4 Apriltags recognition

4.1 experimental goals

This lesson mainly learns the machine code recognition function, according to the recognized machine code, box out, and print the relevant data.

The reference code path for this experiment is： CanMV\05-AI\find_apriltags.py

4.2 experimental procedure

The factory firmware of the module has integrated the AI vision algorithm module. If you have downloaded other firmware, please burn it back to the factory firmware before doing the experiment.

1. Import the relevant libraries and initialize the camera and LCD display.

```
import sensor, image, time, math, lcd

lcd.init()
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QQVGA)
sensor.skip_frames(time = 100)
sensor.set_auto_gain(False)
sensor.set_auto_whitebal(False)
clock = time.clock()
```

2. Set the machine code family members to be identified, and comment out that line if you don't need to identify which one. The default here is the TAG 36 H 11 family.

```
tag_families = 0
tag_families |= image.TAG16H5   # comment out to disable this family
tag_families |= image.TAG25H7   # comment out to disable this family
tag_families |= image.TAG25H9   # comment out to disable this family
tag_families |= image.TAG36H10  # comment out to disable this family
tag_families |= image.TAG36H11  # comment out to disable this family (default
family)
tag_families |= image.ARTOOLKIT # comment out to disable this family
```

3. Create a new function to convert the family name to a string.

```
def family_name(tag):
    if(tag.family() == image.TAG16H5):
        return "TAG16H5"
    if(tag.family() == image.TAG25H7):
        return "TAG25H7"
    if(tag.family() == image.TAG25H9):
        return "TAG25H9"
    if(tag.family() == image.TAG36H10):
        return "TAG36H10"
    if(tag.family() == image.TAG36H11):
        return "TAG36H11"
    if(tag.family() == image.ARTOOLKIT):
        return "ARTOOLKIT"
```

4.Create a new while loop, call the find apriltags function to find the machine code in the image, box it when found, and print the relevant information.

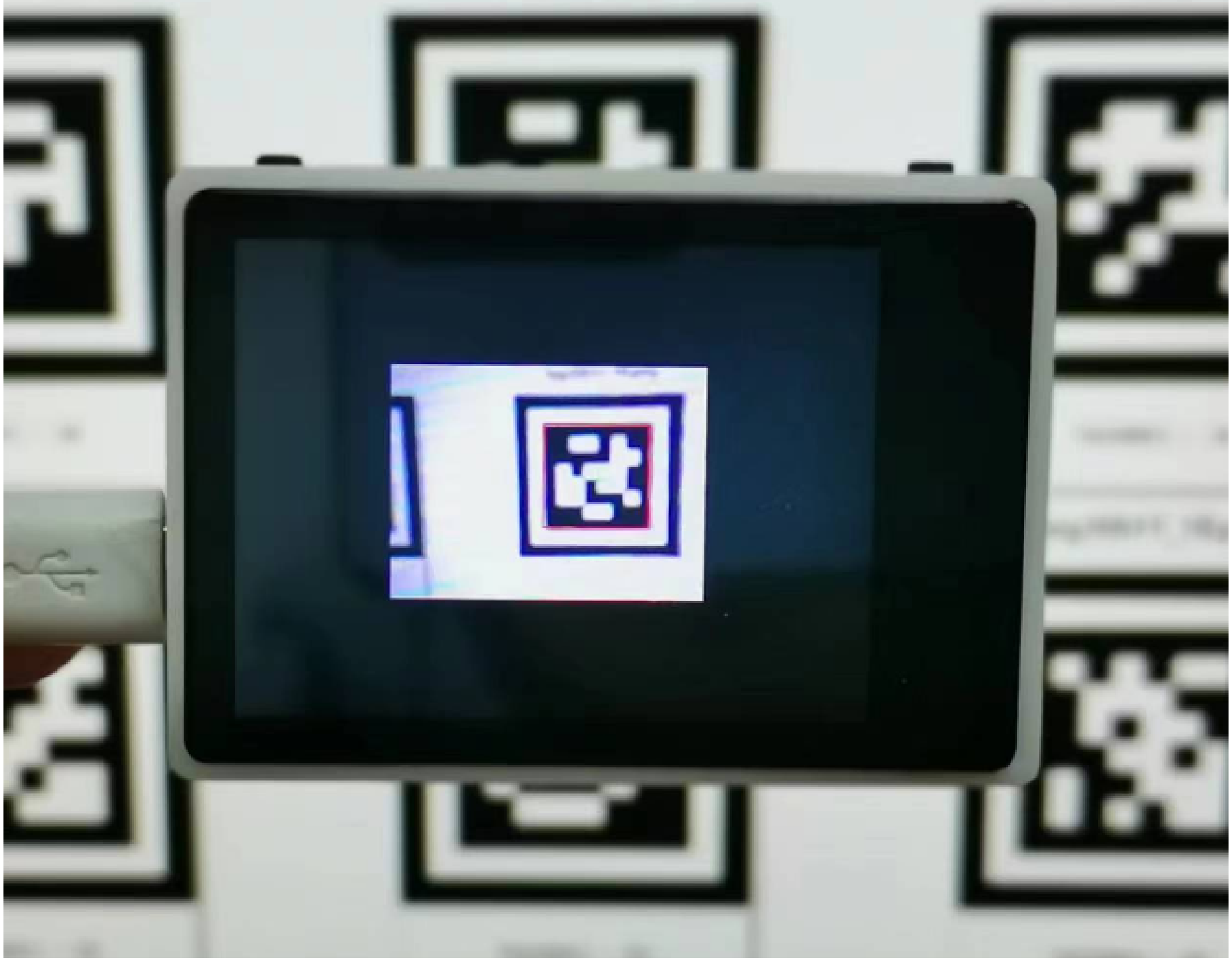
```
while(True):
    clock.tick()
    img = sensor.snapshot()
    #img = img.resize(280, 195)
    #img = img.resize(292, 210)
    for tag in img.find_apriltags(families=tag_families):
        img.draw_rectangle(tag.rect(), color = (255, 0, 0))
        img.draw_cross(tag.cx(), tag.cy(), color = (0, 255, 0))
        print_args = (family_name(tag), tag.id(), (180 * tag.rotation()) / math.pi)
        print("Tag Family %s, Tag ID %d, rotation %f (degrees)" % print_args)
    lcd.display(img)
    #print(clock.fps())
```

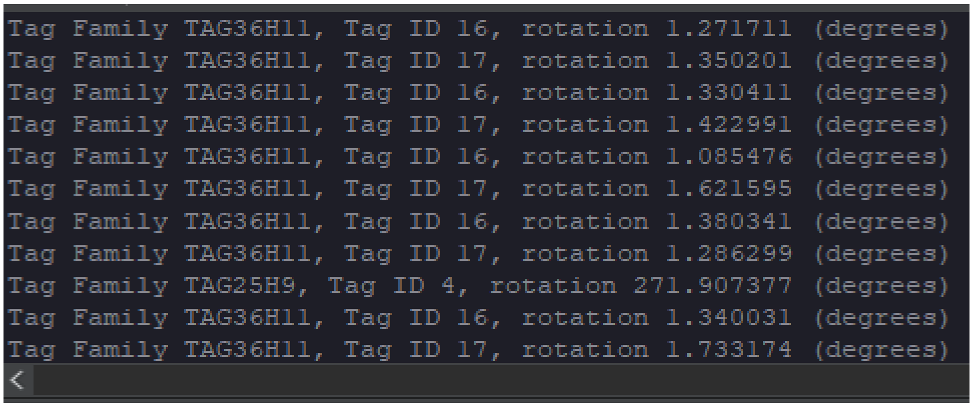
4.3 experimental results

Connect the K210 module to the computer through the microUSB data cable, CanMV IDE click the connect button, after the connection is completed click the Run button to run the routine code. You can also download the code as main.py and run it in the K210 module.

After waiting for the system initialization to be completed, the LCD displays the camera screen, and the camera is used to shoot the machine code, and the types of identification include: TAG16H5, TAG25H7, TAG25H9, TAG36H10, TAG36H11, ARTOOLKIT. Due to the limited processing power of the K210, processing the AprilTag machine code requires a lot of storage space and computing power, so it is currently impossible to set the full-screen resolution size screen.

You can see that the machine code is framed and the machine code information is printed out in the serial terminal at the bottom of the IDE.





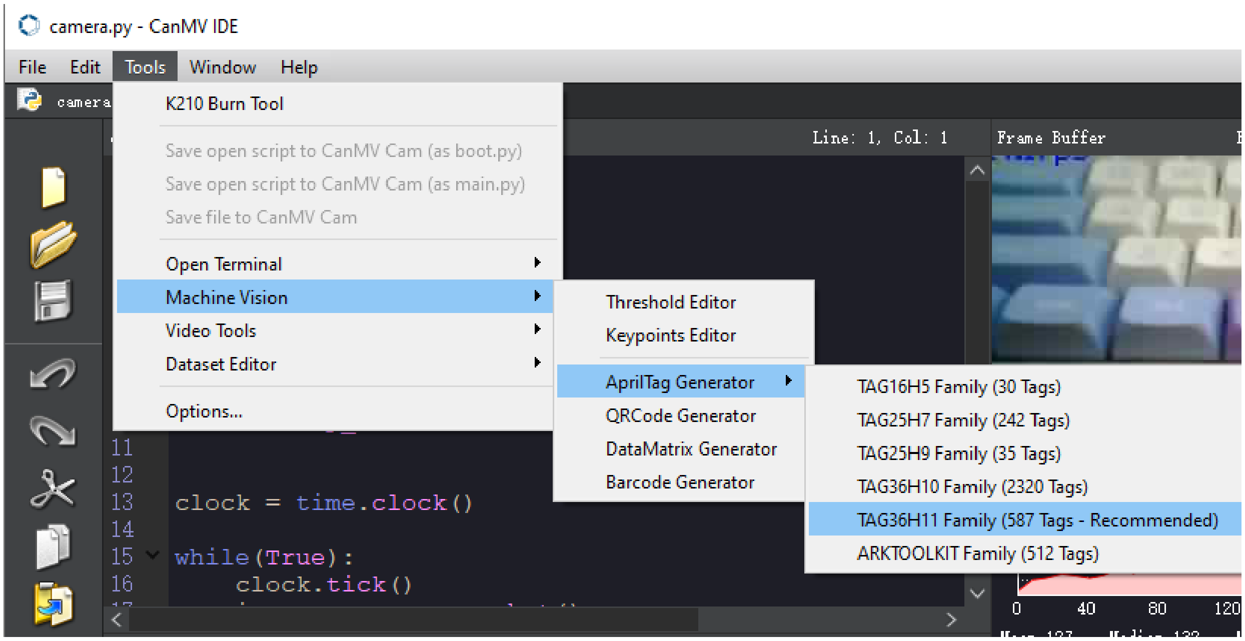
```
Tag Family TAG36H11, Tag ID 16, rotation 1.271711 (degrees)
Tag Family TAG36H11, Tag ID 17, rotation 1.350201 (degrees)
Tag Family TAG36H11, Tag ID 16, rotation 1.330411 (degrees)
Tag Family TAG36H11, Tag ID 17, rotation 1.422991 (degrees)
Tag Family TAG36H11, Tag ID 16, rotation 1.085476 (degrees)
Tag Family TAG36H11, Tag ID 17, rotation 1.621595 (degrees)
Tag Family TAG36H11, Tag ID 16, rotation 1.380341 (degrees)
Tag Family TAG36H11, Tag ID 17, rotation 1.286299 (degrees)
Tag Family TAG25H9, Tag ID 4, rotation 271.907377 (degrees)
Tag Family TAG36H11, Tag ID 16, rotation 1.340031 (degrees)
Tag Family TAG36H11, Tag ID 17, rotation 1.733174 (degrees)
```

4.4 experiment summary

Since machine code recognition requires a lot of calculations, the K 210 itself has limited conditions, so the screen cannot be set to full screen.

If you need to display a larger screen, you can set the sensor.set_framesize (sensor. QVGA), and then after capturing the camera image, increase img = img.resize(292, 210) to modify the image size. However, this will reduce the video frame rate and recognition rate, please do it with caution.

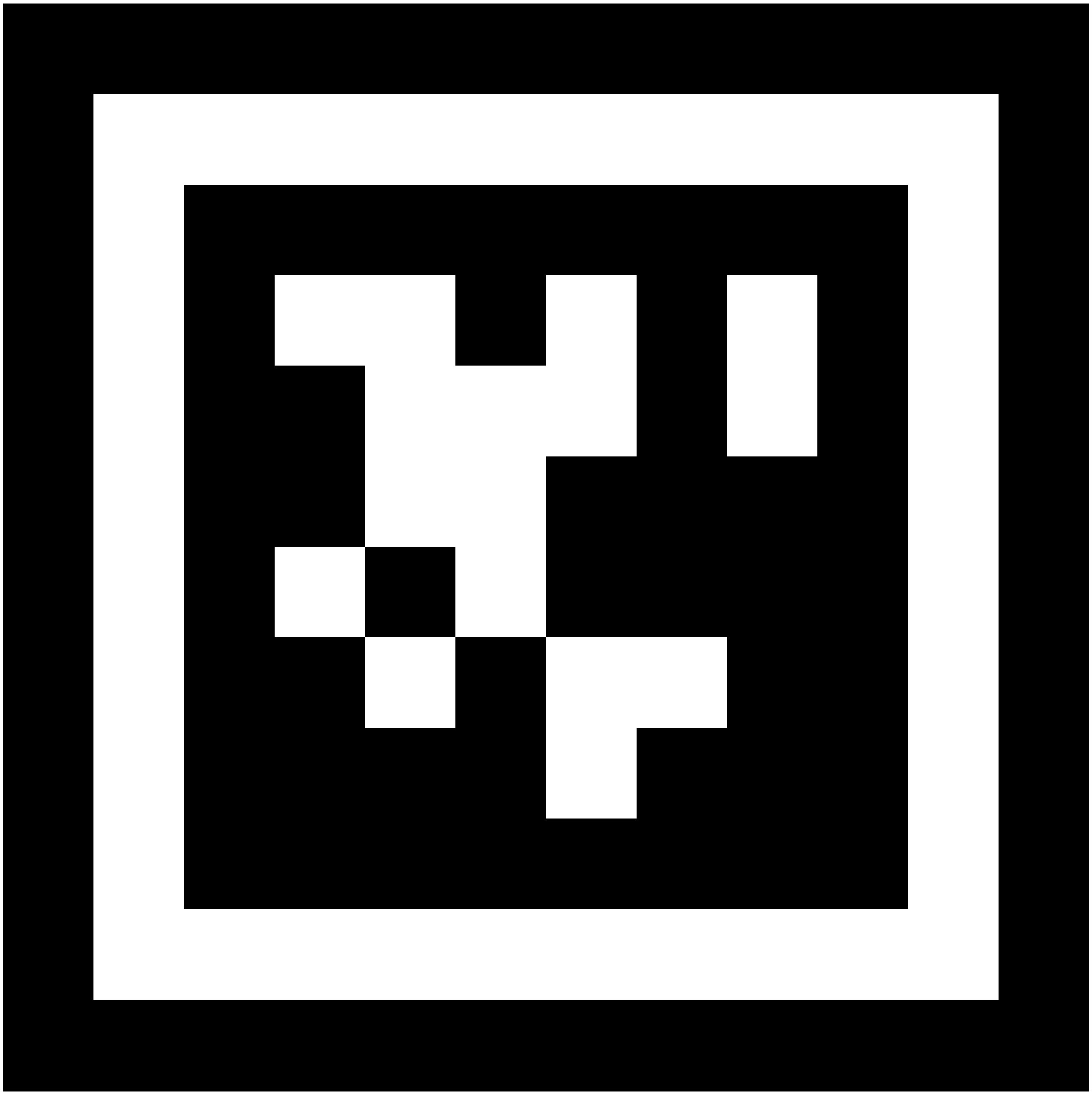
The machine code test uses the TAG36H11 family machine code by default, if you need to apply for other machine code, you can click Tools - > Machine Vision - > AprilTag machine code generator on the CanMV IDE.



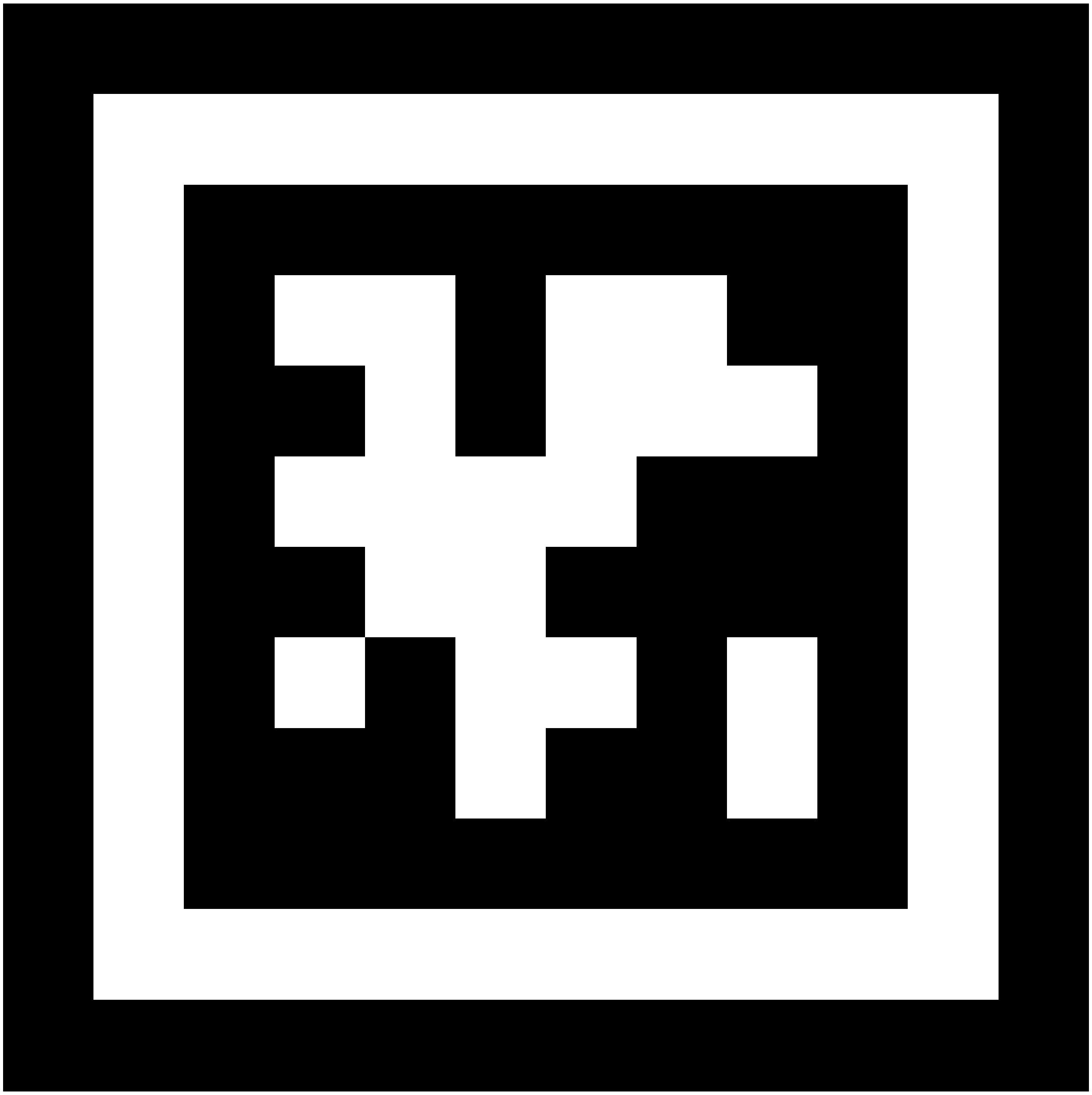
```
clock = time.clock()

while(True):
    clock.tick()
```

TAG36H11 photo:



TAG36H11 - 0



TAG36H11 - 1