

## 8.4、机器码识别

### 8.4、机器码识别

#### 8.4.1、实验目标

#### 8.4.2、实验过程

#### 8.4.3、实验效果

#### 8.4.4、实验总结

### 8.4.1、实验目标

本节课主要学习机器码识别功能，根据识别到的机器码，框出来，并且打印相关的数据。

本次实验的参考代码路径为：K210\_Broad\05-AI\find\_apriltags.py

### 8.4.2、实验过程

模块的出厂固件已经集成AI视觉算法模块，如果下载过其他固件，请烧录回出厂固件再进行实验。

1. 导入相关库，并初始化摄像头和LCD显示屏。

```
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. 设置要识别的机器码家族成员，如果不需要识别哪个就注释掉那一行。这里默认是TAG36H11家族。

```
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.ART00LKIT # comment out to disable this family
```



3. 新建一个函数用来将家族名转化成字符串。

```
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.新建一个while循环，调用find\_apriltags函数查找图像中的机器码，找到之后框出来并且打印相关信息。

```
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())
```

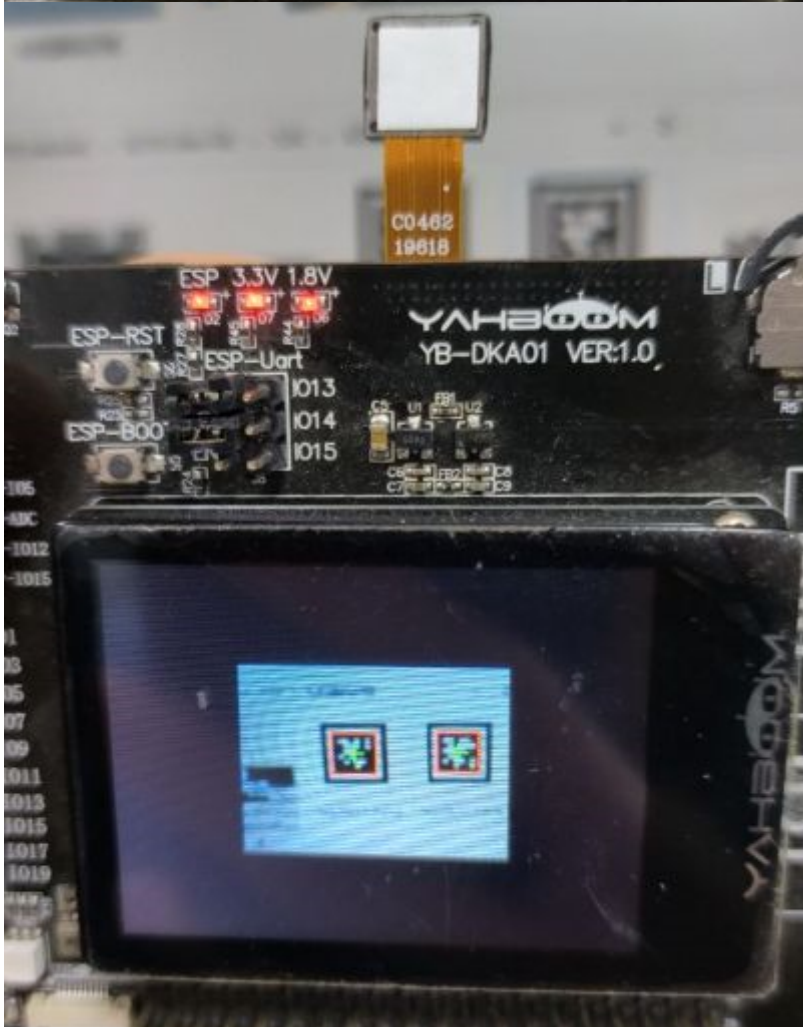


### 8.4.3、实验效果

将K210开发板通过TYPE-C数据线连接到电脑上，CanMV IDE点击连接按钮，连接完成后点击运行按钮，运行例程代码。也可以将代码作为main.py下载到K210开发板上运行。

等待系统初始化完成后，LCD显示摄像头画面，用摄像头拍摄机器码，识别的类型包括：TAG16H5，TAG25H7，TAG25H9，TAG36H10，TAG36H11，ARTOOLKIT。由于K210处理能力有限，处理AprilTag机器码需要大量储存空间和算力，所以目前无法设置全屏分辨率大小画面。

可以看到机器码被框出来，并且在IDE底部的串行终端打印出机器码的信息。





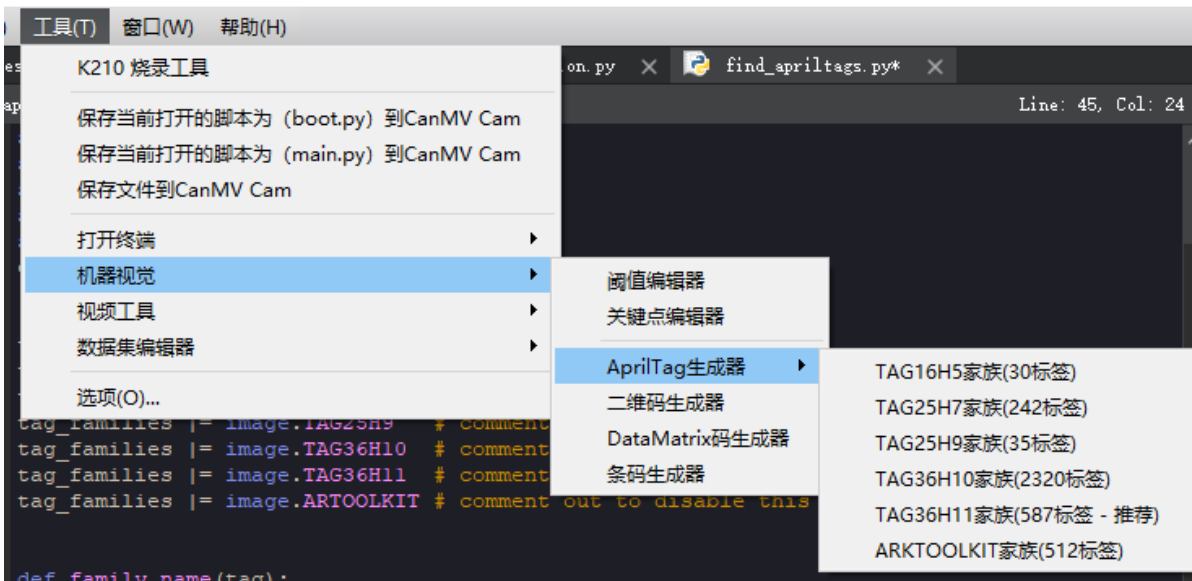
```
串行终端
Tag Family TAG36H11, Tag ID 16, rotation 1.271711 (degrees)
Tag Family TAG36H11, Tag ID 17, rotation 1.350201 (degrees)
Tag Family TAG36H11, Tag ID 16, rotation 1.330411 (degrees)
Tag Family TAG36H11, Tag ID 17, rotation 1.422991 (degrees)
Tag Family TAG36H11, Tag ID 16, rotation 1.085476 (degrees)
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)
<
搜索结果 串行终端
```

### 8.4.4、实验总结

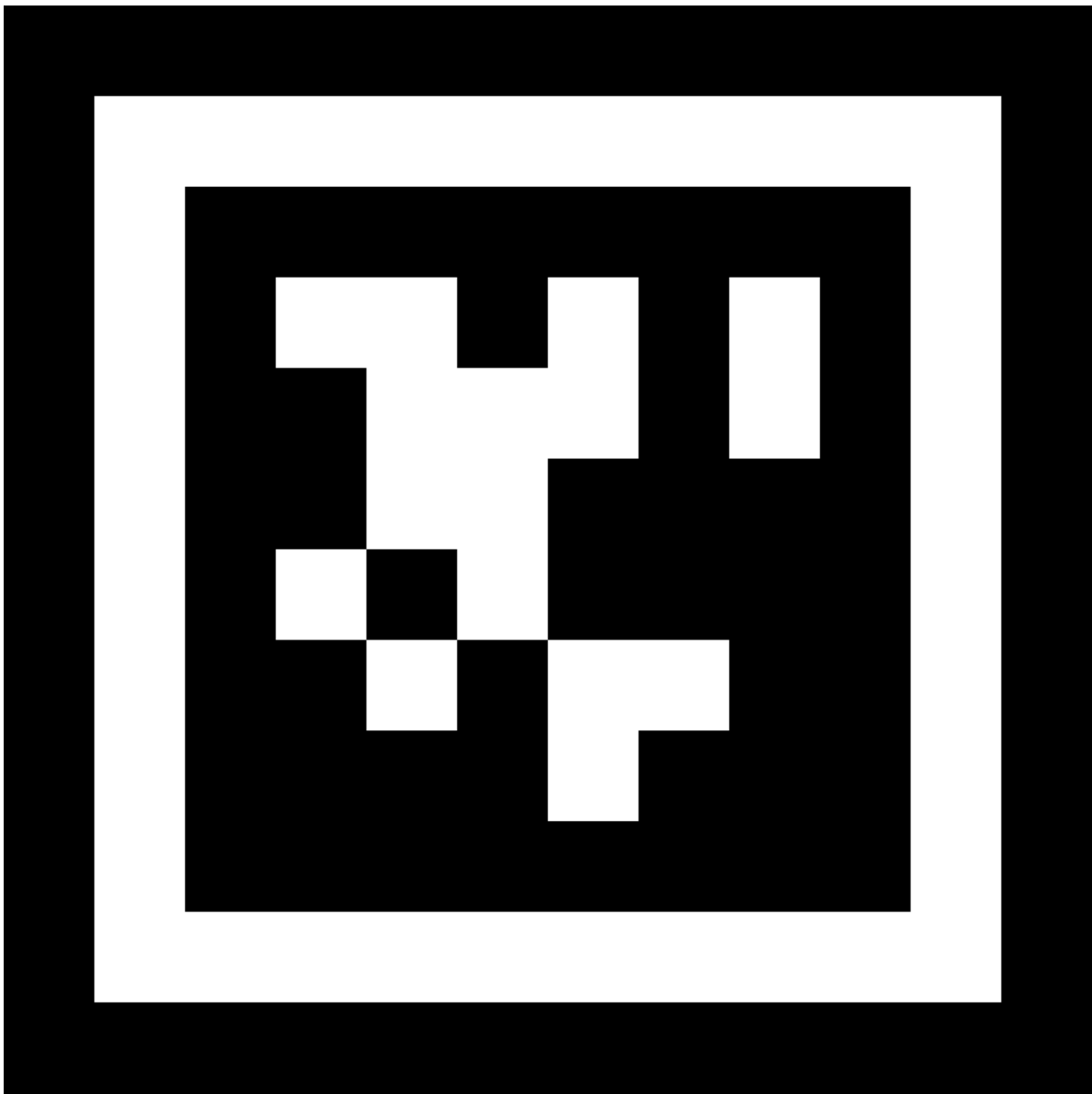
由于机器码识别需要大量计算，K210本身条件有限，所以屏幕无法设置全屏。

如果需要显示更大屏幕，可以设置`sensor.set_framesize(sensor.QVGA)`，然后在捕获摄像头图像后，增加`img = img.resize(292, 210)`修改图片大小。但是这样会降低视频帧率和识别率，请谨慎操作。

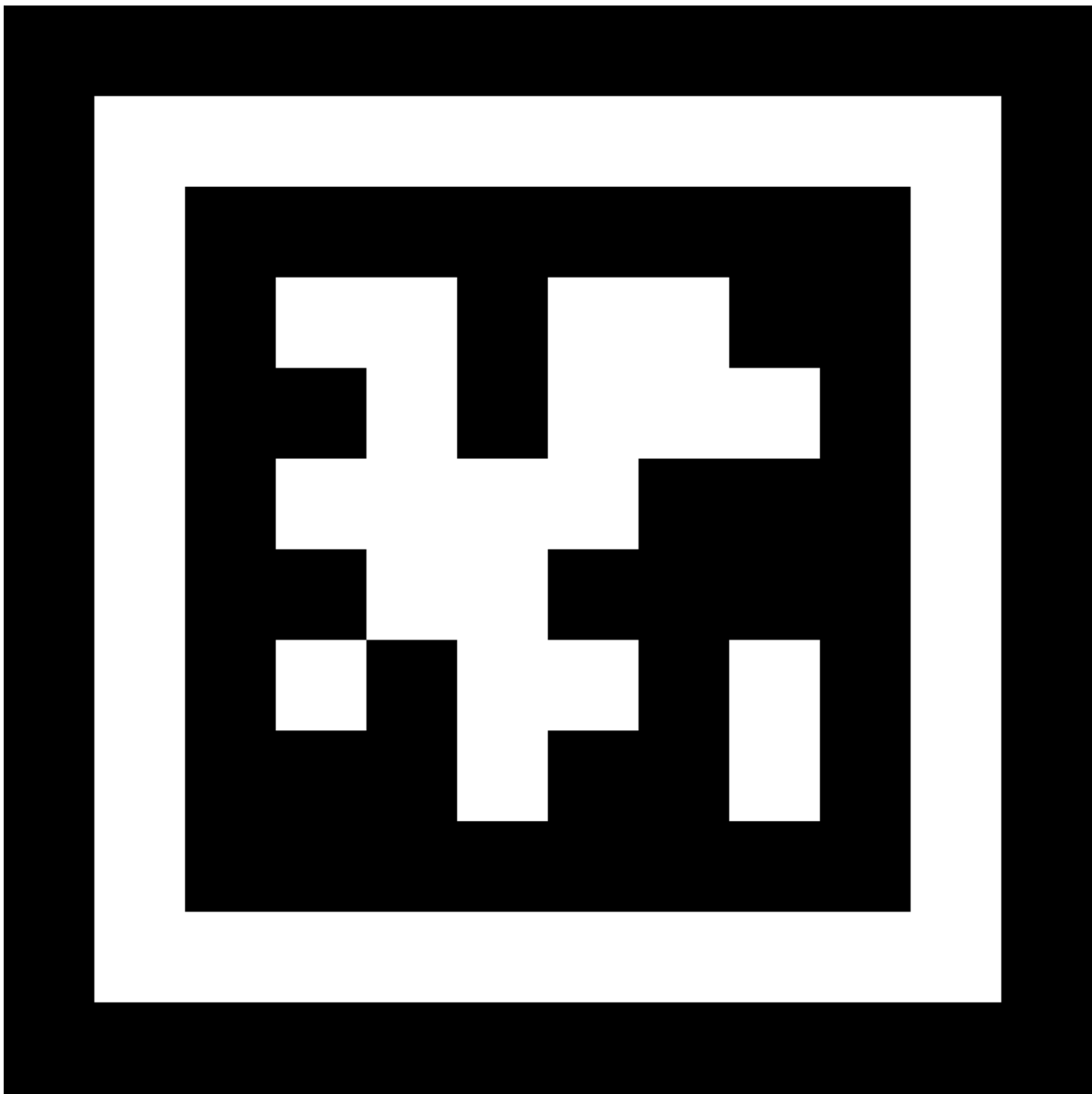
机器码测试默认使用TAG36H11家族机器码，如果需要申请其他机器码，可以在CanMV IDE上点击工具->机器视觉->AprilTag机器码生成器。



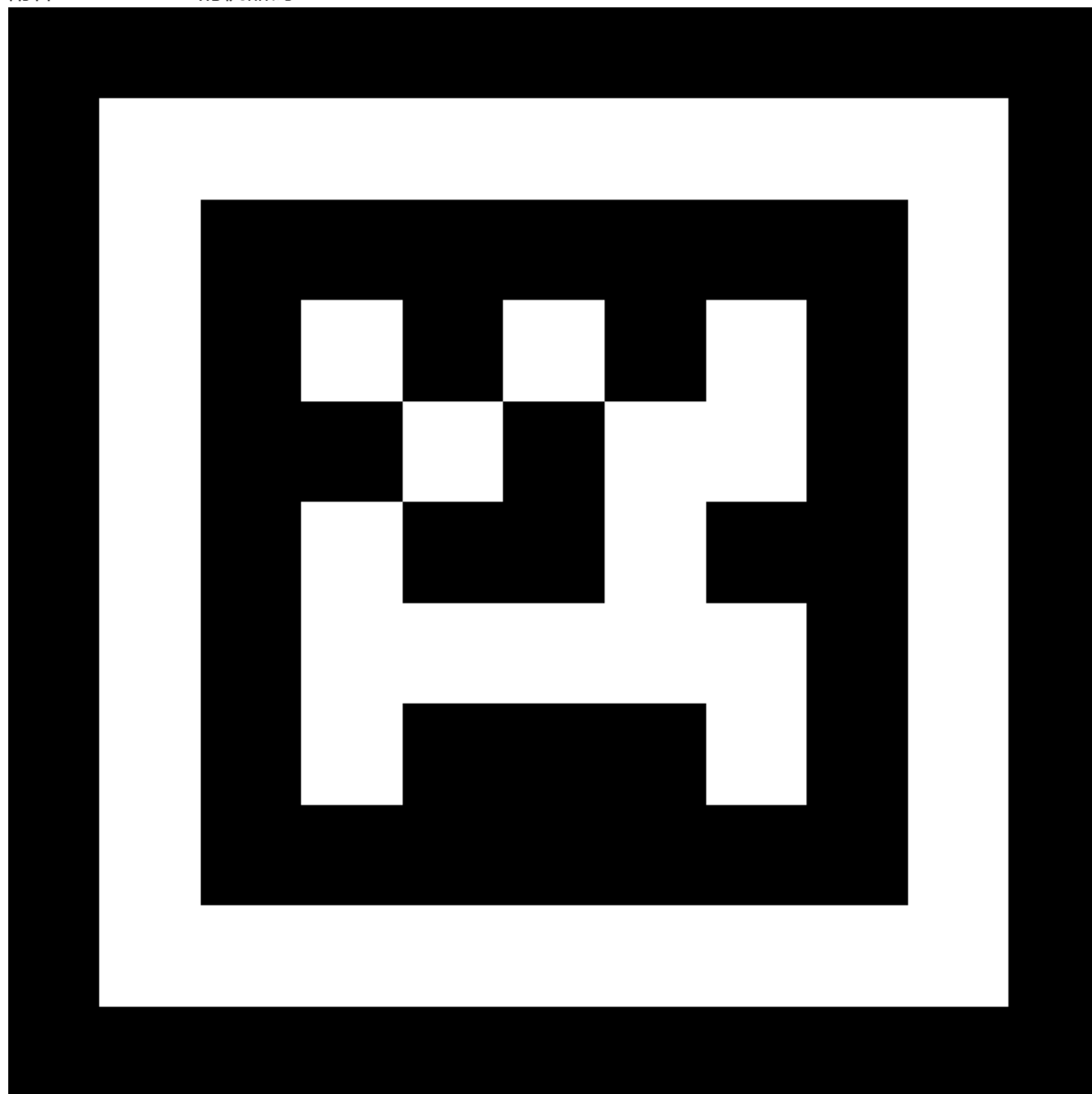
附件1：TAG36H11的机器码



TAG36H11 - 0



TAG36H11 - 1



TAG25H9 - 0