

Monitoramento e Controle de Temperatura e Umidade em Ambientes de Produção de Frangos de Corte

Douglas Lopes de Souza¹, Paulo João Martins¹

¹Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC)
Av. Universitária, 1105 - CEP: 88806-000 – Criciúma – SC – Brasil

douglaslopsou@gmail.com, pjm@unesc.net

Abstract. *Brazilian poultry farming is very competitive, which requires that producers invest in new technologies and have strict control over production. Ensuring the health of the birds is essential to obtain better results, thermal comfort is the main factor in terms of productivity gains and mortality reduction. This research aims to apply wireless data communication so that it is possible to monitor and control the temperature and humidity with parameters of a broiler production environment. The application captures the temperature and humidity through a wireless sensor and sends it to a central module that makes the data available to an actuator module that activates the temperature and humidity controls.*

Resumo. *A avicultura brasileira é muito competitiva o que exige que os produtores invistam em novas tecnologias e tenham um controle rigoroso da produção. Garantir a saúde das aves é fundamental para obter melhores resultados, o conforto térmico é o principal fator de ganho de produtividade e redução de mortalidade. Esta pesquisa tem como objetivo aplicar a comunicação de dados sem fio para que seja possível monitorar e controlar a temperatura e umidade com parâmetros de um ambiente de produção de frangos de corte. A aplicação capta a temperatura e umidade por meio de um sensor sem fio envia a um módulo central que disponibiliza os dados para um módulo atuador que ativa os controles de temperatura e umidade.*

1. Introdução

O Brasil se destaca mundialmente na avicultura, sendo o terceiro maior produtor e maior exportador mundial de carne de frango, a região sul se destaca neste cenário, sendo responsável pela maior porção da produção nacional [ABPA 2021]. Desde o início da produção de frangos de corte no Brasil o meio vem se modernizando, com o objetivo de sempre melhorar o desempenho do setor, reduzir custos e otimizar a produção. Nacionalmente tem se destacado como um dos setores mais organizados evidenciado pelos resultados não só em produtividade mas também com volume de abate e desempenho econômico contribuindo muito para a economia nacional. Mesmo com o expressivo crescimento nas exportações e no progresso científico verificado na avicultura, são necessários constantemente estudos sobre formas de alojamento que busquem a melhor ambiência para criação das aves [Andreazzi et al. 2018].

Garantir a saúde do frango é fundamental para obter um ótimo potencial genético e aproveitamento nutricional. Para que isso seja possível, medidas devem ser tomadas para monitorar e controlar o aviário, identificando os riscos à saúde das aves e do homem. Desde a chegada das aves é necessário adotar boas práticas de produção até o abate do lote, isso é crucial para o desempenho. Os frangos necessitam de condições ideais para seu desenvolvimento como temperatura, umidade relativa, pressão

atmosférica, luminosidade e nível sonoro. Melhorias nas instalações e nos equipamentos na avicultura brasileira contribuíram para o crescimento da produção de frangos de corte, pois, os produtores passaram a aumentar a quantidade de aves alojadas por metros quadrados. A ambiência em um aviário é de suma importância. Com um ambiente onde se tem um conforto térmico é possível criar uma maior quantidade de frangos por metro quadrado de galpão e ainda ter um melhor desempenho [Balduino Jorge et al. 2020].

O conforto térmico que os avicultores vem buscando está sendo obtido através de tecnologias e equipamentos que são responsáveis pela climatização, bem como máquinas automatizadas, dentre as tecnologias empregadas há exaustores ou ventiladores (ventilação forçada), nebulizadores (controlar a umidade) e *inlets* (painéis evaporativos) que refrigera os frangos na fase final da criação [Balduino Jorge et al. 2020].

As aves são animais homeotérmicos, ou seja, ficam em troca térmica contínua com o ambiente e para encontrar conforto ambiental a temperatura deve estar dentro de certos limites para evitar estresse térmico por calor. Com cobertura corporal dotada de penas isso dificulta a troca de calor com o meio. Isso faz com que a ave retenha mais calor do que dissipa. Com esses aspectos o estresse por calor é um dos maiores problemas na produção de frangos de corte, principalmente em regiões onde as altas temperaturas predominam. Fisiologicamente as aves com estresse térmico por calor reduzem a ingestão de alimentos assim como diminuem o metabolismo o que consequentemente resulta em uma redução no ganho de peso e desempenho alimentar [Melo et al. 2016].

De acordo com Camusso et al. (2021) na avicultura um ambiente é tido como confortável para aves adultas quando apresenta temperaturas de 15-18°C a 22-25°C e umidade relativa do ar na faixa de 50 a 70%, esses valores dificilmente são obtidos nas condições proporcionadas pelo clima brasileiro especialmente no verão. À medida que a temperatura sobe acima de 26°C a capacidade que as aves têm de dissipar o calor torna-se menos efetiva o que as deixa mais vagarosas e podem até parar de comer impactando diretamente no ganho de peso.

Para manter a temperatura e umidade nesses números ideais são empregados dois tipos de sistemas, os manuais e os automatizados. Sistemas manuais não apresentam custos, muito diferentes dos sistemas automatizados que possuem custos muito elevados. Os sistemas manuais podem ser competitivos perante aos automáticos que necessitam de um investimento maior em instalações e equipamentos, entretanto a tendência é que com produtores mais capitalizados e mais preparados para continuar no mercado meios para automatizar a produção sejam implantados nestes aviários [EMBRAPA 2021].

Diante da constante busca por tecnologias diferenciadas mais qualificadas, com o objetivo de promover o bem-estar dos animais de corte, otimizar a cadeia produtiva, reduzir custos, essas tecnologias tornam-se cada vez mais essenciais. O controle automatizado é mais eficiente, e muito mais preciso que o método manual, no qual os avicultores precisam se deslocar várias vezes ao dia até o aviário e utilizam apenas termômetros para medir a temperatura e umidade, ferramentas bem menos precisas que os sensores atuais, outro fator seria que no modelo tradicional o monitoramento não é

feito de forma constante, a temperatura e umidade podem se alterar rapidamente, gerando riscos para as aves [Avicultura Industrial 2021].

Segundo CNPSA (2021) existem formas de manter a temperatura e umidade nestes ambientes, pode-se utilizar o método por pressão positiva para ventilação a qual é posicionado os ventiladores no sentido longitudinal do aviário. Nesse processo as cortinas laterais do aviário permanecem fechadas e bem vedadas para tornar a ventilação tipo túnel eficiente. O ar entra por uma das extremidades do aviário e é dissipado pelos ventiladores, que são posicionados ao longo do comprimento, e pressionados a sair pela extremidade oposta que permanece aberta conforme a Figura 1. Nesse sistema o controle da ventilação é mais facilitado pois não sofre tanta influência da ventilação natural como nos outros sistemas e caso mesmo com a ventilação não seja possível atingir a temperatura desejada é empregado um sistema de nebulização (controle da umidade).

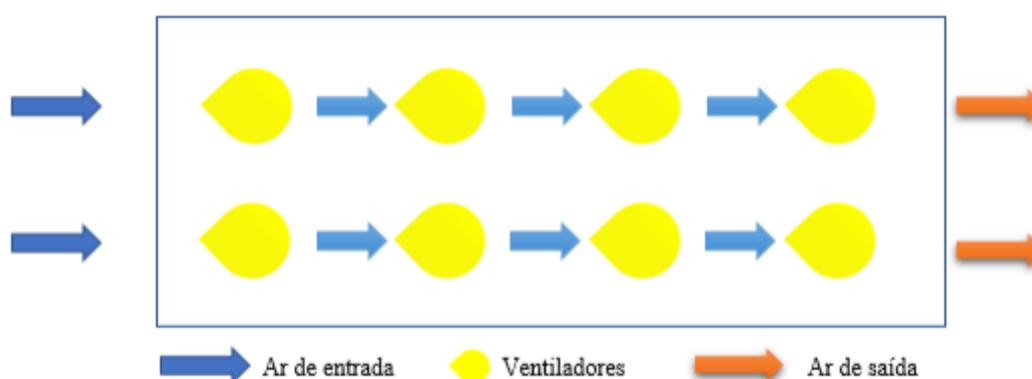


Figura 1. Ventilação por pressão positiva tipo túnel

Para a conectividade de um dispositivo sem fio geralmente está empregando WiFi, Bluetooth, Zigbee, LoRaWAN e alguns outros protocolos. Entre os módulos habilitados para WiFi, ESP8266 é um dos mais populares na comunidade de código aberto. O ESP8266 recebe tanta atenção em razão do seu custo benefício, incluindo transceptor de 2,4 GHz com amplificador de potência, CPU, ADCs, temporizadores, GPIOs, dentre outros. O ESP8266 tem sido empregado em várias aplicações, seja em fase experimental ou de produção [Yoppy et al. 2018].

Com base nos cenários citados, este protótipo propõe monitorar através de sensores e controlar a temperatura e umidade através de atuadores utilizando comunicação de dados sem fio com o ESP8266 e sensores tendo como base um sistema de ventilação por pressão positiva (ventilação artificial) e os parâmetros de temperatura e umidade, estes definidos de acordo com as fases do frango de um ambiente de produção de frangos de corte com a finalidade de manter o ambiente o mais próximo possível da zona de conforto térmico utilizando mecanismos de monitoramento e controle automatizados.

2. Materiais e Métodos

A presente pesquisa caracteriza-se por aplicada e de base tecnológica. Desenvolveu-se o protótipo de um sistema para monitorar e controlar a temperatura e umidade com parâmetros de ambientes de produção de frangos de corte. É constituído por 4 módulos sendo 2 de Software e 2 de Hardware. Os módulos de Software foram denominados como BAM-Backend e BAM-Dashboard, já os de Hardware foram definidos como BAM-Device e BAM-Controller (Figura 2).

Os módulos integrados empregaram a aplicação da comunicação de dados. O protótipo integra todos os módulos de forma que consiga monitorar a temperatura e umidade captadas pelo módulo BAM-Device. O módulo BAM-Backend é o módulo central que se comunica com todos os outros módulos, tanto para armazenar quanto para fornecer os dados parametrizados. O BAM-Dashboard é uma aplicação Frontend (Aplicação Web) que apresenta as informações servidas pelo BAM-Backend, é por meio dele que todas as parametrizações responsáveis pelo funcionamento do protótipo são realizadas previamente. O módulo BAM-Controller é o responsável por acionar os dispositivos que irão controlar a temperatura e a umidade, baseado nos parâmetros recebidos através da comunicação com o BAM-Backend e da ambiência atual. A conexão entre os módulos de Hardware é feita mediante conexão sem fio Wifi com protocolo HTTP (Hyper Text Transport Protocol) e formato JavaScript Object Notation (JSON)(Figura 2).

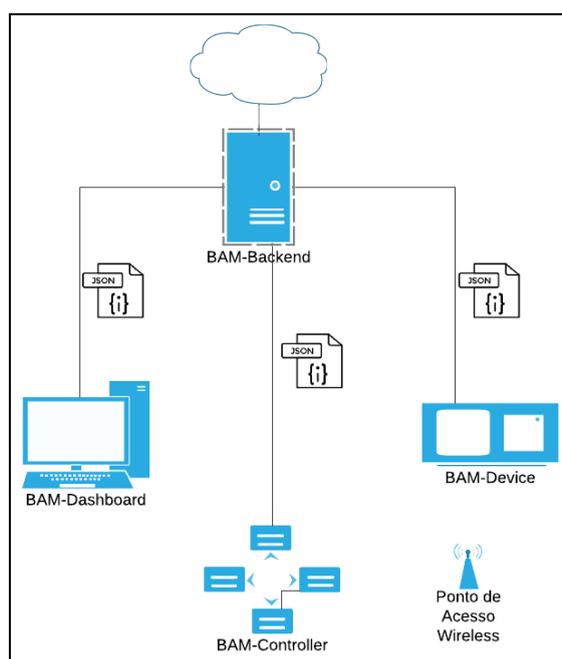


Figura 2. Comunicação entre os módulos

2.1 BAM-Backend

O módulo BAM-Backend foi desenvolvido em linguagem JavaScript (JS) no ambiente de programação Visual Studio Code, versão 1.56.0. Para executar o código JS em backend foi usado o Node.js, versão 12.16.3. Foram usadas as bibliotecas para o

desenvolvimento do módulo, são elas: Express, Typeorm. As bibliotecas foram disponibilizadas de forma gratuita pelo gerenciador de pacotes yarn, versão 1.22.4.

Quando executada a aplicação por meio do Node.js é iniciado o serviço na porta 3333. Após executada a aplicação possibilita que seja feita a comunicação através de requisição e resposta. O formato dos dados de requisição e resposta é JSON.

Como demonstrado na Figura 2 a solução proposta usa comunicação através do protocolo HTTP, uma forma de conversa entre dispositivos, que permite transferir hipertexto de um lado a outro, daí o nome Hyper Text Transport Protocol. O BAM-Backend utiliza a arquitetura RESTful para receber requisições e dar um retorno, ambos em formato JSON.

O módulo BAM-Backend fornece comunicação através de rotas, que são utilizadas para executar as ações em CRUD (Create (criar), Read (ler), Update (atualizar) e Delete (excluir)), as rotas disponíveis são /Ambience, /Configuration, /Device e /DeviceData. A rota /Ambience é utilizada para obter e manipular as regras de ambiência de forma que seja possível definir a temperatura e umidade ideal em determinados períodos entre duas datas. A rota /Configuration é utilizada para manipular através do BAM-Dashboard o range de temperatura e o tempo de cada ciclo e fornecer essas informações ao módulo BAM-Controller que irá realizar os controles da ambiência utilizando estes parâmetros. A rota /Device foi desenvolvida para manipular os dados de dispositivos BAM-Device permitindo que seja possível utilizar mais de um dispositivo para realizar as leituras e armazenar a informação registrada por cada dispositivo separadamente. A rota /DeviceData é utilizada para obter e manipular os dados de temperatura e umidade captados por um BAM-Device. Todos os dados recebidos pelo BAM-Backend são armazenados em um banco de dados Postgres.

2.2 BAM-Dashboard

Para o módulo BAM-Dashboard foi utilizado o template gratuito Light Bootstrap Dashboard React sob a *MIT License* o mesmo foi desenvolvido em linguagem Javascript (JS). Para executar o código JS foi usado o framework REACT.

O template foi customizado para a pesquisa e foram agregadas algumas bibliotecas de forma que possa atender as necessidades, uma delas foi a Axios. A biblioteca Axios permite que seja feita a comunicação com o módulo BAM-Backend onde toda a comunicação é feita na arquitetura REST a qual os dados de requisição e retorno são em formato JSON. Com os recursos do template foram criadas para manipular e apresentar as informações obtidas pelo módulo BAM-Backend três páginas que são Dashboard, Configurações e Ambiência.

A página inicial apresenta as informações em *real time* bem como dá uma visão geral dos dados gerados pelos módulos BAM-Controller e BAM-Device. São apresentados as informações de Temperatura e Umidade atuais enviadas pelo BAM-Device, nível Atual do BAM-Controller e dois gráficos sendo um com os dados de temperatura e outro da umidade.

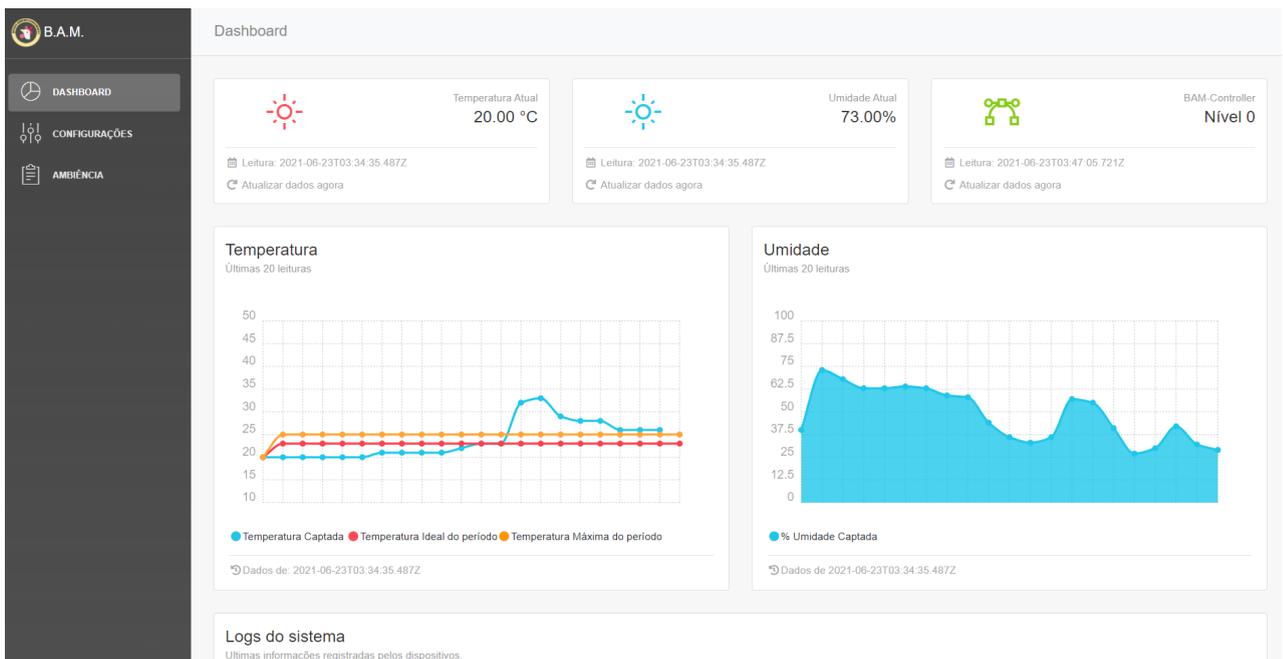


Figura 3. Tela inicial do módulo BAM-Dashboard

A página de Configurações permite configurar o range de temperatura em graus *celsius* e o delay do ciclo em milissegundos, estes parâmetros são requisitados pelo módulo BAM-Controller e servem como base para o cálculo de temperatura máxima (temperatura atual + range de temperatura) e o tempo de cada ciclo (intervalo de mudança de níveis) respectivamente, bem como para o cálculo do tempo em que os dispositivos do nível ficarão ativos (delay do ciclo /2).



Figura 4. Página de configurações do módulo BAM-Dashboard

A página de Ambiência tem como finalidade definir a temperatura e umidade de um determinado período entre duas datas, não sendo possível configurar mais de uma ambiência para o mesmo período. Os dados parametrizados nesta página são recebidos

pelo módulo BAM-Controller mediante a requisição e servem como base para definir o nível de controle.

ID	TÍTULO	DATA DE INICIO	DATA DO FIM	TEMPERATURA	UMIDADE
1	Ambiência Teste	28/05/2021	31/12/2021	20°C	40%

Figura 5. Página de ambiência do módulo BAM-Dashboard

2.3 BAM-Device

No desenvolvimento do protótipo BAM-Device estudou-se acerca do hardware, os equipamentos e tecnologias empregados, bem como a montagem do circuito eletrônico. No Software foi necessário desenvolver o algoritmo para se conectar a uma rede Wifi, enviar os dados de temperatura e umidade captados pelo sensor DHT11, serializar essas informações e se comunicar com o BAM-Backend em formato JSON. Este módulo foi o primeiro a ser desenvolvido para se familiarizar com as tecnologias de hardware que foram aplicadas também no módulo BAM-Controller o qual envolveu uma estrutura muito mais complexa. O custo total para montagem do dispositivo aproximou-se de R\$ 130,00.



Figura 6. Módulo BAM-Device no protótipo

2.3.1 Montagem do Circuito do BAM-Device

No desenvolvimento lógico do circuito eletrônico utilizou-se uma placa controladora Lolin NodeMcu V3 Esp8266 com uma alimentação entre 3.3 volts (V) a 5V e uma dimensão de 49x25x7mm. Para monitorar a temperatura e umidade foi acoplado um sensor DHT11 com uma alimentação de 3V a 5V e um tamanho de 23x12x5mm, que é devidamente calibrado em laboratório e proporciona extrema precisão de leitura. O coeficiente de calibração é armazenado em forma de programa na memória OTP (One Time Programed), que é utilizado pelo processo interno de leitura do sensor. A interface de comunicação serial de uma via tornou a integração do sensor ao microcontrolador rápida e possibilitou logo nas primeiras tentativas a leitura dos dados.

Para alimentação da placa base do NodeMCU foi utilizada uma bateria de 9V. A placa base pode ser alimentada com 6V a 24V através da entrada para fonte de alimentação, posteriormente a tensão de entrada é convertida para 3.3 volts a 5V que é a tensão de operação da placa controladora que como citada fica acoplada na placa base.

A Lolin NodeMcu V3 Esp8266 comunica-se mediante a interface *Inter-Integrated Circuit (I²C)*. Para estabelecer essa comunicação, conectaram-se os pinos S, GND e V do DHT11 aos pinos D4, GND e 3V da placa base, respectivamente, que são as portas usadas para comunicar os equipamentos de acordo com a ambiência atual, o módulo pode ser visualizado através da Figura 7.

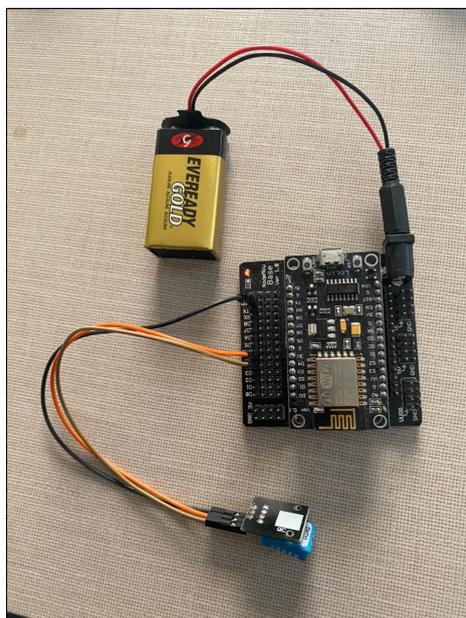


Figura 7. Circuito do Módulo BAM-Device

2.3.2 Desenvolvimento do Software do BAM-Device

No desenvolvimento do software foi usado o Arduino IDE em sua versão 1.8.13, com a linguagem de programação C++. Para auxiliar no desenvolvimento da aplicação foram utilizadas quatro bibliotecas, são elas: DHT.h, ESP8266HTTPClient, ESP8266WiFi.h, WifiClient.h. Todas as bibliotecas foram disponibilizadas pelo próprio Arduino IDE, por isso, não foi possível identificar a versão.

A biblioteca DHT.h foi usada para estabelecer a comunicação da placa controladora com o sensor via interface I²C e também para escrever e ler valores captados. A ESP8266HTTPClient foi usada para se comunicar com os outros módulos através do protocolo HTTP. A ESP8266WiFi.h foi usada para estabelecer conexão a uma rede Wi-Fi.

O código do sistema pode ser separado em três etapas, são elas: Configuração (método setup), Execução (método loop) e registro dos dados lidos (método deviceData). Na etapa de configuração é feita a conexão a uma rede Wi-Fi. A etapa de execução voltou-se à leitura dos dados de temperatura e umidade e posteriormente a leitura é chamado o método deviceData. O método deviceData realiza a conexão ao módulo BAM-Backend mediante o seu endereço endpoint, os dados são serializados em um JSON e enviados através de requisição HTTP no formato POST para o BAM-Backend que dá um retorno código 200 caso tenha sucesso.

2.4 BAM-Controller

No desenvolvimento do protótipo BAM-Controller estudou-se acerca do hardware, os equipamentos e tecnologias empregados, bem como a montagem do circuito eletrônico. No Software foi necessário desenvolver o algoritmo para se conectar a uma rede Wifi, obter as configurações e os parâmetros de ambiência do BAM-Backend e de acordo com esses parâmetros ativar os dispositivos protótipos de controle. Os dispositivos de controle são: cinco coolers (ventiladores), um servo motor (cortinas) e uma bomba de água (nebulizador). Os cinco ventiladores e a bomba de água são acionados através de um módulo relé de 8 canais, a cortina é aberta pelo servo motor através do microcontrolador. Os controles são divididos em seis níveis e o seu acionamento é feito de forma gradual de um nível por vez. Periodicamente pelo tempo configurado no BAM-Dashboard é feita a comunicação com o BAM-Backend para se obter os parâmetros de ambiência por meio desses dados são realizadas verificações para definir o nível de controle. Toda a comunicação com o BAM-Backend é realizada através do protocolo HTTP no formato JSON. Este módulo foi o último a ser desenvolvido devido a sua quantidade de dispositivos de controle conectados ao *Hardware* e complexidade do código do *Software* e comunicação com outros dispositivos. O custo total para a montagem do dispositivo aproximou-se de R\$ 400.

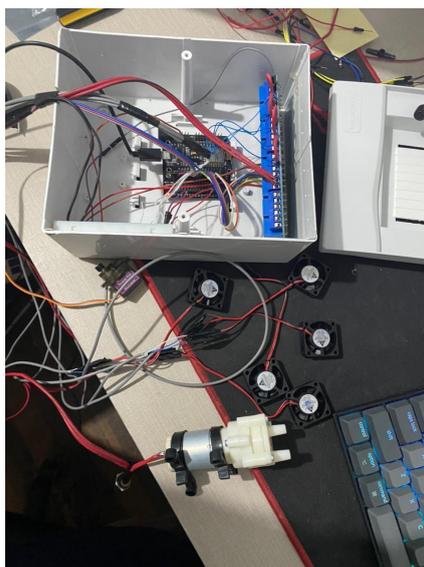


Figura 8. Módulo BAM-Controller

2.4.1 Montagem do Circuito do BAM-Controller

No desenvolvimento lógico do circuito eletrônico utilizou-se uma placa controladora Lolin NodeMcu V3 Esp8266 com uma alimentação entre 3.3 volts (V) a 5V e uma dimensão de 49x25x7mm. Para os controles foram utilizados cinco coolers Gostime modelo GDT3010S12B de 5V e 0.1A para ventilação, o micro servo motor Tower Pro MG90S para cortina e uma bomba de água Gaotu de 12V para nebulização. Para acionamento dos ventiladores e da bomba de água foi utilizado um módulo relé de 8 canais.

A Lolin NodeMcu V3 Esp8266 comunica-se mediante a interface *Inter-Integrated Circuit (I²C)*. Para estabelecer essa comunicação, conectaram-se os pinos do Esp8266 às entradas do módulo relé bem como feita a ligação da GND de ambos os dispositivos. O módulo foi alimentado com uma fonte de 5V.

Os ventiladores foram conectados no relé ao circuito normalmente aberto. Por serem acionados individualmente cada ventilador possui sua saída no módulo relé. O positivo de cada ventilador foi alimentado pela fonte de 5V e o negativo foi conectado à saída do circuito normalmente aberto, por utilizarem da mesma alimentação todos os pinos comum deste circuito foram jumpeados de forma que se pudesse otimizar o protótipo.

O circuito normalmente aberto do relé que foi utilizado para a bomba nebulizadora foi alimentado por uma fonte de 12V.

O servo motor foi conectado diretamente no NodeMCU e alimentado pela tensão de 3V do próprio dispositivo.

Um cabo USB foi utilizado para alimentar a placa base do NodeMCU através dele é possível alimentar a placa controladora e caso esteja conectada ao computador é também possibilita ver informações pela porta serial.



Figura 9. Circuito do Módulo BAM-Controller

2.4.2 Desenvolvimento do Software do BAM-Controller

No desenvolvimento do software foi usado o Arduino IDE em sua versão 1.8.13, com a linguagem de programação C++. Para auxiliar no desenvolvimento da aplicação foram utilizadas cinco bibliotecas, são elas: ESP8266HTTPClient, ESP8266WiFi.h, WifiClient.h, ArduinoJson.h e Servo.h. Todas as bibliotecas foram disponibilizadas pelo próprio Arduino IDE, por isso, não foi possível identificar a versão.

A biblioteca ESP8266HTTPClient foi usada para se comunicar com os outros módulos através do protocolo HTTP. A ESP8266WiFi.h foi usada para estabelecer conexão a uma rede Wi-Fi. A WifiClient.h foi utilizada por possuir alguns recursos como *Client Wifi*, e a ArduinoJson.h para serializar e deserializar os dados em formato JSON. A biblioteca Servo.h foi utilizada para manipular o Servo Motor.

O código do sistema pode ser separado em quatro etapas, são elas: Configuração (método setup), Execução (método loop), manipulação dos níveis (método ativarNivel), registros de logs (método log). Na etapa de configuração é feita a conexão a uma rede Wi-Fi, configurados os pinos D1, D2, D3, D4, D5 e D6 que são dos ventiladores e da bomba nebulizadora como saída e definido o pino D7 para a cortina. Todas as saídas foram definidas para um valor inicial e todos os dispositivos como inativos. A etapa de execução voltou-se à obtenção de parâmetros e ativação dos controles de acordo com algumas verificações. É realizada uma requisição no formato GET pelo protocolo HTTP ao módulo BAM-Backend mediante o seu endereço endpoint, é recebido um retorno código 200 caso tenha sucesso juntamente com os dados no formato JSON. Os dados são deserializados em variáveis que são usadas como base para definir o funcionamento dos controles. Os valores obtidos são: temperatura e umidade ideal para o período definidos na página de Ambiente no módulo BAM-Dashboard, range de temperatura e delay cycle definidos na página de Configurações também do BAM-Dashboard e temperatura e umidade atuais captados pelo módulo BAM-Device.

Após todos os dados serem obtidos é feito o cálculo do tempo ativo com a fórmula delay do ciclo /2. Em seguida é calculada a temperatura máxima que o ambiente deve ter com a fórmula `ambiente_temperature + configuration_range_temperature`. Com os cálculos realizados é iniciada a verificação para definir se um nível será aumentado, diminuído ou mantido.

Para definir o nível é utilizado o algoritmo da Figura 10, este verifica se a temperatura atual é superior a máxima calculada anteriormente, caso a condição seja atendida é verificado se o nível atual é inferior ao nível 6, caso seja o nível é aumentado, caso não seja quer dizer que já está no nível máximo e o mesmo é reativado, após finalizado o ciclo é ativado o delay do ciclo onde o sistema fica em standby até que as mesmas verificações sejam feitas novamente. Caso a temperatura atual seja inferior ou igual a temperatura máxima definida para o período, o nível é diminuído novamente e o delay do ciclo é ativado assim respectivamente. O sistema realiza a troca de níveis de forma automática e caso a temperatura se mantenha dentro do parametrizado o nível continua sendo diminuído até que atinja o nível zero, neste nível todos os controles são desligados e a cortina é totalmente fechada. É possível visualizar o algoritmo dos níveis na (Figura 10).

```
if(temperaturaAtual > maxTemperature){
    if(nivelAtual < 6) {
        nivelAtual+=1;
    }
    ativarNivel(nivelAtual, tempoAtivo);
}else{
    if(nivelAtual > 0) {
        nivelAtual-=1;
    }
    ativarNivel(nivelAtual, tempoAtivo);
}
```

Figura 10. Algoritmo que define o nível

2.4.3 Níveis de Controle do BAM-Controller

Como citado anteriormente o BAM-Controller possui sete dispositivos acoplados que são cinco ventiladores, uma bomba nebulizadora e uma cortina. Os dispositivos são acionados conforme seu nível. No nível 1 é acionado o ventilador 1 e a cortina é aberta em 20%. No nível 2 são acionados os ventiladores 1 e 2 e a cortina é aberta em 40%. No nível 3 são acionados os ventiladores 1, 2 e 3 e a cortina é aberta em 60%. No nível 4 são acionados os ventiladores 1, 2, 3 e 4 e a cortina é aberta em 80%. No nível 5 são acionados os ventiladores 1, 2, 3, 4 e 5 e a cortina é totalmente aberta. Caso o sistema tenha passado por todos os níveis de controle e mesmo assim a temperatura não chegou no ideal é ativado o nível 6 o qual ativa os ventiladores 1, 2, 3, 4, 5, abre toda a cortina e a bomba nebulizadora é também acionada. O sistema muda apenas um nível por vez aumentando ou diminuindo os mesmos de forma gradual e os dispositivos de cada nível permanecem ativos durante metade do tempo de delay de cada ciclo.

2.5 Ambiente para Simulação

Foi desenvolvido um protótipo parcialmente vedado para realizar os experimentos. O mesmo foi utilizado para acoplar o BAM-Controller e o BAM-Device de forma que se pudesse produzir calor internamente e este fosse dissipado para fora pelos dispositivos de controle podendo manter assim a temperatura e umidade interna. Internamente foi revestido de manta asfáltica para que pudesse manter o calor pelo máximo de tempo possível, os ventiladores foram posicionados como na ventilação do tipo túnel (Figura 1), uma das extremidades permaneceu aberta como entrada de ar e a outra fechada pela cortina desenvolvida, o protótipo pode ser observado na Figura 11.

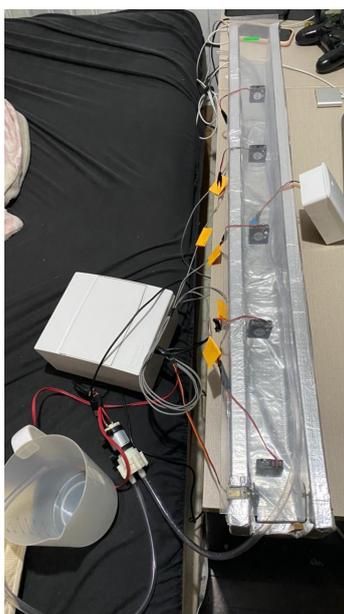


Figura 11. Circuito do Módulo BAM-Controller

3. Experimentos

Nesta pesquisa realizaram-se experimentos no ambiente de simulação que foi desenvolvido. Para realizar os experimentos foi necessário aquecer o interior do protótipo e para isso foi utilizado uma lâmpada incandescente a mesma foi acesa para aquecer o interior do protótipo. Foram realizados dois testes um com a lâmpada acesa durante 20 minutos com tempo de leitura de 5 minutos e sem o módulo BAM-Controller, os resultados podem ser observados na Tabela 1, e outro com a Lâmpada acesa durante 20 minutos com delay de cada ciclo de 150 segundos com o BAM-Controller ligado e o resultado pode ser observado na Tabela 2. Foram configurados no módulo BAM-Dashboard a temperatura ideal de 22°C com range de 3°C, o módulo deve aumentar o nível quando a temperatura for superior a 25°C.

Tabela 1. Leitura do BAM-Device sem o módulo de controle

Modulo	Leitura	Tempo
BAM-Device	Temperatura Atual: 21.00 Umidade Atual:35.00	11:41:48.707936

BAM-Device	Temperatura Atual: 20.00 Umidade Atual:34.00	11:46:49.135861
BAM-Device	Temperatura Atual: 27.00 Umidade Atual:20.00	11:51:49.580879
BAM-Device	Temperatura Atual: 28.00 Umidade Atual:8.00	11:56:49.890602
BAM-Device	Temperatura Atual: 30.00 Umidade Atual:14.00	12:01:50.300417
BAM-Device	Temperatura Atual: 32.00 Umidade Atual:11.00	12:06:50.748314
BAM-Device	Temperatura Atual: 30.00 Umidade Atual:4.00	12:11:51.470654

Tabela 2. Resultados Obtidos com o módulo BAM-Controller Ativo

Módulo	Leitura	Tempo
CICLO 1		
BAM-Controller	Temperatura Máxima: 25, Temperatura atual: 23 Umidade Atual:12.00 Nível 0 - Controles desligados, cortina fechada!	13:18:37.247628
CICLO 2		
BAM-Controller	Temperatura Máxima: 25 Temperatura atual: 25 Nível 0 - Controles desligados, cortina fechada!	13:28:38.493992
CICLO 3		
BAM-Controller	Temperatura Máxima: 25 Temperatura atual: 37 Umidade Atual:01.00 Nível 1 - Ventilador 1 ligado! Cortina aberta 20%!	13:33:39.546253
CICLO 4		

BAM-Controller	Temperatura Máxima: 25 Temperatura atual: 32 Umidade Atual:03.00 Nivel 3 - Ventiladores 1, 2 e 3 ligados! Cortina aberta 60%!	13:44:57.295789
CICLO 5		
BAM-Controller	Temperatura Máxima: 25 Temperatura atual: 33 Umidade Atual:02.00 Nivel 5 - Ventiladores 1, 2, 3, 4 e 5 ligados! Cortina aberta 100%!	13:52:32.669745
CICLO 6		
BAM-Controller	Temperatura Máxima: 25 Temperatura atual: 37 Umidade Atual:05.00 Nivel 6 - BOMBA DE ÁGUA e todos os ventiladores ligados! Cortina aberta 100%!	13:57:53.416571
CICLO 7		
BAM-Controller	Temperatura Maxima: 25 Temperatura atual: 28 Umidade Atual:22.00 Nivel 6 - BOMBA DE ÁGUA e todos os ventiladores ligados! Cortina aberta 100%!	14:00:56.15257
CICLO 8		
BAM-Controller	Temperatura Máxima: 25 Temperatura atual: 28 Umidade Atual:37.00 Nivel 6 - BOMBA DE ÁGUA e todos os ventiladores ligados! Cortina aberta 100%!	14:03:54.15297
CICLO 9		
BAM-Controller	Temperatura Máxima: 25 Temperatura atual: 26 Umidade Atual:42.00% Nivel 6 - BOMBA DE ÁGUA e todos	14:07:06.638374

	os ventiladores ligados! Cortina aberta 100%!	
CICLO 10		
BAM-Controller	Temperatura Máxima: 25 Temperatura atual: 24 Umidade Atual:39.00% Diminuiu o nível, nível atual:5 Nível 5 - Ventiladores 1, 2, 3, 4 e 5 ligados! Cortina aberta 100%!	14:10:08.874143

Como apresentado na Tabela 2, o módulo operou conforme o esperado, aumentando os níveis quando a temperatura foi superior a 25°C. No momento que o nível 6 foi alcançado a umidade lida foi de 5% e ao chegar no ciclo 9, três ciclos depois a umidade chegou a 42%. No ciclo 10 a temperatura chegou a 24 °C evidenciando que a temperatura está dentro do nível ideal, o nível foi diminuído, após esse momento os níveis foram diminuídos até atingir o nível 0, neste todos os controles foram desligados e a cortina foi fechada.

4. Resultados e Discussões

O protótipo demonstrou ser um sistema de monitoramento e controle eficaz dado as proporções que foram utilizadas. A comunicação com o BAM-Backend foi através da arquitetura RESTful e os métodos foram testados sem autenticação para facilitar no momento dos testes porém foi desenvolvido toda a estrutura de autenticação com dados criptografados utilizando JWT e caso opte por implantar esse nível de segurança do protótipo basta utilizar este *middleware* nos métodos e incluir a autenticação nos *firmwares* dos módulos de *hardware*.

No desenvolvimento do protótipo de simulação a ideia era construir um ambiente que tivesse as características de um aviário com sistema de ventilação por pressão positiva o qual o ar entra por uma das extremidades e sai pela outra com o auxílio de ventiladores.

O módulo BAM-Device o qual é responsável pela captação de temperatura e umidade foi desenvolvido com comunicação sem fio com o intuito de possibilitar a leitura de mais de um sensor pois em um ambiente de aviário há uma grande extensão e há uma diferença de temperatura de acordo com a posição do sensor e para isso seria interessante possuir vários sensores ao longo do aviário captando a temperatura e fazer uma média.

O módulo BAM-Controller este que funciona como atuador foi desenvolvido com o intuito de ser o mais próximo possível de um produto comercial. Durante os testes realizados o mesmo manteve sua comunicação estável, o delay do ciclo contribui

para que não sejam necessários muitos recursos do servidor tornando o protótipo estável com o uso de uma boa rede de Internet. O uso do relé de até 250V possibilita substituir os ventiladores atuais por dispositivos mais potentes, podendo assim aplicar em um ambiente real sem ter que fazer muitas modificações. Os resultados obtidos com o uso do módulo como evidenciados na Tabela 2 foram satisfatórios.

O BAM-Dashboard possibilitou a apresentação dos dados captados pelos dispositivos bem como alterar os parâmetros utilizados pelo controlador.

Todo o sistema comportou-se bem durante os testes realizados. O servidor trabalhou de forma satisfatória na comunicação de dados entre os dispositivos, não houve nenhum tipo de instabilidade ou falha na comunicação, o protocolo HTTP juntamente com a arquitetura REST além de ser fácil de utilizar funcionou corretamente. O BAM-Device realizou todas as leituras e se comunicou com o BAM-Backend que serviu os dados ao BAM-Controller que se comportou de acordo com todos os parâmetros recebidos e efetuou todas as mudanças de nível conforme o esperado. A rede local sem fio não influenciou no desempenho da aplicação pois como há um delay em cada ciclo os dispositivos não consomem muito recurso do servidor e as operações com banco de dados demonstraram serem leves com o uso do Postgres.

Os dados coletados estão representados na Tabela 1. É possível visualizar que os controles foram ativados respeitando o tempo de delay de cada ciclo que foi previamente configurado.

O pesquisador Oliveira (2019) empregou a implementação e avaliação de um sistema automatizado de monitoramento e controle térmico utilizando tecnologia IoT para testar se é possível aplicar em um aviário convencional. Como na presente pesquisa [Oliveira 2019] também desenvolveu uma aplicação web para gestão do ambiente e apresentação dos dados. O mesmo também construiu módulos sensores e módulos atuadores o que nessa presente pesquisa foi intitulado de BAM-Device e BAM-Controller respectivamente. Além do desenvolvimento, Oliveira (2019) aplicou a pesquisa em um aviário do laboratório de avicultura da FZEA/USP. O sensor que realizou as medições bem como o microcontrolador foram os mesmos utilizados nesta pesquisa. De acordo com todos os estudos realizados por Oliveira (2019) concluiu-se que os resultados apresentados permitem que um sistema automatizado pode ser empregado no controle térmico uma vez que através do sistema desenvolvido foi possível manter a homogeneidade térmica. Ainda segundo o pesquisador, o sistema se mostrou eficiente de acordo com os protocolos implementados nos módulos sensores e atuadores embarcados no ambiente no qual a pesquisa foi aplicada.

De acordo com Camusso et al. (2021) as condições ideais para o frango de corte adultas são com temperaturas de 15-18°C e 22-25°C e umidade relativa de 50 a 70%. O pesquisador desenvolveu um protótipo com o intuito de manter o ambiente interno nessas faixas. Para o protótipo utilizou um ambiente fechado com um sensor interno e como forma de aquecer o ambiente e gerar uma carga térmica utilizou uma lâmpada incandescente. Dentre os testes realizados foram testados o ambiente sem ventilação forçada e sem carga térmica e o resultado foi de uma temperatura de 29°C e umidade de 58%, sem ventilação forçada e com carga térmica a temperatura atingiu 50°C e a umidade 25%, com ventilação forçada e com carga térmica a temperatura atingida foi de

35°C e umidade de 43%. Mesmo não atingindo as faixas ideais, foi possível verificar que com o uso do sistema automatizado a temperatura e a umidade mudaram consideravelmente mesmo com carga térmica.

O pesquisador Wibowo et al. (2018) desenvolveu um protótipo de alimentação dos frangos, monitoramento e controle da temperatura e umidade utilizando conceitos de Internet das Coisas. Dentre os testes realizados o mesmo testou o mecanismo para alimentação desenvolvido onde as 8 horas da manhã o servo abria a comporta a cada 5 segundos e fechava novamente. Para monitorar a temperatura e umidade o mesmo empregou um mecanismo em que caso a temperatura estivesse abaixo de 25°C uma lâmpada incandescente era acesa e quando a temperatura estivesse acima de 30°C a lâmpada desliga. Quando a temperatura for maior ou igual a 25°C e menor ou igual a 30°C a lâmpada incandescente e o ventilador estarão desligados indicando que a temperatura está nas condições ideais. Todos os dados gerados pelo sistema foram processados pelo Raspberry Pi e quando o cliente acessa a página web os dados são apresentados semelhante a presente pesquisa. Após todos os testes o autor concluiu que, pode ser criado um sistema automático de alimentação e monitorar a temperatura e umidade aplicando o conceito de Internet das Coisas com Raspberry Pi. A presente pesquisa também provou que é possível monitorar a temperatura e umidade, porém não foi empregado o controle da alimentação.

Os estudos de Zhang e Chen (2020) acerca de monitorar o ambiente de produção de frango baseado em Android, utilizou um sistema de autoria própria que armazena um histórico e gera avisos, bem como monitora os dados em tempo real utilizando a plataforma android e um hardware para captação de dados. Após realizados vários testes com a experiência de usuário, os resultados apresentados mostraram que o sistema é estável e de confiança e possui uma boa acurácia, quando o parâmetro ambiental é excedido um aviso é gerado o que possibilita que os responsáveis podem monitorar remotamente como na presente pesquisa.

5. Conclusão

Este trabalho teve por objetivo desenvolver o protótipo para monitorar e controlar a temperatura e umidade em ambientes de produção de frangos de corte. A aplicação demonstrou ser prática e fácil de utilizar, visto que muitos dos trabalhos correlatos utilizam de aviários reais o que torna o ambiente para o monitoramento e controle mais complexo, e com resultados mais realísticos. A dinâmica utilizada para troca de níveis possibilita que seja feita a mudança de forma gradual, ao que tange a ventilação, o que tem relação direto ao conforto dos animais. Bastando apenas ter uma conexão de rede *Wi-fi* para conectar todos os módulos e alimentação para o módulo controller a aplicação se mostrou funcionar de forma eficiente. Os resultados alcançados confirmaram a eficácia na comunicação de dados entre os dispositivos, porém o uso de ventiladores menos potentes dificultaram no resfriamento do protótipo, em relação a comunicação foi possível monitorar e controlar os dispositivos e registrar toda a comunicação e ações feitas pelos módulos.

Levando em conta que os frangos produzem calor com seu movimento foi necessário utilizar algo que pudesse produzir calor e também ser resfriado e para isso empregou-se uma lâmpada incandescente no interior do protótipo, durante o uso pelo

mesmo ser estreito e a parte superior ser de policarbonato aconteceu de em razão do aquecimento houve um derretimento do policarbonato na região mais próxima da lâmpada, isso foi contornado utilizando uma madeira na parte mais próximo do local.

Como base nos conhecimentos adquiridos acerca do nicho bem como com os resultados obtidos sugere-se para trabalhos futuros utilizar ventiladores com maior potência e maior quantidade dos mesmos em um ambiente maior; desenvolver a cortina com maior proximidade do que é aplicado no ambiente real; aplicar o sistema em um aviário real; realizar um estudo de outras tecnologias que podem ser aplicadas dentro de ambientes de produção de frangos de corte como métodos de verificar a saúde do frango de forma individual e com baixo custo ou também empregar a alimentação.

Referências

- ABPA (2021). Relatório Anual 2021. https://abpa-br.org/wp-content/uploads/2021/04/ABPA_Relatorio_Anual_2021_web.pdf, [acesso em março 21].
- Andreazzi, M. A., Pinto, J. S., Santos, J. M. G. Dos, et al. (2018). DESEMPENHO DE FRANGOS DE CORTE CRIADOS EM AVIÁRIO CONVENCIONAL E DARK-HOUSE. REVISTA DA UNIVERSIDADE VALE DO RIO VERDE, v. 16, n. 1.
- Avicultura Industrial (2021). Sistema que controla clima de aviário pode evitar perdas na produção de frangos. <https://www.aviculturaindustrial.com.br/imprensa/sistema-que-controla-clima-de-aviario-pode-evitar-perdas-na-producao-de-frangos/20201119-122403-y795>, [acesso em abril 21].
- Balduino Jorge, M. P., Gomes Marçal Vieira Vaz, R., Calixto da Silva, M. and De Araujo Costa, H. (1 dec 2020). DESEMPENHO DE FRANGOS DE CORTE CRIADOS SOBRE DOIS TIPOS DE CAMA E DUAS DENSIDADES. DESAFIOS - Revista Interdisciplinar da Universidade Federal do Tocantins, v. 7, n. Especial, p. 221–228.
- CNPSA (2021). Instalações. <http://www.cnpsa.embrapa.br/SP/aves/Ventila-artificial-2.html>, [acesso em abril 21].
- Melo, A. da S., Fernandes, Raimunda Thyciana Vasconcelos, Marinho, J. B. M., et al. (nov 2016). Relação temperatura e nutrição sobre o desempenho de galinhas poedeiras. Pubvet, v. 10, n. 11, p. 855–860.
- Oliveira, M. E. De (4 out 2019). Implementação e avaliação de um sistema automatizado de monitoramento e controle térmico em um aviário convencional utilizando tecnologia IoT. Biblioteca Digital de Teses e Dissertações da Universidade de São Paulo.
- Wibowo, E. P., Wibisono, A., Nawangsari, S. and Suritalita, A. (1 oct 2018). Prototype of feeding devices, temperatures and humidity monitoring at broiler chickens breeders with the internet of things concept. In Proceedings of the 3rd International

Conference on Informatics and Computing, ICIC 2018. . Institute of Electrical and Electronics Engineers Inc. , [accessed on Jun 6].

Yoppy, Arjadi, R. H., Candra, H., Prananto, H. D. and Wijanarko, T. A. W. (2 jul 2018). RSSI Comparison of ESP8266 Modules. In 2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar, EECCIS 2018. . Institute of Electrical and Electronics Engineers Inc. , [accessed on Jun 1].

Zhang, W. and Chen, C. (20 nov 2020). Design and Implementation of an Android-based Broiler Production Environment Monitoring System. In 2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering, AUTEEE 2020. . Institute of Electrical and Electronics Engineers Inc. , [accessed on Jun 1].