

Anemômetro

Os ventos do Jardim

Gabriel Santos Silva Araújo
Universidade de Brasília
Faculdade Gama
Gama, Distrito Federal
gabrielsantos_s96@hotmail.com

Yasmine Silveira Andrade
Universidade de Brasília
Faculdade Gama
Gama, Distrito Federal
andradeyasmine96@gmail.com

Resumo — O presente documento consiste na descrição do processo de confecção de um anemômetro que beneficiará o Jardim Botânico de Brasília informando a velocidade e direção do vento para auxiliar em combates a incêndios.

Keywords—*anemômetro; vento; Jardim Botânico de Brasília; direção; velocidade do vento*

I. INTRODUÇÃO

O bioma Cerrado constitui 23,92% do território brasileiro, sendo o segundo maior bioma nacional com uma área de aproximadamente 2.036.448 Km² [1]. Conhecido como a savana mais rica do mundo em biodiversidade, ele possui diversas espécies e é berço de várias nascentes, como dos principais rios das bacias Amazônica, da Prata e do São Francisco.

Na importante tarefa de conhecer e proteger este bioma, o Jardim Botânico de Brasília – JBB, conhecido como “Jardim do Cerrado”, é uma área de proteção ao cerrado vinculada à Secretaria de Meio Ambiente do Distrito Federal – SEMA/DF, que possui um grande acervo de espécies e promove pesquisas e educação ambiental nesta área [2]. Porém, um dos grandes problemas enfrentados pelo JBB são as queimadas que ameaçam a fauna e flora ali presente.

As queimadas são comuns no Cerrado devido a abundância de matéria orgânica em seu solo – como folhas secas e restos de vegetação – e a baixa umidade no período de seca, mas, com diversas ações do homem, essas queimadas se intensificaram, sendo prejudicial à diversidade deste bioma.

Um dos fatores que influenciam a propagação do fogo é a velocidade e a direção do vento. Para obtenção destes dados são utilizados anemômetros.

Existem diversos tipos de anemômetros, porém, vários deles apresentam limitações que restringem seu uso, dependente das condições climáticas de umidade e temperatura, necessidade de manutenção devido ao

desgaste das partes mecânicas e faixa de velocidade do vento encontrada no local a ser aferido.

Um dos mais utilizados é o anemômetro de pás, constituído por três pás mecânicas que giram com as correntes de vento. A partir da quantidade de giros por tempo e do o perímetro em que as pás se movimentam, é possível aferir a velocidade em que o vento se encontra.

II. OBJETIVO

Este projeto tem como objetivo, auxiliar o jardim botânico de Brasília em um problema recorrente no cerrado brasileiro: a contenção de incêndios através do uso de informações a respeito do vento, utilizando um anemômetro de baixo custo e eficiência.

III. BENEFÍCIOS E REQUISITOS

O anemômetro foi projetado de modo a atender as necessidades do Jardim Botânico de Brasília, agindo para facilitar o controle de incêndios, pois é de extrema importância o conhecimento da velocidade e direção do vento para que se possa calcular a velocidade de propagação do fogo. Essas informações serão utilizadas para facilitar o trabalho dos bombeiros e ajudar a preservar a fauna e flora do JBB.

Os dispositivos que permitem mensurar esses dados vendidos comercialmente, além de possuir um preço elevado, necessitam de um custo maior com energia e manutenção. Pensando nisso, este projeto conta com sensores e controladores de baixo custo, baixo consumo energético e fácil manutenção.

Dentre os requisitos necessários, o anemômetro projetado deverá medir velocidades entre 5Km/h e 30Km/h, baixo consumo energético e resistente às condições climáticas no local.

IV. CONSTRUÇÃO DO ANEMÔMETRO

Durante a realização do proposto, buscou-se soluções viáveis que resultassem em dados confiáveis de modo a atender os requisitos.

Em resumo, o funcionamento é dado por três pás mecânicas, ligadas a um disco, que giram com a força do vento. O disco contém um ímã que aciona um sensor sempre que passa por ele. Sabendo a circunferência do disco, é possível aferir a velocidade do vento pelo o número de voltas por um determinado tempo, descontando-se a resistência mecânica do equipamento. Já a direção do vento é dada por uma chapa presa à ponta de uma haste de modo de, quando o vento bate, rotaciona a chapa, deixando-a na mesma direção e sentido dele, que é detectada por outro sensor.

Para melhor compreensão, a construção do anemômetro foi subdividida em três partes: estrutura mecânica, hardware e software.

A. Construção Mecânica

A estrutura do protótipo foi montada de acordo com a fig.1 presente no final deste tópico. Ela conta com uma parte móvel formada por três pás, em formato de meia esfera ocas, presas a uma peça circular plástica por raios de bicicleta de aço inox. A peça circular foi encaixada na parte superior de um cano pvc, já na parte inferior, foi colocado um cd, com um ímã em sua extremidade, seguido de um rolamento para que o suporte não impeça o giro da estrutura.

Um tubo de caneta é encaixado no centro do rolamento e fixado à base do anemômetro, deixando assim, a parte móvel livre para girar.

A base é formada por uma caixa retangular de madeira com um furo circular no centro, para a passagem de uma haste de metal, e outro retângular à 6 cm do primeiro para passagem do módulo sensor magnético. A haste possui uma pá em forma de disco em sua ponta superior e é presa ao encoder dentro da caixa para que, quando o vento mover a pá para sua direção, o encoder é rotacionado.

A caixa possui também um furo em sua lateral para a passagem do cabo de alimentação do microcontrolador contido na mesma.

Os materiais utilizados no processo de montagem, excluindo-se sensores e o microcontrolador, são descritos na tabela I.

6	CD	Acrílico	1	Ø8cm
7	Rolamento	Aço	1	Ø23mm
8	Ímã	Neodímio	1	10 x 12mm
9	Tubo de caneta	Acrílico	1	25 x Ø7mm
10	Haste metálica	Aço Inox	1	22 x Ø4mm
11	Caixa	Madeira	1	12 x 22 mm

a.

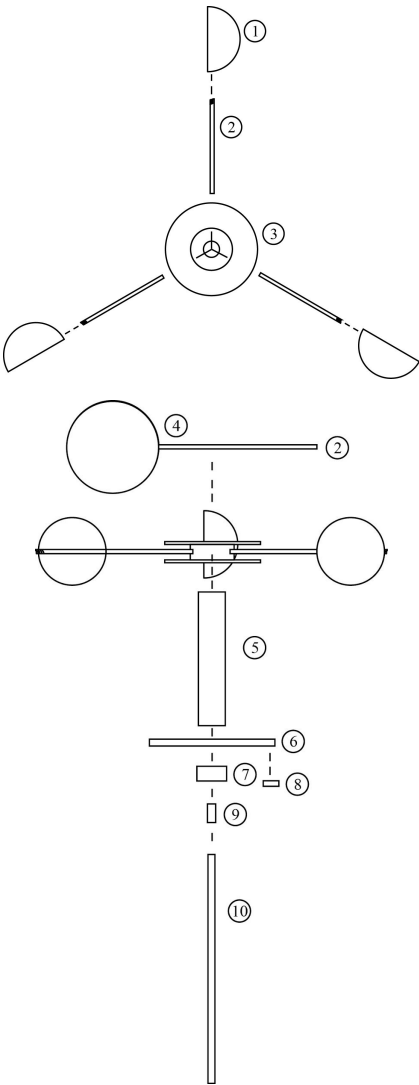


TABELA I. LISTA DE MATERIAIS UTILIZADOS PARA MONTAGEM MECÂNICA

Nº	Bill Of Materials			
	Descrição	Material	Quantidade	Dimensões
1	Pá mecânica	Alumínio	3	Ø45 mm
2	Raio bicicleta	Aço inox	4	180 mm
3	Carretel de fio de nylon	Plástico	1	Ø6 cm
4	CD	Acrílico	1	Ø12 cm
5	Cano PVC	PVC	1	18 x Ø25mm

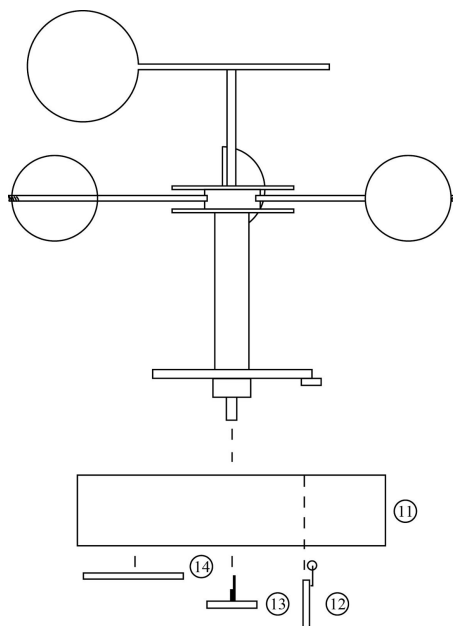


Fig. 1. Diagrama de montagem do anemômetro

Finalizada essa descrição da parte mecânica, é possível detalhar sensores e o microcontrolador necessários para o equipamento.

B. Descrição de Hardware

Primeiramente escolheu-se o hardware que seria utilizado.

Para a medição da velocidade do vento foi utilizado um módulo sensor magnético, mostrado na fig. 2, que contém um Reed-Switch, com contato normal aberto, e é acionado por indução magnética.



Fig. 2. Módulo sensor magnético.

Para aferir a direção do vento, foi utilizado um encoder rotativo, observado na fig 3.



Fig. 3. Módulo encoder 360° P17.

Este sensor gera dois sinais digitais com defasagem de 90°, os sinais estão fixos em nível lógico alto e geram pulsos sempre que o pino do encoder é rotacionado. A partir da combinação dos dois sinais é possível identificar qual o sentido de giro, como mostrado na fig.4.

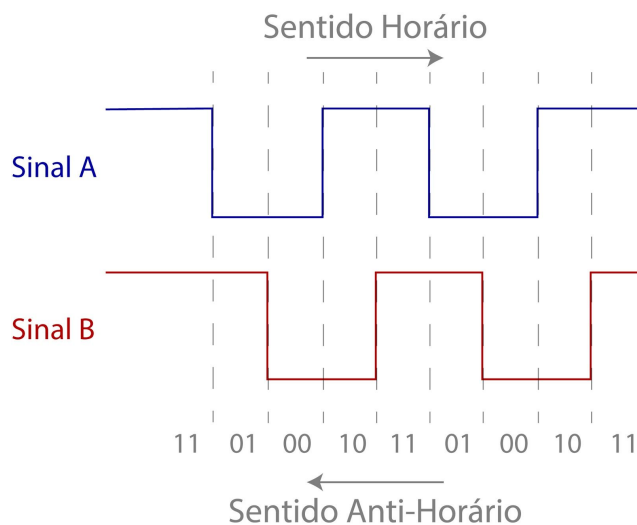


Fig. 4. Sinal de onda do encoder rotativo.

Para fazer a programação e a interação entre os sensores, foi o microcontrolador responsável é o MSP430G2553.

A seguir o diagrama blocos da fig. 4 de demonstra a disposição destes componentes e, abaixo dela, há a lista dos hardwares citados.

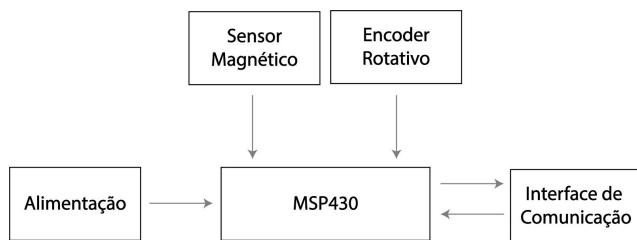


Fig. 4 Diagrama de blocos da disposição do hardware.

TABELA 2. LISTA DE MATERIAIS UTILIZADOS

Nº	Bill Of Materials			
	Material	Quantidade	Custo Unitário (R\$)	Fornecedor
13	Módulo Encoder 360° P17	1	25.90	GBK Robotics
12	Módulo Sensor Magnético	1	13.90	H.U infinito
14	MSP430G2553	1	30.00	Texas Intruments
-	Jumpers	6	0.25	Baú da Eletrônica

Por fim, foi redigido o código utilizado.

C. Descrição do software

Para realização do projeto foi necessário utilizar o software Code Composer, que permite a confecção de códigos em linguagem C e ASSEMBLY. Com essa ferramenta em mãos, foi elaborado o código para controlar o sensor magnético e indicar a velocidade em seu terminal.

No início do código para medição da velocidade, a função e Init_UART() é chamadas para fazer as especificações de comunicação serial necessárias para ler os dados do sensor.

Após a chamada da função de configuração da uart e das declarações iniciais, o temporizador com um clock de 2 Hz entra em loop infinito, como mostrado no fragmento de código 1, sendo utilizado para indicar o tempo em que a contagem de voltas é realizada.

```

cnt = 4;
TA0CTL = TASSEL_2 + ID_3 + MC_1;
TACCR0 = 62500-1; // clock de 2 hz;
while(cnt--)
{
    while((TACTL & TAIFG)==0);
    TACTL &=~TAIFG;
}

```

Fragmento de Código 1.

Em seguida, quando a contagem está sendo realizada, há uma verificação de voltas dadas pelo disco do anemômetro, sendo que a cada volta completa a variável “voltas” é incrementada.

O código abaixo mostra também a lógica de detecção do sinal analógico recebido, tratado como entrada.

A borda de descida é detectada caso o valor da variável “sensor” mude de 20, quando está sendo acionado, para 10, quando deixa de ser acionado, sendo o último valor atribuído à variável “new_sensor”. Assim, se new_sensor = 20 e sensor = 10 houve uma volta a mais.

Quando a contagem chega ao fim, uma interrupção é chamada.

```

while(1)
{
    cnt = 32;
    while(cnt--)
    {
        TA0CTL = TASSEL_2 + ID_3 + MC_1;
        TACCR0 = 62500-1; // clock de 2 hz;

        while((TACTL & TAIFG)==0)
        {
            if ((P1IN & AD_IN)==0) {sensor = 20;}
            else if ((P1IN & AD_IN)!=0) {sensor = 10;}
            else if ((new_sensor = 20) && (sensor = 10)) {voltas++;}
            new_sensor = sensor;
        }
        TACTL &=~TAIFG;
    }
    P1OUT ^=inter;
}

```

Fragmento de Código 2.

Na interrupção é feito o cálculo da distância percorrida e da velocidade. Em seguida, a função “posição()” é chamada para identificar qual a direção e sentido do vento, como mostrado no fragmento de código 3.

```

#pragma vector = PORT1_VECTOR
__interrupt void port_1(void){
    dist = (voltas * 0.36758*3.6)/2*0.867;
    P1OUT ^=LED;
    voltas = 0;
    posição();
    P1IFG = 0x00;
}

```

Fragmento de Código 3.

A distância percorrida é calculada a partir do número de voltas do disco e do valor de sua circunferência sempre que contagem é reiniciada, ou seja, cnt = 1. Sabendo o diâmetro do disco (12cm), sua circunferência é dada por:

$$\begin{aligned} \text{Circunferência} &= 2.\pi.0.06 \\ \text{Circunferência} &= 0.3632743 \end{aligned}$$

A partir desses dados, obtém-se a distância percorrida em 2 segundos, ou seja, a velocidade em m/s. Para observar os dados com mais facilidade, a velocidade foi multiplicada por 3.6 pra converter de m/s para Km/h.

A função “posição();” consiste em determinar a direção do vento de acordo com dois sinais digitais fornecidos pelo módulo sensor encoder rotativo, como mostrado na descrição do hardware.

Com base nessa lógica, e sabendo que o *range* do sensor para dar uma volta completa é 24 contagens, a posição do encoder é calculada a partir do seguinte fragmento de código 4.

```
void posicao();
{

P1DIR &= ~(reA + reB);
P1IFG &= ~(reA );

P2IE |= reA ;
P2IES &= ~reA;

}
```

Fragmento de Código 4.

De acordo com a quantidade de deslocamentos realizados para direita ou esquerda é possível obter a direção e sentido do vento.

```
void direcao (void)
{
    int n = pos%24;
    if ((n >= 0 && n < 6) | (n >= -24 && n < -18))
        dir = 'N';
    else if ((n >= 6 && n < 12) | (n >= -18 && n < -12))
        dir = 'E';
    else if ((n >= 12 && n < 18) | (n >= -12 && n < -6))
        dir = 'S';
    else if ((n >= 18 && n < 24) | (n >= -6 && n < 0))
        dir='O';
}
```

Fragmento de Código 5.

Com a velocidade e a posição calculada, a função printval(); é chamada para mostrar no terminal do code composer os valores de velocidade medidos. Então é zerado o valor da quantidade de voltas e a flag da interrupção é zerada, para retornar o código de onde foi chamada e então recomeçar a contagem.

V. RESULTADOS

Com o fim das três etapas mencionadas (fig. 5, 6 e 7), foi possível realizar testes para constatar a funcionalidade do protótipo construído.



Fig. 5. Ímã e sensor magnético instalados.



Fig. 6. Vista frontal do anemômetro construído.



Fig. 7. Vista superior do anemômetro construído.

Foi possível verificar então a partir dos testes realizados que o protótipo é capaz de calcular a direção e sentido do vento de modo adequado a aplicação desejada, incluindo os sentidos, norte (N), leste (E), sul (S) e oeste (O). No entanto houve um problema com o encoder incremental e

o novo sensor adquirido não chegou na data esperada, impossibilitando a demonstração prática.

Já a velocidade a partir da correção de erros anteriormente mencionados apresentou funcionalidade comprovada onde foi possível visualizar seu funcionamento a partir da impressão dos dados de velocidade e número de voltas obtidos a partir de testes experimentais com o protótipo criado.

Os resultados apresentados sob a influência de uma velocidade de vento previamente estabelecida estão demonstrados na tabela III.

TABELA III. RESULTADO DAS MEDIÇÕES

Medição	Velocidade aprox. 10 Km/h	Velocidade aprox. 13.4 Km/h
1	11	11
2	11	12
3	10	13
4	9	13
5	10	14

VI. CONCLUSÃO

Assim, a partir dos testes realizados, é possível observar que para a utilização no jardim botânico o protótipo precisa de alguns ajustes e de um instrumento de medição para realizar a calibração, fazendo com que os dados apresentados sejam mais confiáveis.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] IBGE. Mapa de Biomas e de Vegetação. Disponível em: <https://ww2.ibge.gov.br/home/presidencia/noticias/21052004bioma.shtml>. Acesso em: 26 de março de 2018.
- [2] JBB. Quem Somos. Disponível em: <http://www.jardimbotanico.df.gov.br/institucional/quem-somos/>. Acessado em: 26 de março de 2018.
- [3] A.S.Dionisio, "ANEMOMETRO UTRASSÔNICO BASEADO EM SENSOR DE DISTÂNCIA". UFRN, Setembro 2014.
- [4] M. R. S. Ribeiro¹, R. Bergamasco¹, M. G. Ribeiro¹, S. V. Aredes², L. F. W. Barbosa³, "DESENVOLVIMENTO DE UM ANEMÔMETRO DIGITAL ULTRA-SÔNICO". UNIVAP/FEAU.

VII. ANEXO

```
#include <msp430g2553.h>
#include <legacymsp430.h>

#define inter BIT5
#define reA BIT0
#define reB BIT7
#define AD_IN BIT3
#define AD_INCH INCH_0
#define RX BIT1
#define TX BIT2
#define LED BIT6

char dir;
int pos;
int new_sensor;
int dist;
int cnt;
int voltas;
int sensor;

void direcao(void);
void posicao();
void printval();
void Send_Data(volatile unsigned char c);
void Send_Int(int n);
void Send_String(char str[]);
void Init_UART();

int main(void)
{
    WDTCTL = WDTPW + WDTHOLD;
    BCSC1 = CALBC1_1MHZ;
    DCOCTL = CALDCO_1MHZ;
    P1DIR &= ~(reA + reB);
    P1IFG &= ~(reA);
    P2IE |= reA;
    P2IES &= ~reA;
    P1DIR &= ~AD_IN;
    P1REN |= AD_IN;
    P1OUT |= AD_IN;
    P1DIR &= ~inter;
    P1REN |= inter;
    P1IE = inter;
    P1IES = inter;
    P1IFG = 0x00;
    P1OUT &= ~LED;
    P1DIR |= LED;

    Init_UART();
```

```
_enable_interrupts();

while(1)
{
    cnt = 32;
    while(cnt--)
    {
        TA0CTL = TASSEL_2 + ID_3 + MC_1;
        TACCR0 = 62500-1; // clock de 2 hz;

        while((TACTL & TAIFG)==0)
        {
            if ((P1IN & AD_IN)==0) {sensor = 20;}
            else if ((P1IN & AD_IN)!=0){sensor = 10;}
            else if ((new_sensor = 20) && (sensor = 10))
            {voltas++;}
            new_sensor = sensor;
        }
        TACTL &= ~TAIFG;
    }
    P1OUT ^= inter;
}

#pragma vector = PORT1_VECTOR
__interrupt void port_1(void){
    dist = (voltas * 0.36758*3.6)/2;
    P1OUT ^= LED;
    printval(dist);
    voltas = 0;
    P1IFG = 0x00;
}

void printval()
{
    Send_String("velocidade = ");
    Send_Int(dist);
    Send_String("\r\n");
    Send_String("direção = ");
    Send_Int(dir);
    Send_String("\r\n");
}

void Send_Data(volatile unsigned char c)
{
    while((IFG2&UCA0TXIFG)==0);
    UCA0TXBUF = c;
}

void Send_Int(int n)
{
    int casa, dig;
    if(n==0)
    {
        Send_Data('0');
        return;
    }
```

```

}
if(n<0)
{
    Send_Data('-');
    n = -n;
}
for(casa = 1; casa<=n; casa *= 10);
casa /= 10;
while(casa>0)
{
    dig = (n/casa);
    Send_Data(dig+'0');
    n -= dig*casa;
    casa /= 10;
}
}

```

```

void Send_String(char str[])
{
    int i;
    for(i=0; str[i]!='\0'; i++)
        Send_Data(str[i]);
}

```

```

void Init_UART()
{
    // Habilita os pinos para transmissao serial UART
    P1SEL2 = P1SEL = RX+TX;
    // Configura a transmissao serial UART com 8 bits de
    dados,
    // sem paridade, comecando pelo bit menos
    significativo,
    // e com um bit de STOP
    UCA0CTL0 = 0;
    // Escolhe o SMCLK como clock para a UART
    UCA0CTL1 = UCSSEL_2;
    // Define a baud rate
    UCA0BR0 = 104;
    UCA0BR1 = 0;
    UCA0MCTL = UCBRF_0+UCBRS_1;
}

```

```

void posicao()
{
    if (reA != reB)

    { pos++; }

    else { pos--; }

    P1IFG = 0x00;

    direcao();
}

```

```

}

void direcao (void)
{
    int n = pos%24;
    if ((n >= 0 && n < 6 ) | ( n >= - 24 && n < -18 )) { dir
    = 'N';}
    else if (( n >= 6 && n < 12 ) | (n >= -18 && n < -12))
    {dir = 'E';}
    else if (( n >= 12 && n < 18) | (n >= -12 && n < -6))    {
    dir = 'S';}
    else if (( n >= 18 && n < 24) | (n >= -6 && n < 0))    {
    dir = 'O';}
}

```