

dez 02, 15 20:49	passaro.as	Page 1/17
;Passaro Bamboleante		
;83449 Duarte Galvao		
;83539 Pedro Caldeira		
;Turno segunda feira, 11:00-12:30, grupo 13		
;85080 Joao Pina		
; ----- CONSTANTES DO JOGO -----		
; -----		
SP_INICIAL	FFFh	
INT_MASK_ADDR	FFFh	
INT_MASK	000000000000111b	
RANDOM_MASK	100000000010110b	
BIT_MASK	1000000000000000b	
IO_DISPLAY	FFF0h	
LCD_DISP_C	FFF4h	
LCD_DISP_W	FFF5h	
TEMP_UC	FFF6h	
TEMP_PC	FFF7h	
ATIVAR_TEMP	0001h	
LED_DISPLAY	FFF8h	
TXT_DISPLAY_C	FFFCh	
TXT_DISPLAY_W	FFFeh	
LIMPAR_JANELA	FFFFh	
DELAY_COUNT	0001h	
NIBBLE_MASK	000fh	
NUM_NIBBLES	4	
NIBBLE	4	
BYTE	8	
COL_PASSARO	20	
POS1_PASSARO	0C00h	
CH_ESPACO	' '	
CH_LIMITE	'-'	
CH Obstaculo	'X'	
CH_ZERO	'0'	
TAM_PASSARO	2	
NUM_L_SUBIR	2	
DISPLAY_W	79	
LIM_TOPO	0100h	
ULTIMA_LINHA	1700h	
NUM_BOTOES	14	
LINHA	0100h	
GRAVIDADE	20	
INTERVALO_OBS	5 ; Espaco entre cada coluna de obstaculos	
NUM_MAX_OBS	14	
ESPACO_OBS	0600h ; Espaco por onde o passaro passa	
BASE_DEC	10	
MAX_NIVEL	16	
; Palavra de memoria que contem a variavel de contagem		
ORIG	8000h	
STR	'O>'	
MsgInicL1	10, 'Prepare-se'	; Comprimento, Mensagem
MsgInicL2	22, 'Prima o interruptor I1'	; Comprimento, Mensagem
MsgFimL1	11, 'Fim do Jogo'	; Comprimento, Mensagem
MsgFimL2	4, '0000'	; Comprimento, Mensagem
MsgLcd	10, 'Distancia:'	; Comprimento, Mensagem
; ----- VARIAVEIS DO JOGO -----		
; -----		
; R1=Contador da dificuldade		

dez 02, 15 20:49	passaro.as	Page 2/17
<pre> ;R3=Ni (Random) ;R4=Posicao do cursor ;R5=(y0)Posicao da ultima vez que o passaro subiu ;R6=(t)Tempo desde a ultima vez que o passaro subiu (em ciclos do temporizador) ;R7=Interrupcoes ativadas ;(T, I0, I1, I2, ...) (primeiro bit para T e por ai em diante) </pre>		
Distancia	<b>TAB</b> 1	; Distancia percorrida
Nivel	<b>TAB</b> 1	; Nivel invertido (1-16) (16-N)
PosPassaro	<b>TAB</b> 1	; Posicao do passaro em virgula
PosAPassaro	<b>TAB</b> 1	; Posicao anterior do passaro (e
m v.f.)		
LinhasASubir	<b>TAB</b> 1	; Linhas que faltam subir
EspObstaculos	<b>TAB</b> NUM_MAX_OBS	; (8043) No. maximo de obstaculos
FaseObstaculos	<b>TAB</b> 1	; Fase, de 0 a 5, da movimentacao dos obsta
culos		
NumObstaculos	<b>TAB</b> 1	; Numero de obstaculos existente
s na Janela		
EstadoLEDs	<b>TAB</b> 1	; Conteudo dos LEDs
<pre> ; ----- ; ----- TABELA DE INTERRUPTCOES ----- ; ----- ; ----- </pre>		
<b>INT0</b>	<b>ORIG</b> FE00h	
<b>INT1</b>	<b>WORD</b> RInt0	
<b>INT2</b>	<b>WORD</b> RInt1	; Diminui dificuldade
	<b>WORD</b> RInt2	; Aumenta dificuldade
	<b>ORIG</b> FE0Fh	
<b>INT15</b>	<b>WORD</b> RIntTemp	
<pre> ; INICIO DO CODIGO <b>ORIG</b> 0000h <b>JMP</b> Inicio </pre>		
<pre> ; ----- ; ----- INTERRUPTCOES ----- ; ----- ; ----- Rotina de servico a interrupcao do botao I0 </pre>		
Entradas: R7		
Saídas: R7		
Efeitos: ---		
RInt0:	<b>PUSH</b> R1	
	<b>MOV</b> R1, BIT_MASK	
	<b>ROR</b> R1, 1	
	<b>OR</b> R7, R1	
	<b>POP</b> R1	
	<b>RTI</b>	
<pre> ; RInt1: Rotina de servico a interrupcao do botao I1 ; Entradas: R7 ; Saídas: R7 ; Efeitos: --- </pre>		
RInt1:	<b>PUSH</b> R1	
	<b>MOV</b> R1, BIT_MASK	
	<b>ROR</b> R1, 2	
	<b>OR</b> R7, R1	
	<b>POP</b> R1	
	<b>RTI</b>	
<pre> ; RInt2: Rotina de servico a interrupcao do botao I2 ; Entradas: R7 </pre>		

dez 02, 15 20:49	passaro.as	Page 3/17
<pre>; Saídas: R7 ; Efeitos: --- RInt2:  PUSH       MOV R1, BIT_MASK       ROR R1, 3       OR  R7, R1       POP R1       RTI  ; RIntTemp: Rotina de servico a interrupcao do temporizador ; Entradas: R7 ; Saídas: R7 ; Efeitos: --- RIntTemp:  PUSH           OR  R7, BIT_MASK           POP R1           RTI  ; ----- ROTINAS AUXILIARES ----- ; ; ApagaEcra: Rotina que apaga o ecra na sua totalidade ; Entradas: --- ; Saídas: --- ; Efeitos: Altera R4, Altera M[TXT_DISPLAY_C1] ApagaEcra:  PUSH R2           MOV R1, CH_ESPACO           MOV R4, R0            ; Enquanto o cursor nao chegar a ultima poiscao,           o ciclo continua           AE_C1:  MOV M[TXT_DISPLAY_C1], R4                   MOV M[TXT_DISPLAY_W], R1                   INC R4                   MOV R2, ULTIMA_LINHA                   ADD R2, DISPLAY_W                   CMP R4, R2                   BR.NZ AE_C1                    POP R2                   POP R1                   RET  ; DesenhaRim: Rotina que desenha uma linha de um carater a escolha ; Entradas: R4 ; Saídas: --- ; Efeitos: Altera o R4 DesenhaRim:  PUSH R1             PUSH R2 ; Carater a escrever              MOV R1, DISPLAY_W             MOV R2, CH_LIMITE              ; Enquanto o cursor nao chegar a ultima coluna,             o ciclo continua             DL_L1:  MOV M[TXT_DISPLAY_C1], R4                     MOV M[TXT_DISPLAY_W], R2                     INC R4                     DEC R1</pre>		

dez 02, 15 20:49	passaro.as	Page 4/17
	<pre>BR.NZ DL_L1  POP R2 POP R1 RET  ; EscLinhaC: Rotina que escreve uma linha no meio do ecra ; Entradas: (pilha) Linha, Mensagem ; Saídas: --- ; Efeitos: --- EscLinhaC:  PUSH R1 ; Linha           PUSH R2 ; Endereco da Mensagem           PUSH R3 ; Comprimento da mensagem            MOV R1, M[SP+6]           MOV R2, M[SP+5]           MOV R3, M[R2]           INC R2;Passar um a frente pois o primeiro el e o comprimento           MOV R4, DISPLAY_W           SUB R4, R3           SHR R4, 1           SHL R1, 8           ADD R4, R1 ; Endereco onde começa a mensagem            ELC_C1:  MOV M[TXT_DISPLAY_C1], R4                   MOV R1, M[R2] ; R1 = Letra atual                   MOV M[TXT_DISPLAY_W], R1                   INC R2                   INC R4                   DEC R3                   BR.NZ ELC_C1            POP R3           POP R2           POP R1           RETN 2  ; MataPassaro: Rotina que acaba o jogo ; Entradas: --- ; Saídas: --- ; Efeitos: --- MataPassaro:  CALL ApagaEcra               PUSH 12               PUSH MsgFimL1                CALL EscLinhaC               PUSH M[Distancial]               CALL CalcPontuacao               CALL AtMsgFimL2               PUSH 14               PUSH MsgFimL2               CALL EscLinhaC                MP_C1:  BR MP_C1                     RET  ; ResetTemp: Rotina que ativa o temporizador ; Entradas: --- ; Saídas: --- ; Efeitos: --- ResetTemp:    PUSH R1</pre>	

dez 02, 15 20:49	passaro.as	Page 5/17
	<pre>MOV R1, DELAY_COUNT MOV M[TEMP_UC], R1 MOV R1, ATIVAR_TEMP MOV M[TEMP_PC], R1  POP R1 RET  ; TelaDeJogo: Rotina que desenha os limites ; Entradas: --- ; Saídas: --- ; Efeitos: --- TelaDeJogo: MOV R4, R0 CALL DesenharLim MOV R4, ULTIMA_LINHA CALL DesenharLim RET  ; ----- CALCULO E ESCRITA DA DISTANCIA E PONTUACAO ----- ; ----- ; AtMsgFimL2: Atualizar a MsgFimL2 conforme a pontuacao ; Entradas: Pilha - pontuacao ; Saídas: M[MsgFimL2] ; Efeitos: --- AtMsgFimL2: PUSH R1 ; Pontuacao PUSH R2 ; Endereco a escrever PUSH R3 ; Algarismo a escrever  MOV R1, M[SP+5] MOV R2, MsgFimL2 ADD R2, M[R2]  ; Enquanto nao forem escritos na memoria todos os elementos, o ciclo conti nua AMFL2_C1: MOV R3, BASE_DEC DIV R1, R3  MOV M[R2], R3 DEC R2 CMP R1, R0 BR.NZ AMFL2_C1  POP R3 POP R2 POP R1 RETN 1  ; CalcPontuacao: Rotina que efectua o calculo da pontuacao ; Entradas: Pilha - distancia ; Saídas: Pilha - pontuacao ; Efeitos: --- CalcPontuacao: PUSH R1 ; Distancia PUSH R2 ; DISPLAY_W - COL_PASSARO + 1  ; Formula: ; (1) Pontuacao = 0, se R1 &lt; R2 ; (2) Pontuacao = (R1 - R2)/(INTERVALO_OBS + 1) + 1, se R1 &gt;= R2 ; Inicializar R1 MOV R1, M[SP+4] ; Inicializar R2</pre>	

dez 02, 15 20:49	passaro.as	Page 6/17
	<pre>MOV R2, DISPLAY_W SUB R2, COL_PASSARO INC R2  ; Verificar se R1 &lt; R2 SUB R1, R2 BR.NN CP_1  ; Se R1 &lt; R2, executar (1) MOV R1, R0 BR CP_2  ; Se R1 &gt;= R2, executar (2) MOV R2, INTERVALO_OBS INC R2 DIV R1, R2 INC R1  ; Devolver pontuacao MOV M[SP+4], R1  POP R2 POP R1 RET  ; EscDistancia: Rotina que escreve a distancia percorrida no display LCD. ; Entradas: M[Distancia] ; Saídas: ---- ; Efeitos: Altera M[LCD_DISP_C] EscDistancia: PUSH R1 ; Coordenada do LCD PUSH R2 ; Distancia PUSH R3 ; Algoritmo a ser escrito PUSH R4 ; Numero de algoritmos da distancia  CALL EscLCD  ; Descobrir o numero de algoritmos da distancia MOV R2, M[Distancia] MOV R4, R0 INC R4 MOV R3, BASE_DEC DIV R2, R3 BR.NZ ED_C1  ; Escrever a distancia no LCD MOV R2, M[Distancia] ADD R1, R4 MOV R3, BASE_DEC DIV R2, R3 ADD R3, CH_ZERO MOV M[LCD_DISP_C], R1 MOV M[LCD_DISP_W], R3 DEC R1 CMP R2, R0 BR.NZ ED_C2  POP R4 POP R3 POP R2 POP R1 RET</pre>	



dez 02, 15 20:49	passaro.as	Page 9/17
	POP R2 POP R1 RET	
	<pre>; DescerPassaro: Rotina que desce o passaro conforme a acao da gravidade ; Entradas: M[PosPassaro], M[PosAPassaro], R5, R6 ; Saidas: M[PosPassaro], M[PosAPassaro] ; Efeitos: --- DescerPassaro:  PUSH R1                 PUSH R2                  MOV R1, M[PosPassaro]                 MOV R2, ULTIMA_LINHA                 SUB LINHA                 MOV M[PosAPassaro], R1                 CMP R1, R2 ; R1 &gt; R2 (ultima linha)                  CALL.P MataPassaro                 ; Y = y0 + v0t + (1/2)*g*t^2                 ; v0 = 0, y0 = R5, t = R6, g = GRAVIDADE                 ; R1 = GRAVIDADE * R6 * R6 / 2 + R5                 MOV R1, GRAVIDADE                 MOV R2, R6                 MUL R2, R1                 MOV R2, R6                 MOV R2, R1                 MUL R2, 1                 SHR R1, 1                 ADD R1, R5                 MOV M[PosPassaro], R1                  POP R2                 POP R1                 RET</pre>	
	<pre>; SubirPassaro: Rotina que sobe o passaro ; Entradas: --- ; Saidas: --- ; Efeitos: --- SubirPassaro:  PUSH R1                 PUSH R2                  MOV R6, R0 ;tempo inicial e zero                 MOV R1, M[PosPassaro]                 MOV M[PosAPassaro], R1;Guardar a posicao anterior do passaro                 MOV R2, NUM_L_SUBIR                 SHL R2, BYTE;Converter 000Xh em 0X00h estava na parte das                                 ;colunas e passa                                 a estar na parte das linhas                 SUB R1, R2                 SUB R1, M[LinhasASubir]                 CMP R1, LIM_TOPO ;Se Pos-NUM_L_SUBIR &lt; 0100h: Pos = 0100h                 BR.N SP_1                 MOV R5, R1 ;guardar a posicao da ultima vez que o passaro s                  SUB R2, LINHA                 ADD M[LinhasASubir], R2                 MOV R1, LINHA                 SUB M[PosPassaro], R1</pre>	

dez 02, 15 20:49	passaro.as	Page 10/17
	BR SP_Fim  SP_1:  MOV R1, LIM_TOPO MOV R5, R1 ; guardar a posicao da ultima vez que o passaro subiu  MOV M[PosPassaro], R1  SP_Fim:  POP R2 POP R1 RET	
	<pre>; VPosPassaro: Rotina que verifica se o passaro tem linhas a subir ou nao. ; Entradas: --- ; Saidas: --- ; Efeitos: --- VPosPassaro:  PUSH R1                  CMP M[LinhasASubir], R0                 BR.Z VPP_DescPass                 MOV R1, M[PosPassaro]                 MOV M[PosAPassaro], R1                 MOV R1, LINHA                 SUB M[PosPassaro], R1                 SUB M[LinhasASubir], R1                 BR VPP_Fim                  VPP_DescPass:  INC R6                                 DescerPassaro                                  VPP_Fim:  POP R1                                 RET</pre>	
	<pre>; ----- OBSTACULOS ----- ; DesX: Rotina que desenha um X e apaga o carater a direita. ; Entradas: R4 - Coordenada do ecrã ; Saidas: --- ; Efeitos: --- DesX:  PUSH R1          R4         INC R4         MOV M[TXT_DISPLAY_C], R4         MOV R1, CH_ESPACO         MOV M[TXT_DISPLAY_W], R1         DEC R4         ; Desenha um X         MOV M[TXT_DISPLAY_C], R4         MOV R1, CH_OBSTACULO         MOV M[TXT_DISPLAY_W], R1          POP R1         RET</pre>	
	<pre>; ApagaUltCol: Rotina que apaga a ultima coluna da esquerda da janela de texto ; Entradas: --- ; Saidas: --- ; Efeitos: Altera R4, M[TXT_DISPLAY_C], M[TXT_DISPLAY_W] ApagaUltCol:  PUSH R1 ; Coordenada atual                 PUSH R2 ; Ultima linha</pre>	

dez 02, 15 20:49	passaro.as		Page 11/17
AUC_C1:	<b>PUSH</b>	R3 ; Espaco	
	<b>MOV</b>	R1, LINHA	
	<b>MOV</b>	R2, ULTIMA_LINHA	
	<b>MOV</b>	R3, CH_ESPACO	
	<b>MOV</b>	M[TXT_DISPLAY_C], R1	
	<b>MOV</b>	M[TXT_DISPLAY_W], R3	
	<b>ADD</b>	R1, LINHA	
	<b>CMP</b>	R1, R2	
	<b>BR.NZ</b>	AUC_C1 ; Enquanto a linha nao for a ultima linha	
	<b>POP</b>	R3	
	<b>POP</b>	R2	
	<b>POP</b>	R1	
	<b>RET</b>		
	; DesObs: Rotina que desenha os obstaculos		
	; Entradas: ---		
	; Saidas: ----		
	; Efeitos: ---		
	<b>PUSH</b>	R1 ; Numero de obstaculos	
	<b>PUSH</b>	R2 ; Letra a desenhar	
	<b>PUSH</b>	R3 ; Ultima linha para desenhar obs	
	; ----- R4 : Posicao do cursor		
	<b>PUSH</b> R5 ; (registro auxiliar)		
	<b>PUSH</b>	R6 ; Endereco do vetor aleatorio	
	<b>PUSH</b>	R7 ; Numero aleatorio	
	<b>MOV</b> R1, M[NumObstaculos]		
	<b>CMP</b>	R1, R0 ;se o numero de obstaculos for 0,a rotina que	
	<b>JMP.Z</b>	DO_Fim ; desenha os obstaculos e passada a frente	
	<b>CALL</b>	ApagaUltCol	
	<b>MOV</b>	R3, ULTIMA_LINHA	
	<b>SUB</b>	R3, LINHA	
DO_C1:	<b>MOV</b>	R6, EspObstaculos	
	<b>MOV</b>	R4, DISPLAY_W	
	<b>DEC</b>	R4	
	<b>SUB</b>	R4, M[FaseObstaculos]	
	<b>MOV</b>	R7, M[R6]	
	<b>ADD</b>	R4, LINHA	
	<b>MOV</b>	R5, R4	
	<b>SHR</b>	R5, BYTE	
	<b>CMP</b>	R7, R5	
	<b>BR.NZ</b>	DO_NaoFazerEsp	
DO_NaoFazerEsp:	;Ate chegar ao numero aleatorio,escreve X.		
	;quando chegar, salta o n de posicoes definidas		
	;e continua a escrever X.		
	<b>ADD</b>	R4, ESPACO_OBS	
	DesX		
	<b>CMP</b>	R4, R3	
	<b>BR.N</b>	DO_C1 ; Cursor<Ultima linha	
	<b>INC</b>	R6	
	<b>MOV</b>	R7, M[R6]	
	<b>MVBH</b>	R4, R0 ; Colocar cursor na linha zero	
	<b>SUB</b>	R4, INTERVALO_OBS	
	<b>DEC</b>	R4	
	<b>BR.N</b>	DO_Fim ; Se a coluna for negativa (fora do ecra	
	)		
	<b>DEC</b>	R1	
	<b>BR.NZ</b>	DO_C1	

dez 02, 15 20:49	passaro.as		Page 12/17
DO_Fim:	<b>POP</b>	R7	
	<b>POP</b>	R6	
	<b>POP</b>	R5	
	<b>POP</b>	R3	
	<b>POP</b>	R2	
	<b>POP</b>	R1	
	<b>RET</b>		
	; DeslocaObs: Rotina que desloca todos os obstaculos para a esquerda		
	; Entradas: ---		
	; Saidas: ---		
DeslocaObs:	Efeitos: ---		
	<b>PUSH</b>	R1 ; Fase dos obstaculos	
	<b>PUSH</b>	R2	
	<b>MOV</b>	R1, M[FaseObstaculos]	
	<b>CMP</b>	R1, INTERVALO_OBS	
	<b>BR.NZ</b>	DO_1 ; (Se a fase nao	
	for a ultima)		
	<b>MOV</b>	M[FaseObstaculos], R0	
	<b>CALL</b>	InserVetObs	
DO_1:	<b>CALL</b>	IncNumObs	
	<b>BR</b>	DO_2	
	<b>INC</b>	M[FaseObstaculos]	
	<b>INC</b>	M[Distancia]	
	<b>CALL</b>	EscDistancia	
	<b>CALL</b>	EscIO	
	<b>POP</b>	R2	
	<b>POP</b>	R1	
	<b>RET</b>		
	; IncNumObs:Rotina que incrementa o numero de obstaculos se nao atingiu o maximo		
	; Entradas: M[NumObstaculos]		
	; Saidas: M[NumObstaculos]		
	; Efeitos: ---		
	<b>PUSH</b>	R1	
	<b>MOV</b>	R1, NUM_MAX_OBS	
	<b>CMP</b>	M[NumObstaculos], R1	
	<b>BR.Z</b>	INO_Skip	
	<b>INC</b>	M[NumObstaculos]	
INO_Skip:	<b>POP</b>	R1	
	<b>RET</b>		
	; InsereVetObs: Rotina que insere um novo elemento no vetor dos obstaculos		
	; Entradas: ---		
	; Saidas: ---		
	Efeitos: ---		
	<b>PUSH</b>	R1	
	<b>PUSH</b>	R2	
	<b>CALL</b>	MemObsFrente	
r de Ni	<b>CALL</b>	RandomN	
	<b>MOV</b>	R1, R3	
	; R1 toma o valo		
	; 2 <= N_ALEATORIO <= ULTIMA_LINHA - ESPACO_OBS - 2		
	; 0 <= N_ALEATORIO - 2 <= ULTIMA_LINHA - ESPACO_OBS - 4		
	<b>MOV</b>	R2, ULTIMA_LINHA	
	<b>SUB</b>	R2, ESPACO_OBS	
	<b>SHR</b>	R2, BYTE	



dez 02, 15 20:49	passaro.as	Page 15/17
	<pre>MOV R2, MAX_NIVEL CMP M[Nivel], R2 BR.Z DecNivel_Skip INC M[Nivel] SHL M[EstadoLEDs], 1 MOV R2, M[EstadoLEDs] MOV M[LED_DISPLAY], R2  DecNivel_Skip: POP R2 RET  ; IncNivel: Rotina que aumenta a dificuldade do jogo ; Entradas: --- ; Saídas: --- ; Efeitos: --- IncNivel: PUSH R2  MOV R2, 1 CMP M[Nivel], R2 BR.Z IncNivel_Skip DEC M[Nivel] SHRA M[EstadoLEDs], 1 MOV R2, M[EstadoLEDs] MOV M[LED_DISPLAY], R2 ; Ao diminuir o N, verificar que o contador nao ultrapassa N-1 MOV R2, M[Nivel] DEC R2 ; R2 = N - 1 CMP R1, R2 ; Contador &lt;= N - 1 BR.NP DecNivel_Skip MOV R1, R2 ; Se ultrapassar, Contador = N - 1  IncNivel_Skip: POP R2 RET  ; TempObstaculos: Rotina que trata da temporizacao dos obs conforme o nivel ; Entradas: --- ; Saídas: --- ; Efeitos: --- TempObstaculos: PUSH R2  MOV R2, M[Nivel] DEC R2 CMP R1, R2 BR.NZ TO_SkipObs ; Se contador estiver no maximo MOV R1, R0 ; Reset ao contador CALL DeslocaObs ; Execucao do deslocamento dos obstaculo  S CALL DesObs BR TO_Fim ; Caso contrario  TO_SkipObs: INC R1 TO_Fim: POP R2 RET  ; ----- INICIO ----- ; ----- ; Inicio do programa Inicio: MOV R1, SP_INICIAL MOV SP, R1 MOV R1, INT_MASK</pre>	

dez 02, 15 20:49	passaro.as	Page 16/17
	<pre>MOV M[INT_MASK_ADDR], R1 ; Inicializar variaveis do jogo MOV M[Distancia], R0 MOV R2, MAX_NIVEL MOV M[Nivel], R2 MOV R2, POSI_PASSARO MOV M[PosPassaro], R2 MOV M[PosAPassaro], R0 MOV M[LinhasASubir], R0 MOV R2, INTERVALO_OBS MOV M[FaseObstaculos], R2 MOV M[NumObstaculos], R0 MOV R2, BIT_MASK MOV M[EstadoLEDs], R2  ; Inicializar porto de controlo da janela de texto MOV R4, LIMPAR_JANELA MOV M[TEXT_DISPLAY_C], R4 CALL ApagaEcrã MOV R7, R0 MOV R3, R0 PUSH 12 PUSH MsgInicL1 CALL Esclinhac PUSH 14 PUSH MsgInicL2 CALL Esclinhac ; Inicializar variaveis do passaro MOV R5, M[PosPassaro] MOV R6, R0 MOV R2, BIT_MASK SHR R2, 2 ENI  Esperar_Inic: INC R3 TEST R7, R2 BR.Z Esperar_Inic SUB R7, R2 CALL ApagaEcrã ; Ativar temporizador OR R1, BIT_MASK MOV M[INT_MASK_ADDR], R1 ; Desenharm limites do jogo CALL TelaDeJogo ; Inicializar Temporizador CALL ResetTemp MOV R1, R0 ; Inicializar LEDs MOV R2, M[EstadoLEDs] MOV M[LED_DISPLAY], R2  Ciclo_1: ENI MOV R2, BIT_MASK DSI  ciclo_global ; As interrupcoes sao desligadas durante todo o ; para assegurar a ordem correta das operacoes. CMP M[PosAPassaro], R0 ; Se a posicao anterior for diferente de zero, ; significa que houve uma subida do passaro, ; logo tera que desenha-lo de novo.</pre>	



dez 02, 15 20:49	passaro.as		Page 17/17
	CALL	BR.Z ApagaPassaro	Ciclo_ITemp
	CALL	DesPassaro	
	MOV	M[PosAPassaro], R0	
Ciclo_ITemp:	TEST	R7, R2	
	BR.Z	Ciclo_Int0	
	CALL	ResetTemp	
	CALL	VPosPassaro	
	CALL	TempObstaculos	
	SUB	R7, R2	
Ciclo_Int0:	ROR	R2, 1	
	TEST	R7, R2	
	BR.Z	Ciclo_Int1	
	CALL	SubirPassaro	
	SUB	R7, R2	
Ciclo_Int1:	ROR	R2, 1	
	TEST	R7, R2	
	BR.Z	Ciclo_Int2	
	CALL	DecNivel	
	SUB	R7, R2	
Ciclo_Int2:	ROR	R2, 1	
	TEST	R7, R2	
	JMP.Z	Ciclo_1	
	CALL	IncNivel	
	SUB	R7, R2	
	JMP	Ciclo_1	