

```
1: 'Projeto SADA "Sistema de Automação da Disponibilização de Água"
2: 'Versão 3.2.11 - 18/12/2007 18:09:45
3:
4: 'Filippo Pardini - www.robotica.eng.br
5: '
6: '-----
6: 'Funcionamento do sistema SADA
7:
8: 'Subsistemas:
9:
10: ' 1. Sabesp (S)
11: '   Fornecimento público de água potável. O controle do fluxo de água é feito através de uma
12: '   eletroválvula (Vs) (220v).
13:
14: ' 2. Caixa inferior (Ci)
15: '   Caixa enterrada de 12.000 litros que tem por finalidade manter um estoque de segurança de
16: '   água que é coletada do sistema Sabesp ao nível do chão, portanto com pressão mínima.
17: '   A funcionalidade desta caixa é composta por uma bóia mecânica para caixa cheia, uma bomba
18: '   vibratória (B) (220v ; 1000 l/h), um sensor (NCi) de nível mínimo de água para proteção da
19: '   bomba e uma eletroválvula (Vb) (220v) na saída da bomba para controle do fluxo de água.
20:
21: ' 3. Caixa superior (Cs)
22: '   Caixa de 2.000 litros sob o telhado da casa que serve a casa toda através de um sistema de
23: '   pressurização. Esta caixa recebe água diretamente da Sabesp ou da caixa inferior (Ci) através
24: '   da bomba (B). A funcionalidade desta caixa é composta por um "ladrão" para segurança, por um
25: '   sensor de entrada de água (Fa), por um sensor de nível máximo (N5), por um sensor para
26: '   acionamento da entrada de água (N4) e por um par de sensores de nível mínimo de água (N2 e N3)
27: '   para proteção do subsistema de pressurização.
28:
29: ' 4. Pressurização (Pr)
30: '   Subsistema (Dancor) composto por uma bomba (220v), um tanque de 100 litros e um pressostato
31: '   para controle da pressão da água nas tubulações da casa. Este subsistema mantém automaticamente
32: '   uma pressão de 40 psi na tubulação, através da bomba controlada pelo pressostato.
33:
34: 'Funcionamento:
35:
36: ' Em condições normais (existência de água da Sabesp) a Cs estará oscilando entre os níveis N4 e
37: ' N5 sendo alimentada diretamente pela Sabesp, o pressurizador estará desbloqueado alimentando a
38: ' casa e a Ci estará cheia, controlada pela boia mecânica. Quando o nível cai abaixo de N4, a
39: ' eletroválvula da Sabesp (Vs) é acionada, permitindo a entrada de água. Quando o nível N5 é
```

40: ' atingido, Vs é desligada bloqueando a entrada de água.
41:
42: ' No caso de falta de água da Sabesp, quando o nível em Cs cai abaixo de N4 ela passa
43: ' automaticamente a ser alimentada pela Ci através da bomba (B) iniciando um ciclo que somente
44: ' reverterá para a Sabesp (se o fornecimento voltar) após Cs atingir o nível N5. Se continuar
45: ' faltando a água da Sabesp, a Cs continuará oscilando entre os níveis N4 e N5 mas agora alimentada
46: ' pela Ci através da bomba (B) e da eletroválvula (Vb).
47:
48: ' Independentemente da alimentação da Cs (Sabesp ou Ci), caso o nível em Cs caia abaixo de N2,
49: ' o pressurizador é bloqueado para proteger sua bomba. Quando o nível volta acima de N3,
50: ' o pressurizador é desbloqueado.
51:
52: ' Quando a Cs estiver sendo alimentada pela Ci através da bomba (B), o nível da água em Ci começa
53: ' a baixar (porque não há entrada de água da Sabesp, caso contrário a alimentação de Cs teria
54: ' revertido para a Sabesp) e poderá chegar a ser menor que N_{Ci}. Neste momento, a bomba (B) é
55: ' bloqueada para proteção. Com a volta da água da Sabesp, e o nível em Ci acima de N_{Ci}, a bomba é
56: ' desbloqueada, o sistema reverte a alimentação de Cs para a Sabesp e Ci passa a ser alimentada
57: ' pela Sabesp até ficar cheia e a boia mecânica ser acionada.
58:
59: ' Quando a alimentação de Cs estiver sendo feita pela Ci através da bomba (B) e o sistema perceber
60: ' que a bomba não está mais funcionando (ou queimada), é dado um alarme visual e sonoro para avisar
61: ' da falha da bomba. Apesar disso, o sistema não é bloqueado de modo que se nesse ínterim a água
62: ' da Sabesp voltar, a alimentação de Cs é revertida para a Sabesp e o sistema volta ao normal
63: ' porém com a bomba bloqueada e com alarme de bomba inoperante. Após a troca da bomba, o sistema
64: ' precisa ser reinicializado para incorporar a bomba (B).
65:
66: ' Em alguns casos detectáveis de falha nos sensores de nível, o sistema se inibe bloqueando tudo
67: ' e dando alarme visual e sonoro de sistema em colapso.
68:
69: ' Os elementos de interação com o sistema são:
70: ' Al (Al) - Alarme sonoro
71: ' Led1 (L1) - Led vermelho
72: ' Led2 (L2) - Led amarelo
73: ' bit0 (L5) - Led verde
74: ' bit1 (L4) - Led verde
75: ' bit2 (L3) - Led verde
76: ' (S1) - Botão para o acionamento do interrupt (ver descrição na rotina Interrupt)
77: ' S2 (S2) - Botão para acionar o display do Status via os leds (ver descrição na rotina Mostra)
78: ' (S3) - Botão para o reset de hardware (reinicializa o microcontrolador)

```
79: ' S4 (S4) - Botão para silenciar o alarme sonoro
80: ' (R4) - Trimpot para ajuste da voltagem do ADC. Cuidado! ele deverá ser ajustado para uma leitura
81: ' de 5VDC em JP1 correspondendo ao fim de escala (ponto mais alto) do sensor de Cs.
82:
83: ' O display das variáveis de estado do sistema nos permitirá analisar o funcionamento do sistema,
84: ' especialmente quando for sinalizado algum problema.
85: '
86: '
87: 'Variaveis e constantes
88: Public Const CCS As Byte = 15 ' Pino de seleção para ADC0831
89: Public Const Clock As Byte = 16 ' Pino de clock para o ADC0831
90: Public Const In_D As Byte = 17 ' Pino de dados para o ADC0831
91: Public Const N6 As Byte = 21 ' Pino para leitura de Nci
92: Public Const NFa As Byte = 22 ' Pino para leitura de Fa
93: Public Const PVb As Byte = 23 ' Pino para acionamento da eletrovalvula da bomba
94: Public Const PVs As Byte = 24 ' Pino para acionamento da eletrovalvula da Sabesp
95: Public Const PBp As Byte = 25 ' Pino para bloqueio/desbloqueio do pressurizador
96: Public Const PBb As Byte = 26 ' Pino para bloqueio/desbloqueio da bomba
97: Public Const PB As Byte = 27 ' Pino para ligar/desligar a bomba
98: Public Const Led1 As Byte = 28 ' Pino para acionar o Led1. Problema nos sensores ou placa
99: Public Const Led2 As Byte = 32 ' Pino para acionar o Led2. Bomba queimada
100: Public Const Al As Byte = 33 ' Pino para acionar o alarme
101: Public Const S4 As Byte = 38 ' Pino para desligar o alarme
102: Public Const S2 As Byte = 37 ' Pino para mostrar status
103: Public Const bit0 As Byte = 34 ' Pino para valor 1 - L5
104: Public Const bit1 As Byte = 35 ' Pino para valor 2 - L4
105: Public Const bit2 As Byte = 36 ' Pino para valor 4 - L3
106: Public Const um As Byte = bx00000001 ' Valor 1
107: Public Const dois As Byte = bx00000010 ' Valor 2
108: Public Const quatro As Byte = bx00000100 ' Valor 4
109: Public Vs As Byte ' Variável de estado da eletrovalvula da Sabesp
110: Public Ea As Byte ' Variável de estado da água (Sabesp ou bomba)
111: Public Eb As Byte ' Variável de estado da bomba (queimada ou bom estado)
112: Public B As Byte ' Variável de estado da bomba (ligada ou desligada)
113: Public Bp As Byte ' Variável de estado do pressurizador (bloqueado ou não)
114: Public Bb As Byte ' Variável de estado da bomba (bloqueada ou não)
115: Public Fa As Byte ' Variável de estado do sensor de fluxo de água
116: Public Cs As Byte ' Variável de estado da caixa superior
117: Public Ci As Byte ' Variável de estado da caixa inferior
```

```
118: Public AgS As Byte ' Variável para indicar se tem que aguardar Sabesp
119: '
120: Public Sub Main()
121: 'Lógica principal
122:
123: Dim I As Integer ' Contador
124: Dim N As Integer ' Contador
125: Dim First As Boolean ' Indicador de primeira vêz
126: Dim T1 As Integer ' Número de minutos que deve durar o ciclo interno
127: Dim T2 As Integer ' Intervalo em minutos para verificação da volta da Sabesp
128: Dim Tc As Integer ' Ciclos necessários para completar T1 - T1*20
129: Dim Ts As Integer ' Ciclos necessários para completar T2 - T2/T1
130:
131: Sleep 1.0 ' Ajuda download
132:
133: '*****
134: Do ' Repete eternamente. Este ciclo somente executa a inicialização, ou no início do programa ou
135: ' em casos exepcionais
136: '*****
137:
138: If Not Inicialize() Then ' Tenta inicializar o sistema
139: 'Se não conseguir
140: Interrupt(2) ' Trata o êrro e aguarda S1
141: LigaTudo ' Liga todos os leds e o alarme
142: Do ' Fica aqui eternamente
143: Delay 5.0
144: Loop
145: End If
146:
147: First = True ' Primeira vez que vai executar o ciclo interno
148: T1 = 5 ' Aqui define o número de minutos do For de espera
149: T2 = 5 ' Aqui define o número de minutos de intervalo para verificação da Sabesp
150: Tc = T1 * 20 ' Ciclos necessários para completar T1
151: Ts = FixI(CSng(T2)/CSng(T1)) - 1 ' Ciclos necessários para completar T2
152:
153: '*****
154: Do ' Repete eternamente. Este ciclo é repetido a cada T1 minutos, sendo que aproximadamente 10
155: ' segundos são utilizados para a leitura do sensor Cs (via ADC com tratamento estatístico).
156: ' No restante do tempo, é executado um loop que só mostra condições existentes via leds e
```

```
157:      ' permite, se desejado, desligar o eventual alarme sonoro (via S4) e mostrar as variáveis
158:      ' de estado (via S2).
159:      '*****
160:
161:      ' Lê sensores e calcula Ci, Cs e Fa
162:      Ci = GetNCi() ' Variável de estado da caixa inferior
163:      Fa = GetFa()  ' Variável de estado do sensor de fluxo de água
164:      Cs = GetCs()  ' Variável de estado da caixa superior - leva ~5 segundos
165:
166:      If VerErros1() Then ' Verifica se tem erros de leitura nos sensores
167:          Interrupt(1)  ' Trata o erro e aguarda S1
168:          LigaTudo 'Liga todos os leds e o alarme
169:          Do ' Fica aqui eternamente
170:              Delay 5.0
171:          Loop
172:      End If
173:
174:      If VerErros2() Then ' Verifica se tem erros de incongruências operacionais
175:          Interrupt(4)  ' Trata o erro e aguarda S1
176:          LigaTudo 'Liga todos os leds e o alarme
177:          Do ' Fica aqui eternamente
178:              Delay 5.0
179:          Loop
180:      End If
181:
182:      ' Analisa Ci
183:      Select Case Ci
184:          Case 1 ' Caixa inferior vazia
185:              BloqueiaB ' Bloqueia bomba
186:          Case 2 ' Caixa inferior com água
187:              DesbloqueiaB ' Desbloqueia bomba
188:      End Select
189:
190:      '*****
191:      If Ags = 0 Then ' Não tem que aguardar Sabesp
192:          '*****
193:
194:          ' Analisa Cs
195:          Select Case Cs
```

```
196:      Case 1 ' Caixa cheia
197:         DesbloqueiaP ' Desbloqueia pressurizador
198:         DesligaB ' Desliga bomba
199:         DesligaS ' Desliga Sabesp
200:         Ea = 0 ' Predispõe para Sabesp
201:      Case 2 ' Caixa quase cheia
202:         DesbloqueiaP ' Desbloqueia pressurizador
203:      Case 3 ' Caixa precisando de água
204:         DesbloqueiaP ' Desbloqueia pressurizador
205:         PrecisaAgua ' Verifica quem vai fornecer água e aciona
206:      Case 4 ' Caixa Vazia
207:         BloqueiaP ' Bloqueia pressurizador
208:         PrecisaAgua ' Verifica quem vai fornecer água e aciona
209:      Case 5 ' Caixa em processo de enchimento
210:         DesbloqueiaP ' Desbloqueia pressurizador
211:      Case Else ' Problema nos sensores
212:         Interrupt(3) ' Erro Leitura Cs. Trata o erro e aguarda S1
213:         LigaTudo 'Liga todos os leds e o alarme
214:         Do ' Fica aqui eternamente
215:             DesligaAlarme ' Verifica se é para desligar o alarme
216:             MostraStatus(1) ' Verifica se é para mostrar Status
217:         Loop
218:     End Select
219:     '*****
220: Else ' Tem que aguardar Sabesp
221:     '*****
222:
223:     ' Verifica Cs
224:     Select Case Cs
225:         Case 3 ' Caixa precisando de água
226:             DesbloqueiaP ' Desbloqueia pressurizador
227:         Case 4 ' Caixa Vazia
228:             BloqueiaP ' Bloqueia pressurizador
229:         Case Else ' Problema nos sensores
230:             Interrupt(3) ' Erro Leitura Cs. Trata interrupt e aguarda S1
231:             LigaTudo 'Liga todos os leds e o alarme
232:             Do ' Fica aqui eternamente
233:                 Delay 5.0
234:             Loop
```

```
235:      End Select
236:
237:      ' Verifica se passou o tempo programado para ver se voltou a Sabesp
238:      If N >= Ts Then ' Se passou o intervalo, verifica se voltou a Sabesp
239:          N = 0
240:          If LigaS() Then ' Ligou Sabesp e tem fluxo de água
241:              Ags = 0 ' Não aguarda Sabesp
242:          End If
243:      End If
244:
245:      '*****
246:      End If
247:      '*****
248:
249:      ' Ciclo para aguardar um certo tempo, mostrando certas ocorrências via leds e permitindo
250:      ' ações via S2 (mostra status) e S4 (desliga alarme). Cada ciclo é de aproximadamente de
251:      ' 3 segundos, portanto para conseguir um tempo de T1 minutos, o ciclo deverá ser repetido
252:      ' T1*60/3 => T1*20
253:      For I = 1 To Tc ' Ciclo que deverá durar T1 minutos
254:          DesligaAlarme ' Verifica se é para desligar o alarme
255:          MostraStatus(0) ' Verifica se é para mostrar Status
256:          If Ags = 1 Then ' Tem que esperar Sabesp
257:              Pisca 1, 1 ' Pisca led vermelho para indicar que está aguardando a Sabesp
258:              If Eb = 0 Then ' Bomba queimada, pisca L5
259:                  Pisca 3, 1 ' Pisca led amarelo para indicar que está em ciclo normal
260:                  Pisca 7, 1 ' Pisca led verde(L5) para indicar bomba queimada
261:              Else ' Bomba bloqueada
262:                  Pisca 3, 1 ' Pisca led amarelo para indicar que está em ciclo normal
263:                  Pisca 8, 1 ' Pisca led verde(L4) para indicar bomba bloqueada
264:              End If
265:          Else ' Não tem que esperar Sabesp
266:              Pisca 3, 3 ' Pisca led amarelo para indicar que está em ciclo normal
267:          End If
268:      Next
269:
270:      ' Verifica se foi o primeiro ciclo
271:      If First Then
272:          N = 0
273:          First = False
```

```
274:     else
275:         N = N + 1
276:     End If
277:
278:     '*****
279:     Loop ' Repete a lógica
280:     '*****
281:
282:     '*****
283:     Loop ' Reinicializa
284:     '*****
285:
286: End Sub
287: '
288: Sub PrecisaAgua()
289: 'Rotina para decidir fonte de água
290:
291: If Ea = 0 Then ' Predisposição para Sabesp
292:     If LigaS() Then ' Ligou Sabesp e tem fluxo de água
293:         AgS = 0 ' Não precisa aguardar Sabesp
294:         Exit Sub
295:     End If
296:     ' Tentou ligar Sabesp e não tem fluxo de água, vamos tentar a bomba
297:     Ea = 1 ' Variável de estado que indica na Bomba
298: End If
299:
300: ' Predisposição para bomba
301: If Eb = 1 Then ' Bomba em bom estado
302:     If Bb = 0 Then ' Bomba não bloqueada
303:         If LigaB() Then ' Ligou Bomba e tem fluxo de água
304:             AgS = 0 ' Não precisa aguardar Sabesp
305:             Exit Sub
306:         Else ' Tentou ligar bomba e não tem fluxo de água
307:             Eb = 0 ' Acabou de queimar a bomba
308:             PutPin Al, 1 ' Liga alarme
309:             AgS = 1 ' Precisa aguardar Sabesp
310:             Ea = 0 ' Predispõe para Sabesp
311:         End If
312:     Else ' Bomba bloqueada
```



```
313:     AgS = 1 ' Precisa aguardar Sabesp
314:     Ea = 0 ' Predisposição para Sabesp
315:     Exit Sub
316: End If
317: Else ' Bomba já queimada
318:     Exit Sub
319: End If
320: End Sub
321: '
322: Function GetNCs() As Byte
323: 'Função para leitura do sensor de nível na caixa superior,
324: 'via ADC0831
325:
326:     Dim N as Integer ' Contador
327:     Dim ADCbit as Byte ' Bit lido
328:
329:     PutPin CCS, 1 ' ADC0831 não selecionado
330:     GetNCs = 0 ' Limpa campo
331:     PutPin In_D, 3 ' Pino In_D como entrada
332:     PutPin Clock, 0 ' Clock baixo
333:     PutPin CCS, 0 ' Seleciona ADC0831
334:     For N = 1 TO 8 ' Lê os 8 bits de dados
335:         GetNCs = GetNCs * 2 ' Shift para a esquerda
336:         PulseOut Clock, 0.00001, 1 ' Pulsa Clock com 10us
337:         ADCbit = GetPin (In_D) ' Lê o bit
338:         GetNCs = GetNCs + ADCbit ' Adiciona o novo bit
339:     Next
340:     PutPin CCS, 1 ' ADC0831 não selecionado
341:     GetNCs = GetNCs * 2 ' Shift para a esquerda
342: End Function
343: '
344: Function GetCs() As Byte
345: 'Função para o cálculo do estado da caixa superior, em função das leituras do sensor de nível e do
346: 'sensor de fluxo de água. São feitas 10 leituras via ADC e eliminadas as que ultrapassam + ou - 10%
347: 'da média. o valor utilizado é a média das restantes.
348:
349:     Dim I As Byte ' Contador
350:     Dim N As Byte ' Contador
351:     Dim XX As Integer ' Acumulador de leituras válidas
```

```
352: Dim X As Byte ' Valor a ser considerado
353: Dim Xmed As Integer ' Valor médio
354: Dim Xmed10 As Integer ' 10% da média
355: Dim Xi(1 To 10) As Integer ' Vetor das leituras
356: Const N1 As Byte = 51 ' Nível abaixo do qual é considerado erro
357: Const N2 As Byte = 81 ' Nível de bloqueio do pressurizador
358: Const N3 As Byte = 92 ' Nível de desbloqueio do pressurizador
359: Const N4 As Byte = 162 ' Nível para acionar fornecedor de água
360: Const N5 As Byte = 228 ' Nível para desacionar fornecedor de água
361:
362: Xmed = 0 ' Inicializa valor médio
363:
364: For I = 1 To 10 ' Lê 10 valores do sensor com intervalo de 1 segundo
365:     Xi(I) = CInt(GetNCs()) ' Lê byte e transforma para integer
366:     Xmed = Xmed + Xi(I) ' Acumula em integer
367:     Delay 1.0 ' Aguarda 1 segundo
368: Next
369:
370: Xmed = FixI(CSng(Xmed)/10.0) ' Calcula a média dos 5 valores como integer
371: Xmed10 = FixI(CSng(Xmed)/10.0) ' Calcula 10% da média
372: N = 0 ' Inicializa contador de leituras válidas
373: XX = 0 ' Inicializa acumulador de leituras válidas
374:
375: For I = 1 To 10 ' Verifica quais dos 10 valores estão dentro de um desvio de 10% da média
376:     If Not (Abs(Xi(I) - Xmed) > Xmed10) Then ' Se o desvio for menor que 10%
377:         N = N + 1 ' Conta a leitura
378:         XX = XX + Xi(I) ' Acumula a leitura
379:     End If
380: Next
381:
382: If N = 0 Then ' Problemas!
383:     Interrupt(5) ' Trata erro e aguarda S1
384:     LigaTudo 'Liga todos os leds e o alarme
385:     Do ' Fica aqui eternamente
386:         Delay 5.0
387:     Loop
388: End If
389:
390: X = FixB(CSng(XX)/CSng(N)) ' Calcula a média das leituras válidas como byte
```

```
391:
392: If (X >= N5) And (X <= 255) Then
393:   X = 1 ' Caixa cheia
394: ElseIf X > 255 Then
395:   X = 6 ' Erro
396: Else
397:   If Fa = 0 Then ' Caixa em estado de esvaziamento
398:     If (X < N5) And (X >= N4) Then
399:       X = 2 ' Caixa quase cheia
400:     ElseIf (X < N4) And (X >= N2) Then
401:       X = 3 ' Caixa precisando de água
402:     ElseIf ((X < N2) And (X >= N1)) Or ((X < (N2 + 5)) And (X >= N1)) Then
403:       X = 4 ' Caixa vazia
404:     Else
405:       X = 6 ' Erro
406:     End If
407:   ElseIf Fa = 1 Then ' Caixa em estado de enchimento
408:     If (X < N5) And (X >= N3) Then
409:       X = 5 ' Caixa em estado de enchimento
410:     ElseIf (X < N3) And (X >= N1) Or ((X < (N3 + 5)) And (X >= N1)) Then
411:       X = 4 ' Caixa vazia
412:     Else
413:       X = 6 ' Erro
414:     End If
415:   Else
416:     X = 6 ' Erro
417:   End If
418: End If
419: GetCs = X ' Devolve valor da leitura
420: End Function
421: '
422: Function GetNCi() As Byte
423: 'Função para o calculo do estado da caixa inferior
424: 'GetNCi = 1 Caixa vazia, bloqueia bomba (Nci = 0)
425: 'GetNCi = 2 Caixa com água, desbloqueia bomba (NCi = 1)
426:
427: Dim X As Byte
428:
429: PutPin N6, 3 ' Pino N6 como entrada
```

```
430: X = GetPin (N6)
431: If X = 1 Then
432:   GetNCi = 2
433: Else
434:   GetNCi = 1
435: End If
436: End Function
437: '
438: Function GetFa() As Byte
439: 'Função para leitura do sensor de fluxo de água
440:
441:   Dim X As Byte
442:
443:   PutPin NFa, 3 ' Pino NFa como entrada
444:   X = GetPin (NFa)
445:   If X = 0 Then
446:     GetFa = 1 ' Existe fluxo de água
447:   Else
448:     GetFa = 0 ' Não existe fluxo de água
449:   End If
450: End Function
451: '
452: Function Inicialize() As Boolean
453: 'Inicialização do sistema
454:
455:   Inicialize = True
456:
457:   ' Desliga todos os leds e alarme
458:   DesligaTudo
459:   Fa = 0 ' Sem fluxo de água (válvulas fechadas)
460:   Eb = 1 ' Bomba virtualmente em bom estado
461:   Ea = 0 ' Predispõe para Sabesp
462:   AgS = 0 ' Não tem que aguardar Sabesp
463:
464:   B = 1 ' Bomba virtualmente ligada
465:   DesligaB ' Desliga bomba
466:   Vs = 1 ' Válvula da Sabesp virtualmente ligada
467:   DesligaS ' Desliga Sabesp
468:
```

```
469: ' Lê sensores e calcula Ci e Cs
470: Ci = GetNCi() ' Variável de estado da caixa inferior
471: Cs = GetCs() ' Variável de estado da caixa superior
472:
473: ' Verifica Ci
474: Select Case Ci
475:     Case 1 ' Caixa inferior vazia, Bloqueia bomba
476:         Bb = 0 ' Bloqueio da bomba virtualmente desligado
477:         BloqueiaB ' Bloqueia bomba
478:     Case 2 ' Caixa inferior com água, Desbloqueia bomba
479:         Bb = 1 ' Bloqueio da bomba virtualmente ligado
480:         DesbloqueiaB ' Desbloqueia bomba
481: End Select
482:
483: ' Verifica Cs
484: Select Case Cs
485:     Case 1,2,3 ' Caixa cheia, quase cheia ou precisando de água
486:         Bp = 1 ' Bloqueio do pressurizador virtualmente ligado
487:         DesbloqueiaP ' Desbloqueia pressurizador
488:     Case 4 ' Caixa vazia
489:         Bp = 0 ' Bloqueio do pressurizador virtualmente desligado
490:         BloqueiaP ' Bloqueia pressurizador
491:     Case Else ' Problema nos sensores
492:         Inicialize = False
493:         Exit Function
494: End Select
495:
496: If VerErros1() Then ' Verifica êrros
497:     Inicialize = False
498:     Exit Function
499: End If
500:
501: If VerErros2() Then ' Verifica êrros
502:     Inicialize = False
503:     Exit Function
504: End If
505:
506: Pisca 6,3 ' Mostra que inicializou piscando os leds 1 e 2 alternadamente
507: End Function
```

```
508: '

---

509: Function LigaB() As Boolean  
510: 'Rotina para ligar a bomba  
511:  
512: If B = 0 Then 'Bomba desligada, liga e verifica chegada de água  
513:     PutPin PVb, 1 'Aciona relé que liga valvula da bomba  
514:     Delay 3.0 'Dá um tempo para ligar a bomba  
515:     PutPin PB, 1 'Aciona relé que liga bomba  
516:     PutPin bit0, 1 'liga Led5 - bomba ligada  
517:     Ea = 1 'Variável de estado que indica na bomba  
518:     B = 1 'Variável de estado que indica bomba ligada  
519:     Delay 10.0 'Aguarda subir água  
520: End If  
521: 'Bomba ligada, verifica chegada de água  
522: Fa = GetFa()  
523: If Fa = 1 Then 'Tem água  
524:     LigaB = True  
525: Else 'Não tem água  
526:     LigaB = False  
527:     PutPin PB, 0 'Aciona relé que desliga bomba  
528:     Delay 3.0 'Dá um tempo para desligar a valvula da bomba  
529:     PutPin PVb, 0 'Aciona relé que desliga valvula da bomba  
530:     PutPin bit0, 0 'Desliga Led 5 - bomba desligada  
531:     B = 0 'Variável de estado que indica bomba desligada  
532: 'Ea = 0 Variável de estado que indica na Sabesp  
533: 'DesligaB  
534: End If  
535: End Function  
536: '

---

537: Sub DesligaB()  
538: 'Rotina para desligar bomba  
539:  
540: If B = 1 Then 'Bomba ligada, desliga  
541:     PutPin PB, 0 'Aciona relé que desliga bomba  
542:     Delay 3.0 'Dá um tempo para desligar a valvula da bomba  
543:     PutPin PVb, 0 'Aciona relé que desliga valvula da bomba  
544:     PutPin bit0, 0 'Desliga Led 5 - bomba desligada  
545:     B = 0 'Variável de estado que indica bomba desligada  
546: End If
```

```
547: End Sub
548: '
549: Function LigaS() As Boolean
550: 'Rotina para ligar Sabesp
551:
552: If Vs = 0 Then ' Sabesp desligada, liga e verifica chegada de água
553: PutPin PVs, 1 ' Aciona relé que liga válvula da Sabesp
554: PutPin bit1, 1 ' Liga Led4 - Sabesp ligada
555: Ea = 0 ' Variável de estado que indica na Sabesp
556: Vs = 1 ' Variável de estado que indica válvula da Sabesp aberta
557: Delay 10.0 ' Aguarda subir água
558: End If
559: ' Sabesp ligada, verifica chegada de água
560: Fa = GetFa()
561: If Fa = 1 Then ' Tem água
562: LigaS = True
563: Else ' Não tem água
564: LigaS = False
565: PutPin PVs, 0 ' Aciona relé que desliga válvula da Sabesp
566: PutPin bit1, 0 ' Desliga Led4 - Sabesp desligada
567: Vs = 0 ' Variável de estado que indica válvula da Sabesp fechada
568: ' Ea = 1 ' Variável de estado que indica na Bomba
569: ' DesligaS
570: End If
571: End Function
572: '
573: Sub DesligaS()
574: 'Rotina para desligar Sabesp
575:
576: If Vs = 1 Then ' Sabesp ligada, desliga
577: PutPin PVs, 0 ' Aciona relé que desliga válvula da Sabesp
578: PutPin bit1, 0 ' Desliga Led4 - Sabesp desligada
579: Vs = 0 ' Variável de estado que indica válvula da Sabesp fechada
580: End If
581: End Sub
582: '
583: Sub BloqueiaP()
584: 'Rotina para bloquear pressurizador
585:
```

```
586:   If Bp = 0 Then ' Pressurizador desbloqueado, bloqueia
587:     PutPin PBp, 1 ' Aciona relé que bloqueia pressurizador
588:     PutPin bit2, 1 ' Liga Led3 - pressurizador bloqueado
589:     Bp = 1 ' Variável de estado que indica bloqueio do pressurizador
590:   End If
591: End Sub
592: '
593: Sub DesbloqueiaP()
594: 'Rotina para desbloquear pressurizador
595:
596:   If Bp = 1 Then ' Pressurizador bloqueado, desbloqueia
597:     PutPin PBp, 0 ' Aciona relé que desbloqueia pressurizador
598:     PutPin bit2, 0 ' Desliga Led3 - pressurizador desbloqueado
599:     Bp = 0 ' Variável de estado que indica desbloqueio do pressurizador
600:   End If
601: End Sub
602: '
603: Sub BloqueiaB()
604: 'Rotina para bloquear bomba
605:
606:   If Bb = 0 Then ' Bomba desbloqueada, bloqueia
607:     PutPin PBb, 0 ' Aciona relé que bloqueia bomba
608:     Bb = 1 ' Variável de estado que indica bloqueio da bomba
609:   End If
610: End Sub
611: '
612: Sub DesbloqueiaB()
613: 'Rotina para desbloquear bomba
614:
615:   If Bb = 1 Then ' bomba bloqueada, desbloqueia
616:     PutPin PBb, 1 ' Aciona relé que desbloqueia bomba
617:     Bb = 0 ' Variável de estado que indica desbloqueio da bomba
618:   End If
619: End Sub
620: '
621: Sub Interrupt(ByVal Id As Byte)
622: 'O registro de status será gravado na eeprom a partir do endereço 20000
623: 'e o microcontrolador aguardará o interrupt para ser rebootado.
624: 'Id = 1 Erro nos sensores
```



```
625: 'Id = 2  Erro na inicialização
626: 'Id = 3  Erro de leitura Cs
627: 'Id = 4  Erro incongruência operacional
628: 'Id = 5  Erro em GetCs
629:
630:   PutPin Al, 1  ' Liga alarme
631:
632:   'Grava Status na EEPROM
633:   PutEEPROM 20000,Id,1
634:   PutEEPROM 20001,Vs,1
635:   PutEEPROM 20002,Ea,1
636:   PutEEPROM 20003,Eb,1
637:   PutEEPROM 20004,B,1
638:   PutEEPROM 20005,Bp,1
639:   PutEEPROM 20006,Bb,1
640:   PutEEPROM 20007,Fa,1
641:   PutEEPROM 20008,Cs,1
642:   PutEEPROM 20009,Ci,1
643:   PutEEPROM 200010,AgS,1
644:
645:   Do  ' Fica aqui eternamente, só sai com S1
646:     PutPin Led1, 1  ' Liga Led1 vermelho
647:     DesligaAlarme  ' Verifica se é para desligar o alarme - S4
648:     MostraStatus(1) ' Verifica se é para mostrar Status - S2
649:     If Sai() Then  ' Verifica se é para sair - S1
650:       Exit Do
651:     End If
652:   Loop
653: End Sub
654: '
655: Sub Mostra(ByVal Ch As Byte)
656: 'Rotina para mostrar valores via leds.
657: 'Usa os leds L3,L4 e L5. L3 indica o valor 1, L4 o valor 2 e L5 o valor 4, ou seja, os 3 leds
658: 'indicam o valor em binário ( de 0 a 7).
659: 'Enquanto o led amarelo (L2) estiver ligado (5 segundos), o valor nos leds numéricos
660: 'é valido. Na mudança de valor, o led vermelho acende e qualquer valor
661: 'mostrado nos leds numéricos não é valido. O Status é sempre mostrado na ordem:
662: 'Id,Vs,Ea,Eb,B,Bp,Bb,Fa,Cs,Ci,AgS
663:
```

```
664: Dim Val(1 To 11) As Byte ' Vetor de variáveis
665: Dim I As Byte ' Contador
666: Dim Id As Byte ' Identificador do interrupt
667:
668: If Ch = 1 Then ' Le dados da EEPROM
669:   GetEEPROM 30000,Id,1
670:   GetEEPROM 30001,Vs,1
671:   GetEEPROM 30002,Ea,1
672:   GetEEPROM 30003,Eb,1
673:   GetEEPROM 30004,B,1
674:   GetEEPROM 30005,Bp,1
675:   GetEEPROM 30006,Bb,1
676:   GetEEPROM 30007,Fa,1
677:   GetEEPROM 30008,Cs,1
678:   GetEEPROM 30009,Ci,1
679:   GetEEPROM 300010,AgS,1
680: Else ' Usa dados da memória
681:   Id = 0 ' Não teve interrupt
682: End If
683:
684: Val(1) = Id
685: Val(2) = Vs
686: Val(3) = Ea
687: Val(4) = Eb
688: Val(5) = B
689: Val(6) = Bp
690: Val(7) = Bb
691: Val(8) = Fa
692: Val(9) = Cs
693: Val(10) = Ci
694: Val(11) = AgS
695:
696: DesligaTudo ' Desliga todos os leds e o alarme
697:
698: For I=1 To 11
699:   PutPin Led1, 1 ' Liga led vermelho
700:   PutPin bit2, 0 ' Desliga L3
701:   PutPin bit1, 0 ' Desliga L4
702:   PutPin bit0, 0 ' Desliga L5
```

```
703:      ' Ve quais os leds numéricos que tem que ligar
704:      If (Val(I) And um) = 1 Then
705:          PutPin bit0, 1 ' Liga L5
706:      End If
707:      If (Val(I) And dois) = 2 Then
708:          PutPin bit1, 1 ' Liga L4
709:      End If
710:      If (Val(I) And quatro) = 4 Then
711:          PutPin bit2, 1 ' Liga L3
712:      End If
713:      Delay 2.0 ' Sinaliza mudança de variável
714:      PutPin Led1, 0 ' Desliga led vermelho
715:      PutPin Led2, 1 ' Liga led amarelo
716:      Delay 5.0 ' Aguarda 5 segundos
717:      PutPin Led2, 0 ' Desliga led amarelo
718:  Next
719:
720:  DesligaTudo ' Desliga todos os leds e o alarme
721: End Sub
722: '
723: Function VerErros1() As Boolean
724: 'Rotina para verificar êrro nos sensores
725:
726:  Dim Cod As Integer
727:
728:  VerErros1 = False
729:
730:  ' Verifica êrro nos sensores
731:  Cod = CInt(Ci) * 100 + CInt(Fa) * 10 + CInt(Cs)
732:  If (Cod = 105) OR (Cod = 205) OR (Cod > 216) OR (Cod < 101) OR ((Cod > 106) And (Cod < 111)) _
733:  OR ((Cod > 116) And (Cod < 201)) OR ((Cod > 206) And (Cod < 211)) Then
734:      VerErros1 = True
735:  End If
736: End Function
737: '
738: Function VerErros2() As Boolean
739: 'Rotina para verificar impssibilidades operacionais
740:
741:  VerErros2 = False
```

```
742:
743:  ' verifica impossibilidades operacionais
744:  If ((Vs=1) And (B=1)) Or ((Ea=0) And (B=1)) Or ((Ea=1) And (Vs=1)) Or ((Fa=1) And (Vs=0) And _
745:  (B=0)) Or ((Fa=1) And (Vs=0) And (Eb=0)) Or ((Fa=1) And (Vs=0) And (Bb=1)) Then
746:    VerErros2 = True
747:  End If
748: End Function
749: '
750: Sub Pisca(ByVal Modo As Byte, ByVal N As Integer)
751: 'Rotina para acionar os leds.
752: 'Modo = 1 Pisca só Led1, 2 Pisca só Led1 com Led2 ligado, 3 Pisca só Led2,
753: '4 Pisca só Led2 com Led1 Ligado, 5 Pisca os 2 juntos, 6 Pisca os dois alternados
754: 'N - Número de piscadas
755:
756:  Dim I As Integer  ' Contador
757:
758:  Select Case Modo
759:  Case 1
760:    For I = 1 To N
761:      PutPin Led1, 1  ' Liga Led1
762:      Delay 0.5
763:      PutPin Led1, 0  ' Desliga Led1
764:      Delay 0.5
765:    Next
766:  Case 2
767:    PutPin Led2, 1  ' Liga Led2
768:    For I = 1 To N
769:      PutPin Led1, 1  ' Liga Led1
770:      Delay 0.5
771:      PutPin Led1, 0  ' Desliga Led1
772:      Delay 0.5
773:    Next
774:    PutPin Led2, 0  ' Desliga Led2
775:  Case 3
776:    For I = 1 To N
777:      PutPin Led2, 1  ' Liga Led2
778:      Delay 0.5
779:      PutPin Led2, 0  ' Desliga Led2
780:      Delay 0.5
```

```
781:     Next
782:   Case 4
783:     PutPin Led1, 1 ' Liga Led1
784:     For I = 1 To N
785:       PutPin Led2, 1 ' Liga Led2
786:       Delay 0.5
787:       PutPin Led2, 0 ' Desliga Led2
788:       Delay 0.5
789:     Next
790:     PutPin Led1, 0 ' Desliga Led1
791:   Case 5
792:     For I = 1 To N
793:       PutPin Led1, 1 ' Liga Led1
794:       PutPin Led2, 1 ' Liga Led2
795:       Delay 0.5
796:       PutPin Led1, 0 ' Desliga Led1
797:       PutPin Led2, 0 ' Desliga Led2
798:       Delay 0.5
799:     Next
800:   Case 6
801:     For I = 1 To N
802:       PutPin Led1, 1 ' Liga Led1
803:       Delay 0.5
804:       PutPin Led1, 0 ' Desliga Led1
805:       Delay 0.5
806:       PutPin Led2, 1 ' Liga Led2
807:       Delay 0.5
808:       PutPin Led2, 0 ' Desliga Led2
809:       Delay 0.5
810:     Next
811:   Case 7
812:     For I = 1 To N
813:       PutPin bit0, 1 ' Liga L5
814:       Delay 0.5
815:       PutPin bit0, 0 ' Desliga L5
816:       Delay 0.5
817:     Next
818:   Case 8
819:     For I = 1 To N
```

```
820:         PutPin bit1, 1  ' Liga L4
821:         Delay 0.5
822:         PutPin bit1, 0  ' Desliga L4
823:         Delay 0.5
824:     Next
825: End Select
826: End Sub
827: '
828: Sub LigaTudo()
829: 'Rotina para ligar todos os leds e o alarme
830:
831: PutPin Al, 1
832: PutPin Led1, 1
833: PutPin Led2, 1
834: PutPin bit2, 1
835: PutPin bit1, 1
836: PutPin bit0, 1
837: End Sub
838: '
839: Sub DesligaAlarme()
840: 'Rotina para verificar se é para desligar o alarme
841:
842: Dim X As Byte
843:
844: PutPin S4, 3  ' Pino S4 como entrada
845: X = GetPin (S4)
846: If X = 0 Then
847:     PutPin Al, 0  ' Desliga alarme
848: End If
849: End Sub
850: '
851: Sub MostraStatus(ByVal Id As Byte)
852: 'Rotina para verificar se é para mostrar Status
853:
854: Dim X As Byte
855:
856: PutPin S2, 3  ' Pino S2 como entrada
857: X = GetPin (S2)
858: If X = 0 Then
```

```
859:     Mostra(Id)  ' Mostra Status
860:   End If
861: End Sub
862: '
863: Sub DesligaTudo()
864: 'Rotina para desligar todos os leds e o alarme
865:
866:   PutPin Al, 0  ' Desliga alarme
867:   PutPin Led1, 0  ' Desliga Led1
868:   PutPin Led2, 0  ' Desliga Led2
869:   PutPin bit0, 0  ' Desliga Led5
870:   PutPin bit1, 0  ' Desliga Led4
871:   PutPin bit2, 0  ' Desliga Led3
872: End Sub
873: '
874: Function Sai() As Boolean
875: 'Rotina para verificar se S1 foi apertado
876:
877:   Dim X As Byte
878:
879:   PutPin 13, 3  ' Pino 13 como entrada
880:   X = GetPin (13)
881:   If X = 0 Then
882:     Sai = True
883:   Else
884:     Sai = False
885:   End If
886: End Function
887: '
888:
```