Architetture dei Sistemi Di Elaborazione **Extra point #1** **Max 2 points**	Delivery date: 9th Gennaio 2023 Expected delivery of extapoint1.zip must include: - zipped project folder - this document compiled in pdf - A 4 minutes video with audio (.mp4 or .avi) explaining how the project works; the video has to show a software debug session with all significant peripheral windows opened; the audio must be a voice recording of you describing (in Italian or English) the behavior of the running system.

Purpose of Part 1: to acquire full confidence in the usage of the KEIL **software debug** environment to emulate the behaviour of the LPC1768 and the LANDTIGER Board.

This part is evaluated to assign a maximum of 2 extra-points for qualified students taking the exam with a mark >= 18

Make a Tamagotchi!

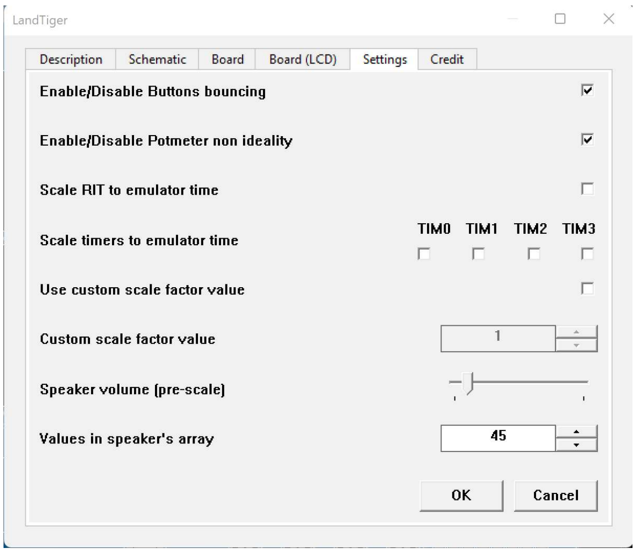
Tamagotchi is a famous toy from the end of the '90s and the beginning of the new millennium (but they still make them with genetics, NFC, and Bluetooth!). The goal of the game was to take care of a virtual pet, taking care of its basic needs, such as hunger and happiness. Of course, in the end, many kids just had tons of them on their consciences.

In Keil µVision, use the LANDTIGER **emulator** (or the actual board if you are able to have one) for implementing a basic Tamagotchi!

Please deliver a zip folder with all the files of your project. At the end of the Keil project folder, please add "EMULATOR" if you developed it using the emulator or "BOARD" if you had the opportunity of developing the project on the actual board.

Example: extrapoint1_emulator.zip **or** extrapoint1_board.zip

Please attach any useful comments and your emulator configurations (substitute the image below). Your project will be evaluated according to your specification!!


LandTiger

Description | Schematic | Board | Board (LCD) | Settings | Credit

Enable/Disable Buttons bouncing	✓
Enable/Disable Potmeter non ideality	✓
Scale RIT to emulator time	☐
Scale timers to emulator time	TIM0 ☐ TIM1 ☐ TIM2 ☐ TIM3 ☐
Use custom scale factor value	☐
Custom scale factor value	1
Speaker volume (pre-scale)	
Values in speaker's array	45

OK | Cancel

Additional Comments (C-variable/defines defined in your code, e.g., scaling factors) :
!!TESTATO SU EMULATORE!!

In *GraphicVariables.h* sono contenuti i vari #define inerenti alle dimensioni/posizioni degli elementi grafici, oltre alle matrici di colori RGB565 usate dalle funzioni per disegnare gli elementi grafici. Nota: nel video in un certo momento alcuni dei #define appaiono come variabili statiche, in quanto mi sono accorto successivamente dell'errore (post editing del video).

In "IRQ_RIT.c" sono presenti diverse variabili utilizzate come contatori, in quanto i tempi richiesti dalla traccia sono tutti multipli di 1 sec e quindi facilmente gestibili attraverso soltanto il RIT (che lavora a 50 ms per il polling del joystick):

- timer_char_refr: usata per l'animazione del personaggio (ciclo su 1 sec)
- battery_counter: usata per l'aggiornamento dei valori di happiness e satiety, e di conseguenza delle batterie
- timer_counter: usato per lo scorrere del tempo (Age del personaggino)
- busy_counter: usato per l'animazione del mangiare (o per le scritte che appaiono se il personaggio ha già i valori massimi). Il ciclo dura poco più di un secondo, perché oltre al mangiare è presente il movimento del personaggino verso il cibo.

Oltre ai counter sono presenti anche dei flag che sbloccano (o disabilitano) funzioni e animazioni:

- is_busy: indica se il personaggio è occupato a mangiare, disabilitando le animazioni normali e abilitando quelle adatte. Disabilita inoltre la possibilità di selezionare i pulsanti mediante il joystick
- is_going: indica che la sequenza di fuga è stata innescata
- is_sad: usato se il valore di almeno una batteria scende sotto al due. Cambia l'animazione del personaggio in una più triste per indicare che c'è il rischio di perdere
- reset: se settato a 1 significa che è stata chiamata la funzione Resetter() (dal main o a seguito della fuga del personaggio). Disabilita le operazioni del RIT_IRQHandler

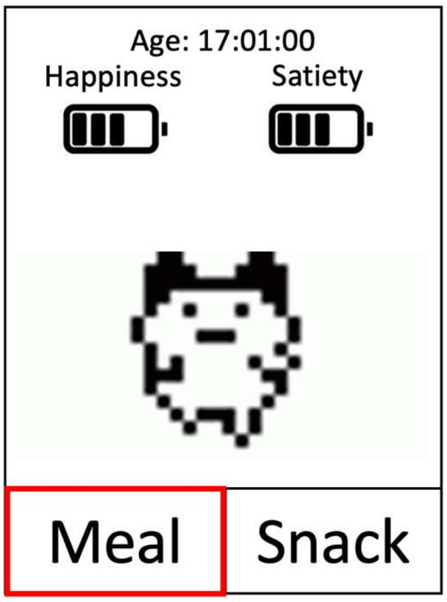
Nota: Nel video e per velocizzare i test ho adoperato un valore di RIT pari a 10 ms, a causa di problemi avuti con il RIT Scaler dell'emulatore (dovuto probabilmente alla lentezza dell'operazione di disegno). Tuttavia avendo provato sulla scheda il funzionamento delle parti di animazione del personaggio (perlomeno per quanto riguarda la parte di fuga e standard, niente col cibo) posso dire che non dovrebbe verificarsi flickering una volta settato il valore di RIT corretto. In caso di flicker su scheda, comunque, basterebbe modificare il valore di confronto con i counter (es la n-esima animazione farla a counter==20 invece che ==10, etc).
In ogni caso, se voleste testare il codice così come ho fatto io, oltre alle impostazioni presenti nell'immagine sopra per quanto riguarda l'emulatore, dovreste decommentare la riga 46 e commentare la riga 47 del main.

Specification 1)

Use the display library to create and show your own Tamagotchi. Be creative! A basic “animation” is also required for this point. You are free to decide what kind of animation to create and how complex you want to make it but remember that there is a limit on the dimension of the code that you can compile and device performances, so just try and be ready to simplify it if needed.

Requirement 1: You are required to display the Tamagotchi with the shape of your choice in the middle of the LCD.
The animation has a refresh rate of 1 second.

Make it alive!



Specification 2)

Create the logic of your virtual pet.

At the top of the screen, the STATUS of your virtual pet is shown. <u>The status view must always be visible at the top of the screen, together with Tamagotchi.</u> There are three information in this status view:

a) The "Age" of your Tamagotchi (in hours:minutes:seconds);
b) The "Happiness" of your Tamagotchi (on a scale of your choice);
c) The "Satiety" of your Tamagotchi (again on a scale of your choice).

Age: 17:01:00
Happiness
Satiety

Create a FOOD MENU on the bottom of the screen. <u>The food menu must always be visible at the bottom of the screen, together with Tamagotchi.</u>
Your icons may be different or simplified text abbreviations. Note that the RED square is the currently selected menu. The option in the food menu should be:

a) "Meal" to satisfy the <u>hungry</u> level of your Tamagotchi;
b) "Snack" to satisfy the <u>happiness</u> level of your Tamagotchi.

Selecting a food or a snack triggers an animation with a sketch of the selected food (it may be a simple circle or square, just differentiate the snack from the meal) and the Tamagotchi "eating" it. It's not necessary to make a complex animation, but the Tamagotchi must move toward the food/snack (appearing in a fixed position on the screen) for eating.

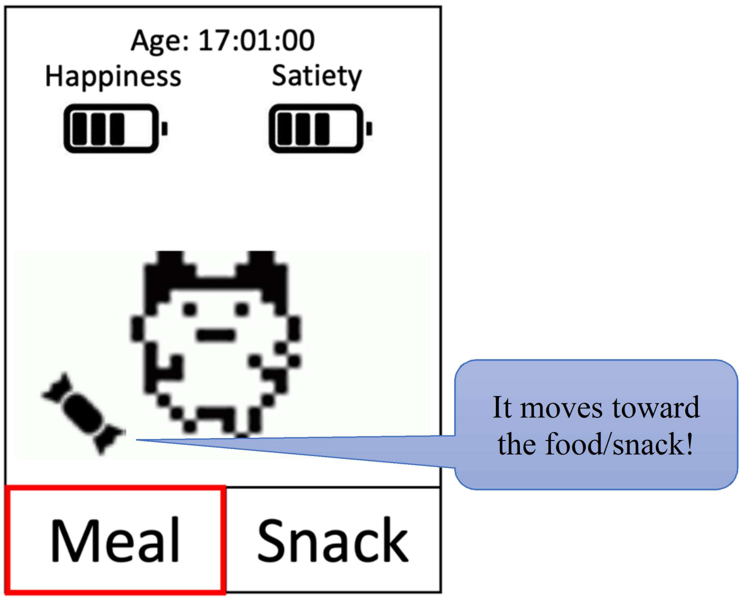
The eating must last for at least 1 second.
Food selection during eating is disabled.

- Eating a snack increases the happiness of your Tamagotchi by one bar (value of your scale).
- Eating a meal increase the Satiety of your Tamagotchi by one bar (value of your scale).

Requirement 2: You are required to develop the Food/Status menu.
Note that images are only for explanation purposes.

Note that "Happiness" and "Satiety" decrease every 5 seconds of one bar (value of your scale).

The JOYSTICK is used to select the food options (**LEFT/RIGHT** for choosing the desired food and **SELECT** to choose it).



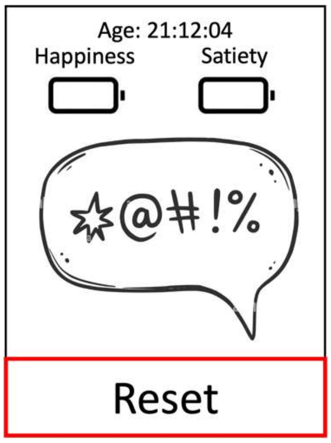
The eating must last for at least 1 second.
Food selection during eating is disabled.

Your Tamagotchi runs away!

Specification 2)

Your Tamagotchi should leave you if you don't take care of it. If you reach the end of the Happiness or Satiety scale, the runaway sequence is triggered.
You should have a very basic animation of the Tamagotchi leaving the screen. After it's run away:

a. A message or icon should be placed in the screen to signal that your Tamagotchi left you.
b. The <u>age counter stops</u> counting the age of your Tamagotchi.
c. A **reset button replaces the food options.**



By clicking the reset button, a complete reset of your Tamagotchi is triggered, with a new Tamagotchi appearing on your screen, with **full Happiness and Satiety and with 00:00:00 as age.**

Requirements 3:
<u>You are required to develop the runaway sequence and the reset functionality.</u>
<u>Note that images are only for explanation purposes.</u>

For the Reset functionality, you simply must press the joystick (**SELECT**).