;apaga leds de "record" e "solicitude de tiro"
;@@@@@@@@
SETB P1.6
SETB P3.0
SETB P3.1
;@@@@@@@@
MOV TH1,#85H ;PARA QUE DESBORDE AOS 31.5 ms
MOV TL1,#0
MOV TH0,#00
MOV TL0,#00 ;conta o máximo = 65ms
L3A: JB P3.2,L3A ;detecta un nivel baixo en P3.2
SETB TR0 ;acéndese o temporizador
SETB TR1
SETB EA ;habilita interr de T1
CLR P1.7 ;acende led R que detecta paso pola barreira
aaaaaaaaaaaaaaaaaaaaa
;aaaaaaaaaaaaaaaaaaaaa
MOV R2,# 3CH ;retardo de aprox 100us antirrebotes
L99: DJNZ R2,L99
;aaaaaaaaaaaaaaaaaaaaa
;aaaaaaaaaaaaaaaaaaaaa
L20: JNB P3.2,L20 ;espera a que DEIXE DE PASAR O BALON
CLR EA ;deshabilitanse as interrupci ns
CLR TR0 ;apágase T0
CLR TR1
CLR TF0
CLR TF1
MOV R1,TH0 ;gárdanse TH0 e TL0 en R1 e R0
MOV R0,TL0
JNB 25H.0,L9
MOV R3,#00H
MOV R4,#00H
MOV R5,#00H
CLR 25H.0
;@@@@@@@@
CLR P3.1
;@@@@@@@@@@@@@@
AJMP L1
;***** determinaci n se   o record ou non
L9: MOV A,R1

```

```

CLR C
SUBB A,#1CH ;se a velocidade é maior de 110Km/h pon a 0
JNC L9A
MOV R3,#00H
MOV R4,#00H
MOV R5,#00H
;oooooooooooo
CLR P3.0
;oooooooooooooooo
AJMP L1
L9A: CLR C
MOV A,R1
SUBB A,21H ;comparo co gardado en 21h e 20h
JNZ L4
CLR C
MOV A,R0
SUBB A,20H
JNZ L5
SJMP L6
L4: JNC L6
L7: CLR P1.3 ;acender led de record
MOV 20H,R0 ;se é record gárdase en 20h e 21h
MOV 21H,R1
SJMP L6
L5: JNC L6
SJMP L7
;***** Redondeo de R1 (valor de TH0)
L6: CLR C
MOV A,R0
SUBB A,#80H ;se R0 é maior de 79h increméntase R1
JC L8
INC R1
L8: CLR C
;***** Conversión do valor redondeado de TH0 (R1) a velocidade
;faise mediante unha consulta á taboa de velocidades.
MOV DPTR,#TABVEL
MOV A,R1
SUBB A,#1CH ;porque o valor mínimo de TH0 é 1Ch
CLR C
MOVC A,@A+DPTR
MOV B,#64H ;divido por 100 para ter o 1º dígito
DIV AB

```

```

MOV R5,A ;gárdase en R5 o díxito máis alto
MOV A,B
MOV B,#0AH ;divido o resto anterior por 10 para 2º dix.
DIV AB
MOV R4,A ;gárdase en R4 o díxito do medio
MOV R3,B ;gárdase en R3 o díxito máis baixo
;***** Visualización dos tres díxitos
ACALL VISUAL
LJMP L1
;***** Descodificador dos tres díxitos
;faise consultando unha taboa de xeito que os segmentos
;acéndense con "1"
VISUAL:
MOV DPTR,#DESCO
CLR C
MOV A,R3
SUBB A,#5H
JZ L30
SJMP L31
L30: CLR C
MOV A,R4
SUBB A,#5H
JZ L32
SJMP L31
L32: CLR C
MOV A,R5
SUBB A,#2H
JZ L33
SJMP L31
L33: CLR C
MOV R3,#0
MOV R4,#0
MOV R5,#0 ;ata aquí
L31: MOV A,R3
MOVC A,@A+DPTR
MOV P2,A
CLR P1.0 ;acéndese o díxito de menor peso
SETB P1.1
SETB P1.2
ACALL TEMPO ;modificado o tempo da subrutina de 80ms
;a 5ms para que non flutúe
MOV A,R4

```

```

MOVC A,@A+DPTR
SETB P1.0
MOV P2,A
CLR P1.1 ;acéndese o díxito do medio
SETB P1.2
ACALL TEMPO
MOV A,R5
MOVC A,@A+DPTR
SETB P1.1
MOV P2,A
CLR P1.2 ;acéndese o díxito de menor peso
SETB P1.0
ACALL TEMPO
SETB P1.2 ;apágase o 3º díxito e quedan todos así
RET
;***** subrutina de tempo de espera de 5ms aprox
TEMPO:
MOV R7,#10H
L10: MOV R6,#9CH
L11: DJNZ R6,L11
DJNZ R7,L10
RET
;***** interrupción por desbordamento de T1
;no caso de tiro moi lento deixa o contador a 0
DESBORD:
SETB 25H.0
CLR TR1
CLR TF1
CLR IE1
CLR EA ;deshabilita as interrupcións
;@@@@@@@@@@@@
CLR P1.6
;@@@@@@@@@@@@
RETI
;***** táboas de datos
DESCO:
DB 03FH, 06H,5BH,4FH,66H,6DH
DB 7DH,7H,7FH,67H
TABVEL:
DB 6EH,6BH,67H,64H,61H,5EH
DB 5BH,58H,56H,54H,51H,4FH
DB 4DH,4BH,4AH,48H,46H,45H

```

DB 43H,42H,40H,3FH,3EH,3DH
DB 3BH,3AH,39H,38H,37H,36H
DB 35H,34H,34H,33H,32H,31H
DB 30H,30H,2FH,2EH,2DH,2DH
DB 2CH,2CH,2BH,2AH,2AH,29H
DB 29H,28H,28H,27H,27H,26H
DB 26H,25H,25H,24H,24H,24H
DB 23H,23H,22H,22H,22H,21H
DB 21H,21H,20H,20H,20H,1FH
DB 1FH,1FH,1EH,1EH,1EH,1DH
DB 1DH,1DH,1DH,1CH,1CH,1CH
DB 1CH,1BH,1BH,1BH,1BH,1AH
DB 1AH,1AH,1AH,1AH,19H,19H
DB 19H,19H,19H
END

2.2.8. Tarefa 8. Sensores e actuadores en instalacións domóticas

Esta tarefa consiste na realización dunha práctica de conexión de sensores e actuadores en instalacións domóticas.

BOLETÍN DE PRÁCTICAS

Instalacións domóticas

CONEXIÓN DE SENSORES E ACTUADORES

Obxectivo

Durante a realización desta práctica o alumno debe comprender e utilizar os seguintes conceptos:

1. Sensores para domótica
2. Actuadores para domótica
3. Tipos de entradas dos módulos domóticos
4. Tipos de saídas dos módulos domóticos

Material

O alumno deberá utilizar para a realización desta práctica o seguinte material:

Sensores para domótica: detector de fume, sensor de calor, sensor de inundación, termóstato de roda con saída por relé, detector de movemento, detector crepuscular, contacto magnético para detectar apertura de portas ou ventás, detector de gas, pulsador convencional, pulsador inversor para persianas

Actuadores para domótica: avisador acústico-luminoso, electroválvula de auga, contactor, enchufe, lámpada, pechadura eléctrica, relé, motor de persiana eléctrica

Módulos domóticos con 3 tipos distintos de entradas dixitais:

-Winkel wk0400 (ten entradas dixitais libres de potencial)

-Winkel wk0100 (ten entradas dixitais de continua tipo pnp)

-Autómata B&B Electronics TECO SG2-10HR-A (ten entradas dixitais de alterna)

Para calquera destes 3 módulos o alumno usará un programa tipo flip-flop (un flanco na entrada dixital i modifica o estado da saída i)

Tamén usaremos un módulo domótico específico para controlar o motor dunha persiana cun programa para subir e baixar unha persiana usando pulsadores convencionais:

-Winkel wk0300 (ten 2 entradas dixitais libres de potencial e 2 saídas por relé conectadas a 230Vac para conectar directamente ao motor dunha persiana)

Bibliografía

http://www.bb-elec.com/Products/Manuals/SG2V3_UserManual.pdf

http://winkelhel.com/downloads/Ficha_tecnica_WK0100.pdf

http://winkelhel.com/downloads/Ficha_tecnica_WK0300CR.pdf

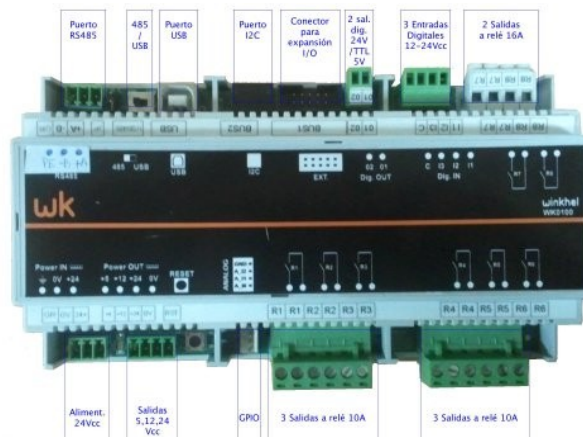
http://winkelhel.com/downloads/Ficha_tecnica_WK0402.pdf

Enunciados

1. Activación dun avisador acústico-luminoso mediante detector de fume (Usamos I1 dixital e saída R1 dixital) Entrada I1 libre de potencial



2. Activación dun avisador acústico-luminoso mediante un sensor de calor (Usamos I2 dixital e saída I2 dixital) Entrada I2 de continua tipo pnp



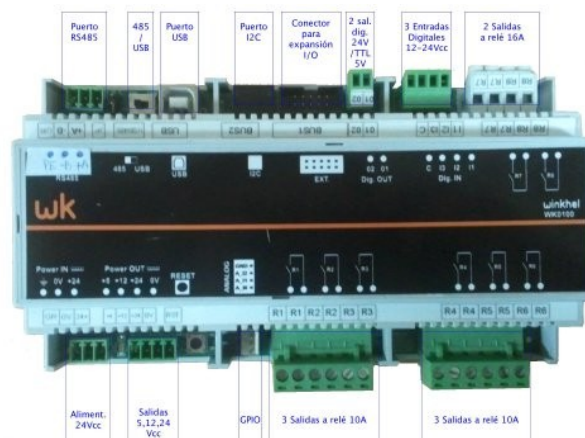
3. Control subministro de auga mediante electroválvula cando se detecte unha inundación (Usamos entrada dixital I1 e saída dixital Q1) Entrada I1 de alterna. **IMPORTANTE: comprobar se o relé do detector soporta una tensión de 230Vac e en caso contrario intercalar un relé externo que si a soporte**



4. Control dun enchufe para calefacción eléctrica con contactor e termóstato de roda con saída por relé (Usamos entrada dixital I3 e saída dixital R3) Entrada I3 libre de potencial



5. Activación dun punto de luz cando se detecte movemento e sexa case de noite (Usamos entrada dixital I2 para conectar os 2 detectores e saída dixital I2 para a lámpada) Entrada I2 de continua tipo pnp



6. Apertura duna pechadura eléctrica cando se detecte movemento (Usamos entrada dixital I3 para conectar o detector e saída dixital R3 para a pechadura) Entrada I3 de alterna. **IMPORTANTE: comprobar se o relé de**

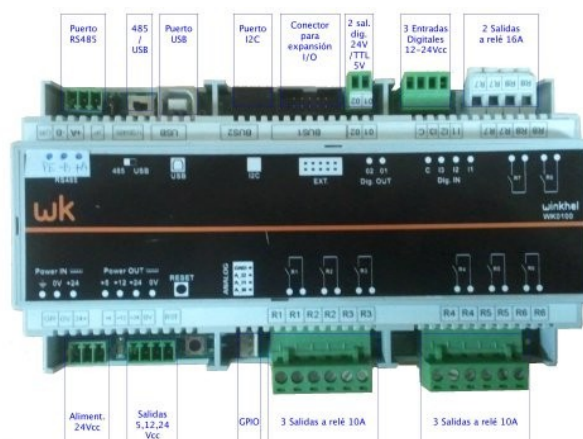
detector soporta 230Vac e en caso contrario intercalar un relé externo que si a soporte



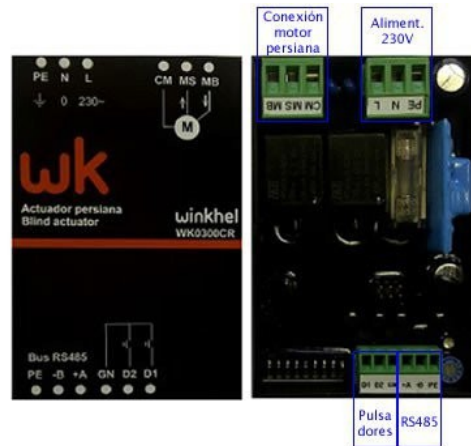
7. Activación de avisador acústico-luminoso cando se detecte apertura de calquera de 3 ventás dunha vivenda (Usamos entrada dixital I1 para conectar os 3 sensores magnéticos e saída dixital I1 para o avisador) Entrada I1 libre de potencial



8. Activación dunha lámpada cando se detecte gas (Usamos entrada dixital I2 e saída dixital R2) Entrada I2 de continua tipo pnp



Antes de controlar o motor cun módulo domótico débese realizar o control do motor unicamente utilizando un pulsador inversor e debuxar o esquema:



3. Avaliación

A avaliación desta actividade se realizará a través das probas de funcionamento dos módulos realizados, a verificación dos programas elaborados e probas escritas. Como exemplo, axúntase a continuación un modelo de proba escrita:

PROBA ESCRITA – CONVERTEDORES AD/DA-SENSORES-COMUNICACIÓNS

Nome:DATA:/...../.....

- 1) O teorema de Nyquist-Shannon establece que a frecuencia mínima de mostraxe dun sistema dixital ten que ser o dobre da máxima frecuencia de entrada ($f_s > 2f_{max}$). Responde ás seguintes cuestións:
 - a) Cal será a mínima frecuencia de funcionamento dunha tarxeta de son de ordenador se a frecuencia máxima de audio é de 20KHz?
 - b) Se nun CD se emprega como frecuencia de dixitalización 44,1KHz cúmprese o teorema?
- 2) Explica brevemente por qué é necesaria a conversión dixital a analóxica e analóxica a dixital. Pon un exemplo de uso.
- 3) Indica os tipos que recordes de convertedores analóxico-dixitais
- 4) Explica en qué consiste un convertedor analóxico a dixital de aproximacións sucesivas
- 5) Explica en qué consiste un convertedor dixital a analóxico de rede de resistencias R-2R
- 6) Qué é a resolución dun convertedor dixital a analóxico? ¿Qué relación ten co número de bits?
- 7) En qué consiste o erro de offset dun convertedor D/A?
- 8) Podemos empregar un convertedor A/D de tipo serie nun circuito de elevada velocidade? Por qué?
- 9) Qué tipo de convertedor A/D é o máis empregado en polímetros dixitais?

- 10) Explica por qué en circuitos de audio se empregan convertedores A/D cun número de bits elevado (16 ou máis).
- 11) Indica os tipos de sensores que recordes según a magnitude física que miden
- 12) Queremos construír unha estación meteorolóxica con sensores I2C de temperatura, humidade e presión atmosférica conectados a un microcontrolador. É necesario que todos os sensores estean situados no mesmo lugar? Fai un esquema de bloques do sistema indicando o cableado necesario.
- 13) Fai un esquema para conectar unha resistencia LDR a un convertedor analóxico a dixital
- 14) Compara os buses I2C e SPI indicando as vantaxes e inconvenientes de cada un.
- 15) O relé tipo reed é un elemento que se usa frecuentemente como sensor magnético. Cómo usarías un relé reed combinado cun imán para medir a velocidade nunha roda de bicicleta? Fai un esquema de bloques.
- 16) Os sensores piroeléctricos
- a) miden o calor emitido no seu contorno (radiación infravermella lonxana)
 - b) miden a temperatura
 - c) miden o movemento
- 17) Cita aplicacións dos sensores de forza e aceleración como o MMA7660.
- 18) Os sensores ultrasónicos como o Devantech SRF02...
- a) pode usarse como detectores de distancia
 - b) poden usarse como detectores de presenza
 - c) poden usarse como detectores de velocidade
- 19) Cita tipos de sensores de temperatura indicando se son analóxicos ou dixitais.
- 20) Un microcontrolador como o AT89S4051 leva integrado un porto serie RS232. É posible conectalo a unha rede ethernet, wifi, ou bluetooth? E a un ordenador con USB? Qué tipo de elementos serían necesarios?