**NAME: Jiarui Huang**

**ID: 202283890036**

**Project #1: Face recognition with Micro:bit model and ting:bit robot**

**DATE: 25/6/25**

**GitHub URL:**

**<https://github.com/JerryHuang0424/-ProjectRobot1_2025.git>**

**Video URL:**

**<https://www.bilibili.com/example/video>**

**Table of Contents**

**Part 1 Document The Agile Process**

**Part 2 Implement the face recognition**

1. **Introduction  
   2. Description of Implementation  
    2.1 Task 1:** Building the Robot **2.2 Task 2:** Display images on led matrix **2.3 Task 3:** Recognize faces **2.4 Task 4:** Protect faces data  
    **2.5 Task 5:** Complete a video **2.6 Task 6:** Complete a final Report  
   **3. Final Source Code  
   4. Git Command History**

**Part 1 Document The Agile Process**

*SCRUM(Plan the Quick Reaction Pi Game) Duration:5 Days*

*|*

*|\_\_\_\_\_\_\_\_\_\_>SPRINT 1 - Duration = 5 days*

*Part 1 :*

*Task 1 : Plan Steps*

*Task 2 : Write User Stories*

*Task 3 : Create Burndown Chart*

*Task 4 : Write Agile section of the report*

*Part 2 :*

*Task 5 : Building the Robot*

*Task 6 : Display images on led matrix*

*Task 7 : Recognize faces*

*Task 8 : Protect faces data*

*Task 9 : Complete a video presentation of the project*

*Task 10:Complete a final Report*

· **As a developer**, I want to **build a micro:bit robot**.

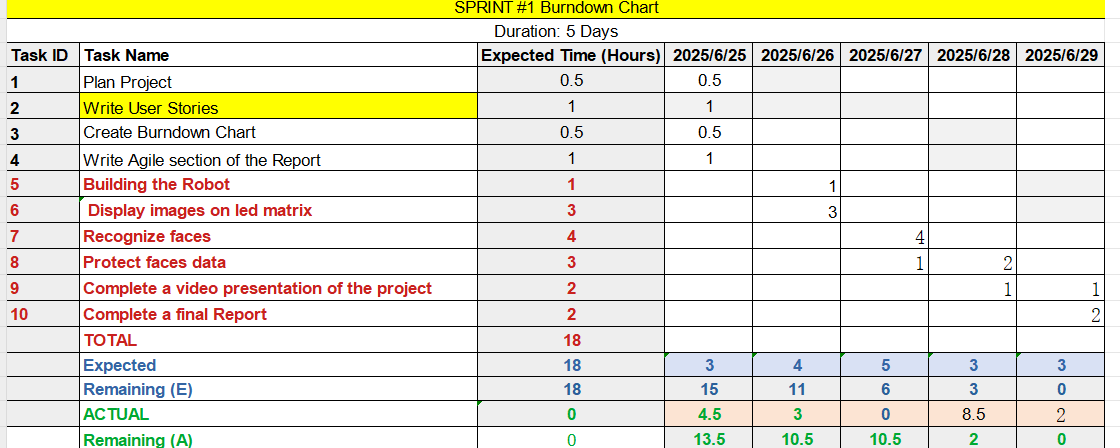
· **As a developer**, I want to **be able to recognize different faces** using micro:bit robot

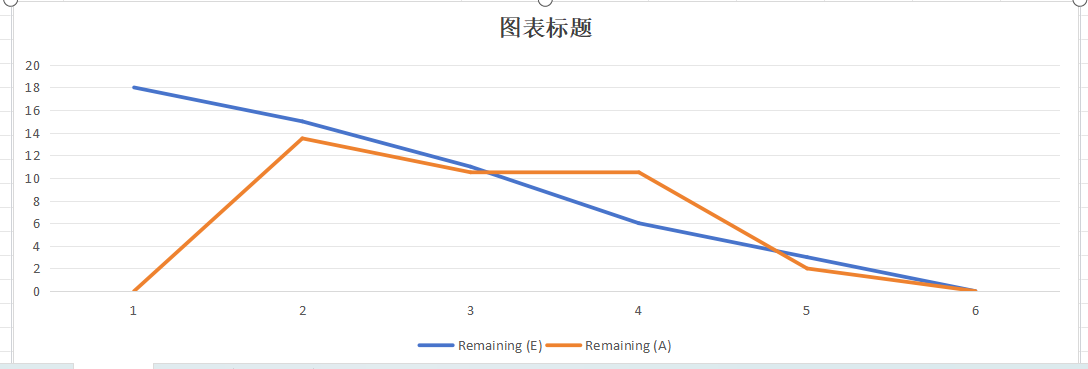
· **As a developer**, I want to be able to **show numbers on micro**:bit led matrix

· **As a developer**, I want to be able to **link different face with different number image.**

· **As a Scrum Master**, I want to **create a burndown chart**, so I can visually monitor our team’s progress during the sprint.

· **As a documentation lead**, I want to **write the Agile section of the final report**, so that our development process is well-documented.





**Part 2 Implement the face recognition**

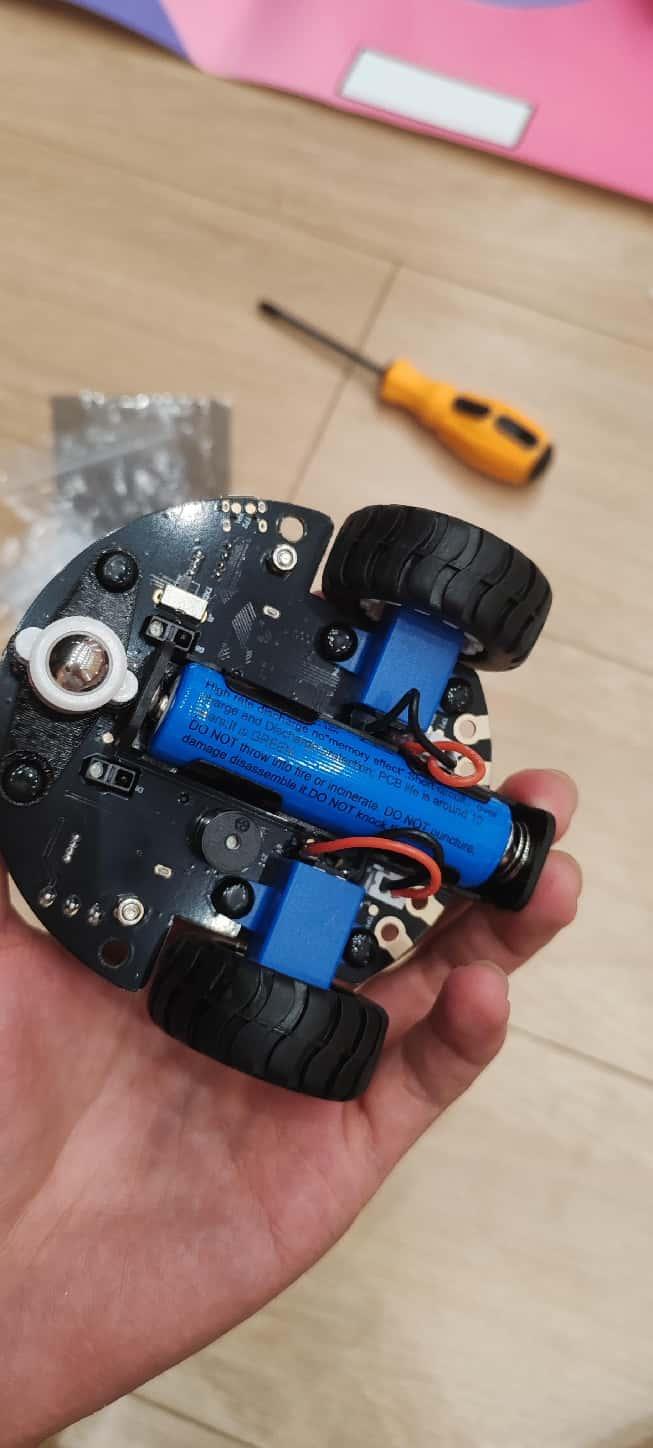
1. **Introduction**

In recent years, the integration of artificial intelligence with embedded systems has gained significant attention, especially in educational and prototyping environments. This experiment focuses on developing an intelligent two-wheeled mobile car capable of facial recognition using the tiny-bit smart car kit combined with the micro:bit micro controller. By leveraging the compact and programmable features of the micro:bit and the expandability of the tiny-bit platform, we designed a small autonomous vehicle that can detect and respond to human faces in real time. This project not only demonstrates the practical application of computer vision in robotics but also provides a hands-on understanding of how embedded systems, sensors, and AI algorithms can work together to create interactive and adaptive machines. The goal of the experiment is to explore the feasibility of implementing face recognition on a resource-constrained platform and evaluate the performance and behavior of the mobile robot in dynamic environments.

1. **Description of Implementation**
   1. **Task 1:** **Building the robot**

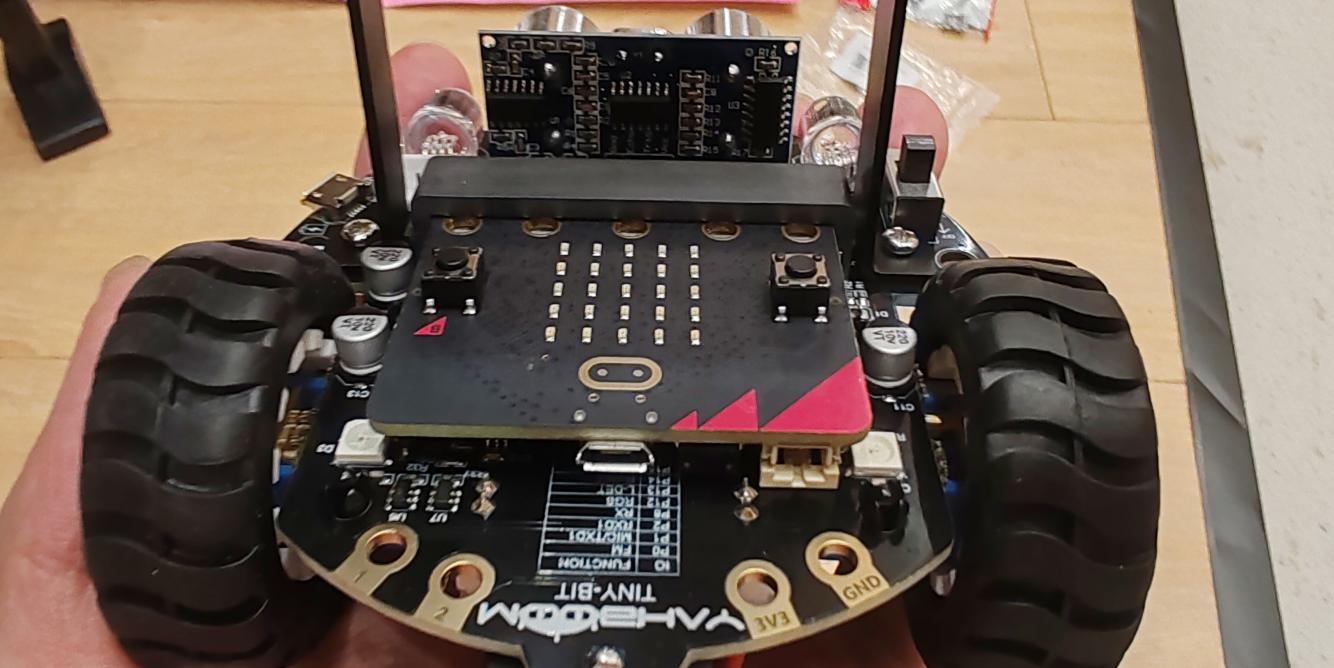
**Assembling the car body**

First, we should implement accessories for the tiny:bit car body, including two tires, one caster wheel, and the central battery.

****

**Implement Micro:bit model**

The micro:bit embedded development board can be directly inserted into the designated slot to complete the installation.

****

**Assembling face recognize model**

First, it is necessary to install the adjustable-angle bracket for the visual recognition module, then fix the K210 visual recognition module onto the bracket.

****

* 1. **Task 2: Display images on led matrix**

**Set face recognition variable state**

First, we need to write code that is used to control the activity of the Tiny:bit and Micro:bit. We detect that the Tiny:bit can only store three faces, so we use a variable to store the face state. Then, the LED matrix will display the number of the face on the LED matrix.

k210\_models.initialization()

music.set\_built\_in\_speaker\_enabled(True)

face = -2

basic.clear\_screen()

def on\_forever():

    if face == 0:

        music.\_play\_default\_background(music.built\_in\_playable\_melody(Melodies.POWER\_UP),

            music.PlaybackMode.UNTIL\_DONE)

        basic.clear\_screen()

    if face == 1:

        music.\_play\_default\_background(music.built\_in\_playable\_melody(Melodies.JUMP\_UP),

            music.PlaybackMode.UNTIL\_DONE)

        basic.clear\_screen()

    if face == 2:

        music.\_play\_default\_background(music.built\_in\_playable\_melody(Melodies.JUMP\_DOWN),

            music.PlaybackMode.UNTIL\_DONE)

        basic.clear\_screen()

basic.forever(on\_forever)

def on\_forever2():

    global face

    face = k210\_models.face\_reg()

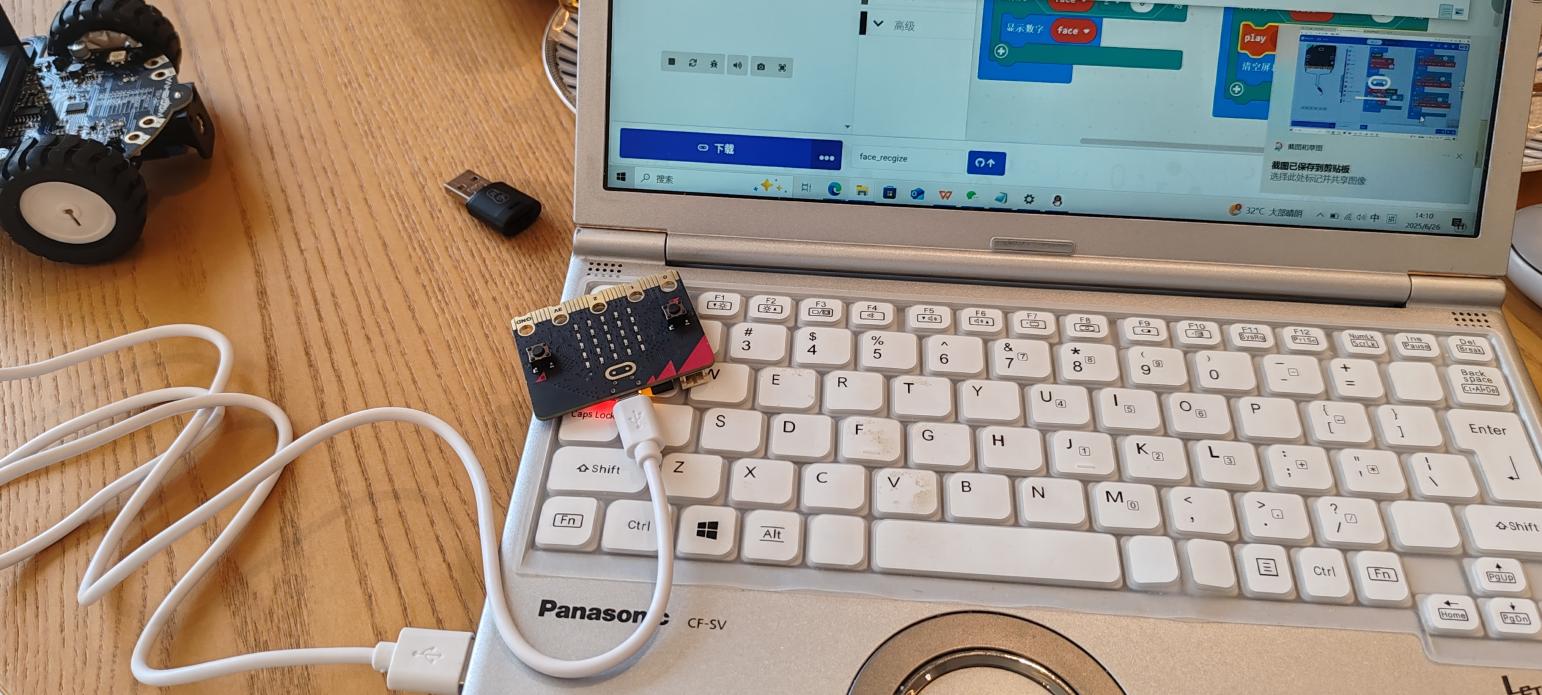
    if face >= 0:

        basic.show\_number(face)

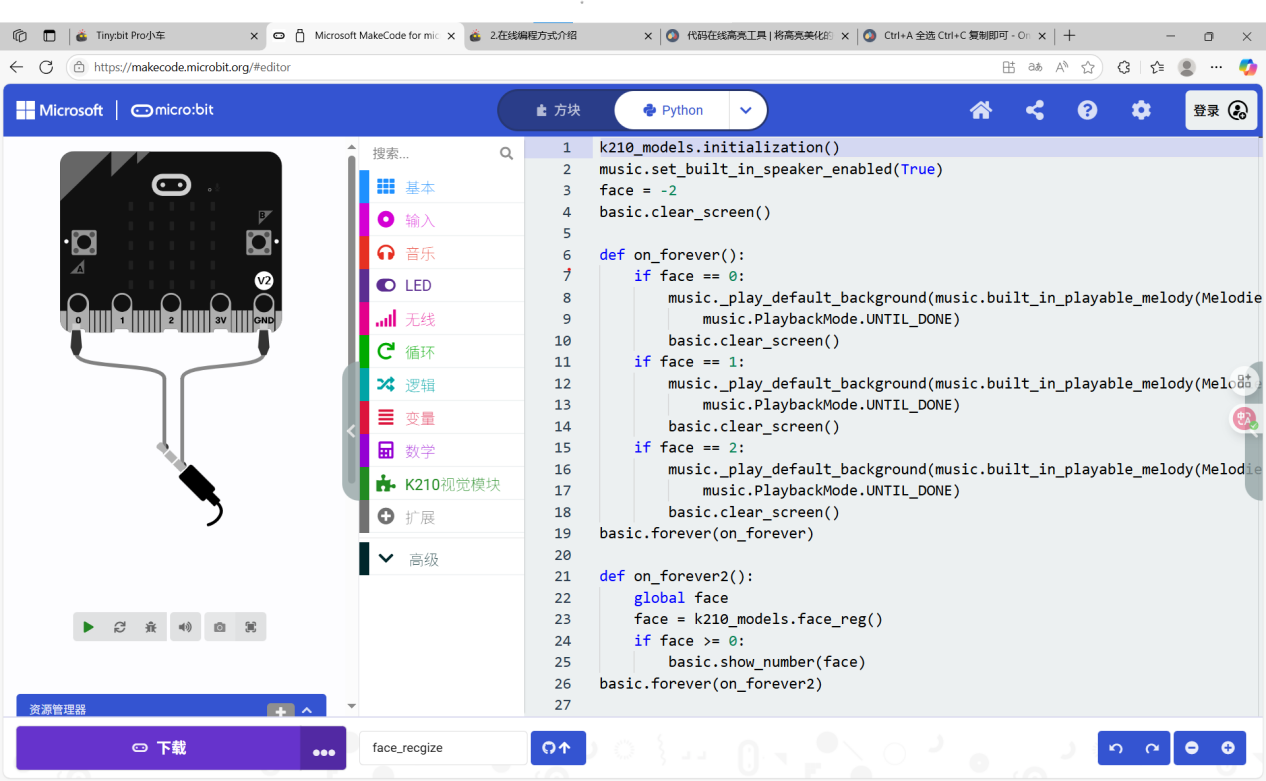
basic.forever(on\_forever2)

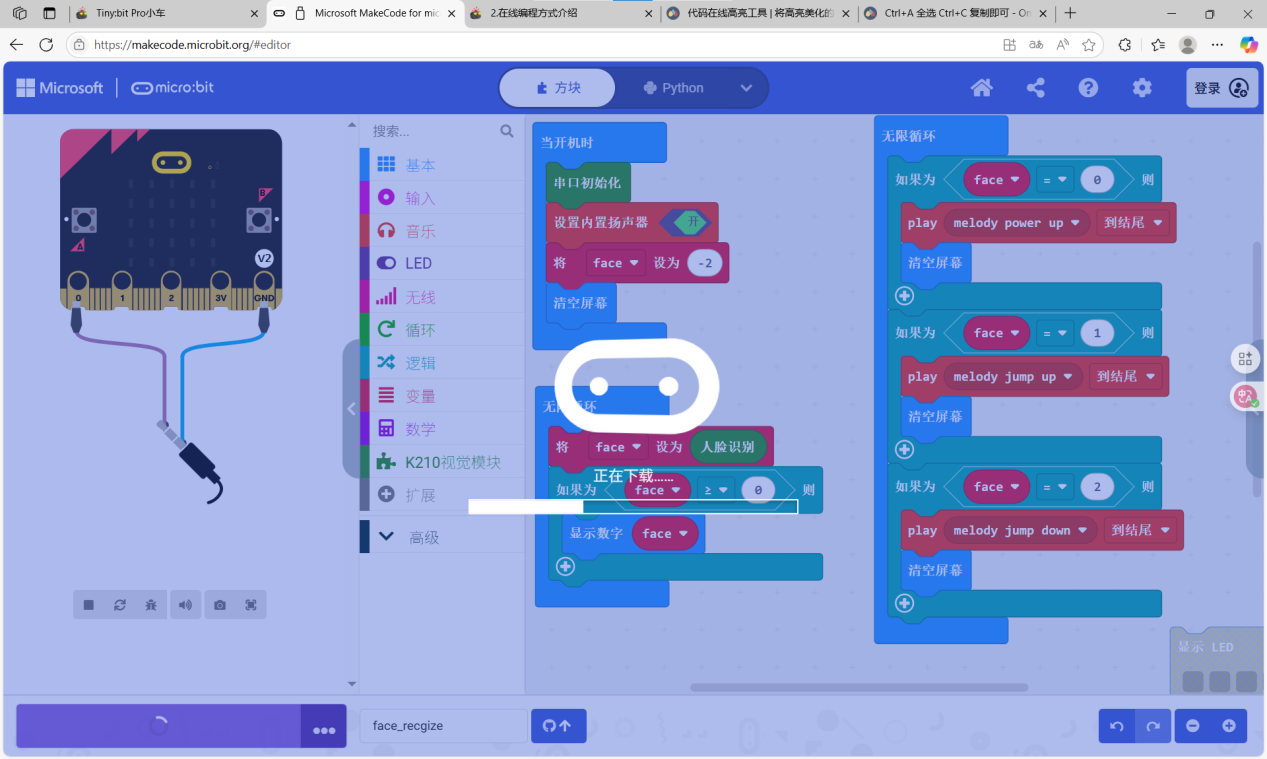
**Implement the code on the Micro:bit**

After write the code, we need to seed the code onto Micro:bit controller, we need connect micro:bit controller with our laptop.

****

Then we can send code onto the micro:bit controller





After deploying the code into ting:bit robot, the face identity can be working,

****

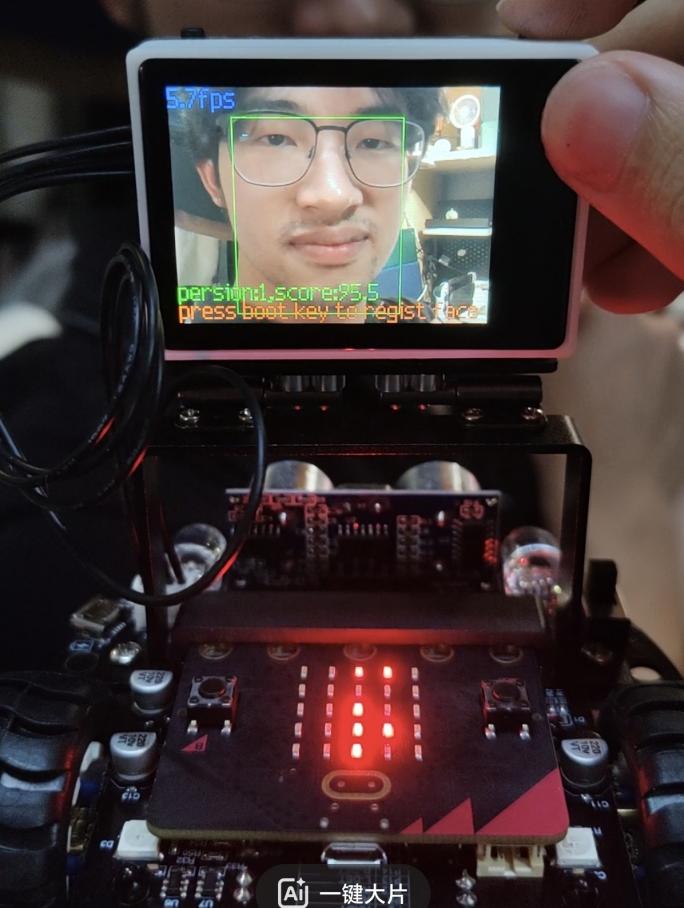
When the robot detects that the recorded face appears in the depth camera, the LED matrix will display an image. In this case, the image displayed is "0".

* 1. **Task 3: Recognize faces**

To identity a face you should point the camera at your face，if the camera detected a new face,it will display a white border to frame the face, if the camera detect a recorded face, it will use green border to frame the face.

****

An unrecorded face inform

****