

Câmara de Eco Digital

EUDOXIO G. RAMOS JUNIOR

É raro encontrarmos na Internet ou em revistas de Eletrônica aplicações do microcontrolador PIC em controle ou processamento de áudio digital.

Além das aplicações já conhecidas como acender *leds*, ler teclado ou comandar *displays*, o PIC pode ser usado em aplicações de áudio não só para a geração de tons, mas também para o processamento e tratamento de sinais de áudio em circuitos de filtragem, equalização e geração de efeitos sonoros.

Neste artigo descrevemos uma aplicação onde o PIC é usado para criar efeitos de eco.

Referimo-nos aqui somente à parte que cabe ao PIC, pois a teoria geral de funcionamento de Câmaras de Eco digitais já foi publicado em artigos passados nesta revista.

Se bem que no exemplo prático descrito, o PIC não manipula diretamente o sinal de áudio digitalizado, aqui ele é responsável pelo trabalho de comandar os conversores de áudio, memória, teclas e *leds* e também criar uma linha de atraso digital necessária ao efeito.

Em artigos futuros mostraremos aplicações práticas onde o PIC manipula e processa diretamente sinais de áudio no formato PCM com resolução de 16 bits estéreo, na criação de efeitos mais sofisticados.

Neste circuito, usamos uma memória RAM dinâmica de 64k bits como linha de atraso. O PIC neste caso tem a função de gerar os sinais necessários para esta memória, afim de coordenar a movimentação dos dados dentro dela.

Para gerar os endereços de RAS e CAS para a memória, criamos no PIC, dois contadores de oito bits chamados RASCONT e CASCONT. Estes dois contadores trabalham em conjun-

A Câmara de Eco Digital CE01-pcb é um *kit* eletrônico completo que possibilita a produção de efeitos de eco a partir de sinais de áudio ou voz.

Pode ser conectada com grande eficiência em fontes de som como microfones, guitarras, instrumentos musicais eletrônicos, pré-amplificadores, mesas de som e sistemas para Karaoke.

Simple e fácil de usar, o kit CE01-pcb já vem completamente montado e calibrado, bastando para seu funcionamento um transformador com saída de 14 a 16 V AC.

O kit CE01-pcb trabalha com conversão AD/DA do tipo Delta Sigma de 1 bit possibilitando melhor resultado sonoro quando comparado com o sistema BBD.

to formando um contador de 16 bits capaz de contar até 64000, que é a capacidade total da memória.

Os valores destes dois contadores são enviados à memória através do barramento formado pelo PORT B do PIC.

No PORT A do PIC temos os pinos responsáveis pela sinalização dos pinos RAS, CAS e WE da memória, e também um pino de entrada para a tecla de seleção do tempo de atraso.

O tempo de atraso é calculado levando-se em conta o tamanho da me-

mória e a velocidade do clock durante os ciclos de leitura e escrita.

Os conversores AD e DA são devidamente sinalizados aproveitando-se o sinal presente no pino RAS da memória.

Na figura 1 é mostrado o Diagrama de Blocos da aplicação.

DESCRIÇÃO DO PROGRAMA

Um programa para gerar uma linha de atraso é bem simples e curto.

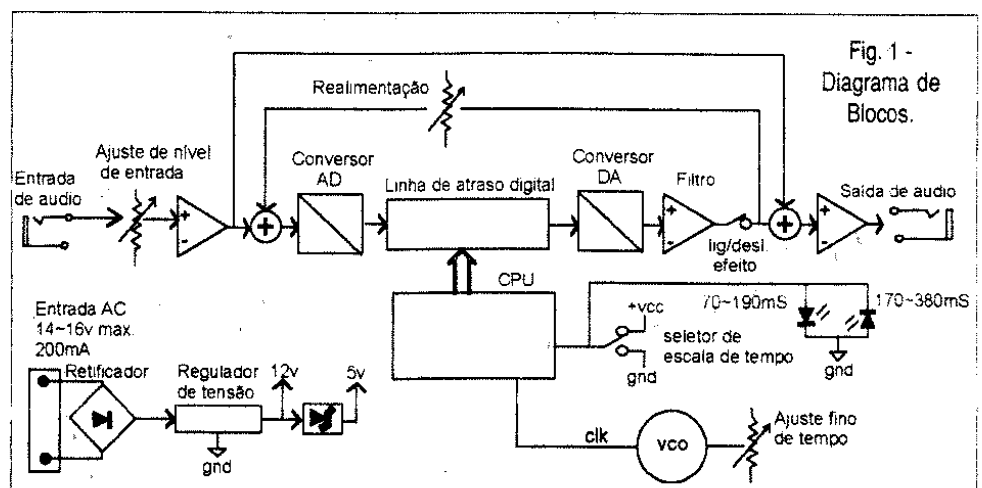
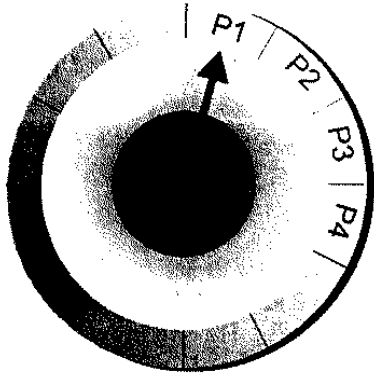


Fig. 1 - Diagrama de Blocos.

Na página seguinte temos uma descrição detalhada de cada linha deste programa. O que está a direita do ponto e virgula (;) é apenas comentário e não é tratado pelo compilador do PIC.



A linha de atraso gerada pelo programa é chamada de *Buffer Circular*, e para maior clareza a representamos como na figura abaixo.

P₁, P₂, P₃ etc. são as posições de memória endereçadas pelo ponteiro. O ponteiro, neste caso, são os contadores RASCONT e CASCONT.

Na figura 2 é ilustrado o esquema elétrico completo.

CARACTERÍSTICAS

Entrada de microfone com ajuste de nível.

Saída de Linha.

Ajuste do tempo de atraso de 70 ms a 380 ms.

Chave Liga/Desliga o efeito.

Controle de realimentação.

Resposta em frequência direta: 40 Hz a 20 kHz.

Resposta em frequência do efeito: 40 Hz a 7 kHz.

Tensão de alimentação: 14 a 16 V AC, 200 mA.

Distorção Harmônica Total: 0,1%

Nível de ruído: -80 dB a -68 dB.

Dimensões da placa: 66 mm x 196 mm

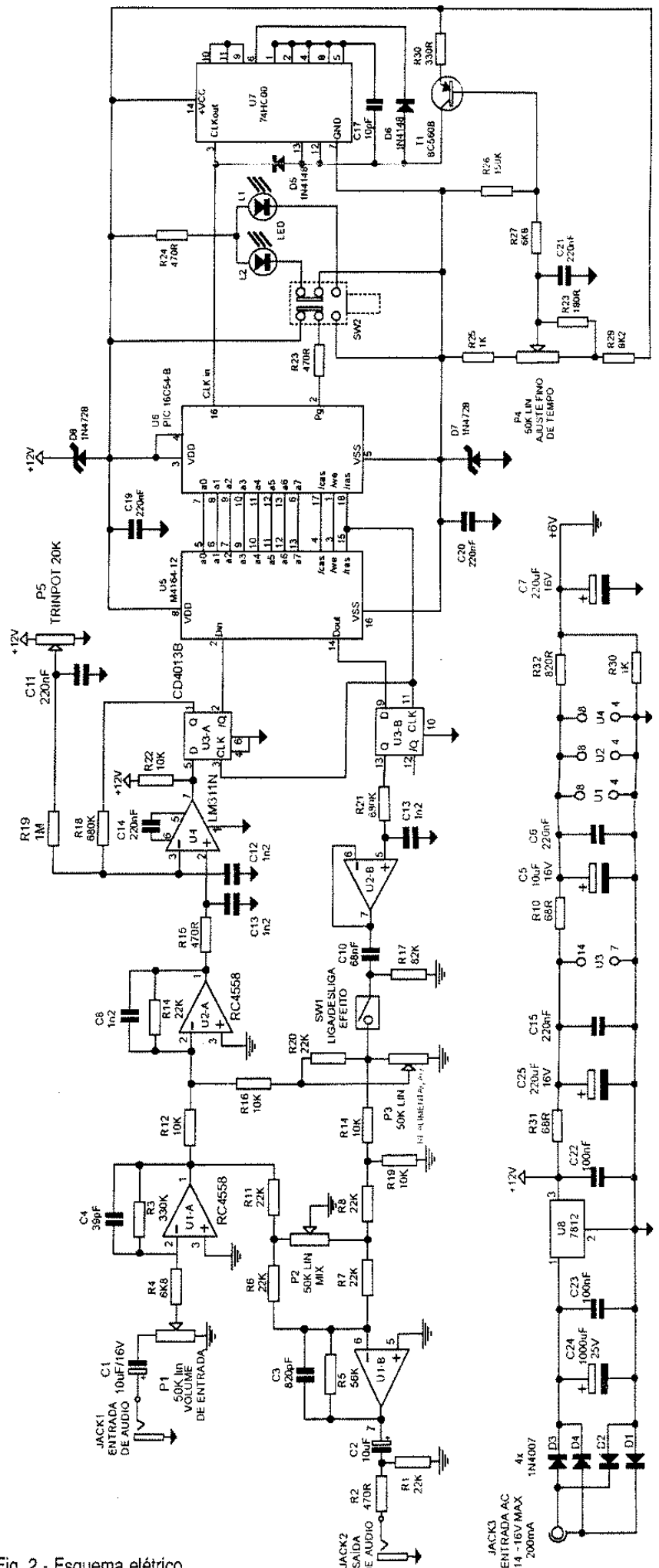


Fig. 2 - Esquema elétrico.

O circuito efetua um ciclo de leitura e gravação em cada posição da memória apontada por este ponteiro. Assim, P_1 é lido e o seu conteúdo é enviado ao conversor D/A. Após isto o dado disponível na saída do conversor A/D é gravado nesta mesma posição. Ato contínuo, o ponteiro é incrementado e este ciclo de leitura e gravação é repetido nesta nova posição. Depois de percorrer toda a memória, o ponteiro volta a P_1 , onde encontraremos o dado anteriormente gravado.

O tempo de atraso provocado no sinal de áudio irá depender de dois parâmetros: do tamanho da memória e da velocidade de deslocamento do ponteiro. Em nosso projeto trabalhamos com estes dois parâmetros a fim de ajustar o tempo de atraso.

O tempo de atraso é dividido em duas escalas selecionadas pela chave SW2.

Na primeira escala, o tempo de atraso pode variar de 70 ms até 190 ms. Na segunda escala de 170 ms a 380 ms. A variação do tempo dentro da escala selecionada é feita através do potenciômetro P_3 que ajusta a frequência do *clock* enviado ao PIC.

O programa em *assembler* mostrado aqui, é o mesmo que está gravado no PIC usado em nosso kit.

Nele, só falta a rotina de leitura de tecla, que é fácil de ser implementada. Nesta, quando a tecla é acionada, o programa carrega em CASCONT um valor menor daquele carregado na inicialização.

Isto faz com que o tamanho do *Buffer* seja diminuído.

Algumas questões interessantes aparecem, quando olhamos para a figura do *Buffer Circular*.

1°- Como se apresentará o sinal de áudio, se durante a varredura variarmos a velocidade do ponteiro?

2°- O que acontecerá se alterarmos a direção de rotação do ponteiro durante a varredura?

3°- Podemos acrescentar outros ponteiros neste mesmo *Buffer*?

4°- O que acontecerá se conectarmos diversos *Buffers* em série ou em paralelo?

Tente imaginar o resultado de cada situação acima. Muitos processadores de efeitos encontrados no mercado as aplicam na geração de efeitos sonoros interessantes. Futuramente falaremos sobre isso. ■

Lista de Material

U ₆ - CPU PIC16c54-b	ELETROLÍTICO
U ₅ - 4164-12 - MEMÓRIA	C ₈ , C ₁₂ , C ₁₃ - 1n2/63 V - POLIÉSTER
U ₃ - CD4013B - FLIP-FLOP	C ₉ - 1n5/62 V - POLIÉSTER
U ₇ - 74HC00 - QUAD NAND	C ₁₀ - 68 nF/63 V - POLIÉSTER
U ₈ - LM7812 - REGULADOR 5 V	P ₁ , P ₂ , P ₃ , P ₄ - 50 KB - Potenciômetro
U ₁ , U ₂ - RC4558 - OPERACIONAL	R ₁₀ , R ₃₁ - 68 Ω - 1/4 W 5%
U ₄ - LM311N - COMPARADOR	R ₂₈ - 180 Ω - 1/4 W 5%
T ₁ - BC560B - TRANSISTOR	R ₃₀ - 330 Ω - 1/4 W 5%
D ₅ , D ₆ - 1N4148 - DIODO	R ₂ , R ₁₅ , R ₂₃ , R ₂₄ - 470 Ω - 1/4 W 5%
D ₁ , D ₂ , D ₃ , D ₄ - 1N4007 - Retificador	R ₃₃ - 820 Ω - 1/4 W 5%
D ₇ , D ₈ - 1N4728 - ZENER 3,1 V	R ₂₅ , R ₃₂ - 1 kΩ - 1/4 W 5%
L ₁ , L ₂ - LED 3mm - VERMELHO	R ₂₉ - 8,2 kΩ - 1/4 W 5%
P ₅ - TRIMPOT 20 k - Mini, Horizontal	R ₄ , R ₂₇ - 6,8 kΩ - 1/4 W 5%
C ₁₇ - 10 pF/63 V - CERÂMICO	R ₉ , R ₁₂ , R ₁₃ , R ₁₆ , R ₂₂ - 10 kΩ - 1/4 W 5%
C ₄ - 39 pF/63 V - CERÂMICO	R ₁₁ , R ₆ , R ₇ , R ₈ , R ₁₁ , R ₁₄ , R ₂₀ - 22 kΩ - 1/4 W
C ₃ - 820 pF/63 V - CERÂMICO	R ₅ - 68 kΩ - 1/4 W 5%
C ₆ , C ₁₉ , C ₂₀ , C ₂₂ , C ₂₃ - 100KpF/50V - CERÂMICO	R ₁₇ - 82 kΩ - 1/4 W 5%
C ₁₁ , C ₁₄ , C ₁₅ , C ₁₆ , C ₁₈ , C ₂₁ - 220 KpF/50 V - CERÂMICO	R ₂₆ - 150 kΩ - 1/4 W 5%
C ₁ , C ₂ , C ₅ - 10 μF/16V RADIAL	R ₃ - 330 kΩ - 1/4 W 5%
ELETROLÍTICO	R ₂₁ , R ₁₈ - 680 kΩ - 1/4 W 5%
C ₇ , C ₂₅ - 220 μF/16 V RADIAL	R ₁₉ - 1 MΩ - 1/4 W 5%
ELETROLÍTICO	SW ₁ , SW ₂ - CHAVE SPJ - C/ TRAVA
C ₂₄ - 1000 μF/25 V - RADIAL	JACK ₁ , JACK ₂ - JACK P ₁₀ - TB P/ CI
	JACK ₃ - JACK FONTE - CI
	PCI A ₃ - PLACA

O PROGRAMA

```

MOV LW 0xFF ; INICIALIZA OS CONTADORES
MOVWF CASCONT ; CASCONT E RASCONT
MOVWF RASCONT ; COM O VALOR 255.

INICIO
MOVF RASCONT, W ; MOVE O CONTADOR RASCONT
MOVWF PORTB ; PARA A PORTA B.
BCF PINO_RAS ; ZERA O PINO RAS DA MEMÓRIA
MOVF CASCONT, W ; MOVE O CONTADOR CASCONT
MOVWF PORTB ; PARA A PORTA B.
BCF PINO_CAS ; ZERA O PINO CAS DA MEMÓRIA.
; A PARTIR DESTES PONTOS, O DADO LIDO
; JÁ ESTARÁ DISPONÍVEL NO PINO DE
; SAÍDA DA MEMÓRIA. ESTE DADO SERÁ
; CARREGADO NO CONVERSOR D/A NA
; SUBIDA DO PINO_RAS LOGO ADIANTE.
BCF PINO_WE ; DESCE O PINO WE DA MEMO PARA
; GRAVAR UM NOVO DADO
; NA MESMA POSIÇÃO DO DADO LIDO.
BSF PINO_RAS ; SOBE OS PINOS RAS, CAS E WE
; E TERMINA O CICLO DE
BSF PINO_CAS ; LEITURA E ESCRITA.
BSF PINO_WE ; OS CONVERSORES SÃO ATUALIZADOS
; NA SUBIDA DO PINO RAS.
DECFSZ RASCONT, F ; DECREMENTA O CONTADOR RASCONT
GOTO INICIO ; SE ESTE CONTADOR NÃO CHEGOU A ZERO,
; O PROGRAMA REPETIRÁ TODO O
; PROCESSO ACIMA, NA PROXIMA
; POSIÇÃO DA MEMÓRIA.
DECFSZ CASCONT, F ; SE CHEGOU A ZERO, DECREMENTA
GOTO INICIO ; O CONTADOR CASCONT
; E RECOMEÇA NOVAMENTE.

```