Universidad Carlos III de Madrid

Laboratory report

Microprocessor based digital systems

Group 16 :
Members:

-Yousri Aboudaoud

-LUCIA BARRANCO MORENO

-MARIA FERNANDA MONTIEL ZAVALA

Summary

Block diagram

Hereafter the block diagram of the whole project:

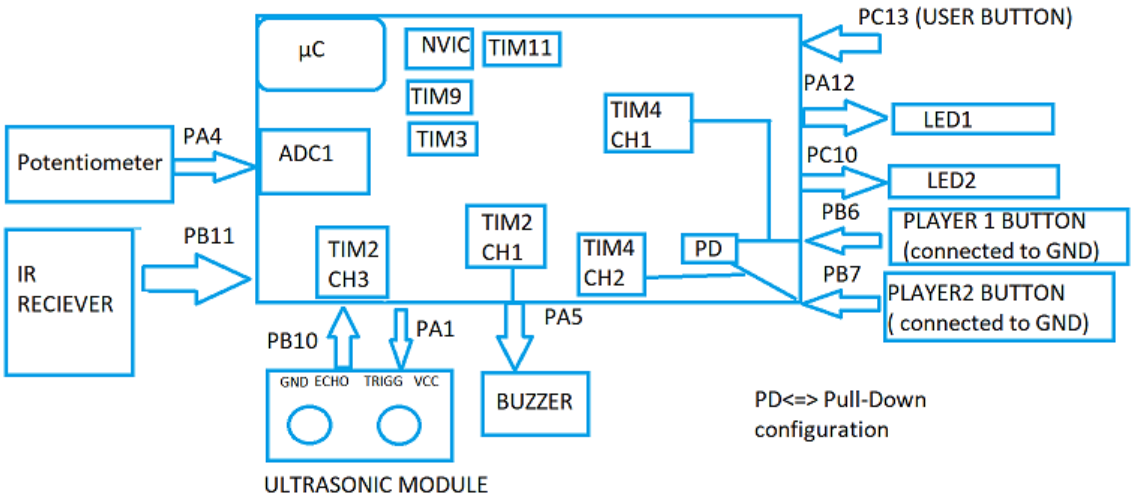

Figure 1 Block diagram, game of reflexes + improvements

Schematics and description of connections, showing the board used, and the components

The following schematic describes the connections between the board and the external components (in this case the **Nucleo board** is represented by the **Arduino Uno** board).

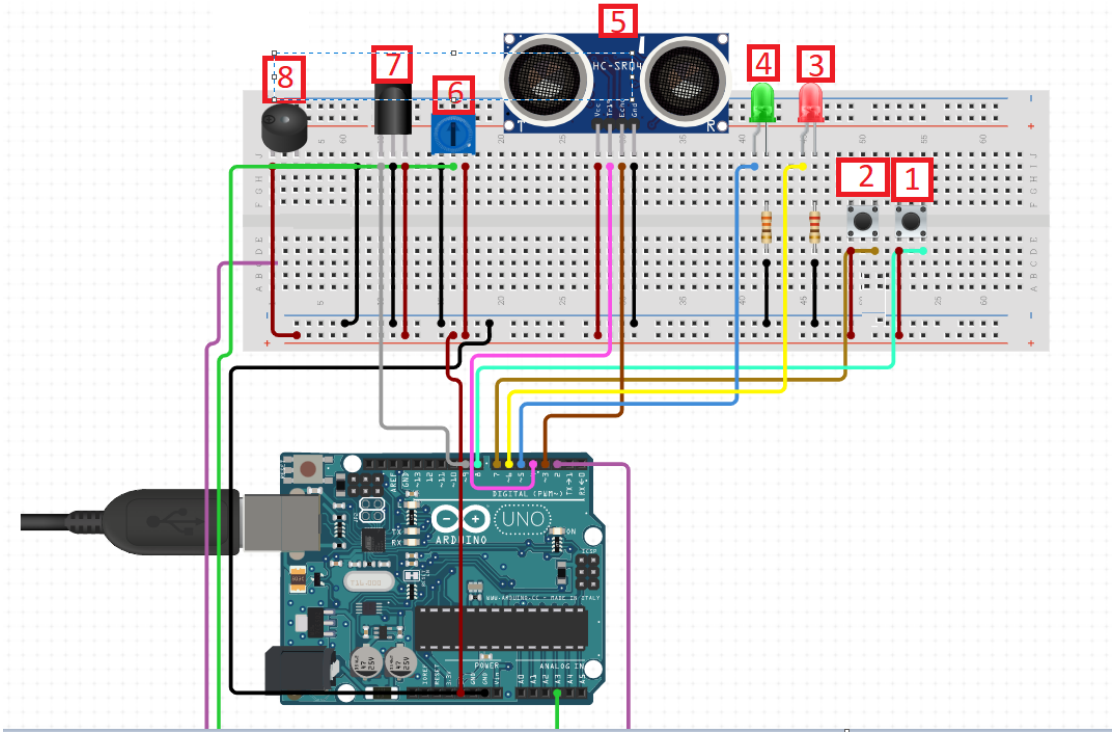

Figure 2 Whole schematic of the project

From figure 2 we can notice that:

- Components 1 & 2 are the push buttons of the players
- Components 3 & 4 are the LEDs associated with each player
- Component 5 is the HC-SR04 Ultrasonic Sensor Module (added as an improvement)
- Component 6 is the PT-15, 15 mm Carbon Potentiometer; used in order to change the difficulty of the game N°2
- Component 7 is the AX-1838HS is miniaturized infrared receivers for remote control (added as an improvement, used for controlling game number 3 another improvement)
- Component 8 is the buzzer employed to generate melodies (this is possible due to the properties of the piezoelectric material composing the buzzer) depending on the outcome of the game

Brief explanation about the microcontroller peripheral used and their basic setup & which IRQ have been used and the functionality achieved by their ISRs

This peripherals has been as follows, we shall describe each use of the peripherals by first stating the pin used and thereafter its basic configuration, and while describing the way the peripherals associated with it achieve the functionality:

- PC13: digital input, used as the USER button to change the game, with interrupt to change the game each time press it
- PA12 & PC10: digital output, used for turning on and off the external LEDs of each player
- PB6: alternate function, pull-up configuration, connected to the channel 1 of TIM4, connected to the push button of the player 1. TIM4 configured as TIC with interrupts to count the time it has taken the player 1 to press his/her button
- PB7: alternate function, pull-up configuration, connected to the channel 2 of TIM4, connected to the push button of the player 2. TIM4 configured as TIC with interrupts to count the time it has taken the player 2 to press his/her button
- PA5: alternate function, connected to the channel 1 of TIM2, the latter is configured as a TOC in toggle configuration (hardware output enabled); combined with the channel 2 of TIM3 (the latter is used to count the duration of each melody), TIM2 generates a square signal and subsequently generates melody
- PA1: digital output, used with a timer to generate the signal Trigger (the timer (in this case TIM2 channel 2) counts 100ms, and then count 10µs (one channel has been used to count two quantities of time) during which the PA1 pin is set to one)
- PB10: Alternate function, connected to the channel 3 of TIM2, the latter is configured as a TIC with an interrupt configuration to both edges (the rising edge is the start time of the signal echo, the falling is the end of the signal; by storing the two times we can compute the width of the signal and thus the distance using formula of distance=speed*time; speed being that of the sound)
- PB11: digital input, with interrupts, to read the data of the IR receiver encoded in NEC protocol, used along the TIMER11 to duration of the data sent (since the NEC protocol send information by packets of 8bits (cf. Figure 3)

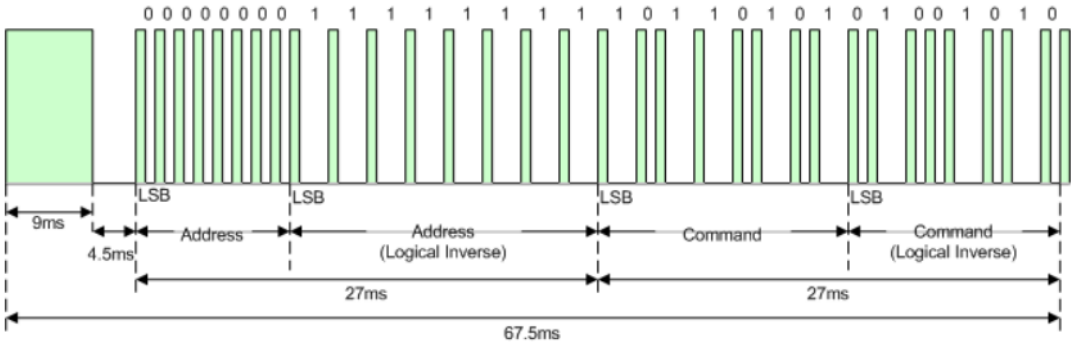

Figure 3 The NEC protocol, each time the user presses a button of the remote control the output signal could be visualised using the oscilloscope as shown above, after 9ms the signal starts and transmits the address and the command (and their respective inverse, to ensure the goodness of transmission)

- PA4: analog pin, connected with the ADC1, convert the analog signal provided by the potentiometer, with a resolution of 12 bits, without conversion and the scan disabled (one sequence)
- PA3: Alternate Function (USART2_RX) PC connection through USB cable, used to select the game by pressing 1,2 or 3 on the keyboard of the PC, used with HAL_UART_RxCpltCallback in order to handle the interrupts associated with this peripheral. Used also to generate software reset whenever the player presses ‘X’
- PA2: Alternate Function (USART2_TX) PC connection through USB cable, used in order to print messages on the screen of the PC
- TIM9 has been used for the delays needed in the program (employed in a function “delay”)

Flowcharts of the ISRs developed

A quick description of each flow chart has been provided in the legend of each illustration.

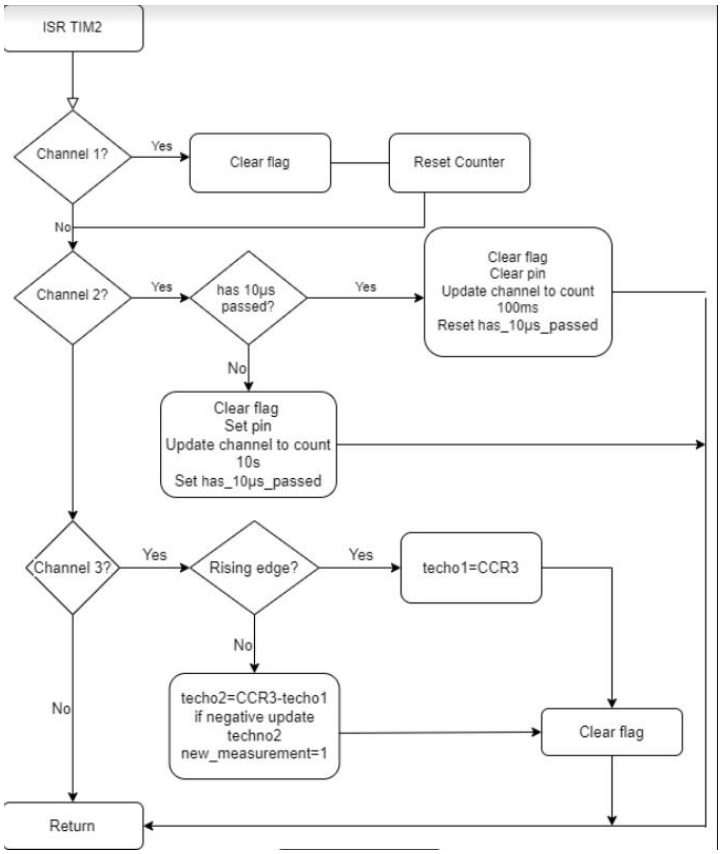

Figure 4 ISR of timer 2 with two external components, channel 1 is associated with the buzzer to generate melodies, channel 2 and 3 are used for the ultrasonic module, the former is used to manage the generation of the trigger signal; the latter (channel 3) is used to measure the width of the signal echo (a measurement used to compute the distance

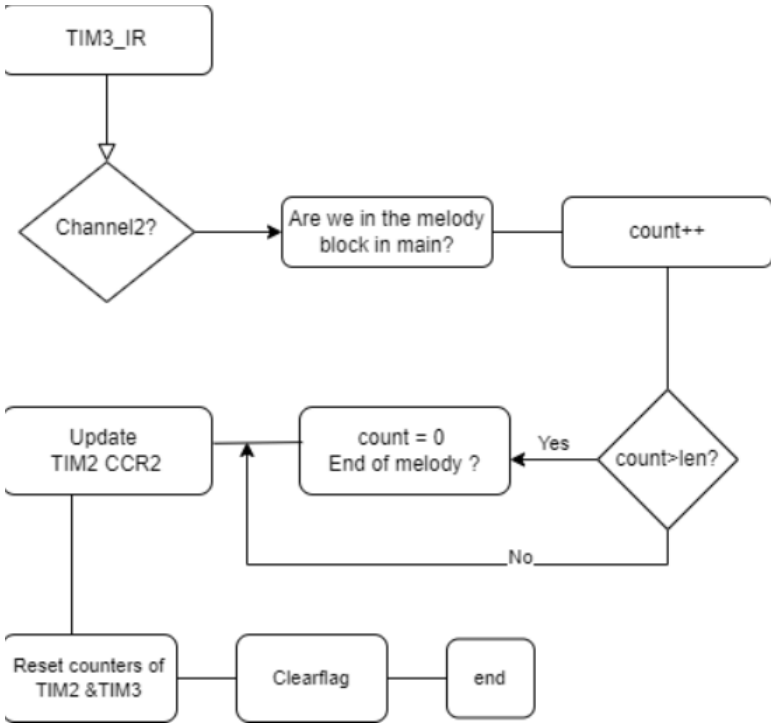

Figure 5 ISR of TIM3, we check which channel had risen a flag, and we verify if we are in the melody block in main (this condition was added in the case we used the channels of the timer for other purposes), after that we proceed to the algorithm that will configure the timer to generate the melodies

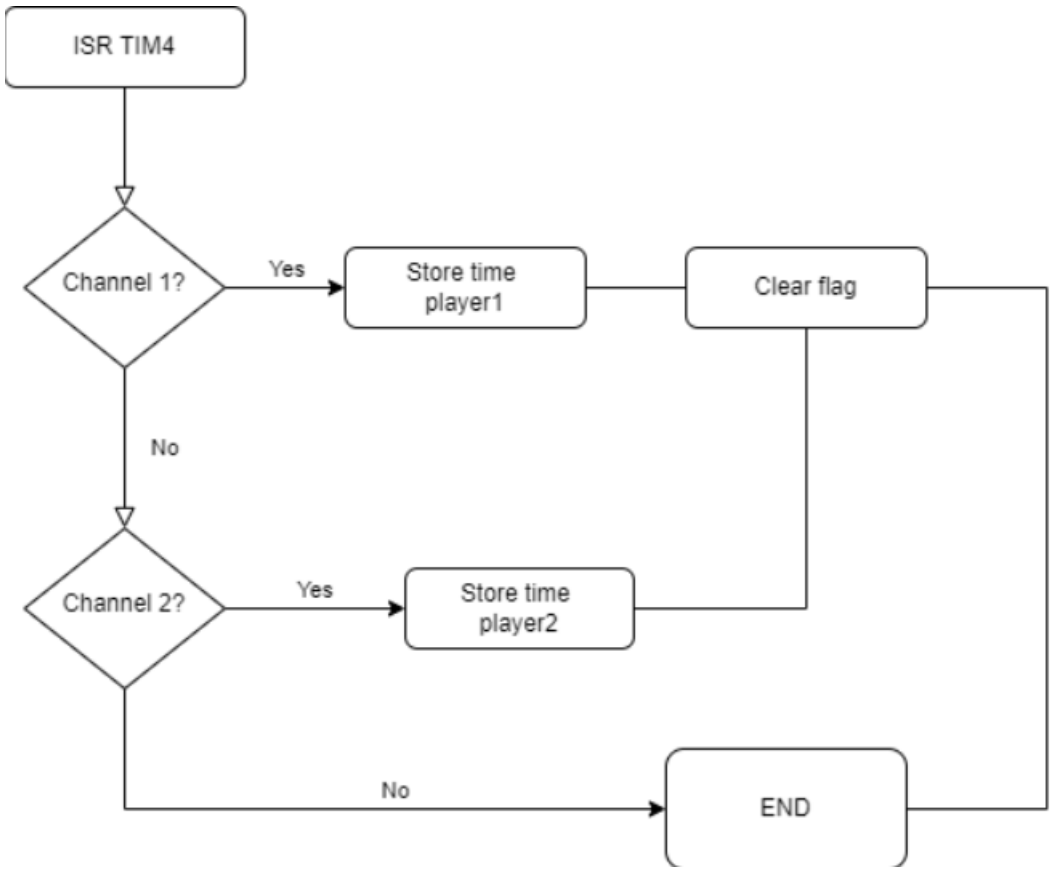

Figure 6 ISR of TIM4, we check first which channel has triggered the interrupt, and then we store the time of each player in a global variable

Conclusions

All task at hand has been completed, during the project we faced multiple problems, which allowed us to think “out of the box” taking each problem from multiple perspectives, subsequently solving all the problems, and in the same time improving ourselves and our skills.