

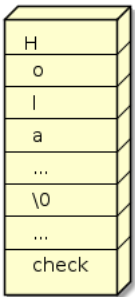
Dispositius Programables

Durada: 2h

Final - Gener 2023

Punts totals: 13

1. Una aplicació de comunicació entre dos Arduinos intercanvia missatges de text. L'aplicació es pot assemblar (compilar) per a diferents mides màximes dels missatges. La mida màxima ve determinada per la constant MAXT. El format del missatge un cop està desat a memòria és el següent:



Aquesta estructura de dades es troba a partir de la posició de memòria de l'etiqueta **missatge:**. En primer lloc es troben cadascun dels caràcters, que acaben delimitats pel caràcter no imprimible que fa de sentinella corresponent al byte 0. El nombre màxim de caràcters pot arribar a ser de MAXT. A la última posició disponible es troba el byte *check* que servirà per fer una detecció de possibles errors. Cal fer notar que ni el caràcter sentinella ni el *check* formen part de la mida del missatge.

La recepció d'aquests missatges es fa a través de la interrupció RX del port sèrie. El caràcter de retorn de carro (valor 13) serveix per marcar tant l'inici d'un nou missatge com el final de l'anterior.

El càlcul del byte de *check* consisteix en sumar el valor actual del *check* amb el caràcter rebut tenint en compte que si hi ha carry es torna a sumar aquest carry. El valor inicial del *check* és 0.

L'estructura de dades **missatge:** queda completament actualitzada cada vegada que arriba un nou caràcter pel port sèrie. Això vol dir que tan el sentinella ocupa la posició que li correspon com el *check* té el valor que li correspon.

Disposeu de les següents macros/subrutines: **ledon**, **ledoff**, **delay1s** (genera un retard de 1 segon).

Algunes directives d'assemblador són: `{.byte value}`, `{.space size, value}`, `{.fill repeat, size, value}`.

- a) Defineix a memòria de dades la reserva d'espai per contenir l'estructura de dades corresponent a un missatge amb l'etiqueta **missatge:**. Aquest missatge inicialment ha d'estar buit. Indica la secció de memòria on està definit.
- b) De quina manera faries referència al *check* de **missatge:** a partir de conèixer l'adreça de l'etiqueta **missatge:**?
- c) Defineix la subrutina **mten** que té com a paràmetre un missatge qualsevol a través del registre X i retorna a r18 la mida de caràcters que té.
- d) Quins canvis s'han de fer a **mten** per a que també retorni a X l'adreça on es troba el sentinella (byte 0).
- e) Dissenyau la subrutina **addcar** que té com a paràmetres r16 i r17. Fa la suma dels 2 paràmetres i si hi ha carry, incrementa aquest carry al resultat. El resultat es retorna al mateix r16.
- f) (2 punts) Dissenyau la rutina d'atenció a la interrupció de RX que actualitza l'estructura de dades **missatge:**. Aproveiteu les subrutines que s'han definit als apartats anteriors.
- g) Amb l'objectiu de millorar l'eficiència en temps d'execució de la rutina d'interrupció RX, descriu com la dissenyaries.

- h) Dissenya la subrutina **comprova** que encén el led en el cas que la mida de **missatge:** sigui parell i el *check* siguin tot 1's. En cas contrari apaga el led.
 - i) Es vol cridar a **comprova** cada segon. Es proposen 2 alternatives. Indica quina o quines d'elles són adequades pensant en la integritat de les dades de l'estructura de dades de **missatge:**. Si és possible i cal, quins són els canvis que faries per adequar-les. En el cas que ii. sigui adequat, indica quin timer seria candidat i perquè.
 - i. El bucle principal crida a **comprova** seguit de **delay1s** repetidament
 - ii. Una interrupció de timer cada segon crida a **comprova**
2. Respon si són certes o falses les següents afirmacions. Justifica la resposta. Una resposta sense justificació compta la meitat. Les respostes incorrectes resten la meitat, excepte si la justificació té sentit. Els pesos d'aquests apartats són 0.5 punts.
- a) El contingut inicial de les variables definides a la memòria de dades, secció .data es troben ocupant espai a la memòria de programa.
 - b) Per dividir per 2 el valor d'un registre s'han de desplaçar els bits a la dreta 1 posició mantenint el bit de més pes al mateix valor que havia abans del desplaçament.
 - c) LDI r16, 254 i LDI r16, -2 és diferent a ulls del programador però és equivalent des del punt de vista de la CPU.
 - d) En l'AVR el pas de paràmetres per pila permet passar més dades que a través de registres.
 - e) Passar dades per referència (passar l'adreça d'on es troben les dades) és més eficient que passar les dades per valor quan les dades són de mida gran
 - f) Els registres X, Y, Z són la millor manera de passar dades per referència amb l'AVR.