



MaquinaV – Automatic Sales Machine

Assessment Component P2 (20%) of Computer Architecture

School year: 2022/2023

Delivery date and discussion: 26-04-2023

1. Description of the work number two

1.1. General overview

In this project, students should develop a program that allows the sale of products in a machine. The graphical interface (or display) should simulate the selection and sale of a product, as well as the visualization of the machine's stock by authentication. The processor to be used will be PEPE, as shown in Annex I.

The initial interface should have two options as shown in the following figure:

```

-----
M A C H I N E   M A D E I R A
      W E L C O M E
-----
1 )   P r o d u t s
2 )   S t o c k
-----

```

Figure 1 – Initial selection menu

After the user selects the products, a menu with the product categories should be displayed, as shown in the example in Figure 2.

```

-----
- - -   C a t e g o r y   - - -
-----
1 )   D r i n k s
2 )   S n a c k s
7 )   C a n c e l
-----

```

Figure 2 - Product category menu

After selecting the category, another sub interface should be displayed with at least 3 products, as well as the respective monetary value, where the user can choose:

```

-----
- - -   S n a c k s   - - -
-----
1 )   M & M           1 . 0 0
2 )   B o u n t y     0 . 5 0
3 )   M a r s         0 . 8 0
4 )   B a c k

```

Figure 3 – Product category sub-menu

As soon as the user selects the product, a payment menu should be displayed where the user can select the monetary amount to be entered and make the purchase. The machine only accepts the following monetary values (coins): 0.10€, 0.20€, 0.50€, 1.00€ and 2.00€ and 5.00€.

Once the payment has been made, a receipt must be printed, which must contain the selected product with its monetary value, the monetary value entered in the payment and the change, as shown in the following figure:

```

-----
      R E C E I P T
-----
B r i s a . . . . . 1 . 0 0
I n s e r t e d . . . . 2 . 6 0
C h a n g e . . . . . 1 . 6 0
-----

```

Figure 4 – Receipt

In the initial interface, if the user selects the stock, the following interface should be displayed:

```

----- S t o c k -----
-----
      I n s e r t
      P a s s w o r d

1 )   C o n f i r m
4 )   B a c k

```

Figure 5 - Authentication for viewing stock

When entering the password (authentication), the user must have access to the stock where he can view the quantities of products and the monetary values present in the machine, as shown in the following example:

```

- -   S t o c k   2 / 3   - - -
B u r n . . . . . 0 0 0 0 5
5 E u r o s . . . . . 0 0 0 0 0
2 E u r o s . . . . . 0 0 0 0 6
1 E u r o . . . . . 0 0 0 0 4
5 0 C e n t . . . . . 0 0 0 0 5
1 )   N e x t

```

Figure 6 - Stock and quantities

Stock management, any necessary error notifications, product names and product monetary value are free to choose.

Note: The keyword must contain at least 4 characters with the following requirements:

- At least one number;
- At least one lower case;
- At least one capital letter;
- At least one special character;

In order to limit the choice of keyword, Annex II presents the characters to be used.

1.2. User interface

The user interface will be performed through a display (output peripheral), dimensions 7x16 (7 lines of 16 characters – bytes). The user's interaction with the system is carried out through a single input peripheral called PER_EN.

In this project it is expected that the student will be able to develop a personalized solution and framed in the presented theme. **The project must be running in Run mode of the simulator.**

2. Work Plan

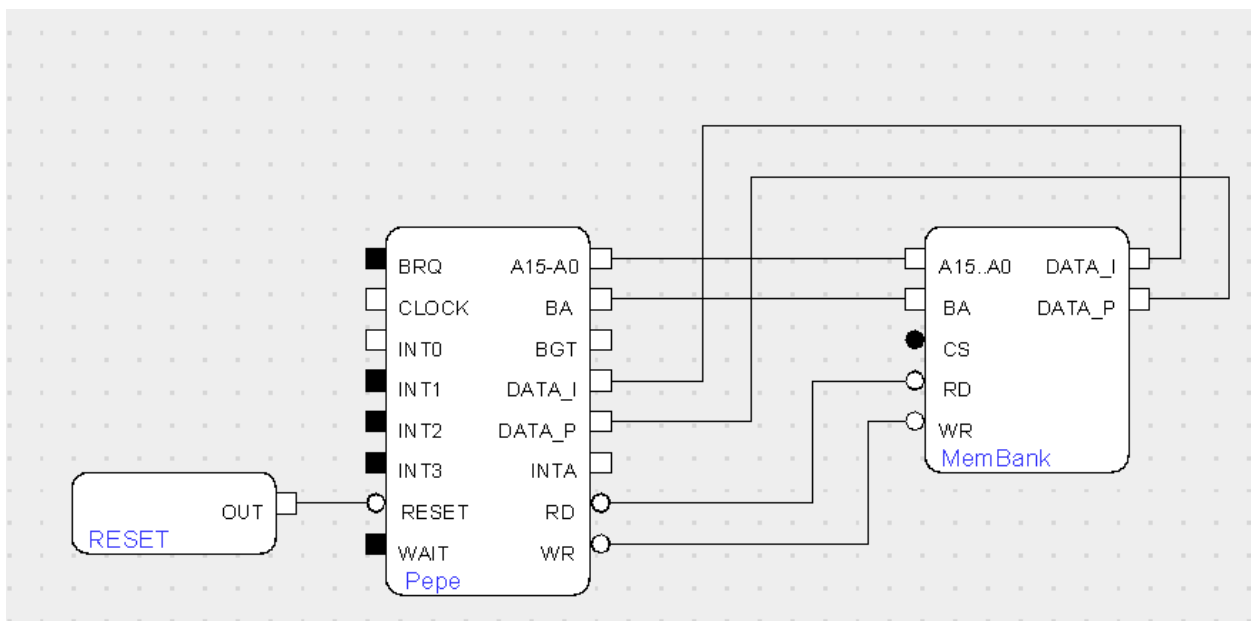
The following items will be taken into account in this evaluation work:

- Specification and design of the dataflow diagrams;
 - Programming in *assembly* language and implementation;
 - Implementation of extra functions such as:
 - The management of the stock of coins to be returned to the user (example: the monetary value to be returned is 1.00€. It will be removed from the stock coins 2 x 0.50€ or 1x 0.50€, 2 x 0.20€ and 1 x 0.10€ or 5 x 0.20€ or 10 x 0.10€);
 - If the monetary value of the payment is equal to, or greater than, the value of the product, the purchase is processed automatically without the user having to introduce further inputs;
 - Separation of stock between products and coins;
 - If there are not enough coins in stock, an extra menu is displayed that allows the user to continue with the purchase, without receiving part or all of the change, or to cancel the purchase;
 - Different monetary values in each product;
 - Turn ON/OFF the vending machine.
 - Elaboration of the report.
-
- Specification and dataflow diagrams
 - Brief study of requirements for the *software* solution;
 - Drawing of dataflow diagrams for the main control modes and secondary procedures.
 - Programming in *assembly* and implementation
 - Programming the algorithm in *assembly* language;
 - Experimental verification of the program.
 - Work report
 - Maximum of 8 pages;
 - Cover with course, teachers and students' identification;
 - Project description and analysis of results;
 - Annex A with dataflow diagrams;
 - Annex B with the code in *assembly* language, which should be organized and commented.

3. Important information

- The second evaluation work (report and oral) represents 20% of the final grade and has a minimum grade of 8/20;
- The work should be done individually or in groups of two elements;
- Code should have comments, otherwise penalties will be given;
- The report in PDF, the file with the *assembly* code (.asm) and the simulator used should be zipped in one file, and sent to the Student Support Office ("trabalhos@uma.pt") no later than 26-04-2023. In the email body that will be sent, you should state your name and student number, the course name, the work identification and the teachers' name.
- On the day of the discussion (26-04-2023), each student has to show the program working correctly, and thus it is asked that you bring your personal laptop with you.
- In the discussion evaluation it will be taken into account if any extra/additional function to those proposed in the work was implemented, with a value of 1 point out of 20.

Annex I - PEPE processor and connection to the data memory



Annex II – ASCII code

Bin	Oct	Dec	Hex	Sinal	Bin	Oct	Dec	Hex	Sinal	Bin	Oct	Dec	Hex	Sinal
0010 0000	040	32	20	(espaço)	0100 0000	100	64	40	@	0110 0000	140	96	60	`
0010 0001	041	33	21	!	0100 0001	101	65	41	A	0110 0001	141	97	61	a
0010 0010	042	34	22	"	0100 0010	102	66	42	B	0110 0010	142	98	62	b
0010 0011	043	35	23	#	0100 0011	103	67	43	C	0110 0011	143	99	63	c
0010 0100	044	36	24	\$	0100 0100	104	68	44	D	0110 0100	144	100	64	d
0010 0101	045	37	25	%	0100 0101	105	69	45	E	0110 0101	145	101	65	e
0010 0110	046	38	26	&	0100 0110	106	70	46	F	0110 0110	146	102	66	f
0010 0111	047	39	27	'	0100 0111	107	71	47	G	0110 0111	147	103	67	g
0010 1000	050	40	28	(0100 1000	110	72	48	H	0110 1000	150	104	68	h
0010 1001	051	41	29)	0100 1001	111	73	49	I	0110 1001	151	105	69	i
0010 1010	052	42	2A	*	0100 1010	112	74	4A	J	0110 1010	152	106	6A	j
0010 1011	053	43	2B	+	0100 1011	113	75	4B	K	0110 1011	153	107	6B	k
0010 1100	054	44	2C	,	0100 1100	114	76	4C	L	0110 1100	154	108	6C	l
0010 1101	055	45	2D	-	0100 1101	115	77	4D	M	0110 1101	155	109	6D	m
0010 1110	056	46	2E	.	0100 1110	116	78	4E	N	0110 1110	156	110	6E	n
0010 1111	057	47	2F	/	0100 1111	117	79	4F	O	0110 1111	157	111	6F	o
0011 0000	060	48	30	0	0101 0000	120	80	50	P	0111 0000	160	112	70	p
0011 0001	061	49	31	1	0101 0001	121	81	51	Q	0111 0001	161	113	71	q
0011 0010	062	50	32	2	0101 0010	122	82	52	R	0111 0010	162	114	72	r
0011 0011	063	51	33	3	0101 0011	123	83	53	S	0111 0011	163	115	73	s
0011 0100	064	52	34	4	0101 0100	124	84	54	T	0111 0100	164	116	74	t
0011 0101	065	53	35	5	0101 0101	125	85	55	U	0111 0101	165	117	75	u
0011 0110	066	54	36	6	0101 0110	126	86	56	V	0111 0110	166	118	76	v
0011 0111	067	55	37	7	0101 0111	127	87	57	W	0111 0111	167	119	77	w
0011 1000	070	56	38	8	0101 1000	130	88	58	X	0111 1000	170	120	78	x
0011 1001	071	57	39	9	0101 1001	131	89	59	Y	0111 1001	171	121	79	y
0011 1010	072	58	3A	:	0101 1010	132	90	5A	Z	0111 1010	172	122	7A	z
0011 1011	073	59	3B	;	0101 1011	133	91	5B	[0111 1011	173	123	7B	{
0011 1100	074	60	3C	<	0101 1100	134	92	5C	\	0111 1100	174	124	7C	
0011 1101	075	61	3D	=	0101 1101	135	93	5D]	0111 1101	175	125	7D	}
0011 1110	076	62	3E	>	0101 1110	136	94	5E	^	0111 1110	176	126	7E	~
0011 1111	077	63	3F	?	0101 1111	137	95	5F	_					