

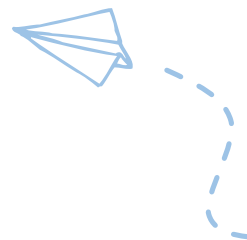
2019 RoboMaster冬令营

第6组

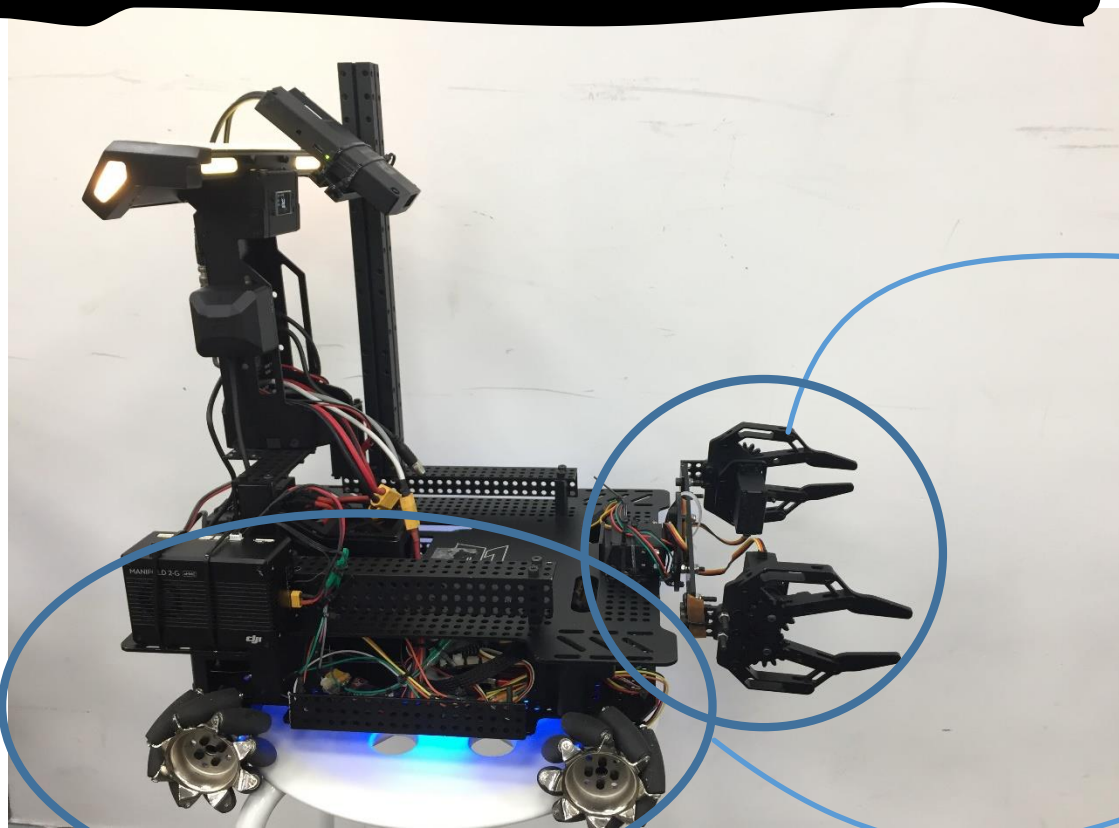




机械设计



工兵机器人



旋转夹

改装版教育套件底盘

根据规则，一次最多只能夹两个块，我们选择了夹子的方案。

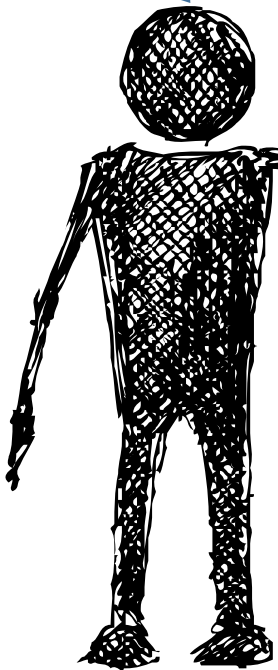
夹取机构的方案

Plan A 摩擦轮

优点：更加方便控制多个方块。
缺点：不能准确控制落点，
方块与摩擦轮的衔接调试时间长，需要更多时间开发。

Plan B 夹子

优点：可从现有的套件上开发，合理利用现有材料，节省时间。控制精准，方便工兵机器人和解密机器人的对接。
缺点：不能同时控制多个方块。





旋转夹

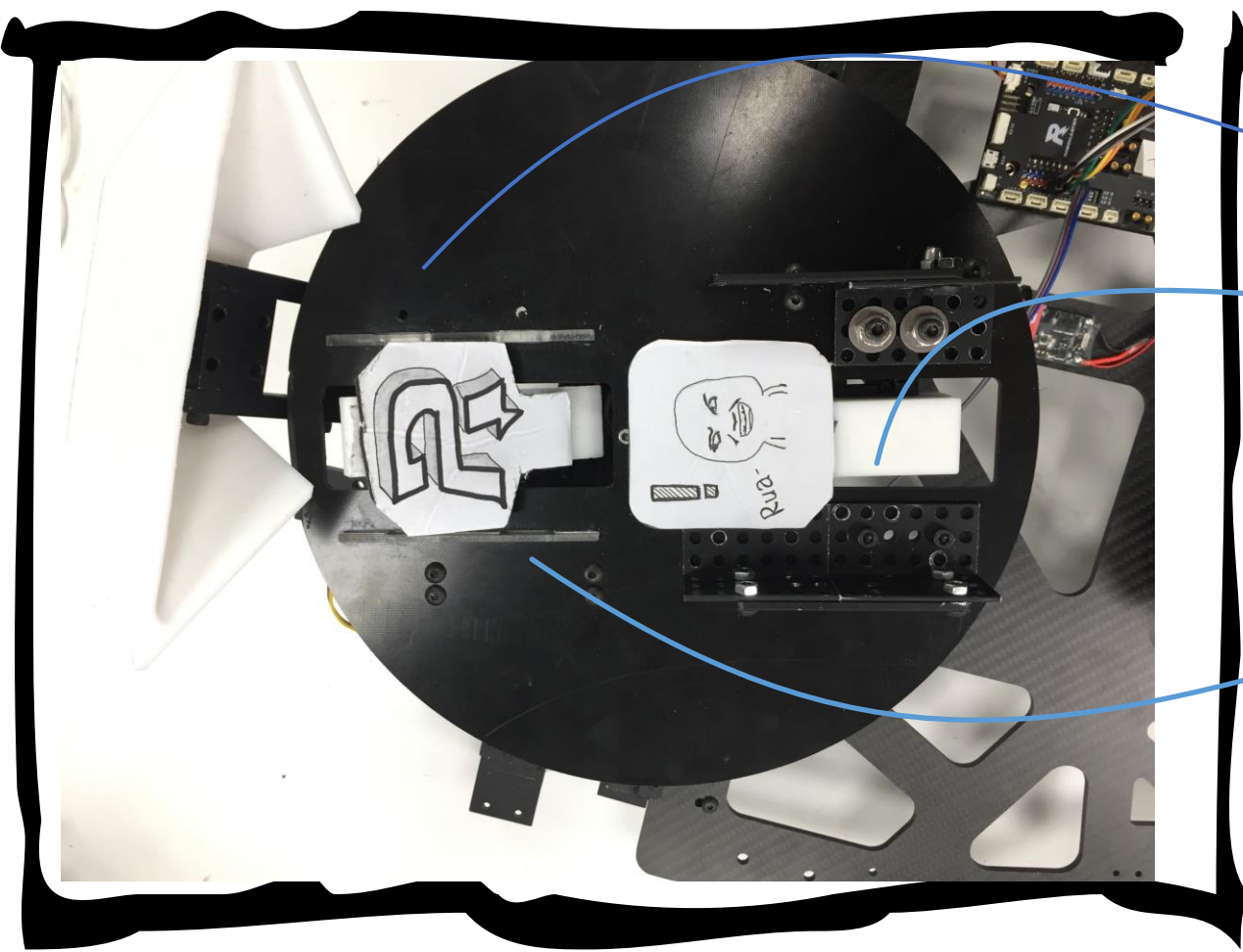
优点:

- 1.兼具夹子的优点，但是同样可控制多个方块。
- 2.上升方式简单、快速。

缺点:

- 1.旋转轴的受力较大，受力点较小，降低电机寿命。
- 2.夹子的线容易绕在一起。

解密机器人



转盘

弹出装置

限位器

可选用的结构

镜子

优点：
机械结构简单
缺点：
加工难度大，
材料获取难度高

转盘

优点：
制作简单
缺点：
容易绕线，
加装翻转结构难度大

机械臂

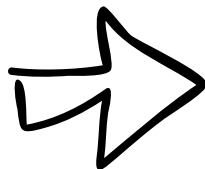
优点：
种类多，可实现的翻转形式多。
缺点：
机械结构复杂。

解密机器人的数次迭代



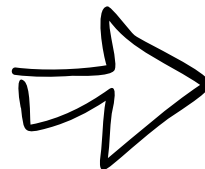
弹出装置

从套件拼接到3D
打印一体成型：
弹出轨迹可控



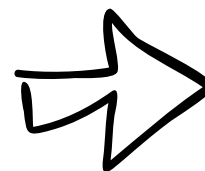
左右限位器

从铝制金属梁到亚
克力板拼接：**限位
但不遮挡视野**



翻斗

漏斗式设计：**轨迹
可控并真正利用旋
转夹**

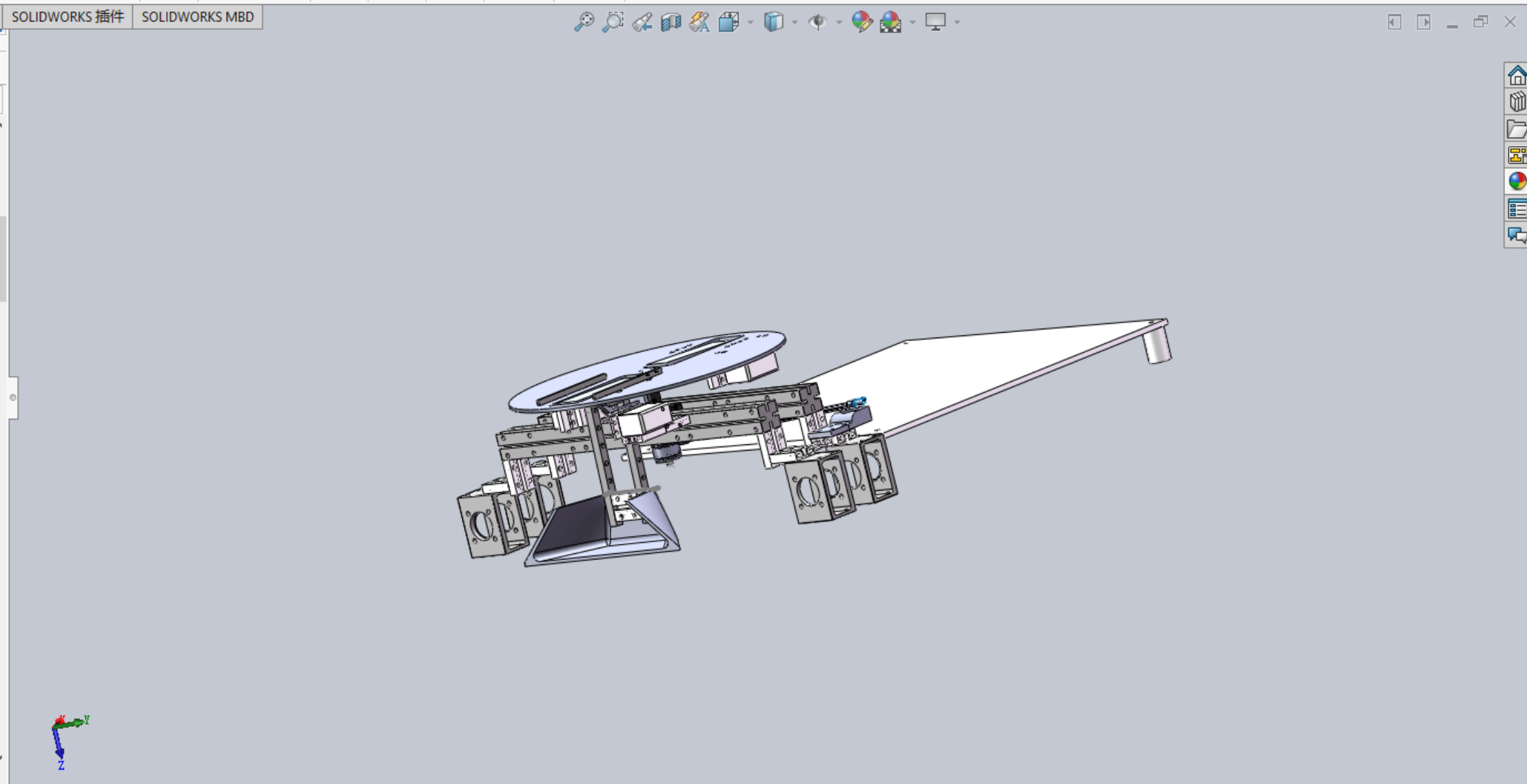




装配体 布局 草图 评估 SOLIDWORKS 插件 SOLIDWORKS MBD



- 板<2> (默认<<默认>>_显示状 ^
- (-) 支撑4<1> (默认<<默认>>_
- (-) 支撑4<2> (默认<<默认>>_
- (-) 裁判系统主控灯柱<1> (默认<<默认>>_
- (-) 侧挡板<1> (默认<<默认>>_
- (-) 侧挡板<2> (默认<<默认>>_
- (-) 支撑6<1> (默认<<默认>>_
- 32mm双孔梁<1> (默认<<默认>>_
- 32mm双孔梁<2> (默认<<默认>>_
- 电机座Z80123<5> (默认<<默认>>_
- (-) 32mm双孔梁<4> (默认<<默认>>_
- (-) 圆柱<1> (默认<<默认>>_显
- 32mm双孔梁<8> (默认<<默认>>_
- 32mm双孔梁<9> (默认<<默认>>_
- (-) 纹盘<1> (默认<<默认>>_显
- 上部<1> (默认<<默认>>_显示
- 下部<1> (默认<<默认>>_显示
- 32mm双孔梁<11> (默认<<默认>>_
- 32mm双孔梁<12> (默认<<默认>>_
- 32mm双孔梁<14> (默认<<默认>>_
- 32mm双孔梁<15> (默认<<默认>>_
- 32mm双孔梁<16> (默认<<默认>>_
- 32mm双孔梁<17> (默认<<默认>>_
- (-) 支撑<5> (默认<<默认>>_显
- (-) 支撑 - 副本<5> (默认<<默认>>_
- (-) 3x3L型连接片 (自制)
- 112<1> (Default<<Default>>_

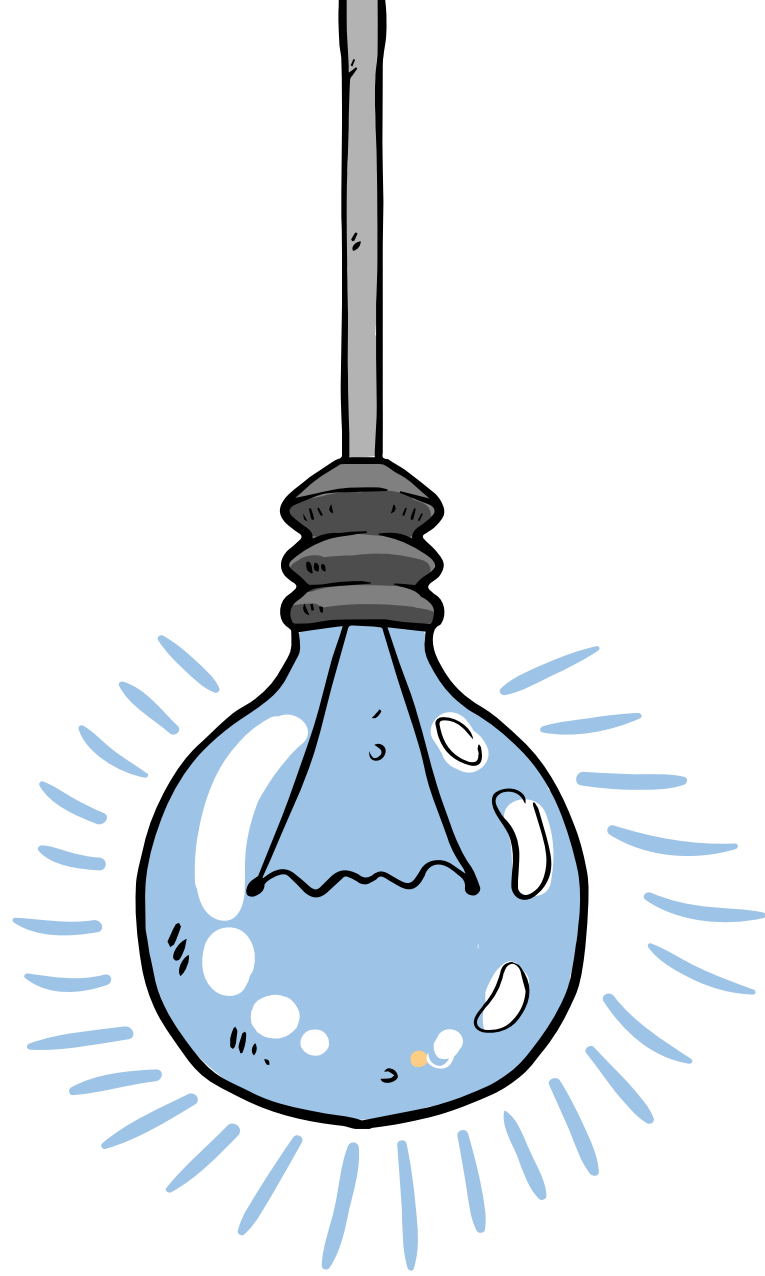


2019

RoboMaster冬令营嵌入式部分-工兵机器人



By 张开源



大体可分为
是三个部分

嵌入式开发过程中

1 算法通讯接口

2 控制逻辑架构

3 自动控制函数

迭代过程

设计
目标

初代

初步构想

次代

针对性改进

终代

最终定稿

初代初步构思

01

算法通讯接口

- 利用FreeRTOS分配，使得路径规划算法作为独立线程运行在主控板之上，使用全局变量或者调用栈进行通信以极坐标思想获得方向和位移长度两个信息
- 打乱地盘控制周期，拖慢算法运行速率，造成控制的滞后

02

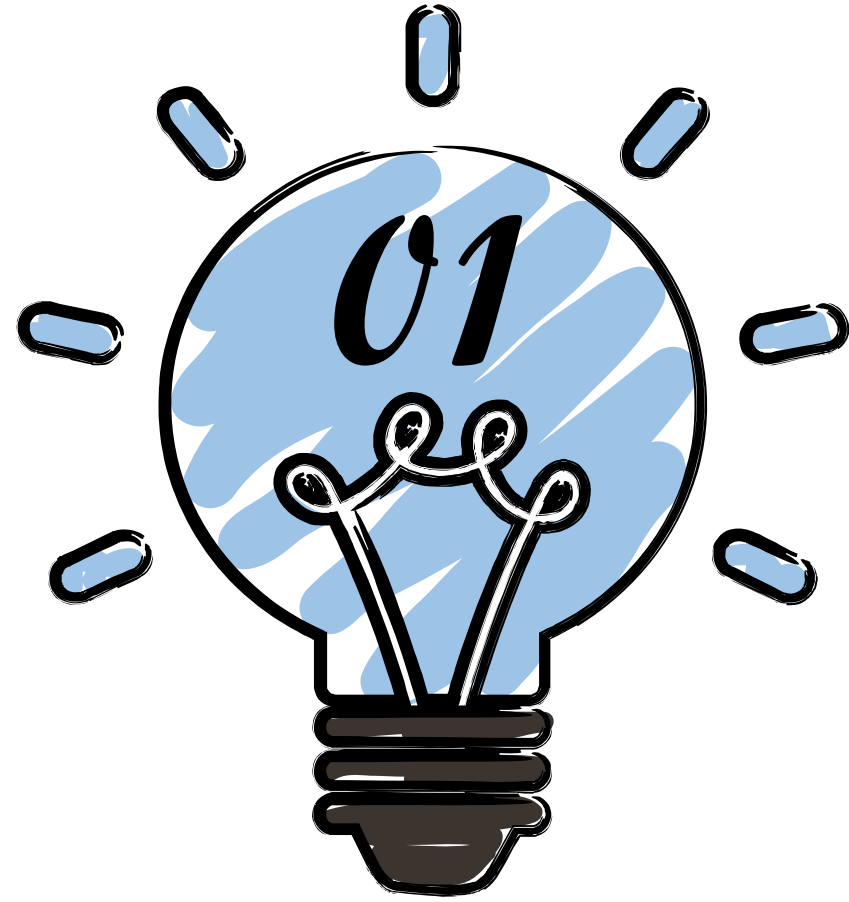
控制逻辑架构

- 基本沿用步兵控制架构，由右侧拨杆的挡位在移动、机械臂、自控间切换
- 无法满足操作手对功能性及可操作性的需求

03

自动控制函数

- 通过光电传感器采集位移量对角速度闭环循迹，通过RFID采集绝对和相对轨迹里程计闭环90°转角
- 差速驱动法对于麦克纳姆轮效率不高，RFID暂时无法读取



迭代更新简介



算法通讯接口升级

次代 1.采用pc/妙算上位机，提高算法运行效率，为嵌入式部分进程保留更多运算力和时间

2 采取事先将路径一次性压栈，由嵌入式部分自主提取，防止通讯时的错误

终代 采用妙算2上位机，两者之间采用uart异步串口通信加强互动，提高算法部分對於操作手介入等突發事故時重新規劃路徑

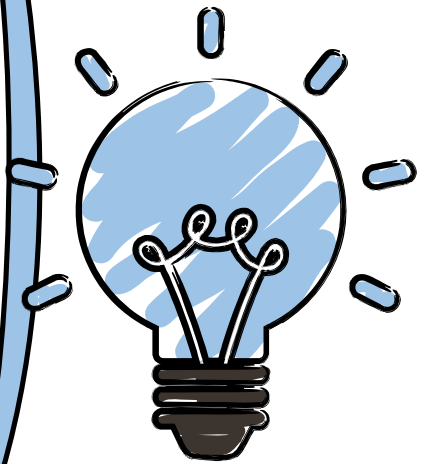
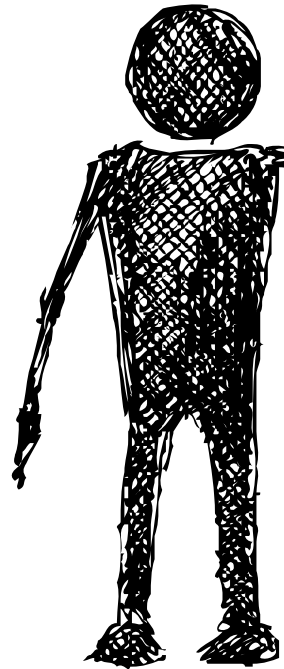
控制邏輯架構升級

次代

- 1、將三種模式切換替換為左側撥桿，右側撥桿加裝一鍵90°轉角功能功能，
- 2、在機械臂控制模式下采用左側搖桿控制主臂舵機右側遙感xy位移量分別控制兩隻機械爪的開合

終代

- 1、應操作手要求將一鍵90°模式替換為一鍵舵機復位功能
- 2、將主臂舵機的控制全部集成于一鍵功能中，將兩隻小機械爪的控制改為左側右側搖桿兩各一，規避同一搖桿操控時對xy軸之間干涉導致的誤操作

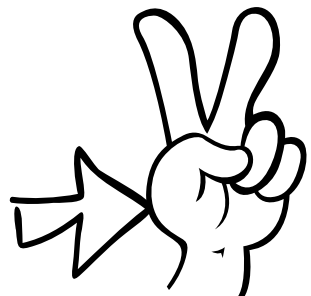


移動控制函數升級



step1

見初代構
想



step2

- 1、加裝imu單軸姿態傳感器直接監控角度進行角度pid閉環控制，製作90°固定轉角子函數
- 2、由於RFID暫時無法讀取，采用記錄十字黑綫的辦法判斷走過的節點數，配合方向和起始坐標建立相對和絕對移動路經記錄以及車輛坐標
- 3、將黑綫對角速度的差速控制改為對橫向移動速度的控制，同時引入角度閉環，使得車輛在進入直行狀態時的方向上進行鎖止，優化了機器人循跡的移動效率和軌跡，漸少了脫綫的風險



step3

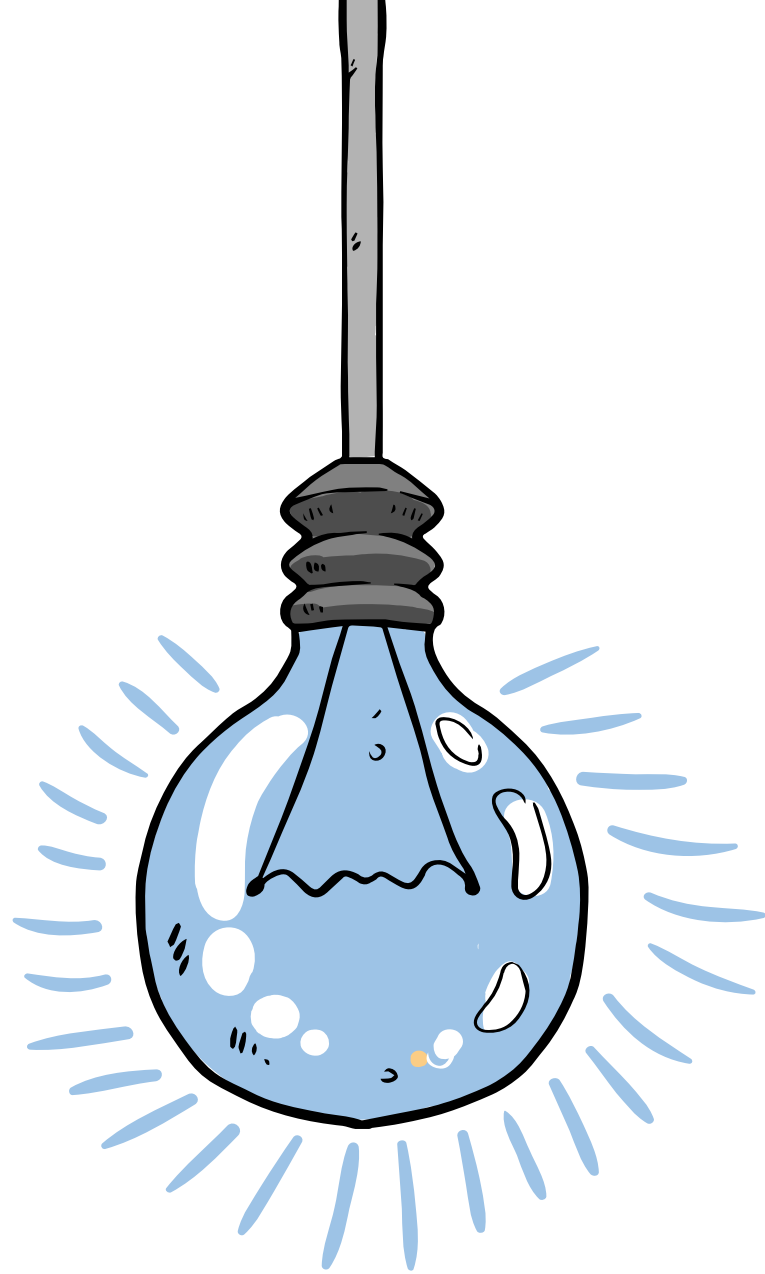
- 1、讀取IMU角速度，形成對角速度的閉環控制，建立角度-角速度-電機轉速-電流的串級pid控制
- 2、利用官方歷程加串口通信的方式建立RFID讀取上位機獲得RFID數據並通過坐標系及向量運算計算相對位移
- 3、利用此前函數建立參數為方向-距離的自動循跡函數，完成針對實現算法命令的封裝函數溝通算法嵌入式的研究成果



嵌入式部分-解密机器人

BY 郭婧汝





目录

1. 明确机器功能
2. 初步实现功能
3. 优化实现方法
4. 装置完善改进



明确机器功能



解密功能明确

能够读取工兵机器人运送的物块，
并读取二维码

（最优：六面二维码皆可读取）



初步实现功能

初步功能实现方案 【尝试】

启动条件：
串口字符='1'

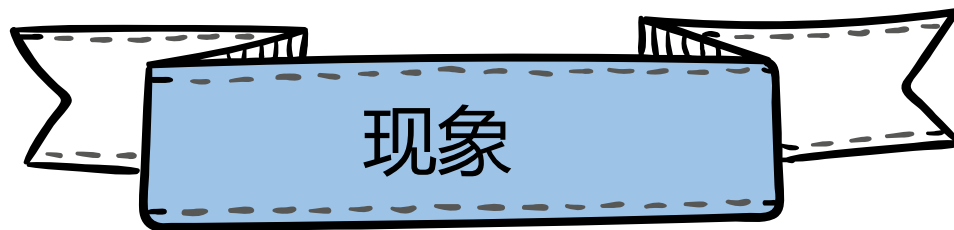
电机正转270度
【恒定电流】

舵机1先转90
再转回原角度

电机反转180度
Delay
【恒定电流】

电机反转90度
【恒定电流】

舵机1抬起90度
舵机2抬起90度
舵机1放回
舵机2放回



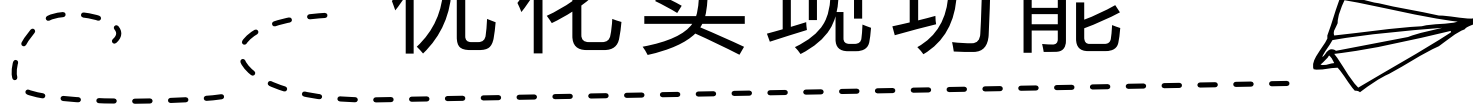
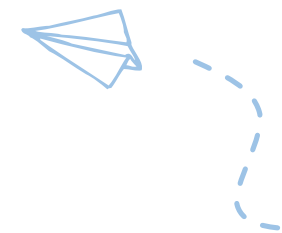
可达到偏转效果，但误差较大，并且需转速极小以保证二维码可被读出。

改进措施

采用PID进行角度闭环使转角更加稳定



优化实现功能





对应知识



1

舵机、电机
控制



2

PID控制



3

CAN与串口
特性

解密 - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
uint16_t test_b=1000;
int16_t Angle=0,pulse=0,t=0,X=80,Target_angle=90,Angular_velocity=10;
double PWM_time = 0;
int x = 90;
int d1=110;
int d2=60;
int d3=0;

int i=0;
int j=0;
int m=0;
uint8_t can1_send_data[8];
pid_t pid_test = {0};
float kp = 1,ki = 0.2,kd = 0;
```

```
void Usart3_callback(void)
{
  a=usart3_rcv[0];
}

void servo(uint8_t ID, double angle)
{
  set_pwm_group_param(PWM_GROUP1,20000);
  start_pwm_output(ID);

  double PWM_Time = 1500.0 + 600.0/x*angle;

  if(angle < 0)
  {
    PWM_Time = -PWM_Time;
  }

  set_pwm_param(ID,PWM_Time);
}
```

```
void execute_task(const void* argu)
{
    uart_init(USER_UART3,9600,WORD_LEN_8B,STOP_BITS_2,PARITY_EVEN);
    uart_rcv_callback_register(USER_UART3,Usart3_callback);
    uart_receive_start(USER_UART3,usart3_rcv,4);
    osDelay(1000);
    pid_init(&pid_test,600,0,kp,ki,kd);
    can_device_init();
    can_rcv_callback_register(USER_CAN1,can1_rcv_callback);
    can_receive_start();
    ... ..
}
```

```
for( ;i<=9600;i+=10)
{
    current=pid_calc(&pid_test,moto_test.total_angle,i);
    can1_send_data[6]=current>>8;
    can1_send_data[7]=current;
    write_can(USER_CAN1,0x1ff,can1_send_data);
    osDelay(5);
}
current=0;
can1_send_data[6]=current>>8;
can1_send_data[7]=current;
write_can(USER_CAN1,0x1ff,can1_send_data);
```

```
//DJ  
osDelay(1000);  
servo(PWM_IO2,d1);  
osDelay(50);  
servo(PWM_IO2,d1-95);  
osDelay(1000);  
servo(PWM_IO2,d1);  
osDelay(1000);
```

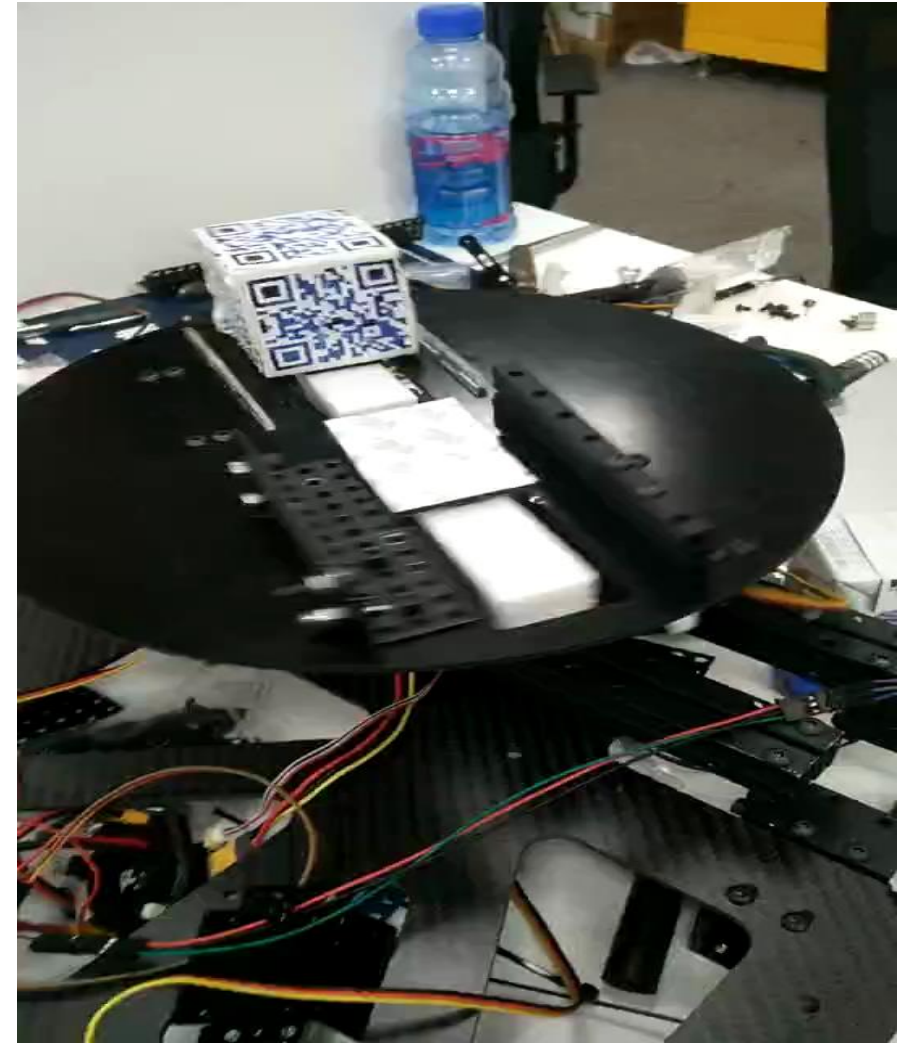


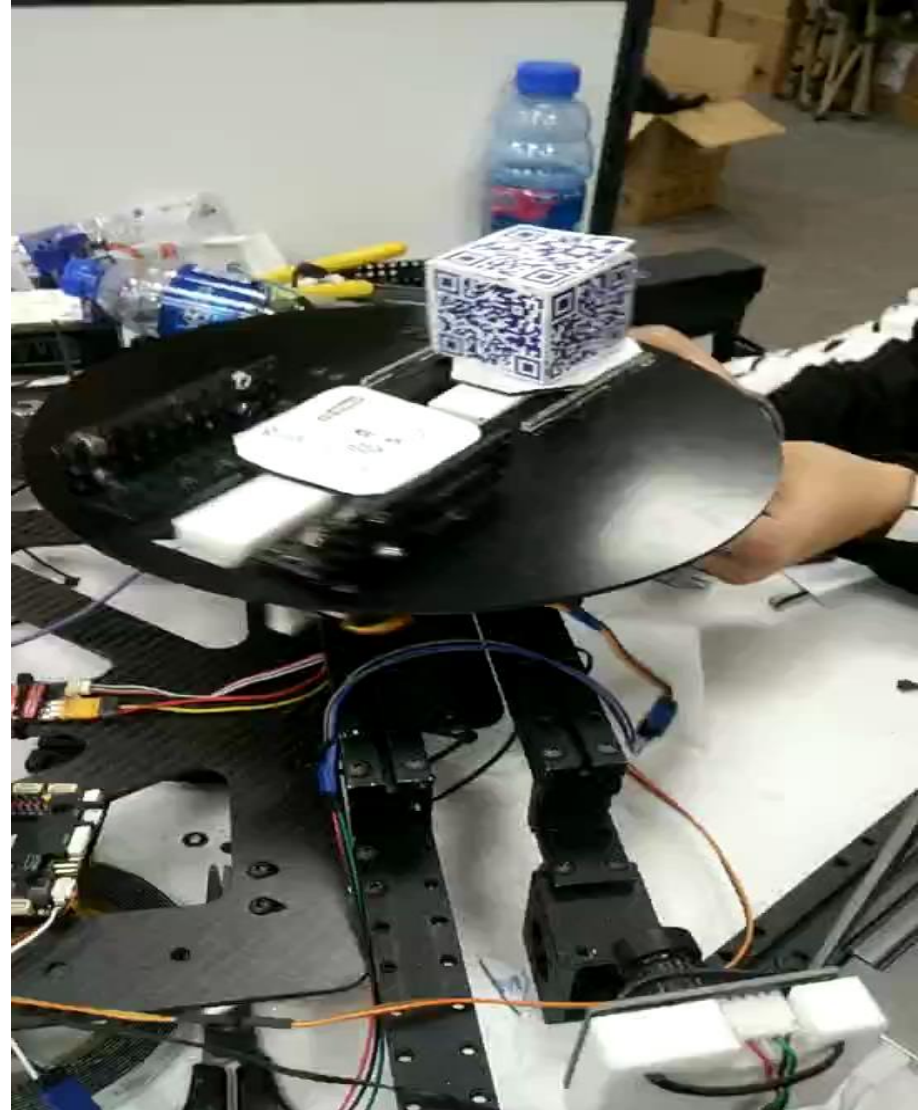
```
for(;i>=3240;i-=10)
{
    current=pid_calc(&pid_test,moto_test.total_angle,i);
    can1_send_data[6]=current>>8;
    can1_send_data[7]=current;
    write_can(USER_CAN1,0x1ff,can1_send_data);
    osDelay(5);
}
current=0;
can1_send_data[6]=current>>8;
can1_send_data[7]=current;
write_can(USER_CAN1,0x1ff,can1_send_data);
```



装置完善改进







需修改细节

1

为解决机械结构在两拨杆之间的缺陷，采用搭桥方式，使物块顺利被拨出
【且调整舵机速度，保证物块被较精准拨动】

2

为减少工兵机器人在解密区停留过长时间，机械在圆盘下部设置斗，在第一个物块扫描结束后将斗中物块翻上
【翻上时速度要快，以免物块被带下】

3

为解决工兵机器人失误，或误拨动导致串口检测a恒为0或恒为1，设想增加微动开关，以增加扫描条件

BFS-based Pathfinding Algorithm

Presented by 鄺磊 (Kuang, Lei) / KLsz
from the 6th group

Main Function

Find the better way to collect blocks and tell the STM32 to control the robot.

Main Logic

1. Preload the map of all situations to the program;
2. Find the shortest path to the next position dynamically with the BFS algorithm;
3. Check whether the robot would pass the door before it rotate, and if it wouldn't, re-find the shortest path using the map for the rotated situation;
4. Tell the STM32 to go, step-by-step, when STM32 tells me it has finished the most recent move.

Main Logic Implement

```
void start() noexcept {
    if(num_boxes_on_car == MAX_NUM_BOXES_ON_CAR) aimed_pos = read_pos;
    else if(!is_secret_key_visited) aimed_pos = secret_key_pos;
    // ...
}
void Route() noexcept {
    graphs[PLATE_TOGGLE_SITUATION].Route_getRoute(step_stack, end_pos);
    // ...
}
void Run() noexcept {
    // ...
    while(!step_stack.empty()) {
        write(static_cast<uint8_t*>(step_stack.top()));
        step_stack.pop();
    }
    // ...
}
```

Advantages and Disadvantages

(Advantages)

- Possible to be fully automated;
- Efficient when semiautomated;

(Disadvantages)

- Requiring the robot to move accurately from pixel to pixel;
- Requiring at least an ARMv7 device to reach the minimum requirement on calculation and memory.

Technical Key Point

- Read map using File I/O commands;
- Pathfinding using BFS (Breadth-First Search) algorithm;
- Predict the door rotating situation when arriving at the door by estimating time;
- Communicate with STM32 using built-in Serial;
- Build the whole system on Linux aarch64 platform.

Key Point BFS Implement

```
    // ...
public:
    void Count_setStartPos(pos_t& count_start_pos) noexcept;
    // ...
public:
    size_t Count_getStepCount(pos_t& count_end_pos) noexcept;
    size_t Count_getDoorCount(pos_t& count_end_pos) noexcept;

    pos_t Route_findFirstDoor(pos_t& route_end_pos) noexcept;
    void Route_getRoute(std::stack<Step>& step_stack, pos_t& route_end_pos) noexcept;
    // ...
public:
    tmr_t Route_getTime(std::stack<Step> step_stack) noexcept;
    // ...
```

Key Point Serial Implement

```
void Init() {  
    if(!ser.Init()) FatalError(static_cast<err_t>(2));  
}  
  
void Write(uint8_t* data) {  
    data[0] += 48; data[1] += 48;  
    if(ser.write(data, 2) != 2) {  
        FatalError(static_cast<err_t>(3));  
    }  
}  
  
uint8_t Read() noexcept {  
    uint8_t data[1];  
    ser.Read(data, 1);  
    return *data;  
}
```

License

The presentation is licensed under CC BY-SA 4.0;
The code is licensed under GPLv3, while DJI is authorized to have another full access to the code;
CopyLeft KŁsz 2019.

STM32 is a registered trademark of STMicroelectronics International N.V..

Thank you for listening.

Please welcome 周子皓 (Zhou, Zihao)
to introduce the implement of the Decoder.

解 密 算 法

6组 周子皓



目录页

01
OPTION



解密例程分析

02
OPTION



解密例程改进

03
OPTION



串口通信

04
OPTION



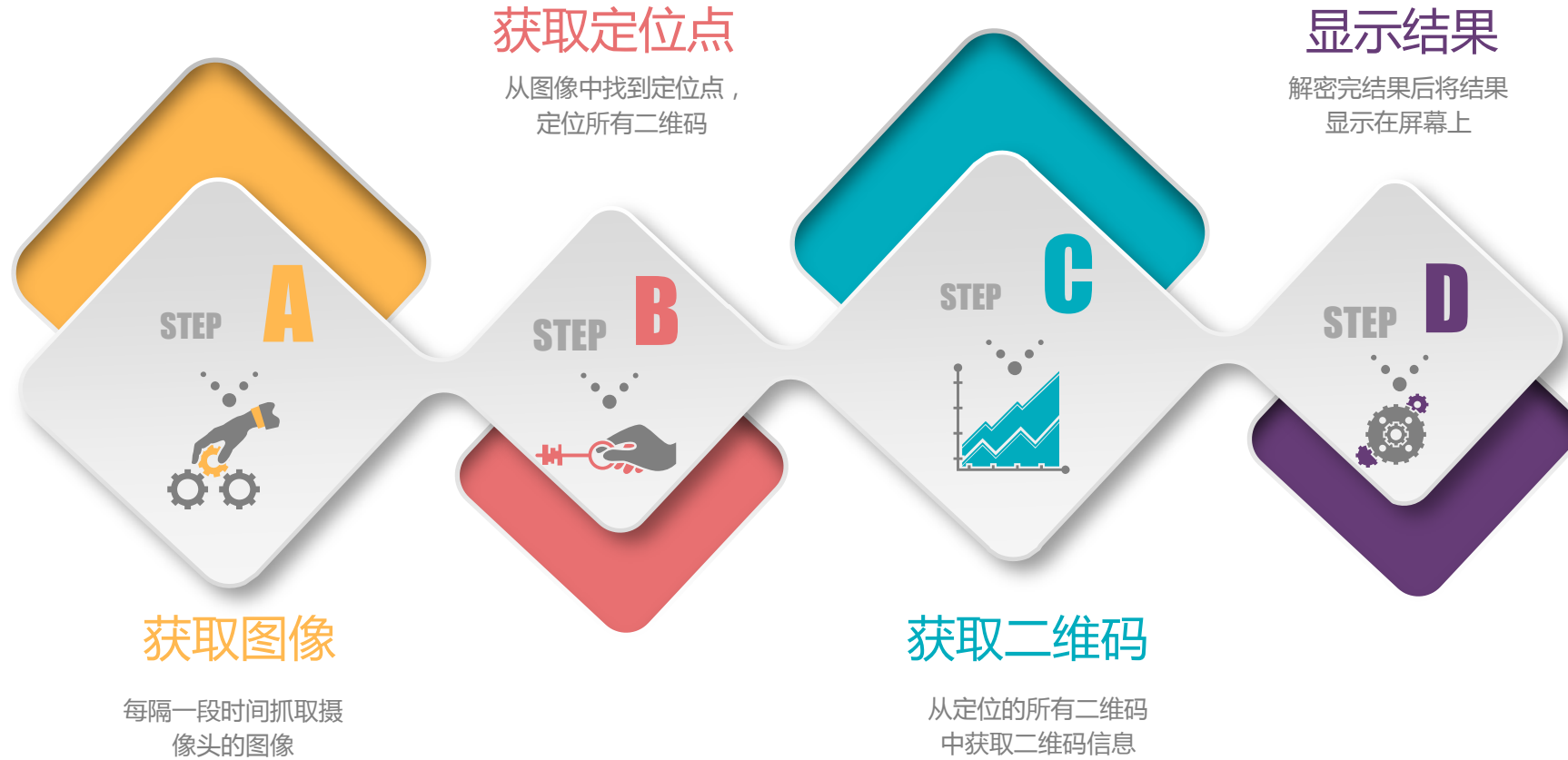
地图导入

01
OPTION

解密例程分析



例程功能分析



例程缺点分析



BUG

解密失败的时候会卡死，必须先扫描密钥方块才能正常运行



操作复杂

完全解密完一个密文方块必须要再次扫描密钥方块才能继续解密下一个密文方块

01



错误扫描

有小概率出现扫到了空文本的情况

02

03

显示有限

只能同时显示一组结果

04

02
OPTION

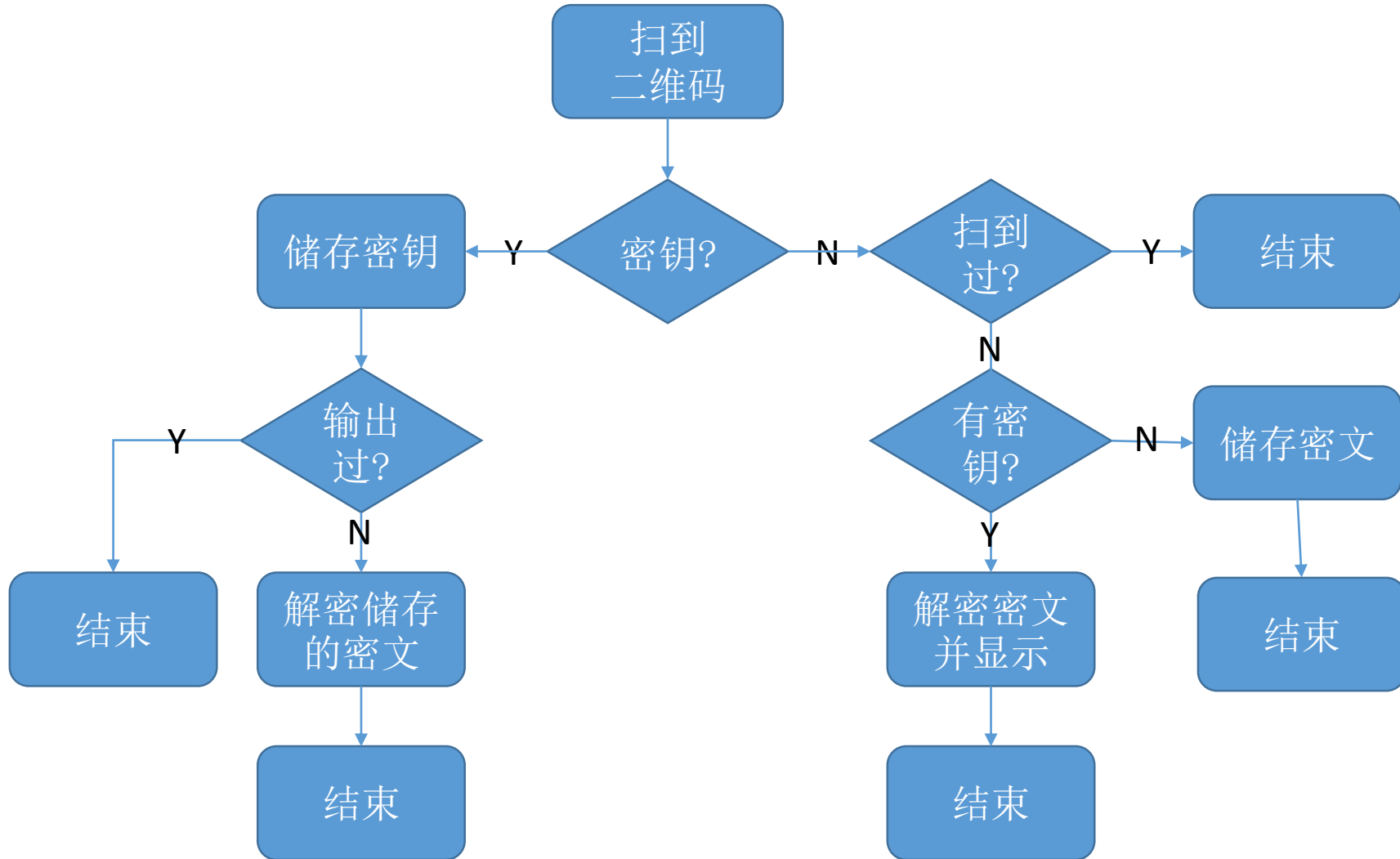
解密例程改进



解决空字符串

```
while(symbolCount--)  
{  
    uint8_t*base64_decode_hex ;  
    //输出二维码内容  
    std::string origin_str = symbol->get_data();  
    const char* origin_char_str = origin_str.c_str();  
    std::cout<<"text:"<<origin_str<<" "<< std::endl;  
    if(origin_str == "")  
        continue;  
    result[0] = '1';  
    size_t base64_decode_len = 0;  
    size_t final_str_len = 0;  
    if(memcmp(origin_char_str, "key:", 4) == 0)  
    {  
        ...  
    }  
    else  
    {  
        ...  
    }  
}
```

解决顺序问题



显示多组结果

```
203 int group(int x)
204 {
205     if(x>=-1&&x<=5) return 0;
206     if(x>=6&&x<=11) return 1;
207     return 2;
208 }
```

返回该字符串属于第几组

```
458 text.putText(resultFrame, final_str + 2, cvPoint(50, 30 * str_index + 50 + 200 * group(word - 1)), CV_RGB(0, 0, 0));
```

```
349 resultFrame = cvCreateImage(CvSize(800, 800), IPL_DEPTH_8U, 3);
```

增大结果
框大小

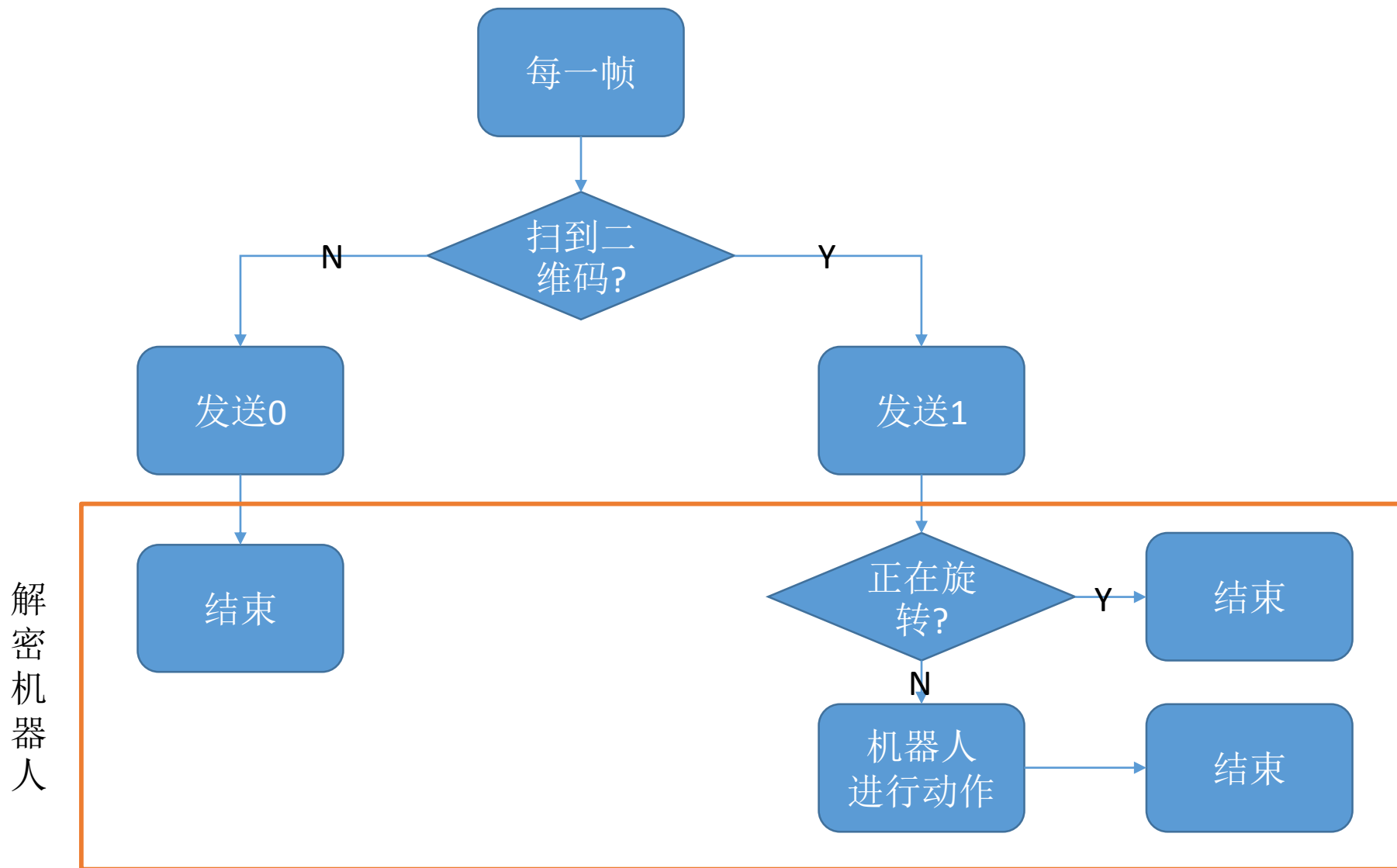
变量word储存该
密文是第几个

03
OPTION

串口通信



串口通信



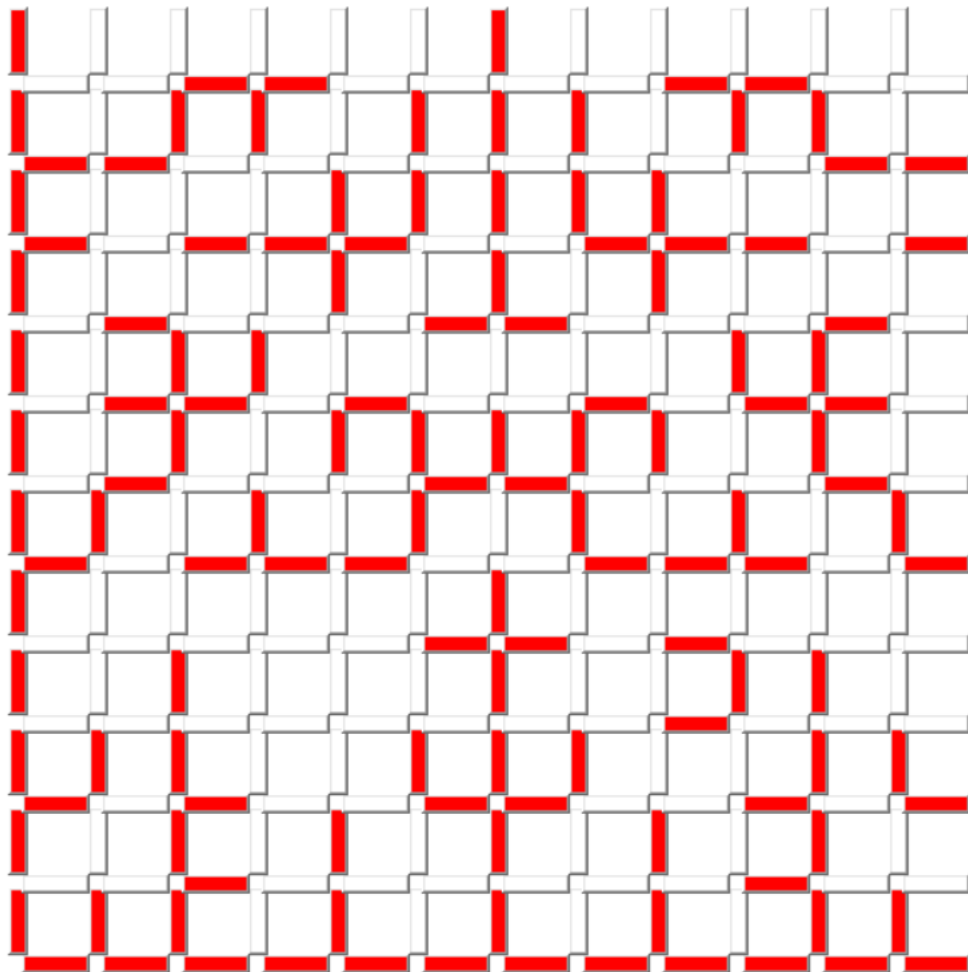
04
OPTION

地图导入



地图导入

RM2019Winter



```
333232323233  
103010101210  
313003310213  
101000102110  
100002302000  
312321032303  
121013311030  
123120020330  
120012301020  
302231133202  
321101110132  
102200102200
```

输出

