

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO**

**SISTEMA DE MEDIÇÃO DO CONSUMO DE ÁGUA
PARA CONSCIENTIZAÇÃO DURANTE O BANHO
USANDO MICROCONTROLADORES**

**RELATÓRIO DA DISCIPLINA DE MICROCONTROLADORES PARA
AUTOMAÇÃO**

Prof. Robinson Figueiredo de Camargo

Bruno Gabriel Flores Sampaio

**Santa Maria, RS, Brasil
2019**

RESUMO

Com as crescentes notícias a cerca do uso de forma inadequada da água, uma das preocupações de toda comunidade científica e industrial mais conscientes, está sobre como reduzir o consumo de água através da conscientização das pessoas e criação de aparelhos que consigam economizar os recursos hídricos. Como forma de conscientização a Organização das Nações Unidas (ONU) criou no dia 22 de março de 1992, o Dia Mundial da Água, divulgando a Declaração Universal dos Direitos da Água, entretanto, pouca coisa mudou de lá para cá, pois, as pessoas não tem consciência de quantos litros de água elas gastam por dia e de que forma elas gastam esses recursos.

Como forma de conscientização, nesse projeto será proposto a criação de um dispositivo que mede o volume de água gasto durante um banho e, que assim, a pessoa que esteja fazendo o uso do chuveiro, possa acompanhar esse consumo durante o processo. Acredita-se que com a visualização do volume total de água gasto durante um banho, o usuário tome consciência de quantos litros de água ele está desperdiçando, muitas vezes com coisas desnecessárias, e assim, o seu tempo de banho seja reduzido, economizando água consequentemente.

Sabe-se da importância da conscientização a respeito do uso da água e é necessário ter-se os órgãos de apoio ao nosso lado nessa luta, para que se consiga, com sabedoria, usar a água plenamente e assim deixar-se um bom legado para as futuras gerações. Com cada um fazendo a sua parte e economizando, na medida do possível, conseguiremos manter um mundo mais saudável e sem o medo de como será no futuro, tudo graças a tecnologia. A água é um bem de todos e deve ser respeitada.

SUMÁRIO

1.INTRODUÇÃO	7
2.OBJETIVO GERAL	8
3.OBJETIVO ESPECÍFICO	9
4.ESPECIFICAÇÕES.....	10
4.1Sensor de fluxo.....	10
4.2Microcontroladores Arduino	11
4.3Módulo Transmissor e Receptor RF 433MHz FS1000A / MX-RM-5V	12
4.4Display LCD 16x2	13
4.5Alimentação	13
5.DESENVOLVIMENTO	14
5.1HARDWARE.....	15
5.1.1Transmissor	15
5.1.2Receptor	19
5.1.3Estimativa de manutenção.....	22
5.1.4Conclusão de Hardware	23
5.2SOFTWARE	24
5.2.1Código do dispositivo Transmissor.....	25
5.2.2Código do dispositivo Receptor	28
5.2.3Conclusão	31
6.CONCLUSÃO	32
BIBLIOGRAFIAS.....	35

LISTA DE FIGURAS

Figura 1: Sensor de fluxo YF-S201	11
Figura 2: Arduino UNO a Direita e Arduino Nano a Esquerda	12
Figura 3: Módulo receptor RF a esquerda e módulo transmissor RF a direita.....	12
Figura 4: Display LCD 16x2	13
Figura 5: Arduino NANO e pinagens (Retirado de www.filipeflop.com)	16
Figura 6: Transmissor FS1000A visto de cima	17
Figura 7: Sensor de fluxo YF-S201	18
Figura 8: Fonte de alimentação 9V do transmissor	19
Figura 9: Módulo receptor MX_RM_05	20
Figura 10: Cores RGB (Red, Green, Blue).....	21
Figura 11: Anodo Comum e Catodo Comum.....	22
Figura 12: Mensagem do display	31

LISTA DE TABELAS

Tabela 1: Tabela de componentes do transmissor	15
Tabela 2: Tabela de componentes do receptor	19
Tabela 3: Pinos do LCD	21
Tabela 4: Tabela de alcance do transmissor/receptor em 1Hz	23
Tabela 5: Gastos anuais e economia de luz com o dispositivo operando – Parte 1	33
Tabela 6: Gastos anuais e economia de luz com o dispositivo operando - Parte 2	33

LISTA DE EQUAÇÕES

Equação 1: Taxa de fluxo YF-S201	10
Equação 2: Dimensionamento de antenas de RF.....	17

1. INTRODUÇÃO

Pesquisas acerca do uso indiscriminado da água no planeta Terra vem sendo feitas a muito tempo, e as estimativas são sempre preocupantes, a água está acabando. Segundo Claudio Angelo, no seu livro “O Aquecimento Global” (ANGELO, s.d), em até 2025, cerca de 2,7 bilhões de pessoas no mundo podem sofrer com a falta de água de alguma forma, principalmente nas regiões em que a escassez de água já é um problema hoje. O Brasil ainda não sente esse efeito em grande parte do país, principalmente por possuir grandes reservas de água distribuídas por grandes extensões do seu território, no entanto, isso nos traz grandes responsabilidades acerca do seu uso.

Um levantamento feito a respeito do consumo urbano de água no Brasil, relacionou aspectos demográficos e o consumo residencial e um dos aspectos descobertos foi que, apesar da diminuição do crescimento no número de pessoas, houve um crescente aumento no volume de água gasto pelas residências (CARMO, 2014). Um dos fatores para o aumento do volume de água gasto pelas famílias, certamente se dá ao fato de muitas tecnologias terem sido desenvolvidas para aumentar o bem-estar das pessoas. Com o comodismo criado por essas novas tecnologias, o desperdício dos recursos hidroelétricos aumentaram como consequência da facilidade de se usar tais dispositivos, pois agora temos o prazer de conseguir tomar banhos mais quentes nos dias mais frios, regar os quintais quase que religiosamente e manter as calçadas e carro sempre limpos, tudo a custo de “um pouquinho” mais de água.

Pensando no âmbito de consciência pessoal, é difícil se ter consciência de um problema, quando não sabemos o impacto das nossas ações para o agravamento dos mesmos. Muitas vezes em atividades que julgamos inocentes, estamos causando um desperdício que poderia ser contornado se tivéssemos noção dos nossos gastos e desperdícios. Tarefas de higienização pessoal ou de bens, como limpeza de casa, carros e pessoal, são essenciais e não podemos deixar de usufruir dos comodismos que temos, no entanto, precisamos sim saber dos nossos gastos e de como isso afeta nossa vida e a vida de todos no planeta Terra, direta e indiretamente.

Portanto o primeiro passo para isso, é criar um sistema de medição de vazão em sistemas hidráulicos que sejam de fácil implementação e que permitam a visualização em tempo real do consumo individual para os dispositivos de consumo hidroelétricos já existentes. Dentro do ambiente residencial podemos fazer a identificação de vários pontos onde esses dispositivos poderiam ser instalados, como por exemplo, em tubulações de mangueiras de jardim, caixa

d'água, torneiras e tanques, chuveiros e máquinas de lavar. Podemos regar o jardim todos os dias sem perceber dos impactos dessas ações, ou estendermos o banho por minutos a mais desnecessariamente.

2. OBJETIVO GERAL

Como forma de tentar conscientizar a população, uma das formas de se forçar esse pensamento, se dá com a visualização dos gastos que esse volume de recursos hidroelétricos traz, gerando um uso consciente. A fim de criar um dispositivo que seja capaz de fazer essa visualização, foi proposto a criação de um dispositivo que conseguisse fazer a leitura do volume de água gasto durante um banho, mostrando-o durante o processo, a fim de que assim, a pessoa que esteja usando-o, esteja ciente de quantos litros de água ela está gastando enquanto ensaboa o corpo ou lava os cabelos, por exemplo.

Uma das formas de se alcançar esse objetivo é usando algum dispositivo eletrônico ligado ao encanamento, que consiga fazer a medição do volume de água que está sendo entregue ao chuveiro e, dessa forma, calcular os gastos que irão ser gerados. Buscando um meio de conseguir fazer essa medição, nada mais prático do que fazer o uso de um microcontrolador comercial de código aberto, que possa ser replicado com certa facilidade e fazendo o uso de um sensor de vazão, seríamos capazes de prever com considerável precisão e então, através de um display, tornamos esses valores visíveis a quem estiver no chuveiro.

Esse mesmo sistema poderia ser expandido nos demais sistemas hidráulicos da casa, como mangueiras e máquinas de lavar, como citados anteriormente. Por isso a preocupação de criar um sistema descentralizado de sensoramento e que consiga interligar todos os sistemas em um único ponto, a fim de mostrar onde toda a água está sendo usado e em que quantidade. Esse seria um sistema ideal e que toda residência deveria possuir, visto os impactos positivos no consumo consciente que esses dispositivos trariam.

Portanto, para atingirmos tal objetivo, devemos criar um sistema de fácil aplicação, que possua um custo benefício considerável e bom, para não contradizermos o instinto humano de economizar ao se deparar com gastos e tratar nosso sistema como um “desperdício”, que não possua limitações físicas que impeçam nossa medição ou transmissão do dados captados e que sejam de fácil manutenção, além claro, de tornar-se um sistema atraente a quem o vê, a fim de que ele seja chamativo e bonito, para que consiga cumprir seu papel da melhor forma possível.

3. OBJETIVO ESPECÍFICO

Para atingir esse objetivo, será utilizado um sensor de fluxo, acoplado em série ao encanamento dos dispositivos ao qual faremos o sensoriamento, nosso dispositivo em questão será o chuveiro. O funcionamento do Sensor de Fluxo de água baseia-se em uma turbina estilo hélice que gira conforme o fluxo de água e detecta a quantidade de voltas dadas. A leitura dos dados faz-se através de um ímã posicionado em uma das pás da hélice e paralelamente a um sensor de efeito hall, responsável pelo envio de sinal digital. Dessa forma, usando um microcontrolador da família Arduino, seremos capazes de detectar qual o volume de água que está fluindo pelo encanamento e assim, fazer o monitoramento.

O microcontrolador utilizado será da família Arduino devido ao fácil acesso dos pinos de leitura digital e sinais de pulso modulados *pwm* (*pulse width modulation*). Além da medição do fluxo de água, o microcontrolador, também será utilizado para fazer a transmissão dos dados coletados com o sensor para outro microcontrolador da mesma família, a fim de conseguir fazer pontos independentes de monitoramento, com um único ponto de recepção para controle dos gastos. Dessa forma teremos o controle do sistema, dividido em dois pontos independentes, interligados de forma virtual e controlados pelos Arduinos de forma que a programação de cada um seja feita conforme as necessidades dele e assim, tornando o sistema modular, sendo bastante fácil de se fazer a expansão do sistema para mais dispositivos dispersos sob a casa.

O microcontrolador UNO assim como o Nano, possuem características muito similares, exceto pelas dimensões físicas. No entanto, as demais diferenças são irrelevantes para o projeto, uma vez que os microcontroladores são suficientes para o projeto, possuindo pinos digitais suficientes e entradas e saídas *pwm* que são os requisitos mais importantes. A programação de ambos pode ser feita na mesma *IDE*(*interface de desenvolvimento integrada*) e não possuem diferença de linguagem alguma.

Pensando na transmissão dos dados coletados pelo sensor de Fluxo, a primeira ideia que salta aos olhos, é fazer via transmissão em rádio frequência. Existem módulos desse tipo que podem ser usados com o microcontrolador da Arduino, em espectros de frequências legais na faixa de 433MHz. Tais módulos apesar de conseguirem, não precisam possuir alcance maior que alguns metros e podem usar antenas compactas para isso.

Para fazer a ponte entre o sistema e o usuário, faremos uso de uma *IHM*(*interface homem máquina*) simples, que pode ser um pequeno display sem muitas funções, além da função de

mostrar o registro do consumo de água e o desperdício gerado. Dessa forma, teremos um sistema completo e bastante simples, com possibilidade de expansão e que consegue obter todos os objetivos propostos.

4. ESPECIFICAÇÕES

Os sensores, módulos e microcontroladores utilizados para a execução desse projeto, foram escolhidos de acordo com a familiarização dos mesmos para o desenvolvimento do projeto, custo benefício e acessibilidade. Será dada uma breve explicação de como funciona cada um dos elementos base que serão utilizados no projeto, tendo como objetivo de familiarizar o leitor às formas de funcionamento dos dispositivos. Assim, quando for mostrada as formas de desenvolvimento do projeto, ficará mais fácil de entender o papel de cada peça e como cada um deles funciona.

4.1 Sensor de fluxo

O sensor de fluxo utilizado para o projeto é o sensor *YF-S201*. Esse sensor usa a força mecânica da água que flui do encanamento para dentro dele, girando alguns pares de hélices que se encontram dentro do sensor. Ao girar as hélices, o sensor cria um pulso magnético que pode ser captado através do efeito hall. Esses pulsos definem o fluxo da água que escoar sobre o sensor e possuir uma taxa definida pela Equação 1.

Equação 1: Taxa de fluxo YF-S201

$$\text{Pulso (Hz)} = [7.5 \times \text{Taxa do fluxo 'Q' (L / min)}]$$

Isso significa que a cada pulso dado por segundo, o fluxo de água é proporcional a 7.5 vezes o valor de 1 litro de água por minuto. De acordo com a folha de dados do sensor (*datasheet*), a taxa de erro que o sensor pode atingir é de 1 a 3% tanto para cima, quanto para baixo. A Figura 1 mostra as dimensões do sensor retiradas diretamente dessa folha de dados.



Figura 1: Sensor de fluxo YF-S201

4.2 Microcontroladores Arduino

Arduino é uma empresa de hardware livre que fabrica placas com um microcontrolador e um *IDE* com as que qualquer usuário pode criar projetos de todo tipo. A ideia é que os usuários, tanto os avançados como os novatos, possam montar uma placa próprias, criando um design único para suas necessidades e usá-la livremente sem ter que comprar uma placa pré-fabricada (GONZÁLEZ, 2016). Dessa forma, a empresa conta com milhares de desenvolvedores que não medem esforços para ajudar os mais novos no ramo, tornando assim a plataforma mais acessível e confiável.

Contando com isso, escolhemos um microcontrolador que fosse acessível e que ao mesmo tempo conseguisse atender as demandas do projeto. No site da Arduino conseguimos ter todas as informações a respeito dos produtos que a empresa oferece e suas especificações completas (ARDUINO).

Para o projeto, serão necessárias duas controladoras, uma para fazer a captação dos dados do fluxo do chuveiro, por meio do sensor citado anteriormente e outra para fazer o processamento desses dados. Pensando que a primeira controladora estará acoplada no encanamento e sujeita a condições adversas de temperatura e umidade, foi escolhido o Arduino Nano, devido as suas dimensões físicas reduzidas, enquanto para fazer o processamento dos dados, escolhemos o Arduino UNO, por ser mais versátil e compatível com as necessidades do

projeto, além de ser bastante manejável. A Figura 2 mostra a diferença de tamanho entre o Arduino UNO, a Direita e o Nano, a Esquerda.

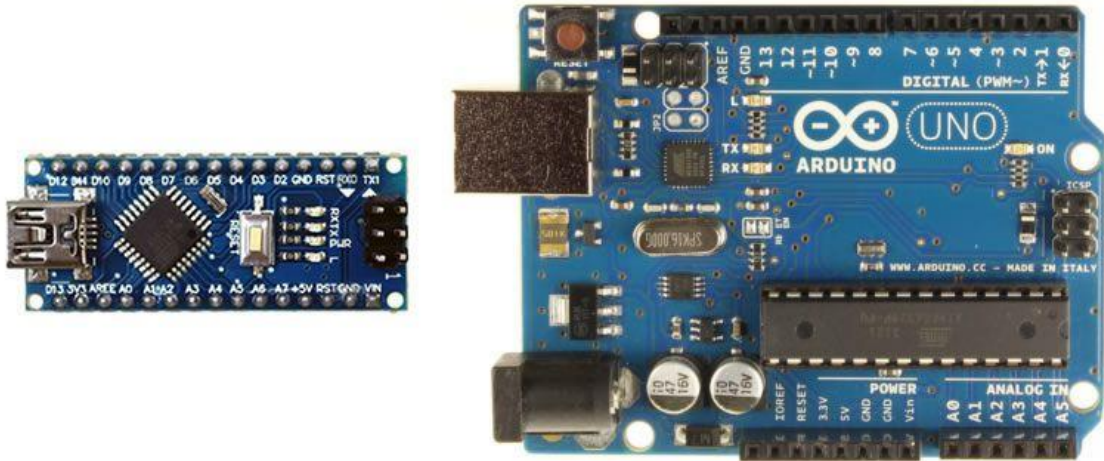


Figura 2: Arduino UNO a Direita e Arduino Nano a Esquerda

4.3 Módulo Transmissor e Receptor RF 433MHz FS1000A / MX-RM-5V

Para se fazer a transmissão dos dados, será utilizado um par de transmissores de rádio frequência, que operam no espectro de 433MHz. O transmissor é do modelo *FS1000A* e o receptor é do modelo *MX-RM-5V*. Esses módulos operam com modulação *ASK (Amplitude Shift Keying)*, ao qual através de uma onda portadora de alta frequência, transmitem dados de forma digital a uma boa distância em metros, motivo esse pelo interesse nos módulos. Na Figura 3 temos a foto de um par do módulo, estando o transmissor a direita e o receptor a esquerda.

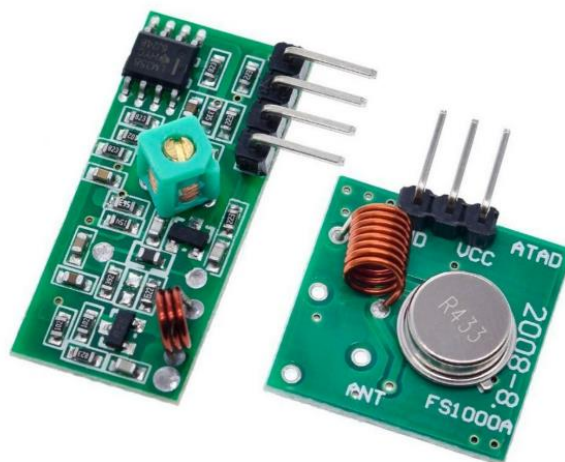


Figura 3: Módulo receptor RF a esquerda e módulo transmissor RF a direita

O sistema será do tipo *simplex*, ou seja, haverá comunicação somente em um sentido. O transmissor envia os dados para o receptor e não há inversão. Para garantir a segurança na entrega dos dados, haverá um *FCS (Frame Check Sequence)* para a detecção de erros. Dessa forma, se um dado esperado não for recebido ou então não chegar inteiro (Checagem de erro falhar), o sistema não irá contabilizar esses dados distorcidos ou inexistentes, garantindo que não seja mostrado ao usuário, informações erradas ou então que o sistema falhe por falta de coesão ou precisão.

4.4 Display LCD 16x2

A fim de acontecer uma relação entre o usuário e sistema, será incorporado uma *IHM* em forma de display. Esse display em questão é o *LCD (Display de Cristal Líquido)* que cria uma interface com o usuário, onde será impresso todas as informações que coletaremos e que julgamos necessárias. Esse display *LCD* tem 16 colunas e 2 linhas, com luz de fundo (*backlight*) verde e letras na cor preta. Para conexão, são 16 pinos, dos quais usamos 12 para uma conexão básica, já incluindo as conexões de alimentação, *backlight* e contraste (Figura 4).

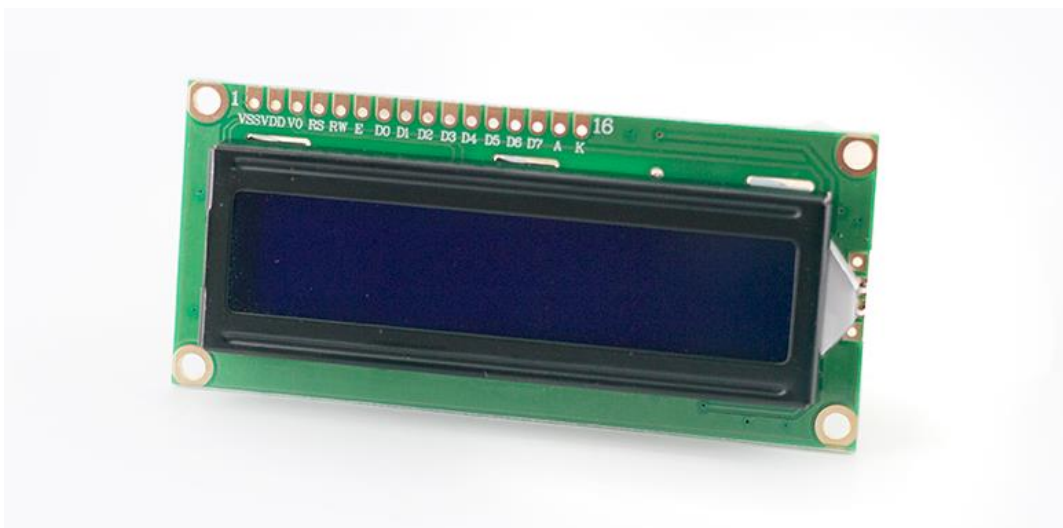


Figura 4: Display LCD 16x2

4.5 Alimentação

Para conseguirmos alimentar nosso sistema, será necessário se levar em consideração alguns aspectos, como o local onde será implementado o sistema, quais as condições desse ambiente e quais os recursos teremos disponível ali. Como o sistema, a primeiro momento, será

dividido em duas partes, teremos que levar em conta dois locais de instalação e, portanto, teremos duas situações.

A primeira e mais simples das duas, será o local onde estará a interface de comunicação com o usuário. Esta estará disposta na altura dos olhos, dentro do local de banho e por se tratar de um local bastante úmido e com riscos de choques, de acordo com a NBR5410, não é permitido tomadas ou qualquer outra fonte de energia a menos de 60 centímetros de boxes ou locais de banho (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2004, pag 170). Portanto, nosso sistema deverá utilizar uma bateria de 9V que consiga suprir a demanda do sistema.

Para o segundo caso, já podemos usar mais artifícios, uma vez que, apesar do local inapropriado, onde há um chuveiro, haverá também uma fonte de alimentação bastante forte, capaz de suprir nosso pequeno sistema. Nesse ponto, estaremos nos referindo ao sistema sendo ele composto pelo Arduino, o sensor de fluxo e o módulo transmissor, portanto não há a necessidade de uma fonte de energia grande ou então o risco de sobrecarga da rede, uma vez que o gasto energético do sistema é quase nulo, quando comparado ao da instalação de um chuveiro. Portanto, nesse caso, será utilizado uma fonte de alimentação própria que será conectada diretamente na rede, utilizando os próprios conectores do chuveiro ou então outro conector comercial do mesmo padrão.

5. DESENVOLVIMENTO

Para demonstração das etapas de desenvolvimento do sistema, foi necessário se separar em dois grandes tópicos, sendo o primeiro tópico, a parte de **Hardware**, onde será montado o sistema físico do projeto e através da folha de dados dos dispositivos, feito o dimensionamento do sistema para que consiga operar conforme o esperado e com a máxima redução de erros. Enquanto o segundo tópico fala a respeito da implementação de **software** do sistema, abordando pontos importantes para o funcionamento pleno do sistema lógico.

5.1 HARDWARE

A primeira coisa feita após a definição das dores que o sistema iria sanar e como ele iria fazer, foi a escolha das peças que comporiam o sistema como um todo. Cada sensor ou dispositivo usado, foi escolhido com alguma tarefa em particular que ele deveria solucionar e com isso, criar o sistema como uma peça única. Cada peça tem uma particularidade sua e funciona com algum protocolo de operação próprio, portanto, a familiarização com os componentes utilizados, foi um fator importante para a sua escolha, mais até que o seu preço ou desempenho. Dessa forma, outros dispositivos podem ser substituídos em réplicas ou adaptações desse projeto, desde que cumpram com as mesmas funções dos dispositivos originalmente escolhidos.

O sistema está plenamente modularizado, de tal forma que é possível se analisar componente por componente, individualmente, sem que se perca precisão da ideia.

5.1.1 Transmissor

Para o sistema de transmissão dos dados, estão inclusos os seguintes componentes:

Tabela 1: Tabela de componentes do transmissor

Nome do componente	Preço médio	Descrição	Modelo
Arduino NANO	R\$ 35,00	Microcontrolador	ATMega328p
Transmissor RF de 433MHz	R\$ 6,00	Transmite os dados coletados	FS1000A
Sensor de Fluxo 1/2"	R\$ 35,00	Mede o fluxo de água do encanamento	YF-S201
Fonte de Tensão 9V	R\$ 15,00	Alimentação do sistema	#
Total	R\$ 86,00	#	#

5.1.1.1 Arduino NANO

O Arduino NANO é o microcontrolador do sistema transmissor. Ele é o responsável por fazer toda a aquisição dos dados do sensor de fluxo, converter e envia-los ao receptor. O Arduino NANO opera em uma tensão de 5V, no entanto, é especificado que seja alimentado com pelo menos 9V para que opere com integridade.

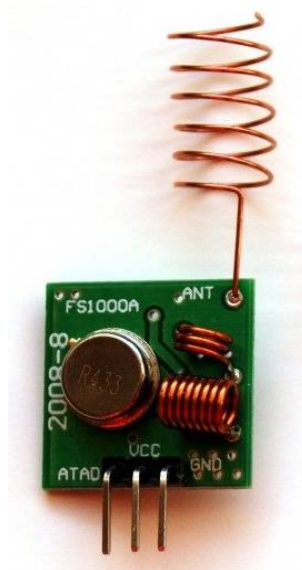


Figura 6: Transmissor FS1000A visto de cima

O módulo transmissor opera em um espectro eletromagnético de 433MHz, tornando-o um transmissor omnidirecional, ou seja, ele envia sinais eletromagnéticos em todas as 3 dimensões com o seu epicentro na antena. Em geral, esse tipo de transmissão costuma ser mais eficaz, ou seja, costuma a enviar sinais a distancias maiores, de acordo com a tensão aplicada nos pinos de alimentação *VCC*, podendo chegar até 200 metros em campo aberto quando acoplado uma antena eficaz.

Será utilizado no transmissor, uma antena dimensionada de 17.30cm utilizando um fio de cobre rígido 24awg. O dimensionamento da antena segue a Equação 2.

Equação 2: Dimensionamento de antenas de RF

$$\lambda \cdot f = c$$

A Equação 2 é conhecida mundialmente e imutável. Composta por λ sendo o comprimento da onda eletromagnética que queremos encontrar, f a frequência de operação de transmissão e c a velocidade da luz, uma constante que vele aproximadamente 300Kkm/s. Aplicando os valores de $f = 433.000.000 \text{ Hz}$ e $c = 299\,792\,458 \text{ m/s}$, obtemos o valor aproximado de 0,6923m. No entanto, o esperado de uma antena é que ela tenha pelo menos o comprimento de múltiplos de 1/4 de λ . Nesse caso $0,6923\text{m}/4 = 0,17309\text{m}$ ou 17,30cm. No entanto, visando melhor qualidade, iremos usar um múltiplo de 1/2 de λ enrolado em forma helicoidal de 5mm de diâmetro interno e mantendo o passo o mais uniforme possível.

5.1.1.3 Sensor de fluxo

O sensor de fluxo *YF-S201*, possui um diâmetro interno de $1/2''$ e pode ser acoplado diretamente no encanamento, com o uso de luvas. O sensor ao girar a hélice interna com o movimento mecânico água, irá gerar pulsos magnéticos que através do efeito hall, geraram pulsos que poderão ser captados em forma de interrupção no microcontrolador.

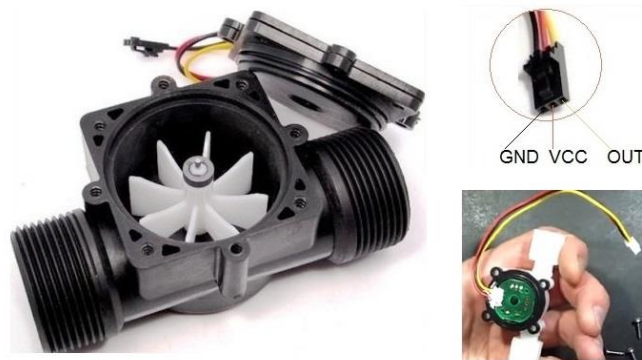


Figura 7: Sensor de fluxo YF-S201

Na figura 7 podemos ver a estrutura interna do sensor e as respectivas pinagens, sendo elas (1)*GND*, (2)*VCC* e (3)*OUT*, responsáveis respectivamente pela alimentação negativa, positiva e pela saída dos pulsos. Os pulsos são gerados pelo pequeno ímã no eixo de rotação da turbina.

5.1.1.4 Fonte de energia

A fonte utilizada, será uma fonte chaveada com entrada 110 ~ 220v e saída de 9V, com capacidade de até 1A (Figura 8). A fonte será ligada diretamente na rede elétrica, de tal forma que o sistema consiga sempre estar ligado, evitando assim o contato com o dispositivo, para que não torne o método de ligar e desligar o sistema, algo cansativo e desencoraje o uso.

A ponteira da fonte de alimentação é do tipo P4, no entanto, haverá um estágio de adaptação para que o conector encaixe no Arduino NANO, uma vez que ele deverá ser ligado diretamente nos pinos de alimentação *Vin* na própria placa (*onboard*) e ser devidamente isolado, evitando riscos de choque elétricos por ser exposto em ambientes muito úmidos.



Figura 8: Fonte de alimentação 9V do transmissor

5.1.2 Receptor

Para o sistema de recepção dos dados vindos do transmissor, foram utilizados:

Tabela 2: Tabela de componentes do receptor

Nome do componente	Preço médio	Descrição	Modelo
Arduino UNO	R\$ 50,00	Microcontrolador	ATMega328p
Receptor RF 433Mhz	R\$ 5,00	Receptor dos sinais RF	MX_RM_05
Display LCD 16x2	R\$ 20,00	Interface supervisória	
Led RGB	R\$ 1,00	Indicador luminoso	Anodo comum 5mm
Botões	R\$ 2,00	Configurações extra	#
Total	R\$ 78,00	#	#

5.1.2.1 Arduino UNO

Da mesma forma que o transmissor, o receptor usará um Arduino, porém dessa vez do modelo UNO, a pesar de ser o mesmo chip microcontrolador ATmega328P, a placa é um pouco diferente, como mostra a Figura 2.

O receptor usará um Arduino UNO para controlar os sinais recebidos pelo receptor de RF e processá-los, assegurando que os dados sejam recebidos de forma integra e assim, passando tais informações para a interface do Display e o led RGB. O Arduino será o cérebro por trás de todo processo de controle e atuará como um agente regulador, sendo o responsável pela alimentação e manipulação de todo o sistema.

5.1.2.2 Receptor RF 433MHz

O receptor RF do modelo *MX_RM_05* será o responsável pela recepção dos quadros recebidos do transmissor. Da mesma forma que o transmissor possui uma antena de 17,3cm para enviar os quadros, o receptor também irá possuir uma antena para receber as ondas eletromagnéticas que carregam toda informação útil, evitando que pedaços se percam no caminho ou que não consigam ser captados. O controle de integridade dos quadros recebidos, porém, não será processado pelo módulo receptor, mas sim pelo microcontrolador.

Na Figura 9 vemos o receptor de cima.

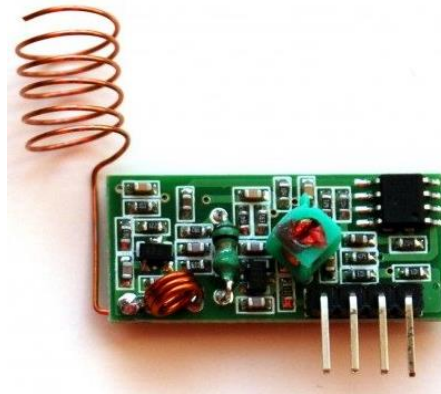


Figura 9: Módulo receptor MX_RM_05

A pesar de possuir 4 pinos, o receptor só utiliza 3 deles, uma vez que os pinos 2 e 3 estão interligados e representam os pinos de dados. Os pinos são (1)*GND*, (2,3)*Dados* e (4)*VCC* vistos da esquerda para a direita na Figura 9. Ao contrário do transmissor, o receptor só opera em faixas de tensão de 5V.

5.1.2.3 Display LCD 16x2

O Display de 16 colunas e 2 linhas, funciona seguindo os protocolos *SPI(serial peripheral interface)* através de uma série de registradores internos que fazem todo o controle do que será impresso e onde dentre as colunas e linhas será impresso, no display. Na Figura 4 vemos os pinos do display em um total de 16 pinos, no entanto, ao contrário do que pode se pensar, apenas 4 deles são pinos de dados. Na Tabela 3 podemos ver qual a pinagem, em ordem crescente, do Display e para que servem.

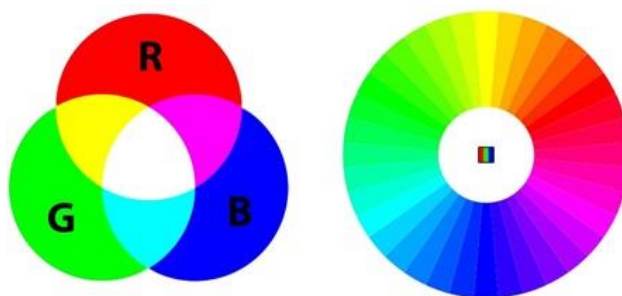
Tabela 3: Pinos do LCD

Pino LCD	Nome	Função
1	Vss	Alimentação positiva (5V)
2	Vdd	Alimentação negativa (ground)
3	V0	Ajuste de brilho do display
4	RS	Reset
5	R/W	Read / Write
6	EN	Enable
7 ~ 10	D0~D3	Pinos de dados para função Read (Não usados)
11~14	D4~D7	Pinos de dados para função Write (Usados)
15 e 16	Anodo e Catodo	Pinos de ajuste de LED background (Não usados)

Para o projeto, serão utilizados os pinos de dados *D4*, *D5*, *D6* e *D7*, para a função de *Write* no display, além dos pinos *RS* e *EN*, importantes para o controle de escrita na tela. Além dos pinos digitais, usaremos um pino analógico do microcontrolador para fazermos o ajuste de *V0*, sem o uso de um potenciômetro externo. *Vss* e *Vdd* serão conectados na alimentação do microcontrolador.

5.1.2.4 LED RGB e Botões

Será utilizado também um LED RGB como forma de aviso luminoso ao usuário do dispositivo. O LED RGB nada mais é que três LEDs em um só. Ele é formado por um LED vermelho (R de *Red*), um verde (G de *Green*) e um azul (B de *Blue*) Figura 10. Por ser um led multicolorido, podemos variar sua cor e brilho, a fim de gerar efeitos de urgência e para chamarmos atenção ao dispositivo.

**Figura 10: Cores RGB (Red, Green, Blue)**

O LED em questão é do tipo Catodo Comum, o que significa que ele possui seu pino de alimentação Negativo, comum aos demais LEDs. Portanto, para acendermos determinada cor, precisamos enviar um sinal com tensão positiva para os respectivos pinos. Na Figura 11 vemos os dois tipos de LEDs disponíveis, sendo eles do tipo Catodo comum e Anodo comum, um é o oposto do outro.

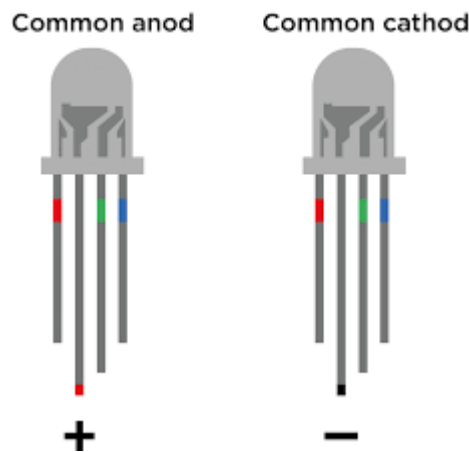


Figura 11: Anodo Comum e Catodo Comum

Para ligarmos os LEDs, cada cor possui uma tensão de operação, sendo elas 2V para a cor vermelha, 3.2 para as cores Azul e Verde. Para ligarmos esse LED RGB no microcontrolador, usaremos 4 pinos analógicos para que assim, consigamos entregar as tensões necessárias para cada um dos pinos sem precisarmos fazer o uso de resistores.

5.1.3 Estimativa de manutenção

Como o dispositivo eletrônico vai estar instalado em um ambiente bastante úmido por natureza, é importante que medidas de combata ao desgaste desses dispositivos e falhas contra eventuais queimas prematuras. Um sistema de isolamento é essencial para que a umidade não entre em contato com os circuitos eletrônicos, tal isolamento pode ser feito com a criação de uma caixa de proteção que junte as partes sensíveis e deixem expostas somente as partes que não podem ser guardadas. Havendo um sistema de isolamento eficiente, a vida útil do sistema se estenderá por mais tempo, caso contrário, falhas devido a oxidações ou curtos-circuitos podem acontecer, desgastando as peças e assim, havendo necessidade de troca de equipamentos.

5.1.4 Conclusão de Hardware

Como o maior problema do dispositivo está na comunicação RF entre transmissor e receptor, é fundamental que estes estejam alinhados para que evite que quadros sejam perdidos no ambiente e assim, o sistema perca qualidade. Para isso foi feito um teste como mostra a Tabela 4. Esse teste visou avaliar a qualidade da transmissão e a taxa de perdas no caminho, avaliando quantos quadros foram enviados e quantos recebidos, alterando-se a distância em linha reta entre eles e após, adicionando obstáculos no caminho.

Tabela 4: Tabela de alcance do transmissor/receptor em 1Hz

Distância	Obstáculos	Quadros enviados	Quadros recebidos	Taxa de sucesso
1m	Não	60	60	100%
5m	Não	60	60	100%
10m	Não	60	60	100%
15m	Não	60	59	98%
20m	Não	60	30	50%
1m	Sim	60	60	100%
5m	Sim	60	59	98%
10m	Sim	60	28	46%
15m	Sim	60	0	0%
20m	Sim	60	0	0%

Podemos perceber então, que o transmissor é eficaz para transmitir dados para distancias de até 15 metros em linha reta, sem obstáculos. No entanto, adicionando paredes e mais elementos domésticos, o seu alcance cai para até 5m com eficiência, mas consegue captar até 10m, recebendo quase a metade dos quadros que são enviados.

Diante disso, com uma proporção de sucesso de quase 100%, tratando de ambientes abertos em distância de 10m e com obstáculos a 5m, consideramos que mesmo a nível de protótipo, com os contatos elétricos longe do ideal, o sistema se mostrou muito eficiente, portanto, será capaz de realizar as tarefas para o qual foi proposto que realiza-se, com uma segurança.

5.2 SOFTWARE

Para a análise de *Software*, destinaremos um tempo especial para olharmos de perto, como cada um dos componentes do sistema Transmissor e Receptor irão trabalhar em conjunto, eletricamente, para que o sistema funcione plenamente. O cérebro do sistema serão os microcontroladores da família Arduino, que possuem a capacidade de serem programáveis e assim, adaptáveis, às diversas aplicações. O Arduino será tratado como unidade central e os sensores e atuadores, serão os periféricos do sistema.

Para a programação do sistema, a plataforma Arduino possui uma *IDE* próprio, um ambiente de desenvolvimento integrado criado para facilitar a programação das diversas plataformas de desenvolvimento, usando a linguagem de programação C/C++. Essa *IDE* pode ser instalada diretamente no seu computador pessoal ou então ser acessada diretamente no site da Arduino.cc de forma online (ARDUINO, 2020).

A programação Arduino é feita em alto nível se comparado à linguagem de máquina, a linguagem a qual o microprocessador irá interpretar, geralmente feita utilizando a linguagem de programação PASCAL. No entanto é vista como uma linguagem de baixo nível quando comparada as demais linguagens de programação que podem ser usadas, pois aproxima-se da máquina, podendo lidar diretamente com registradores e contadores do chip microprocessador.

No entanto, apesar do código estar escrito em linguagem C/C++, não há a necessidade de se entender profundamente a linguagem, pois ela será tratada de forma genérica e tratada como um bloco. Porém é necessário que se haja conhecimento básico de programação e dos conceitos fundamentais de qualquer linguagem.

Para a análise de *software*, analisaremos primeiro o código transmissor, em como ele fará a aquisição dos dados do sensor de fluxo, como eles serão processados e em seguida entregues ao transmissor RF para serem destinados ao receptor. Em seguida analisaremos o código do receptor, partindo da aquisição dos dados no receptor RF e processados pelo microcontrolador que irá então entrega-los à *IHM*.

5.2.1 Código do dispositivo Transmissor

5.2.1.1 Inicialização do código

Para iniciar o código, importamos a biblioteca *VirtualWire* na primeira linha e declaramos algumas variáveis importantes para a transmissão dos quadros nas linhas seguintes:

```
#include <VirtualWire.h>
uint8_t mensagem[4];
uint8_t tam_mensagem;
```

As variáveis *mensagem* e *tam_mensagem* são variáveis que irão ser as responsáveis por armazenar os bytes que irão ser transmitidos e o tamanho desse buffer.

Outras variáveis irão fazer o controle de vazão e contadores de todo o sistema que iremos transmitir na mensagem. Esses valores são declarados como ponteiros como vemos:

```
uint8_t *contaPulso = &mensagem[0];
uint8_t *Seg        = &mensagem[1];
uint8_t *Min        = &mensagem[2];
uint8_t *Hor        = &mensagem[3];
```

Essas variáveis são respectivamente, as variáveis de pulsos por segundo (*Hz*), vindas do sensor de vazão, segundos passados após a primeira contagem de pulsos, minutos e horas. É interessante essa declaração das variáveis como ponteiros, pois assim, podemos manipular elas e estaremos assim, escrevendo diretamente no *buffer* que será entregue na mensagem.

Outra declaração importante de ser feita é a que será utilizada nas interrupções que o sensor de fluxo irá causar no sistema, vemos ela em:

```
void incPulso(){
    contaPulso++;
}
```

Essa é uma declaração da função *incPulsos()*, a responsável por gerar um incremento positivo na variável *contaPulso*, a variável que é responsável por contar o volume da vazão de água no encanamento.

Essas declarações são feitas antes do código principal ser executado e é de importância para o controle do código.

5.2.1.2 Função Setup

Para a função de inicialização Setup, temos as seguintes linhas:

```
void setup() {  
    pinMode(Pino_transmissor, OUTPUT);  
    pinMode(Pino_sensorFluxo, INPUT);  
    attachInterrupt(0, incPulso, RISING);  
    vw_set_tx_pin(Pino_transmissor);  
    vw_setup(2000);  
}
```

As funções *pinMode()* são as responsáveis por definir os pinos de entrada e saída do Arduino. Vemos dois parâmetros sendo usados, *OUTPUT* e *INPUT*, sendo eles, parâmetros que definem se os periféricos são de saída ou entrada no Arduino. O pino que vai ao transmissor é um pino de saída de dados, sendo classificado como um periférico de saída, ao contrário, o pino que é conectado ao sensor de fluxo, é um periférico de entrada, pois ele é o responsável por dar entrada aos pulsos gerados pelo fluxo de água.

Para fazermos a medição desses pulsos gerados pelo sensor, usamos a função *attachInterrupt()*, recebendo os parâmetros 0 que defini o pino de interrupção sendo o pino digital 2, a função que será chamada ao receber a interrupção *incPulsos* e o parâmetro *RISING*. Esses parâmetros podem ser checados no site da Arduino.cc na parte de referências (ARDUINO REFERENCIAS, 2020).

As últimas duas linhas do código do setup são as responsáveis por iniciar a transmissão dos dados usando a biblioteca *VirtualWire*. Por vezes veremos funções que iniciam por VW, isso significa que são funções dessa biblioteca. A função *vw_set_tx_pin()* é a responsável por definir o pino que será utilizado o transmissor (*TX*) e a função *vw_setup()* define a velocidade em bit por segundo dessa transmissão. No nosso código está definido como 2000 bits por segundo.

A função setup, tem por definição, definir os parâmetros essenciais para o código rodar, sendo a primeira execução que o Arduino faz ao ser iniciado. A função setup só é executada uma única vez.

5.2.1.3 Função Loop

A função loop principal de todo o código, é nela onde o código será executado infinitamente, formando um ciclo de repetições que nunca se acaba. Na função loop algumas coisas interessantes irão acontecer indefinidamente, vamos ver as primeiras linhas:

```
contaPulso = 0;
delay (1000);
```

A primeira coisa que vemos é a variável *contaPulso* sendo zerada, isso garante que toda vez que uma nova contagem de pulsos seja iniciada, não existam valores sobrepostos. Em seguida temos os pulsos do sensor de fluxo sendo contados durante pelo menos 1 segundo. Dessa forma temos a frequência dos pulsos em Hertz, ou seja, em pulsos por segundo.

Após feita as contagens de pulsos, incrementamos o tempo. Sabendo que a contagem de pulsos leva 1 segundo para ser feita, incrementamos a variável que guarda os segundos *Seg*.

```
Seg++;
```

Dessa forma, podemos calcular o tempo corrido desde a primeira contagem de pulsos, até o final do banho, quando os pulsos zeram. A contagem do tempo se dá em Segundo, Minutos e Horas.

Outro ponto relevante ao processo é saber quando os pulsos cessarem, significando que o chuveiro está desligado, uma vez que não há fluxo de água sob o sensor. Para isso, criamos uma nova variável chamada *flagStopSend*:

```
if (!contaPulso)
    flagStopSend = true;
else
    flagStopSend = false;
```

Com isso, uma vez que o chuveiro esteja desligado, será utilizada essa flag para desligar a transmissão dos dados, enviando a contagem zero na primeira vez e após isso, só será enviado novos quadros assim que houver uma nova contagem no sensor.

Após contado os pulsos, basta enviar os quadros salvos no buffer com a variável *mensagem*, se a bandeira *flagStopSend* for falsa:

```
if(!flagStopSend){
    vw_send((uint8_t *)mensagem, tam_mensagem);
    vw_wait_tx();
}
```

A função *vw_send()* é a responsável por enviar a mensagem salva nas variáveis *mensagem*. A função leva dois argumentos, sendo o primeiro o buffer onde estará a mensagem

para ser enviado, ela deve ser do tipo `uint8_t`, por isso é importante garantirmos isso, fazendo um casting na variável (`uint8_t`). A segunda variável é que diz qual o tamanho do buffer em bytes.

Dessa forma podemos concluir que os dados salvos no buffer mensagem serão entregues sempre que houver uma nova contagem de pulsos com saldo positivo, permitindo que o sistema fique sempre ligado, sem a necessidade de implementarmos um sistema de *espera* (*StandBy*), que mantém o sistema adormecido até que algo o desperte.

5.2.2 Código do dispositivo Receptor

O código do receptor será um pouco mais complexo, uma vez que haverá necessidades adicionais para assegurar que os quadros recebidos estão íntegros, ou seja, não possuem dados perdidos no meio do caminho, além disso, haverá botões com funcionalidades específicas e o controle do display.

5.2.2.1 Inicialização

A primeira coisa a ser feita para a recepção dos quadros, é importar também a biblioteca *VirtualWire*, a mesma biblioteca utilizada para o envio dos quadros. Porém de forma contrária, agora definimos duas variáveis que serão os buffers de entrada de dados.

```
#include <VirtualWire.h>
#define MAX_BUFF 32
uint8_t buff_len = MAX_BUFF;
uint8_t buff[MAX_BUFF];
```

Como os quadros enviados são do tipo `uint8_t`, é sensato se pensar que o *buffer* deva ser do mesmo tipo, para fazer a recepção dos dados. Com isso temos o vetor variável *buff* do tamanho estipulado pela definição *MAX_BUFF* e também a temos a variável *buff_len* que representa o tamanho, bytes, do buffer.

Para o uso do display, temos que importar uma biblioteca especial também, uma vez que o LCD possui um mecanismo mais complexo para imprimir as informações na tela, é mais fácil de se fazer isso com a biblioteca *LiquidCrystal*:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(RS, EN, D4, D5, D6, D7);
```

Na segunda linha, iniciamos o objeto *lcd* do tipo *LiquidCrystal* e passamos os parâmetros *RS*, *EM*, *D4*, *D5*, *D6* e *D7*, que podem ser vistos na Tabela 3. Após criarmos esse

objeto `lcd`, podemos usar funções como `print()`, `setCursor()` e `clear()`, que serão abordadas mais à frente.

Novamente, após recebermos o quadro que será recebido pela variável `buff`, temos que quebra-la em partes que nos interessam, no mesmo formato que enviamos pelo transmissor, da mesma forma, declaramos:

```
uint8_t *pulsos = &buff[0];
uint8_t *Seg    = &buff[1];
uint8_t *Min    = &buff[2];
uint8_t *Hor    = &buff[3];
```

Novamente, a declaração das 4 variáveis é no formato de ponteiro, para o endereço de memória do buffer `buff`. Dessa forma, toda vez que o buffer receber um valor, as variáveis serão automaticamente carregadas com esse valor.

Para ligarmos o LED RGB, declaramos os pinos correspondente a cada uma das pernas do led, da seguinte forma:

```
const int VERM    = A1;
const int CONT    = A2;
const int VERD    = A3;
const int AZUL    = A4;
```

Sendo as variáveis `VERM` correspondente à cor vermelha no pino analógico A1, `CONT` o pino de controle do LED, correspondente ao pino de Catodo comum, no pino A2, `VERD` a cor verde, colocada no pino A3 e `AZUL` correspondendo à cor azul no pino A4.

5.2.2.2 Função Setup

A função `setup` irá inicializar as variáveis globais, definindo se os periféricos serão de entrada ou saída e também, iniciando alguns parâmetros importantes. A ideia do receptor segue a do transmissor, portanto só serão abordados pontos diferentes, como:

```
pinMode(CONT, OUTPUT);
analogWrite(CONT, LOW);
pinMode(A0, OUTPUT);
analogWrite(A0, 120);
lcd.begin(16, 2);
vw_rx_start();
```

Nas primeiras duas linhas, definimos o pino *CONT* como saída e em seguida definimos que a sua saída deve ser igual a 0. Sabendo que esse pino é o Catodo comum do LED, nós definimos que a sua saída deve ser igual a 0V, transformando-a em um pino de alimentação negativa.

Nas próximas duas linhas seguintes, fazemos o mesmo processo, no entanto, esses pinos são os pinos que usaríamos comumente para a instalação de um potenciômetro para controlarmos a luz de fundo do LCD, o chamado *BackLight*. Quando definimos esse pino como saída e passamos o parâmetro 120 no pino, estamos passando uma saída de 120 em uma escala de 1023 valores, que significam o nível de tensão no pino. Em regra de 3, sabendo que 1023 é igual a 5V e 0 igual a 0V, 120 equivale a tensão de 0.59V. Essa tensão seria obtida com a ajuda de um potenciômetro, no entanto, com o uso de uma saída analógica, conseguimos o mesmo resultado. O valor de 120 foi encontrado na tentativa e erro, até a tela ligar.

A próxima linha *lcd.begin(16,2)*, inicia as configurações do LCD e os argumentos 16 e 2 equivalem ao tamanho físico do LCD, sendo 16 colunas e 2 linhas.

A última linha, é a responsável por permitir que o receptor comece a escutar, ou seja, receber os dados vindos do transmissor. Caso queiramos desligar a escuta do receptor, bastaríamos usar um comando similar *vw_rx_stop()*.

5.2.2.3 Função Loop

Logo ao iniciar a função loop, nosso sistema irá esperar até que alguma transmissão seja feita, para isso, basta chamarmos a função *vw_wait_rx_max()*, mostrado logo a baixo:

```
vw_wait_rx_max(1000);
if (vw_have_message())
    if (vw_get_message(buff, &buff_len))
```

Essa função possui o argumento 1000 que significa o tempo máximo de espera para receber alguma transmissão. Essa função roda em loop até que receba uma mensagem, quando ela receber a mensagem, o loop é encerrado e então fazemos a verificação. *if(vw_have_message())*, que verifica a existência de uma mensagem.

Na linha seguinte, verificamos novamente se a mensagem foi recebida de forma íntegra. Nesse ponto do código, já sabemos que recebemos uma mensagem, mas não sabemos de quem foi e nem se ela chegou completa, portanto se o segundo *if* retornar falso (*false=0*), isso significa que a mensagem se perdeu no caminho, caso retorne verdadeiro (*true=1*), significa que a contagem de bites com o FCS está correto e a mensagem pode ser lida de forma confiável.

Como a comunicação é *simplex* não podemos mandar um *feedback* para o transmissor, pedindo que a mensagem seja reenviada, portanto o quadro errado, só será descartado e uma média dos resultados antigos será feita.

Quando o quadro recebido pelo receptor estiver correto, a mensagem será impressa na tela do lcd como mostra a Figura 12.

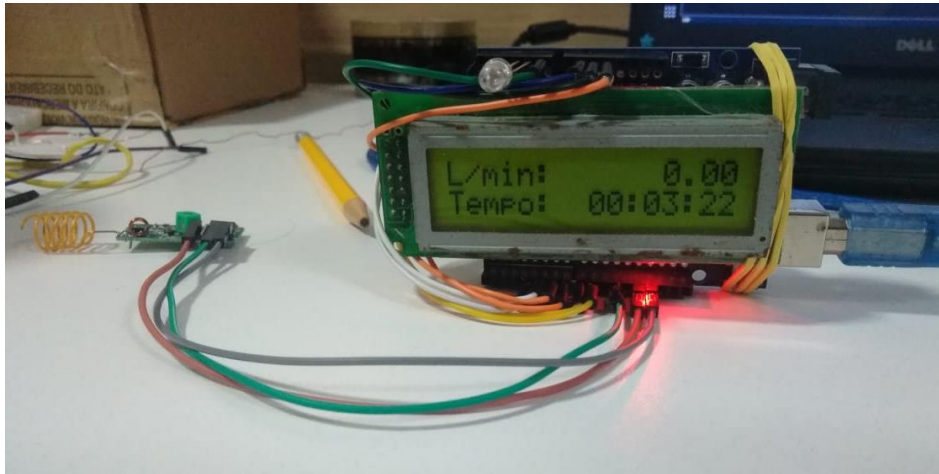


Figura 12: Mensagem do display

As demais funções do código possui o intuito de complementar o sistema como um todo, possuindo funções genéricas, como definir um valor em litros para ser consumido antes que o led ilumine sinalizando que o valor foi obtido ou está perto de extrapolar ou então, botões para desativar o sistema em casos de desconforto ou então, a fim de economizar, ainda mais, a bateria em casos de longas viagens, dessa forma, o sistema pode ser modificado a escolha do usuário. Dessa forma, não se faz necessário explicar as linhas de código usadas para as funções de botões ou leds.

5.2.3 Conclusão

Dessa forma, podemos perceber que diferente do imaginado à primeira vista, a coleta e transmissão de dados obtidos dos sensores pode ser feita de uma forma muito simples e sem a necessidade de se ter um conhecimento avançado em programação. Esse é um fator importante para a réplica do projeto para os interessados e entusiastas em domótica e IoT.

Vale salientar que o código fonte do projeto estará disponível de forma gratuita e aberta através da plataforma GitHub, no seguinte endereço Web:

<https://github.com/iOsnaaente/Medicao-de-consumo-de-agua>

6. CONCLUSÃO

O desenvolvimento do projeto está a nível de protótipo. A montagem dos dois dispositivos (Transmissor e Receptor) foram feitas a nível de bancada e estão totalmente expostos como mostram a Figura x e Figura y. Considerando o dispositivo como um produto final, ainda existem etapas adicionais para a implementação plena desse produto, como a criação de uma carcaça isolante e criação de uma placa de circuitos impressa com implementação de um microcontrolador *onboard* do tipo *standalone*.

O Arduino é um microcontrolador de código aberto (*opensource*), portanto, é comum a criação de placas pessoais e especializadas para serviços específicos. Para o dispositivo em questão, seria muito interessante a criação de uma placa própria, que cumpra as funções requisitadas e que tenham os módulos e dispositivos integrados em uma mesma placa. As vantagens de se fazer tal placa são várias, como, optimização de espaço físico, segurança de contatos elétricos, isolamento apropriado contra oxidações resultantes do ambiente úmido e facilidade de manutenção.

No entanto, se limitando a nível de protótipo, nosso projeto cumpriu bem os requisitos propostos por ele e mostrou bastante eficaz na aquisição dos dados de volume de água e comunicação a longas distâncias com boa eficiência e entrega íntegra de quadros, conseguindo comunicar o sistema em dois pontos distintos, tratando o sistema como um único sistema centralizado.

Além do mais, podemos ver que não houve um investimento astronômico para fazermos a aquisição de todos os componentes, se levando em conta os benefícios a longo prazo e pensando na economia de água e também elétrica. O gasto médio para se comprar e montar o circuito receptor foi de R\$ 91 e para o circuito transmissor foi de R\$ 78, dando ao total um gasto de R\$ 169. À primeira vista parece que há a necessidade de um investimento bastante gordo para algo que “não há necessidade”.

No entanto, quando analisamos a Tabela 5, usada para calcularmos o gasto de um chuveiro de potência média entre 5.5kWh e 7.5kWh e, presumindo que o tempo de banho de 3 casas com números de pessoas diferentes e tempos de banho diferentes, analisamos. Efeito da conta de luz em reais sem o uso do sistema e com o uso do sistema, presumindo que ele reduza o tempo de banho em 33% do tempo sem o sistema. O sistema se paga em questão de meses,

considerando a tarifa de R\$ 0.56 a cada kWh que foi cobrada no mês de junho de 2020 na região centro-oeste do rio grande do sul - Brasil. (ANEEL, 2020).

Tabela 5: Gastos anuais e economia de luz com o dispositivo operando – Parte 1

Casa	Potencia kWh	Nº pessoas	Tempo banho	Tempo total da casa	Tempo com dispositivo	Custo kWh	Gasto kWk antes	Gasto kWk depois
Casa1	6.5	4	15	60	20	R\$0,56	6.50	2.15
Casa2		3	10	30	10		3.25	1.07
Casa3		2	10	20	6.5		2.17	0.72
Total		9		110	36.5		11.91	3.93

Tabela 6: Gastos anuais e economia de luz com o dispositivo operando - Parte 2

Diferença em kW	Valor Diferença em um dia	Em uma semana	Em um mês	Em um ano	Gasto no ano sem o sistema
4.35	R\$ 2,44	R\$ 17,07	R\$ 73,16	R\$ 890,15	R\$ 2.145,00
2.18	R\$ 1,22	R\$ 8,54	R\$ 36,60	R\$ 445,10	R\$ 1.072,50
1.45	R\$ 0,81	R\$ 5,70	R\$ 24,40	R\$ 296,70	R\$ 715,00
7.98	R\$ 4,47	R\$ 31,30	R\$ 134,10	R\$ 1.621,96	R\$ 3.932,50

Nas condições da hipótese levantada, teríamos uma economia de luz de aproximadamente R\$ 1.620 em um ano, somando as três casas, com diferentes números de pessoas e diferentes tempos de banho e potencias de chuveiro. Com essa economia, seria suficiente para criarmos mais 10 sistemas como esse e assim sucessivamente, gerando uma grande economia na conta de luz, ajudando no desperdício de água e ajudando o meio ambiente de uma forma voluntária.

Além da economia em energia elétrica, fator pelo qual o sistema funcionará, haverá a economia direta do volume em litros de água que são desperdiçados diariamente de forma inconsciente. Na mesma proporção que economizaríamos 30% na conta de luz no final do dia, teremos a mesma proporção na economia de água, sendo assim, em um banho de 10 minutos, que são gastos em média 30 litros de água, teríamos uma economia de 10 litros por banho no dia. Usando o total de 9 pessoas da Tabela 5 e 6, podemos assumir que haveria uma economia de 90 litros de água por dia, 630 litros por semana e 32.850 litros por ano. Economia essa que

se dará de forma espontânea, sem o pesar ou a dor de ter que reduzir seu tempo, prazeroso, de banho, já que o bolso pesa mais que a consciente em muitos casos.

Tendo atingido os objetivos visados no momento da criação do projeto, finalizo demonstrando extrema alegria com os resultados adquiridos e pensando no efeito positivo que poderia alcançar com difusão desse método de economia voluntário. O meio ambiente é um espaço dividido com todos nós e cuidar dele é um dever que temos com as gerações que virão depois de nós.

BIBLIOGRAFIAS

- [1] ANGELO, Claudio; et al. **Governo do estado do Paraná: Uma previsão catastrófica marca o colapso da água no mundo para o ano 2025**, s.d. Disponível em: < http://www.diaadiaeducacao.pr.gov.br/portals/roteiropedagogico/publicacao/2519_A_era_da_falta_de_agua.pdf >. Acesso em: 15 de maio de 2020.
- [2] CARMO, Roberto Luiz do; DAGNINO, Ricardo de Sampaio; JOHANSEN, Igor Cavallini. **Transição demográfica e transição do consumo urbano de água no Brasil**. Rev. bras. estud. popul., São Paulo , v. 31, n. 1, p. 169-190, Jun 2014.
- [3] GONZÁLEZ, Gabriela. Blogthinkbig.com. **Por que Arduino é útil e o que é possível criar com ele?**. 5 de maio de 2016; Disponível em < <http://br.blogthinkbig.com/> >. Acesso em 22 de maio de 2020.
- [4] ARDUINO. **Página inicial**. Disponível em: < <https://www.arduino.cc/> >. Acesso em: 16 de maio de 2020.
- [5] ARDUINO REFERENCIAS. **Referencias**. Disponível em: < <https://www.arduino.cc/reference/pt/> >. Acesso em: 06 de jun de 2020.
- [6] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 5410:2004 - Instalações elétricas de baixa tensão**. Rio de Janeiro, 2004, p. 170.
- [7] ANEEL, **Ranking de tarifas por região**. Disponível em: <<https://www.aneel.gov.br/ranking-das-tarifas>>. Acesso em 08 de jun de 2020.