

F1 Reaction game

Introduction

What is the project?

The project represents Formula 1 reaction game that requires 2 players. It contains a series of leds (each player has 3 leds), each one with a corresponding button. Players must press the correct button as soon as its led lights up to earn points. After a certain time, the winner will be the player with the most points.

Functionalities of the game:

- **Random selection of the leds**
- **Input detection** (Button pressing)
- **Scoring** (points for each button pressed correctly)
- **Global timer** (the players have 3 selections for the timer: 15 seconds, 30 seconds, 60 seconds)
- **Statistics** (it will be calculate the median time for each player to press the button after the game)
- **Theme song** (there will be a buzzer that sings)

What is the purpose of the project?

For me, the purpose of this project is to create a real-world device from nothing. This project is interesting because it helps me understand better how the hardware components and the software connects.

What was the idea that got you started?

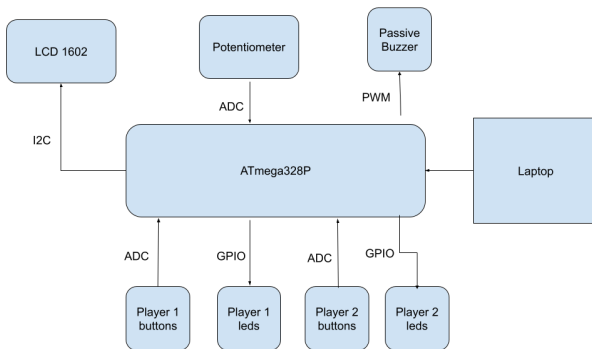
So I am a big fan of Formula 1 and I thought what i can do about this topic. I remembered that I played a version of the game that I want to implement in a museum, but it was in singleplayer mode. I thought it will be fun to have a version of the game that was multiplayer.

Why do you think this project is useful for others and for you?

Just like a Formula 1 driver reacting to track signals, this game challenges you to press the right button the second the led is up (which will improve your reflexes). In addition, it creates a lot of fun with your friends because of the multiplayer mode.

General Description

The block diagram for this project is:



How the game will be played?

1. First the players will choose the time of the game. (there will be 3 options: 15 seconds, 30 seconds, 60 seconds)
2. The game waits for the players to write their name and send it to the microcontroller via uart.
3. For each player, one random led will light up, and the player need to pressed the correct button for the led to go out. At this process will be repeated as long as the timer is bigger than zero. For each correct button pressed the player will get one point.
4. After the time runs out, the winner will be the player with the most points.
5. After the winner is being written on the lcd, the microcontroller will send via uart the average reaction time for each player.

While the game is being played, the theme song from Formula 1 will play in the background.

Hardware Design

The list of components:

| Components | Number of pieces | Protocol |
|---------------|------------------------|----------|
| Atmega328P | 1 | USB |
| Potentiometer | 1 | ADC |
| LCD 1602 2x16 | 1 | I2C |
| Leds | 6 (3 for each player) | GPIO |
| Buttons | 6 (3 for each player) | ADC |
| Buzzer | 1 | PWM |

The detailed structure of the components (their role and how they are connected to the board):

1. ATmega328P

- This is the center of the project.
- Flash memory: 32 Kib
- RAM memory: 2 Kib
- Voltage: 3.3V - 5V

2. Potentiometer

- Role: Helps to choose the time of the game
- Has 3 pins that needs to be connected to work. We will controlled the values from him with adc.

| PIN | Connected |
|-----|-----------|
| 1 | VCC (5V) |
| 2 | PC2 (A2) |
| 3 | GND |

I connected the second pin to the PC2, because I need to read the analog values that the potentiometer gives me as input. That values I am gonna place it in intervals to design the time of the game.

3. LCD 1602 2×16

- Role: To show the score of the game, how much time is left and the winner of the game.
- The microcontroller will communicate with the lcd using the i2c protocol. I am using a module for consuming less pins on the microncontroller (will use 4 pins)

| PIN | Connected |
|---------|-------------|
| 1 (VCC) | VCC (board) |
| 2 (GND) | GND (board) |
| 3 (SCL) | SCL (PC5) |
| 4 (SDA) | SDA (PC4) |

4. Leds

- Role: For each player, there will be one led turn on randomly that needs to be turned off. (for scoring points)
- Each led will be connected to a resistor (330 ohmi) and a pin on the microcontroller
- Each player will have 3 leds.

| LED | PIN |
|-----|-----|
| 1 | PD7 |
| 2 | PD6 |
| 3 | PD5 |
| 4 | PD4 |
| 5 | PD2 |
| 6 | PD1 |

I choose this pins on the board, because I need digital pins for the leds.

5. Buttons

- Role: To turn off the corresponding led. Each led will have a corresponding button.
- Each player will have 3 buttons. Normally, a button should use one pin, but this will consume a lot of pins of the board. My solution is to connect 3 buttons on one pin and read the values using adc for determined which button is being pressed.
- For this part I need to create a voltage divider. Each button will have his own resistor.

This is the electrical scheme for how the buttons and the resistors is being connected.



I will need a pull-down resistor for the input to read 0 when no button is being pressed.

For pins I will use 2 analog pins (PC0 for player 1 and PC1 for player 2).

6. Buzzer

- Role: To sing the F1 theme song.
- It will be controlled using PWM.
- It will be connected to the PD3 pin, because the OC2B is on that pin on the atmega328p microcontroller.



This is the electric scheme of my project, that represent the connections that I detailed in the structure of the components.

Software Design

Link repository: [Repository](#)

The element that is new to this game is the multiplayer mode. Initially the game was a training method for the formula 1 drivers and of course was singleplayer. Because it will be a multiplayer game, will be fun to play with friends and train your reflexes.

For this project, I am gonna use things that I learned from this laboratories:

- **Laboratory 0: GPIO:** for leds
- **Laboratory 1: UART:** for communicating with the players (asking their names and send the statistics for each player at the end of the game)
- **Laboratory 2: Interrupts, Timers:** for implementing debounce and external interrupts for buttons. In addition, implementing timers for counting the seconds.
- **Laboratory 3: PWM:** for the buzzer to sing the formula 1 theme song
- **Laboratory 4: ADC:** for the buttons for each player (along with an interrupt) and for the potentiometer
- **Laboratory 6: I2C:** for showing information about the score, time on the lcd

The structure of my project is:

```
Project
| - i2c.c
| - i2c.h
| - adc.c
| - adc.h
| - timers.c
| - timers.h
| - usart.c
| - usart.h
| - utils.c
| - utils.h
| - main.c
```

I choose this structure for organizing (each protocol and feature will have a file). This is easier for me to read the code and debug.

For this project I am gonna use:

- **A state machine:** I am gonna have 3 states (start of the game, middle of the game, end of the game). I will use this concept for dividing my code into 3 parts (I will have interrupts that need to be working just in one state of the game).
- No external library

Algorithms and data structures used:

- **States of the game:** this is an enum
- **UART Protocol:** For finding out the name of the players (and send it to the lcd) and communicating their average time for the game
- **I2C Protocol:** For showing the score, time, names on the lcd

Rezultate Obținute

Concluzii

Download

Jurnal

3 May 2026: Adding the introduction and the general description.

4 May 2026: Adding the block diagram.

11 May 2026: Complete the hardware design.

16 May 2026: Explaining the features of the game.

27 May 2026: Finish the code for the project.

Bibliografie/Resurse

[Export to PDF](#)

From: <http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link: http://ocw.cs.pub.ro/courses/pm/prj2026/theodor_ioan.buliga/alexandru.ocanoaia 

Last update: **2026/05/27 06:12**