

基于ESP32的虚拟电子围栏设计与实现

杨子弘, 胡诚羽, 刘佳*

武汉商学院信息工程学院, 湖北 武汉

Email: 1921931219@qq.com, 1944534045@qq.com, *mikeliujia@wbu.edu.cn

收稿日期: 2021年6月20日; 录用日期: 2021年7月16日; 发布日期: 2021年7月26日

摘要

随着物联网技术的不断发展, 许多领域已经开始了新老技术的更迭, 安防领域也不例外, 基于对居家安全以及动物园区存在的安全隐患的考虑, 该文中设计出一款以ESP32为核心的虚拟电子围栏系统, 该系统能够实时检测经过红外传感器的物体并实现对相关数据的上云, 客户端利用该数据能够判别异常发生的地点。在通过具体的功能和性能测试后, 虚拟电子围栏系统能够实时判别是否有物体通过, 同时它也具有高效的数据上传能力以及较好的可靠性和稳定性。最终的系统能够强化传统动物园的安防手段, 进而解决目前动物园区发生意外后反应不及时的问题。

关键词

物联网, 物联网中间件, 传感器, 数据通信

Design and Implementation of Virtual Electronic Fence Based on ESP32

Zihong Yang, Chengyu Hu, Jia Liu*

School of Information Engineering, Wuhan Business University, WBU, Wuhan Hubei

Email: 1921931219@qq.com, 1944534045@qq.com, *mikeliujia@wbu.edu.cn

Received: Jun. 20th, 2021; accepted: Jul. 16th, 2021; published: Jul. 26th, 2021

Abstract

With the continuous development of Internet of Things technology, many fields have started the replacement of new and old technologies, and security field is no exception. Based on the consideration of home safety and the hidden danger existing in animal park, this paper designs a virtual

*通讯作者。

electronic fence system with ESP32 as the core. The system can detect the object passing through the infrared sensor in real time and realize the cloud of the relevant data. The client can use the data to identify the location of abnormal occurrence. After passing the specific function and performance test, the virtual electronic fence system can determine whether the object passes in real time. At the same time, it also has efficient data uploading ability and good reliability and stability. The final system can strengthen the traditional zoo security means, and then solve the problem of the current animal park after the accident response is not timely.

Keywords

The Internet of Things, Internet of Things Middleware, The Sensor, Data Communication

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

2021年5月6日,杭州转塘疑似发现豹子事件的发生,直到2021年5月8日杭州动物园确认发生安全事件结束,金钱豹外逃的事实就已经确认。此次的事件所带来的社会恐慌是不小的,由此事件可以看出畜牧场、动物园等园区在面临着日益复杂的社会环境的情况下,通过使用传统的安全防护技术手段已经不足以实现安全防护的任务,并且传统的安全防护手段不仅需要大量的安全保卫力量,防护效果也不尽如人意,金钱豹外逃事件就能够说明,传统的安防措施存在着反应速度慢、可靠性差等弊端。

近年来,在国外以 Langworthy Adam D [1]和 Southwest Farm Press [2]所发表和出版的文章为例,他们提出了一种电子围栏,这实际上是一种电力围栏,目的是为了限制和引导牧场中的动物在固定区域吃草,使其他区域的牧草正常生长,从而提升经济效益,其功能与本系统略有不同,本系统设立的目的是为了检测避开了其他安防手段的动物的动向,方便进行对动物的追踪。同样的,在国内以刘敬[3]和吴坤阳[4]等人所提出的虚拟电子围栏为例,其功能均与本系统的功能类似,但因应用领域不同,所以导致系统底层所采用的传感器也各不相同,刘敬的虚拟电子围栏基于摄像头进行,应用于工程施工现场或其他有可能坠落物体的区域,防止有人进入危险区域,而吴坤阳的虚拟电子围栏则是采用船舶 AIS (船舶自动识别系统)以及海事 VTS (船舶交通服务系统)等技术形成一种概念上的围栏,实现对船舶的安全监督工作。虽然在国内外暂未发现相同定位的设计方案,但这些定位于其他领域的类似应用仍然能够为本系统的设计提供一些参考意义。

基于当前社会对于畜牧场、动物园等园区的安全需求有待加强的考虑,文中提出并实现了一款虚拟电子围栏系统,可以从其根源上将其归类为智慧安防领域,这套系统能够检测避开了传统防护手段的动物,用于解决使用传统安全防护手段的畜牧场、动物园等园区所存在的无法实现24小时值守以及动物出逃后反应速度慢等问题,除此之外,本系统也可放置在家中,作为安防设备,对居室入侵进行预警和防范。

2. 系统总体设计

虚拟电子围栏系统以物联网系统四层架构中的下三层作为设计的总体框架,分别为感知层、传输层以及平台层,各层模块分布如表1系统各层模块分布表所示。

Table 1. System layer module distribution table
表 1. 系统各层模块分布表

工作层	工作模块
感知层	ESP32, 红外传感器, 温度传感器, 雷达传感器, 蜂鸣器
传输层	ESP32
平台层	基于 Python 与 TCP 协议的数据接收与存储云平台

以表 1 中的系统总体框架为基础, 本系统能够实现的功能为:

- 1) 实现对经过虚拟电子围栏物体的检测;
- 2) 在物体经过虚拟电子围栏时, 使用蜂鸣器发出警报;
- 3) 当周围环境温度达到设置的阈值时, 切换为雷达传感器进行检测;
- 4) 每间隔 5s 向云端上传一次节点数据;
- 5) 在数据上传成功后使蜂鸣器停止;
- 6) 云端实现对终端节点数据的接收、判别以及存储。

3. 系统硬件选型与设计

3.1. MCU 选型

本系统定位于智慧安防领域, 目的是为了在出现异常情况时, 能够快速的将该情况上传至预警中心, 做出一些防范措施。因此, 在进行 MCU 的选型时, 应当注意其功耗和体积等与应用场景密切相关的性能, 以及其是否板载 WIFI 模块。

结合本系统 MCU 应当具有的性能要求, 选出了两款 MCU 进行对比, 分别是零知 ESP32 与零知 ESP8266, 其余品牌的 ESP32 与 ESP8266 因其性能大都相似并且体积也略大一些, 故不作讨论。表 2 中结合零知官方公布的 ESP32 与 ESP8266 的相关参数进行了对比[5]。

Table 2. ESP32 and ESP8266 related parameters comparison table
表 2. ESP32 与 ESP8266 相关参数对比表

	ESP32	ESP8266
处理器	32 位双核(160~240 MHz)	32 位单核(80 MHz)
SRAM	520 KB	160 KB
ROM	448 KB	无
Wi-Fi (802.11 b/g/n)	HT20	HT20
以太网	10/100 Mbps	无
GPIO 引脚数	34	16
UART	3	2
ADC	18 (12-bit)	1 (10-bit)
工作电流	5 uA	20 uA

结合表 2 所展示的参数, ESP32 的大部分性能优于 ESP8266, 唯一的缺点是价格相比 ESP8266 略高一些, 但是以本系统的应用场景来看, 数据传输速度以及 GPIO 引脚数是系统所需要的, 更多的 GPIO 引

脚数能够支持更多的红外对射式开关传感器进行工作,而不需要重新设置一个模块。因此,本系统的 MCU 采用的是 ESP32。

3.2. 红外传感器选型

本系统中红外传感器主要需要实现的功能就是能够检测到是否有物体经过,因此,红外对射式开关传感器更加适合本系统,其内部的主要器件为热释电红外传感器。这种传感器相比于其他类似功能的红外传感器,功能单一、结构简单、功耗低以及价格低廉,其感应距离能够达到 1 m 至 1.2 m,工作电压的要求为直流 4.5 V 至 5.5 V,价格在 5 元左右,但其唯一的缺点就是在距离固定的情况下容易受到光源以及热源的影响。红外对射式开关传感器外形结构如图 1 所示,它拥有一个发射端和一个接收端,由发射端发出红外光束,接收端接收到红外光束时,输出端输出低电平,当有物体经过,未接收到红外光束时,输出端输出高电平。



Figure 1. Infrared light switch sensor
图 1. 红外对射式开关传感器

3.3. 温度传感器选型

在红外传感器选型中介绍了红外对射式开关[6]的主要器件是热释电红外传感器,但这种传感器容易受到光源、热源以及距离的影响。在光源较强的情况下,该传感器会失效,即使在理想光源以及固定距离的情况下,红外热释电传感器输出电压与温度呈线性关系[7],环境温度与经过物体温度相似时,传感器会短暂失效。因此,需要考虑在特殊情况下红外对射式开关会失效的情况,所以系统在设计过程中加入了雷达传感器与温度传感器,保证系统能够在红外对射式开关失效的情况下使用雷达传感器继续工作。

温度传感器应当满足低功耗、高可靠性的特点,在这里选出了两款温度传感器并结合其公布的参数进行了对比,如表 3 所示,分别是 DHT11 以及 LMT70 两种常见温度传感器。

Table 3. DHT11 and LMT70 temperature sensor performance comparison
表 3. DHT11 与 LMT70 温度传感器性能对比

	DHT11	LMT70
工作电压	3.3~5.5 V	2~5.5 V
工作温度	0°C~50°C	-55°C~150°C
温度测量精度	±2°C	±0.36°C
引脚	4	4
输出	数字输出	模拟电压值
低功耗	是	是

结合上面的参数对比,这两款温度传感器的参数比较上较为明显,但需要注意的是,DHT11 温度传感器不能长时间暴露在太阳光下或强烈的紫外线辐射中,长时间的暴晒会使性能降低[8],这一点缺点不符合本系统在动物园等园区的应用时的要求,而 LMT70 的限制则相对要小,因此在本系统中,温度传感器采用的是 LMT70。

但 LMT70 温度传感器不具有 DHT11 所具有的数字输出功能,因此需要对 LMT70 温度传感器所输出的模拟电压值利用温度转换公式进行换算。

3.4. 雷达传感器选型

雷达传感器选用的是 HLK-LD116-24G 毫米波雷达传感器,这是一款定位于安防、智能家居等领域的小型化雷达传感器,它的工作频率为 24G~24.25 GHz,在 5 V 供电的情况下,功耗为 55 mA,并且其不受温度、噪声、光暗等影响,抗干扰性强,这些特性均能够满足系统的要求。其他的雷达模块价格相对较贵,并且大多数的雷达模块定位的应用领域于本系统所应用的领域不同,故不展开深入讨论。

雷达传感器的优点较为突出,它的扫描范围呈现以该模块为顶点的圆锥,在保证雷达模块的准确性的情况下,并且该圆锥可高达 3 m,直径达 6 m,但是其缺点也是明显的,首先雷达模块的安装必须要相对稳定,如果发生抖动,则扫描范围内的物体发生相对移动,容易使雷达模块产生误识别,其次,雷达模块对于其扫描范围内的运动物体也有要求,当物体与雷达扫描区域发生切向移动时,容易判断失灵,正向移动则正常,如图 2 所示。

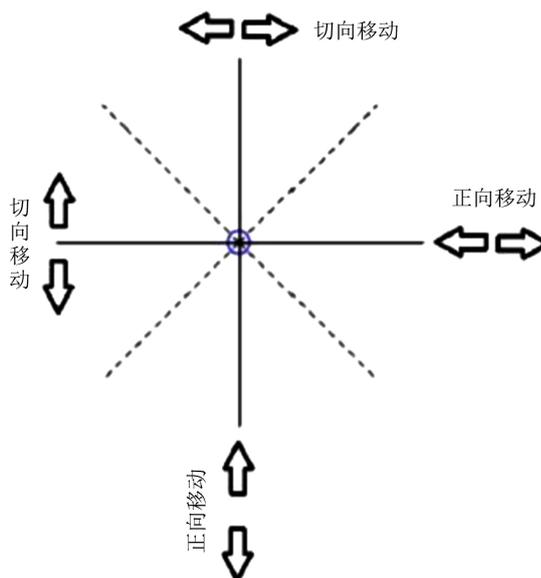


Figure 2. Forward and tangential movement diagrams

图 2. 正向移动与切向移动图示

因此,结合雷达传感器的特点,它只能作为本系统的备用计划存在,当红外对射式开关发生故障时,由雷达传感器代替检测。

3.5. 接线设计

本系统中需要使用的传感器过多,因此,为了防止过载的情况,本系统中加入了一个电源模块,该模块可以使用一块 12 V 的锂电池供电,并产生 15 路 5 V 以及 3.3 V 的输出。

在进行接线的过程中, 需要考虑到 ESP32 引脚复用的问题, 将蜂鸣器以及红外对射开关接在 ESP32 的普通 GPIO 引脚上, 留出其他复用了 ADC、串口、定时器等功能的引脚, 方便其他复杂扩展模块的接入。系统具体接线设计如图 3 所示。

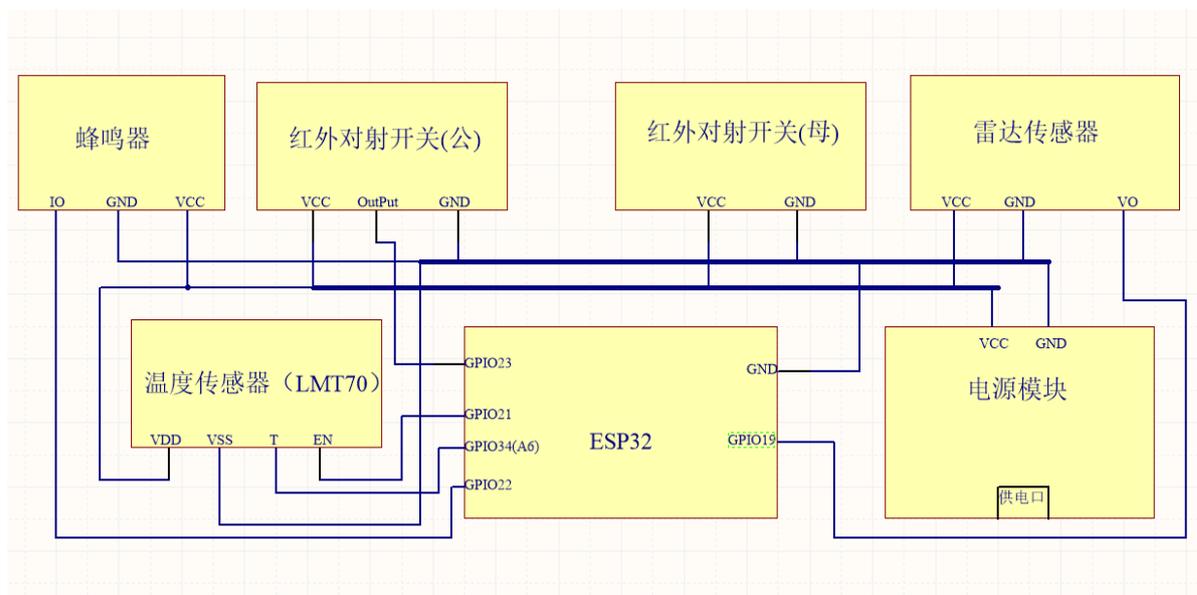


Figure 3. System wiring design
图 3. 系统接线设计

4. 系统软件设计

4.1. 驱动程序设计

驱动的开发完全采用库函数进行, 避免了大量的底层驱动的编写, 大大缩短了功能开发的时间成本。

为了保证低功耗, 应当在软件层面上在保证功能可靠的同时降低芯片对各个外设的读取频率, 并控制数据上传的时间间隔[9]。

对经过物体的检测使用红外传感器实现, 此种传感器的数据引脚会有高低电平的变化, 因此只需要读取相关 GPIO 口即可实现控制, 但为了防止电流毛刺引起的高电平, 程序中加入了防误触机制, 只有在同一时刻连续触发 5 次以上, 才会引起蜂鸣器的鸣叫, 提示有异常情况的发生。

系统的预警功能通过蜂鸣器实现, 具体的控制采用与其连接的 GPIO 引脚, 当有异常情况时, 会控制引脚输出高电平, 发出预警, 只有当数据发送函数将此次异常状态成功发送, 才会将蜂鸣器复原。

数据发送函数使用软件定时器进行控制, 时间间隔为 5s, 该函数在成功或者失败的发送了一次数据后, 会更改数据发送状态变量的值, 在该变量被置位的情况下, 表示数据发送失败, 预警功能中蜂鸣器的鸣叫由该变量进行控制。数据发送函数的定时器定义如下:

```
tickerSendMessage.attach_ms(5000, send_message,0);
```

LMT70 的输出信号为模拟电压值, 在使用 ADC 模块采集了模拟电压值之后, 需要使用 LMT70 数据手册中给出的公式进行温度换算。在进行数据采集的实现过程中, 采用了三阶温度换算公式:

$$T_M = a(V_{TAO})^3 + b(V_{TAO})^2 + c(V_{TAO}) + d$$

其中, $a = -1.809628E-09$, $b = -3.325395E-06$, $c = -1.814103E-01$, $d = 2.055894E+02$ 。

对于 LMT70 温度传感器的数据采集为 5s 一次，其定时器的定义如下：

```
tickerCollectTemperature.attach_ms(5000,collect_temperature,0);
```

最终程序能够实现的功能如下

- 1) 能够检测是否有物体经过；
- 2) 读取 LMT70 输出的模拟电压，并转换为实际温度值；
- 3) 实现在不同温度下使用不同的传感器进行检测；
- 4) 每 5s 发送一次节点数据。

驱动程序流程图如图 4 所示。

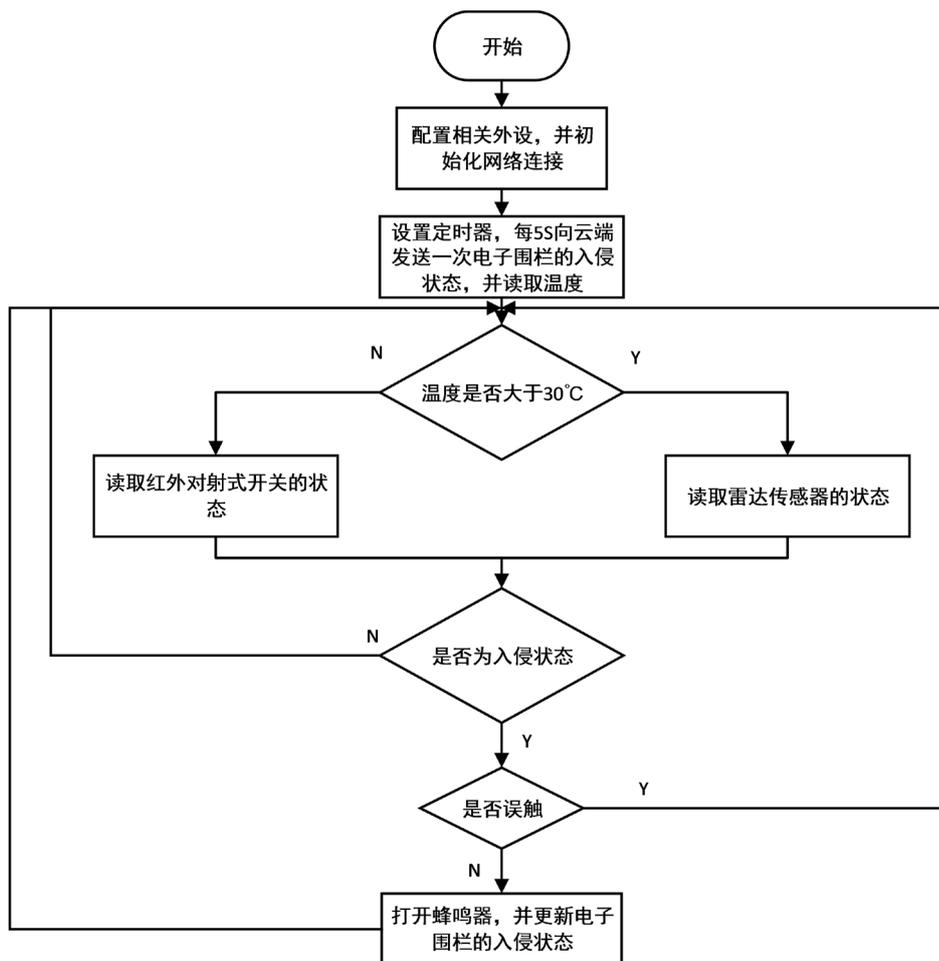


Figure 4. Driver flow chart

图 4. 驱动程序流程图

4.2. 数据发送函数设计

数据发送函数作为系统的数据上报模块，应当保证采用的数据的格式充分的简单和清晰，并且方便提取数据，基于这一点考虑，在进行数据封装时采用了 JSON 格式，JSON 格式是一种轻量级的数据交换格式，它的结构清晰，并且独立于编程语言，即使用其他的编程语言也能够进行数据的解析，这对于本系统的云端平台来说是必要的。

系统的通信协议选择的是 TCP 协议，在目前的阶段相较于 MQTT 协议更加简洁，方便设计初期的测试工作，如果采用大量的节点进行测试，可以使用 MQTT 协议。

函数中封装的数据结构如下：

```
typedef struct sensorData {
    String UID;           //设备编号
    int STATUS;          //电子围栏的入侵状态
    int TEMPERATURE;    //温度
}SensorData;
```

在进行具体的实现过程中，采用了第三方库 ArduinoJson，能够直接实现将本地数据打包成 JSON 数据格式的过程。

本系统的应用场景中所存在的不确定因数较多，在数据发送的过程需要考虑到来自各种环境因数的影响，例如网络波动、断网等，因此在程序设计的过程中需要加入异常捕获机制，避免因为网络问题导致的系统崩溃。在数据进行发送前检测与服务器的连接状态，可以确保数据的成功发送，具体如下：

```
if(connect_status == 0){
    if (!client.connect(host, Port)){
        connect_status = 0;
        return;
    }
}
```

4.3. 基于 Python 与 TCP 协议的数据接收与存储云平台设计

该平台设计的核心是对来自终端节点的数据的接收、处理以及存储。

平台的实现采用了第三方库 json 以及 pymysql，通过这两个库，能够实现对 JSON 数据的解码、键值对的提取以及 MySQL 数据库的访问。

在平台的设计过程中加入了数据接收的异常捕获机制、数据过滤模块、设备身份识别模块、数据存储模块等，实现了对数据的接收、过滤以及存储。

其中数据过滤模块的主要目的是为了防止非法数据的大量传入而导致大量的预警信息的产出，主要是通过 UID 所划分的范围以及 UID 中所携带的校验字段实现。

该平台基于 Python 进行开发，工作在 Linux 系统上，它最终能够实现的功能如下：

- 1) 实现对数据的接收；
- 2) 实现对 JSON 数据的解析，并将非法数据丢弃；
- 3) 提取 JSON 数据中相关数据并存入 MySQL 数据库。

因代码量较大，这里只对其工作流程进行展示，如图 5 所示。

4.4. 数据库设计

本系统的数据库编码采用 UTF-8，排序规则采用 utf8_general_ci。

为了保证终端节点在故障之后能够发现其故障时间，以及意外情况发生时的发生时间，在进行数据表的设计时，加入了日期和时间，方便对历史数据的查看。具体表结构如表 4 所示(UID 为设备识别号)。

本系统在安装过程中，通常会在一个固定的位置，因此客户端可以在异常情况发生时，根据设备的 UID 号来判断异常发生的地点。

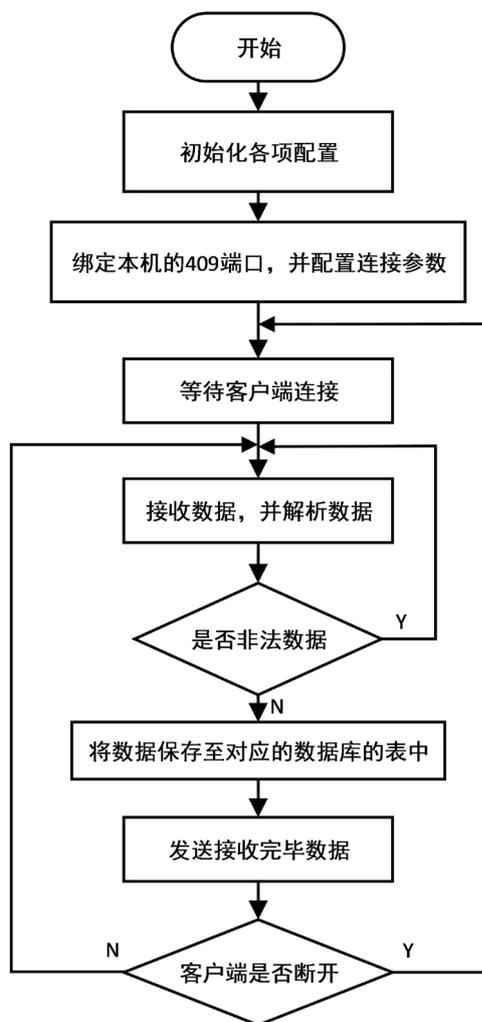


Figure 5. Cloud platform workflow
图 5. 云平台工作流程

Table 4. Data table structure
表 4. 数据表结构

Field	Type	NULL	Key	Default
DATE	date	No	No	NULL
TIME	time	No	No	NULL
UID	varchar(8)	No	No	NULL
STATUS	int(1)	No	No	NULL
TEMPERATURE	Int(2)	No	No	NULL

5. 系统测试

5.1. 正常测试

正常情况下, 室温 5°C, 下午 17 点, 正常亮度, 红外光电开关间隔 1 m。

Table 5. Normal test results**表 5.** 正常测试结果

	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次
入侵状态	0	1	0	1	1
温度	5	4	5	4	5

在正常情况下, 系统测试实验一共进行了 30 次, 没有出现过误识别。表 5 中的数据是由 30 次实验结果中抽取的, 图 6 为云端数据库中所接收到的数据, 每 5 s 一次, 其中第四列为入侵状态(0 表示入侵, 1 表示正常), 第五列为温度。

2021-03-11	17:33:46	00100001	0	5
2021-03-11	17:33:51	00100001	1	4
2021-03-11	17:33:56	00100001	0	5
2021-03-11	17:34:01	00100001	1	4
2021-03-11	17:34:06	00100001	1	5
2021-03-11	17:34:11	00100001	1	5

Figure 6. Data received in the cloud database**图 6.** 云端数据库中所接收的数据

5.2. 特殊情况测试

系统的应用场景所面对的主要变量为光照以及温度, 因此在特殊情况中只对光照以及温度两个变量进行测试[10]。

Table 6. Test results when changing the intensity of the light**表 6.** 改变光照强度时的测试结果

	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次
入侵状态	0	1	1	1	0
温度	4	4	4	5	4

系统在光照改变的情况下进行了若干次实验, 这里截取了如表 6 所示的结果。在第 1 次实验中, 手电被置于系统上方 1 m 处, 系统能够正常识别入侵。在第 5 次实验中, 手电筒被贴在红外光电开关的发射端, 此时发射端周边受光影响, 全部变白, 红外对射式开关发生误识别, 但此种情况较为极端, 多半为人的主观行为。

Table 7. Test results when the temperature is changed**表 7.** 改变温度时的测试结果

	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次
入侵状态	0	1	1	0	1
温度	25	28	30	32	31

如表 7 所示, 在改变温度的情况下(只是加热了温度传感器), 截取了温度上升过程中的 5 次测试结果, 在第 1 次实验中, 温度为 25℃, 红外对射式开关能够正常识别入侵。在第 4 次实验中, 温度为 32℃, 雷达模块能够正常识别入侵。

5.3. 测试总结

通过正常测试和特殊情况的测试结果来看:

- 1) 本系统在正常情况下具有良好的可靠性, 在一些较为特殊的情况下也能够正常工作。
- 2) 当光照强度过高时, 可以考虑加入光照传感器, 设定阈值然后使用雷达传感器进行识别, 但测试中所出现的极端情况在实际应用过程中出现几率较小。
- 3) 在温度测试中, 虽然没有改变环境温度, 但最终结果均是切换为雷达模块进行识别, 从测试结果来看, 在保证雷达模块位置固定的情况下, 检测结果准确。

6. 结语

本系统采用 ESP32 作为主控芯片, 使用红外对射式开关检测到是否有物体经过, 并使用雷达传感器在红外对射式开关失效的情况下继续工作, 保证系统的可靠性。整个系统的需求和功能均在文中做出了详尽的分析和规划, 并对系统所需要的器件进行了对比选型。在完成整个系统的设计后, 进行了完整的性能测试。经过实际验证, 文中所设计的基于 ESP32 的虚拟电子围栏系统完成了最初的预期效果, 能够在大多数情况下正常工作。针对系统中仍然存在的不足, 在文中也提出了一些供参考的方案用来改进, 使得系统能够适应各种应用场景。相信智慧安防的相关产品在未来会在各种可能出现的安全问题中起到重要的作用。

参考文献

- [1] Langworthy, A.D., *et al.* (2021) Virtual Fencing Technology to Intensively Graze Lactating Dairy Cattle. I: Technology Efficacy and Pasture Utilization. *Journal of Dairy Science*, **104**, 7071-7083.
<https://doi.org/10.3168/jds.2020-19796>
- [2] Hehemann, W. (2019) Electric Fencing Saves Money, Controls Weeds on Pasture. *Beef*.
- [3] 刘敬. 基于视频监控的虚拟电子围栏系统设计研究[D]: [硕士学位论文]. 济南: 济南大学, 2020.
- [4] 吴坤阳. 虚拟“电子围栏”在船舶进出港报告监管中的应用[J]. *交通企业管理*, 2020, 35(2): 98-100.
- [5] DLGG. ESP8266、ESP32 和 ESP32-S2 对比[Z/OL].
<https://blog.csdn.net/tiandiren111/article/details/107344274>, 2020-07-14.
- [6] 张昭, 刘洋, 吕梁辉, 刘海朋, 张操, 王春罡, 吴春蕾. 基于红外对射和视频监控的变电站电子围栏设计[J]. *山东电力高等专科学校学报*, 2020, 23(6): 31-33+51.
- [7] 盛明远, 赵敬之, 刘锦, 朱怡然, 王昭杰, 谢海芬. 热源温度和距离对红外热释电传感器的影响研究[J]. *物理与工程*, 2021, 31(1): 114-119.
- [8] 360 百科.dht11 [Z/OL]. <https://baike.so.com/doc/6838529-7055792.html>, 2020-09-25.
- [9] 李其元. 基于 STM32 的张力式电子围栏的软硬件开发[D]: [硕士学位论文]. 绵阳: 西南科技大学, 2017.
- [10] 聂元秀. 脉冲电子围栏的运行维护及异常处理分析[J]. *通讯世界*, 2019, 26(12): 212-213.