Alunos: ***Amós Nunes e Gustavo de Carvalho***
Matrícula: ***17/0098621 | 17/0011879***

Ponto de Controle 2 - Gerador de Funções

Eletrônica Embarcada

1) Desenvolvimento de um protótipo funcional

Antes de finalizar um projeto, é de extrema importância a produção de protótipos funcionais. Tanto para entender o comportamento do produto, como para apresentar passos de evolução para o cliente e/ou Fornecedor. Então, como o objetivo de incentivar os alunos à boa prática de prototipagem é proposto o Ponto de Controle 2. Portanto, com ferramentas básicas, porém usuais, foi desenvolvido o protótipo 1 do Gerador de Funções.

O protótipo desenvolvido possuí boa parte das funcionalidades de um gerador de funções. A fim de facilitar a prototipagem, o gerador de funções foi limitado a um só tipo de onda, a saber, a onda quadrada. Foram instalados, também, dois componentes analógicos ajustáveis. Possibilitando assim a variação da Frequência e do *Duty Cycle* da onda gerada. A frequência, por decisões de projeto, foi limitada a um *range* de $100Hz$ a $3kHz$, e o *Duty Cycle* pode variar de 0% a 95%. A amplitude da onda está fixa em $3,3V$.

1.1) Componentes necessárias

Para o desenvolvimento desse protótipo foram utilizados:

(a) **2 Potenciômetros**;
Dois potenciômetros de $1k\Omega$ foram utilizados como entrada analógica para a MSP430. Utilizados para variação da Frequência e do *Duty Cycle*.

(b) **2 Resistores**;
Dois resistores de 330Ω para tratamento da queda de tensão gerada na saída analógica da MSP430.

(c) **Jumpers**;
Fios para usos diversos.

(d) **MS430F5529**;
E, obviamente, o microcontrolador MSP430. Foram utilizados os pinos de entrada analógica 6.0 e 6.1, o pino de saída 2.0 da onda quadrada e os conectores de V_{cc} e GND.

1.2) Código em C

Apesar da recomendação de utilizar a plataforma *Energy IDE*, os projetistas acharam por bem utilizar o *software* de apresentação final do projeto, o *Code Composer Software* (CSS).

Foram utilizados os *timers* e as interrupções geradas pelos ADC's (Conversores Analógicos-Digitais). Para gerar o comportamento gráfico da onda desejada. Como mostra o código comentado abaixo:

```
#include <msp430f5529.h>

#define pwm_out BIT0

unsigned int duty=0;
unsigned int frec=10000;

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;    // Stop watchdog timer

    P2DIR |= pwm_out;            // Atribui a direcao de saida do pwm do pino 2.0
    P2SEL |= pwm_out;            // Seleciona a funcao especifica do pino

    TA1CCTL1 = OUTMOD_7;         // Modo de Reset/Set do timer
    TA1CCR0 = 10000;
    // Valor inicial do Timer para divisao do SMCLK (Modifica a frequencia)
    TA1CCR1 = 0;                 // Valor inicial para o Duty (Modifica o Duty Cyclo)
```

```
    TA1CTL = MC_1 | TASSEL_2;    // Conta no modo UP e seleciona o SMCLK


    volatile unsigned int i;

    P6SEL |= BIT0 | BIT1;        // Habilita os canais de entrada analogica A0
    REFCTL0 &= ~REFMSTR;
    // Reseta o REFMSTR. Valor de referencia dos potenciometros

    ADC12CTL0 = ADC12ON+ADC12MSC+ADC12SHT02+ADC12REFON+ADC12REF2_5V;
    // Ativacao do conversor analogico-digital.

    ADC12IE =  ADC12IE0 | ADC12IE1;       // Habilita as interrupcoes dos ADC's
    ADC12CTL1 = ADC12SHP + ADC12CONSEQ_1;// Multiplos canais e habilita a amostragem
    ADC12CTL2 = ADC12RES_0;               // Resolucao de 8 Bits

    ADC12MCTL0 = ADC12SREF_1;
    // Utiliza a referencia de 2,5v de entrada para o primeiro potenciometro
    ADC12MCTL1 = ADC12SREF_1 | ADC12INCH_1;
    // Referencia de 2,5v para o segundo, seleciona o canal 6.1 de entrada.

    for ( i=0; i<0x30; i++);              // Delay para inicializacao da referencia


    ADC12CTL0 |= ADC12ENC | ADC12SC;      // Habilita as conversoes ADC.


    __enable_interrupt();                 // Habilita as interrupcoes

    while(1){
        while (!(ADC12IFG & BIT0));
        LPM1;                             //Low Power Mode 1
    }
}


{
        switch (__even_in_range(ADC12IV,8)) // Caracteriza prioridades de interrupcoes.
        {
            case 0x06:
                duty = (frec/250)*ADC12MEM0;
                // Atribui a variacao do potenciometro para a variavel Duty
                TA1CCR1 = duty;               // Atribui o valor de duty para o Timer 1
                ADC12CTL0 |= ADC12SC;         // Habiita novamente as conversoes
            break;

            case 0x08:
                frec = 10000 - (10000/250)*ADC12MEM1;
                // Atribui a variacao do potenciometro para a variavel frec
                TA1CCR0 = frec;               // Atribui o valor de frec para o Timer 2
                ADC12CTL0 |= ADC12SC;         // Habiita novamente as conversoes
            break;

            default:
            break;
        }
}
```

1.3) Testes

O protótipo foi testado conectando a saída da MSP430 na entrada de um osciloscópio. Como indica as Figuras 1 e 2:

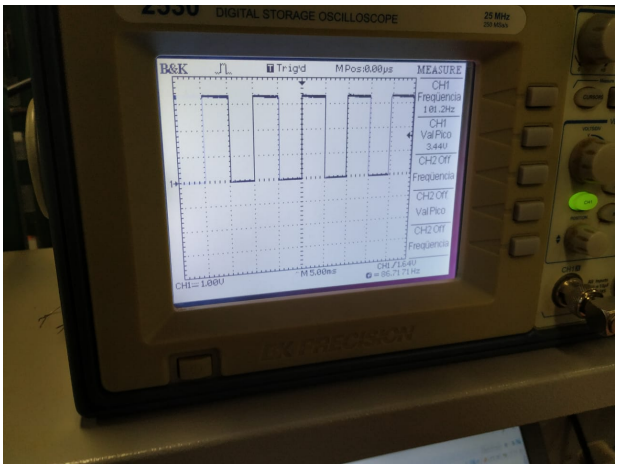

Figura 1: Imagem capturada durante o experimento. Onda gerada.

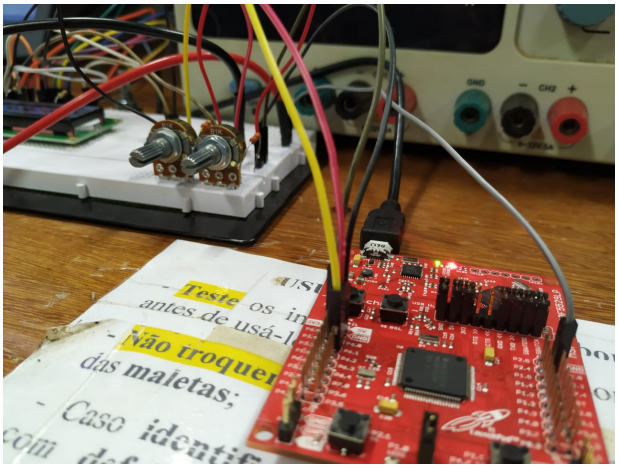

Figura 2: Imagem capturada durante o experimento. Circuito montado.