2021

MICROPROCESSORS PROJECT: MENTAL SPEED

GILAD KANGISSER - 2367017

UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

I DECLARE THAT FOLLOWING DOCUMENT IS MY OWN WORK.

School of Electrical and Information Engineering

University of the Witwatersrand, Johannesburg

ELEN 2021 – Microprocessors

Gilad Kangisser – 2367017

Date: 18 May 2021

1. Circuit Diagram and Photograph of the game

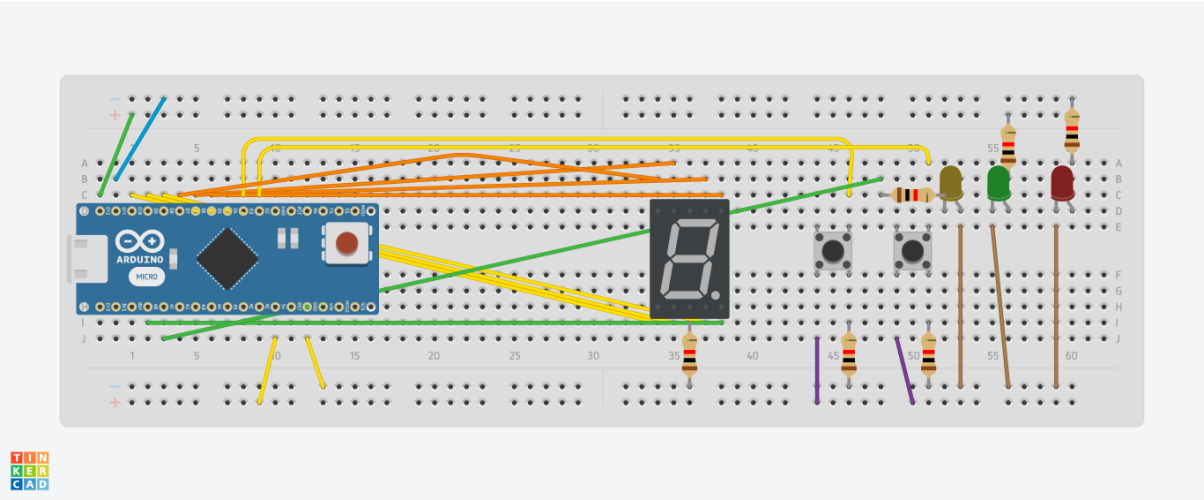
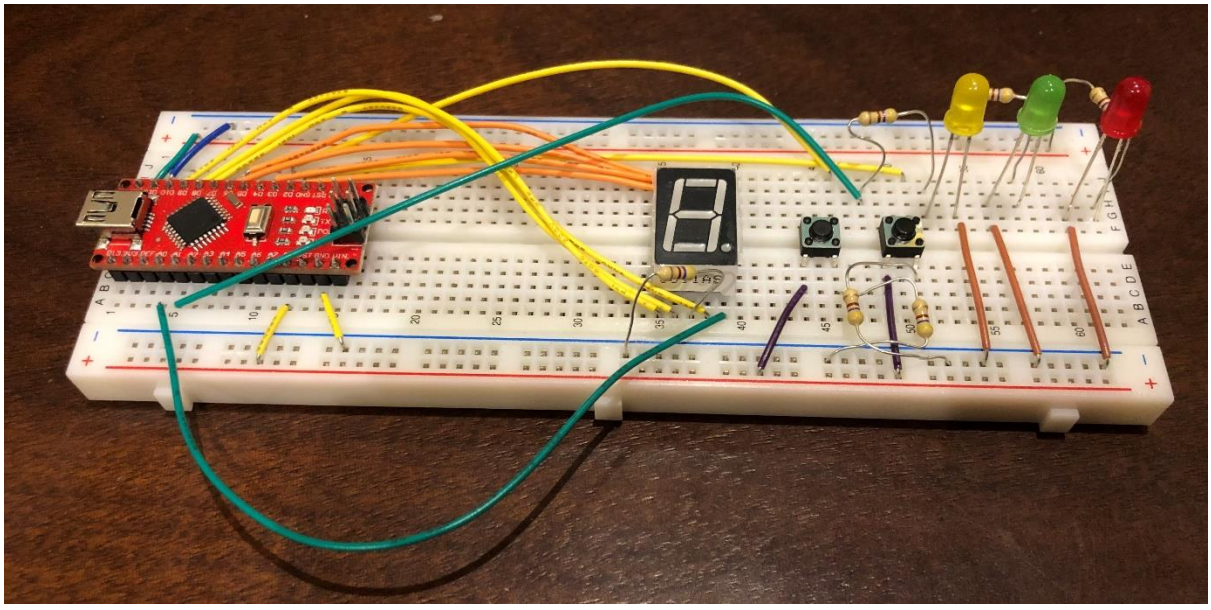


Diagram of circuit representing the game.



Photograph of Actual Game.

2. Equipment Used

Equipment	Use in Project
1 Arduino Nano	To control the logic of the game in order to process the inputs and outputs necessary**.**
1 Large Breadboard	To connect the relevant circuit components to each other as well with the microprocessor.
1 7-digit Display	To display the current number being shown during the game as well as the results, and setup choices.
2 Pushbuttons	For the player to select a divisor, select a speed, skip a number and select a number during the game.
6 Resistors	To connect the circuit components to avoid circuit malfunctions such as short circuiting.
1 Green LED	To communicate to the player that they have chosen the correct answer and have scored a point. It also Is used in together with the red LED to represent that the result is being displayed.
1 Red LED	To communicate to the player that they have chosen the incorrect answer and have been deducted 2 points. It also Is used in together with the green LED to represent that the result is being displayed.
1 Yellow LED	To distract the player by flashing at pseudo-random instances.
21 Assorted Wire Connectors	To connect the different circuit elements to each other, such as, a push-button connected to a microprocessor port.

3. Functionality and Instructions

- ### The Basic Game:

  In order to begin the game, the user, or player, can push the first pushbutton (the right most button on the breadboard). Following an initial 3 second start time, the first number is displayed, if the number is divisible by 3, the player is able to use the push button in order to select the total to be an answer. If the user is correct, the Green LED is lit to tell the user that the answer is correct and that their score has increased by 1. If the player is incorrect, the Red LED is lit and thus the players score is decreased by 2, or to 0 - whichever is higher. 60 seconds after the game began, the game will end and the display will loop, stopping on each state for 2 seconds, the cycle includes, a blank display, followed by the first, then second digit, if applicable. During this time, the user, is able to push the button to start a new game.

- ### Speeding up the game:

  If a player wants to control the speed of their game, this option has also been provided, the second button (the left button) enables this customization. If the pushbutton is pressed before the game has started, the display will show a number, beginning with 1, and by continuingly pressing the button will increment the displayed number. The current number represents the time interval that each number will be shown for. This enables the user to play a faster paced game by choosing an interval of 1-4 seconds. Furthermore, when an instance of the game ends, the player can once again press the second push-button in order to choose a new speed for their next game. The default of this is 3 seconds as described in the project brief.

- Changing the Divisor:

  If a player wants to choose a different divisor, other than the default 3, the second button (the left button) enables this option, using a similar method to changing the speed of the game. After one has selected the speed of the game, the left pushbutton can once again be pressed, the display will show a number, beginning with 2, and by continuingly pressing the button will increment the displayed number. The current number represents the divisor that will be used during the game. Therefore, when the player pushes the first/right button to begin the game, whatever number was last displayed will be the divisor for that specific game. This enables the user to play using a divisor of 2-9. Furthermore, when an instance of the game ends, the player can once again choose a new divisor for their next game.

- Next Number:

  The second button is also used by the player in order to skip to the next number. During the game, if the player pushes the button, either after selecting the current number or not, the next number will display, and the player can then answer the new number. This allows for less wait time and a higher possible score to achieve in the 60 seconds.

- Light Distraction:

  In order to add an extra element of difficulty to the game, a third, yellow, LED has been connected in order to function as a distraction to the player. The yellow LED has been set to shine at semi-random intervals as to confuse the player. The function that controls this, does so by lighting when the current 'sum' is 1 less than a prime number – so that it may or may not - be a multiple, and will therefore, confuse the player.

4. Code Implementation and Discussion

Pin	Use
PortD Pin 2	Used for INT0, to act as the input to register button pushes.
PortD Pin 3	Used for INT1, to act as the input to register button pushes.
PortD Pins (4-7)	Used to output to the top half of the 7-digit display.
PortB Pins (0-2)	Used to output to the bottom half of the 7-digit display.
PortB Pin 3	Used to turn on/off the red LED.
PortB Pin 4	Used to turn on/off the green LED.
PortA Pin 0	Used to control the decimal point on the 7-digit display.
PortA Pin 1	Used to turn on/off the yellow LED.

- Code Structure
  a) At the top of the program, aliases for binary numbers that will be used in the game, indirect addresses for registers and addresses for the interrupt vector table (0x00, 0x02, 0x04, 0x16) are included.
  b) The variables and interrupts that were needed before the game started were initialized, as well as the code needed to setup variables once the game was started.
  c) The rest of the methods can be categorized in this order: change divisor, change speed, first pushbutton interrupt, second pushbutton interrupt, check divisibility, skip number, end game and displaying score/numbers.
  d) Some problems were faced, such as button debouncing causing double clicks, displaying the results as instructed, and the need to manipulate the timers in order to ensure a 60 second game. However, all of these issues were resolved as can be seen in the code.

5. References and Acknowledgements

1. Tinkercad | From mind to design in minutes (2021). Available at: https://www.tinkercad.com/dashboard (Accessed: 18 May 2021).
2. Debouncing via RC Filter - Digilent Reference (2021). Available at: https://reference.digilentinc.com/learn/microprocessor/tutorials/debouncing-via-rc-filter/start (Accessed: 24 May 2021).
3. TCNT1 = 0 with inline assembly (2008). Available at: https://www.avrfreaks.net/forum/tcnt1-0-inline-assembly (Accessed: 24 May 2021).
4. Arduino, U. and Questions, P. (2016) Accessing TCNT1H, Arduino Forum. Available at: https://forum.arduino.cc/t/accessing-tcnt1h/368811/4 (Accessed: 24 May 2021).
5. (2021) Ww1.microchip.com. Available at: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf (Accessed: 24 May 2021).

I acknowledge that I received help from the resources above and from the tutors and lecturers.