



Automatização de Tanques da Aquaponia: Nivelador de água

Nomes: Kauê Yotsuo Pereira Takada, Giovani Carregalo

Professor: Wilian França Costa

Universidade Presbiteriana Mackenzie (UPM)
Rua da Consolação, 930 Consolação, São Paulo - SP, 01302-907 – Brazil

Abstract. *This article aims to enunciate a project that aims to use elements from the field of Internet of Things for the theme of aquaponics. The project aims to solve a problem faced in this scenario: the water gap in the fish tank caused by evaporation, causing risks to animals. The implementation of the IoT will automate the regulation of the water level in these tanks through level sensors programmed by an Arduino.*

Resumo. *Este artigo tem como objetivo enunciar um projeto que tem como objetivo usar elementos do campo da Internet das Coisas para o tema da aquaponia. O projeto tem como objetivo resolver um problema enfrentado nesse cenário: o desnível da água do tanque dos peixes provocado pela evaporação, causando riscos aos animais. A implementação da IoT irá automatizar a regularização do nível da água nesses tanques através de sensores de nível programados por um Arduino.*

1. Introdução

A tecnologia está extremamente presente em todos os tipos de Mercado nos dias de hoje, porém existem cenários onde a tecnologia é pouco usada por não ser viável ou com um preço elevado.

Um panorama onde isso é encontrado é na aquaponia onde segundo o canal do Youtube Aquaponia MS(2019) o problema do desnivelamento da água dos tanques na maioria das vezes não é automatizado e quando é, o Sistema é caro.

Através dessa perspectiva o projeto tem como objetivo criar um sistema que irá nivelar a água dos tanques usados na aquaponia com materiais de baixo custo controlados pelo celular por um protocolo MQTT.

2. Materiais e Métodos

Arduino Uno - Serve como centro do projeto, onde ele dá comandos aos itens conectados a ele e em sequência certa

Figure 1-Arduino Uno



Fonte: Google(2021)

Modulo relê - Conecta o arduino e a bomba, o arduino dá um comando ao modulo relê e o modulo liga ou desliga a bomba.

Figure 2-Modulo relê



Fonte: Google(2021)

Sensor de nível - No projeto há 2 sensores de nível, sendo 1 deles para medir a altura ideal da água de um dos tanques e o outro serve para medir a altura recomendada para funcionar sem problemas no reservatório.

Figure 3-Sensor de nível



Fonte: Google(2021)

LED - o LED serve para indicar quando a água do reservatório está abaixo do nível recomendado.

Figure 4-LED



Fonte: Google(2021)

Bomba - a bomba serve para enviar a água do reservatório para o tanque quando o nível de água está abaixo do recomendado.

Figure 5- Bomba



Fonte: Google(2021)

Fios - liga todo o projeto, dando energia aos componentes.

Figure 6-Fios



Fonte: Google(2021)

Resistor de 220 ohms - o resistor serve para transformar a energia do arduino para o LED.

Figure 7-Resistor



Fonte: Google(2021)

Cabo de energia - o cabo de energia será o que dá energia a todo o projeto.

Figure 8-Cabo de energia



Fonte: Google(2021)

Placa de ensaio - a placa de ensaio serve para organizar o projeto, e se possível reduzir a quantidade de fios utilizados.

Figure 9-Placa de ensaio

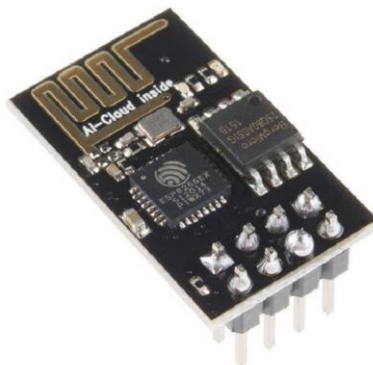


Fonte: Google(2021)

MQTT – Para o uso do MQTT foi utilizado dois tipos duas bibliotecas da adafruit e alguns componentes.

ESP-01 – Responsável pela comunicação com a internet ao Arduino e suas devidas programações

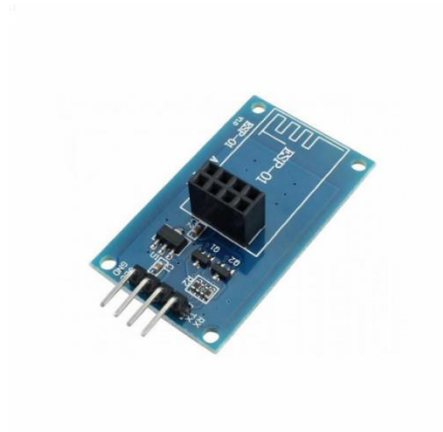
Figure 11-Esp-01



Fonte: [Google](#)(2021)

Adaptador – Simplifica a conexão com o ESP-01

Figure 12- Adaptador Esp-01

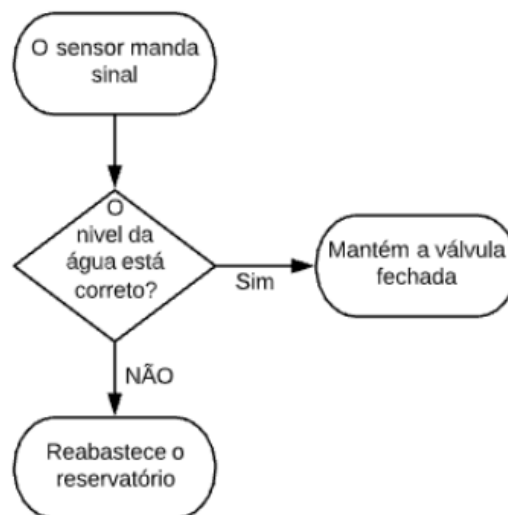


Fonte: Google(2021)

2.1 Diagramas

Fluxograma: Diagrama para entendimento do processo do Sistema.

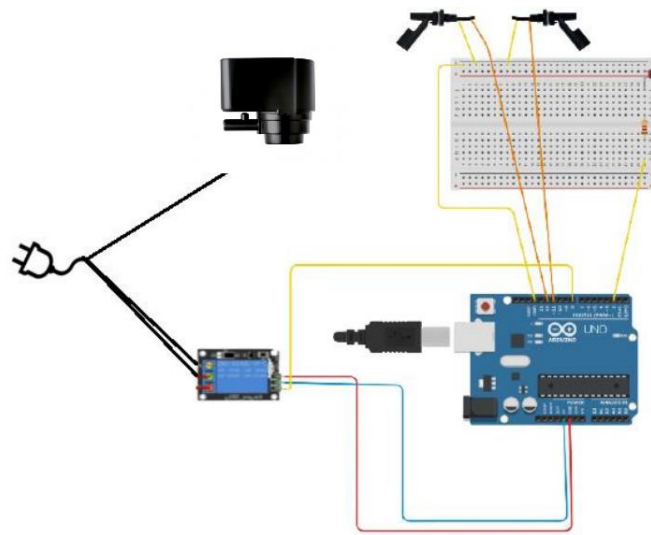
Figure 13- Fluxograma do sistema



Fonte: Autoral

Tinkercad: Diagrama do hardware do projeto, mostrando as ligações que devem ser feitas para funcionamento do mesmo.

Figure 14- Tinkercad do sistema



Fonte: Autoral

2.3 Software

O software consta com duas sketches para o funcionamento completo do sistema:

2.3.1 Skecth Master – Código para conexão do mqtt e sua devida configuração para o acesso no Adafruit

Figure 15- Skecth Master

```
46 lines (46 xloc) 1.19 KB
1 #include <Ada.h>
2 #include "config.h"
3 #define endereco 500
4 #define tempoAtualizacao 500
5 AdafruitIO_Feed *reserv = io.feed("reserv");
6 AdafruitIO_Feed *bomba = io.feed("bomba");
7 Ada arduinoSlave;
8 void configuraMQTT();
9 unsigned long controleTempo = 0;
10 bool estadoReserv = LOW;
11 bool estadoBomba = LOW;
12 bool estadoReservAnt = LOW;
13 bool estadoBombaAnt = LOW;
14 void setup() {
15   Serial.begin(9600);
16   while (! Serial);
17   configuraMQTT();
18   arduinoSlave.begin(0, 2);
19   Serial.println("Fin Setup");
20 }
21 void loop() {
22   io.write();
23   if (millis() > controleTempo + tempoAtualizacao) {
24     estadoReserv = arduinoSlave.variableRead(endereco, 0);
25     if (estadoReserv != estadoReservAnt) {
26       reserv->write(estadoReserv);
27     }
28     estadoReservAnt = estadoReserv;
29     estadoBomba = arduinoSlave.variableRead(endereco, 1);
30     if (estadoBomba != estadoBombaAnt) {
31       bomba->write(estadoBomba);
32     }
33     estadoBombaAnt = estadoBomba;
34     controleTempo = millis();
35   }
36 }
37 void configuraMQTT() {
38   Serial.print("Conectando ao Adafruit IO");
39   io.connect();
40   while (io.status() < AIO_CONNECTED) {
41     Serial.print(".");
42     delay(500);
43   }
44   Serial.println();
45   Serial.println(io.statusText());
46 }
```

Fonte: Autoral

2.3.2 Skecth Slave – Código Utilizado para o funcionamento completo do mecanismo de reposição e monitoramento de água dos tanques com os devidos loops de verificação para o acionamento e definições dos pinos

Figure 16- Skecth Slave

```
47 lines (47 sloc)  1.22 KB
1  #include <A2a.h>
2  #define endereco 0x08
3  #define pinSensorTanque 11
4  #define pinSensorReserv 12
5  #define pinHele  8
6  #define pinAlerta 3
7  bool estadoTanque = HIGH;
8  bool estadoReserv = HIGH;
9  A2a arduinoMaster;
10 void setup() {
11     arduinoMaster.begin(endereco);
12     arduinoMaster.onReceive(receberDados);
13     arduinoMaster.onRequest(enviarDados);
14     pinMode(pinSensorTanque, INPUT_PULLUP);
15     pinMode(pinSensorReserv, INPUT_PULLUP);
16     pinMode(pinHele, OUTPUT);
17     pinMode(pinAlerta, OUTPUT);
18     Serial.begin(9600);
19 }
20 void loop() {
21     estadoTanque = digitalRead(pinSensorTanque);
22     estadoReserv = digitalRead(pinSensorReserv);
23     if (estadoReserv) {
24         if (estadoTanque) {
25             digitalWrite(pinHele, LOW);
26             arduinoMaster.wireWrite(1, LOW);
27         }
28         else {
29             digitalWrite(pinHele, HIGH);
30             arduinoMaster.wireWrite(1, HIGH);
31         }
32         digitalWrite(pinAlerta, LOW);
33         arduinoMaster.wireWrite(0, LOW);
34     }
35     else {
36         digitalWrite(pinHele, LOW);
37         arduinoMaster.wireWrite(1, LOW);
38         digitalWrite(pinAlerta, HIGH);
39         arduinoMaster.wireWrite(0, HIGH);
40     }
41 }
42 void receberDados() {
43     arduinoMaster.receiveData();
44 }
45 void enviarDados() {
46     arduinoMaster.sendData();
47 }
```

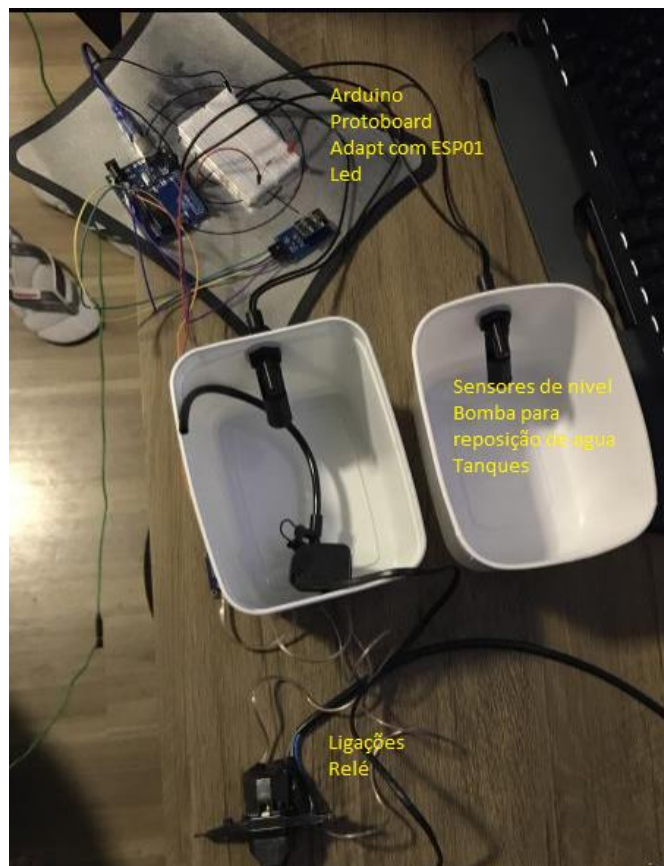
Fonte: Autoral

3. Resultados

3.1 Descrição

O projeto foi criado em menor escala para mostrar o funcionamento do sistema, como esperado o tanque de reposição despejou a água tratada no tanque que representaria o tanque dos peixes, além disso todo o percurso foi monitorado e representado no site da adafruit coordenado pelo MQTT implantado no hardware.

Figure 16- Projeto completo e seus componentes



Fonte: Autoral

3.2 Problemas enfrentados

Durante a criação do projeto ocorreram alguns problemas com o funcionamento do MQTT, porém após algumas pesquisas e modificações nas instalações obtivemos sucesso em criar a Plataforma na Adafruit.

4. Conclusões

4.1 Objetivos

O objetivo foi completado uma vez que todas as funcionalidades propostas foram atendidas (monitoramento e reposição dos tanques).

4.2 Vantagens e desvantagens do projeto:

Uma vantagem que o projeto tem é o baixo custo, levando em conta o tanto de funcionalidades que ele desempenha e o tempo que o mesmo economiza para o usuário.

As desvantagem desse projeto estão diretamente relacionadas aos pequenos defeitos que ele apresenta em sua montagem e que serão melhor explicados no próximo tópico.

4.3 Melhorias para o projeto:

O projeto poderia ter alguns sistemas e funções que melhorariam seu desempenho como um sensor de temperatura da água e um código com delay para não ter o acionamento da bomba por ações “não naturais”.

5. Links do Github e do vídeo explicativo do Youtube.

- Github: [Link](#)
- Youtube: [Link](#)

6. Referências

Referências

Aquaponia MS. Youtube, 2019. Disponível em: <<https://www.youtube.com/channel/UChPkO6sN9lG40NveH0OyJAw>>. Acesso em: 26 de mar. 2021.

Oliveira, Andrade. Teste Sensor de nível de água interruptor e mini boia. 2017. Disponível em: <<https://www.youtube.com/watch?v=hyC9qSxDhwg>>. Acesso em: 26 de mar. 2021