



西安电子科技大学
XIDIAN UNIVERSITY

详细设计说明书

第 32 组-多功能签到系统

任俊杰 (组长)

刘旭森

韩一畅

张珈齐

目录

1 引言	5
1.1 编写目的	5
1.2 背景	5
1.3 定义	5
1.4 参考资料	5
2 程序系统的结构.....	7
3 程序 1(FACEREC()、READ_DATA_FROM_XML())设计说明	9
3.1 程序描述	9
3.2 功能	9
3.3 性能	9
3.4 输入项	9
3.5 输出项	10
3.6 算法	10
3.7 流程逻辑	10
3.8 存储分配	10
3.9 限制条件	10
3.10 测试计划	10
4 程序 2 (GET_TIME()、CACLU_TIME_DIFFER()) 设计说明	11

4.1 程序描述	11
4.2 功能	11
4.3 性能	11
4.4 输入项	11
4.5 输出项	11
4.6 流程逻辑	11
4.7 存储分配	12
4.8 限制条件	12
4.9 测试计划	12
5 程序 3(GET_ANG()、GET_MULTI_FACE_ANG())设计说明	12
5.1 程序描述	12
5.2 功能	12
5.3 性能	12
5.4 输入项	12
5.5 输出项	13
5.6 算法	13
5.7 流程逻辑	13
5.8 存储分配	14
5.9 限制条件	14
5.10 测试计划	14

6 程序 4(NEW_FACE())设计说明	14
6.1 程序描述	14
6.2 功能	14
6.3 性能	14
6.4 输入项	14
6.5 输出项	14
6.6 算法	15
6.7 流程逻辑	15
6.8 存储分配	16
6.9 限制条件	16
6.10 测试计划	16

详细设计说明书

1 引言

1.1 编写目的

该文档在概要设计的基础上，进一步的细化系统结构，展示了软件结构的图标，物理设计、数据结构设计、及算法设计、详细的介绍了系统各个模块是如何实现的，包括涉及到的算法，逻辑流程等。

由于硬件部分和物联网部分涉及知识较多，且程序调用较为简单，故只对人脸部分进行说明。

预期的读者：程序员

1.2 背景

- a. 待开发软件系统的名称：多功能签到系统
- b. 项目的任务提出者：任俊杰
- c. 项目的开发者：任俊杰
- d. 项目的用户：各高校或公司的全体学生或员工
- e. 运行该软件的计算站（中心）：单独终端，树莓派 4b 或 X86 开发板

1.3 定义

树莓派：Raspberry Pi，只有信用卡大小的微型电脑，其系统基于 Linux。

Blinker：一个物联网平台，提供免费的 APP 接入，开发简单可靠。

阿里云：物联网平台提供安全可靠的设备连接通信能力，支持设备数据采集上云，规则引擎流转数据和云端数据下发设备端。专业的物联网平台，开发较为复杂。

虹软人脸识别 SDK：虹软公司提供的免费人脸识别服务，支持人脸检测、人脸识别、活体检测、年龄预测。

OpenCV：一个开源的计算机视觉库，现已被广泛应用于各种场合。

1.4 参考资料

[1] 卜秋月. 基于 OpenCV 的人脸识别系统的设计与实现[D]. 吉林: 吉林大学, 2015.

[2] 孙志. 基于 OpenCV 的人脸识别算法实验平台研究与实现[D]. 吉林: 吉林大学, 2014.

[3] 公衍宇. OpenCV 在人脸门禁系统的研究与应用[D]. 河北:河北工业大学, 2012.

[4] 雷静. 基于 OpenCV 的人脸跟踪识别系统研究[D]. 西安:西安电子科技大学, 2010.

[5] 麦敬椿. 复杂背景下基于神经网络的人脸识别方法研究与实现[D]. 广州:华南理工大学, 2014.

[6] 江泓政, 汤军, 黄建, 等. 基于人脸识别的课堂点名系统 [J]. 微型机与应用, 2016, 35(12):13-15.

[7] 陈凯文, 文进宇, 黄涛, 等. 基于 OpenCV 的人脸识别门禁系统的设计与实现[J]. 电脑与信息技术, 2016, 23(6):1005-1228.

[8] Xianghua Fan, Fuyou Zhang, Haixia Wang, Xiao Lu, et al. The System of Face Detection Based on OpenCV[J]. Chinese Control and Decision Conference, 2012(24):648-651.

[9] M Hassaballah Kenji, Murakami Shun. Face detection evaluation: a new approach based on the golden ratio[J]. Journal Citation Reports, 2013(7):307-316.

[10] O Herbawi, J Teeti, Y Hmeed. Raspberry Pi and Computers-Based Face Detection and Recognition System[J]. International Conference on Computer and Technology Applications, 2018(4):171-174.

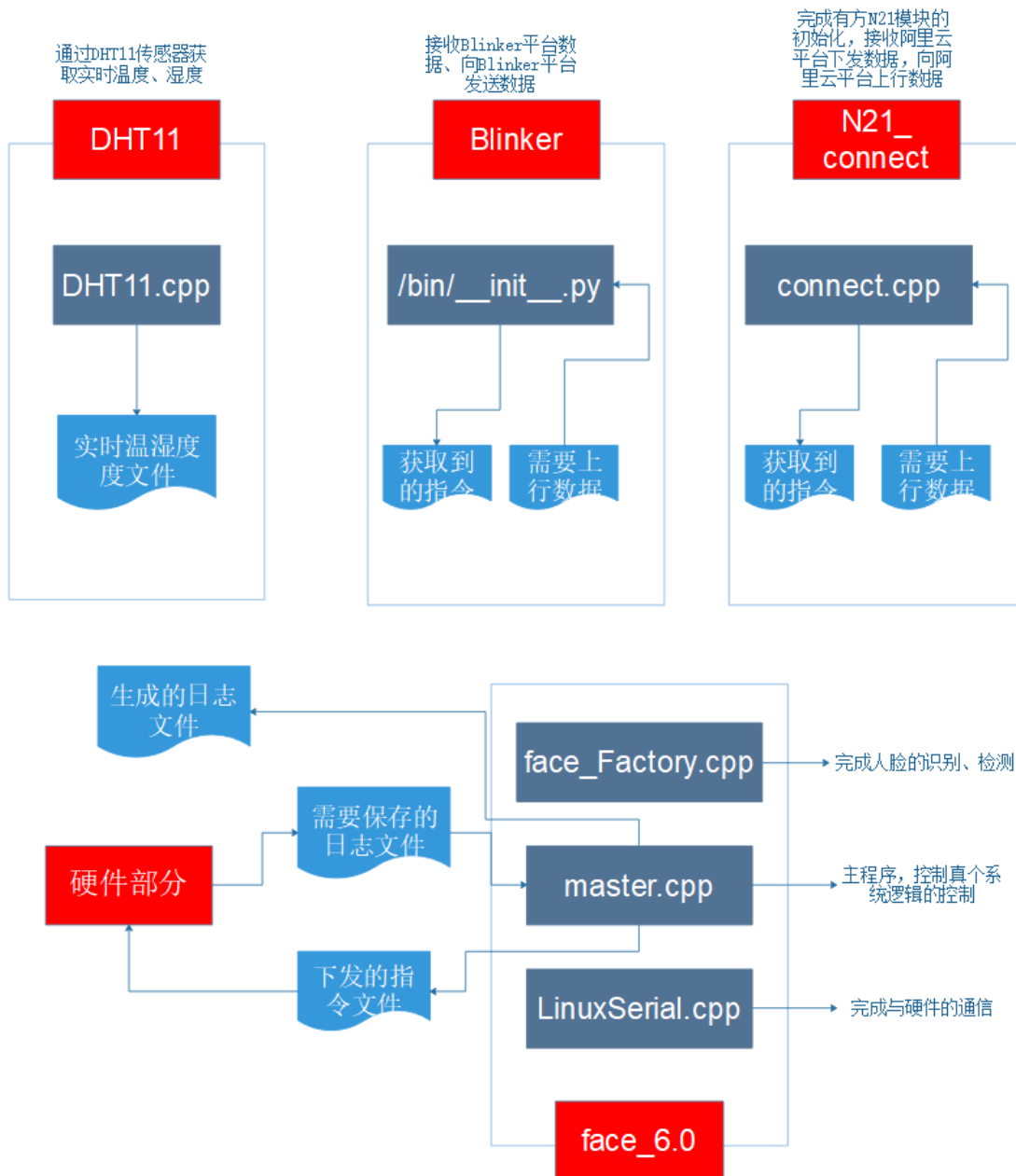
[11] Patrick Laytner, Chrisford Ling, Qinghan Xiao. Robust Face Detection from Still Images[J]

[12] HaiWu Lee, FanFan Peng, XiuYun Lee. Research on Face Detection under Different Lighting[J]. Proceedings of IEEE International Conference on Applied System Innovation, 2018:1145-1148

[13] 概要设计说明书 (G8567-88)

[14] 《软件工程》 张秋余等编著, 西安电子科技大学社出版 2014 年 12 月, ISBN 978-7-5606-3510-1/TP

2 程序系统的结构



由于硬件部分和物联网部分涉及知识较多，且程序调用较为简单，故只对 face_6.0 人脸识别模块中的 face_factory.cpp 部分进行说明。

```
class FaceRec{
public:
    FaceRec(){
        read_Data_From_Xml();
        last_Data_Send_Time = "2019-10-11----9-06-35";

        std::ofstream arduino_Receive("../receive_arduino_data/ARDU
INO_COMMAND.txt");
```

```

        std::ofstream blinker_Receive("../../BLINKER/db/BLINKER_COM
MAND.txt");
        ofstream NEWEST_DATA("../../BLINKER/db/NEWEST_DATA.txt");
        arduino_Receive << 'X' <<endl << 0;
        blinker_Receive << 'f' <<endl << 0;
        NEWEST_DATA << "[2019-10-17---19-55-55]-[Name of Unlocker]
:1-[Unlocking mode]:人脸解锁"<<endl<<"[Door State]:0"<<endl<<0;
        arduino_Receive.close();
        blinker_Receive.close();
        NEWEST_DATA.close();
    }
    cv::Point3f get_Ang(int center_x,int center_y,int width);

    string get_Time();

    int caclu_Time_Differ(string tmp1,string tmp2);

    bool get_Multi_Face_Ang(ASF_MultiFaceInfo &detectedFaces,std::v
ector<cv::Point3f> &faceAngle,float &minDis);

    bool new_Face();

    bool delete_Face();

    int faces_Db_Comparsion(ASF_FaceFeature &feature);

    int recongnise_Face(bool loop_flag);

    bool init_Engine();

    bool save_Data_To_Xml();

    bool read_Data_From_Xml();

    void mark_faces(cv::Mat &img,ASF_MultiFaceInfo &facesPosData,st
d::vector<cv::Point3f> facesAngle,cv::Scalar color);

    void mark_faces(cv::Mat &img,string label,cv::Rect _position);

    void record_unlock_message(string unlock_name,string unlock_way
,string door_state);

    void record_Recon_Data(string str,cv::Mat face_record);

```



```

void printAllDatas();

int size(){
    return FACES_StoredData.size();
}

MHandle handle = NULL;
MRESULT res;

bool faceReconFlag = 0;
private:
    string last_Data_Send_Time;
    std::vector<FACE_SingleData> FACES_StoredData;
};

```

3 程序 1(FaceRec()、read_Data_From_Xml()) 设计说明

3.1 程序描述

整个自定义人脸识别类的构造函数，主要用于读取上次程序意外关闭保存的人脸数据、初始化内部通信的几个文件，保证系统启动后可以正常运行。该类中的私有段包含了常驻在内存中的当前录入完毕的人脸数据，和上次信息发送的时间。

3.2 功能

读取之前保存在 xml 文件中的数据；
初始化各个模块间的通信文件。

3.3 性能

必须在短时间内读取 xml 中的数据，并保证读取的数据无差错。

3.4 输入项

无

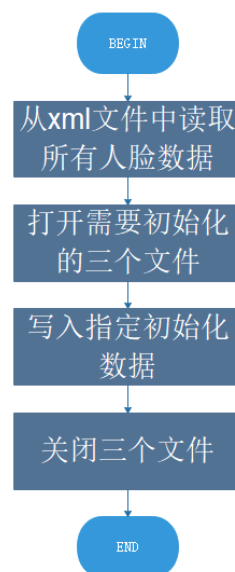
3.5 输出项

`std::vector<FACE_SingleData> FACES_StoredData;` 一个 `FACE_SingleData` 类型的容器，长度为上次录入的人脸数量。

3.6 算法

无算法，只是顺序执行。

3.7 流程逻辑



3.8 存储分配

一个 `FACE_SingleData` 类型的容器，长度不指定，根据需要确定。

3.9 限制条件

需要有相应的两个文件

3.10 测试计划

无需输入，直接运行，然后查看相对应生成的两个文件，看内容是否与程序中一致，内容一致且无报错则说明运行成功。

4 程序 2 (get_Time()、caclu_Time_Differ()) 设计说明

4.1 程序描述

get_Time():获取系统当前时间;

caclu_Time_Differ():计算两次开锁/录入的时间差,避免过快记录数据。

4.2 功能

获取系统当前时间、计算两次时间差。

4.3 性能

对 get_Time()无性能要求。

对 caclu_Time_Differ():需要保证计算精度在秒级别。

4.4 输入项

get_Time(): 无

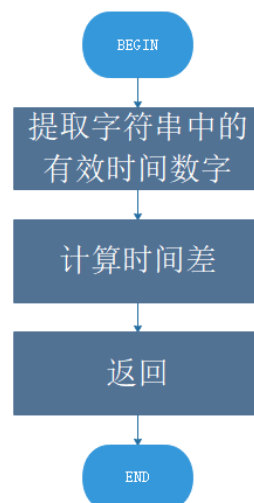
caclu_Time_Differ(): 两次时间的字符串,格式如下: "%Y-%m-%d---%H-%M-%S"

4.5 输出项

get_Time(): 当前时间的字符串,如: "2019-10-17---19-55-55"

caclu_Time_Differ(): 两个时间的时间差,单位为 s,范围 0-65535。

4.6 流程逻辑



4.7 存储分配

无

4.8 限制条件

计算时间差只能精确到月，假如上次发送时间是上月某天某时刻，则下个月同时刻则计算有误，但实际可能性较低。

4.9 测试计划

Get_Time(): 无需输入，打印输出，检查是否为当前系统时间。

caclu_Time_Differ(): 输入两个符合标准的时间字符串，打印输出，检查结果是否正确。

5 程序 3(get_Ang()、get_Multi_Face_Ang()) 设计说明

5.1 程序描述

get_Ang(): 通过 PNP 解算出单张人脸的三维坐标，用于对人脸识别距离的控制。

get_Multi_Face_Ang(): 获取整个图像所有人脸三维坐标。

5.2 功能

获取图像中人脸的三维坐标，得到距离设备最近的人脸距离，用于识别的距离控制。

5.3 性能

识别误差不能超过 5cm，对于多人脸具有较高的识别效率。

5.4 输入项

一帧中所有检测到的人脸坐标，即人脸中心位置图像坐标和图像宽度。

5.5 输出项

`std::vector<cv::Point3f> &faceAngle`: 一个 `Point3f` 类型的容器，长度为图像中检测到的人脸个数。用于存储对应于每张人脸的三维坐标。

`float &minDis`: 该图像中距离设备最近人脸的距离。

5.6 算法

用到了基本的相机标定和单目相机姿态解算 PNP 的算法，由于算法及其复杂，只列出重要公式：

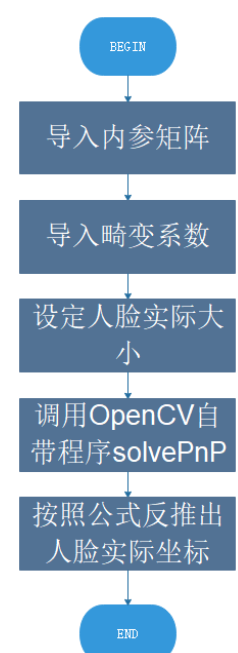
坐标变换公式：

$$\frac{1}{Z_c} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{S_x} & r & u_0 \\ 0 & \frac{f}{S_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R_{3 \times 3} & T_{3 \times 1} \\ O & 1 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = K_{3 \times 3} \cdot \begin{bmatrix} R_{3 \times 3} & T_{3 \times 1} \\ O & 1 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

消除畸变公式：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2p_1 xy + p_2(r^2 + 2x^2) \\ 2p_1(r^2 + 2y^2) + 2p_2 xy \end{bmatrix}$$

5.7 流程逻辑



5.8 存储分配

`std::vector<cv::Point3f> &faceAngle:` 一个 `Point3f` 类型的容器，长度为图像中检测到的人脸个数。用于存储对应于每张人脸的三维坐标。

5.9 限制条件

无

5.10 测试计划

运行程序，输入视频流，打印结果，看人脸位置变动是否会因人脸位置变动而变动，实际测量距离，判断与程序计算的距离是否一致。

6 程序 4(new_Face())设计说明

6.1 程序描述

用于录入人脸，执行该功能后，系统会自动检测人脸，检测到人脸即可以录入，录入完成之后自动放入人脸数据库。

6.2 功能

录入人脸

6.3 性能

最多的录入人脸数量可以由用户设定，需要保证录入的速度。

6.4 输入项

视频流中的许多帧图像

6.5 输出项

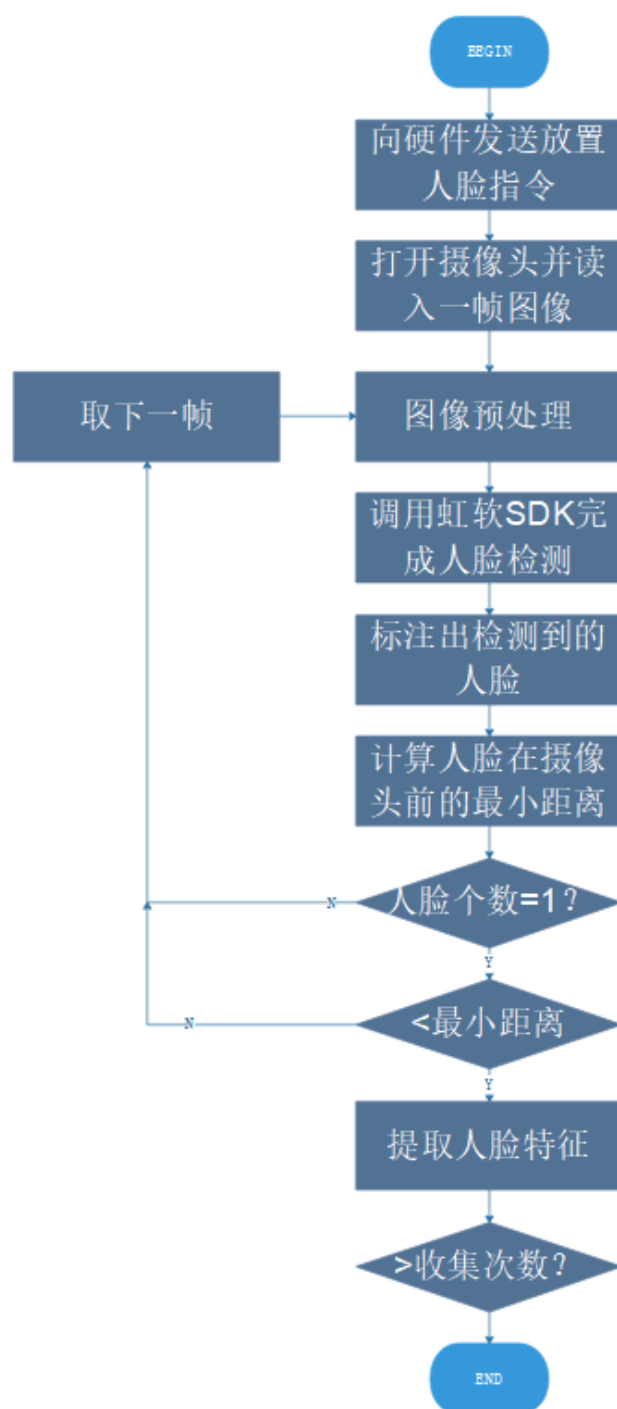
在上述提到的 `std::vector<FACE_SingleData> FACES_StoredData;` 中增添数据（即新录

入的人脸)。

6.6 算法

无

6.7 流程逻辑



6.8 存储分配

ASF_MultiFaceInfo detectedFaces = { 0 }：存储检测到的人脸位置数据

ASF_SingleFaceInfo SingleDetectedFaces = { 0 }：存储检测到的单张人脸数据

ASF_FaceFeature feature = { 0 }：存储单张人脸特征

6.9 限制条件

无

6.10 测试计划

正常启动程序，放置人脸到指定位置，等待程序自动录入完成后结束。查看.xml 文件中是否多出了新的人脸数据。