Architetture dei Sistemi Di Elaborazione **Laboratory 9**	Delivery date: 16th December 2022 Expected delivery of lab_09.zip must include: - zipped project folder - this lab track completed and converted to pdf format.

Exercise 1) – Read and convert a MORSE message.
Samuel Finley Breese Morse invented the Morse code in 1836. It is based on long and short signals (dashes and points) representing numbers and alphabet letters.

A ·−	J ·−−−	S ···	2 ··−−−
B −···	K −·−	T −	3 ···−−
C −·−·	L ·−··	U ··−	4 ····−
D −··	M −−	V ···−	5 ·····
E ·	N −·	W ·−−	6 −····
F ··−·	O −−−	X −··−	7 −−···
G −−·	P ·−−·	Y −·−−	8 −−−··
H ····	Q −−·−	Z −−··	9 −−−−·
I ··	R ·−·	1 ·−−−−	0 −−−−−

Use the LANDTIGER board and the System-On-Chip LPC1768 for implementing a system that deciphers a message in Morse code and returns:

- Its ASCII conversion.
- The total number of translated Morse symbols.

The system must work as follow:

- The message to translate is in a vector of 8-bit unsigned integers with a maximum length of 100 elements. The vector can only contain 0,1,2,3,4 as values.
- **The 0 is used for the dot symbol, 1 for the dash symbol, 2 indicates the character end, 3 for the space, and 4 for the end of the sentence.**

For example, the vector can be initialized as shown below:

H O L A M U N D O 1 2

0000211120100201311200121021002111301111200111 4

The sentence *HOLA MUNDO 12* can be represented above, and it contains 13 symbols. The space has to be considered as a symbol.

1. By pressing **KEY1**, it starts the message reading(in C).
   - All LEDs are switched off.
   - At every change of symbol (number or letter), LEDs show the number of read symbols until that point.
2. At the end of the message, all LEDs are switched on, independently from the number of read symbols.
3. By pressing **KEY2**, it starts the message conversion in ASCII(in Assembly).
   - During the conversion phase, the buttons **INT0** and **KEY1** are disabled.
   - The assembly function for translating the message is invoked.

The assembly function has the following prototype:

int translate_morse(char* vett_input, int vet_input_lenght, char* vett_output, int vet_output_lenght, char change_symbol, char space, char sentence_end);

The function arguments are:

- char* vett_input, Morse code vector to convert.
- int vet_input_lenght, length of the vector to convert.
- char* vett_output, ASCII output vector.
- int vet_output_lenght, length of the output vector.
- char change_symbol, the value representing the change of symbol in the input vector.
- char space, the value representing the space in the input vector.
- char sentence_end, the value representing the end of the sentence in the input vector.

The function returns the total number of converted Morse symbols into an integer variable called RES. The space has to be considered as a converted symbol.
In *vett_output* there is the translation in ASCII of the input vector.

Al rientro dalla funzione ASM:

- All LEDs show the total number of converted symbols (letters, numbers, and spaces).
- Buttons **INT0** and **KEY1 must be enabled,** by pressing **INT0** the process has to restart from point 1).

Hint (Morse code):

A	01	**J**	0111	**S**	000	**2**	00111
B	1000	**K**	101	**T**	1	**3**	00011
C	1010	**L**	0100	**U**	001	**4**	00001
D	100	**M**	11	**V**	0001	**5**	00000
E	0	**N**	10	**W**	011	**6**	10000
F	0010	**O**	111	**X**	1001	**7**	11000
G	110	**P**	0110	**Y**	1011	**8**	11100
H	0000	**Q**	1101	**Z**	1100	**9**	11110
I	00	**R**	010	**1**	01111	**0**	11111