



# Receptor Morse

## Dispositius Programables — Enginyeria de Sistemes TIC

Jordi Bonet      Francisco del Àguila

7 de desembre de 2022

### Índex

<b>1</b>	<b>Objectiu</b>	<b>1</b>
<b>2</b>	<b>Material necessari</b>	<b>1</b>
<b>3</b>	<b>Descripció del receptor morse</b>	<b>2</b>
<b>4</b>	<b>Implementació</b>	<b>4</b>
<b>5</b>	<b>Estudi previ</b>	<b>4</b>
<b>6</b>	<b>Treball pràctic</b>	<b>6</b>

### 1 Objectiu

L'objectiu d'aquesta pràctica és realitzar un receptor morse, que rebrà el senyal morse per un dels pins d'entrada digital de l'AVR. Aquest senyal caldrà que el processem abans d'arribar al pin d'entrada, per tal que els *símbols morse* (*punts o ratlles*) d'1kHz es converteixin en una tensió contínua de 5 V i els silencis en una tensió de 0 V. Per fer aquest processament usarem el circuit de detecció d'envolupant que coneixeu de la pràctica *boost converter* de *Circuits i Sistemes Lineals*. Cada vegada que l'AVR detecti una seqüència de símbols i silencis que pertanyi a l'alfabet morse, transmetrà pel port sèrie la lletra o dígit corresponent.

### 2 Material necessari

Els elements que intervenen són:

1. Plataforma Arduino
2. Circuiteria necessària per fer el detector d'envolupant
3. Conjunt amplificador amb altaveus, per escoltar el senyal generat
4. Placa *protoboard*
5. Sondes necessàries, per monitorar els diferents senyal a l'oscil·loscopi

6. Trossos de cable prim per interconnectar la protoboard i l'Arduino

Els altaveus es facilitaran en el moment de la pràctica.

### 3 Descripció del receptor morse

Per realitzar aquest receptor morse caldrà fer servir els següents perifèrics:

1. port serie
2. *INT0*: interrupció externa causada per un canvi en un pin d'entrada digital
3. *timer 1*: control de la durada del punt (1 unitat), la ratlla (3 unitats) i el silenci entre símbols (1 unitat).

En aquesta ocasió es farà servir el mòdul d'interrupcions externes de l'AVR (apartat 13 de [ATmega328p]).

Les variables i constants que s'han de fer servir són les següents:

**U\_TEMPS** Els símbols i silencis tenen una durada que és múltiple d'aquesta *unitat* de temps expressada en milisegons. A la taula 1 s'indica la durada dels símbols i de cadascun dels silencis del codi morse.

**LLINDAR\_L** És el valor del comptador del *timer 1*, considerant el *prescaler* a 1024 i el rellotge del sistema de 16 MHz, quan ha passat un temps de  $2 U\_TEMPS$ . Aquest llindar diferencia entre punt (1 unitat de temps) i ratlla (3 unitats de temps) quan l'entrada digital està a nivell *high* o entre la pausa entre símbols (1 unitat de temps) i pausa entre caràcters (3 unitats de temps) quan l'entrada digital està a *low*. Aquesta constant s'ha de calcular en temps de compilació.

**LLINDAR\_H** És el valor del comptador del *timer 1*, considerant el *prescaler* a 1024 i el rellotge del sistema de 16 MHz, quan ha passat un temps de  $5 U\_TEMPS$ . Aquest llindar diferencia entre la pausa entre caràcters (3 unitats de temps) i la pausa entre paraules (7 unitat de temps) quan l'entrada digital està a *low*. Aquesta constant s'ha de calcular en temps de compilació.

**rAA** És el registre on anem emmagatzemant el tipus de símbol que rebem. Els punts els codifiquem amb un bit de valor 0 i les ratlles amb un bit de valor 1.

**rBB** És el registre de valor igual al nombre de símbols rebuts. Observeu que usem dos bytes per a codificar el codi morse rebut (a diferència del transmissor morse en què en un sol byte codificàvem el codi morse usant 5 bits per als símbols i 3 per a la seva longitud).

El mètode de recepció del senyal morse segueix els següents passos:

1. En el pin d'entrada que dispara la interrupció externa *INT0* es rep el senyal morse després de passar per un detector d'envolupant com l'usat a la pràctica *boost converter* de *Circuits i Sistemes Lineals*, [CSL-P3].
2. Aquesta pin d'entrada està inicialment a nivell *low* i s'està en el procés d'esperar l'arribada del primer símbol d'un caràcter morse.

3. Quan es rep un símbol (nivell *high*) apareix un flanc de pujada que dispara la interrupció *INT0* que ha d'inicialitzar i posar en marxa el *timer 1*, per comptar el temps des que ha aparegut el símbol fins al següent esdeveniment.
4. Quan finalitza el símbol, s'inicia un silenci (nivell *low*) i apareix un flanc de baixada que dispara de nou *INT0*. En la interrupció es consulta el valor del comptador del *timer 1* i es compara amb *LLINDAR\_L* per tal de decidir el símbol rebut i, també, s'inicialitza altre cop el *timer 1*, per comptar el temps des que ha aparegut el silenci fins al següent esdeveniment.
5. Si abans que es desbordi el *timer 1* es produeix l'arribada d'un nou símbol, es dispara de nou la interrupció *INT0* en la que es consulta el valor del comptador del *timer 1* i es compara amb *LLINDAR\_L* i *LLINDAR\_H* per tal de decidir el tipus de silenci i l'acció a realitzar:
  - pausa de símbol: es continua el procés de rebre el caràcter morse esperant un nou símbol.
  - pausa de caràcter: es consulta la taula 2 i es transmet pel port sèrie el caràcter morse codificat en ASCII i s'inicia el procés de rebre un altre caràcter morse.
  - pausa de paraula, es fa el mateix que amb la pausa de caràcter i, a més, es transmet el caràcter espai codificat en ASCII.
6. Si, en canvi, es produeix el desbordament del *timer 1* abans de l'arribada d'un nou símbol, es dispara la interrupció d'*overflow*. Aquest fet indica l'existència d'una pausa molt llarga, que s'interpreta com la finalització del codi morse i, per tant, s'actua com davant una pausa de paraula però, a més, cal tornar a l'estat inicial en el que s'està en el procés d'esperar l'arribada del primer símbol d'un caràcter morse.

En funció d'aquesta descripció la manera d'implementar el receptor morse es basarà en una màquina d'estats on els esdeveniments que poden provocar els canvis d'estat són:

1. Interrupció de flanc de pujada al pin *INT0* de l'AVR
2. Interrupció de flanc de baixada al pin *INT0* de l'AVR
3. Interrupció d'*overflow* del *timer 1*

No es definirà cap més interrupció, ja que el port serie es fa servir per transmetre i no cal fer servir interrupcions.

Quan s'ha decidit el símbol rebut, es modifica el registre *rAA*, que actua com a variable global. La manera de modificar el registre consisteix en introduir un nou bit usant desplaçament cap a l'esquerra. D'altra banda, s'incrementa el valor del registre *rBB*, per tal d'actualitzar el nombre de símbols del caràcter morse.

Quan s'ha decidit que el tipus de pausa és de caràcter o de paraula, a partir d'*rAA* i *rBB* s'ha de construir un byte, que sobreescrirà *rAA*, i que codifiqui un caràcter de l'alfabet morse de la mateixa manera que es feia amb el transmissor morse (els 5 bits de major pes amb el *codi*, i els 3 bits de menor pes amb la *longitud*). Poseu atenció a com recupereu el *codi* quan la *longitud* és inferior a 5. Aquest byte serà el que, en consultar la taula 2 ens dirà el codi ASCII a transmetre pel port sèrie.

Senyals	punt	ratlla	pausa	pausa caràcter	pausa paraula
Unitats de temps	1	3	1	3	7

Taula 1: Unitats temporals de morse

## 4 Implementació

Consulteu l'apartat 16.3 de [ATmega328p] per accedir a registres de 16 bits, com el comptador del *timer 1* que en algun moment voldreu inicialitzar.

En el disseny de la màquina d'estats, considereu totes les situacions possibles (i que per simplicitat no s'han aparegut en l'explicació sintètica anterior). En particular considereu com actuar en aquells casos que no compleixen la mètrica d'un caràcter morse, com un símbol de durada molt llarga, per exemple superior a *llindar<sub>H</sub>*, o un *codi* que no formi part del vostre alfabet morse. Seria bo que en aquests casos s'indiqués la anomalia amb un missatge transmès pel port sèrie i es tornés a l'estat inicial en què s'espera el primer símbol d'un caràcter.

Feu totes les rutines transparents i aprofiteu, sempre que pugueu, codi de pràctiques anteriors.

## 5 Estudi previ

**TASCA PRÈVIA 1** Dibuixeu el diagrama d'estats que descriu el receptor morse. Indiqueu amb claredat els esdeveniments i les accions. Especifiqueu els registres *rAA* i *rBB* escollits.

**TASCA PRÈVIA 2** Dissenyeu la rutina *crea\_codi* que, a partir d'*rAA* i *rBB*, retorni en el registre *rAA* la codificació binària del caràcter morse, 3a columna de la taula 2.

**TASCA PRÈVIA 3** Dissenyeu la rutina *troba\_codi* que té com a paràmetre *r16*, que conté la codificació binària del caràcter morse, i que retorna, també a *r16*, la codificació ASCII del caràcter morse (3a i 1a columna respectivament de la taula 2).

**TASCA PRÈVIA 4** Configureu la interrupció *INT0*. Per comprovar el bon funcionament d'aquesta interrupció, dissenyeu el programa *p10-codi1.s* que transmeti 'H' / 'L' pel port sèrie quan es produeixi un flanc de pujada / baixada en el pin *INT0*.

**TASCA PRÈVIA 5** Configureu el *timer 1* en el mode de comptador simple. Per verificar el seu funcionament, dissenyeu el programa *p10-codi2.s* que aprofiti la interrupció *INT0* per transmetre pel port sèrie '+' / '-' quan el temps entre dues interrupcions sigui superior / inferior a *U\_TEMPS*. Si abans de produir-se una nova interrupció *INT0* es produeix una interrupció d'*overflow* del *timer 1* cal transmetre pel port sèrie 'O'. Podeu utilitzar un generador de funcions per generar els senyals de prova.

**TASCA PRÈVIA 6** Dissenyeu el programa *p10-codi3.s* que implementi parcialment la màquina d'estats del receptor morse. Per parcialment cal entendre que faci tots els canvis d'estat quan es produeixi l'esdeveniment adequat, però només les accions necessàries per fer la verificació que es detalla a continuació. Per verificar el seu

Caràcter	Morse	Codificació binària
A	..	0b01000010
B	...-	0b10000100
C	-...-	0b10100100
D	...-	0b10000011
E	.	0b00000001
F	...-	0b00100100
G	--..	0b11000011
H	....	0b00000100
I	..	0b00000010
J	..-.-	0b01110100
K	-.-	0b10100011
L	...-	0b01000100
M	--	0b11000010
N	..-	0b10000010
O	---	0b11100011
P	--..	0b01100100
Q	-...-	0b11010100
R	-.-	0b01000011
S	...-	0b00000011
T	-	0b10000001
U	...-	0b00100011
V	...-	0b00010100
W	..-.	0b01100011
X	...-	0b10010100
Y	..-.-	0b10110100
Z	-...-	0b11000100
1	..-.-.-	0b01111101
2	...-.-	0b00111101
3	...-.-	0b00011101
4	....-	0b00001101
5	.....	0b00000101
6	.....	0b10000101
7	-....	0b11000101
8	--...-	0b11100101
9	-----	0b11110101
0	-----	0b11111101

Taula 2: Equivalència ASCII - morse

funcionament, usareu un senyal morse *desmodulat* (amb detecció d'envolupant ja realitzada) que es connectarà al pin *INT0* i que teniu disponible en el pin *PB5* de la pràctica anterior, *Transmissor morse*. En funció del temps transcorregut entre flancs i el sentit dels flancs cal transmetre pel port serie:

- '?' quan es rep un punt
- '-' quan es rep una ratlla
- 'S' quan es rep una pausa entre símbols
- 'C' quan es rep una pausa entre caràcters
- 'P' quan es rep una pausa entre paraules
- 'E' quan es produeix alguna anomalia

Utilitzeu la constant *DEBUGAR* per activar el mode *debugar*, en el qual es transmet pel port sèrie l'estat de la màquina d'estats, codificat amb uns caràcters que no entrin en conflicte amb la codificació anterior (per exemple '0', '1', '2'...).

TASCA PRÈVIA 7 Dissenyu el programa final *p10-codi4.s* que realitzi la recepció morse.

## 6 Treball pràctic

TASCA 8 **El treball al laboratori** consisteix en la comprovació de les tasques de l'estudi previ.

Per tal d'agilitzar la feina al laboratori utilitzeu el senyal morse que teniu disponible en el pin *PB5* del *Transmissor morse* de la pràctica anterior per tal de prescindir del detector d'envolupant. Fibalitzada la pràctica amb aquest senyal, us podreu plantejar usar el senyal morse *genuí* que teniu disponible en el pin *OC2A* del *Transmissor morse* de la pràctica anterior seguit d'un detector d'envolupant.

## Referències

[ATmega328p] Atmel. ATmega328P datasheet; <https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf>

[CSL-P3] Pràctica del *boost converter* de *Circuits i Sistemes Lineals*; <https://ocwitic.epsem.upc.edu/assignatures/csl/practiques/practica-3-boost-converter/view>