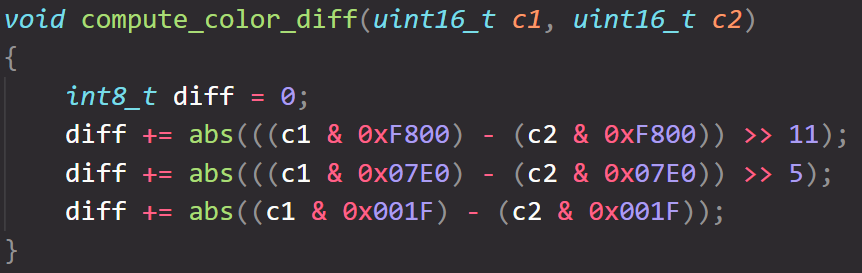
# 运动目标追踪

1. 画面变动率：两帧之间变化像素的变化量。假设有2个像素发生了变化，像素1变化了27，像素2变化了10，则该数值为27+10=37。将两个像素点R,G,B三个分量的差值求和即为变化量。

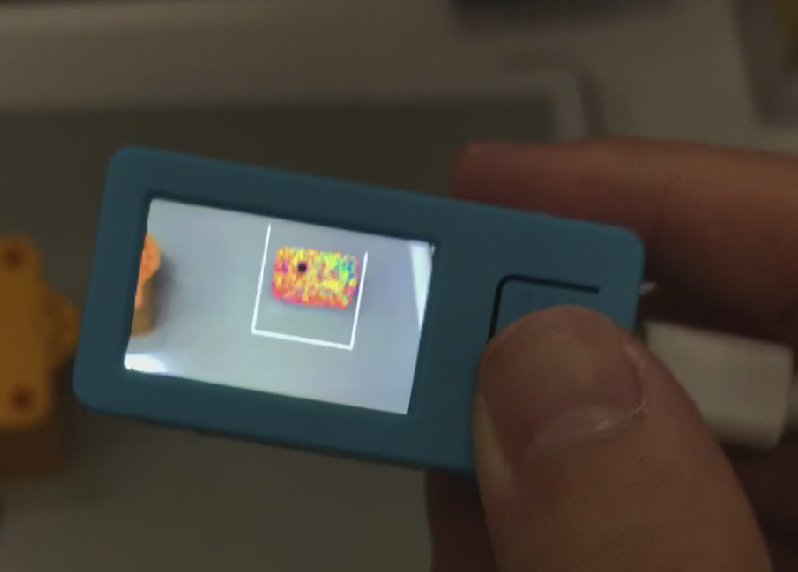


1. 最高变化率：变化最剧烈的像素的变化量
2. 目标ROI：目标的边界框，xy为矩形左上角坐标，wh为长宽。
3. area：变化像素的数量，假设有2个像素发生了变化，该值就是2。
4. 边界框长宽阈值：若（3）中目标ROI的长宽小于该值，则不发送这个边界框。
5. 扫描间隔：检测目标时的步进值，假设step x = 2， step y = 4，则横向每隔2个像素检测一次目标，纵向每隔4个像素检测一次目标。加大该项数值可以有效提升目标检测速度，但有可能漏掉较小的目标。
6. 变化率阈值：按（1）中的计算方式，变化量小于该数值的像素将不被认为发生了变化。
7. 动态检测模式：不断拍摄图像，比对前后两帧之间的变化。
8. 静态检测模式：拍一张照片并保存，比对新拍摄的图像与照片直接的变化。发送该指令时则自动拍摄，若要重新拍摄新的照片，需要先切换回动态检测模式再发送该指令。
9. （\*此项设置暂未支持）边界框合并阈值：合并两个距离小于该值的边界框。



# 目标追踪

1. 框出目标在画面中的位置，即可自动追踪框中的目标。目标最好具有显著的颜色特征。



# 色块识别

1. area：该颜色的像素数量
2. LAB颜色阈值：是LAB颜色空间的颜色值，在该范围外的颜色将会被过滤。这个值的选取需要专用的阈值编辑器实现，后续我会写一个小工具。
3. 检测步进，长宽阈值，合并阈值与运动目标追踪中所述一致。

# 人脸识别

1. 使用JSON作为数据包格式
2. {
3. "FUNC": "FACE DETECT",  *// 功能说明*
4. "count": 3,   *// 识别到的人脸数量*
5. "2": {  *// 人脸编号*
6. "x": 97,    *// ROI*
7. "y": 26,
8. "w": 64,
9. "h": 86,
10. "value": 0.859508,  *// 置信率*
11. "classid": 0,
12. "index": 2,
13. "objnum": 3
14. },
15. "1": {
16. "x": 70,
17. "y": 157,
18. "w": 38,
19. "h": 63,
20. "value": 0.712100,
21. "classid": 0,
22. "index": 1,
23. "objnum": 3
24. },
25. "0": {
26. "x": 199,
27. "y": 145,
28. "w": 31,
29. "h": 40,
30. "value": 0.859508,
31. "classid": 0,
32. "index": 0,
33. "objnum": 3
34. }
35. }

# 二维码识别

{

    "count": 1,

    "FUNC": "FIND QRCODE",

    "0": {

        "x": 57,

        "y": 16,

        "w": 197,

        "h": 198,

        "payload": "m5stack", //二维码数据

        "version": 1, //二维码版本

        "ecc\_level": 1, //二维码ECC水平

        "mask": 2, //二维码掩码

        "data\_type": 4, //二维码数据类型

        "eci": 0 //返回二维码的ECI。

    }

}