

Sistema de Quantificação de Dosagem Medicamentosa

Victor Santos Cruz 10/48473
Universidade de Brasília-UnB
vycthur@hotmail.com

6 de maio de 2018

Resumo

Esse trabalho tem como objetivo principal; projetar um sistema que quantifique e alerte a dosagem de medicamentos. O objetivo é mensurar a quantidade medicamentosa que é inalada por um paciente e com isso alertar a dosagem e, caso seja, bloquear a aplicação do medicamento. O problema alvo está nas doenças respiratórias; Asma, Bronquiolite, Bronquite, entre outras. Para isso será usado como base de estudo a substância Bromidrato de fenoterol, princípio ativo do medicamento Berotec.

Palavras-chaves: Medicamento.Quantificação.MSP430.

1 Justificativa

A automedicação é um problema mais comum do que se imagina. No Brasil, o problema causa cerca de 20 mil mortes por ano, segundo a Associação Brasileira das Indústrias Farmacêuticas (Abifarma). A automedicação pode causar também entre outros fatores a dependência química.

Nesse contexto, temos também a questão da superdosagem que é quando o paciente extrapola na quantidade do medicamento ministrado no intuito de solucionar rapidamente o problema da enfermidade. Infelizmente, também é comum relatos de superdosagem por profissionais da saúde. Há muitos casos em que crianças e recém-nascidos são vítimas desses tipos de erros.

Visando a questão da segurança na posologia, e em tempos de doenças que acometem o aparelho respiratório, principalmente em crianças, tais como; Bronquite, Bronquiolite, Asmas entre outras enfermidades. O projeto aborda essa questão da rigorosidade da apli-

cação dos medicamentos e segurança na hora da aplicação.

2 Objetivo

O intuito é, experimentalmente, quantificar a dosagem de medicamento inalado por cada indivíduo seguindo sua faixa etária. Para tanto, será analisado a dosologia com a substância química Bromidrato de Fenoterol, princípio ativo do medicamento Berotec. Berotec é um Broncodilatador das vias respiratórias indicado para o tratamento dos sintomas da Asma e de outras doenças que provocam o estreitamento das vias respiratórias. Ele age de forma a dilatar os músculos da região afetada pela doença.

3 Requisitos

Para conseguir fazer a medição da quantidade de medicamento que está sendo injetado pelo frasco/bombinha será feito uma análise da quantidade do princípio ativo que

consta em cada 1ml de medicamento. Assim pode ser feita uma comparação da quantidade do princípio ativo, que é recomendada e prescrita pelo médico, à quantidade que realmente está sendo inalado pelo paciente. Para essa finalidade, será projetado um circuito que leia a quantidade de medicamento e retorne uma ação. Essa ação terá a princípio dois (2) sinais:

- * Alerta que a quantidade de medicamento recomendada foi atingida;
- * Sinal broqueando a frasco/ bombinha, caso o paciente insista em aplicar outra dose.

Portanto, esse circuito constará de um sensor que fará a leitura da quantidade de medicamento e um microcontrolador para o processamento dessa informação. Outros componentes serão acrescentados posteriormente para construção da bancada de teste.

3.1 Sensor

Para essa medição será utilizado o sensor de umidade Dht11 (Sensor de Temperatura e Umidade). O Dht11 é um sensor com saída digital. A leitura do sensor é possível por meio da comunicação com um microcontrolador, sendo o sinal de leitura serial de uma via.

Parâmetros de funcionamento do sensor:

- * A faixa de medição de umidade está entre 20% a 90% RH. :
- * A tensão de alimentação 5V:
- * Corrente de alimentação 200 a 500 mA:

3.2 Microcontrolador

O microcontrolador utilizado no projeto será o MSP430 da TEXAS INSTRUMENTS. É um microcontrolador de baixo consumo de energia, possui alto desempenho e baixa tensão de operação. Nesse microcontrolador será desenvolvido o software que

controlará o circuito do projeto. Ele fará a aquisição e tratamento do sinal fornecido pelo sensor, assim como também o processamento dos cálculos de medidas e informações de aviso ao usuário.

4 Benefícios

O produto fim do projeto pode ser uma alternativa de controle de medicações. Indicado principalmente a pessoas leigas que necessitam usar medicações que podem ocasionar um risco iminente a saúde, caso não seja usado a dosagem correta. O produto também pode servir de utensílio para o tratamento de pessoas que adquirem algum vício em decorrência do prolongamento de medicações adversas ou medicações de uso contínuo.

5 Processo de Fabricação

5.1 Algoritmo

Para o funcionamento do circuito formado pelo Microcontrolador e o Sensor de umidade é preciso de um algoritmo que controle o sistema. Nesse algoritmo está previsto a aquisição dos sinais vindos do sensor e os cálculos para tomada de decisão. Nessa parte do projeto será usado o compilador de código do ENERGIA.



```
#include <msp430.h>

// Definição de variáveis com funções booleanas

#define TST(x,y) (x & (y)) // TST recebe x AND y
#define SET(x,y) (x |= (y)) // SET recebe x que recebe x OR y
#define CLR(x,y) (x &= ~(y)) // CLR recebe x que recebe x AND complemento y
#define TOG(x,y) (x ^= (y)) // TOG recebe x que recebe x XOR y

// Definição do pino de dados do sensor de temperatura DHT11

#define DPIN BIT4

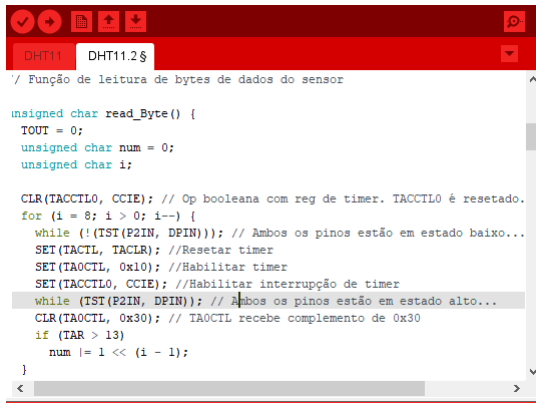
extern unsigned char volatile TOUT; // Variável da função de leitura

// Definição de funções usadas na leitura dos dados do sensor

void start_Signal(void);
```

Figura 1 – Declaração de variáveis

Na figura figura2 está a função onde será feito a leitura do sensor.



```

// Função de leitura de bytes de dados do sensor

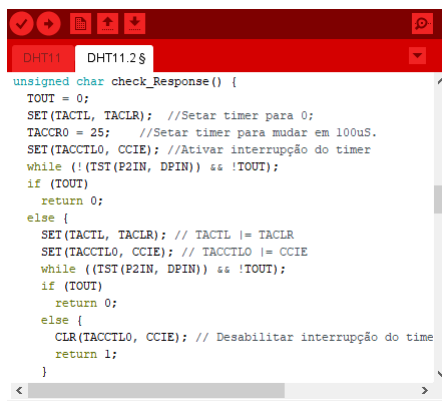
unsigned char read_byte() {
    TOUT = 0;
    unsigned char num = 0;
    unsigned char i;

    CLR(TACCTL0, CCIE); // Op booleana com reg de timer. TACCTL0 é resetado.
    for (i = 8; i > 0; i--) {
        while (!(TST(P2IN, DPIN))); // Ambos os pinos estão em estado baixo...
        SET(TACTL, TACLRL); //Resetar timer
        SET(TA0CTL, 0x10); //Habilitar timer
        SET(TACCTL0, CCIE); //Habilitar interrupção de timer
        while (TST(P2IN, DPIN)); // Ambos os pinos estão em estado alto...
        CLR(TA0CTL, 0x30); // TA0CTL recebe complemento de 0x30
        if (TAR > 13)
            num |= 1 << (i - 1);
    }
}

```

Figura 2 – Leitura do Sensor

A figura figura3 representa a função que verifica a resposta do sensor e ativa e desativa a interrupção do Timer. Essa função só retorna verdadeiro se a interrupção é devidamente ativada e desativada.



```

unsigned char check_Response() {
    TOUT = 0;
    SET(TACTL, TACLRL); //Setar timer para 0;
    TACCR0 = 25; //Setar timer para mudar em 100uS.
    SET(TACCTL0, CCIE); //Ativar interrupção do timer
    while (!(TST(P2IN, DPIN))) && !TOUT;
    if (TOUT)
        return 0;
    else {
        SET(TACTL, TACLRL); // TACTL != TACLRL
        SET(TACCTL0, CCIE); // TACCTL0 != CCIE
        while ((TST(P2IN, DPIN)) && !TOUT);
        if (TOUT)
            return 0;
        else {
            CLR(TACCTL0, CCIE); // Desabilitar interrupção do time
            return 1;
        }
    }
}

```

Figura 3 – checar resposta



```

// Função para iniciar a leitura de sinal

void start_Signal() {
    SET(P2DIR, DPIN); // Setar pino de Dados como saída
    CLR(P2OUT, DPIN); // Setar estado falso
    __delay_cycles(25000); // Delay de quase 18ms
    SET(P2OUT, DPIN);
    __delay_cycles(30); // Deixar em true por 30us
    CLR(P2DIR, DPIN); // Setar pino de Dados como entrada
}

```

Figura 4 – Inicializar leitura de sinal



```

// Definição de interrupção

#pragma vector = TIMER0_A0_VECTOR
__interrupt void CCR0_ISR(void) {
    SECOND_TIMER++;
    TOUT=1;
    TOG (P1OUT, 0x01);
    CLR (TACCTL0, CCIFG);
}

int main(void) {
    init(); //Inicialização

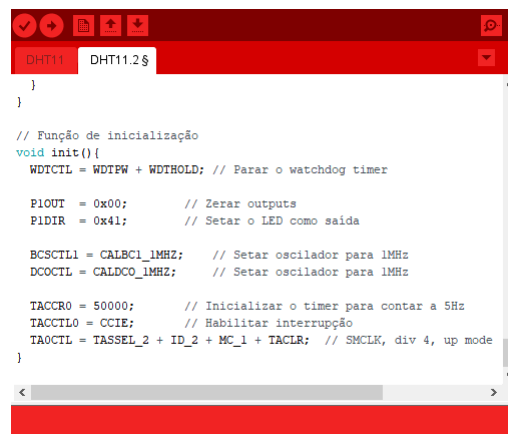
    while(1){

        if(SECOND_TIMER >= 5){ //5 @ CCR0 = 50000 & div 4

            // Adquirindo os dados e armazenando

```

Figura 5 – Definição de interrupção



```

// Função de inicialização
void init(){
    WDCTL = WDTPW + WDTHOLD; // Parar o watchdog timer

    P1OUT = 0x00; // Zerar outputs
    P1DIR = 0x41; // Setar o LED como saída

    BCSCTL1 = CALBC1_1MHZ; // Setar oscilador para 1MHz
    DCOCTL = CALDCO_1MHZ; // Setar oscilador para 1MHz

    TACCR0 = 50000; // Inicializar o timer para contar a 5Hz
    TACCTL0 = CCIE; // Habilitar interrupção
    TA0CTL = TASSEL_2 + ID_2 + MC_1 + TACLRL; // SMCLK, div 4, up mode
}

```

Figura 6 – Função de Inicialização

6 Bibliografía

- <https://noticias.r7.com/saude/automedicacao-e-uso-incorreto-de-remedios-podem-levar-a-morte-30032015>
- <https://noticias.r7.com/distrito-federal/medica-que-receitou-superdose-de-adrenalina-a-crianca-que-morreu-sera-intimada-pelo-ministerio-publico-22052013>
- <https://www.tdtec.com.br/produto/modulo-sensor-dht11-temperatura-e-umidade-arduino-pic/413296>
- <http://www.medicinanet.com.br/bula/899/berotec.htm>

7 Anexos

7.1 Diagramas

Sistema de Quantificação de Dosagem Medicamentosa

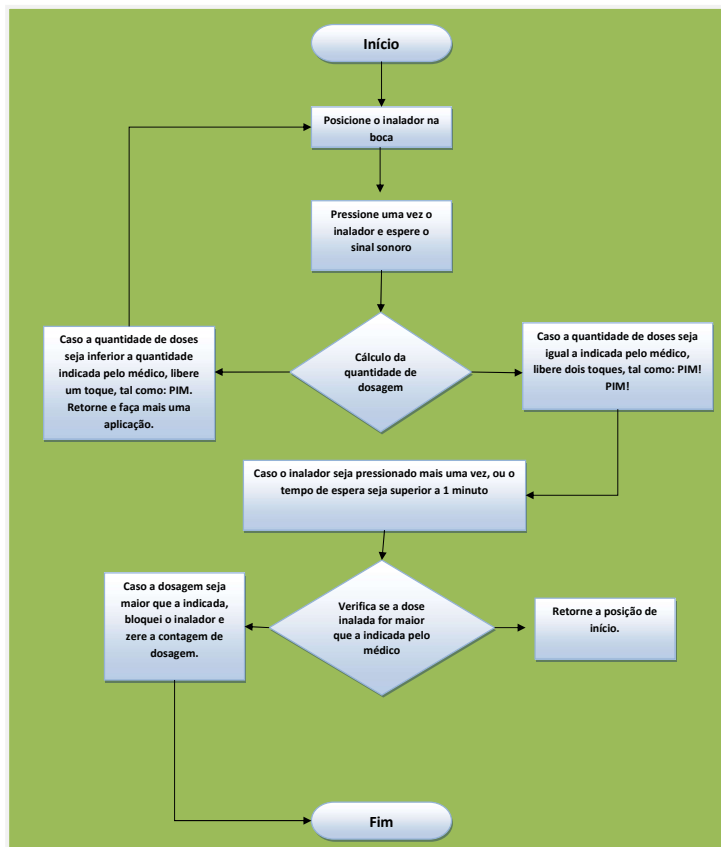


Figura 7 – Fluxograma de funcionamento