

Project specifications

Introduction

For the project, you are free to choose one of the following three projects. You are also free to propose your own project (with the elements you have in the lab kit, or with other if you have some additional elements at home). In that case, please contact the teaching staff so that we can agree on the specifications of your self-proposed project. We remind you that the project has to be done in groups of 3-4 and that it counts for $\frac{1}{3}$ of the final grade of the course.

1 Distress beacon

The aim of this project is to build an emergency beacon that is capable of sending three types of distress signals, as shown in Figure 1 :

1. a luminous signal sending morse code ;
2. an audio signal sending morse code ;
3. a visual flag type signal sending semaphore code.

Morse code is a succession of short and long luminous or audio signals, which can describe the letters of the alphabet. Semaphore code uses two flags to describe the different letters of the alphabet as shown in this image (note that your servos can only rotate 180° , so not all letters can be done. You are free to change the combination of flags for those letters). Different inputs will be used to allow different functionalities of the distress beacon.

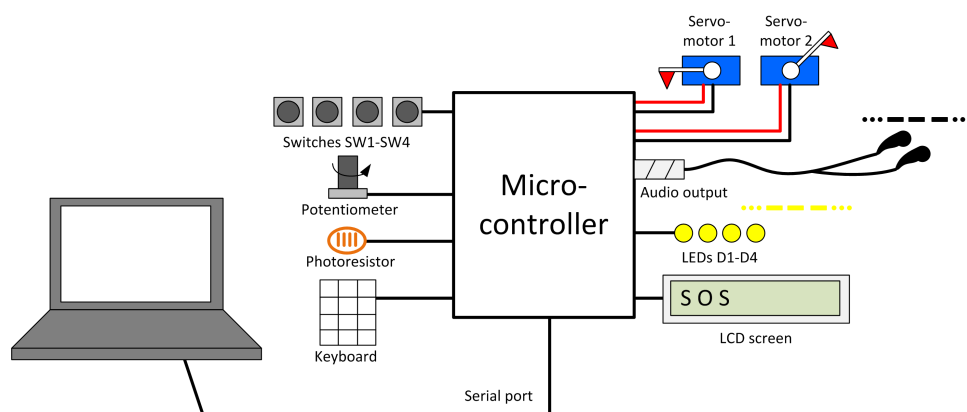


FIGURE 1 – Schematic representation of the distress beacon inputs and outputs.

The inputs should be configured as follows.

- **SW1** : When pressing switch SW1, the LEDs should light up and the audio output should emit a sound as long as the switch is pressed. When the switch is unpressed, the LEDs should go out and the audio output should stop emitting a sound.

- **SW2** : When pressing switch SW2, the LEDs and audio output should emit a short morse signal. The short morse signal should be 250 ms long.
- **SW3** : When pressing switch SW3, the LEDs and audio output should emit a long morse signal. The long morse signal should be 750 ms long.
- **SW4 and Pot** : When turning the potentiometer (Pot), the flag should be going up according to the turning of the potentiometer. When pressing SW4, we change the flag that is controlled with the potentiometer.
- **Photoresistor** : The photoresistor can detect whether there is ambient light or not. When there is light, only the audio output and semaphore flags should be active. When there is no light, only the audio output and the LEDs should be active.
- **Keyboard** : We can use the keyboard to send a series of pre-recorded message. For example, pressing “1” and “*” should start sending “SOS” in morse and semaphore code. Pressing “2” and “*” should send “BEAMS” in morse and semaphore code. Other messages are for you to chose.
- **Serial port** : The user computer can send ASCII characters through the serial port, which are then sent in morse and semaphore code by the distress beacon. The microcontroller also sends two types of characters to the user computer over the serial ports : “.” and “-” (which represents short and long morse signals that are sent by the distress beacon).
- **LCD screen** : The LCD prints the ASCII characters that are sent by the distress beacon (this feature can only be used when using the keyboard or the serial port). You can also chose to print additional information on the LCD screen.

2 Never-lose

The aim of this project is to build a system that can play Google Chrome's *dino* game autonomously (without ever losing of course). It's the game we have all played at some point when there were no more internet connection. You can play it by opening a Chrome browser and typing the following URL. When you press the spacebar, the game will launch. Your *Never-lose* system should :

1. use the photoresistors to detect incoming obstacles on the screen ;
2. use the servomotors to press the spacebar and the “down” arrow.

You can attach the photoresistors to your screen with some duck tape, the servomotors should be taped to the keyboard or held down with some weights to ensure that they don't move when pressing the computer's keys.

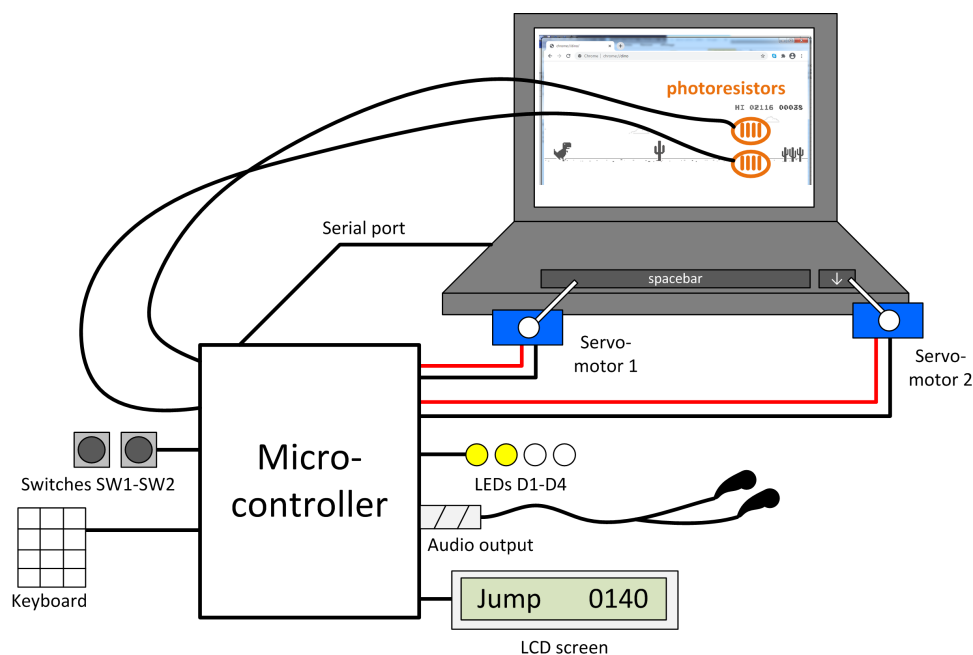


FIGURE 2 – Schematic representation of the never-lose system.

The inputs and outputs should be configured as follows.

- **SW1** : When pressing SW1, the servo controlling the spacebar should press the spacebar.
- **SW2** : When pressing SW2, the servo controlling the down-arrow key should press the down-arrow key.
- **SW3 and score counting** : When pressing SW3, the score should be reset to zero. The score should start increasing the first time the spacebar is pressed after resetting the score to zero (in the dino game, the score increases with 10 points per second). The score should be printed on the second half of the LCD.
- **LED1-LED2** : These LEDs should light up when the dinosaur is jumping.
- **LED3-LED4** : These LEDs should light up when the dinosaur is ducking.
- **Photoresistor** : The photoresistors should be used to detect incoming obstacles (one photoresistor for the ground obstacles, one for the flying obstacles).

- **Keyboard :** You can pick two keys of the microcontroller keyboard that should have the same effects as SW1 and SW2.
- **Serial port :** The user computer can send the “jump” and “duck” command through the serial port (for example for launching the game), which should result in the correct servo pressing the correct key. The microcontroller should also send the following messages to the user computer over the serial ports : “Jump” and “Duck” when the dinosaur is jumping/ducking.
- **LCD screen :** The LCD screen should print “Jump” when the dinosaur is jumping, “Duck” when the dinosaur is ducking. The second part of the LCD should print the score.
- **Audio out :** The audio output should emit one sound when the dinosaur is jumping, another sound when the dinosaur is ducking.

As a bonus, the following elements can be implemented :

- a system to detect when the day changes into night on the game ;
- ...

3 One-dimensional gimbal

The aim of this project is to realize a one-dimensional gimbal, i.e. a system that will maintain a platform horizontal, even if the whole object (in this case the PSoC board) is rotated. To maintain the horizontal board horizontally, the servo motor (attached to the PSoC board) will be used to rotate the horizontal board with respect to the PSoC board. Rotations will be detected using a ADXL335Z 3-axes accelerometer (datasheet 1 and datasheet 2) , which can be attached to the PSoC board or to the horizontal board. Note that an accelerometer will measure the earth's vertical gravitation vector and project it along it's 3 axes, which can be used as a measure of orientation of the accelerometer. Get in touch with us if you go for this project because you will need some extra equipment.

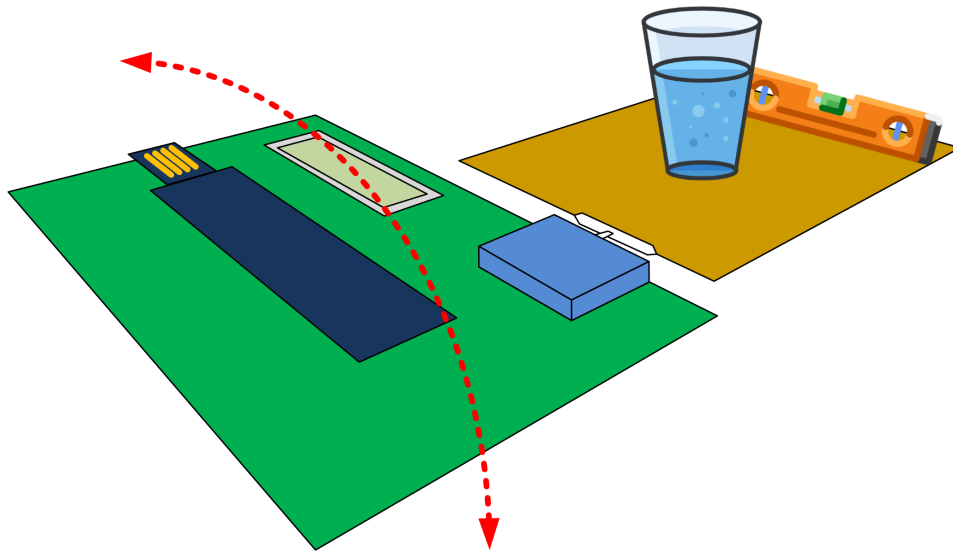


FIGURE 3 – Schematic representation of the 1-D gimbal system. The brown platform should remain horizontal, even if the PSoC is rotated.

The gimbal system will have two modes. In the first mode (the testing mode), the horizontal platform can be rotated to the left or to the right (this is mainly for testing purposes). In the second mode (the gimbal mode), the gimbal functionality should be activated, and the platform should remain horizontal.

The inputs and outputs should be configured as follows.

- **SW1** : When pressing SW1, we should switch between testing mode and gimbal mode. This should also be reflected with the LEDs.
- **LED1-LED4** : When in testing mode, the LEDs should turn off. In gimbal mode, the LEDs should be on.
- **Pot** : When in testing mode, turning the potentiometer to the left should rotate the platform to the left, turning the potentiometer to the right should rotate the platform to the right.
- **Keyboard** : Pressint the “1” should switch between testing mode and gimball mode. When in testing mode, pressing “2” should rotate the platform to the left, pressing “3” should rotate the platform to the right.
- **Audio out** : The audio output should generate a sine tone whose frequency is proportional to the tilting of the platform.

- **LCD screen :** The LCD screen should print the value that is captured with the accelerometer (and tentatively convert it to orientation).
- **ADC :** the ADC will be used to measure the acceleration from the ADXL335Z evaluation board.
- **Serial port :** the PSoC should send the same data to the computer as is sent to the LCD screen. On the other hand, the user should be able to send the same commands as the keyboard from the computer to the PSoC.

4 Flipper-Mania

The aim of this project is build a small flipper. This can be played by pressing the switch to activate the servomotor to propel the ball. The flipper itself is a simple inclined piece of cardboard or 3D printed stand. You may required additional equipment for this project.

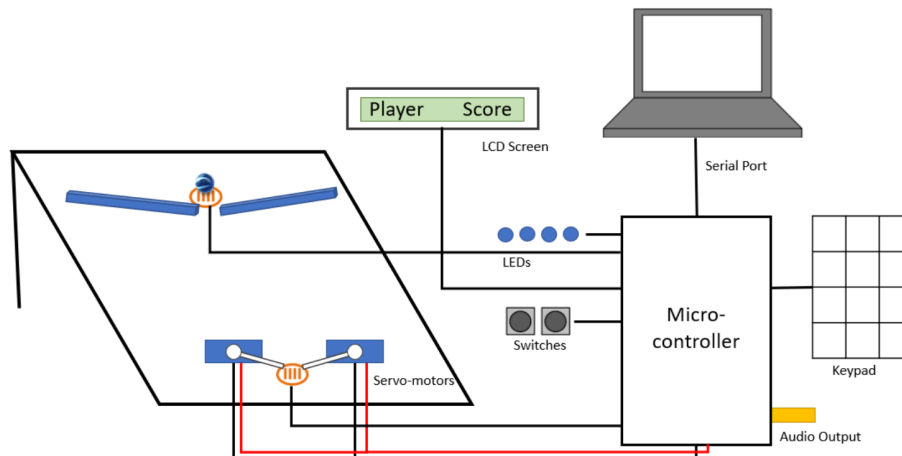


FIGURE 4 – Schematic representation of the small flipper.

The inputs and outputs should be configured as follows.

- **SW1** : When pressing the SW1, the left servo should be activated.
- **SW4** : When pressing the SW4, the right servo should be activated.
- **SW3** : When pressing SW3, the score should be reset to zero.
- **LED1-LED4** : Turn the LEDs on when the game ends.
- **Photoresistor** : The photoresistors detect and start or end the game according to the position of the ball (if the sensibility of the photoresistor allows it).
- **Keyboard** : You can pick several keys to assign the different action of the flipper (left and right servo command, start the game, reset the score ...).
- **Serial port** : The computer user can enter his name through the serial port.
- **LCD screen** : : The LCD screen should first print the name of the player then the score of the player.
- **Audio out** : The audio output should emit the music of the Flipper.