

2022 年全国大学生电子设计竞赛试题

设计报告



基于物联网的养殖场环境监测辅导系统（创新专题）

Environmental Monitoring and Guidance System for

Livestock Farms Based on Internet of Things

【本科组】

摘要:

根据 2020 年 09 月 27 日国务院办公厅发表关于促进畜牧业高质量发展的意见, 其中指出: 提升畜牧业机械化水平。制定主要畜禽品种规模化养殖设施装备配套技术规范, 推进养殖工艺与设施装备的集成配套。目前来看, 传统养殖对畜禽生产也不能进行有效的监督, 当其流向市场时, 更不能保证其质量安全, 这对畜禽产品市场是一个很大的潜在危害性因素。

鉴于此以及个体养殖户对养殖场监控环境因素监控力匮乏, 难以形成最适合的养殖环境, 从而导致动物出现疾病、瘟疫等等情况, 本团队开发出一款软硬件结合的基于物联网的养殖场环境检测辅导系统。此设备能实时检测养殖场内的温湿度、氨气的影响动物生长的重要环境因素。通过蓝牙模块上传到应用程序端, 能方便养殖户实时监控养殖场的环境数据。

关键词: 人工智能、计算机视觉、软硬件结合、实时数据监控、远程监控

Abstract:

According to the opinions issued by the General Office of the State Council on Promoting the High-quality Development of Animal Husbandry on September 27, 2020, it is pointed out that the level of mechanization of animal husbandry should be improved. Formulate technical specifications for large-scale breeding facilities and equipment for major livestock and poultry breeds, and promote the integration of breeding technology and facilities and equipment. At present, traditional breeding can not effectively supervise the production of livestock and poultry, and when it flows to the market, its quality and safety cannot be guaranteed, which is a great potential harmful factor for the livestock and poultry products market.

In view of this and the lack of monitoring power of individual farmers to monitor environmental factors on farms, it is difficult to form the most suitable breeding environment, resulting in diseases, plagues and other situations in animals, the team developed a combination of software and hardware based on the Internet of Things farm environmental detection and counseling system. This equipment can detect in real time the temperature and humidity in the farm, ammonia and important environmental factors affecting animal growth. Uploaded to the application side through the Bluetooth module, it is convenient for farmers to monitor the environmental data of the farm in real time.

Keyword: Artificial intelligence、Computer vision、Combination of hardware and software、Real-time data monitoring、Remote monitoring

一、系统方案

本系统分为主从机两个部分。主机由单片机 STM32F103ZET6 模块、ILI9341 的 2.8 寸 ALIENTEK TFTLCD 显示模块、0.96 寸 OLED 模块、OV7725 摄像头模块、蓝牙通信模块、app 模块、电源模块组成；而从机则由单片机 STM32F103ZET6 模块、串口通信模块、DHT11 温湿度传感器模块、DS18B20 温度传感器模块、MQ137 氨气传感器模块、电源模块组成。下面分别论证这几个模块的选择。

1、单片机的论证与选择

方案一：采用传统的 51 系列单片机。

传统的 51 单片机为 8 位机，价格便宜，控制简单，但是运算速度慢，片内资源少，存储容量小，难以存储大体积的程序和实现快速精准的反应控制。并且受时钟限制，计时精度不高，外围电路也增加了系统的不可靠性。

方案二：采用以增强型 80C51 内核的 STC 系列单片机 STC12C5A60S2，其片内集成了 60kB 程序 Flash，2 通道 PWM、16 位定时器等资源，操作也、较为简单，具有在系统调试功能（ISD），开发环境非常容易搭建。但实际使用了多种传感器对内存要求较高的外设，无法很好地符合设计的需要。

方案三：采用以 ARM Cortex-M3 为内核的 STM32F1 系列控制芯片，STM32 系列芯片时钟频率高达 72MHz，具有 64K 字节 SRAM，512K 字节的 FLASH 容量，具有极强的处理计算能力。

综合以上三种方案，由于方案三比较有挑战性，且内存大，选择方案三。

2、显示屏的论证与选择

方案一：采用 OLED 显示，当有电流通过时，有机材料就会发光，操作方便，功能丰富，可显示汉字、ASCII、图案等，OLED 显示屏幕可视角度大，功耗低。但是其显示屏显示区域较小，更加适合小系统，可用于本系统的实时文字监测数据显示。

方案二：采用 ALIENTEK TFTLCD 显示，其具有 320×240 的分辨率，57.6mm*47.2mm 的显示区域，方便摄像头检测图像的显示。高亮背光，背光亮度高功耗低。通用的接口，可用于 ALIENTEK 系列开发板，易上手。可用于本系统的实时摄像监控显示。

通过权衡比较，方案一可用于本系统的实时文字监测数据显示。方案二显示屏的显示区域较大、方便后期开发、也容易实现。从系统指标要求来看，对显示区域和分辨率要求较高。采用方案二分辨率恰好达到 OV7725 的 QVGA 的大小，同时显示区域较大，达到系统指标要求。

综合以上两种方案，选择方案一、二并用。

3、蓝牙通信的论证与选择

方案一：采用蓝牙通信，低功耗，方便电池供电的设备工作;价格便宜，可应用于低成本设备，数据和声音传输同时管理，低延迟。传输距离适中，数据

传输速率为 24Mb/s，不同设备间协议不兼容性，需要本地数据记录，以确保数据不间断可用。

方案二：采用连线串口通信，直接使用杜邦线将主从机的 TXD 和 RXD 串口相连，共地，即可进行通信，操作简单，但受距离限制严重。

方案三：采用 4G 模块通信，其可以透明传输和定位，用户通过串行通信将需要发送的数据(及原始经纬度信息)发送给模块，模块严格按照原始数据将用户数据发送到客户预设的指定服务器，服务器下发数据通过模组串口传输到用户主控，进而可实现远程通信。但是其成本高，配置高于本产品需求。

综合以上分析，蓝牙通信符合本系统主从机距离较远通信的特点、设计简单、成本较低、可靠性好，所以选择方案一。

4、电源模块的论证与选择

方案一：采用 5V 电源适配器供电，可直接满足部分模块的供电电压要求，且可方便自行设计 5V 转化为 3.3V 供电的稳压器。但是这种供电方式限制了系统的放置位置，需要外接电源插座。

方案二：采用可充放电 3.7V 锂电池供电。可使系统减少放置位置的束缚，需要增加 3.7V 转 5V 的升压模块，继而使用 5V 转化为 3.3V 供电的稳压器，满足系统供电要求。

方案三：采用 5 号电池升降压供电。可使系统减少放置位置的束缚，同时提高了稳定供电时间，需要增加 3V 转 5V 的升压模块，继而使用 5V 转化为 3.3V 供电的稳压器，满足系统供电要求。

经过讨论，三种方案各有利弊，且没有必然的对立关系，故而决定选择以上三种方案供电，任用户选择。

5、传感器的论证与选择

方案一：采用 DHT11 温湿度传感器监测环境的温湿度指标。DHT11 数字温湿度传感器是一款含有已校准数字信号输出的温湿度复合传感器。它应用专用的数字模块采集技术和温湿度传感器传感技术，确保产品具有优质的可靠性与长期稳定性。但是其温度检测的精度不高，温度变化不明显，还需要进一步提升。

方案二：采用 DS18B20 温度传感器监测环境的温度指标。DS18B20 的数字温度传感器，可以用来对环境温度进行定量的检测。支持多点组网功能，多个 DS18B20 可以并联在唯一的三线上，实现多点测温。而且其精度相对较高，能够较好地实时体现环境温度的变化。

方案三：采用 MQ137 氨气传感器监测环境的氨气指标。当传感器所处环境中存在氨气时，传感器的电导率随空气中氨气的浓度的增加而增大。使用简单的电路即可将电导率的变化转换为与该气体浓度相对应的输出信号。其竞品较少，在较宽的浓度范围对氨气有良好的灵敏度，长寿命、低成本、驱动电路简单。

综合以上分析，湿度的检测选择方案一。为了能够较好地实时体现环境温度的变化，温度的检测选择方案二。氨气的检测选择方案三。

6、图像识别模型数据读取的论证与选择

方案一：采用同步数据读取，数据读取与模型训练串行。当模型需要数据时，才运行数据读取函数获得当前批次的数据。在读取数据期间，模型一直等待数据读取结束才进行训练，数据读取速度相对较慢。

方案二：采用异步数据读取，数据读取和模型训练并行。读取到的数据不断的放入缓存区，无需等待模型训练就可以启动下一轮数据读取。当模型训练完一个批次后，不用等待数据读取过程，直接从缓存区获得下一批次数据进行训练，从而加快了数据读取速度。

综合以上分析，同步数据读取方式较为简单，且本模型的数据集较少，数据读取与模型训练串行速度也不会太低，所以选择方案一。

7、图像识别模型网络结构的论证与选择

方案一：采用全连接神经网络，虽然使用经典的全连接神经网络可以提升一定的准确率，但其输入数据的形式导致丢失了图像像素间的空间信息，这影响了网络对图像内容的理解。

方案二：采用卷积神经网络，卷积神经网络针对视觉问题的特点进行了网络结构优化，可以直接处理原始形式的图像数据，保留像素间的空间信息，因此更适合处理视觉问题。卷积神经网络由多个卷积层和池化层组成。卷积层负责对输入进行扫描以生成更抽象的特征表示，池化层对这些特征表示进行过滤，保留最关键的特征信息。

综合以上分析，虽然全连接神经网络较为简单，但是对于计算机视觉问题，效果最好的模型仍然是卷积神经网络。所以选择方案二。

8、图像识别模型损失函数的论证与选择

方案一：采用均方误差，当标签的可能取值是一个连续的实数值，以模型输出的数值和真实标签值差距作为损失函数（Loss）是合理的。均方误差作为损失函数多出现于回归模型。

方案二：采用交叉熵，交叉熵损失函数的设计是基于最大似然思想：最大概率得到观察结果的假设是真的。交叉熵只计算对应着“正确解”标签的输出自然对数，其值是由正确标签所对应的输出结果决定的。常用交叉熵误差作为分类问题的损失衡量。

综合以上分析，本模型的样本标签是一个个离散确定值，更贴近分类问题，交叉熵作为损失函数的效果更好。所以选择方案二。

9、图像识别模型优化算法的论证与选择

方案一：采用 SGD 算法，随机梯度下降算法，每次训练少量数据，抽样偏差导致的参数收敛过程中震荡。

方案二：采用 Momentum 算法，引入物理“动量”的概念，累积速度，减少震荡，使参数更新的方向更稳定。每个批次的数据含有抽样误差，导致梯度更新的方向波动较大。如果我们引入物理动量的概念，给梯度下降的过程加入一定的“惯性”累积，就可以减少更新路径上的震荡，即每次更新的梯度由

“历史多次梯度的累积方向”和“当次梯度”加权相加得到。历史多次梯度的累积方向往往是从全局视角更正确的方向。

方案三：采用 AdaGrad 算法，根据不同参数距离最优解的远近，动态调整学习率。学习率逐渐下降，依据各参数变化大小调整学习率。但是其过程中可能会出现学习率急剧下降的问题。

方案四：采用 Adam 算法，由于动量和自适应学习率两个优化思路是正交的，因此可以将两个思路结合起来，这就是当前广泛应用的算法。

综合以上分析，无法直观地判断哪一个优化算法更适合本模型，故而都进行了试验，Adam 优化算法效果更好，能较快得到损失尽可能小的参数，并且后期波动较小。所以选择方案四。

10、远程监控客户端的论证与选择

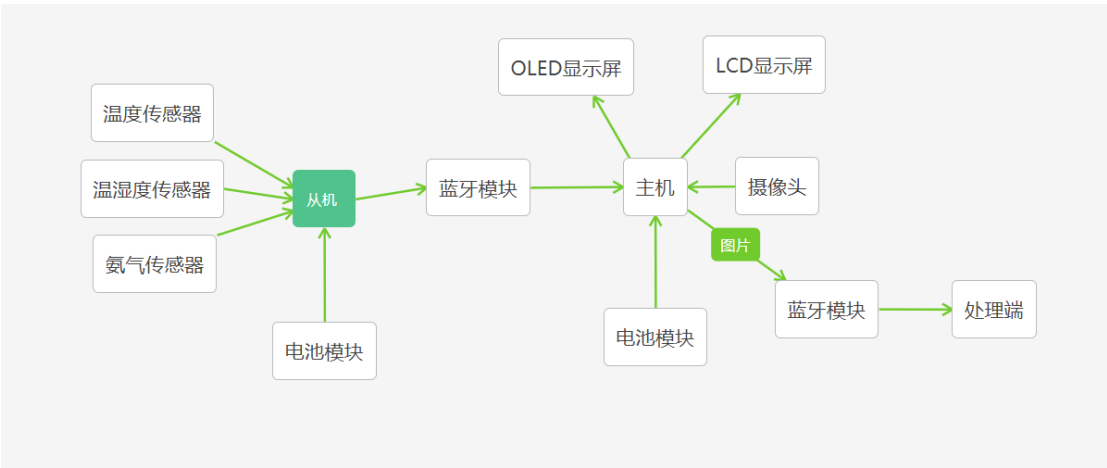
方案一：通过微信小程序进行数据的监控。微信小程序是很普及的，小程序相比客户端应用，它不需要下载安装安装，因而更像是一种微信深度定制版的 Web 应用，比较方便用户的使用。但是如果需要发行微信小程序到大众中的话需要拥有合法的域名，而申请合法域名的成本较高。

方案二：通过 APP inventor 开发一款我们适用的 app 进行数据的监控。相比微信小程序，app 发行要简单许多。使用简洁的代码块拼接开发，对初学者很友好。对我们的项目而言，此款通过 appinventor 所开发的 app 也十分适用。

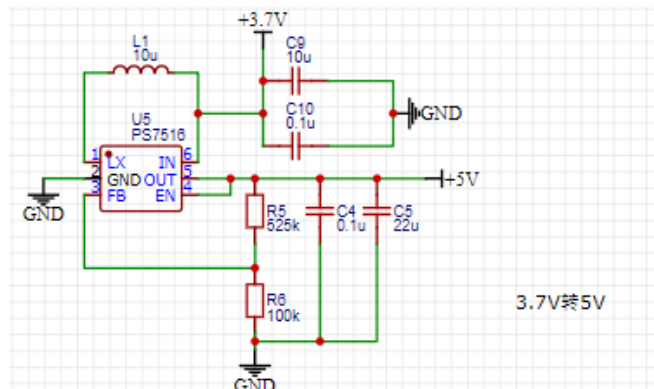
综合上面两个方案的优缺点，为了方便远程监控数据，本项目最终决定采取方案二进行数据的监控。

二、系统理论分析与计算

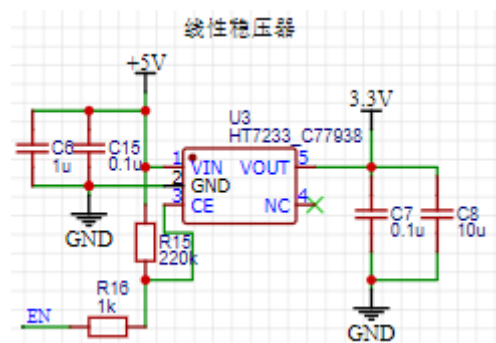
1、系统框图



2、电源模块理论分析

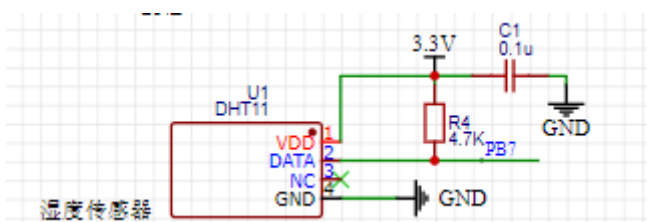
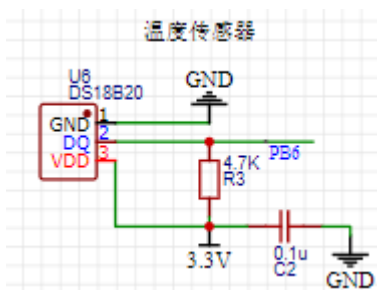


该模块基于 PS7516 锂电池同步整流升压 IC 而搭建,用于将锂电池的 3.7V 输出转换成 5V 输出。其 IC 效率可达到 96%。PS7516 输入电压范围: 2.5V-5.5V, 适宜用在锂电池升压输出 5V1A 的应用中。其计算公式为 $V_{out}=0.8V*(1+R5/R6)$ 。



该模块是基于 HT7233 芯片而设计的电路,用于作为 5V 转 3.3V 的线性稳压。没有采用常用的 LM2596 芯片,而使用该 LDO 是因为其具有高输出电压精度,静态功耗电流,高工作电压可达 8V,耐干扰性强。

3、温湿度传感器模块理论分析



DHT11 数字湿温度传感器和 DS18B20 温度传感器都采用单总线数据格式,且外围电路一致。即单个数据引脚端口完成输入输出双向传输。温湿度传感器数据包由 5Byte (40Bit) 组成。数据分小数部分和整数部分,一次完整的数据传输为 40bit,高位先出。DHT11 的数据格式为: 8bit 湿度整数数据+8bit 湿

度小数数据+8bit 温度整数数据+8bit 温度小数数据+8bit 校验和。其中校验和数据为前四个字节相加。

byte4	byte3	byte2	byte1	byte0
00101101	00000000	00011100	00000000	01001001
整数	小数	整数	小数	校验和
湿度		温度		校验和

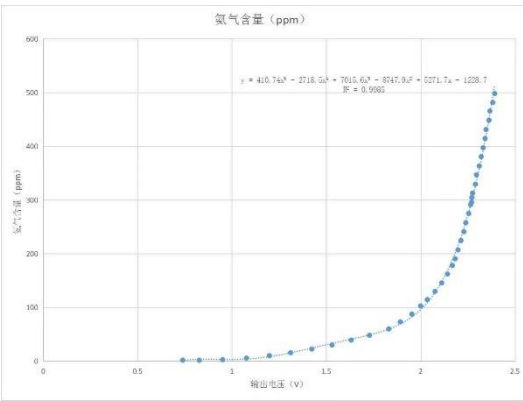
传感器数据输出的是未编码的二进制数据。数据(湿度、温度、整数、小数)之间应该分开处理。由以上数据就可得到湿度和温度的值，计算方法：

湿度= byte4 . byte3=45.0（%RH）

温度= byte2 . byte1=28.0（℃）

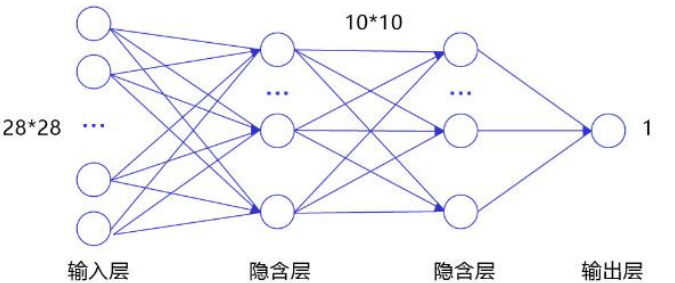
校验= byte4+ byte3+ byte2+ byte1=73(=湿度+温度) (校验正确)

4、MQ137 氨气传感器模块理论分析



ppm 是用溶质质量占全部溶液质量的百万分比来表示的浓度，也称百万分比浓度，图为输出电压对应的氨气含量值的函数图。

5、深度学习模型理论分析



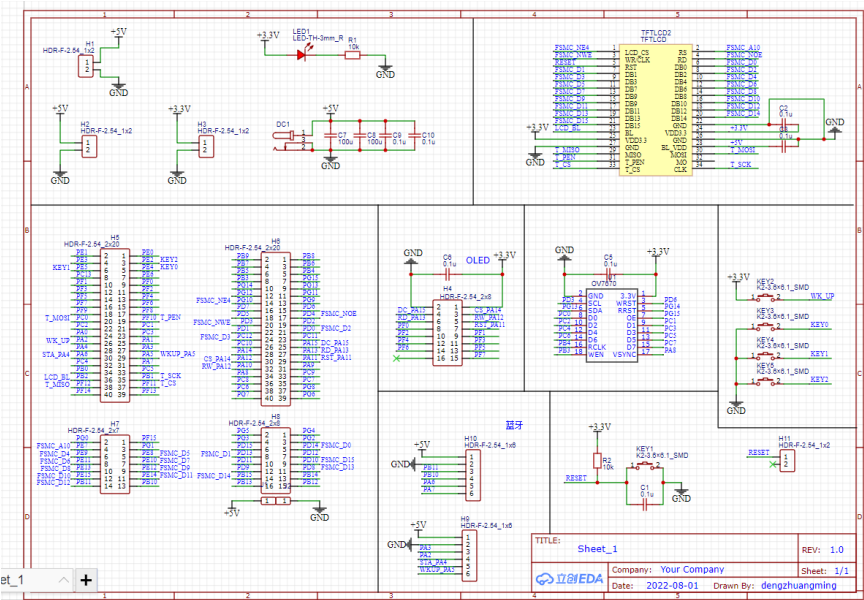
基于深度学习的模型可以简单分成 5 步处理，分别是数据处理、模型设计、训练配置、训练过程、模型保存。在该模型中，所使用的数据集是来源于智慧农业赛题——牛只图像分割竞赛、猪只盘点挑战赛等公开的数据集。模型的网络结构设计采用多层网络结构，引入卷积层和池化层，更好地去提炼图像

识别物的特征。而损失函数采用分类模型常用的交叉熵。为了使损失函数往梯度降低的方向前进，提高模型的预测能力，需要优化算法。在训练配置中，通过对比四种常用的优化器效果，可得 Adam 作为该模型的优化器更合适。并且在训练过程中，随着训练的进行，数据的准确性不断提高并趋于 100%，进一步验证了模型的可靠性。进而保存模型并将其封装为 API 以便调用。

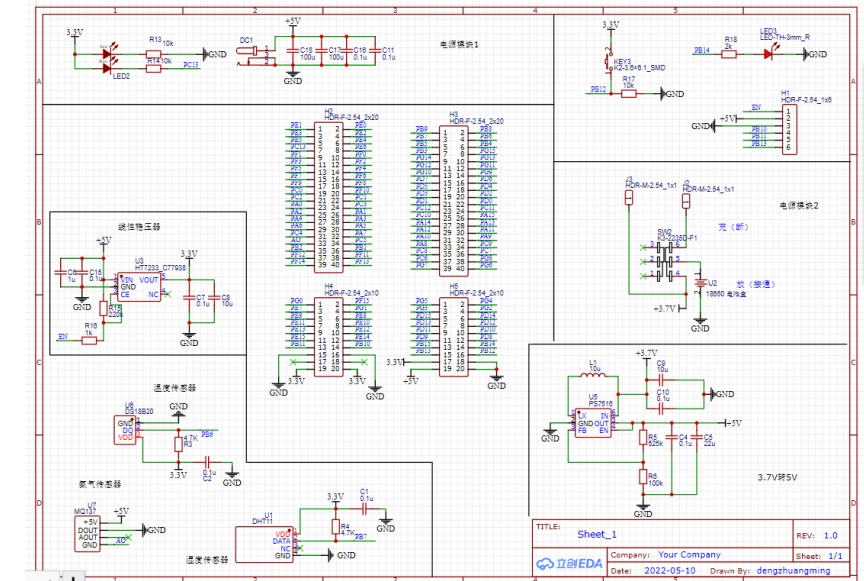
三、电路与程序设计

1、电路原理图

1.1 主机电路原理图

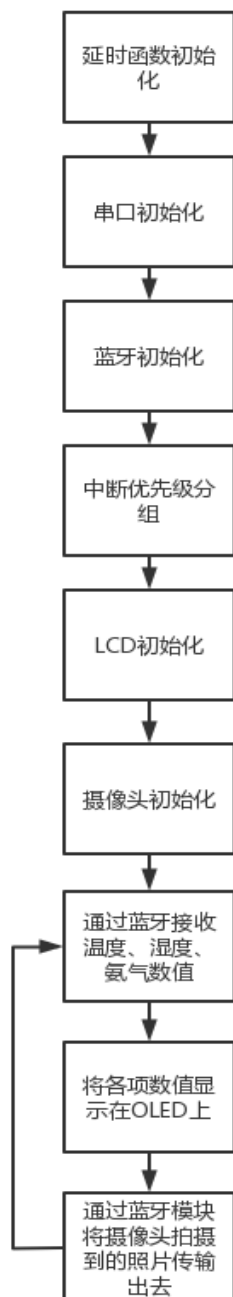


1.2 从机电路原理图

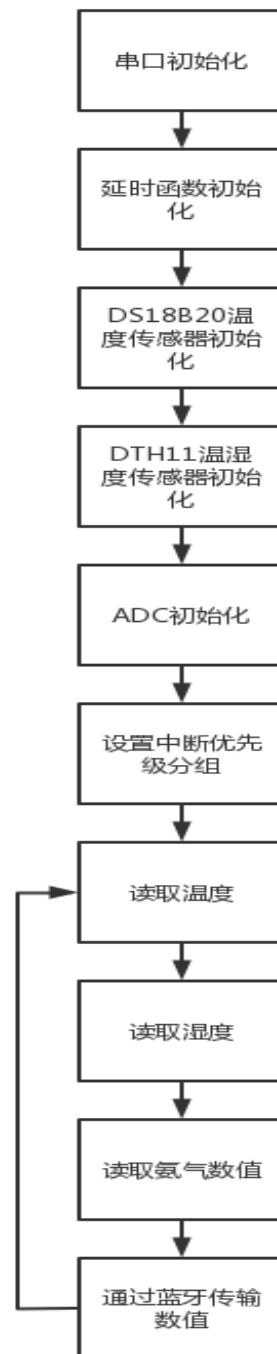


2、程序框图

2.1 主机程序框图:



2.2 从机程序框图:

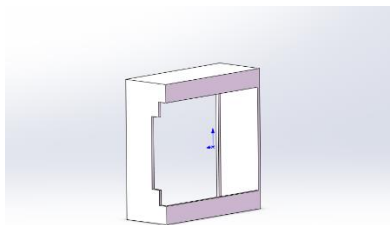


程序子流程：

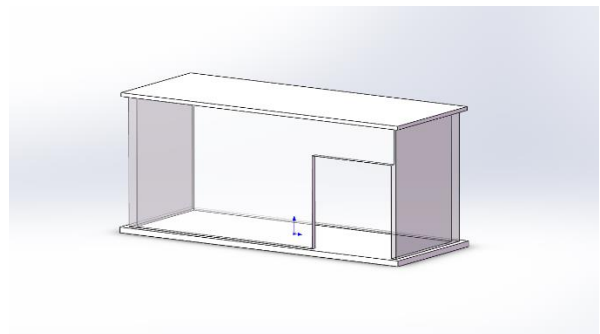
- 温度读取过程为：复位→发 SKIP ROM 命令→发开始转换命令→延时→复位→发送 SKIPROM 命令→发读存储器命令→连续读出两个字节（即温度）
- 湿度读取过程为：复位→等待响应→读取 5 个字节→校验→返回数值
- 氨气读取过程：通过 AD 转换得到数字量→通过相应公式将数字量转换为电压值显示→通过气体含量与模拟电压的计算公式，进行气体含量计算
- 串口初始化：串口时钟使能、GPIO 时钟使能→GPIO 端口模式设置→串口参数初始化
开启中断并且初始化 NVIC→使能串口

3、外观设计：

从机外壳



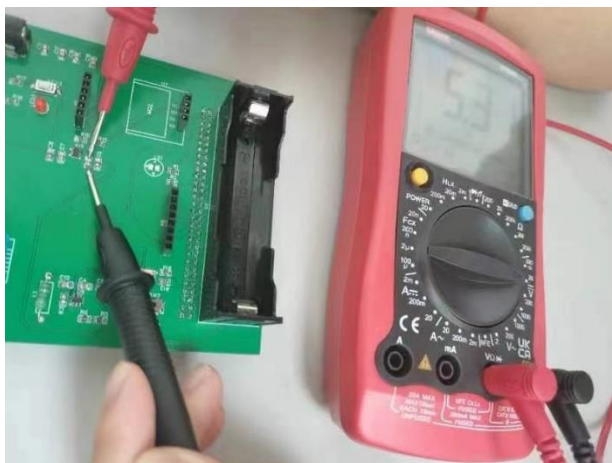
养殖场模型



四、测试方案与测试结果

1、电源测试方案及结果

测试方案：



两种供电模式分别上电测试，测试各模块输出的电压是否正常。电源适配器供电，各模块的电压值都接近理论值。电池供电测试时，发现电感功率太低，故换之，再测试电压值，一切供电正常。

测试设备：万用表、可调直流稳压电源

测试结果：电源指示正常，测试电源符合预想值。

2、传感器测试方案及结果

测试方案：

当从机、主机正常运行时观察主机的 OLED 显示显示屏上 DS18B20 温度传感器、DTH11 温湿度传感器、氨气传感器所测到的数值，并且通过温度计、湿度计等来验证 DS18B20 温度传感器、DTH11 温湿度传感器、氨气传感器所测到的数值的准确性，并且通过用手捂住温度传感器、湿度传感器来验证其是否能实时监测温度、湿度并且可以验证温度传感器、湿度传感器的灵敏性，可以通过将装有氨气浓度比较高的瓶子靠近氨气传感器来验证其灵敏性及实时监测的功能。

测试设备：

从机、蓝牙模块、DS18B20 温度传感器、DTH11 温湿度传感器、氨气传感器、OLED 显示屏、LCD 显示屏、摄像头、主机。

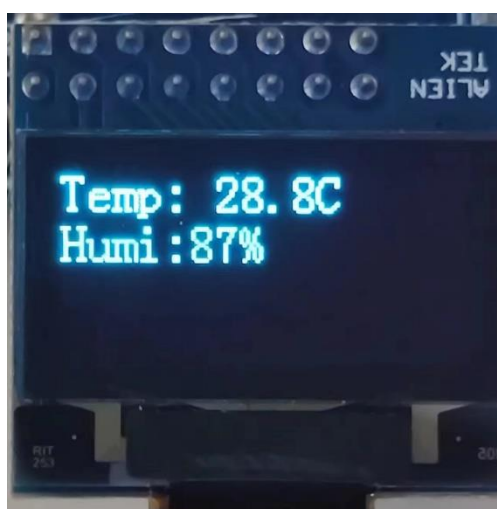
测量数据：温度、湿度、氨气。

测试结果：

温度传感器所测量的数值与当前室内的温度相比比较准确，误差在 0.1 摄氏度左右，且当用手捂住温度传感器时，其测量的数值也快速上升，可见其灵敏性较高并且测量的数值依然准确；湿度传感器所测量的数值与当前的室内湿度相比较为准确，误差在 1% 左右，且当用手捂住温湿度传感器时、其所测得数值也快速变化、可见其灵敏性较高并且测量的数值依然准确；氨气传感器的测量数值也比较准确，并且当将装有氨气浓度比较高的瓶子靠近氨气传感器时，氨气传感器所测量的数值也快速变化，可见其灵敏性较高并且测量的数值依然准确。

实现功能：

温度的实时测量、湿度的实时测量、氨气的实时测量、并且通过蓝牙模块显示在主机的 OLED 显示屏上。



3、功能与指标:

主机: 通过蓝牙模块接收到实时的温度、湿度、氨气的数值, 并将温度、湿度、氨气的数值实时显示到主机的 OLED 的显示屏上, 主机搭载的摄像头模块把监控画面显示在 LCD 上, 并通过蓝牙模块传到客户端, 实现监控功能。

从机: 通过对 DS18B20 温度传感器的控制实现对当前室内实时温度的监控, 并且误差在 0.1 摄氏度左右, 通过对 DHT11 的温湿度传感器的控制实现对当前室内湿度的实时监控, 误差在百分之一左右, 通过对氨气传感器的控制实现对当前室内氨气的数值进行实时监控, 单位为 ppm, 通过蓝牙模块将温度、湿度、氨气的数值实时传输到主机, 可传输的距离相对较远。

App: 通过代码能实现自由选择连接蓝牙, 并通过 visiostudio 设计一款上位机进行对从单片机传输过来的图片数据进行解码并显示。交由深度学习模型进行分析, 同时也方便用户监控养殖场内的情况, 能够更加准确地把控。

附录

(一) 源代码

1、DHT11 温湿度传感器初始化源代码:

```
#include "dht11.h"
#include "delay.h"
//复位 DHT11
void DHT11_Rst(void)
{
    DHT11_IO_OUT(); //SET OUTPUT
    DHT11_DQ_OUT=0; //拉低 DQ
    delay_ms(20); //拉低至少 18ms
    DHT11_DQ_OUT=1; //DQ=1
    delay_us(30);
    //主机拉高 20~40us
}
//等待 DHT11 的回应
//返回 1:未检测到 DHT11 的存在
//返回 0:存在
u8 DHT11_Check(void)
{
    u8 retry=0;
    DHT11_IO_IN(); //SET INPUT
    while
    (DHT11_DQ_IN&&retry<100)//DHT11
```

```
会拉低 40~80us
    {
        retry++;
        delay_us(1);
    };
    if(retry>=100)return 1;
    else retry=0;
    while
    (!DHT11_DQ_IN&&retry<100)//DHT11
    拉低后会再次拉高 40~80us
    {
        retry++;
        delay_us(1);
    };
    if(retry>=100)return 1;
    return 0;
}
//从 DHT11 读取一个位
//返回值: 1/0
u8 DHT11_Read_Bit(void)
{

```

```

    u8 retry=0;
    while(DHT11_DQ_IN&&retry<100)
//等待变为低电平
    {
        retry++;
        delay_us(1);
    }
    retry=0;
    while(!DHT11_DQ_IN&&retry<100)
//等待变高电平
    {
        retry++;
        delay_us(1);
    }
    delay_us(40); //等待 40us
    if(DHT11_DQ_IN) return 1;
    else return 0;
}
//从 DHT11 读取一个字节
//返回值：读到的数据
u8 DHT11_Read_Byte(void)
{
    u8 i, dat;
    dat=0;
    for (i=0; i<8; i++)
    {
        dat<<=1;
        dat|=DHT11_Read_Bit();
    }
    return dat;
}
//从 DHT11 读取一次数据
//temp:温度值(范围:0~50° )
//humi:湿度值(范围:20%~90%)
//返回值： 0, 正常;1, 读取失败
u8 DHT11_Read_Data(u8 *temp, u8 *humi)
{
    u8 buf[5];
    u8 i;
    DHT11_Rst();
    if(DHT11_Check()==0)

```

```

    {
        for(i=0; i<5; i++)
            //读取 40 位数据
            {
                buf[i]=DHT11_Read_Byte();
            }

        if((buf[0]+buf[1]+buf[2]+buf[3])==buf[4])
        {
            *humi=buf[0];
            *temp=buf[2];
        }
        }else return 1;
        return 0;
    }
//初始化 DHT11 的 IO 口 DQ 同时检测
DHT11 的存在
//返回 1:不存在
//返回 0:存在
u8 DHT11_Init(void)
{
    GPIO_InitTypeDef
    GPIO_InitStructure;
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
//使能 PB 端口时钟
    GPIO_InitStructure.GPIO_Pin =
    GPIO_Pin_7;
//PB7 端口配置
    GPIO_InitStructure.GPIO_Mode
    = GPIO_Mode_Out_PP;
//推挽输出
    GPIO_InitStructure.GPIO_Speed
    = GPIO_Speed_50MHz;
    GPIO_Init(GPIOB,
    &GPIO_InitStructure);
//初始化 IO 口
    GPIO_SetBits(GPIOB, GPIO_Pin_7
    ); //PB7 输出高
    DHT11_Rst();

```

<pre>//复位 DHT11 return DHT11_Check();</pre>	<pre>//等待 DHT11 的回应 }</pre>
---	-----------------------------

2、家禽种类及数目识别模型源代码

数据处理

```
def load_data(mode='train'):
    datafile = './work/jiaqin.json.gz'
    print('loading jiaqin dataset from {} .....'.format(datafile))

    data = json.load(gzip.open(datafile))
    # 读取数据集中的训练集, 验证集和测试集
    train_set, val_set, eval_set = data
    # 数据集相关参数, 图片高度 IMG_ROWS, 图片宽度 IMG_COLS
    IMG_ROWS = 28
    IMG_COLS = 28
    # 根据输入 mode 参数决定使用训练集, 验证集还是测试集
    if mode == 'train':
        imgs = train_set[0]
        labels = train_set[1]
    elif mode == 'valid':
        imgs = val_set[0]
        labels = val_set[1]
    elif mode == 'eval':
        imgs = eval_set[0]
        labels = eval_set[1]
    # 获得所有图像的数量
    imgs_length = len(imgs)
    # 验证图像数量和标签数量是否一致
    assert len(imgs) == len(labels), "length of train_imgs({})
) should be the same as train_labels({})".format(len(imgs), len(labels))
    index_list = list(range(imgs_length))
    # 读入数据时用到的 batchsize
    BATCHSIZE = 100
    # 定义数据生成器
    def data_generator():
        # 训练模式下, 打乱训练数据
        if mode == 'train': random.shuffle(index_list)
        imgs_list = []
        labels_list = []
        # 按照索引读取数据
        for i in index_list:
```

```

        # 读取图像和标签，转换其尺寸和类型
        img = np.reshape(imgs[i], [1, IMG_ROWS, IMG_COLS
]).astype('float32')
        label = np.reshape(labels[i], [1]).astype('int64')
        imgs_list.append(img)
        labels_list.append(label)
        if len(imgs_list) == BATCHSIZE:
            yield np.array(imgs_list), np.array(labels_list)
        # 清空数据缓存列表
        imgs_list = []
        labels_list = []
        # 如果剩余数据的数目小于 BATCHSIZE，则剩余数据一起构成
        # 一个大小为 len(imgs_list) 的 mini-batch
        if len(imgs_list) > 0:
            yield np.array(imgs_list), np.array(labels_list)
    return data_generator

```

定义模型结构

```

import paddle.nn.functional as F
# 多层卷积神经网络实现
class JIAQIN(paddle.nn.Layer):
    def __init__(self):
        super(JIAQIN, self).__init__()
        self.conv1 = Conv2D(in_channels=1, out_channels=20,
kernel_size=5, stride=1, padding=2)
        self.max_pool1 = MaxPool2D(kernel_size=2, stride=2)
        self.conv2 = Conv2D(in_channels=20, out_channels=20,
kernel_size=5, stride=1, padding=2)
        self.max_pool2 = MaxPool2D(kernel_size=2, stride=2)
        self.fc = Linear(in_features=980, out_features=3)
    def forward(self, inputs):
        x = self.conv1(inputs)
        x = F.relu(x)
        x = self.max_pool1(x)
        x = self.conv2(x)
        x = F.relu(x)
        x = self.max_pool2(x)
        x = paddle.reshape(x, [x.shape[0], -1])
        x = self.fc(x)
        return x

```

训练模型

```

def train(model):
    model.train()

```



```

        opt = paddle.optimizer.Adam(learning_rate=0.01, weight_decay
=paddle.regularizer.L2Decay(coeff=1e-
5), parameters=model.parameters())
    EPOCH_NUM = 5
    for epoch_id in range(EPOCH_NUM):
        for batch_id, data in enumerate(train_loader()):
            images, labels = data
            images = paddle.to_tensor(images)
            labels = paddle.to_tensor(labels)
            predicts, acc = model(images, labels)
            #计算损失，取一个批次样本损失的平均值
            loss = F.cross_entropy(predicts, labels)
            avg_loss = paddle.mean(loss)
            #后向传播，更新参数的过程
            avg_loss.backward()
            opt.step()
            opt.clear_grad()

```

保存模型参数

```

paddle.save(model.state_dict(), 'jiaqin_regul.pdparams')
model = JIAQIN()
train(model)

```

4、app 程序源代码

```

{
    start = true;
}
//开始读取数据
if (start)
{
    //绘制图像(一行一行地绘制)
    for (int Ycount = 0; Ycount < Iheight; Ycount++)
    {
        //读一行
        while (sp.BytesToRead < data.Length) ;
        sp.Read(data, 0, data.Length);
        if (data[data.Length - 1] != 10)
        {
            start = false;
            return;
        }
        colors = RGB565ToColor(data);
        for (int Xcount = 0; Xcount < Iwidth; Xcount++)
        {
            Image.SetPixel(Xcount, Ycount, colors[Xcount]);
        }
        pictureBox1.Image = (局部变量)intXcount
        textBox1.Text = (Ycount + 1).ToString();
    }
    Frame++;
    Frame = Frame > 0x7fffffff ? 0 : Frame;
    textBox3.Text = "帧数: " + Frame.ToString();
    start = false; //接收数据结束
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
    return;
}
}

```

```

private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
}

private void button2_Click(object sender, EventArgs e)
{
    try
    {
        sp.Close(); //关闭串口
        button1.Enabled = true; //打开串口按钮可用
        button2.Enabled = false; //关闭串口按钮不可用
    }
    catch (Exception err) //一般情况下关闭串口不会出错，所以不需要加处理程序
    {
    }
}

private void button3_Click(object sender, EventArgs e)
{
}

private void radioButton4_CheckedChanged(object sender, EventArgs e)
{
}

private void sp_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    //可能会出现意想不到错误的代码区域
    try
    {
        //检测是否开始发送数据
        if (sp.ReadLine().Equals("image:"))
        {

        }

        {
            InitializeComponent();
            System.Windows.Forms.Control.CheckForIllegalCrossThreadCalls = false;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                sp.PortName = comboBox1.Text;
                sp.BaudRate = Convert.ToInt32(comboBox2.Text, 10); //十进制数据转换
                sp.Open();
                button1.Enabled = false; //打开串口按钮不可用
                button2.Enabled = true; //关闭串口
            }
            catch {
                MessageBox.Show("端口错误, 请检查串口", "错误");
            }
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            for (int i = 1; i < 20; i++)
            {
                comboBox1.Items.Add("COM" + i.ToString());
            }
            comboBox1.Text = "COM1"; //串口号多默认值
            comboBox2.Text = "4800"; //波特率默认值

            /*****非常重要*****/

            sp.DataReceived += new SerialDataReceivedEventHandler(port_DataReceived); //必须手动添加事件处理程序
        }

        private void port_DataReceived(object sender, SerialDataReceivedEventArgs e) //串口数据接收事件
        {

```

（二）作品操作说明

作品硬件部分分为主机和从机。从机上电即可工作，主机上电正常工作后，自动

接收从机传输来的数据，并在 OLED 显示。主机所搭载的摄像头画面也会在 LCD 上显示。可按主机上的按键调节画面的效果，若出现异常，可尝试按复位键。App 端接收从单片机传输过来的图文数据进行解码等处理并显示。

（三）应用资料与参考文献目录

[1]王亮亮. 智能精准畜禽养殖管理系统设计——以猪的养殖为例[J]. 山东商业职业技术学院学报, 2021, 21(02):107-110. DOI:10.13396/j.cnki.jsict.2021.02.025.

[2]王小婷. 北方封闭奶牛舍环境监测系统及氨气浓度预测模型研究[D]. 黑龙江八一农垦大学, 2020. DOI:10.27122/d.cnki.ghlnu.2020.000046.

[3]谭婧. 养殖场环境卫生管理建议[J]. 今日畜牧兽医, 2020, 36(04):59.

（四）查新说明

1. 查新项目的创新要点

产品技术在布局时，采用主从机无线通信的技术方案，达到从机分配到栏监测环境，实现同时监测各栏实时状况，及时发现突发紧急情况，并去除了繁杂的有线连接，具有系统轻便覆盖范围广的优点。从机集结了温湿度、氨气传感器，能实时监测养殖场的环境参数。使用蓝牙模块传输养殖场内监控图片及场内监测数据到用户端，通过应用自主训练基于深度学习的养殖场家禽种类及数目识别模型，实现智能化监控养殖场情况。

2. 查新点与查新要求

查新点：

1. 使用蓝牙模块传输养殖场内监控图片及场内实时监测数据
2. 基于深度学习的养殖场家禽种类及数目识别模型应用于养殖场监测系统

查新要求：

1. 在所查范围内确定有无相同或类似研究；
2. 对查新项目分别或者综合进行对比分析；
3. 对查新项目的新颖性进行判断。

3. 文献检索范围及检索策略

查新使用的数据库：

中国学术期刊网（1994-2022）

CNKI 中国博士学位论文全文数据库（1999-2022）

维普中文科技期刊数据库（1989-2022）

万方学位论文数据库（1977-2022）

万方中国学术会议论文库（1997-2022）

万方中国专利文摘数据库（1985-2022）

万方中国科技成果库 (1985-2022)
Web of Science(1990-2022)
Elsevier(1995-2022)

检索策略:

(一) 检索词:

中文	英文
养殖场	Farms
环境监测	The environmental monitoring
蓝牙传图	Bluetooth transmission figure
计算机视觉	Computer vision

(二) 检索式

1. (养殖场 or 蓝牙传图 or 计算机视觉)and 环境监测
2. (Farms or bluetooth transmission figure or computer vision) and the environmental monitoring

4. 检索结果

经对国内外数据库进行查询,在以上数据库和文献时限内,检索到一些与项目课题有关的文献,现选出 2 篇文献,具体情况如下:

1. 题名: 基于物联网的养殖场环境远程监控系统的设计与实现

作者: 刘生成

文摘: 传统的养殖场环境监控需要人工现场测量和调节,这不仅费时费力,而且监控效率低下。本课题以此为出发点,设计开发了一套可以在智能移动终端上随时随地对养殖场环境进行远程监控系统,并通过搭建软硬件平台和功能实现来论证所设计方案的可行性。论文首先分析了养殖场环境因素对禽畜生长的影响,确定以温度、湿度、光照强度、甲烷浓度、氨气浓度和硫化氢浓度为监测对象,在此基础上结合物联网技术完成了包括软硬件在内的养殖场环境远程监控系统总体方案设计。

硕士电子期刊, 2019 (01)

2. 题名: Barriers to computer vision applications in pig production facilities

作者: JiangongLi、agAngela R.Green-Miller、Xiaodan Hu、Ana Lucic、Mahesh MohanM.R.、RyanN.Dilger、Isabella C.F.S.Condotta、Brian Aldridge、John M.Hart、NarendraAhuja

文摘: In recent years, many studies have explored the use of various technologies and sensors to assist animal caretakers in monitoring animal activity and behavior. Of these technologies, computer vision offers the most consistent promise as an effective aid in animal care, and yet, a systematic review of the state of application of this technology indicates that there are many significant barriers

to its widespread adoption and successful utilization in commercial production system settings.

Elsevier Journal, 2022

5. 查新结论

经对检索出的相关文献进行分析、对比，结论如下：

国内没有发现基于深度学习的养殖场家禽种类及数目识别模型应用于养殖场监测系统的文献。在目前的养殖场环境监测分析文献中，多仅针对传感器对环境参数的采集研究，说明本项目研究内容具有创新性。

且在国外期刊的查阅中，表明计算机视觉在养猪生产设施中的仍存在应用障碍，最重要的障碍之一是识别客观行为标记的错误来源，这些错误无法通过当前的算法性能评估来衡量。仍需不断克服技术难题。

综上所述，本项目具有科学性、可行性以及新颖性。

6. 查新声明

- (1) 报告中陈述的事实是真实和准确的。
- (2) 我们按照大赛查新规范进行查新、文献分析和审核，并做出上述查新结论。