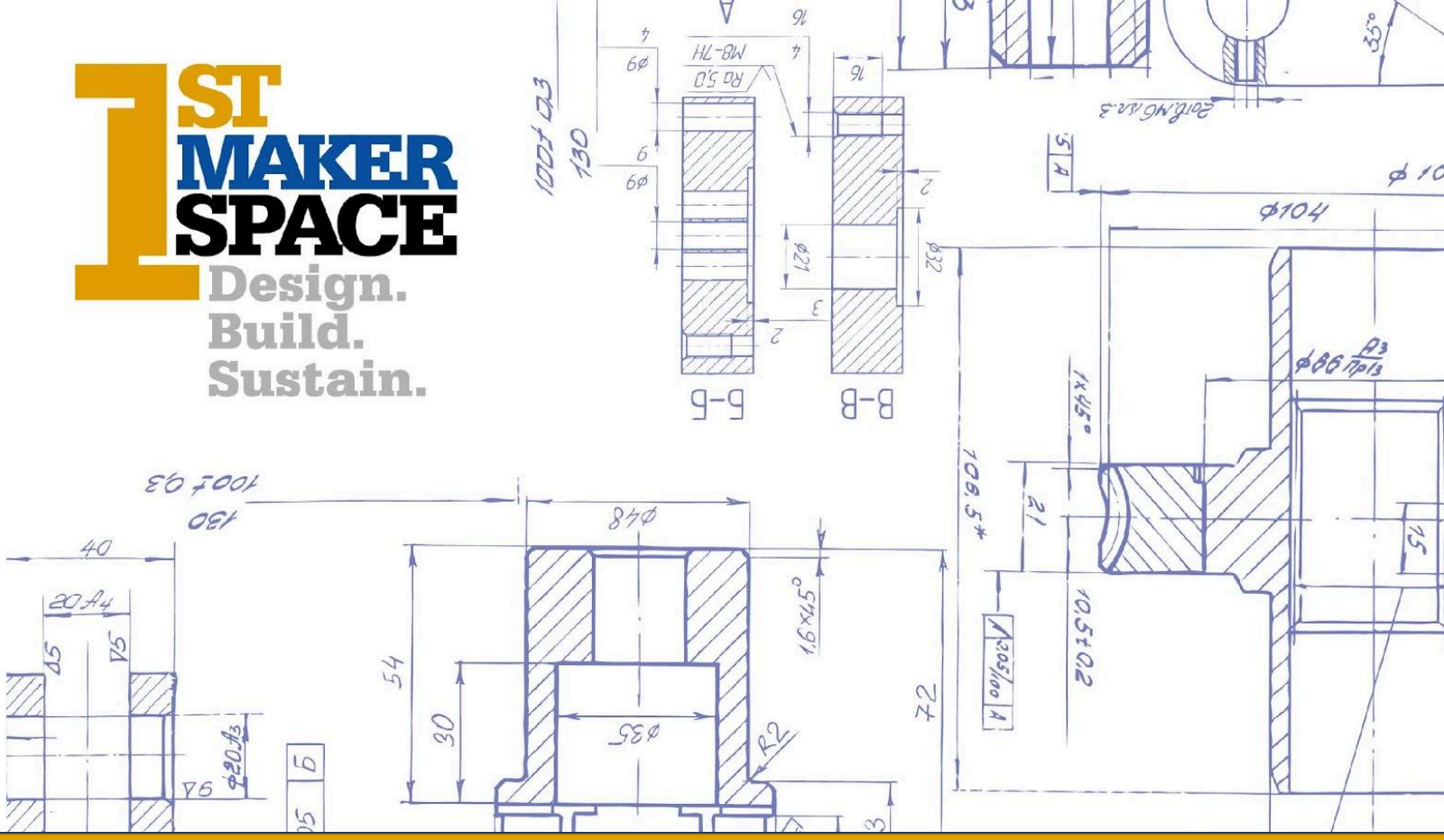
1ST MAKER SPACE
Design.
Build.
Sustain.



Frog Microcontroller Trainer (FMT): Vote

Length of Time: 2 days x 55 Minutes

Recommended for Developmental

Contents

Overview

Objectives

Standards

Materials

Lesson Elements

Resources

© 2024, 1st Maker Space, LLC · https://1stmakerspace.com

Overview:

The Vote program turns the 1st Maker Space Frog (Arduino UNO R4–based board) into a simple thumbs-up / thumbs-down voting machine. Students press one of two buttons to choose YES or NO. The Frog's OLED screen shows a large thumbs-up or thumbs-down image, and the NeoPixel "eyes" change color to match the vote (green for YES, red for NO).

Objectives:

Students will be able to:

Describe how input (buttons) and output (OLED and NeoPixels) work together in an embedded system.

Trace the flow of the Vote program from setup() through loop(), explaining what happens when a button is pressed.

Identify and explain the role of helper functions such as showInstructionsOnce() and showThumbs().

Modify or extend the program to add at least one new feature related to user interaction or display.

© 2024, 1st Maker Space, LLC · https://1stmakerspace.com

Computer Science Teacher Association Standards:

- 3A-AP-16- More advanced students iteratively refine the UI (e.g., adding confirmation screens or a vote counter) based on peer feedback.
- 3A-AP-17- Students decompose the program into logical units (input handling, state management, display updates).

Materials:

- Frog Microcontroller Trainer from 1st Maker Space
- USB Power Cord
- Arduino IDE
- The Vote.ino sketch loaded onto each Frog, along with the 1ST_Maker_Frog library
- PC or Mac
- Chromebooks if using Arduino Create for Education App

Preparation:

- Confirm the Frog boards, library, and Vote sketch compile and upload successfully.
- Run the program yourself and press each button to see the behavior.
- Decide which extensions (e.g., reset button, vote counter, different colors) are appropriate for your students' level.
- Print or distribute the student handouts.

Background Information:

1. Bitmaps and constants

At the top of the program, two large 128×64 pixel images are stored in PROGMEM (program memory). One image is a thumbs-up and the other is a thumbs-down. Storing them in program memory frees up SRAM for variables and makes it possible to draw full-screen images on the small OLED.

2. Tracking the current choice

An enum (enumeration) named Which defines three possible states: NONE, UP, and DOWN. The variable current stores the user's last choice so the program can avoid redrawing the same image over and over when nothing has changed.

3. showInstructionsOnce()

This helper function clears the OLED and prints a short 3-line message:

Vote:
SW1 = Yes
SW2 = No

The text is printed at 2× size so it is easy to read. After displaying the instructions, the function waits 2.5 seconds (delay(2500)) so students have time to read before the main voting UI begins.

4. showThumbs(Which w)

© 2024, 1st Maker Space, LLC · https://1stmakerspace.com

This helper function does the real UI work. It checks the value of w:
• If w == UP, the OLED is cleared and the thumbs-up bitmap is drawn at the top-left corner. Both NeoPixels are set to green using pixel.setPixelColor() and pixel.show().
• If w == DOWN, the OLED is cleared and the thumbs-down bitmap is drawn instead. Both NeoPixels are set to red.
• If w is neither UP nor DOWN, the function does nothing.
Using a separate function like this keeps the code organized and makes it easier to reuse.

5. setup()

The setup() function runs once when the Frog powers up or resets. It:
• Configures BUTTON_ONE and BUTTON_TWO as INPUT_PULLUP, meaning the buttons read LOW when pressed.
• Starts the NeoPixel library, sets the brightness to 64 (out of 255) to keep the lights comfortable, and clears any previous colors.
• Initializes the OLED display. If the display is not found, the program enters an infinite loop that blinks the NeoPixel eyes red as a visible error signal.
• Calls showInstructionsOnce() to show the startup directions to the user.

6. loop()

The loop() function runs repeatedly. Each time through the loop it:
• Reads both buttons and checks if they are PRESSED (active LOW).
• If BUTTON_ONE is pressed and the current choice is not UP, it updates current to UP and calls showThumbs(UP).
• Else if BUTTON_TWO is pressed and the current choice is not DOWN, it updates current to DOWN and calls showThumbs(DOWN).
• Waits 10 milliseconds at the end of the loop. This tiny delay provides simple button debounce and prevents the loop from running too fast.
Because it only reacts when the choice changes, the program avoids unnecessary drawing and flickering.

Key Coding Ideas Highlighted

•Enums for state: Using Which { NONE, UP, DOWN } to manage the current UI state.
•Input handling: Reading digital pins with INPUT_PULLUP and interpreting PRESSED as LOW.
•Output coordination: Matching screen graphics with NeoPixel colors for a clear user interface.
•Error indication: Using blinking red eyes as a visible error message if the OLED fails to initialize.
•Debouncing and efficiency: Using a short delay and only updating the display when the state changes.

© 2024, 1st Maker Space, LLC · https://1stmakerspace.com

Lesson Elements	
Introduction to voting 5-10 minutes	Ask students how groups or apps can quickly collect yes/no opinions. Show the Frog board and explain that it will act as a physical “Thumbs Up / Thumbs Down” voting machine. Invite a quick prediction: What do you think will happen when we press each button?

Access 1MS Code – 15 minutes	Visit the 1st Maker Space website and access the library of code developed specifically for the FMT. Here, you will find a section with various code (called projects) written for each major component of the trainer. The teacher will locate the sketch by clicking on Developmental Lesson 1: Vote.
	
	Here, one can find the sketch to share with students. Students can simply copy the code by selecting the blue copy button.
	
	Project Code:
	
	Copy
	`////////////////////////////////////////////////`
	`// 1.03 - Analog Write`
	
	`byte LED1 = 13;`
	
	`void setup() {`
	`  pinMode(LED1, OUTPUT);`
	`  analogWrite(LED1, 127);`
	`}`
	
	`void loop() {`
	
	`}`
	`////////////////////////////////////////////////`
	
	Have students (individually or in pairs) run the Vote program on their Frog. Ask them to:
	• Press SW1 (BUTTON_ONE) and observe the screen and NeoPixel eyes.
	• Press SW2 (BUTTON_TWO) and observe the changes.
	• Try pressing buttons quickly vs. slowly and notice whether the display flickers or repeats.
	Students record their observations on the handout.

Explain the Program– 15-25 minutes	Bring the class together and show the Vote code on a projector. Walk through the program structure: • Constants and bitmaps. • The Which enum and current variable. • Helper functions: showInstructionsOnce() and showThumbs(). • setup(): what runs once. • loop(): what repeats and how button presses change the state.

© 2024, 1st Maker Space, LLC · https://1stmakerspace.com

	Encourage students to connect each part of the code to what they saw the Frog do.
Extension – 20-30 Minutes	Students choose one or more challenges from the handout to extend the program. Examples include: • Add a way to “reset” the vote to NONE and clear the screen. • Change colors or brightness for accessibility (e.g., color-blind-friendly choices). • Add a simple vote counter using the serial monitor. • Display text labels (YES/NO) alongside or before the thumbs image. Students should test their changes and be prepared to explain what they changed in the code.
Career Exploration: ● A good resource is the IDOE Career Explorer database. ● Another good resource for career information is the Bureau of Labor Statistics	
Practical Application: Programs such as the one in this lesson can be used as a quick way to collect formative feedback.	

Additional Resources:

- 1st Maker Space Frog Microcontroller Library
- 1st Maker Space Learn Arduino with the Frog Microcontroller

Performance Assessment/Check for Understanding :

- Was the student able to successfully load the sketch to the FMT?
- Are students connecting button presses with changes in the program’s state?

Additional Feedback:

© 2024, 1st Maker Space, LLC · https://1stmakerspace.com