

UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RICARDO FERREIRA PINHEIRO

**DESENVOLVIMENTO DE UM SISTEMA DE IRRIGAÇÃO
AUTOMATIZADO PARA PEQUENAS LAVOURAS UTILIZANDO
ARDUINO**

VIÇOSA
2020

RICARDO FERREIRA PINHEIRO

**DESENVOLVIMENTO DE UM SISTEMA DE IRRIGAÇÃO
AUTOMATIZADO PARA PEQUENAS LAVOURAS UTILIZANDO
ARDUINO**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 402 – Projeto de Engenharia Elétrica II – e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Denilson Eduardo Rodrigues

VIÇOSA
2020

RICARDO FERREIRA PINHEIRO

**DESENVOLVIMENTO DE UM SISTEMA DE IRRIGAÇÃO
AUTOMATIZADO PARA PEQUENAS LAVOURAS UTILIZANDO
ARDUINO**

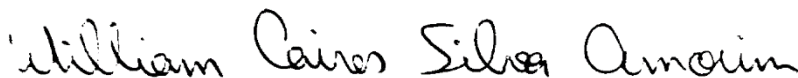
Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 402 – Projeto de Engenharia Elétrica II – e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 09 de dezembro de 2020.

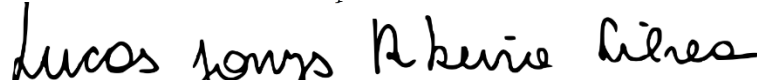
1 COMISSÃO EXAMINADORA



Prof. Dr. Denilson Eduardo Rodrigues – Orientador
Universidade Federal de Viçosa



Prof. Me. William Caires Silva Amorim – Primeiro Avaliador
Universidade Federal de Viçosa



Bel. Lucas Jonys Ribeiro Silva – Segundo Avaliador
Universidade Federal de Viçosa

“Sonhos determinam o que você quer. Ação determina o que você conquista.”
(Aldo Novak)

Dedico este trabalho aos meus pais, irmão, tio e tia maternos, e todos meus amigos.

Agradecimentos

Gostaria primeiramente de agradecer a Deus, por sempre me guiar nas minhas escolhas e não me deixar desistir nos momentos de dificuldades.

Aos meus pais, Cléria e Antônio, meus maiores apoiadores em toda essa jornada, ao meu irmão, Henrique, melhor amigo que a vida me deu, ao meu tio Cleber e tia Carmem, por todo carinho.

Aos meus irmãos de vida, Gabriell Hott, Gabriel Rabelo, Daniel Sossai, Danilo Cunha, Alexandre Pires, Pedro Henrique, Matheus Lacerda e João Augusto, que espero sempre levar em todo caminho que irei trilhar adiante.

Aos meus amigos de curso, com grande carinho pelo Vitor Thinassi e Lucas Jonys, por todos os momentos de dificuldades compartilhados juntos, tendo sempre a parceria em destaque.

Aos todos meus amigos da Lateria e Garras Cheerleaders da Atlético das Engenharias, por todo convívio, alegria e conquistas compartilhadas.

Aos professores e funcionários do Departamento de Engenharia Elétrica da UFV, com ênfase ao professor e orientador Denilson, pelos ensinamentos e orientação ao meu projeto final de curso.

Resumo

Um dos grandes desafios da agricultura é o cumprimento do recurso hídrico necessário para crescimento saudável dos plantios. Nesse contexto, a irrigação vem como uma forte solução para contornar a questão da sazonalidade encontrada na água oriunda das chuvas. Com o alto crescimento populacional, cresce também a quantidade dos plantios, o que por consequência demanda um maior volume de recurso hídrico necessário. Porém, por ser limitado, tem-se a necessidade do uso do recurso hídrico de forma consciente, evitando desperdício.

O objetivo desse trabalho foi a construção de um protótipo de irrigação automatizada, baseado em um microcontrolador Atmega2560, em uma plataforma de prototipagem Arduino. O protótipo é composto por um módulo de ethernet *Shield* W5100, no qual fornece acesso à rede IP via protocolo TCP/UDP para o microcontrolador, operando uma interface *Web* para acesso do usuário às informações no formato tempo real, sensor de temperatura e umidade DHT22, sensor de luminosidade, sensor de chuva e umidade do solo, uma válvula solenoide como atuador elétrico do circuito hídrico, sendo este ligado a um módulo relé, um medidor de fluxo de água, um módulo de relógio e calendário DS3231 e um visor LCD para exibição de informações dos sensores.

O protótipo foi configurado para realizar um ciclo de irrigação por dia, despejando 0,46 litro de água por ciclo. Porém, há configurado uma regra de exceção para início do ciclo, sendo umidade do solo acima de 80% ou a indicação do sensor de chuva acima de 90%, o que causa a suspensão do início do ciclo.

A interface *Web* disponibiliza por meio de uma página codificada no formato HTML as informações de leitura de todos os sensores presentes, com uma taxa de atualização de 5 segundos, sendo a página acessível por todos os usuários conectados na mesma rede LAN no qual o protótipo está conectado.

Palavras-chave: Microcontrolador; Prototipagem; Arduino; Irrigação; Automação, Rede IP.

Abstract

One of the great challenges of agriculture is the fulfillment of the water resources necessary for the healthy growth of plantations. In this context, irrigation comes as a strong solution to circumvent the question of seasonality found in water from rain. With the high population growth, the number of plantations also grows, which consequently demands a greater volume of necessary water resources. However, because it is limited, there is a need to use water resources consciously, avoiding waste.

The objective of this work was the construction of an automated irrigation prototype, based on an Atmega2560 microcontroller, on an Arduino prototyping platform. The prototype is composed of a Shield W5100 ethernet module, in which it provides access to the IP network via TCP / UDP protocol for the microcontroller, operating a Web interface for user access to information in real time format, DHT22 temperature and humidity sensor, light sensor, rain and soil moisture sensor, a solenoid valve as an electrical actuator for the water circuit, which is connected to a relay module, a water flow meter, a DS3231 clock and calendar module and an LCD display for display information from the sensors.

The prototype was configured to carry out an irrigation cycle per day, pouring 0.46 liters of water per cycle. However, an exception rule has been set for the beginning of the cycle, with soil moisture above 80% or the indication of the rain sensor above 90%, which causes the suspension of the beginning of the cycle.

The web interface makes available, through a page encoded in HTML format, the reading information of all sensors present, with a refresh rate of 5 seconds, the page being accessible by all users connected to the same LAN network in which the prototype is connected.

Keywords: Microcontroller, Prototyping; Arduino; Irrigation; Automation, IP Network.

Sumário

1	Introdução.....	12
1.1	Impactos do Projeto	14
1.2	Objetivos.....	14
2	Materiais e Métodos	16
2.1	Software	16
	Fritzing	16
	Arduino IDE.....	16
2.2	Materiais	17
	Arduino MEGA 2560.....	17
	RTC DS3231	18
	Arduino Ethernet Shield W5100.....	18
	Válvula Solenoide de Entrada de Água - 180°	19
	Sensor de Umidade e Temperatura AM2302 DHT22.....	19
	Sensor de Umidade do Solo Higrômetro.....	20
	Sensor de Fluxo de Água 1/2" YF-S201b.....	20
	Módulo Sensor de Chuva YL-83 9SS18.....	20
	Display LCD Nokia 5110 <i>Backlight</i> Azul	21
	Fonte Ajustável Protoboard 3PS02	21
	Módulo Relé 5V SRD-05VDC-SL-C (2 Canais).....	22
2.3	Metodologia.....	22
3	Resultados	25
4	Conclusão	28
5	Referências Bibliográficas	29
	Apêndice A.....	31
	Apêndice B	35

Lista de Figuras

Figura 1 - Método de desvio de cursos d'água para irrigação de plantios	12
Figura 2 - Irrigação automatizada utilizando aspensor	13
Figura 5 - Interface do software Fritzing 0.9.3b.64	16
Figura 6 - Interface gráfica do software Arduino IDE	17
Figura 7 - Arduino MEGA 2560	17
Figura 8 - Módulo RTC DS3231	18
Figura 9 - Ethernet Shield W5100	18
Figura 10 – Válvula Solenoide de Entrada de Água - 180°	19
Figura 11 – Sensor de Umidade e Temperatura AM2302 DHT22.....	19
Figura 12 – Sensor de Umidade do Solo Higrômetro	20
Figura 13 – Sensor de Fluxo de Água 1/2" YF-S201b.....	20
Figura 14 - Sensor de Chuva 9SS18.....	21
Figura 15 - Display LCD Nokia 5110 <i>Backlight</i> Azul	21
Figura 16 - Fonte Ajustável Protoboard 3PS02.....	22
Figura 17 – Módulo Relé 5V SRD-05VDC-SL-C (2 Canais).....	22
Figura 18 - Esquemático do Protótipo	24
Figura 19 - Composição central (elétrica) do protótipo	25
Figura 20 - Composição externa (hídrica) do protótipo	25
Figura 21 - Válvula Solenoide e Sensor de Vazão acoplados ao encanamento	26
Figura 22 - Sensores de Chuva e Solo acoplados ao ambiente da planta doméstica.....	26
Figura 23 - Página WEB de leitura dos dados	27

Lista de Tabelas

Tabela 1 - Aferição do Sensor de Vazão YF-S201	27
--	----

1 *Introdução*

Desde a pré-história o homem já utilizava a irrigação em seus cultivos. Ele entendeu que as plantas ficavam mais verdes e produziam mais quando eram irrigadas além das águas de chuvas. Por meio de desvio dos cursos d'água, tornou-se possível também se estabelecer em regiões áridas e semiáridas de forma permanente (TESTEZLAF, 2017). A Figura 1 ilustra o método de desvio do curso d'água por meio de pequenos canais de irrigação para o cultivo agrícola.

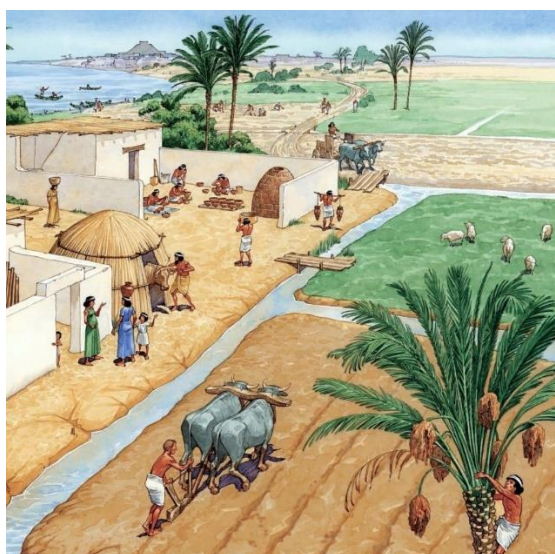


Figura 1 - Método de desvio de cursos d'água para irrigação de plantios (SILVA SOUZA)

Nesse contexto, define-se como irrigação o conjunto de técnicas ou métodos destinados a deslocar água no tempo ou no espaço, de forma a alterar a disponibilidade hídrica agrícolas de cada região. A irrigação tem por objetivo corrigir a distribuição sazonal natural das chuvas (E. F. W. LIMA).

A irrigação é dita como uma tecnologia imprescindível no processo de aumento de produção de bens agrícolas. Para cada tipo de região, sua adoção é requerida de forma diferente baseando-se especialmente quanto a disponibilidade hídrica. Em regiões desérticas e áridas, onde a precipitação anual é inferior a 250 mm, a irrigação é obrigatória de forma constante, para que se tenha um cultivo saudável. Já em regiões onde a precipitação é acima de 600 mm, a irrigação é obrigatória somente em períodos curtos de secas (TESTEZLAF, 2017).

Dessa forma, o manejo racional da irrigação é uma das preocupações atuais, devido ao problema da disponibilidade de recurso hídrico em todo o mundo e a

necessidade de se economizar, visto que diversos produtores, por não adotarem um método controlado de irrigação, usualmente irrigam de forma excessiva seus plantios, temendo que a sua cultura sofra algum estresse hídrico. Esse excesso de irrigação tem como consequência o desperdício de água e de energia elétrica, visto que é necessário o uso de bombas elétricas para o bombeamento de água (E. F. W. LIMA).



Figura 2 - Irrigação automatizada utilizando aspersor (VÍQUA)

Os três principais tipos de irrigação utilizados atualmente são aspersão, localizada e superficial. Aspersão é caracterizada pela distribuição no formato de gotas sobre o solo, com semelhança a chuva, conforme visto na Figura 2. Localizada é caracterizada pela aplicação da água na área ocupada pelas raízes das plantas. Superficial é caracterizada pela condução da água no sistema por meio de bacias d'água.

Pensando então em sistemas controlados e automáticos, o uso de um sistema de irrigação automatizado surge da finalidade tornar a irrigação mais eficaz e menos custosa, em termos de mão de obra e energia elétrica, uma vez que um sistema controlado e automático é definido como sendo um sistema programado para executar determinada função pré-determinada, de forma repetitiva e controlada, por meio da realimentação negativa, ou *feedback* (C. DORF e H. BISHOP, 2001).

Para isso, o mercado apresenta diversos tipos de microprocessadores ou microcontroladores, de baixo até alto custo, com características e capacidades de processamento diversas. A sua importância em um projeto de sistemas controlados e automáticos se deve ao fator de serem eles a unidade lógica, responsável por realizar todo o gerenciamento de leitura dos sinais dos sensores e envio de comandos elétricos, baseado na lógica de programação que lhe é inserido. O Arduino se apresenta como uma solução de prototipagem de baixo custo e alta integração, podendo ser facilmente utilizado por meio do seu Ambiente de Desenvolvimento Integrado próprio e uma vasta quantidade de artigos e tutoriais disponíveis nas comunidades da internet (ARDUINO).

As principais vantagens e desvantagens na implantação de um sistema de irrigação automatizado, do ponto de vista econômico e benefícios para a implantação, são (GORNAT e W. L. C., 1990):

- Economia de energia, visto que as bombas para irrigações são de altas potências e baixa eficiência energética. Otimizando o seu uso, há redução do custo de energia utilizada;
- Redução de mão de obra, pois haverá a substituição dos atuadores manuais por atuadores elétricos;
- Economia de água, devido ao seu uso de forma precisa e exata, por meio do monitoramento do volume de água irrigado;
- Custo elevado, uma vez que a implantação de um sistema de irrigação possui um alto custo de investimento agregado;
- Mão de obra técnica, para realizar a manutenção do sistema, devido a sua complexidade.

1.1 Impactos do Projeto

O presente projeto pretende a criação de um sistema de irrigação automatizada para agricultores de pequena escala, de baixo custo e fácil manutenção. Há muito desgaste de tempo útil humano para monitoramento e execução da irrigação manual, podendo este tempo ser direcionado a outras tarefas complexas requeridas por uma área de agricultura.

Além disso, o projeto permite a redução do desperdício, fazendo assim a irrigação do plantio somente em dias necessários, com carga hídrica previamente definida.

Portanto, esse protótipo tem por impacto o controle e gerenciamento de forma automática do recurso hídrico para a plantação.

1.2 Objetivos

Esse trabalho tem por objetivo desenvolver um protótipo de um sistema de irrigação automatizado para plantas domésticas, em fase de bancada, utilizando-se a plataforma de prototipagem Arduino MEGA2560 com integração via WEB para monitoramento dos dados impressos pelo protótipo.

Dado o objetivo geral, têm-se como objetivos específicos:

- Desenvolver um código C# no Arduino IDE que torne o Arduino capaz de realizar as leituras dos sensores e fazer o controle da válvula solenoide de forma automatizada;
- Desenvolver um sistema de monitoramento do usuário em rede TCP/IP, apresentando dados em tempo real de todos os sensores compostos no projeto;
- Estudar a viabilidade da implementação do protótipo em uma pequena plantação.

2 *Materiais e Métodos*

2.1 Software

Fritzing

O Fritzing, Figura 3, é um software de código aberto, desenvolvido na Fachhochschule Potsdam (*University of Applied Sciences Potsdam*), na Alemanha, com o intuito de ser um ambiente para desenvolvimento de protótipos de circuitos eletrônicos de maneira simples e descomplicada, permitindo criar layouts de circuitos eletrônicos, desenhos de placas de circuitos impressos e simular códigos Arduino, por meio de sua integração ao Arduino IDE. Teve sua primeira versão (0.1b) disponibilizada ao público em novembro de 2008 (KENSHIMA).

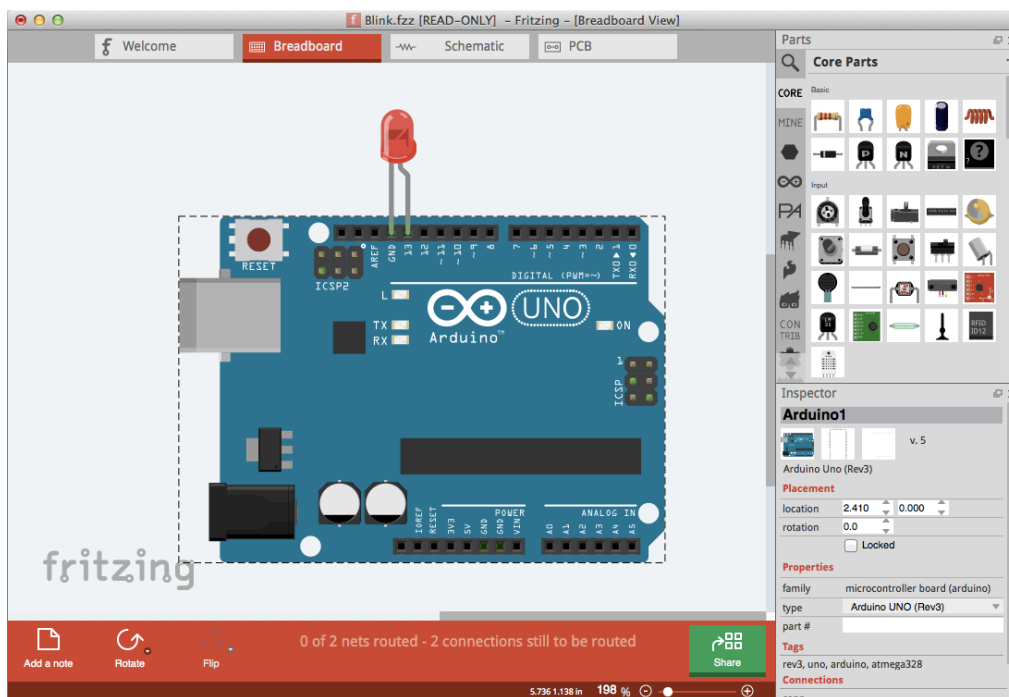


Figura 3 - Interface do software Fritzing 0.9.3b.64 (KENSHIMA)

Arduino IDE

O Arduino IDE (*Integrated Development Environment*, em português, Ambiente de Desenvolvimento Integrado) é um software multiplataforma, escrito na linguagem JAVA, utilizado para escrever e enviar programas para a placa Arduino, no qual suporta as linguagens C e C++ (ARDUINO).

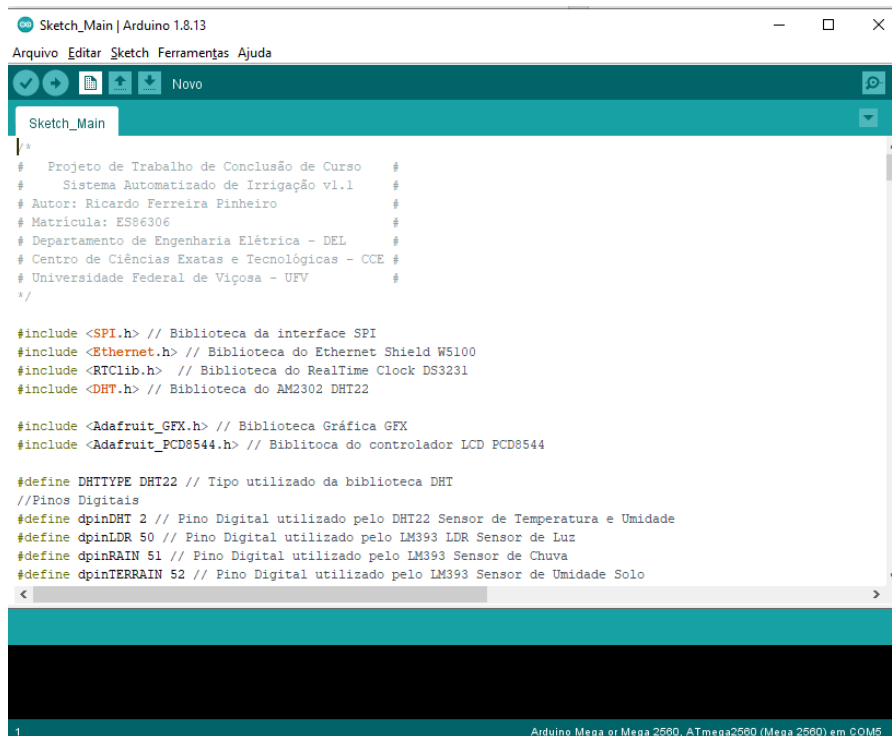


Figura 4 - Interface gráfica do software Arduino IDE (ARDUINO)

2.2 Materiais

Arduino MEGA 2560

O Arduino MEGA 2560 é uma plataforma de prototipagem *open-source* baseada no microcontrolador ATmega2560, da Microchip Technology. Seu robusto hardware dispõe de um conversor analógico para digital de seis canais, disponibilizando assim para o usuário entradas e saídas digitais e entradas analógicas. O seu Ambiente de Desenvolvimento (IDE) open-source é distribuído gratuitamente por sua desenvolvedora, sendo disponível para as plataformas Mac OS X, Windows e Linux, e possui como linguagem de programação de referência C/C++ (ARDUINO).



Figura 5 - Arduino MEGA 2560 (ARDUINO)

RTC DS3231

O RTC DS3231 é um módulo *I²C real-time clock* (RTC) de baixo custo e alta precisão, com um cristal oscilador compensado por temperatura TCXO. O dispositivo possui uma bateria, no qual é responsável por manter a contagem quando a fonte de energia principal é interrompida.



Figura 6 - Módulo RTC DS3231 (K-TECHNICS SOLUTIONS)

O módulo RTC contabiliza segundos, minutos, horas, dia, mês, ano e dia da semana, no qual é automaticamente ajustado para meses com menos de 31 dias, incluindo correções para ano bissexto. O módulo opera tanto em formato 24 horas ou 12 horas, possuindo um indicador de AM/PM. É possível ajustar dois alarmes diários, além de programar uma onda quadrada como saída. Sua comunicação é feita da forma serial através de um canal bidirecional *I²C* (K-TECHNICS SOLUTIONS).

Arduino Ethernet Shield W5100

O módulo Arduino Ethernet Shield W5100 é uma placa baseada no chip ethernet da WIZnet ethernet W5100 que fornece acesso à rede (IP) nos protocolos TCP/UDP e é facilmente utilizado usando a biblioteca Ethernet Library e SD Library. Este Shield suporta até quatro conexões de socket simultâneas. O módulo também disponibiliza de um slot para cartão micro-SD, que pode ser utilizado para armazenar arquivos de um servidor na rede (BAÚ DA ELETRÔNICA).



Figura 7 - Ethernet Shield W5100 (BAÚ DA ELETRÔNICA)

Válvula Solenoide de Entrada de Água - 180°

A Válvula Solenoide é um atuador elétrico de fluxo de água. Ele opera na forma de normalmente fechado, ou seja, é necessário energizá-lo com 12 Volts para que seja permitido a passagem do fluxo de água, e quando não energizado, opera como bloqueador de fluxo (ELETROGATE).



Figura 8 – Válvula Solenoide de Entrada de Água - 180° (ELETROGATE)

Sua pressão de operação é de 0,2 a 8 kgf/cm², onde a vazão mínima é igual a 7 L/min à 0,2kgf/cm², e 40 L/min à 8 Kgf/cm². A temperatura máxima do fluido suportado é de 60°C (ELETROGATE).

Sensor de Umidade e Temperatura AM2302 DHT22

O sensor de Temperatura e Umidade AM2302 DHT22 é formado por um sensor de umidade capacitivo e um termistor. Possui tensão de operação entre 3-5VDC (5,5VDC máximo), com faixas de medições de temperatura entre -40° a +80°C e de umidade relativa entre 0 a 100%, com precisões de medições de temperatura e umidade de $\pm 0,5$ °C e $\pm 2,0\%$ UR, respectivamente (FILIFEFLOP C.E.).

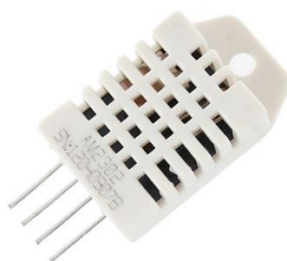


Figura 9 – Sensor de Umidade e Temperatura AM2302 DHT22 (FILIFEFLOP C.E.)

Sensor de Umidade do Solo Higrômetro

O sensor de Umidade do Solo – Higrômetro serve como detector de variação de umidade no solo. Utilizando um comparador LM393, ele aponta saída em nível lógico baixo quando a umidade do solo está alta. O ajuste fino é realizado por meio do potenciômetro presente no LM393. Possui tensão de operação entre 3-5VDC (FILIPEFLOP C.E.).

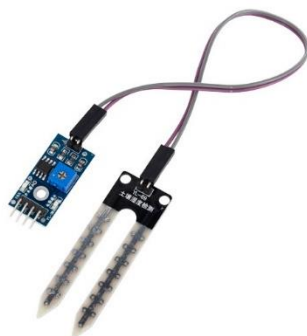


Figura 10 – Sensor de Umidade do Solo Higrômetro (FILIPEFLOP C.E.)

Sensor de Fluxo de Água 1/2" YF-S201b

O sensor de Fluxo de Água 1/2" YF-S201b é um sensor do tipo Efeito Hall, no qual possui o formato de um cata-vento com um ímã acoplado, que em conjunto ao sensor hall envia um sinal PWM. Através desses pulsos emitidos é possível obter a vazão de água. Possui tensão de operação entre 5-24VDC, faixa de fluxo entre 1-30 l/min, pressão máxima de 2,0 MPa e exatidão de 10% (FILIPEFLOP C.E.).



Figura 11 – Sensor de Fluxo de Água 1/2" YF-S201b (FILIPEFLOP C.E.)

Módulo Sensor de Chuva YL-83 9SS18

O sensor de Chuva 9SS18 é um sensor do tipo resistivo, no qual utiliza um comparador LM393 para indicar quando há presença de gotículas de água na superfície do sensor por meio do nível lógico Baixo. O ajuste fino é realizado por meio do

potenciômetro presente no LM393. Possui tensão de operação entre 3-5VDC, e saídas Analógica e Digital (FILIPEFLOP C.E.).



Figura 12 - Sensor de Chuva 9SS18 (FILIPEFLOP C.E.)

Display LCD Nokia 5110 *Backlight* Azul

O display LCD Nokia 5110 com backlight azul é um visor com resolução de 84x48 pixels, com suporte a texto e gráficos, no qual utiliza o driver controlador CMOS LCD PCD8544 de baixa carga. Possui tensão de operação entre 3-5VDC (FILIPEFLOP C.E.).



Figura 13 - Display LCD Nokia 5110 *Backlight* Azul (FILIPEFLOP C.E.)

Fonte Ajustável Protoboard 3PS02

A fonte ajustável Protoboard 3PS02 é uma fonte DC-DC, sendo uma solução prática para alimentação de módulos e componentes acoplados a protoboards. Possui tensão de entrada de 6,5-12VDC, com saídas 3,3VDC e 5VDC selecionadas por meio de *jumpers*, e um botão de liga e desliga a alimentação (FILIPEFLOP C.E.).



Figura 14 - Fonte Ajustável Protoboard 3PS02 (FILIPEFLOP C.E.)

Módulo Relé 5V SRD-05VDC-SL-C (2 Canais)

O módulo Relé 5V (2 Canais) possui capacidade de acionamento de cargas de 200V AC, no qual cada canal possui um LED indicador de estado de saída do relé (FILIPEFLOP C.E.).



Figura 15 – Módulo Relé 5V SRD-05VDC-SL-C (2 Canais) (FILIPEFLOP C.E.)

2.3 Metodologia

Para o estudo de caso, uma bancada de teste foi montada com uma planta doméstica e o protótipo, sendo assim, a planta doméstica o objeto (alvo) a ter a sua irrigação controlada.

A planta doméstica foi equipada com o sensor de umidade de solo (Higrômetro), e irrigada por meio do método superficial, sendo a água canalizada por meio de um cano PVC interligado à Válvula Solenoide e ao Sensor de Fluxo de Água, com origem a uma torneira residencial. O sensor de chuva foi posicionado no arredor da base da planta doméstica, expondo-o à chuva. O módulo de umidade e temperatura e luz ficou acoplado a base estrutural do protótipo.

O Arduino MEGA será programado de forma a fazer varredura constante das variáveis trabalhadas (temperatura, umidade, vazão e estado da Válvula Solenoide), realizando o acionamento do processo de irrigação por meio de comandos pré-programados via software.

Para calibração do Sensor de Fluxo de Água YF-S201b serão realizado três coletas de todo o volume de água liberado por cada ciclo de irrigação (liberação e bloqueio da Válvula Solenoide pelo Arduino), e o volume será pesado por meio de uma balança digital de cozinha, e o volume então obtido pela equação (1), onde “m” é a massa de água pesada e “d” é a densidade da água.

$$v = \frac{m}{d} \quad (1)$$

O módulo Ethernet Shield W5100 será utilizado para criação de uma página web, no qual disponibilizará dados e informações das leituras de cada sensor do protótipo, em tempo real, possibilitando ao usuário um maior monitoramento do processo de irrigação sem a necessidade de intervenção física ao local.

Com o intuito de se isolar o circuito de controle do circuito de carga, fez-se necessário a utilização de duas fontes de tensão contínua de 12 VDC independentes. O circuito de controle caracteriza-se por Arduino, sensores e módulos, e o circuito de carga o atuador, nesse caso, a válvula solenoide.

O ciclo de irrigação será programado para ocorrer a cada dia, iniciando-se as oito horas da manhã (horário de Brasília), com despejo de 0,46 litros de água por ciclo. O ciclo será interrompido ou impedido de ser iniciado caso a umidade do solo esteja acima de 80% ou o indicador do sensor de chuva indique 90% em sua leitura.

O esquemático resultante do protótipo, composto por meio do software Fritzing, é exibido na Figura 16.

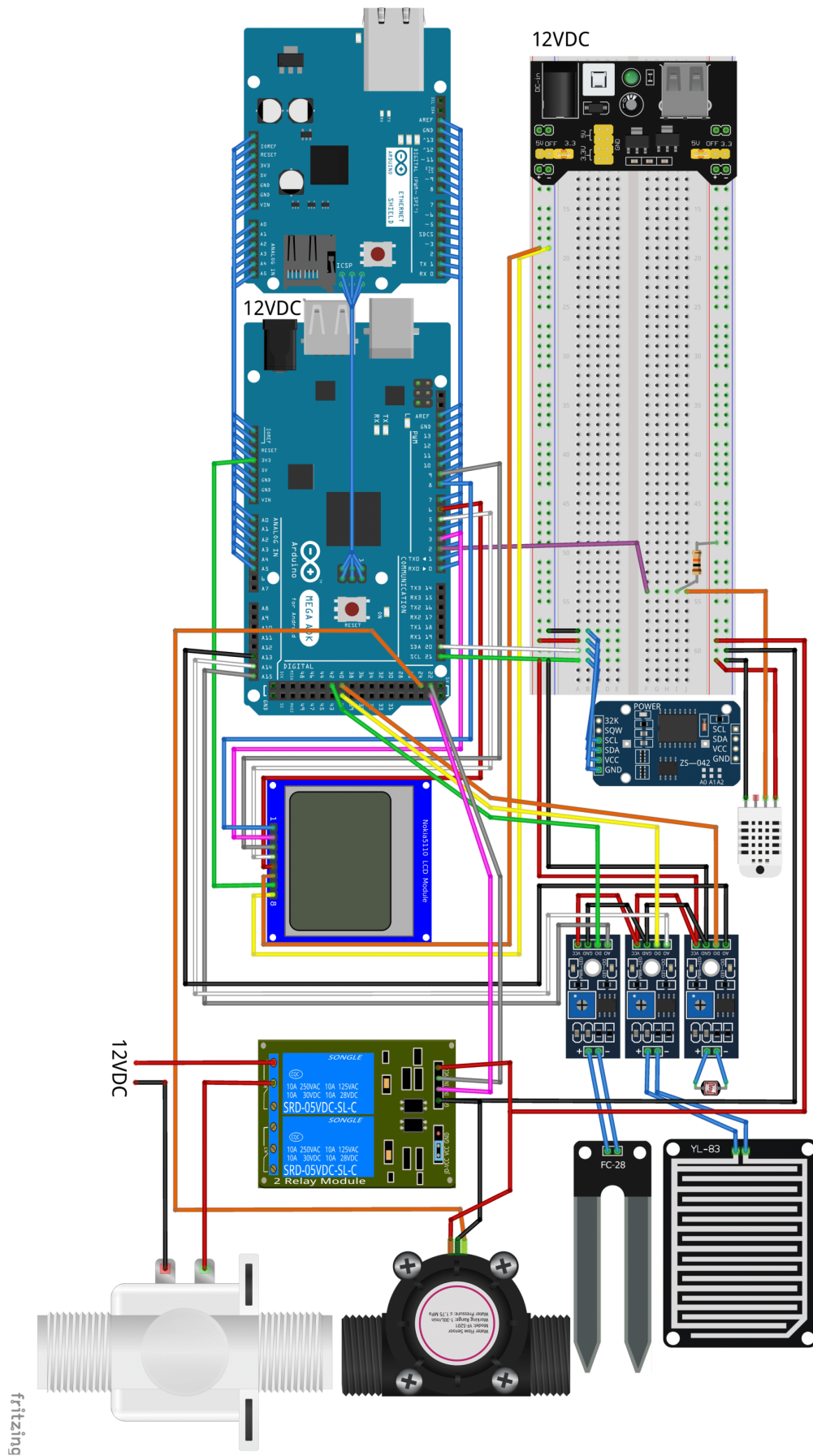


Figura 16 - Esquemático do Protótipo

3 Resultados

Nesta seção serão exibidos os resultados da montagem do protótipo em bancada, utilizando as metodologias e materiais descritos neste trabalho.

O protótipo montado em bancada é exibido nas Figura 17, Figura 18 e Figura 20.

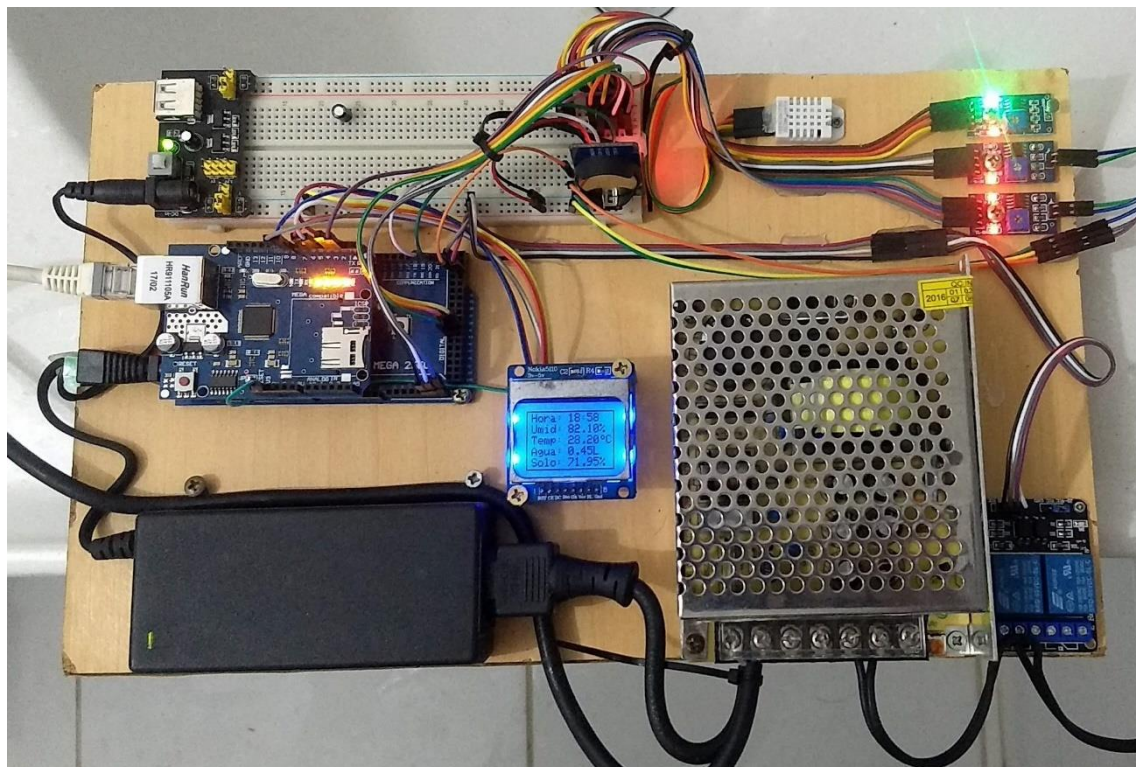


Figura 17 - Composição central (elétrica) do protótipo

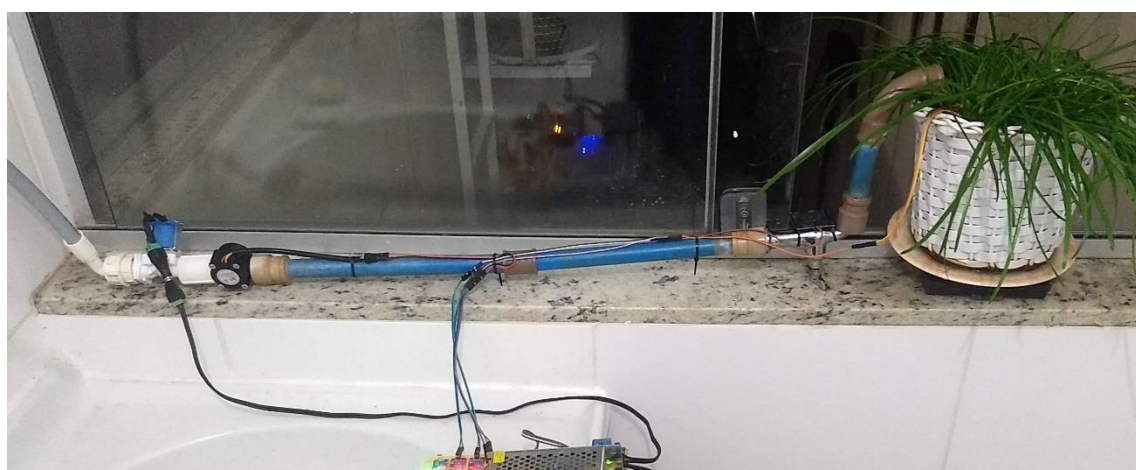


Figura 18 - Composição externa (hídrica) do protótipo



Figura 19 - Válvula Solenoide e Sensor de Vazão acoplados ao encanamento



Figura 20 - Sensores de Chuva e Solo acoplados ao ambiente da planta doméstica

Portando do módulo Ethernet Shield W5100, criou-se uma página web com o objetivo de disponibilizar ao usuário a leitura de todos os dados e informações dos sensores, conforme exibido na Figura 21.

Essa página web é hospedada pelo próprio módulo Ethernet Shield W5100, no qual atua como um servidor TCP/IP, imprimindo a página web para o usuário durante a requisição de acesso HTTP por meio do código gravado no próprio Arduino MEGA. A página web está programada para atualizar de forma automática a cada 1 segundo.

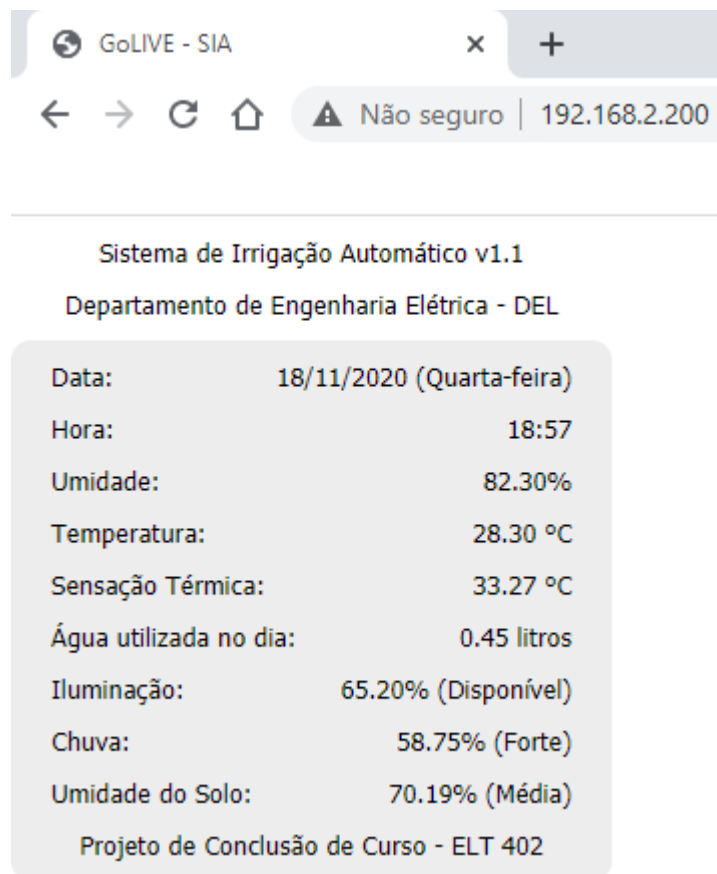


Figura 21 - Página WEB de leitura dos dados

O ciclo de irrigação está programado para ocorrer a cada dia, iniciando-se as oito horas da manhã (horário de Brasília), com despejo de 0,46 litros de água por ciclo. O ciclo é interrompido ou impedido de ser iniciado caso a umidade do solo esteja acima de 80% ou o indicador do sensor de chuva indique 90% em sua leitura.

Os resultados colhidos durante a verificação da quantidade de água despejada pelo protótipo estão exibidos na Tabela 1. Utilizando da equação (1) e considerando a densidade da água igual a 1g/cm^3 , obtém-se os valores de volumes d'água em ml.

Tabela 1 - Aferição do Sensor de Vazão YF-S201

Teste	Volume de Água medido (L)	Peso medido (g)	Volume de Água calculado (L)	Erro (%)
01	0,46	462	0,462	0,43
02	0,46	455	0,455	1,10
03	0,46	463	0,463	0,65

4 Conclusão

Este trabalho tratou da criação de um protótipo de irrigação automática para plantas domésticas, utilizando-se de uma plataforma de prototipagem de fácil uso e baixo custo, facilitando todo o planejamento, construção e execução do projeto. O protótipo é capaz de atuar de forma autônoma na irrigação de um sistema que dependa de recurso hídrico para seu crescimento e evolução, além de disponibilizar via plataforma WEB todas as informações de sensores no formato “tempo real”, facilitando o cotidiano do usuário no quesito noção situacional de seu plantio.

O Arduino MEGA 2560 provou ser um ótimo exemplar de microcontrolador, possuindo tensões de alimentação de sensores muito estáveis, ótima atuação em sensores digitais e analógicos, grande capacidade de memória para execução das etapas e cálculos impostos, além de dispor de diversas portas analógicas e digitais, possibilitando assim uma grande expansão do projeto com adição de novos sensores e atuadores, por exemplo.

Um fato observado com relação a definição de volume d'água a ser despejado é devido ao tempo de leitura do ciclo de processamento ocorreu a presença de um “gap” de 0,06 litros. Ou seja, para configurar um despejo de 0,46 litros, foi necessário programar um despejo de 0,4 litros.

O protótipo dispõe de possibilidades de melhorias futuras, sendo elas: integração de um módulo de internet Wireless; inserção de uma curva de irrigação personalizada para o tipo de plantio a ser trabalho; adição de mais válvulas solenoide, de forma a atuarem em diversos segmentos em um mesmo plantio, bem como a expansão citada anteriormente de diversos outros sensores e atuadores, devido à grande disponibilidade de portas analógicas e digitais pelo Arduino MEGA 2560.

5 *Referências Bibliográficas*

- ARDUINO. Arduino Uno Rev3. **Arduino**. Disponível em: <<https://store.arduino.cc/usa/arduino-uno-rev3>>. Acesso em: 17 nov. 2019.
- BAÚ DA ELETRÔNICA. Ethernet Shield W5100 para Arduino. **Baú da Eletrônica**. Disponível em: <<https://www.baudaeletronica.com.br/ethernet-shield-w5100-para-arduino.html>>. Acesso em: 17 nov. 2019.
- C. DORF, R.; H. BISHOP, R. **Sistemas de Controle Moderno**. 8a. ed. Rio de Janeiro: LTC, 2001.
- E. F. W. LIMA, J. **O uso da Irrigação no Brasil**. EMBRAPA. [S.l.], p. 16. (2).
- ELETROGATE. Válvula Solenóide de Entrada de Água - 180° 1/2" x 1/2" 12V DC. **Eletrogate**. Disponível em: <<https://www.eletrogate.com/valvula-solenoide-de-entrada-de-agua-180-1-2-12v-dc>>. Acesso em: 17 nov. 2019.
- FILIFELOP C.E. Display LCD Nokia 5110 Backlight Azul. Disponível em: <<https://www.filipeflop.com/produto/display-lcd-nokia-5110-backlight-azul/>>. Acesso em: 05 nov. 2020.
- FILIFELOP C.E. Fonte Ajustável Protoboard. Disponível em: <<https://www.filipeflop.com/produto/fonte-ajustavel-protoboard/>>. Acesso em: 05 nov. 2020.
- FILIFELOP C.E. Módulo Relé 5V 2 Canais. **Filipeflop**. Disponível em: <<https://www.filipeflop.com/produto/modulo-rele-5v-2-canais/>>. Acesso em: 17 nov. 2019.
- FILIFELOP C.E. Sensor de Chuva 9SS18. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-chuva/>>. Acesso em: 05 nov. 2020.
- FILIFELOP C.E. Sensor de Fluxo de Água 1/2" YF-S201b. **Filipeflop**. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-fluxo-de-agua-12-yf-s201/>>. Acesso em: 17 nov. 2019.
- FILIFELOP C.E. Sensor de Umidade do Solo Higrômetro. **Filipeflop**. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-umidade-do-solo-higrometro/>>. Acesso em: 17 nov. 2019.

FILIFELOP C.E. Sensor de Umidade e Temperatura AM2302 DHT22. **Filipeflop**. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-umidade-e-temperatura-am2302-dht22/>>. Acesso em: 17 nov. 2019.

GORNAT, B.; W. L. C., S. **Sistemas de Controle e Automatização da Irrigação**. Brasília: ABID, 1990.

KENSHIMA, G. Fritzing: Aprenda o que é e como usar de forma simples! Disponível em: <<https://www.filipeflop.com/blog/introducao-ao-fritzing/>>. Acesso em: 10 nov. 2020.

K-TECHNICS SOLUTIONS. K-Technics. **DS3231N AT24C32 IIC RTC Module Precision Clock Module for Arduino**. Disponível em:

<<https://ktechnics.com/shop/arduino-shields-modules/ds3231-at24c32-iic-precision-real-time-clock-module-memory-module-arduino-3/>>. Acesso em: 17 nov. 2019.

NUNES FILHO, J.; DE LIMA E SÁ, V. A. IRRIGAÇÃO E DRENAGEM - Instituto Agrônomo de Pernambuco - IPA. Disponível em: <<http://www.ipa.br/resp54.php>>. Acesso em: 11 dez. 2020.

SILVA SOUZA, R. Curiosidade: Saiba como os agricultores no Egito Antigo faziam para bombear a água para irrigação. Disponível em:

<https://www.irrigacao.net/outras_noticias/curiosidade-saiba-como-os-agricultores-no-egito-antigo-faziam-para-bombear-a-agua-para-irrigacao/>. Acesso em: 17 nov. 2019.

TESTEZLAF, R. **Irrigação: Métodos, Sistemas e Aplicações**. Campinas: FEAGRI, 2017.

VÍQUA. Conheça as vantagens da irrigação automatizada | Blog ViQUA. **Blog ViQUA**. Disponível em: <<https://www.viqua.com.br/blog/conheca-vantagens-da-irrigacao-automatizada>>. Acesso em: 17 nov. 2019.

Apêndice A

Código-fonte completo utilizado na programação do Arduino MEGA 2560.

```
/*
# Projeto de Trabalho de Conclusão de Curso
# Sistema Automatizado de Irrigação v1.1
# Autor: Ricardo Ferreira Pinheiro
# Matrícula: ES86306
# Departamento de Engenharia Elétrica - DEL
# Centro de Ciências Exatas e Tecnológicas - CCE
# Universidade Federal de Viçosa - UFV
*/

#include <SPI.h> // Biblioteca da interface SPI
#include <Ethernet.h> // Biblioteca do Ethernet
Shield W5100
#include <RTCLib.h> // Biblioteca do RealTime
Clock DS3231
#include <DHT.h> // Biblioteca do AM2302 DHT22

#include <Adafruit_GFX.h> // Biblioteca Gráfica
GFX
#include <Adafruit_PCD8544.h> // Bibliotoca do
controlador LCD PCD8544

#define DHTTYPE DHT22 // Tipo utilizado da
biblioteca DHT
//Pinos Digitais
#define dpinDHT 2 // Pino Digital utilizado pelo
DHT22 Sensor de Temperatura e Umidade
#define dpinLDR 50 // Pino Digital utilizado pelo
LM393 LDR Sensor de Luz
#define dpinRAIN 51 // Pino Digital utilizado pelo
LM393 Sensor de Chuva
#define dpinTERRAIN 52 // Pino Digital utilizado
pelo LM393 Sensor de Umidade Solo
#define dpinFLOW 24 // Pino Digital utilizado pelo
YF-S201 Sensor de Fluxo de Água
#define dpinWATERING1 22 // Pino Digital utilizado
pelo Módulo Relé para ativar a Válvula Solenoide.
//Pinos Analógicos
#define apinLDR A13 // Pino Analógico utilizado
pelo LM393 LDR Sensor de Luz
#define apinRAIN A14 // Pino Analógico utilizado
pelo LM393 Sensor de Chuva
#define apinTERRAIN A15 // Pino Analógico
utilizado pelo LM393 Sensor de Umidade Solo

//Array simbolo grau
byte grau[8]={ B00001100,
               B00010010,
               B00010010,
               B00001100,
               B00000000,
               B00000000,
               B00000000,
               B00000000,};

byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};

char daysOfTheWeek[7][16] = {"Domingo",
"Segunda-feira", "Ter&#231a-feira", "Quarta-
feira", "Quinta-feira", "Sexta-feira",
"S&#225bado"};

float valueLDR = 0;
float valueLDR100 = 0;
float valueRAIN = 0;
float valueRAIN100 = 0;
float valueTERRAIN = 0;
float valueTERRAIN100 = 0;
float valueFLOW = 0;
float valueDHT_HUMIDITY = 0;
float valueDHT_TEMPERATURE = 0;
float valueDHT_TEMPERATUREF = 0;
float valueDHT_HIC = 0;
float valueDHT_HIF = 0;
float valueRTC_TEMPERATURE = 0;
float countWATERDAILY = 0.4;

int countWateringDays = 1;
int flagWatering = 1;

String stateLDR;
String stateRAIN;
String stateTERRAIN;
String lastDAY;

RTC_DS3231 rtc;
IPAddress ip(192, 168, 2, 200);
EthernetServer server(80);
DHT dht(dpinDHT, DHTTYPE);
Adafruit_PCD8544 display = Adafruit_PCD8544(6,
5, 9, 3, 8);

volatile uint16_t pulses = 0;
volatile uint8_t lastflowpinstate;
volatile uint32_t lastflowratetimer = 0;
volatile float flowrate;
SIGNAL(TIMERO_COMPA_vect) {
  uint8_t x = digitalRead(dpinFLOW);

  if (x == lastflowpinstate) {
    lastflowratetimer++;
    return; // nothing changed!
  }

  if (x == HIGH) {
    //low to high transition!
    pulses++;
  }
}
```

```

lastflowpinstate = x;
flowrate = 1000.0;
flowrate /= lastflowratetimer; // in hertz
lastflowratetimer = 0;
}

void useInterrupt(boolean v) {
  if (v) {
    OCROA = 0xAF;
    TIMSKO |= _BV(OCIE0A);
  } else {
    TIMSKO &= ~_BV(OCIE0A);
  }
}

void setup() {
  pinMode(dpinWATERING1, OUTPUT);
  digitalWrite(dpinWATERING1, HIGH);
  pinMode(dpinFLOW, INPUT);
  digitalWrite(dpinFLOW, HIGH);
  Ethernet.init(10);
  dht.begin();
  display.begin();
  Serial.begin(115200);
  Ethernet.begin(mac,ip);

  while (!Serial) {
  }

  if (Ethernet.hardwareStatus() ==
EthernetNoHardware) {
    Serial.println("Ethernet Shield não foi
encontrada. Favor verificar!");
    while (true) {
      delay(1); // Faça nada, impossível seguir sem o
módulo Ethernet.
    }
  }
  if (Ethernet.linkStatus() == LinkOFF) {
    Serial.println("Cabo Ethernet está
desconectado!");
  }

  // start the server
  server.begin();
  Serial.print("Servidor locado no IP: ");
  Serial.println(Ethernet.localIP());

  if (! rtc.begin()) {
    Serial.println("Módulo RTC não foi
encontrado.");
    Serial.flush();
    abort();
  }

  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));

  if (rtc.lostPower()) {
    Serial.println("Módulo RTC ficou sem
alimentação, será realizado calibração.");

```

```

    rtc.adjust(DateTime(F(__DATE__),
F(__TIME__)));
  }
}

void loop() {
  delay(500);
  //----- BEGIN RTC Routine -----
  //-----//

  DateTime now = rtc.now();
  valueRTC_TEMPERATURE = rtc.getTemperature();

  // Serial.print(now.year(), DEC);
  // Serial.print('/');
  // Serial.print(now.month(), DEC);
  // Serial.print('/');
  // Serial.print(now.day(), DEC);
  // Serial.print(" ");
  //
  Serial.print(daysOfTheWeek[now.dayOfTheWeek()
]);
  // Serial.print(" ");
  // Serial.print(now.hour(), DEC);
  // Serial.print(':');
  // Serial.print(now.minute(), DEC);
  // Serial.print(':');
  // Serial.print(now.second(), DEC);
  // Serial.println();

  // Serial.print("Temperature: ");
  // Serial.print(valueRTC_TEMPERATURE);
  // Serial.println(" C");

  //----- END RTC Routine -----
  //-----//

  //----- BEGIN WATERING Routine
  //-----//

  if (lastDAY != String(now.day())) {
    flagWatering = 1;
    pulses = 0;
  }

  if (valueTERRAIN100 > 80 or valueRAIN100 > 90) {
    flagWatering = 0;
  }

  if (valueFLOW < countWATERDAILY and
flagWatering == 1) {
    if(now.hour() >= 8) {
      digitalWrite(dpinWATERING1, LOW);
    }
  }
  else if (valueFLOW >= countWATERDAILY or
flagWatering == 0) {
    digitalWrite(dpinWATERING1, HIGH);
  }
}

```



```

lastDAY = now.day();

//----- END WATERING Routine ---
-----//
//----- BEGIN DHT Routine -----
-----//

// Read humidity as %.
valueDHT_HUMIDITY = dht.readHumidity();
// Read temperature as Celsius (the default)
valueDHT_TEMPERATURE =
dht.readTemperature();
// Read temperature as Fahrenheit (isFahrenheit
= true)
// valueDHT_TEMPERATUREF =
dht.readTemperature(true);

// Check if any reads failed and exit early (to try
again).
if (isnan(valueDHT_HUMIDITY) ||
isnan(valueDHT_TEMPERATURE) ||
isnan(valueDHT_TEMPERATUREF)) {
Serial.println(F("Failed to read from DHT22
sensor!"));
return;
}

// Compute heat index in Fahrenheit (the
default)
// valueDHT_HIF =
dht.computeHeatIndex(valueDHT_TEMPERATUREF,
valueDHT_HUMIDITY);
// Compute heat index in Celsius (isFahreheit =
false)
valueDHT_HIC =
dht.computeHeatIndex(valueDHT_TEMPERATURE,
valueDHT_HUMIDITY, false);

// Serial.print(F("Humidity: "));
// Serial.print(valueDHT_HUMIDITY);
// Serial.print(F("% Temperature: "));
// Serial.print(valueDHT_TEMPERATURE);
// Serial.print(F("°C "));
// Serial.print(F("Heat index: "));
// Serial.print(valueDHT_HIC);
// Serial.println(F("°C "));

//----- END DHT Routine -----
-----//
//----- BEGIN FLOW Routine -----
-----//

lastflowpinstate = digitalRead(dpinFLOW);
useInterrupt(true);
// Serial.print("Freq: ");
// Serial.println(flowrate);
// Serial.print("Pulses: ");
// Serial.println(pulses, DEC);
// if a plastic sensor use the following calculation
valueFLOW = pulses / (7.5 * 60);

// Serial.print(valueFLOW);
// Serial.println(" Litros");

//----- END FLOW Routine -----
-----//
//----- BEGIN SENSORS Routine ---
-----//

valueLDR = analogRead(apinLDR);
valueLDR100 = (1 - valueLDR/1023)*100;
if (valueLDR100 > 50) {
stateLDR = "(Dispon&#237vel)";
}
else {
stateLDR = "(Ausente)";
}
// Serial.print("LDR = ");
// Serial.println(valueLDR);
valueRAIN = analogRead(apinRAIN);
valueRAIN100 = (1 - valueRAIN/1023)*100;
if (valueRAIN100 > 50) {
stateRAIN = "(Forte)";
}
else if (valueRAIN100 > 20) {
stateRAIN = "(Fraca)";
}
else {
stateRAIN = "(Ausente)";
}
// Serial.print("RAIN = ");
// Serial.println(valueRAIN);
valueTERRAIN = analogRead(apinTERRAIN);
valueTERRAIN100 = (1 -
valueTERRAIN/1023)*100;
if (valueTERRAIN100 > 75) {
stateTERRAIN = "(Alta)";
}
else if (valueRAIN100 > 50) {
stateTERRAIN = "(M&#233dia)";
}
else if (valueRAIN100 > 25) {
stateTERRAIN = "(Baixa)";
}
else {
stateTERRAIN = "(Ausente)";
}
// Serial.print("TERRAIN = ");
// Serial.println(valueTERRAIN);

//----- END SENSORS Routine ----
-----//
//----- BEGIN LCD Routine -----
-----//

display.clearDisplay();
display.setTextSize(1);
display.setContrast(15);
display.setTextColor(BLACK);
display.drawRect(0,0, 84,48, BLACK);
display.setCursor(5,2);

```

```

display.println("Hora:");
display.setCursor(40,2);
display.println(String(now.hour(), DEC) + ":" +
String(now.minute(), DEC));
display.setCursor(5,11);
display.println("Umid:");
display.setCursor(40,11);
display.println(String(valueDHT_HUMIDITY) +
"%");
display.setCursor(5,20);
display.println("Temp:");
display.setCursor(40,20);
display.println(String(valueDHT_TEMPERATURE) +
" C");
display.drawBitmap(68,20, grau, 8, 4, BLACK);
display.setCursor(5,29);
display.println("Agua:");
display.setCursor(40,29);
display.println(String(valueFLOW) + "L");
display.setCursor(5,38);
display.println("Solo:");
display.setCursor(40,38);
display.println(String(valueTERRAIN100) + "%");
display.display();

//----- END LCD Routine -----
-----//
//----- BEGIN ETHERNET Routine -
-----//

EthernetClient client = server.available();
if (client) {
  Serial.print("\n");
  Serial.println("Nova conexão:");
  boolean currentLineIsBlank = true;
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      Serial.write(c);
      if (c == '\n' && currentLineIsBlank) {
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println("Connection: close");
        client.println("Refresh: 5");
        client.println();
        client.println("<!DOCTYPE HTML>");
        client.println("<html>");
        client.println("<head><meta charset='utf-
8'>");
        client.println("<title>GoLIVE - SIA</title>");
        client.println("<link
href='https://rpinheiro14.000webhostapp.com/Sty
les.css' rel='stylesheet' type='text/css'></head>");
        client.println("<body><div id='title'>Sistema
de Irrigação Automático v1.1</div><div
id='main'>");
        // RTC //
        client.println("<li>Data:<right>"+
String(now.day(), DEC) + "/" + String(now.month(),
DEC) + "/" + String(now.year(), DEC) + (" (" +
String(daysOfTheWeek[now.dayOfTheWeek()]) +
(")</right></li>");
        client.println("<li>Hora:<right>"+
String(now.hour(), DEC) + ":" +
String(now.minute(), DEC)</right></li>");
        // DHT //
        client.println("<li>Umidade:<right>"+
String(valueDHT_HUMIDITY) + "%</right></li>");
        client.println("<li>Temperatura:<right>"+
String(valueDHT_TEMPERATURE) + "
&#176C</right></li>");
        client.println("<li>Sensa&#231&#227o
T&#233rmica:<right>"+ String(valueDHT_HIC) + "
&#176C</right></li>");
        // FLOW //
        client.println("<li>&#193gua utilizada no
dia:<right>"+ String(valueFLOW) + "
litros</right></li>");
        // SENSORS //
        client.println("<li>Ilumina&#231&#227o:<right>"+
String(valueLDR100) + "% " + stateLDR +
"</right></li>");
        client.println("<li>Chuva:<right>"+
String(valueRAIN100) + "% " + stateRAIN +
"</right></li>");
        client.println("<li>Umidade do Solo:<right>"+
String(valueTERRAIN100) + "% " + stateTERRAIN
+ "</right></li>");
        client.println("</body></html>");
        break;
      }
      if (c == '\n') {
        currentLineIsBlank = true;
      } else if (c != '\r') {
        currentLineIsBlank = false;
      }
    }
  }
  delay(1);
  client.stop();
  Serial.print("Conexão encerrada.");
  Serial.println("");
}
//----- END ETHERNET Routine ---
-----//
}

```

Apêndice B

Tabela com links para download das bibliotecas utilizadas.

Biblioteca	Link para Download
Módulo RTC DS3231	https://www.arduino-libraries.info/libraries/ds3231
Sensor de Fluxo de Água YF-S201b	https://github.com/adafruit/Adafruit-Flow-Meter
Sensor de Temperatura e Umidade DHT22	https://github.com/adafruit/DHT-sensor-library
Biblioteca Interface IO Adafruit	https://github.com/adafruit/Adafruit_BusIO
Biblioteca Gráfica GFX Adafruit	https://github.com/adafruit/Adafruit-GFX-Library
Biblioteca Nokia 5110 LCD	https://github.com/adafruit/Adafruit-PCD8544-Nokia-5110-LCD-library
Biblioteca Ethernet para Arduino	https://github.com/arduino-libraries/Ethernet