Guess the Weight

Guess the Weight is a recreational device that uses a digital scale to create an interactive and fun experience. The game features two distinct modes:

1. Guess the Weight

An object is placed on the scale and weighed. The player must then try to **guess the weight** by inputting a value as close as possible to the actual measurement.

2. Match the Weight

An object is weighed and its weight is stored. The player is then given several smaller objects of known weights. The goal is to **recreate the original weight** by placing a combination of these objects on the scale.

Both modes encourage observation, estimation, and logical thinking while offering a fun and hands-on interaction with real-world measurements.

Introduction

A brief overview of our project:

- The project features two game modes based on weighing an object.
- The goal is to provide a fun and interactive experience for the user.
- The initial idea was to create a functional scale.
- It is useful as it results in a device that can be used for entertainment purposes.

Descriere generală

The “Guess The Weight” system consists of the following main modules:

- **Weight Sensor Module (Load Cell + HX711)** – measures the weight of the object placed on the scale.
- **Microcontroller (Arduino Mega 2560)** – manages the game logic and communicates with all

other components.
- **Display Module (OLED or alternative)** – shows instructions, game mode, weight, and messages for the user.
- **4×3 Keypad** – allows the user to interact with the system and input values.
- **Passive Buzzer** – provides audio feedback during the game (e.g., success/failure sounds).
- **Power Supply (via USB from a laptop)** – powers the entire system through the Arduino board.

Interaction Description:

- The user selects a game mode using the 4×3 keypad.
- The system prompts the user with instructions on the OLED display.
- The user places an object on the load cell.
- The HX711 module sends the weight data to the Arduino via digital communication.
- Arduino processes the data and checks if the condition for the current game mode is met.
- The display and buzzer have different actions depending of the game mode selected.

Hardware Design

# Components List

Component Name	Link	Description
—————	——	————-
Arduino MEGA 2560	[OptimusDigital](https://www.optimusdigital.ro/ro/compatibile-cu-arduino-mega/471-placa-de-dezvoltare-compatibila-cu-arduino-mega-2560-atmega2560-ch340.html)	Development board compatible with Arduino MEGA 2560 (ATmega2560 + CH340)
Passive Buzzer	[OptimusDigital](https://www.optimusdigital.ro/ro/audio-buzzere/634-buzzer-pasiv-de-5-v.html)	5V Passive Buzzer
OLED Display Module	[eMAG](https://www.emag.ro/modul-display-oled-heltec-automation-128x64-interfata-iic-0-96-albastru-galben-zc-01-d/pd/D74PRM3BM/)	Heltec Automation, 128×64, IIC interface, 0.96” blue & yellow
Keyboard Module	[OptimusDigital](https://www.optimusdigital.ro/ro/altele/5825-modul-tastatura.html)	Keyboard module
USB Cable	[OptimusDigital](https://www.optimusdigital.ro/ro/cabluri-cabluri-usb/7313-cablu-albastru-usb-am-la-bm-50-cm-pentru-arduino-mega-i-uno.html)	Blue USB AM to BM 50cm cable for Arduino MEGA and UNO
Weight Sensor	[RoboFun](https://www.robofun.ro/modul-senzor-greutate-1kg-pentru-cantar-electronic.html)	1kg Weight Sensor Module for Electronic Scale
HX711 Module	[OptimusDigital](https://www.optimusdigital.ro/en/others/130-hx711-instrumentation-module.html)	HX711 Instrumentation Module
Breadboard Kit	[OptimusDigital](https://www.optimusdigital.ro/en/kits/2222-breadboard-kit-hq-830-p.html)	Breadboard Kit HQ 830 points

Hardware Connections

1. HX711 + Load Cell (weight sensor)

'`Interface:`' Digital serial communication (2 wires - DT and SCK)

'`Connections:`' * VCC → 5V Arduino * GND → GND Arduino * DT → D2 (digital input) * SCK → D3 (digital output)

'`Voltage:`' 5V logic `Note:` No level shifting needed; both HX711 and Mega operate at 5V logic

2. Passive Buzzer 5V

'`Interface:`' PWM or simple digital ON/OFF signal

'`Connections:`' * Pin + (red) → D9 (PWM digital pin) * Pin - (black) → GND

'`Voltage:`' 5V `Note:` Directly compatible with Arduino Mega digital outputs

3. OLED Display 0.96", I2C

'`Interface:`' I2C (2 wires)

'`Connections:`' * VCC → 5V * GND → GND * SCL → Pin 21 (SCL on Mega) * SDA → Pin 20 (SDA on Mega)

'`Voltage:`' 3.3V–5V `Note:` No conversion needed; internal protection resistors present

4. Matrix Keypad 4x4

'`Interface:`' Digital, column and row scanning (8 pins)

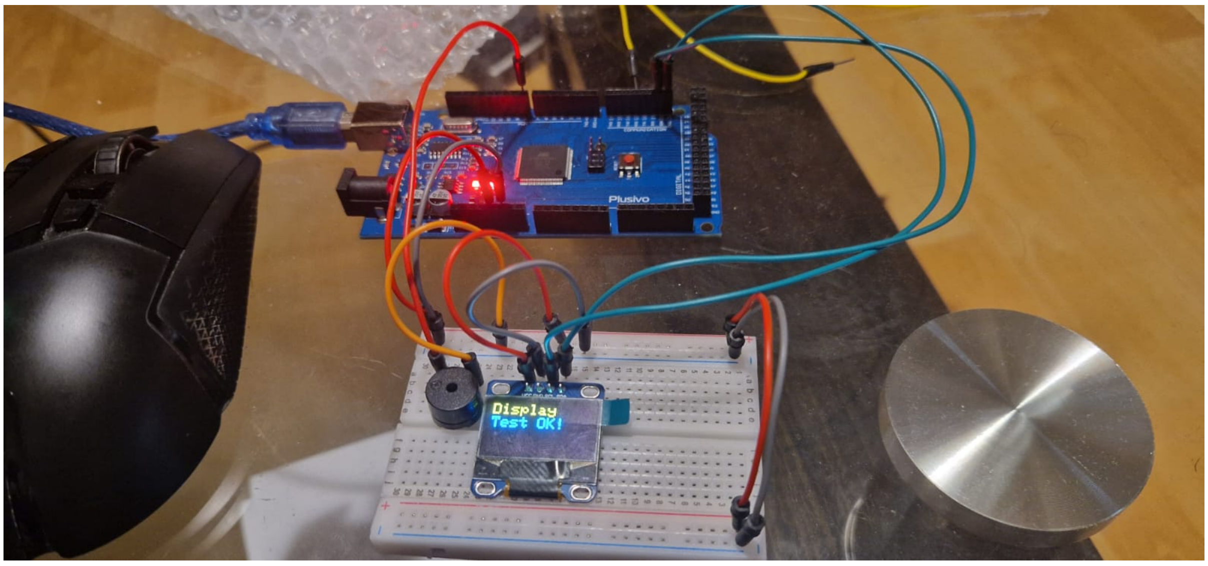
'`Connections:`' * 8 wires connected to digital pins D22–D29

'`Voltage:`' 5V `Note:`' Passive mechanical switches using internal pull-up resistors

5. Breadboard + Power Distribution

'`Powers:`' * OLED Display * Buzzer * HX711

'`Power Source:`' USB AM-BM connected to Arduino Mega `Note:`' 5V and GND lines distributed on both sides of breadboard



Both the display and the buzzer works, the dislpay can be seen in the picture, as for the buzzer, it was tested using a script.

Software Design

Development Environment

The firmware was developed using '`Visual Studio Code`' with the '`PlatformIO`' extension. This setup offers efficient dependency management, integrated serial monitor support, and cross-platform compilation.

Libraries and Third-Party Sources

The following third-party libraries were used:

'`Adafruit_SSD1306`' – controls the OLED display via I2C. '`Adafruit_GFX`' – provides graphics primitives for the display. '`HX711`' – interfaces with the load cell and reads weight data. '`Keypad`' – handles input from the 4×3 matrix keypad.

These libraries simplify hardware communication and allow for clean, readable application logic.

Algorithms and Structures Implemented

A simple '`menu system`' navigated using the keypad, allowing selection between two modes: **'`Guess the Weight`' – user estimates an object's weight and inputs the guess via keypad.**

`'Match the Weight'` – user adds objects to match a previously recorded weight. The `'buzzer'` plays dynamic tones in "Match the Weight" mode. The rhythm (tone interval) increases as the current weight gets closer to the target. A `'state-driven structure'` manages transitions between menu, gameplay, and results. The OLED screen provides `'real-time feedback'`, such as instructions, results, and animated weight indicators.

Implemented Functions and Sources

```
drawMenu()
```

– displays the main menu with selection cursor.

```
startGuessTheWeight()
```

– logic for guessing the weight of an object.

```
startMatchTheWeight()
```

– records a target weight, provides animation and rhythm-based buzzer feedback as the player matches it.

```
playMatchingTones(currentWeight, targetWeight)
```

– adjusts buzzer rhythm according to proximity to target.

```
drawWeightAnimation(animY)
```

– animates a simple visual indicator on the OLED screen.

```
playSuccessTone()
```

– plays a tone sequence when the task is completed correctly.

The software combines interaction, sound, and visual elements to create an engaging user experience.

Justification for Using Laboratory Functionalities in the Project

OLED Display and I2C Communication (Lab 6 - I2C)

The OLED screen is used to display menus, game instructions, and feedback to the user. It communicates via the I2C protocol using the Wire library. Functions such as

```
display.begin()
```

,

```
display.clearDisplay()
```

, and

```
display.display()
```

reflect the practical application of Lab 6 concepts.

Buzzer and Tone Generation (Lab 3 - PWM)

The buzzer is used to provide audio feedback in the “Match the Weight” game. The

```
tone()
```

function utilizes the PWM feature to generate audio signals. As the weight approaches the target value, the rhythm speeds up, demonstrating real-time control over frequency generation.

Weight Reading via HX711 (Lab 4 - ADC)

The weight sensor is interfaced using the HX711 ADC module, which reads analog signals from a load cell. This directly implements the analog-to-digital conversion concepts introduced in Lab 4. The function

```
scale.get_units()
```

reads these values in digital form.

Matrix Keypad Input (Lab 0 - Digital Inputs)

The 4×3 matrix keypad allows the user to input commands and navigate the menu. It is scanned to detect key presses by reading digital signals, which aligns with the digital input techniques introduced in Lab 0.

Serial Communication (Lab 1 - USART)

```
Serial.print()
```

and

```
Serial.println()
```

are used for debugging and displaying values such as the current weight. This is a practical use of USART, as explored in Lab 1.

Timing and Delays (Lab 2 / Lab 3)

Delays using

```
delay()
```

are introduced to manage user interaction timing (e.g., allowing the user to see feedback before the next step). These practices relate to timing management topics covered in Labs 2 and 3.

Summary Table

OLED (I2C)	Lab 6
Buzzer (PWM)	Lab 3
HX711 (ADC)	Lab 4
Matrix Keypad	Lab 0
Serial Communication	Lab 1
Menu / State Logic	Lab 2
Timing / Delays	Lab 2 / Lab 3

Project Structure and Feature Interaction

OLED Display (Adafruit SSD1306 128x64)

Shows the main menu, game instructions, user prompts, and results. Provides all visual feedback to the user during setup and gameplay.

HX711 Weight Sensor + Load Cell

Accurately measures the weight of objects placed on the device. Used in both games to set or check the weight value.

4x3 Keypad

Lets the user navigate menus, select game modes, and enter numerical guesses or commands (such as start, confirm, or return to menu).

Buzzer

Outputs audio signals for feedback. Plays different tones to indicate success or to help the user get closer to the target weight in the matching game.

Arduino Board

Central controller that runs the main program. Reads inputs from the keypad and weight sensor, updates the display, and controls the buzzer.

Functionalities Validation

To test the weight sensor, I weighed an object with a kitchen scale and checked if the values matched.

Sensor Calibration

For the weight sensor, I used an object with a known weight (75 grams) and observed the sensor reading (approximately 148,500). Based on this, I calculated the calibration factor as (sensor_value / actual_weight).

Optimization

Used modular functions for clarity and easier maintenance.

Limited display updates to only when changes occur, reducing flicker.

Grouped hardware initialization in setup for faster start.

Provided immediate feedback with the buzzer for better user experience.

Used keypad library debouncing for reliable input.

Calibrated the weight sensor once for consistent measurements.

Rezultate Obţinute

Demo

Concluzii

Download

guesstheweight.zip

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** şi **Resurse Hardware**.

Export to PDF