

Proposta de Projeto

Cinto de Ecolocalização para Cegos

Adriano Silva de Moraes
Faculdade do Gama
Universidade de Brasília - UnB
Gama-DF, Brasil

Hachid Habib Cury
Faculdade do Gama
Universidade de Brasília - UnB
Gama-DF, Brasil

I. REFINAMENTO EM LINGUAGEM C

O código desenvolvido no Energia foi reescrito no Code Composer Studio. Uma das consequências dessa alteração foi a utilização de recursos de mais baixo nível para controle e acionamento dos sensores. O sensor de distância, por exemplo, utiliza interrupções do timer em subrotinas para gerar os sinais que o controlam. O sinal PWM também é gerado utilizando os

mesmos recursos. O código completo encontra-se em anexo no fim deste documento.

II. REVISÃO BIBLIOGRÁFICA

[1] Davies, J., MSP430 Microcontroller Basics, Elsevier, 2008.

Anexo I – Código Principal do protótipo

```
#include <msp430.h>
#define LED (BIT6 + BIT3)
#define Trig0 BIT4
#define Echo0 BIT5

#define PERIODO 5000

unsigned int Tempol=0;
unsigned int TempoF=0;

//Função Criada para causar atraso em microsegundos
void Atraso_us(volatile unsigned int us)
{
    TACCR0 = us-1;
    TACTL = TASSEL_2 + ID_0 + MC_1;

    while((TACTL & TAIFG)==0);
    TACTL = TACLR;
}

//Função criada para controlar sensores ultrassônicos a uma distância máxima de 2.5m
int SensorUlt(volatile unsigned int Trig, volatile unsigned int Echo)
{
    int distancia;

    P1DIR |= Trig;
    P1OUT &= ~Trig;
    P1DIR &= ~Echo;

    P1OUT |= Trig;           //Colocar Trig em alto durante 10 us
    Atraso_us(10);

    TACTL = TACLR;           // Zerando o TAR
    TACTL = TASSEL_2 + ID_0 + MC_2;    // Configurando clock do timer
    TACCTL2 = CM_3 + CCIS_0 + CAP;    // Configurando modo de captura

    P1OUT &= ~Trig;

    while((P1IN&Echo)==0);
    Tempol=TAR;
    while((P1IN&Echo)==Echo);
    TempoF=TAR;

    distancia = (TempoF - Tempol)/58;    // Problema para Capturar o TAR (disrancia = distancia/58)

    return distancia;
}

void main(void)
{
    BCSCTL1 = CALBC1_1MHZ;
    DCOCTL = CALDCO_1MHZ;           // SMCLK = 1M Hz

    WDTCTL = WDTPW + WDTHOLD;       // Stop WDT

    int distancia;
```

```
P1DIR |= LED;
P1SEL |= LED;
P1SEL2 &= ~LED;
```

```
int DUTY_CYCLE=2500;
```

```
TACCR0 = PERIODO-1;
TACCR1 = DUTY_CYCLE-1;
TACCTL1 = OUTMOD_7;
TACTL = TASSEL_2 + ID_0 + MC_1;
```

```
while(1){
    distancia= SensorUlt(Trig0, Echo0);

    TACCR0 = PERIODO-1;
    TACCR1 = DUTY_CYCLE-1;
    TACCTL1 = OUTMOD_7;
    TACTL = TASSEL_2 + ID_0 + MC_1;

    if(distancia<=10){
        P1SEL |= LED;
        DUTY_CYCLE= PERIODO-2;
    }
    else if (distancia<=30){
        P1SEL |= LED;
        DUTY_CYCLE= PERIODO*0.30;
    }
    else if (distancia<=50){
        P1SEL |= LED;
        DUTY_CYCLE= PERIODO*0.1;
    }
    else if (distancia<=100){
        P1SEL&= ~LED;
        P1OUT&=~LED;
    }
    else if (distancia<=125){
        P1SEL&= ~LED;
        P1OUT&=~LED;
    }
    else {
        P1SEL&= ~LED;
        P1OUT&=~LED;
    }
}
}
```