

SISTEMA DE DESENVOLUPAMENT DS552

INTRODUCCIÓ

L'electrònica digital programable ha evolucionat molt ràpidament en els últims anys. L'aparició del microcontrolador (μ C) que engloba en un únic xip la unitat central de procés, la memòria de dades i programa i les unitats d'entrada i sortida, ha permès que els nous dissenys de circuits electrònics es construeixin d'una manera molt més senzilla i fiable.

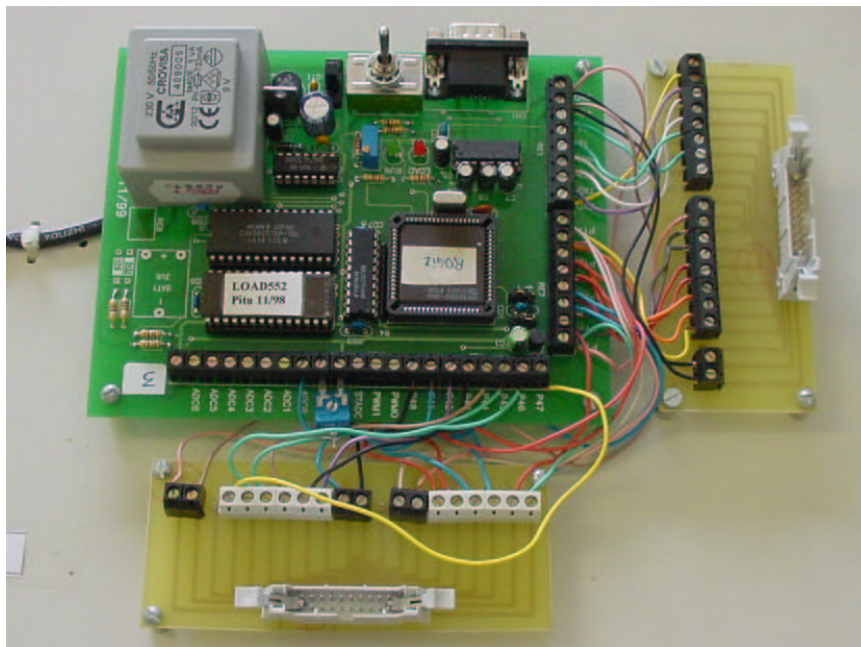
El resultat final serà una placa d'unes dimensions més reduïdes ja que el μ C permet substituir desenes de circuits. Al haver-hi menys components vol dir també una major facilitat a l'hora de fer el circuit imprès.

Per fer qualsevol aplicació on sigui necessari l'ús d'un microcontrolador, ens trobem amb les següents fases:

- Escriure el programa font
- Assemblar / Compilar
- Gravar el microcontrolador externament amb un programador
- Comprovar si funciona correctament

El μ C sol portar una memòria del tipus EPROM. Això vol dir que si el programa no funciona correctament haurèm d'esborrar externament aquesta memòria i tornar a començar el procés. El desenvolupament d'un programa es pot convertir en una feina molt lenta i feixuga.

El DS552 És un sistema de desenvolupament basat amb el 87C552 i que permet carregar un programa (de la família MCS51) via sèrie des del PC i poder-lo executar tantes vegades com sigui necessari.



CARACTERÍSTIQUES

El μ C que fem servir és el 80C552 i el fem treballar com a μ processador. Això vol dir que els ports P0 i P2 no els podrem utilitzar ja que actuen com a bus de dades i d'adreces. També utilitza les línies WR i RD (P3.6 i P3.7) per la memòria RAM. Tot i així ens queden prou línies d'entrada sortida per la majoria d'aplicacions.

Les principals característiques del nostre sistema de desenvolupament DS552 són:

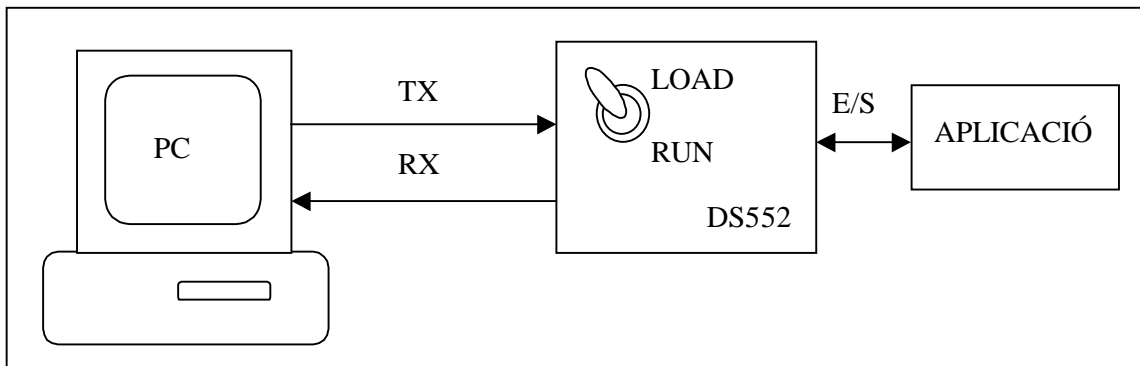
- Unitat Central de Procés 80c552. Sense memòria interna.
- 32K de memòria EPROM externa.
- 32K de memòria RAM alimentada permanentment amb pila.
- 2 Timers/Counters de 16 bits.
- 4 registres de captura i 3 registres de comparació associats a un tercer Timer de 16 bits.
- Convertidor Analògic Digital de 10 bits amb 7 entrades multiplexades.
- 2 Moduladors PWM de 8 bits de resolució.
- 1 Port sèrie UART Full-Duplex.
- 1 Port sèrie Bus I2C.
- Watchdog que es pot habilitar a través d'un jumper.
- Funció IDLE
- 3 Ports d'entrada/sortida més 1 d'entrada, lliures per l'usuari amb regleta de connexió.
- Font d'alimentació de 5v incorporada.
- Commutador LOAD/RUN per carregar/executar el programa
- Connector DB9 per la connexió amb el PC.

La memòria EPROM, porta un programa monitor que s'executa en el moment de donar l'alimentació. Si el commutador està a la posició LOAD, aleshores el programa està esperant que pel port sèrie, que el tenim connectat al PC, l'hi arribi un programa. Aquest el va copiant a la memòria RAM. Quan l'operació de càrrega ha finalitzat, el programa monitor l'hi passa el control i s'executa.

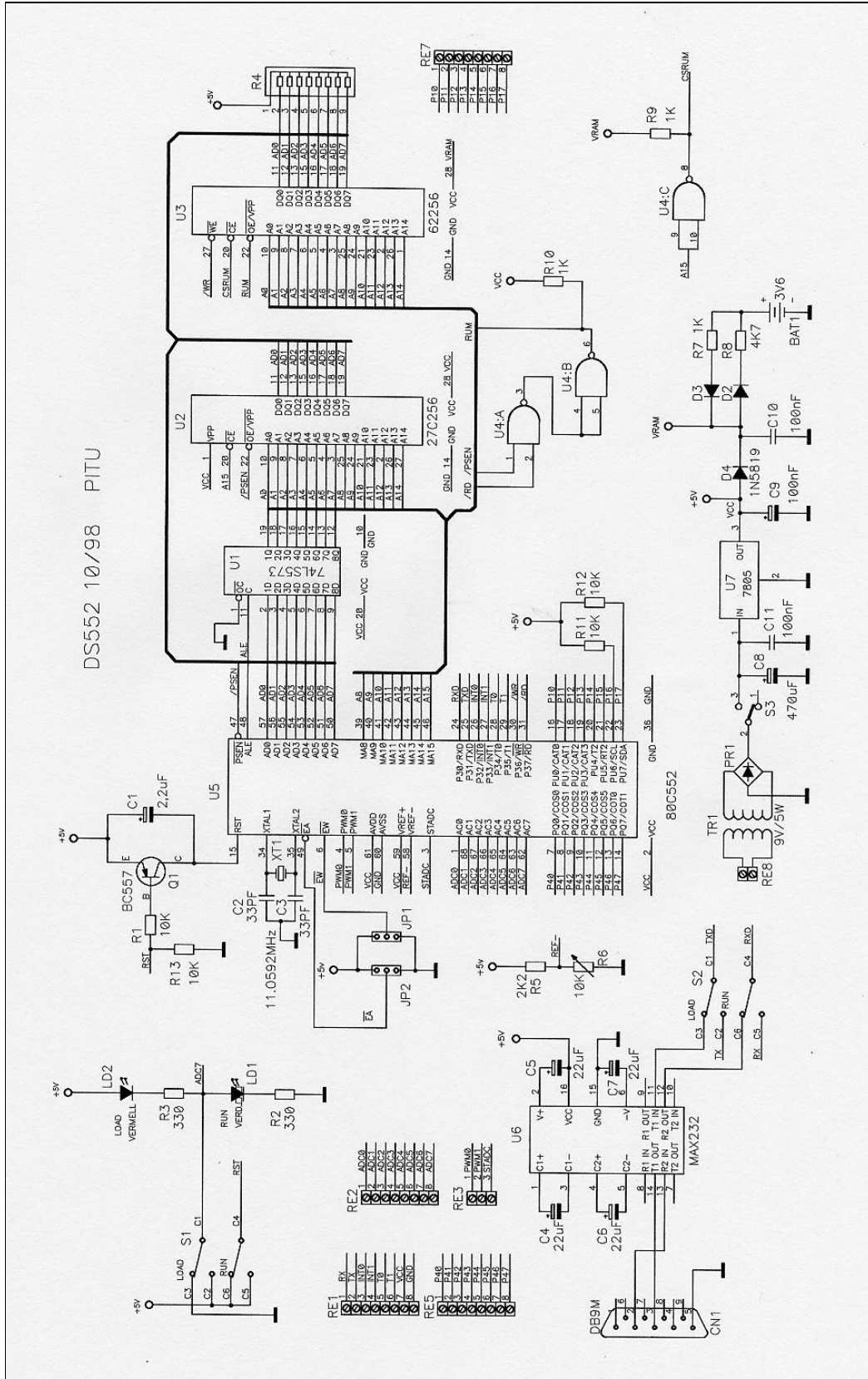
Si veiem que el nostre programa s'encalla podem fer-li un RESET i executar-lo de nou movent el commutador a la posició RUN.

Si el que passa és que s'ha de modificar, aleshores el podem tornar a carregar via sèrie com abans i tornar a comprovar si funciona.

Realment hem aconseguit una eina que permet desenvolupar qualsevol programa amb gran rapidesa i comoditat.



DS552 10/98 PITU



ESQUEMA

Podem distingir quatre parts clarament diferenciades:

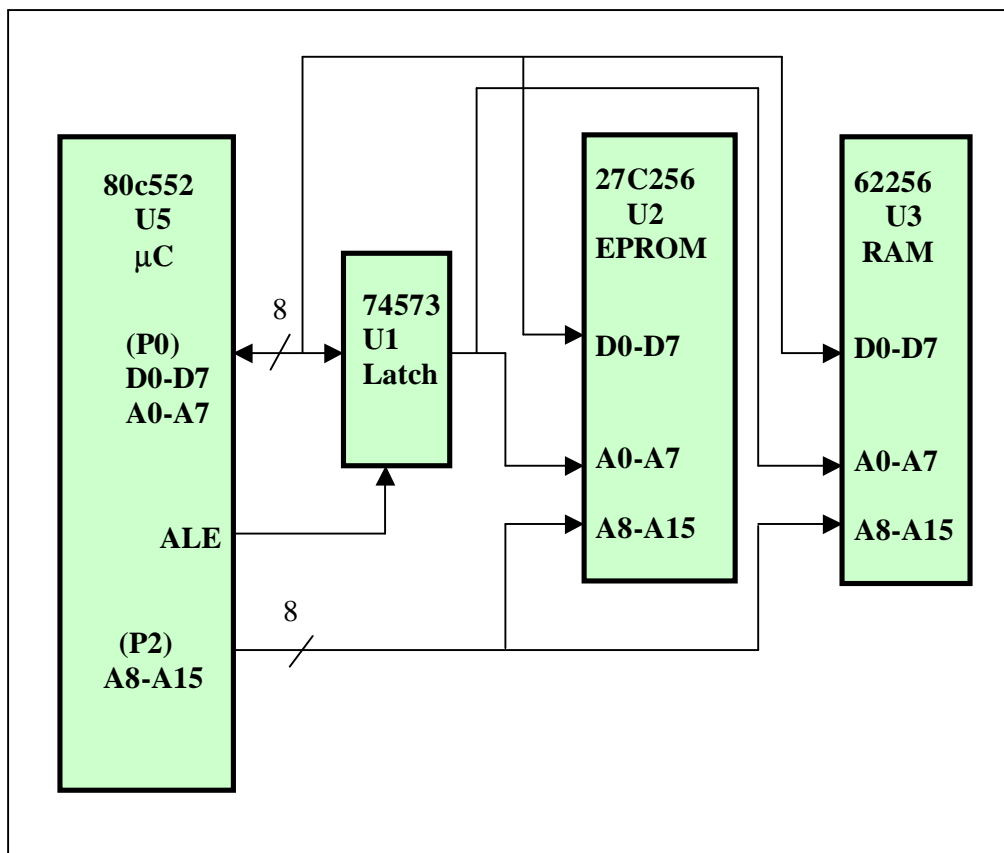
- El μ Controlador i les memòries.
- El circuit de Reset.
- La comunicació amb el PC.
- La Font d'alimentació.

μ Controlador i les memòries.

De fet, el 80c552 (U5) el fem treballar com un microprocessador. La memòria de programa és externa i per tant necessitem el Bus de dades P0 i el Bus d'adreces P0 i P2.

La memòria de programa és una EPROM (U2) de 32K i la de dades la RAM 62256 (U3) també de 32K.

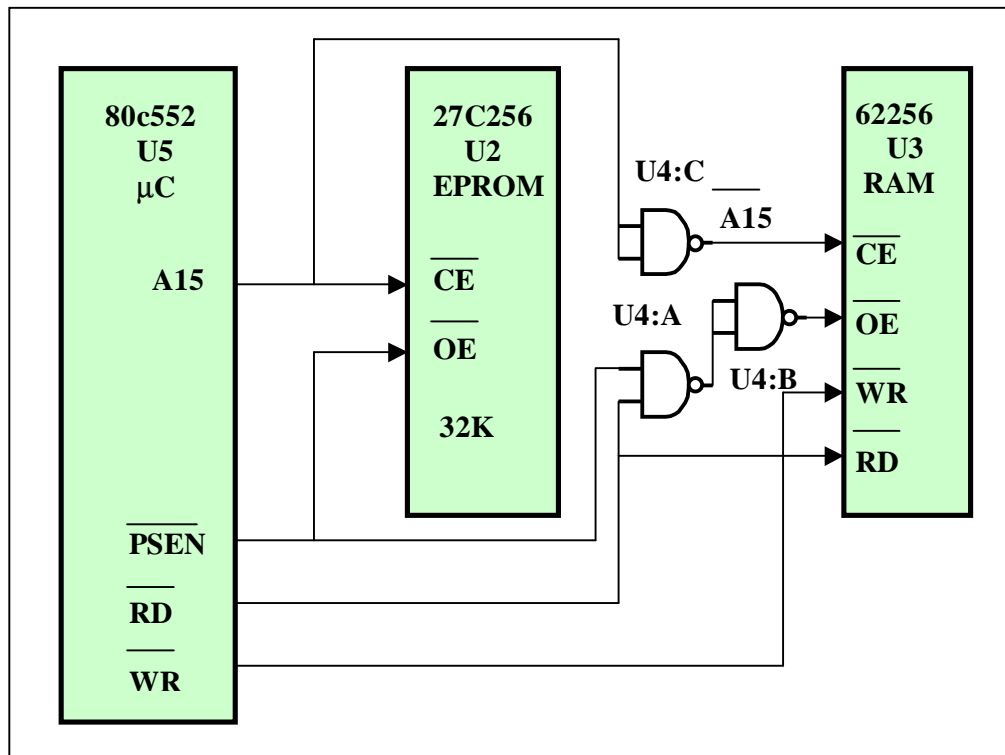
Com que P0 de 8 bits és el bus de dades i també la part baixa del bus d'adreces, necessitem el registre de dades o Latch 74LS573 (U1). Així quan el μ Controlador vol fer una operació amb la memòria, primer posarà l'adreça a P0 (part baixa) i a P2 (part alta). A continuació activa el senyal **ALE** per guardar la part baixa al registre de dades (Latch). Per últim la operació de lectura o escriptura amb la memòria ja es pot realitzar a l'adreça seleccionada i P0 actua ara com a Bus de dades.



Adreçant les memòries

Memòria de programa.

Quan el μ Controlador vol llegir una instrucció, traurà l'adreça que l'hi indica el registre Comptador de Programa (PC) i a continuació activa el senyal $\overline{\text{PSEN}}$. Aquest senyal haurà d'habilitar la memòria de programa (U2) i pel bus de dades el micro recollirà la instrucció. Efectivament, podem veure que $\overline{\text{PSEN}}$ activa el senyal d'habilitació $\overline{\text{OE}}$ de la memòria EPROM (U2).



Control de les memòries

La selecció de xip $\overline{\text{CE}}$ la realitza A15 que és el bit més alt del bus d'adreces. Per tant la memòria quedarà habilitada en l'espai d'adreces entre:

<u>Binari</u>	<u>Hex</u>	<u>Decimal</u>
0111 1111 1111 1111	7FFF	32K
0000 0000 0000 0000	0000	0

Podem veure també que el senyal $\overline{\text{PSEN}}$ es multiplica amb $\overline{\text{RD}}$ en una porta NAND (U4) i el resultat s'inverteix. Es equivalent a la suma $\overline{\text{PSEN}} + \overline{\text{RD}}$. Aquesta suma (RUM) arriba a l'entrada d'habilitació $\overline{\text{OE}}$ de la memòria RAM. Això vol dir que en aquesta memòria també podrem llegir instruccions de programa.

La selecció de xip \overline{CE} la realitza també A15 però ara invertit per U4:C. Aleshores l'espai d'adreces de RAM serà:

<u>Binari</u>	<u>Hex</u>	<u>Decimal</u>
1111 1111 1111 1111	FFFF	64K
1000 0000 0000 0000	8000	32K

En resum. Disposem un total de 64K de memòria externa de programa repartits en dues memòries de 32K cada una.

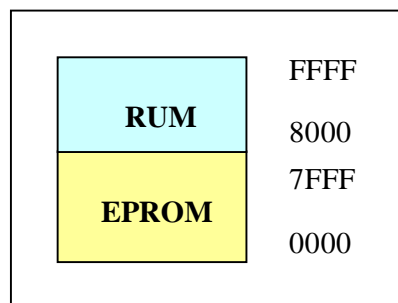
Memòria de dades.

Quan el μ Controlador vol fer una operació amb la memòria de dades, primer treu l'adreça i a continuació activa el senyal \overline{WR} per escriure o el senyal \overline{RD} si vol llegir.

El senyal \overline{WR} arriba a la memòria RAM i per tant únicament aquí el micro podrà escriure el byte que circula pel Bus de dades a l'adreça seleccionada. Recordem que la RAM només està habilitada de la 8000 fins a la FFFF i per tant no serà possible escriure a cap altra adreça.

El senyal \overline{RD} hem vist que es combina amb \overline{PSEN} i permet habilitar la RAM. Aleshores el byte a llegir de l'adreça seleccionada circularà pel bus de dades i el micro el podrà llegir. Igualment l'espai d'adreces serà de la 8000 a la FFFF.

Com que la memòria de programa i la de dades està combinada, El programador és qui decideix l'amplada que tindrà cada una. Els 32K de la EPROM són exclusius per programa i els 32K de RAM tant poden ser de programa com de dades.



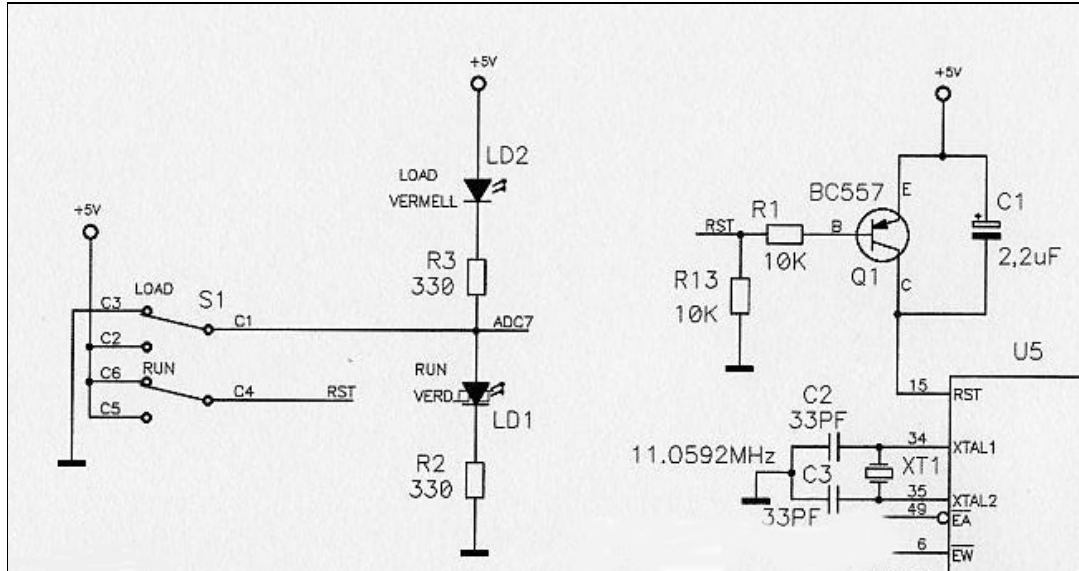
Mapa de memòria

El circuit de RESET.

Per generar un reset, és suficient aplicar un 1 lògic a l'entrada RST durant un dos cicles de màquina que corresponen a 12 impulsos de rellotge. D'això s'encarrega el condensador C1. Quan es connecta l'alimentació C1 està descarregat i aplica un 1 durant el temps que tarda a carregar-se a través d'una resistència que és interna al μ C. Llavors comença a executar-se el programa a partir de la posició 0 de memòria que és on tenim el programa monitor.

El transistor Q1 permet fer un reset amb l'ajut del commutador S1. Quan aquest commutador passa de la posició LOAD a RUN o al revés, T1 es posa a conduir creuant el condensador i per

tant provocant un RESET. Fixeu-vos que T1 només condueix en la transició de les dues posicions del commutador.



Circuit de RESET

El commutador S1 té dues posicions:

- **LOAD:** El Led vermell està encès i s'aplica un zero a l'entrada ADC7. El programa monitor el primer que fa és llegir el valor d'ADC7. Si està a zero llegirà el port sèrie i anirà carregant a la RAM (0x8000) el programa que arriba des del PC. Quan ja està carregat aleshores s'executa i TXD i RXD estan connectats al PC.
- **RUN:** El led verd està encès i s'aplica un 1 a l'entrada ADC7. El programa monitor fa un salt a la posició de memòria 0x8000 que és on hi ha el nostre programa i s'executa. En aquesta posició estem desconnectats del PC i disposem de TXD i RXD per la nostra aplicació.

Comunicació amb el PC

La comunicació amb el PC la necessitem per poder carregar el nostre programa des del PC a la memòria RAM i també per fer servir el teclat i la pantalla del PC com entrada i sortida de dades dels nostres programes.

Tenim un problema de compatibilitats de nivells lògics entre els dos dispositius.

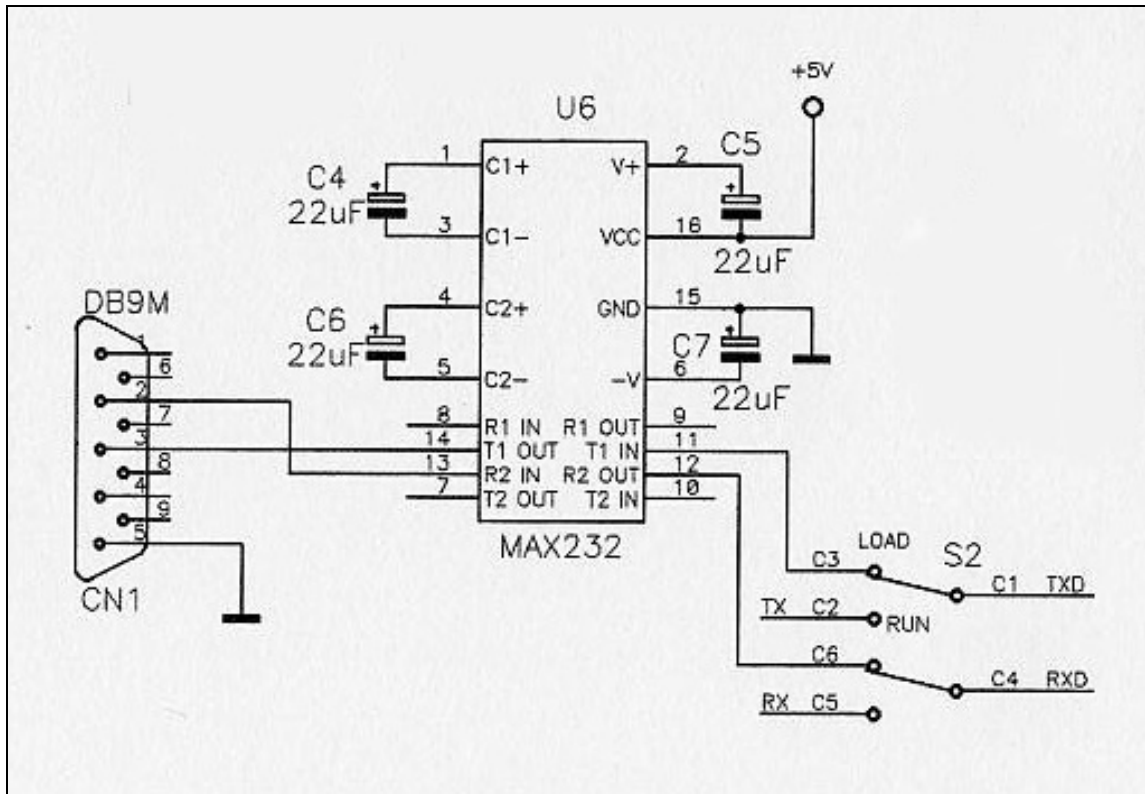
Nivell lògic	PC	µControlador
1	-12 v	+5 v
0	+12 v	0 v

És evident que el PC i el µC no es poden entendre. El MAX232 (U6) és un circuit integrat que adapta aquests nivells i permet que el µControlador i el PC es puguin comunicar sense cap problema.

Únicament farem servir les línies TXD i RXD del port sèrie del PC amb lo qual no permetrà fer cap tipus de protocol.

Quan el commutador S2 (és el mateix S1), està a la posició LOAD, la línia TXD del PC (pin 3 del connector DB9. CN1) es connecta amb TXD del µControlador i RXD del PC (pin 2 del connector DB9. CN1) es connecta amb RXD del µControlador.

Quan el commutador està a la posició RUN, aleshores les línies RXD i TXD del µControlador queden desconnectades del PC i estan disponibles a la regleta RE1 (TX i RX). Això ens permetrà carregar el programa del PC (LOAD) i llavors fer una comunicació sèrie amb un altra dispositiu quan executem el programa a la posició RUN.

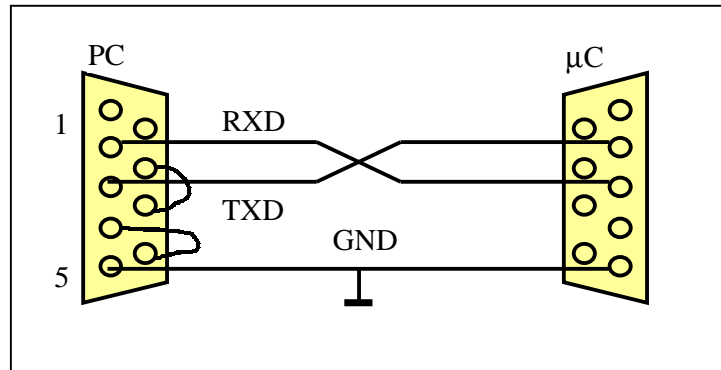


Comunicació amb el PC

El connector DB9 del PC té assignat aquests senyals:

Pin	Nom	Descripció
1	CD – Carrier Detect	Connexió establerta
2	RXD - Received Data	Canal de Recepció
3	TXD - Transmitted Data	Canal de Transmissió
4	DTR - Data Terminal Ready	Terminal preparat
5	GND	GND
6	DSR – Data Set Ready	Equip Preparat
7	RTS – Request To Send	Petició per enviar
8	CTS – Clear To Send	Preparat per enviar
9	RI – Ring Indicator	Indicador de trucada

Com que només fem servir tres fils (RXD, TXD i GND), és necessari fer uns ponts al connector per així enganyar al programa de comunicacions i que es pensi que la comunicació està sempre preparada:

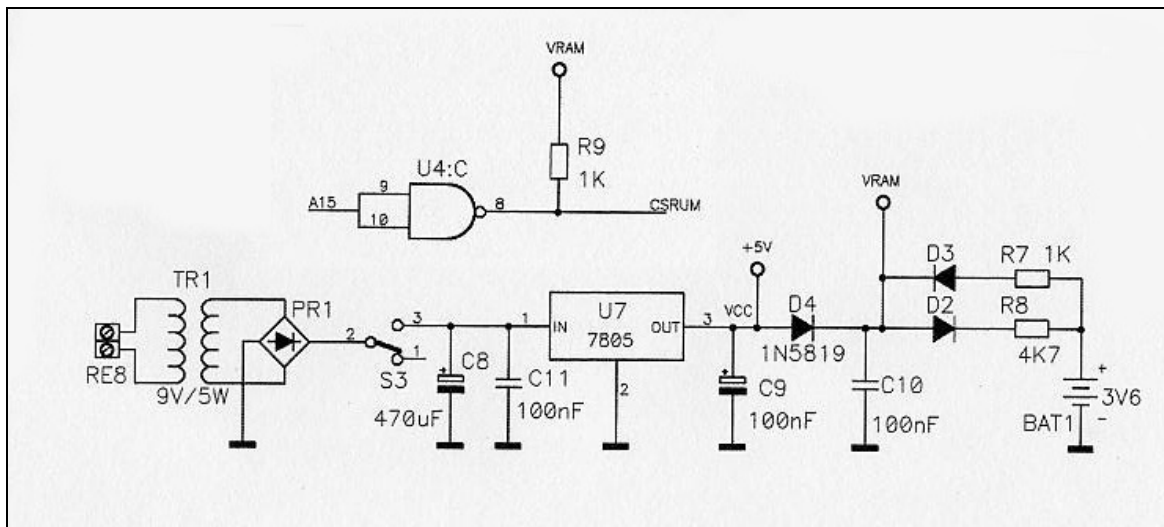


Connexió Modem Null

Els ponts s'han de fer als pins 4 i 6 i també als pins 7 i 8 del costat del PC. Fixem-nos que la línia RXD (2) del PC va a la TXD (3) del µC i al revés. És el que es coneix com una connexió en Modem Null.

Font d'alimentació.

Partim d'un transformador de 9v/5w. Això ens permet una intensitat màxima de 550mA. El regulador és un 7805 que proporciona una tensió de 5v a la línia Vcc.



Font d'alimentació

El commutador S3 serveix per connectar la font.

La bateria BAT1 de 3,6 v, ens permetrà alimentar la memòria RAM quan talem l'alimentació del circuit i així no perdrem el programa. Mentre tenim alimentació, D4 pot conduir i la bateria es va carregant a través de D2 i R8. A VRAM l'hi arriben 4,5V, suficients per alimentar la memòria RAM.

Quan es treu l'alimentació del circuit, D4 deixa de conduir i la tensió de la bateria arriba a VRAM a través de R7 i D3. VRAM únicament alimenta la memòria RAM i el CE que hi aplica un 1 a través de CSRAM per tal d'inhabilitar-la quan hem desconnectat l'alimentació.

D'aquesta manera podem disposar d'una memòria RAM no volàtil.

Llistat de components.

QUAN T.	ELEMENT	REFERÈNCIA	DESCRIPCCIÓ
1	74LS573	U1	LATCH
1	27C256	U2	EPROM 32K
1	62256	U3	RAM 32K
1	74LS03	U4	NAND OPEN COLECTOR
1	80C552	U5	Microcontrolador
1	MAX232	U6	Adaptador nivells RS232
1	7805	U7	Regulador de tensió
1	BC557	Q1	Transistor PNP
1	LED	LD1	LED verd 3mm.
1	LED	LD2	LED vermell 3mm.
1	1N4007	D1,D2,D3	Diodes.
4	10K	R1,R11,R12,R13	Resistència 1/4W.
3	1K	R7,R9,R10	Resistència 1/4W.
2	330	R2,R3	Resistència 1/4W.
1	2K2	R5	Resistència 1/4W.
1	4K7	R8	Resistència 1/4W.
1	10K – Multivolta	R6	Resistència ajustable.
1	10K – RES8SIL	R4	Arrai de resitències.
1	2,2uF/35v	C1	Condensador.
2	33PF	C2,C3	Condensador.
4	22uF/35v	C4,C5,C6,C7	Condensador.
1	470uF/35v	C8	Condensador.
1	100nF	C9,C10,C11	Condensador.
1	XTAL 11.0592MHz.	XT1	Cristall.
1	TRAFO 9V/5W.	TR1	Transformador.
1	PONT RECTIFICADOR	PR1	Pont.
1	BATERIA 3V6	BAT1	Bateria.
1	COMMUTADOR	S1,S2	2 Posicions 4 circuits
1	COMMUTADOR	S3	2 Posicions 1 circuit
1	DB9 MASCLE	CN1	Acolzat per C.I.
2	JUMPERS	JP1,JP2	Commutadors.
4	RBV-8	RE1,RE2,RE5,RE7	REGLETA DE 8
1	RBV-3	RE3	REGLETA DE 3
1	RBV-2	RE8	REGLETA DE 2
1	ZÒCAL PLCC68		
1	ZÒCAL 20 PINS		
1	ZÒCAL 8 PINS		
1	ZÒCAL 7 PINS		
2	ZÒCAL 28 PINS		