



# Transmissor morse

## Dispositius Programables — Enginyeria de Sistemes TIC

Jordi Bonet      Francisco del Àguila

9 de desembre de 2022

### Índex

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

### 1 Objectiu

L'objectiu d'aquesta pràctica és realitzar un transmissor morse. Cada pulsació que es faci d'una lletra o número es transmetrà pel port serie. L'AVR buscarà en una taula el caràcter rebut codificat en ASCII per tal de saber com generar el mateix caràcter codificat en morse en un pin de sortida, a partir dels *símbols morse (punts o ratlles)*.

### 2 Material necessari

Els elements que intervenen són:

1. Plataforma Arduino
2. Polsador
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 polsadors i els altaveus es facilitaran en el moment de la pràctica.

### 3 Descripció del transmissor morse

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

1. port serie
2. timer 2: generació del to d'1 kHz
3. timer 1: control de la durada del punt (1 unitat), la ratlla (3 unitats) i el silenci entre símbols (1 unitat).

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

1. Es rep un caràcter (columna 1 de la Taula 2) codificat en ASCII pel port serie. Aquest caràcter podrà ser una lletra 'A' a 'Z' o 'a' a 'z', un dígit '0' a '9' o un espai.
2. A partir de la Taula 2 d'equivalència d'ASCII a morse, s'obté un altre byte (columna 1 de la Taula 2) on els 5 bits de major pes indiquen els símbols (el punt amb 0 i la ratlla amb 1). Com que cada caràcter de l'alfabet morse pot tenir una longitud variable, entre 1 i 5 símbols, ens cal indicar quants d'aquests 5 bits són útils. Això ho farem amb els 3 bits de menor pes. Així, amb un sol byte codifiquem un caràcter de l'alfabet morse de longitud variable.
3. A continuació s'inicia un bucle de longitud igual a la indicada pels 3 bits de menor pes. En cada pas pel bucle és genera un símbol: punt o ratlla, segons indiquin els bits de major pes. Els símbols consecutius han d'anar separats per un silenci.
4. El timer2 es configura per generar el to d'1 kHz en el pin de sortida *OC2A*. Aquesta sortida s'anirà activant o desactivant en funció de si es vol generar un to o un silenci.
5. La durada del to o silenci generat en el pin *OC2A* es regula amb el timer1. A la Taula 1 s'indica la durada dels símbols i de cadascun dels silencis del codi morse.

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

Taula 1: Unitats temporals de morse

Amb la intenció de fer aquest transmissor el més simple possible i tenint en compte que no s'haurà de realitzar cap altra feina, no es farà servir cap tipus d'interrupció. La decisió de no usar interrupcions es pot justificar pel fet que la tasca de generació morse s'inicia amb la recepció d'un caràcter pel port serie. Un cop s'ha rebut aquest caràcter s'inicia un procés totalment pausat que no requereix ser interromput i, per tant, totes les accions d'espera fins que es compleix una condició es faran usant l'estratègia d'*enquesta*, amb un bucle que consulta contínuament la condició fins que es compleix.

### 4 Estructuració del programa

La temporalització d'aquest programa està regulada pel *timer 1*. El motiu de fer servir el *timer 1*, amb un comptador de 2 bytes, és per poder comptar fins més enllà d'1 s, sempre que es configuri adequadament el *prescaler* i tenint en compte que el rellotge del sistema és de 16 MHz. Per

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

esperar 1 unitat de temps de morse es cridarà a una rutina anomenada *espera*. Si es vol esperar més d'1 unitat de temps es faran tantes crides com calgui. La durada d'1 unitat de temps serà funció d'una constant, definida en temps de compilació, anomenada *U\_TEMPS* que, si no es diu el contrari, valdrà 100 ms. El *timer 1* es configurarà en mode de comparació. En aquest mode, el comptador s'incrementa fins tenir un valor igual al del comparador, moment en que s'activarà un flag. Aquest flag indica que ja ha passat el temps indicat per *U\_TEMPS*. Per tant s'ha de consultar fins que s'activi, i s'ha d'esborrar cada cop que s'activi, per tal que es pugui tornar a activar.

També cal definir les rutines *punt* i *ratlla*, que s'encarregaran de controlar la sortida del *timer 2*, *OC2A*, per generar el to d'1 kHz durant les unitats de temps que calgui i, generar també, el silenci d'1 unitat de temps. Això ho farà ajudant-se de la rutina *espera*.

La següent rutina que cal definir s'anomena *morse* i té com a paràmetre un byte que se li passa pel registre *r16*. Aquest byte conté la codificació binària (columna 3 de la Taula 2) del caràcter morse. Aquesta rutina s'encarrega de generar la transmissió d'aquest caràcter (columna 2 de la Taula 2). Al final de la transmissió, aquesta rutina ha d'afegir un silenci de durada 2 unitats de temps per tal que, amb el silenci d'1 unitat de temps generat per les rutines *punt* o *ratlla*, s'obtingui un silenci total de 3 unitats de temps corresponent al final de caràcter. Per a la seva implementació es suggereix que:

1. Separeu el byte del paràmetre *r16* en els 5 bits de major pes, *codi*, i els 3 bits de menor pes, *longitud*.
2. Feu un bucle amb un nombre d'iteracions igual a *longitud*. En cada iteració s'ha de transmetre el símbol (punt o ratlla) indicat pel bit de *codi* que pertoqui.
3. Testeu aquest bit de *codi* fent un desplaçament i aprofitant el flag de *carry*.

El programa principal es posa en marxa quan es rep un caràcter pel port sèrie. Ha de buscar, a la taula d'equivalència entre ASCII i morse, el caràcter rebut (columna 1 de la Taula 2) en ASCII, retornar la codificació binària del codi morse, amb *codi* i *longitud*, (columna 3 de la Taula 2) i generar en la sortida *OC2A* el senyal morse (columna 2 de la Taula 2) d'aquest caràcter. Si el caràcter rebut és un espai, enlloc d'accedir a la taula, cal generar un silenci de 4 unitats de temps que, sumades a les 3 unitats de temps del final de l'últim caràcter, fan les 7 unitats de temps corresponents al canvi de paraula.

Recordeu que cal preservar i restaurar tots aquells registres que s'utilitzin dins una rutina.

Seguiu el suggeriment d'aprofitar codi de pràctiques prèvies com la de port sèrie o xifrador.

## 5 Estudi previ

**TASCA PRÈVIA 1** Configureu el *timer 2* per generar un to d'1 kHz en el pin *OC2A*. Per verificar aquesta configuració, dissenyeu el programa *p9-codi1.s* que, en funció d'un pin d'entrada, activi o desactivi el to a *OC2A*.

**TASCA PRÈVIA 2** Configureu el *timer 1*. Determineu l'expressió que calcula (en temps de compilació) els valors d' *OCR1AH* i *OCR1AL* en funció del paràmetre *U\_TEMPS* expressat en ms. Disseny la rutina *espera*. Per comprovar el bon funcionament del *timer 1* i la rutina *espera*, dissenyeu el programa *p9-codi2.s* que encengui el LED durant 1 unitat de temps i l'apagui durant 2 unitats de temps.

TASCA PRÈVIA 3 Dissenyeu les rutines *punt* i *ratlla* aprofitant el treball dels apartats anteriors. Per comprovar el bon funcionament d'aquestes dues rutines, dissenyeu el programa *p9-codi3.s* que generi en el pin *OC2A* el missatge *SOS* de forma ininterrompuda.

TASCA PRÈVIA 4 Dissenyeu la rutina *morse* aprofitant el treball dels apartats anteriors. Per comprovar el seu bon funcionament, dissenyeu el programa *p9-codi4.s* que generi en el pin *OC2A* el codi morse (columna 2 de la Taula 2) de la codificació binària (columna 3 de la Taula 2) rebuda pel port serie. Useu *cutecom*.

TASCA PRÈVIA 5 Dissenyeu el programa principal *p9-codi5.s*. Pel port sèrie es rep el codi ASCII del caràcter de la columna 1 de la Taula 2, es busca l'equivalència a la taula morse per obtenir la codificació binària (columna 3 de la Taula 2 i es genera en el pin *OC2A* el codi morse (columna 2 de la Taula 2). Si es rep un espai s'ha de generar el silenci entre paraules.

TASCA PRÈVIA 6 *Avançat*: Considereu l'ús d'interrupcions, comenteu canvis de disseny en el codi si les useu i valoreu la seva implementació.

TASCA PRÈVIA 7 Per tal d'agilitzar la feina al laboratori de la següent pràctica, *Receptor morse*, modifiqueu les rutines *punt* i *ratlla* per tal que, a més de generar símbols amb la modulació d'1 kHz en el pin *OC2A*, genereu símbols sense modulació (de valor 5 V quan hi hauria d'haver modulació i 0 V quan hi ha una pausa) en el pin *PB5* de l'AVR (LED). Aquest senyal també us pot ser útil quan useu un oscil·loscopi amb una freqüència de mostreig baixa que no us permet visualitzar correctament el senyal morse en una escala de segons, a causa del submostreig de la modulació d'1 kHz.

## 6 Treball pràctic

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

### Referències

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