ELECTRICAL TEAM TRAINING

TASK 4

Problem 1: Plan-X

About

With the Decepticons gaining ground, the Autobots are assembling a new front-line warrior: an autonomous battle-ready unit codenamed Plan-X.

To bring this mech to life, the team must ensure it can react intelligently to its environment.

Plan-X will use its **ultrasonic** distance sensors to determine whether the path ahead is clear or obstructed, adjusting its **motor speed** accordingly. When the battlefield is empty, it charges full speed ahead, but when obstacles are near, it slows down to navigate carefully.

To top it off, a **PIR** sensor will act as an intruder detector. Once activated, it signals that the enemy is near and triggers a weapon **LED** (think plasma cannon blast) to fire. This **LED** should remain on as long as motion is detected.

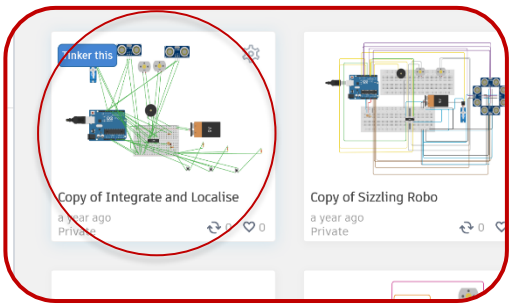
The Autobots need your help. Can you give Plan-X the reflexes it needs to win?

Objectives

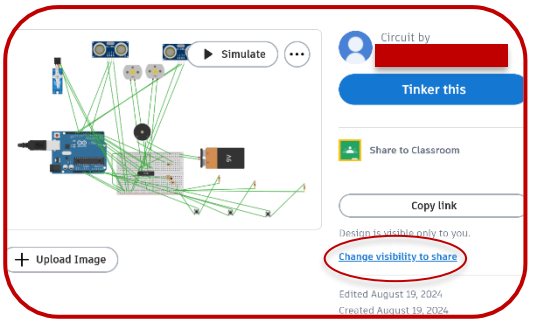
Design a working system that:

- Controls **two DC motors** simulating movement:
  - **High speed** when the **ultrasonic sensor** detects a **large distance** ahead.
  - **Low speed** when the sensor detects a **close object**.
- Using interrupts, detect **motion using a PIR sensor** and turn on an **LED** to simulate a weapon firing when triggered.
- Motor speed must dynamically change based on ultrasonic distance.
- PIR sensor must be wired correctly and trigger the LED only when motion is detected.
- Proper use of analogWrite for speed control and digitalWrite for LED.
- Code must be clear, commented, and modular if possible.
- Clear color-coded wiring.

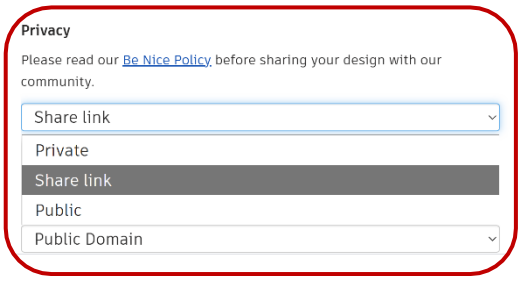
Submission



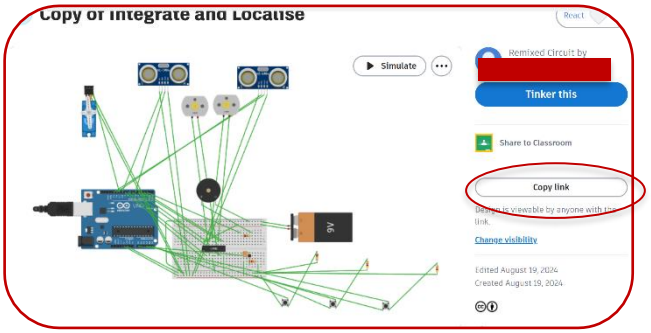
1. After selecting your project



2. Select change visibility



3. Make sure it is set to share link or public and save



4. Submit this link

Problem 2: Autobot Defense System

About

In the ruins of an ancient Autobot facility, deep beneath the surface of Cybertron, lies an Energon relic of immense power. To protect it from Decepticon intruders, the Autobots have left behind a defense module coded into an ancient microcontroller (**STM32F103C8T6**).

Autobot operatives or explorers attempting to access the relic must interact with encrypted Cybertronian runes (**buttons**). Meanwhile, if a Decepticon sabotage attempt is **detected** (trap triggered), the system activates its battle defense mode with flashing **lights** and alarm **buzzers**.

Components

Component	Description
STM32F103C8T6	Brain of the Autobot Defense System
2 Push Buttons	Rune Interfaces (Rune 1, Rune 2)
1 Push Button (EXTI)	Trap trigger (Decepticon attack simulation)
2 LEDs	Visual clues (Autobot torch + clue marker)
1 Buzzer	Defense alarm (Cybertronian siren)

Objective

1. Rune 1 - PA1
   - When pressed, **turns ON LED1 (PC13)** permanently.
   - Represents the discovery of a **clue**.
2. Rune 2 - PA2:
   - Every press **toggle LED2 (PB12)** ON and OFF
   - Represents a **torchlight** used for navigation.
3. Trap Trigger - PA0 (EXTI0):
   - Simulates a **Decepticon trap activation**.
   - Connected to an **external interrupt.**

In ISR (Interrupt Service Routine):

- Set a global flag: trap_triggered = 1
- Turn ON the buzzer (PB13) for alarm

In main loop:

- If trap_triggered == 1:
  - Rapidly **blink LED1** to simulate alert mode
  - **Buzzer stays ON** as a defense warning
  - System can only be silenced/reset manually

Debouncing:

Add a small **delay after each button press** to mitigate **mechanical bounce noise** and ensure smooth triggering.

Bonus

Memory Mapping & Base Address in STM32F103C8

In embedded systems, memory mapping enables direct interaction with hardware components such as peripherals, system registers, and RAM/Flash via pre-defined addresses. For microcontrollers like the STM32F103C8, understanding this concept is critical for writing efficient and low-level embedded code.

1. What is Memory Mapping?

2. STM32F103C8 Memory Regions

3. What is a Base Address?

4. Where are Base Addresses Used

Submission

Problem 1:

- Link to your Tinkercad file

Problem 2:

- .c file, and optional any other files your worked on (eg. Proteus)

Bonus:

- Your research in .txt, .pdf, etc.

- Submission Link
- The Task's deadline is 31/8 11:59PM
- **Cheating is severely penalized**