


Arquitetura de Sistemas Embarcados

Edna Barros (ensb@cin.ufpe.br)


 Centro de Informática – UFPE

Temporizadores, contadores, watchdog timers

- Temporizadores: mede intervalos de tempo
 - Geração de saída para eventos temporais
 - Ex: sinal verde deve permanecer por 10 seg.
 - Para medição de eventos em entradas
 - Ex: medida de velocidade de automóvel

Arquitetura de Sistemas Embarcados 4

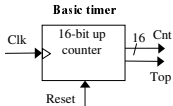
Overview

- Processadores de propósito único
 - Customizados
 - Padronizados
- Processadores de periféricos
 - Temporizadores
 - Contadores
 - UART
 - PWM
 - LCD
 - Teclado
 -

Arquitetura de Sistemas Embarcados 2

Temporizadores, contadores, watchdog timers

- Baseado na contagem de pulsos de clock
 - Considere período do clock = 10ns
 - Se contamos 20.000 pulsos de relógio
 - Então se passaram 200 microsegundos
 - Um contador de 16-bits deveria contar até $65,535 \times 10 \text{ ns} = 655.35 \text{ microseg.}$, resolução = 10 ns



Arquitetura de Sistemas Embarcados 5

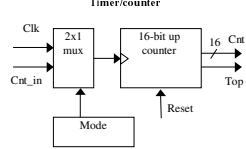
Introdução

- Processadores de propósito único
 - Realizam computação específica
 - Processadores customizados
 - Projetados para uma única tarefa
 - Processadores de propósito único padronizados
 - "Off-the-shelf" --
 - Exemplos:
 - Periféricos
 - Transmissão serial
 - Conversão analógico-digital

Arquitetura de Sistemas Embarcados 3

Contadores

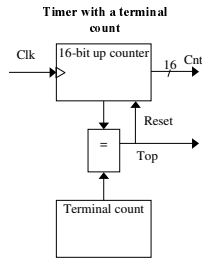
- Contador: semelhante a um temporizador porém conta pulsos de uma entrada genérica (em vez do timer)
 - Ex: contagem de carros passando por um sensor
 - Dispositivo pode ser configurado como temporizador ou contador



Arquitetura de Sistemas Embarcados 6

Outras estruturas temporizadoras

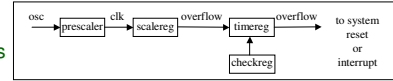
- Temporizadores de Intervalos
 - Indica quando um intervalo de tempo ocorre
 - Setando o Intervalo desejado
 - Número de ciclos de clock = $\frac{\text{Intervalo desejado}}{\text{Período de clock}}$



Arquitetura de Sistemas Embarcados 7

Watchdog timer

- Deve resetar o temporizador a cada X unidades de tempo, caso contrário o temporizador gera um sinal
- Uso comum: detecção de falha, self-reset



```

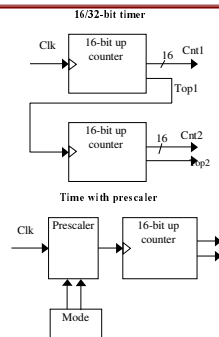
/* main.c */
main()
{
    wait until card inserted
    call watchdog_reset_routine
    while(transaction in progress)
    {
        if(button pressed)
            perform corresponding action
            call watchdog_reset_routine
    }
}
/* if watchdog_reset_routine not called every
< 2 minutes, interrupt_service_routine is
called */

watchdog_reset_routine()
/* checkreg is set so we can load value into
timereg. Zero is loaded into scalereg and
11070 is loaded into timereg */
{
    checkreg = 1
    scalereg = 0
    timereg = 11070
}
void interrupt_service_routine()
{
    eject card
    reset screen
}
    
```

Arquitetura de Sistemas Embarcados 10

Outras estruturas temporizadoras

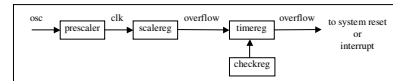
- Contadores em cascata
- Prescaler
 - Divisão do clock
 - Aumenta o range, diminui a resolução



Arquitetura de Sistemas Embarcados 8

Watchdog timer

- Outro uso: timeouts
 - Ex., máquina ATM
 - 16-bit timer, 2 microsec. resolution
 - timereg value = $2^*(2^{16}-1)-X = 131070-X$
 - For 2 min., $X = 120,000$ microsec.



```

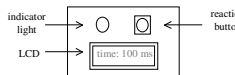
/* main.c */
main()
{
    wait until card inserted
    call watchdog_reset_routine
    while(transaction in progress)
    {
        if(button pressed)
            perform corresponding action
            call watchdog_reset_routine
    }
}
/* if watchdog_reset_routine not called every
< 2 minutes, interrupt_service_routine is
called */

watchdog_reset_routine()
/* checkreg is set so we can load value into
timereg. Zero is loaded into scalereg and
11070 is loaded into timereg */
{
    checkreg = 1
    scalereg = 0
    timereg = 11070
}
void interrupt_service_routine()
{
    eject card
    reset screen
}
    
```

Arquitetura de Sistemas Embarcados 11

Exemplo: Temporizador de Reação

- Medindo o tempo entre acendimento da lampada e o usuário apertar o botão
 - Temporizador de 16-bits, período do clk de 83.33 ns, contador incrementa a cada 6 ciclos
 - Resolução = $6 * 83.33 = 0.5$ microseg.
 - Range = $65535 * 0.5$ microseg = 32.77 mileseg.
 - Para contar milesegundos contador deve ser inicializado com 65535 – $1000/0.5 = 65535$



```

/* main.c */
#define MS_INIT 65535
void main(void)
{
    int count_milliseconds = 0;

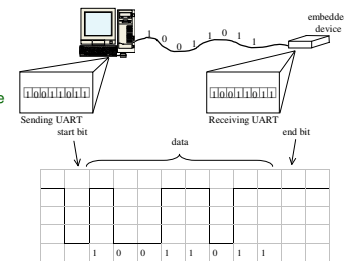
    configure timer mode
    set Cnt to MS_INIT
    wait a random amount of time
    turn on indicator light
    start timer

    while (user has not pushed reaction button)
    {
        if(Top) {
            stop timer
            set Cnt to MS_INIT
            start timer
            reset Top
            count_milliseconds++;
        }
    }
    turn light off
    printf("time: %i ms", count_milliseconds);
}
    
```

Arquitetura de Sistemas Embarcados 9

Transmissão Serial Usando UARTs

- UART: Universal Asynchronous Receiver Transmitter
 - Transmite dado serialmente
 - Recebe dado serialmente e converte para paralelo
- Parity: bit extra para detecção de erros
- Bits de Start e de stop
- Baud rate
 - Mudanças de sinal por segundo
 - Bit rate é usada frequentemente



Arquitetura de Sistemas Embarcados 12

Pulse width modulator - PWM

- Gera pulsos com tempo de high/low especificado
- Duty cycle: % do tempo em nível alto
 - Onda quadrada: 50% duty cycle

Arquitetura de Sistemas Embarcados 13

Controlador LCD

```

void WriteChar(char c)
{
  RS = 1;           /* indicate data being sent */
  DATA_BUS = c;   /* send data to LCD */
  EnableLCD45;     /* toggle the LCD with appropriate delay */
}
    
```

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description
0	0	0	0	0	0	0	0	0	1	Clear all display, return cursor home
0	0	0	0	0	0	0	0	1	1	Return cursor home
0	0	0	0	0	0	0	0	1	S	Set cursor move direction and/or specifies not to shift display
0	0	0	0	0	0	1	D	C	B	Cursor of all displays (A, cursor ON/OFF (C), and home position (B))
0	0	0	0	0	1	S	C	R	L	Move cursor and shift display
0	0	0	0	1	D	L	N	F	*	Set interface data length, number of display lines, and character font
1	0									WRITE DATA

Arquitetura de Sistemas Embarcados 16

Pulse width modulator - PWM

- Uso comum: controle de tensão para equipamento elétrico
 - Mais simples que conversor DC-DC ou conversor digital-analógico
 - Velocidade de motor, lâmpadas com controle de luminosidade
- Outro uso: codificação de comandos, recebe temporização do usuário para decodificar

Arquitetura de Sistemas Embarcados 14

Controlador de Teclado

N=4, M=4

Arquitetura de Sistemas Embarcados 17

Controlando um motor DC motor com um PWM

Input Voltage	% of Maximum Voltage Applied	RPM of DC Motor
0	0	0
2.5	50	1840
3.75	75	6900
5.0	100	9200

```

void main(void)
{
  /* controls period */
  PWM1 = 0x1f;
  /* controls duty cycle */
  PWM1 = 0x7f;
  while(1);
}
    
```

Arquitetura de Sistemas Embarcados 15

Controlador de motor de passo

- Motor de passo: rotaciona número fixo de graus dado um sinal de passos
 - Em contraste: motor DC rotaciona quando potência é aplicada
- Rotação obtida pela aplicação de tensão específica nas bobinas
- Controlador mais simples

Sequence	A	B	A'	B'
1	+	+	-	-
2	-	+	+	-
3	-	-	-	+
4	+	-	+	-
5	+	+	-	+

Arquitetura de Sistemas Embarcados 18

Motor de Passo com Controlador (driver)

```

/* main.c */
sbit clk=P1^1;
sbit csw=P1^0;
void delay(void)
{
    int i, j;
    for (i=0; i<1000; i++)
        for (j=0; j<50; j++)
            i++;
}

void main(void)
{
    /*turn the motor forward */
    csw=1; /* set direction */
    clk=0; /* pulse clock */
    delay();

    /*turn the motor backwards */
    csw=1; /* set direction */
    clk=1; /* pulse clock */
    delay();
}
    
```

The output pins on the stepper motor driver do not provide enough current to drive the stepper motor. To amplify the current, a buffer is needed. One possible implementation of the buffers is pictured to the left. Q1 is an MJE2955T NPN transistor and Q2 is an MJE2955T PNP transistor. A is connected to the 8051 microcontroller and B is connected to the stepper motor.

Arquitura de Sistemas Embarcados 19

Conversão Analógico-Digital

proportionality analog to digital digital to analog

Arquitura de Sistemas Embarcados 22

Motor de Passo sem Controlador (driver)

```

/*main.c*/
sbit noA=P2^0;
sbit noB=P2^1;
sbit noC=P2^2;
sbit noD=P2^3;
sbit dir=P2^4;
void delay()
{
    int i, k;
    for(i=0; i<5000; i++)
        for(k=0; k<10000; k++)
            a++;
}

void main(int dir, int steps)
{
    int y, z;
    /* clockwise movement */
    if(dir == 1)
        for(y=0; y<steps; y++)
            for(z=0; z<4; z++)
                {
                    noA=lookap[z];
                    noB=lookap[z+1];
                    noC=lookap[z+2];
                    noD=lookap[z+3];
                    delay(x);
                }
}
    
```

A possible way to implement the buffers is located below. The 8051 cannot drive the stepper motor, so several transistors were added to increase the current going to the stepper motor. Q1 are MJE2955T NPN transistors and Q2 is an MJE2955T PNP transistor. A is connected to the 8051 microcontroller and B is connected to the stepper motor.

Arquitura de Sistemas Embarcados 20

Conversão Digital Analógico

Simplified binary weighted resistor DAC

8.75V

R - 2R ladder DAC

9.375 max.

Arquitura de Sistemas Embarcados 23

Conversão AD ou DA

- AD: conversão analógico para digital
- DA: conversão digital para analógica

Arquitura de Sistemas Embarcados 21

Conversão Digital Analógico

Four-Bit D/A Converter

One way to achieve D/A conversion is to use a summing amplifier.

Inputs in volts are weighted in the summing amplifier to produce the corresponding analog voltage.

$$1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 13$$

Scaled resistances into a summing junction.

$V_{out} = -1.3V$

A 12 bit DAC of this type would require the largest scaling resistor to be 2048 times the smallest, so this approach quickly becomes impractical.

This approach is not satisfactory for a large number of bits because it requires too much precision in the summing resistors. This problem is overcome in the R-2R network DAC.

Arquitura de Sistemas Embarcados 24

Conversão Digital Analógico

R-2R Ladder DAC

The **summing amplifier** with the R-2R ladder of resistances shown produces the output

$$V_{out} = \frac{R_f}{R} V_{ref} \left[\frac{D_0}{16} + \frac{D_1}{8} + \frac{D_2}{4} + \frac{D_3}{2} \right]$$

where the D's take the value 0 or 1. The digital inputs could be **TTL** voltages which close the **switches** on a logical 1 and leave it grounded for a logical 0. This is illustrated for 4 bits, but can be extended to any number with just the resistance values R and 2R.

[More Detail](#)

Arquitura de Sistemas Embarcados 25

Conversão Digital Analógico

$V_{out} = -V_{in} * (R_{fb}/R_{in})$

$V_{out} = -(-5V) * (15K / 30K) = 2.5V$

$V_{out} = -(V_{ref} * (Digital Input)_{10}) / 256$ (the 10 means "base 10 number").

Arquitura de Sistemas Embarcados 28

Conversão Digital Analógico

R-2R Ladder DAC Details

A logic 1 (e.g., TTL 5V) closes the switch, applying Vref to the summing junction. A logic 0 leaves that input grounded.

A common reference voltage is used because of the variation in what voltage will be produced by a logic 1 or 0.

Having just R and 2R places much less stringent requirements on the required precision of the resistors than the **scaled resistor** approach to DAC.

These weighting factors can be obtained by a **Thevenin** analysis of each input point.

$$V_{out} = \frac{R_f}{R} V_{ref} \left[\frac{D_0}{16} + \frac{D_1}{8} + \frac{D_2}{4} + \frac{D_3}{2} \right]$$

Arquitura de Sistemas Embarcados 26

Conversão Analógico Digital

- Exemplo: valor de entrada no range de 0 a 7.5 que deve ser representada com 4 dígitos. Qual a codificação ideal?
- Fórmula Geral
 - $e/ V_{max} = d/(2^n - 1)$

Arquitura de Sistemas Embarcados 29

Conversão Digital Analógico

Arquitura de Sistemas Embarcados 27

Conversão AD

Simple ramp and comparator ADC

Arquitura de Sistemas Embarcados 30

Conversão Analógico Digital usando Aproximações sucessivas

Given an analog input signal whose voltage should range from 0 to 15 volts, and an 8-bit digital encoding, calculate the correct encoding for 5 volts. Then trace the successive-approximation approach to find the correct encoding.

$5/15 = d/(2^8-1)$ Encoding: 01010101
 $d = 85$

Successive-approximation method

$\frac{1}{2}(V_{\text{min}} - V_{\text{min}}) = 7.5$ volts $V_{\text{min}} = 7.5$ volts.	$\frac{1}{2}(5.63 + 4.69) = 5.16$ volts $V_{\text{min}} = 5.16$ volts.
$\frac{1}{2}(7.5 + 0) = 3.75$ volts $V_{\text{min}} = 3.75$ volts.	$\frac{1}{2}(5.16 + 4.69) = 4.93$ volts $V_{\text{min}} = 4.93$ volts.
$\frac{1}{2}(7.5 + 3.75) = 5.63$ volts $V_{\text{min}} = 5.63$ volts.	$\frac{1}{2}(5.16 + 4.93) = 5.05$ volts $V_{\text{min}} = 5.05$ volts.
$\frac{1}{2}(5.63 + 3.75) = 4.69$ volts $V_{\text{min}} = 4.69$ volts.	$\frac{1}{2}(5.05 + 4.93) = 4.99$ volts $V_{\text{min}} = 4.99$ volts.

00000000
0 0 0 0 0 0 0 0

01010100
0 1 0 1 0 1 0 0

01010100
0 1 0 1 0 1 0 0

01010101
0 1 0 1 0 1 0 1

01010101
0 1 0 1 0 1 0 1

01010101
0 1 0 1 0 1 0 1

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO Arquitetura de Sistemas Embarcados 31

Conversão AD – Aproximação sucessivas

- **Fast process - 1 - 100µsecs**
- **Result always n clocks after start**
- **Used extensively for 12-16bit DAQ systems**

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO Arquitetura de Sistemas Embarcados 32